Some Investigations
of
Spatial and Temporal Constraint Satisfaction Problems

Submitted by

Priti Chandra

In Partial Fulfillment of the

Requirement for the Degree of

Doctor of philosophy, Computer science

Department of Computer and Information Sciences

School of Mathematics and Computer & Information Sciences

University of Hyderabad

Hyderabad, India

March 2007

Department of Computer and Information Sciences
School of Mathematics and Computer & Information Sciences
University of Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that the thesis work entitled "**Some Investigations of Spatial and Temporal Constraint Satisfaction Problems** " being submitted by Priti Chandra in partial fulfillment of the requirement for the award of degree of Doctor of Philosophy (Computer Science) of the University of Hyderabad, is a record of bona fide work carried out by her under my supervision.

The matter embodied in this thesis has not been submitted for the award of any other research degree.

Prof. Arun K. Pujari
Supervisor

Head
Department of Computer and Information Sciences
University of Hyderabad,
Hyderabad, India

Dean
School of Mathematics and Computer & Information Sciences
University of Hyderabad,
Hyderabad, India

DEDICATION

To daddy, mummy

and

chachaji

who are very far away from me,

but watching me fulfill their dream

SELF – DECLARATION

I, Priti Chandra, hereby declare that the work presented in this thesis has been carried out by me under the guidance of Prof. Arun K. Pujari, Department of Computer and Information Sciences, University of Hyderabad, as per the PhD ordinances of the university. I declare, to the best of my knowledge, that no part of this thesis has been submitted for the award of a research degree of any other university.

(**Priti Chandra**)

ACKNOWLEDGMENTS

At the outset I would like to thank immensely my supervisor Prof. Arun K Pujari, Professor, DCIS, University of Hyderabad for suggesting the research problem and for the guidance and suggestions all through the period. There are many people whom I would like to thank for their support throughout my research work. I am thankful to Prof. Arun Agarwal, Head, Department of Computer and Information Sciences for showing me the strong pillars on which the university system stands. He has helped me on many occasions. I am thankful to Prof. B L Deekshatulu, visiting professor, Department of Computer and Information Sciences for his facilitation towards the ending stage of my work.

I am grateful to Defence Research and Development Organization (DRDO) for giving me an opportunity to do my research work with a support of three years of study leave from March-2003 till Feb-2006. I am thankful to Shri N Diwakar, Ex-Director, DLRL for helping me in sanctioning of the leave. I am thankful to Dr. R Sreehari Rao, Director, DLRL for extending his support beyond three years that has helped me to complete this work. I am thankful to Shri S K Gupta, Scientist ‘G’, DLRL for his endless belief in my sincerity, commitment and hard work that has helped me sail across confidently.

I am deeply indebted to the support of my family. This thesis is dedicated to my uncle and my parents from whom I have inherited the qualities of sincerity, truthfulness, commitment and hard work in all walks of life. I am grateful to my husband Murali who is a good advisor and always encouraging and pushing me forward towards the work. I am happy that I have always withstood his busy schedules. I express my thanks to my lovely kids – Soujanya and Nikhil. My son was ten months old when I joined for PhD. My daughter was always inspired to come and study next to me whenever she saw me doing my work at home. I am grateful to Parvathi who has been a strong pillar at home as a continuous support for my kids and having absorbed all my moods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The present thesis investigates tractability in qualitative spatial and temporal frameworks. The focus of the thesis is on Region Connection Calculus (RCC-8) and Interval algebra(IA) networks.

We present a new lattice theoretic formalism for tractability study of qualitative constraint satisfaction problems. We assume strong composition in this study. We propose a domain of objects for which RCC-8 CT is complete and extensional. The proposed formalism helps in supporting theoretical analysis of the existing results and in obtaining new results on tractability for RCC-8. We show that the existence of the least elements of partially ordered sets plays a very important role in identifying tractable sets. This property can also be used to determine consistent instantiation of the constraint network. The present study demonstrates that various tractable sets can be determined and efficient algorithms can be devised using this characterization. We provide theoretical proofs of the several results that rested on machine-assisted proofs so far. We introduce three new tractable classes namely convex class, preconvex class, and $DPConv_8$. We show that any path consistent convex RCC-8 network is minimal. To the best of our knowledge, there is no known class in RCC-8 for which path consistency implies minimality. The partial order together with the dimension of RCC-8 relations can be used to identify the preconvex class. We show that coarsening as a reverse process of refinement determines a new tractable class namely, $DPConv_8$. We propose an algorithm based on the least element instantiation that is more efficient than the existing algorithms for finding consistent singleton label. As a consequence, for any class admitting the existence of least element, it can be shown that path consistency suffices for consistency and hence consistency can be determined in polynomial time. The same result is extended to set of relations J_8 that is not a closed class but path consistency decides consistency.

For interval algebra framework, we analytically answer the question "What makes an IA instance easier?" The least element based interpretation for RCC-8 is tailored to give a new description of the known tractable classes based on high set. We introduce a property of high set to analyze the tractability properties of a class. We prove that the known tractable classes exhibit this property and hence path consistency becomes sufficient condition for existence of a solution. We also establish here that for these classes, we can get a solution by a method called '*highset instantiation*'. We provide a simple analytical proof for all the tractability

results in interval algebra reported so far by other researchers using a number of complex methods. This property is extended to find a consistent instantiation for path consistent tractable networks without any computational overhead of enforcing path consistency intermittently.

We are able to answer the question "What makes an IA instance hard to solve? "for general IA networks. We extend the existing entropy-based interpretation in the weighted IA network paradigm. We investigate the reasons for the incompleteness of the existing highest weight heuristic. We propose a measure of hardness of problem instances for interval algebra networks. Our investigations reveal that the constraints with entropy fluctuations termed as *nasty constraints* are the reason for instance hardness. We provide a theoretical justification for the reason of fluctuation of entropy in IA networks. We show here that the known tractable classes of interval algebra do not exhibit the property of entropy fluctuations due to the high set property. We provide different techniques to relax the nasty constraints so that number of conflicts reduce as the entropy of each constraint stabilizes over the search trajectory. This controls the path that the search trajectory takes resulting into an approximate solution of any instance of general IA networks. This study identifies the nasty constraints in an intelligent manner with the help of the conflicts. The novel method of conflicts proposed here lays a foundation for a new direction towards addressing hardness of any qualitative binary constraint satisfaction problem. We are successful to propose an algorithm that computes an approximate solution on completion of the weighted path consistency algorithm. We show that the inconsistent IA networks can be handled by this algorithm as routine cases of nasty constraints with trivial conflicts.

Lastly we propose a framework to study structure of spatial and temporal networks that represent a scene in the dynamic domain of road traffic. We exploit this structure to predict next scene based on the current scene. We demonstrate benefit of complete instantiation strategy based on the neighborhood property of spatial relations. We investigate different ways in which occlusion can take place. We are able to distinguish between the case of occlusion and an object leaving a scene. This approach is a step away from the conventional backtrack-based approach for an application based on a constraint satisfaction problem.

CHAPTER 1

QUALITATIVE REPRESENTATION AND REASONING

1.1 Introduction

In this chapter, we provide the background that concisely covers the basic fundamentals and the state of research in the area of qualitative representation and reasoning. The discussion is limited to only two frameworks that are the center of this study.

1.2 Binary Constraint Satisfaction Problems

Knowledge about entities or about the relationships between entities is often given in the form of *constraints*. For instance, when trying to place furniture in a room there are certain constraints on the position of the objects. Unary constraints such as "The room is 10 meters in length and 20 meters in width" restrict the domain of single variables, the length and the breadth of the room. Binary constraints like "The desk should be placed in front of the window", ternary constraints like "The table should be placed between the sofa and the armchair", or in general n-ary constraints restrict the domain of 2, 3 or *n* variables. Problems like these are formalized as Constraint Satisfaction Problems (CSP).

Definition 1[Tsang, 1993]: A *Constraint Satisfaction Problem*(CSP) is defined as a triplet $<V,D,C>$ where *V* is a set of variables, *D* is a domain for the set of possible values for each variable in *V*, *C* is a set of constraints restricting variable values.

In this work, we restrict ourselves to binary CSPs i.e. CSPs where only binary constraints are used. Constraints can be processed by propagating the domain restrictions from one variable to others. If the domain of variables is infinite, conventional method of backtracking to solve a CSP is not very helpful and other methods have to be applied. One way of dealing with infinite domains is using constraints over a finite set of binary relations. Binary CSPs are formulated [Ladkin and Maddux, 1994] as relation algebras [Tarski, 1941. This allows treating binary CSPs with finite and infinite domains in a uniform way. The main difference of spatial or temporal CSPs to normal CSPs is that the domains of spatial and temporal variables are usually infinite. For instance, there are infinitely many time points or temporal intervals on the time line and infinitely many regions in two or three-dimensional space. Hence it is not feasible to represent relations as sets of tuples nor is it feasible to apply algorithms that enumerate values of the domains. Instead relations can be used as symbols and reasoning has to be done by manipulating symbols. This implies that the calculus, which deals with extensional relations in the finite case, becomes intensional in the sense that it manipulates symbols, which stand for infinite relations. Two well-known approaches to represent knowledge are quantitative and qualitative approaches with respective advantages. In this study we restrict investigation to only qualitative representation that is closer to human like thinking. Usual way of dealing with qualitative spatial and temporal reasoning is to partition the domain *D* into a family of non-empty binary relations.

1.2.1 Binary Constraints and Operations

Definition 2: A *binary relation R*, consists of a set *A*, called the domain of *R*, a set *B* called the codomain of *R* and a subset of $A \times B$ called the *graph* of *R*.

A relation whose domain is *A*, codomain is *B* is said to be "between *A* and *B*" or "from *A* to *B*". When the domain and the codomain are the same set *A*, we simply say that the relation is "on *A*". It is common to use infix notation "*a R b*" to mean that the pair (*a*,*b*) is in the graph of *R*. A relation consists of a set of binary relations *R* that is closed under the operations of union ($\cup$), intersection ($\cap$) and composition (o).

Definition 3: A set *J* of binary relations on a non-empty domain set *D* is said to be *jointly exhaustive and pair-wise disjoint* (JEPD) if for every pair $(a, b) \in D \times D$ there is exactly one relation $R \in J$ such that *a R b* is true.

If *J* is the set of JEPD relations on *R*, we have a function that maps *D* X *D* to *J*, taking each (*a*, *b*) to the unique relation to which it belongs. In other words the JEPD set of relations is exactly same as partition of *D* X *D*. Among all possible binary relations, the partition selects a finite subset of binary relations. Each possible tuple (*a*, *b*) in *D* X *D* is contained in exactly one relation $R \in J$. Since any two entities are related by exactly one of the atomic relations, they are used to represent definite knowledge. Indefinite knowledge can be specified by disjunctions of possible atomic relations. We denote the set of JEPD relations as atomic relations. The universal relation ***U*** is the disjunction of all the atomic relations. Converse, complement, intersection and disjunction of relations can be obtained by performing the corresponding set theoretic operations. We formally define two basic binary operations of converse and composition on binary relations in the powerset, $\boldsymbol{P} = 2^J$.

Definition 4: If a relation R holds between two variables a and b, then the *converse* of R, denoted by R^{-1}, is a relation that holds between b and a. Converse of any non-atomic relation $R \in \mathbf{P}$ is defined as $R^{-1} = \{r^{-1}: r \in R\}$.

Definition 5: Let $R \in \mathbf{P}$ be a binary relation between variables a and b and $S \in \mathbf{P}$ be a relation between variables b and c, we define *composition* of R and S as a relation between a and c.

The composition of two disjunctive binary relations R and S is obtained as the union of pair-wise compositions of the atomic relations, denoted by $R \text{ o } S = \cup\{r_i \text{ o } s_j\text{: } r_i \in R \text{ and } s_j \in S\}$.

Definition 6 [Bennett et al. 1997]: A *composition table* is defined as a mapping

$$CT: J \times J \rightarrow 2^J$$

i.e. for each ordered pair $<R, S>$ CT specifies $T = CT(R, S)$ where $R, S \in J$ and $T \subseteq 2^J$, called the composition of R and S.

The elements of the set J in above definition are relations that form a JEPD partition of possible relations, which can hold between pairs of objects in the domain under consideration. Any pair of objects is related by exactly one member of J. The set S is the strongest possible disjunction of relations in J. Although the compositions may be hard to derive, once established they may be stored once and for all in a composition table (CT).

Definition 7: A *triad* is defined as a set of any three relations denoted as $<R,T,S>$.

A triad can consist of any type of relations. We denote a triad consisting of only the atomic relations as (s_1, s_2, s_3). When the relation(s) in a triad is (are) disjunctive relation(s), we term the triad as a non-atomic triad. Each cell entry in the composition table can be interpreted as a triad. For instance, composing an atomic relation along a row (s_1) with an atomic relation in a column (s_2) gives a relation (s_1 o s_2) at the corresponding cell entry. This can be represented as CT[s_1][s_2] where CT stands for the two dimensional matrix that holds the basic composition table and s_1, s_2 are the row-wise and column-wise indices respectively. The fundamental mode of reasoning encoded in a CT is to test consistency of triads of relations of the form $R(a, b)$, $S(b, c)$ and $T(a, c)$ where R, S and $T \in J$. We denote a triad to be *CT-consistent* iff $T \in \mathrm{CT}(R, S)$.

1.2.2 Qualitative Spatial and Temporal Reasoning Frameworks

Temporal and spatial domains have been widely addressed in the area of qualitative reasoning. Numerous frameworks have been proposed till date addressing various aspects of expressiveness and computational complexity- Point Algebra [Vilain, Kautz and van Beek 1989] for time events, Allen's algebra [Allen 1983] for time intervals and INDU [Pujari et al. 1999] for interval durations. Spatial domain has a relatively rich structure consisting of topology, size, distance, shape and direction. Qualitative spatial reasoning has also received major attention due to increasing need of real-world applications, namely GIS [Bennett 1996], robot navigation, medical imaging etc. Spatial formalisms include RCC [Randell, Cui and Cohn 1992] for topology, Cardinal algebra for direction [Ligozat 1998], Rectangle algebra, n-block algebra [Balbiani et al. 1999a, b] and 9-intersection model [Egenhofer 1991].

These formalisms of qualitative reasoning for spatial and temporal domains fall into one general framework. Each of these provides a set of jointly exhaustive and pair wise disjoint (JEPD) binary relations and represents the spatial and temporal knowledge as a network of binary constraints. These differ among themselves on expressiveness and the domain of operation (spatial or temporal). Notwithstanding these differences, they exhibit certain common characteristics. A major reasoning problem in this framework has been to decide satisfiability of given information that is expressed in terms of a set of variables and a set of disjunctions of JEPD relations as binary constraints. A widely employed method has been to formulate it as a constraint satisfaction problem (CSP). Although testing satisfiability for most of these formalisms is intractable, in general, there are many fragments for which polynomial time algorithms exist.

1.2.2.1 RCC-8

Region connection calculus (RCC)[Randell, Cui and Cohn 1992] is a topological approach to qualitative spatial representation and reasoning where spatial regions are non-empty regular open subsets of a topological space [Munkres, 1999]. Binary relationships between a pair of spatial regions *a* and *b* are defined in terms of connectedness relation written as *C*(*a*,*b*). In the standard interpretation of the RCC theory, *C*(*a*,*b*) is true if and only if the closure of region [Willard, 1970] *a* is connected to the closure of region *b* i.e. if the closures of the two regions share a common point. Regions themselves do not have to be internally connected, i.e. a region may consist of different disconnected parts and regions are allowed to have holes.

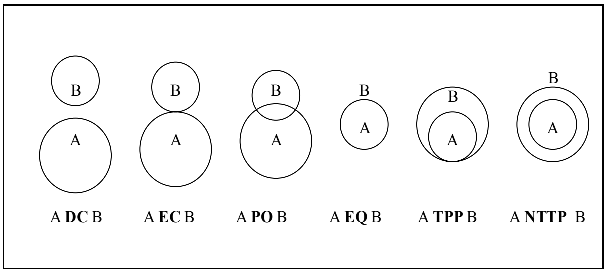


Figure 1. 2D illustrations of relations of RCC-8.

RCC-8 [Randell, Cui and Cohn 1992] is a set of eight JEPD relations definable in RCC theory. The topological relations between any two spatial objects can be described by exactly one of the eight relations of the set **U** = {DC, EC, PO, TPP, NTPP, TPPI, NTPPI, EQ}. The atomic relations have the meaning of *DisConnected*, *Externally Connected*, *Partial Overlap*, *EQual*, *Tangential Proper Part*, *Non-Tangential Proper Part* and their converses. Examples for these eight atomic relations are depicted in Figure 1. The disjunction of these JEPD relations represents indefiniteness of topological relations with one of the disjuncts being true at any instance. The power set ***P*** contains 2^8 possible binary topological relations. RCC-8 CT is included in Appendix – B.

1.2.2.2 Egenhofer's Approach to Topological Spatial Relations

Independently of the Region Connection Calculus a different approach to topological spatial relationship was developed by Egenhofer in the area of geographical information systems (GIS). Relationships between two spatial entities are classified according to nine possible intersections of their interior, exterior and boundary denoted as the 9-intersection model [Egenhofer, 1991]. Depending on the nature of the considered spatial entities, many different relationships can be expressed by this model. For instance, it is possible to use spatial entities of different dimensions or distinguish different degrees of intersection. In his first approach, however, Egenhofer restricted the domain of spatial entities to two-dimensional spatial regions, whose boundary is a closed Jordan curve, i.e. simply connected planar regions that are not allowed to have holes. When looking at only whether the nine intersections of this kind of spatial regions are empty or non-empty and when eliminating all impossible relations, this results in eight different binary topological relationships. The Region Connection Calculus and the 9-intersection model, which are two completely different approaches to topological relationships, lead to exactly (apart from the different constraints on regions) the same set of topological relations. Thus, there seems to be natural agreement about what is a reasonable level of granularity of topological relations.

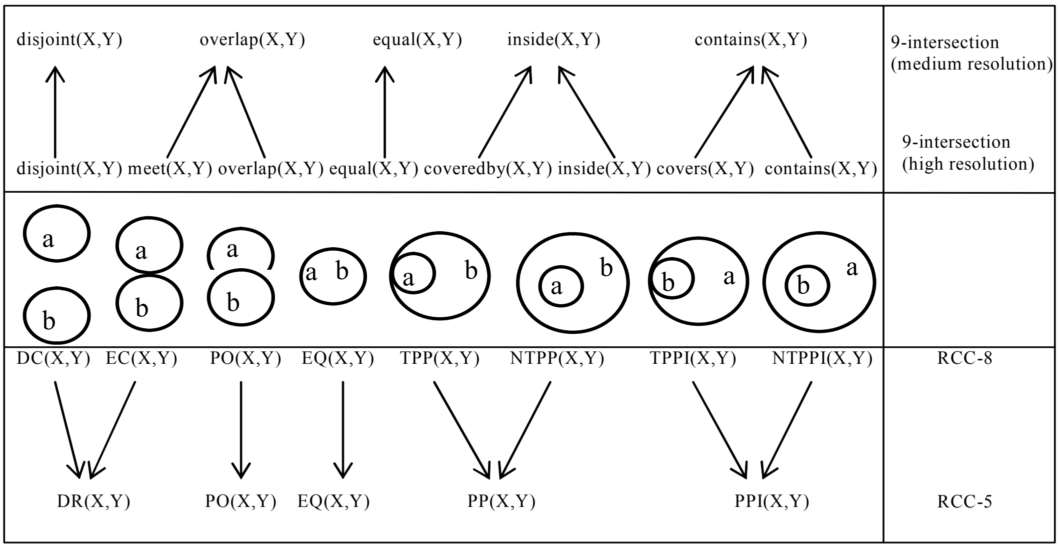


Figure 2. Comparison of different systems of topological relations.

The computational properties of Egenhofer's topological relations were studied [Grigni et al., 1995] for two different notions of satisfiability - the purely syntactical notion of relational consistency and the semantic notion of realizability, which are both different from what we call consistency. Relational consistency means that there is a path consistent refinement of all relations to base relations, realizability means that there is a model consisting of simply connected planar regions. Orthogonal to this distinction, Grigni et al. consider different set of JEPD relations. One of them is the original set of eight base relations suggested by Egenhofer, which they called the *high-resolution* case. Another set of five base relations, the *medium resolution* cases, which are obtained by combining some of the high-resolution relations. Similar to RCC-5[Bennett, 1994a], the medium resolution relations do not distinguish relationships according to the boundary of regions. The difference of this set to RCC-5 is that the relations EC and PO are combined to form a new base relation, whereas for RCC-5 the relations DC and EC are combined as shown in Figure 2. With this distinction, the medium resolution relations can also be used to represent the possible relationships between non-topological sets: *overlap*(*a*, *b*) means that the sets *a* and *b* have a non-empty intersection while none of them is a subset of the other, if *disjoint*(*a*, *b*) holds, the sets *a* and *b* have no elements in common. This set-theoretic interpretation is not possible for RCC-5, since it distinguishes between interior and boundary elements. If two sets share some boundary elements, they are still in the DR relationship.

By reducing the NP-hard string-graph problem [Grigni et al., 1995] show that deciding realizability is NP-hard for the high and medium resolution cases even if only constraints over the base relations are used. It is an open problem whether the realizability problem is in NP and even whether it is decidable. The reduction of the string graph problem, however, is possible only because all regions must be simply connected planar regions. Hence, this result does not carry over to the consistency problem of RCC-8 where spatial regions can be of any dimension and internal connectedness is not required. The relational consistency problem, which is obviously tractable only if base relations are used, was shown to be NP-hard for the high and medium resolution cases if all disjunctions over base relations are permitted.

1.2.2.3 Interval Algebra

Interval Algebra(IA) is an approach of qualitative temporal representation and reasoning based on the notion of time intervals [Allen, 1983]. A time interval is an ordered pair (X^-, X^+) such that $X^- < X^+$ where X^-, X^+ are interpreted as points on the real line. Given any two concrete time intervals, their relative position on the real line can be described by exactly one of the thirteen atomic relations in the universal set **A** = {*b*, *m*, *o*, *s*, *d*, *fi*, *eq*, *f*, *di*, *si*, *oi*, *mi*, *bi*}.

Relation	Symbol	Inverse
x before y	b	bi
x meets y	m	mi
x overlaps y	o	oi
x starts y	s	si
x during y	d	di
x finishes y	f	fi
x equal y	eq	eq

Meaning

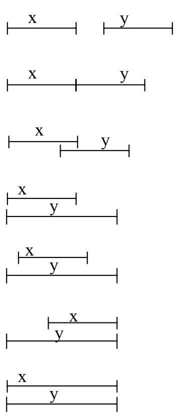


Figure 3. Interval Algebra atomic relations between time intervals.

The atomic relations have the meaning of *b*efore, *m*eet, *o*verlap, *s*tart, *d*uring, *f*inish, *eq* and their converses. The examples of these atomic relations are depicted in Figure 3. The disjunctions of these atomic relations represent the indefinite information. The power set contains 2^{13} possible unions of atomic relations that form the set of binary interval relations, including the empty relation ∅ and the universal relation A. IA CT is included in Appendix – B.

1.2.2.4 Point Algebra

Point Algebra(PA) is a qualitative temporal representation and reasoning framework based on the notion of time points [Vilain and Kautz, 86]. A time point X is a point on the real time line. Given a pair of time points, their relative position on the time line can be described by exactly one of the three atomic relations PA = { <, =, > }. The atomic relations have the meaning of less than, equal, greater than. In order to represent indefinite information, the relation between two points is a disjunction of the atomic relations. The set of possible relations between two points is {∅, <, ≤, =, >, ≥, ≠, ?}, where ? denotes the universal relation **PA**.

1.2.3 Binary Constraint Networks based Reasoning Tasks

Most of the reasoning tasks associated with CSPs are, in general, computationally intractable (NP-Hard), which means that it is not possible to design algorithms that scale efficiently with the problem size. Researchers have focused on developing and improving the performance of general algorithms for solving CSPs, on identifying restricted tractable subclasses that can be solved efficiently, and on developing approximation algorithms. CSPs can be modeled as constraint networks.

Definition 8**:** A *binary constraint* $R_{ij} \in \mathbf{P}$ between the variables A_i and A_j is a disjunction of atomic relations.

Definition 9: A *network of binary constraints* is defined as a directed graph where nodes represent the variables and edges represent the binary constraints between variables.
In the present context, we denote *RCC-8 network* as a binary spatial constraint network consisting of a set of *n* spatial variables $A_0, A_1, \ldots, A_{n-1}$ and a set of binary constraints between

these variables. The spatial variables represent the spatial entities and the domains of the variables are the set of all spatial regions of the underlying topological space.

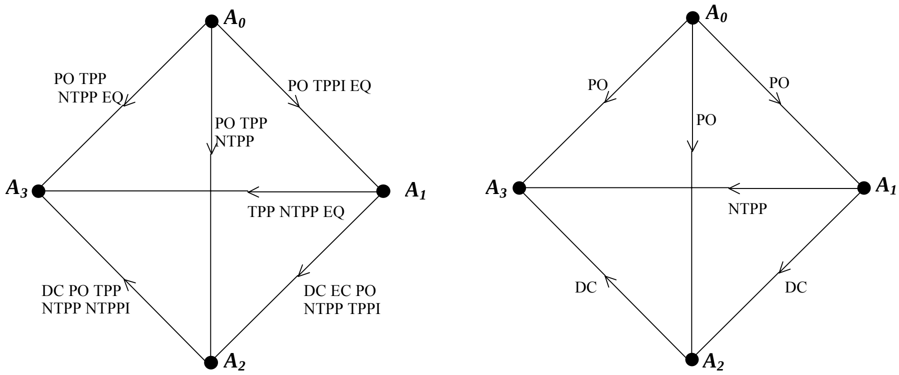
Definition 10: An *instantiation* of a network is an assignment of variables to entities in the respective domain.

An instantiation is said to be *consistent* if entities satisfy all the constraints. Unlike for the CSP with finite domain, for the spatial and temporal CSPs where domain is infinite, there can be infinitely possible instantiations. Hence the consistency property of the network is studied not by instantiating variables but by resolving (asserting) the disjunctions. A binary constraint with a single disjunct is called a *singleton*.

Definition 11: A *singleton labeling* of a binary constraint network assigns to each pair of variables A_i *and* A_j an atomic relation s such that $s \in R_{ij}$.

In the present context of RCC-8 CT, it is worth segregating the two concepts – consistency and solution of a constraint networks. We formally define these concepts.

Definition 12: A given constraint network N is said to be *consistent* if there exists a singleton network S which is consistent with respect to the composition table and $R_{ij}(S) \subseteq R_{ij}(N)$.



(a) Disjunctive network N **(b) Singleton network S**

Figure 4. Example of disjunctive network N and singleton network S.

This notion of consistency purely depends on the composition table irrespective of the properties of the underlying domain. S may be derived from N with the help of any of the solution techniques namely backtrack (or any of its variants), local search. A given binary constraint network N may consist of any number of singleton networks S.

Definition 13: A *solution* of a given constraint network *N* is an instantiation of all its variables in the respective domain such that all the constraints are satisfied.
An inconsistent constraint network does not have a solution, since it is not CT-consistent. The concept of solution of a network means two steps. First – compute *S* from *N*. Second – Start arranging the objects in the given domain as per the singleton relations in *S* till we completely realize *S*. For instance, the solution for an example network *S* is the realization as shown in the Figure 5.

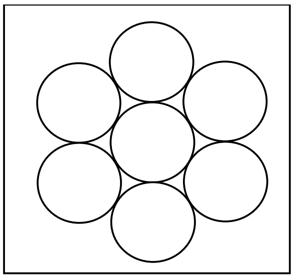


Figure 5. Example S with a realization in domain of equal sized discs.

The two concepts - consistency of a network and solution of a network do not always mean the same. This difference/coincidence depends on the properties of the composition table and the underlying domain. In the case of interval algebra, a consistent disjunctive network will always have a solution. On the other hand, in case of consistent RCC-8 networks, a solution may not always be possible. For instance, a realization of an all-EC network is not possible in the domain of equal sized discs with 7 nodes as shown in Figure 5.

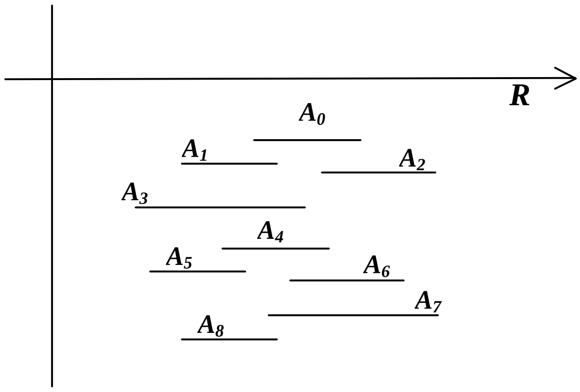


Figure 6. Realization of 9-node (all-EC) network in domain of 1-D time intervals.

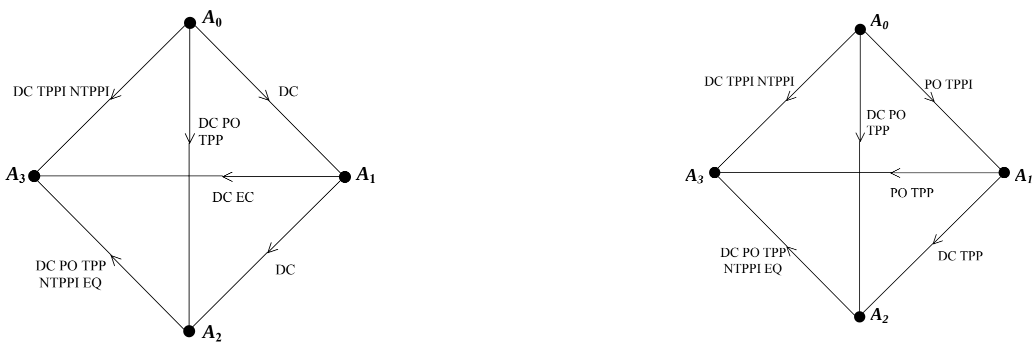
On the contrary, in the domain of one-dimensional time intervals, a realization is possible for any size of the network as shown in the Figure 6. Due to the coincidence of concepts of solution and instantiation in interval algebra solutions of an IA network are its consistent singleton labeling.

Definition 14: An atomic relation $s \in R_{ij}$ is feasible for the pair of variables (A_i, A_j) if and only if there exists one solution where *s* is assigned to this pair.

Definition 15: A *minimal constraint* between A_i and A_j is the set (disjunction) of feasible relations between A_i and A_j.

Definition 16: A *minimal network* is a constraint network where each edge has a minimal constraint.

Figure 7(a) gives an example of a minimal RCC-8 network and the one shown in Figure 7(b) is an example of a RCC-8 network that is not minimal.



(a). Minimal RCC-8 network

(b). Non-minimal RCC-8 network with EQ in R_{23} as the non-feasible relation

Figure 7. Examples of minimal and non-minimal RCC-8 networks.

1.3 Properties of Composition Table in RCC-8 and Interval Algebra

In literature, two important properties of a composition table are studied, namely *completeness* and *extensionality*. We report the related definitions and the state of the art of research in recent times on these two aspects, for the sake of the required background.

Definition 17 [Sanjiang, 2003]: A *model* of <*J*, CT> is a pair <D,v>, where D is the domain set and v is a mapping from *J* to the set of binary relations on D such that $\{v(R): R \in J\}$ is a partition of D × D and $v(R) \circ v(S) \subseteq \cup_{T \in CT(R,S)} v(T)$ $\forall R, S \in J$, where o is usual relational composition.

This means that the composition cell entry may contain some more atomic relations besides that resulting due to the relational composition of the two atomic relations. The condition that is commonly required for CT based reasoning is that the resulting relations(s) must be present. This is commonly termed as the weak composition in the literature[Düntsch, 1999]. This consistency-based definition of composition is given below.

Definition 18 [Sanjiang, 2003]: A model is *consistent* if $T \in \text{CT}(R,S)$ iff ($v(R) \circ v(S)$) $\cap$ $v(T) \neq \emptyset$ $\forall$ where $R, S, T \in J$.

This means that for any three relation symbols *T*, *R* and *S*, *T* is an entry of the cell specified by *R* and *S* if and only if there exist three regions a, b, c in D such that *R*(a,b), *S*(b,c) and *T*(a,c) hold. Thus as a thumb rule of checking consistency of a given triad (*R,T,S*), we need to check if *T* is present in the CT entry specified by *R* and *S*. This helps to check only the transitive relation to exist between objects a,c when the pair wise relations are given between a,b and b,c. For example, the triad (EC, EC, EC) is consistent since EC o EC = { DC EC PO TPP TPPI EQ } and EC $\in$ EC o EC. The triad (EC, NTPP, EC) is not consistent since NTPP $\notin$ EC o EC.

Definition 19: A consistent model is *weak* if $v(R) \circ v(S) \subset \cup_{T\in \text{CT}(R,S)} v(T)$ for all $R, S \in J$.

We illustrate this concept with the help a diagram as shown in the figure 8(a). Suppose the composition of *R* and *S* with respect to the CT results into the relations $\{T_1, T_2, T_3, T_4, T_5, T_6\}$ shown as partition of six relations. The relational composition on the other hand results into the relation $\{T_3, T_4\}$ shown as the subset of CT(R,S) in the diagram.

a. Weak Composition: $\text{CT}(R,S) \supset R \circ S$ b. Strong Composition: $\text{CT}(R,S) = R \circ S$

Figure 8. Partition of binary relations due to weak and strong composition.

1.3.1 Extensionality

Definition 20: A consistent model is *extensional* if v(R) o v(S) = $\cup_{T\in CT(R,S)}$ v(T) $\forall R, S \in J$. This means that the true relational composition of two relations is also that in the weak composition entry in the composition table. The binary relations resulting due to relational composition are the same that resulting due to the CT based composition as shown in Figure 8(b) above. In other words, a possibility of an atomic relation between a and c results due to the composition operator only when there exists a direct relation between a R b and b S c. In such a model, if T is an entry in the cell specified by R and S, then whenever $T(a,c)$ holds, there must exist some b in U such that $R(a,b)$ and $S(b,c)$. Under this circumstance, the model is extensional if the weak composition coincides with the relational composition. This purely extensional definition is strictly stronger that the consistency-based definition. In weak interpretation, we only impose that whenever the interpretation of R holds between two objects a and b, and the interpretation of S holds between b and c, then for some T in (R o S), the interpretation of T holds between a and c. The extensional definition is stronger and the converse must also be true- for any entry in the composition table, if the interpretation of T holds between a and c, where $T \in$ (R o S), if and only if there exists some b in the domain such that R holds between a and b, and S holds between b and c. If a weak composition table is extensional, then as well as providing a means for consistency checking, it can be employed to justify a certain kind of extensional inference: from a relation holding between two objects we deduce existence of a third object related to the original two in a specific way.

A method based on the notion of weak composition may imply consistency of a given network, whereas actually a solution may not exist. As an example, an all-EC network of size 10 is consistent as per the CT but a solution does not exist in the domain of equal sized discs since it is not possible to place more than 7 disks that satisfy all the constraints. In other cases, due to weak composition operator, the method may imply a solution that is different from an actual one. Such a possibility arises when the domain violates the definition of extensionality above. This means that there exists a direct relation between two objects, but it is not possible to place a third object such that an indirect relation holds. Hence the strong composition operator inherently does not limit one only to the satisfaction of given constraints, but forces one to extend a constraint to a third object. This helps in arriving at a physical realization in case of consistent singleton networks.

The RCC-8 composition table (CT) first appeared in [Cui et al. 1993] and is investigated in detail in [Bennett 1994b, 1997, 1998b], [Düntsch1999a, Düntsch et al. 1999b, Düntsch et al. 2001b-c]. Certain peculiar properties are observed for this table that is not exhibited by other well-known qualitative reasoning framework, namely Allen's Interval Algebra(IA). The arguments of temporal relations are intervals and are easily interpreted as pairs of real numbers s and f. It is always possible to find infinitely many points between two given points on real line. If we restrict <s,f> to be integers, then we cannot always find integers between every pair. In such cases, we will not be able to always extend the relation between a pair of intervals to a third interval. It is observed that RCC-8 composition table is not extensional [Bennett et al. 1997]. We investigate the properties of a composition table – extensionality and completeness in the following paragraphs.

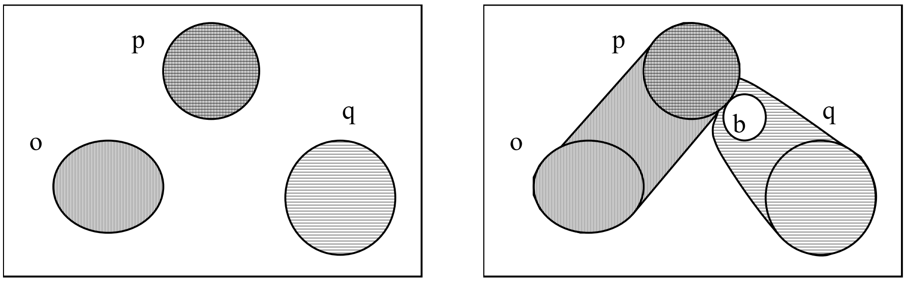
Definition 21: A given composition table (CT) is *extensional* with respect to a given domain D if the following condition is satisfied for the given objects a, b and c in D for all the possible triads $<R,T,S>$ in the CT:

$$\forall\ (a,c \in \mathrm{D})\ T(a,c) \rightarrow (\exists\ b \in D)\ [\ R(a,b) \text{ and } S(b,c)]$$

where $T \in \mathrm{CT}(R, S)$

In the above definition, the extensional interpretation of a triad $<R,T,S>$ means that whenever a relation T exists between any two arbitrarily selected objects in the domain D, it is always possible to place a third object b, such that $R(a,b)$ and $S(b,c)$ are satisfied. The concept of extensional interpretation ensures the basic requirement of consistency of the triad $<R,T,S>$ i.e. $T \in \mathrm{CT}(R,S)$. In addition to ensuring the consistency of the triad, this property ensures the realization of the networks of size three taken independently. The arrangement of objects for the triads that are only consistent is addressed by the concept of extensional composition. The realization of a triad that is inconsistent is in fact filtered out and any argument is avoided on this aspect. Any given triad is not extensional either if it is not consistent or besides being consistent, it is not possible to extend the relation between two objects to a third object. We illustrate such possibilities in RCC-8 with the help of peculiar characteristic properties of objects.

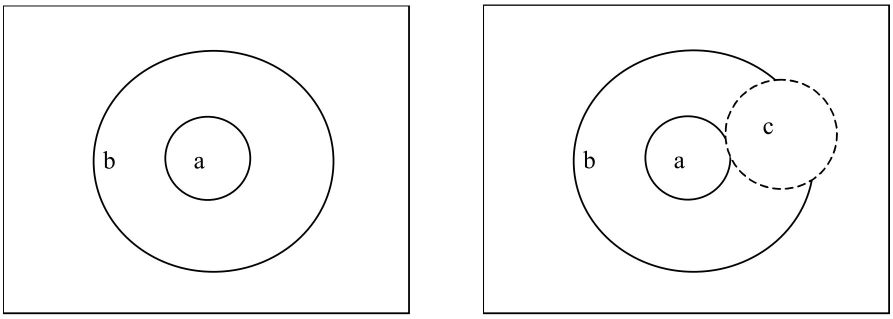
1. For instance, consider three disconnected regions o, p and q. Let $a = o \vee p$, $\mathrm{c} = p \vee q$ as shown in Figure 9.



a. Binary sum operation b. Topological closure

Figure 9. Interpretation of triad <EC, NTPP, PO>

This implies that PO(a,c). According to the RCC-8 CT, PO $\in$ CT(EC, NTPP) but it is easy to see that there is no region *b* such that EC(*a*,*b*) and NTPP(*b*,*c*) to hold at the same time. This suggests that PO $\not\subset$ EC *o* NTPP, where *o* is the true relational composition operator. However, the same becomes possible if we take the regions *a*, *c* as topological closures instead of just the union of the region pieces, denoted as *binary sum operation* [Sanjiang, 2003] as shown in Figure 9(b).



a. b.

Figure 10. Hole relation

2. Another possibility for violation of extensionality arises when a region *a*, which is a hole of another region *b* such that the region *b* completely surrounds *a*, denoted as *hole relation*[Sanjiang, 2003] as shown in Figure 10. For any third region *c* to touch *a*, the region *c* has to overlap *b* as shown in Figure 10(b). Thus any third object to externally touch *a* is not possible without overlapping *b*, since *a* is completely surrounded by *b*.

3. As long as *b* is an ordinary bounded region, regions *a* and *c* satisfying appropriate conditions can be found. But the domain of objects includes a universal region *u* such that every other region is a non-tangential proper part of *u*. For instance, if *a* is an ordinary region and *b* = *u*, then NTPP(*a,b*), but there cannot exist a region *c*, which is a tangential proper part of *b* since this region is unbounded as shown in Figure 11.

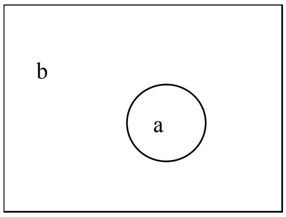


Figure 11. Unbounded region based non-extensionality.

It is suggested [Bennett, 1997] to remove *u* from the domain of possible regions for extensional interpretation. To summarize, three conditions are identified as necessary for a given domain to be extensional [Sanjiang, 2003].

Theorem 1: Necessary conditions for RCC8CT to be extensional for a domain *D* are:

a. Universal region (*u*) not allowed
b. Hole relation between any pair of regions not allowed
c. Objects with discrete components not allowed.

Proof: Let us assume that a given domain *D* is extensional. This implies that for every consistent triad <*R,T,S*> in the RCC-8 CT can be realized. Suppose that this domain does not satisfy at least one of the above three conditions.

a. Suppose the universal region *u* is allowed. Let *a* = *u*, it is true that no region can externally touch(EC) the region *u*. This means that the direct relation between *a* and *c* itself is not satisfied. Thus there cannot exist *R* and *S*, with *T* = EC such that *a R b*, *b S c* are satisfied. This violates the assumption that CT is extensional for D which includes the extensional interpretation of triads <*R,T,S*> where *T* = EC.
b. Suppose the hole relation is allowed in the domain *D*. Suppose Hole(*a,c*) is true i.e. the region *a* is the hole of region *c* such that *a* EC *c* holds true. We cannot place a third object *b* such that it externally touches both *a* and *c* i.e. <EC,EC,EC> though consistent but cannot be realized. This violates the extensional interpretation for *D*.
c. Suppose the domain *D* contains objects with discrete components. Let *a* and *c* be two regions sharing a component i.e. *a* PO *c*. We cannot place a third object *b* such that NTPP(*b,a*) and EC(*b,c*) holds. This implies that extensional interpretation for the triad <EC,PO,NTPP> does not exist, which again violates the assumption. ■

The discovery of this phenomenon- RCC-8 CT is not extensional- led to very interesting research in the recent years. The set of properties that are expected to hold do not hold due to non-extensional (or, weak) CT. Some researchers attempt to define a constrained domain such that the CT becomes extensional. When the domain of the regions is the set of closed discs, the CT is extensional [Düntsch 1999a]. The domain of connected regions bounded by Jordan curves also provides an extensional domain of the RCC-8 CT [Sanjiang et al.2003a].

Definition 22[Sanjiang, 2003b]: *Egenhofer Model*(*E*) is defined as the set of all simple regions bounded by a closed Jordan curve in the Euclidean plane.

Using 9-intersection method, we can identify same set of eight RCC relations. Based on transitivity set inclusion, the weak composition table of this domain is also derived [Egenhofer, 1991]. For any triad <R,T,S> to be extensional, it is shown that for two arbitrarily shaped regions *a,c* ∊ *E* related by RCC-8 relation *T*, there exists some region *b* ∊ E such that *R*(*a*,*b*) and *S*(*b*,*c*). The difficulty due to arbitrary shapes of *a* and *c* is avoided by fixing *a* to be a unit closed disk with centered at o, the origin of the cartesian coordinate system. The aim is to arrive at a closed Jordan curve that defines the boundary of the object *b* depending on the choice of points crucial for the curve. Consider for instance the triad <TPP,DC,TPPI>, the regions *a* and *c* are disconnected as shown in the Figure 12.

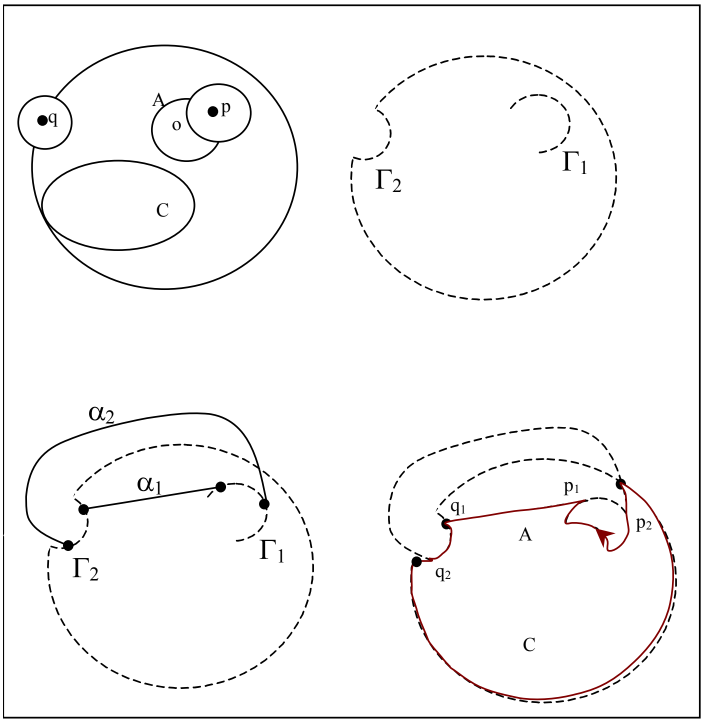


Figure 12. Example to illustrate Jordan curve J for the triad <TPP,DC,TPPI>.

A simple region is b' is drawn that contains both a and c, boundary of c touches b' from inside. Two non-overlapping $\in$-shells with centers (p and q)at boundaries of a and b'. We decide on two arcs- open arc Γ_1 and a closed Jordan curve Γ_2 as shown in Figure 12(b). The two are basically the curves that the object b should contain so that Γ_1 touches the boundary of a from outside whereas Γ_2 touches that of c as well as containing both a and c. p_i, q_i (I=1,2) are points on the two arcs respectively. We have two non-intersecting arcs α_1 and α_2 connecting p_1,q_1 and p_2,q_2 respectively. We take J as follows: We start at p_1 and walk along α_1 till we arrive at q_1. Then walk along Γ_2 clockwise if we will meet C's boundary before meeting q_2 and walk along Γ_2 counterclockwise otherwise. When we arrive at q_2, we then walk along α_2 till we arrive at p_2. Next we walk along Γ_1 clockwise if we will meet A's boundary first and otherwise, walk counterclockwise. We then walk along the arc common between A and the $\in$-shell at p, till we arrive the boundary of the shell. Finally we walk along Γ_1 till we meet p_1 and close the curve as shown in Figure 12. This work is the first ever-exhaustive analysis demonstrating extensionality of all the triads in the RCC-8CT for a domain. This approach does not address the shape of the regions to clearly identify the extensional domain that can be of practical use. Due to the absence of any particular shape, this approach may not be suitable to prove the completeness of RCC-8CT for this domain.

1.3.2 Completeness

The other peculiarity of RCC-8 CT is its *completeness property*. It is necessary to distinguish the concept of consistency with respect to instantiating variables to an element in its domain and the same with respect to the CT. Similarly, the two notions of local consistency- one that refers to extending a partial solution of k variables to a partial solution of k+1 variables and the other that uses CT for enforcing the local consistency, are required to be treated separately in the present context. The issue is due to the fact that the two notions of consistency that do not agree in RCC-8, are otherwise expected to be equivalent. One says that a CSP has a solution if there is an instantiation of all its variables in the respective domain such that all the constraints are satisfied. The other which says that a CSP is path-consistent if for any triad R_{AB}, R_{AC} and R_{BC}, $R_{AC} \subseteq R_{AB} \circ R_{BC}$. The notion of instantiation-based consistency requires the specification of the domain of regions. On the other hand, the notion of local consistency is based on the CT.

Definition 23: A given composition table is complete with respect to a given domain if any CT based singleton consistent network of any size can be realized in the domain.

In the above definition, it is clear that whenever a set of relations is inconsistent can be detected by the composition table. The CT based consistency, *compositional consistency*, is a good algorithmic tool to carry out constraint reasoning. Whenever a singleton network is not realizable, it is possible only when it is not CT-consistent. Completeness of a CT is intimately connected with the relationship between local consistency of triads of relations and overall consistency of the network. Let us consider the domain of regions to be circular discs of equal size. We consider seven such discs such that each is related to other by the atomic relation EC. We see that even if all triads are compositional consistent, we cannot have seven circles externally connected to each other. In [Grigni et al. 1995], it is shown that the RCC-8 CT is not complete for determining consistency of atomic relations when the domain is the set of regions bounded by Jordan curve. It may be recalled that the CT becomes extensional for this domain [Sanjiang, 2003].

1.3.3 Extensionality vs. completeness

The two properties of RCC-8 CT are considered as unrelated [Düntsch, 1999]. These properties of RCC-8CT have been investigated by numerous researchers[Bennett, 1999], [Düntsch, 1999], [Sanjiang, 2003-2004-2005]. These attempts have been successful in identifying necessary conditions that help in verifying whether a given domain is extensional. But none of the researchers have identified sufficient condition(s) for extensionality that could result in the identification of domain for which this CT exhibits extensional interpretation. Moreover no domain is identified so far for which RCC-8CT is extensional and complete.

Extensionality addresses the realization of CT-consistent triplets that are realizable. This indirectly makes it clear that it is meaningless to deal with realization of triplets that are CT-inconsistent. There are some triplets, which are CT-consistent but not realizable. Specific properties of objects in an extensional domain help us in realizing every CT-consistent triplet. The property of extensionality prevents the difficulty of realizing a CT-consistent triplet.
On the other hand, completeness deals with realization of CT consistent networks of larger size. The completeness of a CT lies in identifying the difficulty of extending a partial

instantiation of a network to an instantiation of larger size. If a given CT is complete for a domain, this possibility should arise only due to the inconsistency with respect to CT and not due to the specific properties of the domain. Thus at the level of the networks with three nodes, extensionality implies completeness and vice versa. For networks of size three, if RCC-8CT is extensional in a given domain then it is complete and vice versa.

However, for networks of larger size, same analogy cannot be drawn. If a given CT is extensional with respect to a domain, it may not be always possible to extend the given instantiation infinitely so that the CT is complete as well. However, if a given CT is complete it is possible to conclude extensionality. Consider a hypothetical case of a singleton consistent network *N* with infinitely many objects in the domain such that *N* consists of all the possible consistent triplets of the RCC-8 CT. The fact that the CT is complete, *N* is realizable. This means that a realization is possible for every triplet in the network *N* that in turn implies extensional interpretation of every triad in the CT. The case when the realization is not possible, arises only when the triplet is inconsistent with respect to CT. The proof for extensionality of RCC-8CT is not sufficient to conclude the completeness. Extensionality only assures the existence of a third object for each triad in the CT. The two notions of consistency coincide when the RCC-8 CT is complete and extensional for a domain.

1.4 Constraint Propagation

Constraint propagation technique aims at reducing the search space while maintaining the equivalence. Two CSPs are equivalent if they have the same set of solutions. The constraint propagation algorithms are general methods of checking and enforcing local consistency. A local consistency algorithm tightens the constraints by eliminating any inconsistent relations within a subset of variables. Determining consistency of CSPs with infinite domains is in general undecidable[Hirsch, 1999]. Initially [Montanari, 1974a] developed a form of local consistency, which Mackworth later called path consistency [Mackworth, 1977]. Montanari's notion of path consistency considers all paths between two variables. Mackworth showed that it is equivalent to consider only paths of length two.

Definition 24 [Mackworth, 1977]**:** A constraint network is *path consistent* if and only if for every triplet (A_i, A_j, A_k) of variables, we have that for every instantiation of A_i and A_j that satisfies the direct relation R_{ij}, there exists an instantiation of A_k such that R_{ik} and R_{kj} are also satisfied. Formally, for every triplet of variables A_i, A_j, A_k,

$$\forall A_i, A_j: [(A_i, A_j) \in R_{ij} \rightarrow \exists A_k: ((A_i,A_k) \in R_{ik} \wedge (A_k, A_j) \in R_{kj})]$$

This definition is based on instantiation of variables in the respective domains. [Montanari, 1974b] also developed an algorithm that makes a CSP path consistent, which was later simplified and called path consistency algorithm or enforcing path consistency. Since then a partial method for determining inconsistency of a CSP is the path consistency method [Montanari, 1974], [Mackworth, 1977]. The corresponding CT based definition in this sequel is the following.

Definition 25[Ligozat et al, 2004]: A constraint network is *path consistent* if for every triplet (A_i, A_j, A_k) of variables with constraints R_{ij}, R_{ik} and R_{kj} the relationship holds.

$$\forall \ i, j, k: R_{ij} \subseteq (R_{ik} \circ R_{kj})$$

Generally no distinction is made between the (standard) extensional interpretation of relational composition, which is implicitly assumed in the notion of “path consistency” and so called “weak composition” which is encoded in the composition tables. This distinction has caused problems for many researchers in this area and the confusion is probably due to the fact that for the first set of relations studied in this way, the Allen’s interval relations, the composition table turns out to be extensional, so the two concepts coincide. However the same is not true in the case of RCC-8 relations due to weak composition based CT. If the CT is considered independent of a domain, the relations may not be JEPD. Due to this difference, the two definitions of path consistency are not equivalent in general. The first definition is in terms of instantiations, which ensures 3-consistency. On the other hand, the second definition termed as *algebraic closure* [Düntsch, 1999a-b], [Renz and Ligozat, 2004] is based on weak composition operator.

When a table is not extensional, consistency with a composition table does not enforce path consistency. Thus in order to ensure path consistency, the composition operator requires being the true extensional composition rather than the weak composition given by the CT. For instance, the triad <EC,EC,EC> is certainly CT consistent. Suppose there is no restriction on the underlying domain, we can assign regions a and c such that their sum is the universal region such that a is completely surrounded by c. But for this assignment, there is no instantiation of b which satisfies EC(a,b) and EC(b,c). Thus even this simple CT consistent network though algebraically closed is not path consistent relative to this domain. The two definitions become equivalent for RCC-8 network when we restrict the scope of the domain so that RCC-8 CT is extensional and complete.

The CT based definition gives us an algorithmic tool to enforce consistency whenever it is not so. It is necessary but not sufficient for the consistency of a CSP that path consistency can be enforced i.e. a CSP where path consistency cannot be enforced is not consistent, but a CSP is not necessarily consistent when path consistency can be enforced. A naïve way to enforce path consistency on a CSP is to strengthen relations by successively applying [vanBeek1992] the following operation until a fixed point is reached.

$$\forall\, i, j, k\colon R_{ij} \leftarrow R_{ij} \cap (R_{ik} \circ R_{kj})$$

The resulting CSP is equivalent to the original CSP, i.e. it has the same set of solutions. If the empty relation occurs while performing this operation, then the CSP is inconsistent otherwise the resulting CSP is path consistent. Provided that the composition of relations can be computed in constant time, path consistency can be enforced in $O(n^3)$ time, where n is the number of nodes in the constraint network[Mackworth and Freuder, 1985]. Table 1 contains the $O(n^3)$ time path consistency algorithm [van Beek, 1992]. The idea behind path consistency algorithm is as follows: choose any three vertices i, k, j in the network. The labels on the edges R_{ik} and R_{kj} potentially constrain the label on the edge R_{ij}. To perform this, the algorithm uses ∩ and o operators on the labels and replaces R_{ij} with $R_{ij} \cap (R_{ik} \circ R_{kj})$. If R_{ij} is updated, it may further constrain the other labels and so it is added to the list of labels to be processed. The algorithm iterates until no more changes are possible. The conversions used to compute the constraint are represented by the reverse ordered directed edge.

Algorithm: *Path_Consistency*
Input: A binary constraint network *N* with *n* variables.
Output: fail, if *N* is inconsistent; modified path consistent network *N* otherwise.

```
    Q ← {(i,j,k),(k,i,j) | i<j, k ≠ i, k ≠ j};
        {i indicates the i^th variable in N, analogously for j and k}
    while Q ≠ ∅ do
        select and delete a path (i,k,j) from Q;
        if Revise(i,k,j) then
            if R_ij = = ∅ then return fail
            else Q ← Q ∪ {(i,j,k),(k,i,j) | k ≠ i, k ≠ j};
```

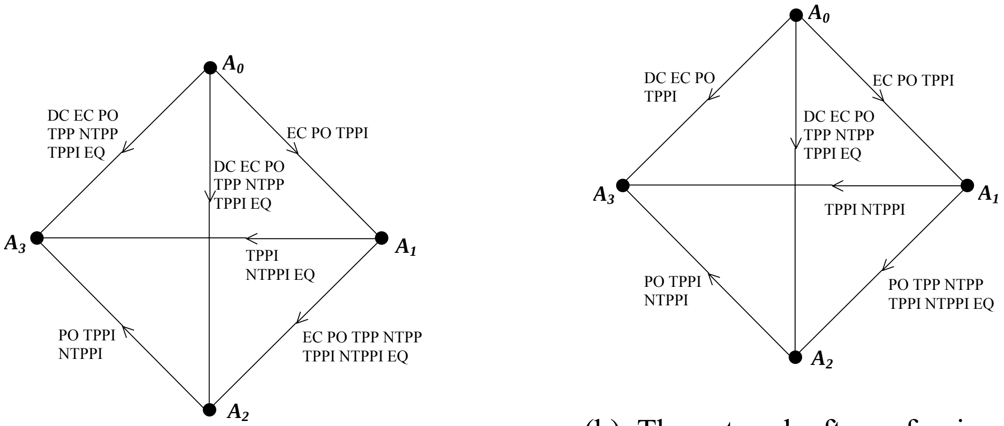
enddo

Function: *Revise*(i,k,j)
Input: three variables i, k and j
Output: true, if R_{ij} is revised; false otherwise.
Side effects: R_{ij} and R_{ji} revised using the operations $\cap$ and o over the constraints involving i, k and j.

oldR = R_{ij}
$R_{ij} = R_{ij} \cap (R_{ik} \text{ o } R_{kj})$
If (oldR = R_{ij}) then return false
$R_{ji} = R_{ij}^{-1}$

return true

Table 1. vanBeek's path consistency algorithm.



(a). Example RCC-8 network.

(b). The network after enforcing path-consistency.

Figure 13. Example network before and after enforcing path consistency.

Consider the example four-node RCC-8 network shown in Figure 13(a). Enforcing path consistency to the network transforms it into an equivalent network having tighter constraints as shown in Figure 13(b).

1.5 Reasoning in RCC-8

One may view RCC-8 as a reasoning formalism independent of any domain. In such cases, the reasoning framework solely depends on the composition table. This restricts the notion of consistency to the compositional consistency so that one can handle the notion of consistency algorithmically. Use of compositional consistency as a tool of reasoning is investigated by various researchers [Nebel, 1995b] and [Renz, 1998]. The basic relations are translated in terms of intuitionistic propositional logics in [Bennett 1994a]. Using this translation, it is shown [Nebel 1995b] that path consistent atomic network is consistent. Continuing this line of research [Renz 1998], it is shown that any path consistent atomic network has a canonical model in any n dimensional Euclidean space when the regions are not forced to be internally connected and for $n \geq 3$ with internally connected regions.

An important reasoning problem in this framework is deciding consistency for a given binary constraint network. The reasoning in RCC-8 is NP-Hard [Renz and Nebel 1999a] for the general RCC-8 network. Path consistency is necessary but not sufficient condition for determining consistency of a CSP. However, there are certain restrictions for which the consistency can be determined in polynomial time. It is shown in [Nebel and Bürckert 1995a] that the complexity of deciding consistency in interval algebra for an arbitrary set of relations ***S*** is same as that of the closure of ***S*** with respect to converse, intersection and composition. Thus any study on tractability analysis can be restricted to the subclasses, or closed sets.

There have been some common approaches to identify these tractable classes and maximal tractable classes. One way is to represent the constraints in some logic theoretic framework such that the satisfiability of CSP becomes equivalent to the satisfiability of the clausal formula in logic. Tractable class of interval algebra as an ORD-Horn clause is derived in [Nebel and Bürckert 1995a] and the similar result for RCC-5 and RCC-8 using Horn clauses in modal logic is derived in [Renz and Nebel 1999a]. Another approach is to interpret the JEPD relations diagrammatically so that a variable instantiation is a point in some Euclidean space and the set of possible instantiations of another entity satisfying a constraint is represented as a region. The concept of convexity was first introduced in [Nökel 1989] and later further investigated by [van Beek and Dechter 1995]. Ligozat [Ligozat 1996] shows that preconvex class is the maximal tractable class for interval algebra(IA). There are similar results for other formalisms too [Balbiani et al. 1999a]. Recently, in [Renz 1999a], a new method is proposed for proving tractability based on a principle of refinement. The tractability of an unknown class is determined by reducing it to a known tractable class by refinement and three maximal tractable classes for RCC-8 namely $\hat{H}_8$, C_8 and Q_8 are found.

$\hat{H}_8$ is shown to be tractable by showing that path consistency for $\hat{H}_8$ is sufficient for consistency. $\hat{H}_8$ is the closure of Horn set of relations H_8, for which path consistency is equivalent to positive unit resolution (PUR) that can be achieved in polynomial time. Refinement [Renz 1999b] is another general method for proving tractability of reasoning over disjunctions of JEPD relations. A classification of tractable classes of RCC-8 is given in

[Renz 1999b] by identifying two new maximal tractable classes namely, C_8 and Q_8, besides $\hat{H}_8$, using refinement. For all these three classes, path consistency is necessary as well as sufficient condition for consistency.

Although the reasoning with RCC-8 is NP-Hard, like the reasoning problem with Interval Algebra, in most cases the problem instances for RCC-8 are not as intractable as the NP-hardness would imply. For random problems with constraints uniformly distributed across the class, hard problems are apparently extremely rare whatever may be the constraint density of the network. Inclusion of significant numbers of tighter constraints seems to massively collapse the search space once $O(n^3)$ compositional inferences are enforced. In [Renz and Nebel 2001], first-ever empirical investigation for RCC-8 framework is carried out. It is shown that algorithms used for qualitative temporal reasoning [van Beek and Manchak 1996] and [Nebel 1997] when adapted for RCC-8 can solve large instances even in the phase transition region. $\hat{H}_8$, C_8 and Q_8 are used to speed up backtracking search by reducing the search space. An orthogonal combination of number of heuristics is able to solve almost all hard instances in the phase transition region up to a certain size in reasonable time.

It is shown by exhaustive enumeration through computer program that that $\hat{H}_8$, Q_8 and C_8 are maximal tractable classes. It is interesting to note that most of the results outlined above are proved using the process of computer-assisted exhaustive enumeration. The closure properties of $\hat{H}_8$, Q_8 and C_8; the tractability of Q_8 and C_8 by way of refining to $\hat{H}_8$ and the maximality of these three classes are proved by exhaustive enumeration and checking. Another interesting use of refinement is to compute a consistent scenario efficiently for $\hat{H}_8$, Q_8 and C_8 in [Renz 1999b]. A path consistent network in any of these classes can be systematically refined to atomic relations in RCC-8 to obtain an equivalent consistent singleton labeling.

1.6 Phase Transition Regions for RCC-8 and Interval Algebra

The famous conjecture [Cheeseman et al, 1991] in regard of the phase transition region in general is:

> *All NP-Complete problems have at least one order parameter and the hard to solve problems are around a critical value of this order parameter. This critical value (a phase transition) separates one region from another, such as over constrained and under constrained regions of the problem space.*

Instances in the phase transition are obviously particularly well suited for testing algorithms on search intensive instances. The purpose of empirically testing the algorithms is to determine the performance of the algorithms and the proposed improvements on typical problems. There are two approaches:

- Collect a set of benchmark problems that are representative of problems that arise in practice.
- Randomly generate the problems and investigate how algorithmic performance depends on problem characteristics and learn to predict how an algorithm will perform on a given problem class.

For interval algebra and RCC-8 networks, there is no existing collection of large benchmark problems that arise in practice – as opposed to the well-known domains of planning and scheduling. We briefly summarize the models for generation of problem instances proposed by several researchers.

1. In [Ladkin and Reinefeld, 1993], it is observed that reasoning in interval algebra has a phase transition region in the range $6 \leq c \times n \leq 15$ for $c \geq 0.5$, where c is the ratio of non-universal constraints to all possible constraints and n is the number of time intervals. This phase transition region is not independent of the instance size, and for this reason does not allow generating arbitrarily hard instances.
2. In [van Beek and Manchak, 1996], two models for IA problem instance generation are proposed:
    a. $B(n)$: Intended to model problems that arise in molecular biology.
        i. Randomly generate the end points of n time intervals. Construct a disjunctive IA network for the set into relations of only two categories: disjoint $\{b, bi\}$ and intersecting $A - \{b, bi\}$.
        ii. Randomly select some of the relations in the network to be non-trivial constraints that closely resemble the Benzer's matrix with 6% are $\{b, bi\}$ and 17% are the rest.
    b. $S(n, p)$: Intended to study performance on sparseness of constraint graph.
        i. Randomly generate a set of edges from ${}^{n}C_{2}$ edges to be present with a probability p. Randomly choose a label for each present edge.
        ii. Randomly generate a consistent singleton network S of n time intervals. Insert S into the above generated network, assuring a consistent IA network.

The hard instances are found with the $p = ¼$, whereas the easy ones are at $p = ½$, 1/8.

3. In [Nebel, 1996], three models for generating IA problem instances are proposed.
    a. *A*(*n, d, s*): Randomly select *nd*/2 edges out of the ${}^{n}C_2$ edges in the network of *n* nodes, termed as the constraint density *d*. For each selected edge, a disjunctive non-universal relation is of size *s* is assigned such that the average size of all the non-universal constraints, termed as the label size s.
    b. *H*(*n, d*): The random instances are generated as in the model *A* above, but instead of random selection of the non-universal constraint, the constraints are selected from particular set of 3006 very hard constraints [Nebel and Bürckert, 1995].
    c. *S*(*n, d, s*): The random instances are generated as in the model *A* above, but the instances are forcibly made satisfiable by adding atomic relations that result from a consistent instantiation.

It is observed that probability of satisfiability drops at $d = 9.5$, $s = 7$. This means that we find over constrained problems around this combination of parameters. Based on the runtime observations, the phase transition region moves to higher values of *d* by varying *s* from 5.0 to 8.0. Thus by varying the constraint density and label size, we can arrive at the complete range of easy and hard problems for networks of all the sizes. The same model underlies the empirical analysis of RCC-8 networks [Renz, J. and Nebel, 2001].

4. In [Pujari and Adilakshmi, 2004], another model is proposed for generating IA instances:
    a. *P*(*n, d, t*): Randomly select d % edges out of the ${}^{n}C_2$ edges in the network of *n* nodes for a constraint density of *d*. For each selected edge, a disjunctive non-universal relation is of label size *t* is assigned to each non-universal constraint. This model does not guarantee any average label size across the complete network. The average size of each instance depends on output of the preprocessing step of path consistency.
    b. *S*(*n, d, t*): The random instances are generated as in the model *P* above, but the instances are forcibly made satisfiable by adding atomic relations that result from a consistent instantiation.

The rule of thumb for any given instance, the hardness can be judged by the order of time taken by the standard complete method of backtrack search only.

1.7 Reasoning in Interval Algebra

1.7.1 Geometric and Lattice Interpretation of Interval Algebra

The atomic relations of interval algebra are represented as pairs of integers between 0 and 4 in a canonical representation [Ligozat, 1991] as shown in Figure 14.

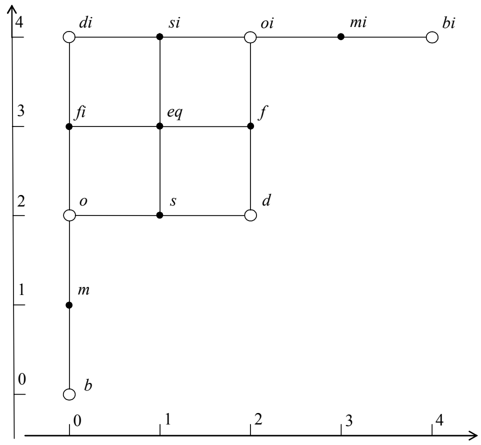


Figure 14. Atomic relations in interval algebra.

This representation gives the set of atomic relations a partial order structure which makes it into a distributive lattice with *b* = (0,0) as the bottom element and *bi* = (4,4) as the top element. Precedence in the IA lattice is defined component wise: $(r_1, r_2) < (s_1, s_2)$ if and only if $r_1 < s_1$ and $r_2 < s_2$. An arbitrary relation is a disjunction of atomic relations is represented as a subset of the lattice. Another representation of interval algebra is in terms of regions in the Euclidean plane. The set of time intervals is identified with the half plane ***H*** defined by the inequation X < Y, where X and Y are the two end points of a time interval. Let (*a.b*) be a fixed interval.

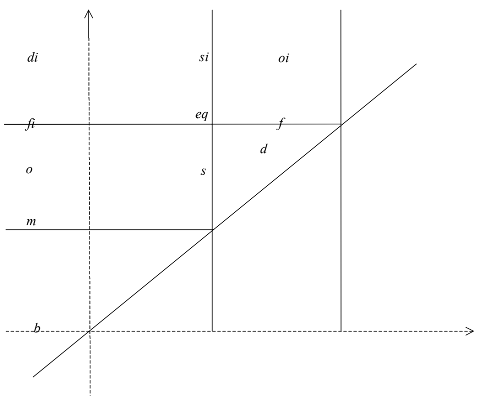


Figure 15. Regions associated to atomic relations in interval algebra.

For each atomic relation, (x,y) is in relation with respect to (*a,b*) if and only if (x,y) belongs to a region in the half plane denoted as reg(r,(*a,b*)) as shown in the Figure 15. The dimension of an atomic relation is defined as the dimension of the associated region. The two-dimensional relations *b, o, d* and their converses are represented as white circles in the IA lattice. One dimensional relations *m, s, f* and their converses are represented as black dots in the lattice. *eq* is the only atom with a dimension of 0 is also a dot in the lattice. The IA convex relations are those relations, which correspond to intervals in the lattice. The IA convex relations are the same set of 82 relations that can be translated into SA relations without the non-equal symbol. The preconvex relations are those relations, which can be obtained from a convex relation by removing one or more lower dimensional relations. There are 868 preconvex relations in interval algebra.

1.7.2 Exact and Approximate Algorithms from Point Algebra

A restricted class of interval algebra networks can be translated without loss of information into PA networks [Vilain and Kautz, 86]. This class of relations is termed as SA relations [van Beek and Cohen, 1990]. In SA networks, the allowed relations between two intervals are only those subsets that can be translated using { ∅, <, ≤, =, >, ≥, ≠, ? } into conjunctions of relations between time interval end points.

An exact $O(n^2)$ algorithm CSPAN, is proposed to find a consistent scenarion for a PA network [van Beek and Cohen, 1990]. A given PA network is condensed into a set of strongly connected components. The strength of the algorithm lies in the fact that throughout the algorithm, path consistency is not used as a pre-processing step. A SA network is translated into a PA network. The consistent scenario of the PA network gives the consistent scenario for SA network.

For any general IA network, CSPAN can be used to determine a consistent scenario if it is a SA networks. For the rest of the cases, the same algorithm is used to improve the performance of backtracking algorithm to compute a consistent scenario. Each label in the network is decomposed into the largest possible sets of atomic relation that are allowed for SA networks. The backtrack algorithm is used to search through these decompositions. This considerably reduces the search space as against decomposing into constituent singletons. The SA sub network is computed using backtrack search followed by $O(n^2)$ time to compute the solution.

The SA relations contain two classes of interval algebra, namely Continuous(C) and Pointizable(P) with 187 relations that are nearly 2% of the IA relations. The continuous class contains 82 relations (1% of complete IA relations), is a proper subset of the pointizable class. The SA relations specific to pointizable class are the SA relations that require ≠ relation for the end-point relation between the time intervals. This can be interpreted as a complete slit in the geometric interpretation of the relations. To summarize, this solution technique can solve only SA networks and cannot be extended as it is for any IA network. This algorithm is exponential in nature in the worst case for general IA networks.

The same author claims it to be useful for the practical planning problems termed as interval graphs [Golumbic, 1980]. The relations in these networks are only two possibilities: disjoint { *b, bi*} or intersecting { *o, oi, d, di, m, mi, s, si, f, fi, eq* }. But the drawback with this representation is that it is too restrictive and one cannot even include if only relation {*b*} is known strongly cannot be stated.

An algorithm FEASIBLE is proposed to find all feasible relations for PA networks [van Beek, 1990] based on the fact that path consistency is sufficient if ≠ is not included, otherwise not sufficient. For a given network the pre-processing step of path consistency is followed by searching for an isomorphic form of a four-node forbidden sub graph and changing the symbol ≤ to <. We can find all the pairs of feasible relations of a SA network by first translating it into PA network, applying FEASIBLE to PA network and translating the result back to SA network. The algorithm is $O(max(mn^2, n^3))$ time, where n is the number of points and m is the number of pairs of points that are asserted to be not equal.

The problem of finding feasible relations in IA network is NP-Complete [Vilain and Kautz, 1986, 1989]. Polynomial time algorithms are proposed to find all consistent instantiations of a general constraint network for finite domains [Freuder, 1978], [Seidel, 1981]. These cannot be extended directly for IA networks, being an infinite domain. We can extend this algorithm for general IA network with the help of SA decomposition based backtrack search. For the minimal label problem, we have to compute all the possible consistent SA sub networks. The algorithm FEASIBLE is repeated for each such network. The feasible relations for IA are just the union of all such solutions. This approach is practical for only small instances and for the instances where only few relations fall outside the allowed SA networks. A better approach to avoid this exponential time of the algorithm approximate algorithm is proposed by the same author. It is proved experimentally that as the percentage of relations out of SA increases, the quality of the solution degrades.

The same author reports an empirical investigation of the above results for IA networks [van Beek, 1992]. The backtrack search based on SA decomposition helps in a three-fold speedup over the conventional singleton based search method. The SA-spit-set algorithm is able to solve instances of size 250 nodes due to the relatively less space requirements. The preconvex-split-set based decomposition showed no additional advantage despite a larger class of relations.

1.7.3 Empirical Analysis Based on Backtrack Search

Reasoning in complete interval algebra in NP-Hard. It is very unlikely to find polynomial-time algorithms that can solve these problems in general. The largest tractable class for interval algebra containing the atomic relations is proposed [Nebel and Burckert, 1994]. This formulation is based on the ORD-Horn relations of interval algebra, which are nearly 10% of the complete set of relations. Extensive machine generated analysis shows that this class is a maximal tractable class such that satisfiability is tractable. The three classes of interval algebra namely the convex class, the pointizable class, and the preconvex class form a strict hierarchy with the branching factors in the increasing order. The complete slit in the pointizable class relations is only a half slit in the preconvex class.

For general IA networks, each label in the network is split into the maximal sets allowed for the preconvex class. This reduces the search space for the backtrack search. An empirical analysis [Nebel, 1996] using the split sets of the pointizable and preconvex classes in the original backtrack search algorithm [Ladkin and Reinefeld, 1992] is performed for the IA problem instances. This investigation is based on the problem instances based on the Nebel-generator explained in the previous section. The parameters of constraint density and label size are varied so that the instance set ranges from easy under constrained cases to the over constrained ones. An orthogonal combination of heuristics of constrainedness, cardinality in backtrack search along with the queue and non-queue based optimizations of path consistency implementations [van Beek and Cohen, 1990] are used in the experimentation. It is observed that the preconvex-split-set based experiments show better performance in the hard region as compared to that of the pointizable-split-set. This conclusion is in contrary to the earlier empirical analysis [van Beek and Manchak, 1996], where it was observed that the pointizable-split-based experiment does not give a major improvement in the performance. This is due to the model generated relatively easy instances as well as the type of the structure of the DNA sequence instances from molecular biology.

In my opinion, all these experiments for IA and RCC-8 are based no doubt on robust and efficient heuristics both for backtrack as well as path consistency. They report very good performance improvements in the hard as well as easy regions. For networks of larger size say beyond n=80, many problem instances have to be terminated with a time out. The algorithms are based on the complete method of backtrack search, but are exponential in the worst case. The generation of problem instances has evolved over a period of time from being size-dependent to size-independent. There seems to be a theoretical void as why some instances, though in the hard region are solved early and some are not. The research has not focused to design the algorithm to intelligently handle the hard instances.

1.7.4 Solution Technique for Over Constrained Interval Algebra Networks

Current research has focused on finding a complete solution or determining that a problem has a solution. If the problem is not solvable then an error is reported and no reasoning is performed on these problems. These are the over constrained problems which are generally the case in real-world problems. In [Thornton, 2002], the first ever research is reported on over constrained temporal constraint satisfaction problem. The technique is based on the partial constraint satisfaction [Freuder and Wallace, 1992], where an optimal partial solution is computed either by removing some of the nodes or relaxing some constraints. Two algorithms based on branch and bound strategies to control the search are proposed. When the backtrack fails, additional atomic relations are inserted and cost is computed at each step or at the end of the search trajectory. An empirical investigation reveals that this approach is suitable only for small-sized problems.

The same author has extended this approach with the help of stochastic local search. An end-point ordering model is introduced that reformulates the TCSP as a standard CSP wherein the variables are instantiated rather than instantiating the constraints with temporal relations. The end-points are assigned a rank in the total ordering of all the end-points. The algorithm starts with randomly instantiating all possible values for every variable. One combination of a value for every variable is picked up as a complete solution of the given network. The cost of a solution is the count of the number of constraints that are not satisfied for the selected combination of values of every variable. A move for deciding the next better solution consists of changing single end-points that involves searching through every possible position for a given interval to obtain best possible solution. Several strategies of optimization such as domain skipping and constraint skipping are also proposed.

It is well known that local search is not a complete method; hence it does not guarantee a solution always. The nature of the search method is such that it is not possible to study the convergence of the method. It is attempt to solve over constrained problem instances, but it does not capture the reason for hardness of the problem instance analytically.

1.8 Reasoning in Weighted Interval Algebra Paradigm

The formalism proposed in [Bhavani and Pujari, 2004] assigns weights to relations in IA network. In an EvIA network, the numerical weights of atomic relations are handled by evidence operations of evidential reasoning. A similar formalism [Rossi et. al, 2002] is studied where the importance of each atomic relation is quantified as weights and used as a heuristic to find a solution of the given network. We reproduce following seven definitions [Pujari and Adilakshmi, 2004] for weighted paradigm for the sake of continuity.

Definition 26: A *weighted IA constraint(W)* is defined as a 13-dimensional vector with the following properties

$$W \in \mathrm{R}^{13} \text{ where } 0 \leq W_m \leq 1,\ 1 \leq m \leq 13 \text{ and } \Sigma W_m = 1$$

where W_m denotes the weight of the corresponding atomic relation IA_m in the disjunctive constraint R_{ij} as per the IA lattice order.

$$W_m^{ij} = \begin{cases} 0, & if\ (r_m \wedge R_{ij}) = \phi \\ 0 \leq W_m \leq 1, & otherwise \end{cases}$$

A weighted IA constraint representation is a deviation from the traditional way of representing a constraint as a disjunction of atomic relations. The weighted paradigm is helpful in translating a discrete problem to a continuous domain.

Definition 27: *A weighted IA network W(N)* is defined as a network where each constraint in the IA network *N* is converted into a weighted IA constraint.

Given an IA network N, we convert it into a weighted IA network by translating each disjunctive constraint R_{ij} in N to a normalized weighted IA constraint W_{ij}. We can have different schemes of assigning weights to the individual atomic relations in a constraint. The initial weights represent our preference or some prior knowledge of the domain. When there is no bias or preference, we can follow the equal weight scheme [Pujari and Adilakshmi, 2004]. The IA-composition table [Dechter, 2003] is represented as a 3-dimensional binary matrix **M**, such that $\mathbf{M}_{ijm} = 1$ if and only if the atomic relation IA_m belongs to the composition of the atomic relations IA_i and IA_j.

***Definition 28*:** The composition of two weighted IA constraints W^{ik} and W^{kj} resulting in a relation $W^{ij}(k)$ is denoted as $W^{ik} \otimes W^{kj}$. Its each component is defined as follows.

$$W_m^{ij}(k) = \frac{\sum_u \sum_v M_{uvm} W_u^{ik} W_v^{kj}}{\sum_m \sum_u \sum_v M_{uvm} W_u^{ik} W_v^{kj}}, \quad 1 \le m \le 13.$$

Definition 29: Converse of a weighted IA constraint W is defined as another weighted IA constraint, denoted as $W^{\sim}$, containing weights for the inverse of the base relations in W.

$$W^{\sim}_{m^{\sim}} = W_m,\ m^{\sim} = (14 - m),\ \ 1 \le m \le 13.$$

***Definition 30*:** The intersection of two weighted relations W^{ij} and V^{ij}, denoted as $U^{ij} = W^{ij} \cap V^{ij}$. Its each component is defined as follows.

$$U_m^{ij} = \frac{W_m^{ij} V_m^{ij}}{\sum_m W_m^{ij} V_m^{ij}},\ 1 \le m \le 13$$

U^{ij}_m is larger in value only if both the weights W^{ij}_m and V^{ij}_m are larger.

The *weighted_pc_iteration* procedure modifies the weights of atomic relations so that the common relation has stronger weight. The algorithm detects inconsistency when the result of intersection of weighted vectors is a 0-vector.

We summarize the work by some researchers based on complete instantiation strategies based on different types of approximations. It is proved [Ligozat, 1994] that the preconvex class of relations in IA is the same set of ORD-Horn relations [Nebel and Burckert, 1994]. The convex closure of a general IA network maintains the consistency of the underlying network [Ligozat, 1996]. In addition to proving the tractability of the preconvex class, a stronger result is the existence of solution consisting of maximal dimension element from each edge [Ligozat, 1996]. The method inherently selects only that atomic relation on edge that minimizes the number of common end points. The choice fails if it implies more number of common end points. Such a drop of a choice is equivalent to a dead end in a backtrack search. This result proves only the existence of such a solution but does not give a straight step to construct a solution. This may not always be true since not every possible singleton network with two-dimension atomic relation is path consistent.

The commonality between a preconvex and the parent convex relation is the set of two-dimensional atomic relations. The analysis is based on the intuition that a preconvex relation is basically a convex relation without some particular atomic relations. In order to find a solution of a given IA network, we can first relax the constraints a bit and then choose a scenario for the corresponding convex network, which is basically consistent. We cannot expect in general to get a solution of the original network. On the same line, a complete instantiation strategy based on an upper approximation of the convex cover network [Pujari and Adilakshmi, 2004]. The solution strategy is based on a highest weight heuristic in a weighted paradigm. The algorithm reports multiple solutions for path consistent convex IA networks. However, the same algorithm fails to show up to the same performance for general IA networks due to the unhandled numerical errors that propagate further in the later stages of the algorithm. We conclude that there does not exist a complete polynomial time algorithm for solving general IA networks to compute either an exact or an approximate solution.

Entropy for IA network is introduced in [Adilakshmi and Pujari, 2002] but the convergence of weighted path consistency iterations was difficult to analyze due to the logarithmic term in the constraint entropy. The Renyi-entropy based interpretation of weighted IA network [Pujari et al, AI-Meth 2005] that we have introduced recently makes the analysis relatively easy. The quantitative measure of the information in a constraint can be computed by Renyi's quadratic entropy (RQE)[Zitnick III, 2003] is given as $-\log\sum_m (W_m^{ij})^2$. In the context of minimizing entropy, without loss of generality, *log* in the above expression is normally dropped.

Definition 31: Entropy of a weighted constraint W^{ij} is defined as follows

$$E_w^{ij} = -\sum_m (W_m^{ij})^2 \ \ where \ \ 0 \le W_m^{ij} \le 1 \ \ and \ \ \sum W_m^{ij} = 1$$

Definition 32: Entropy of a weighted network is defined as the sum of the entropy of all its edges and is given as

$$E_N = \sum_{ij} E_w^{ij}$$

The least entropy of a constraint corresponds to a singleton relation and the highest entropy is when it has non-atomic relations with equal weights. Thus for a network, the least entropy states are singleton labeling and the initial states when we assign equal weights to the disjuncts correspond to highest entropy states. By the law of nature, any given system stabilizes to a low energy state. The behavior of the system in terms of how quickly it stabilizes indicates the extent of disorder present in the system. The path consistency algorithm reduces the entropy of the constraint network as the constraints are tightened up over the iterations. This work is a first ever attempt to study the search trajectory with the help of entropy of the disjunctive IA network.

1.9 Problem Statement and Contribution

1.9.1 Problem Statement

Tractability study for RCC-8 started with identifying H_5 for RCC-5 followed by H_8 for RCC-8 using modal logic. Later C_8 and Q_8 are identified using the method of path consistency preserving refinement. Tractability for the maximal tractable classes is supported by computer-assisted results. To the best of our knowledge, so far no analytical proof has been proposed for RCC-8. Though answers to these questions are undoubtedly important, this aspect of research has somehow escaped the attention of AI researchers. In this thesis we attempt to fill in these gaps.

In this thesis, we propose lattice-theory based representation of RCC-8 to provide a better insight of its expressive power. Basic properties of partially ordered sets are studied to give a new description of the known tractable classes. The disjunctions of RCC-8 relations correspond to partially order sets in the lattice and the existence of the least element plays a very crucial role in analyzing the tractability properties of the set of relations. We show here that the known tractable classes exhibit this property and hence, path consistency becomes sufficient condition for existence of a solution. We also establish here that for these classes, we can get a solution by a process called 'least element instantiation'. We identify a new set J_8, a superset of the three maximal tractable classes of RCC-8. We prove that reasoning in J_8 is a polynomial-time problem. We analytically support the result that path consistency is sufficient for deciding satisfiability in J_8. Tractability proofs for the three existing maximal tractable classes become special cases. The property of lattice intervals in RCC-8 relations is studied to identify two new tractable classes namely $Conv_8$ and $PConv_8$. Interestingly, the path-consistency of the convex class implies minimal network. This study identifies a subset of RCC-8 satisfying the minimality property. The formalism proposed in this thesis can be extended to other frameworks of qualitative reasoning.

Main contributions of this extensive study are four-fold. The novel method proposed here lays a foundation for a new direction of reasoning for any qualitative binary CSP. Secondly, this method helps in providing a simple analytical proof for all the tractability results in RCC-8 reported so far by other researchers using a number of complex methods. Thirdly, we demonstrate the applicability of this method by identifying three new tractable classes of RCC-8. We extend this method to show that the convex class indeed satisfies the minimality property as well. Lastly, we propose an efficient single-step algorithm for constructing a solution for any general RCC-8 constraint network that is path consistent without any computational overhead of enforcing path consistency intermittently.

Surprisingly, number of performance improvements due to the backtrack search based on tractable classes based split set have been enumerated, there is no research reported to identify a measure of hardness of problem instances for qualitative constraint networks. One of the implications of such a measure would be an ease in solving the instances of general IA networks. A question that arises naturally is - Why instances are hard to solve?

Tractability study of interval algebra started with identification of SA networks, identification of convex, preconvex, pointizable class based on different methods - starting from modal logic, lattice and geometric based interpretation and the computer-assisted programs. Analytically, the corner relations are identified that prevent tractability for a set of relations and maximal instantiation strategy for path consistent preconvex IA networks - but not supported by experiments. A set of 3006 constraints is identified using modal logic that is later used to generate hard IA instances. Several empirical analyses conclude by reporting an improved performance profile. A search technique based on local search attempts computing an optimal partial solution for general IA networks. The drawback is that it is theoretically known to be an incomplete method. The entropy-based interpretation of weighted IA networks is proposed but no analysis of the convergence of the algorithm is possible since it is an incomplete one.

To the best of our knowledge, there is gap in research that there is no approach so far that is capable to give a global overall look for the reason for hardness of problem instances. There is no method that can handle the known inconsistent networks also as a special case of the consistent IA networks, instead of just reporting error. As researchers in the field of CSP, we cherish to look for complete algorithms that are polynomial time and do not anywhere depend on backtrack search.

In the second part of this thesis, we attempt to fill in these gaps. We extend the existing entropy-based interpretation of the weighted IA network paradigm. Our investigations reveal that the fluctuations in the entropy of the IA constraints, termed as nasty constraints are the reason for hardness of the instance. The conflict between the highest weight relation on the edge and the most supported relation along the paths is the reason for entropy fluctuations in the IA networks. We show here that the known tractable classes of interval algebra do not exhibit the property of entropy fluctuations. We investigate the reasons for the incompleteness of the highest weight heuristic. We provide different techniques to relax the nasty constraints so that the number of conflicts reduces as the entropy of each constraint stabilizes over the search trajectory. This controls the path that the search trajectory takes resulting into an approximate solution of any instance of general IA networks. This study differs from the existing solution techniques of upper approximation of the IA constraints. These approaches adopt sort of a blind approximation of each constraint without addressing whether any particular relaxation is responsible for hardness. This study identifies the nasty constraints in an intelligent manner with the help of the conflicts. The main contributions of the study on weighted IA networks are five fold. The novel method of conflicts proposed here lays a foundation for a new direction towards addressing hardness of any qualitative binary constraint satisfaction problem. Secondly, this method provides simple analytical proof for ease and difficulty in computing a solution instance wise. Thirdly, we systematically investigate number of methods to relax the identified nasty constraints that has finally helped us to propose an algorithm to compute an approximate solution on completion. Fourthly, we handle the enormous amount of numerical errors that arise due to each and every operation on a weighted IA constraint. This approach is very handy and can be used by anybody in practice. Lastly, we extend this method to show that the inconsistent IA networks can be handled by this algorithm as routine cases of nasty constraints with trivial conflicts.

Apart from the theoretical analysis, we demonstrate an application of the least element-based method to predict a scene that can follow one in the domain of road traffic. The investigation reveals that the weighted paradigm based strategy is not suitable for time-based prediction in this domain. On the other hand, the prediction based on spatial method can be extended to detect occlusion.

In a nutshell, the present thesis tries to answer the following problem:

How to analyze- the tractability beyond known tractable class to find reasons for hardness of instances in qualitative CSPs and to solve any given instance whether consistent or inconsistent.

1.9.2 Thesis Contribution

Though the research presented in this thesis mainly contributes to the tractability analysis of qualitative frameworks of RCC-8 and interval algebra, it can also be used for other qualitative frameworks in general.

1.9.2.1 Lattice based Tractability Analysis of RCC-8

We present a new lattice theoretic formalism for tractability study of qualitative constraint satisfaction problems. The proposed formalism helps in supporting theoretical analysis of the existing results and in obtaining new results on tractability for RCC-8. We show that the existence of the least elements of partially ordered sets plays a very important role in identifying tractable sets. This property can also be used to determine consistent instantiation of the constraint network. The present study demonstrates that various tractable sets can be determined and efficient algorithms can be devised using this characterization. We provide theoretical proofs of the several results that rested on machine-assisted proofs so far. We introduce three new tractable classes namely convex class, preconvex class, and DPConv8. We show that any path consistent convex RCC-8 network is minimal. To the best of our knowledge, there is no known class in RCC-8 for which path consistency implies minimality. The partial order together with the dimension of RCC-8 relations can be used to identify the preconvex class. We show that coarsening as a reverse process of refinement determines a new tractable class namely, DPC8. We propose an algorithm based on the least element instantiation that is more efficient than the existing algorithms for finding consistent singleton label. As a consequence, for any set admitting the existence of least element, it can be shown that path consistency suffices for consistency and hence consistency can be determined in polynomial time. The same result is extended to set of relations J_8 that is not a closed class but path consistency decides consistency.

1.9.2.2 Characterization of hard TCSP

Interval Algebra (IA) based temporal constraint satisfaction problems (TCSPs) are useful in formulating diverse problems. The useful approach to solve IA networks is based on partial instantiation strategy - backtrack search for exact solution. We tailor the formalism proposed in the previous chapter to another framework in qualitative binary CSP – interval algebra. We provide a simple analytical proof for all tractability results in interval algebra reported so far. This helps us to analytically answer the question "What makes an IA instance easier?". We show that high sets plays an important role in tractability. We show here that the known tractable classes exhibit this property. We also establish here that for these classes, this property can also be used to determine consistent instantiation of the constraint network by a method called '*highset instantiation'*. To the best of our knowledge, determining approximate solution for TCSPs is not addressed so far. In this chapter, we propose a new complete instantiation strategy based on a complete algorithm to determine approximate solution of IA networks. We identify a property of constraints called nastiness that disturbs monotonic nature of entropy of a constraint. We go beyond the identification of nasty constraints to pinpoint the singleton to restore normal behavior of entropy. On termination, the algorithm guarantees either an exact or approximate solution depending upon the number of constraints the solution violates. We demonstrate experimentally that solution to general IA networks can be efficiently obtained with the success rate of 95% contrary to exponential exact algorithm.

1.9.2.3 Application: Prediction of Next Road Traffic Scene

Conventional approach in the CSP applications - partial instantiation, enforce path consistency and apply backtrack search. The study in this thesis is completely based on complete instantiation strategies for both the RCC-8 and Interval algebra networks. We demonstrate the application of the least element instantiation strategy in the domain of road traffic scene to detect occlusion. We start with a complete instantiation that represents the starting scene frame. The activity in the scene starting from the first frame till the last frame in the scene can be described by navigation in the lattice order. The immediately following scene is among the generated candidate networks. The partial order is very helpful in navigating from one instantiation to another. We demonstrate that either due to occlusion or an object leaving a scene, the number of objects decrease in the subsequent scene. We propose a strategy to distinguish between instances of occlusion from that of an object exit.

1.9.3 Thesis Outline

The thesis is organized as follows:

Chapter 1: The first chapter entitled "Qualitative Representation and Reasoning", contains a survey of existing techniques and methods on two popular qualitative frameworks. The chapter covers most of the methods and algorithms in an elaborate manner. The chapter covers the research on all the aspects from basics of binary constraint satisfaction problem modeling as constraint networks, composition table, approaches to identify tractable classes and all the solution techniques. We also provide a brief introduction on existing models for generation of problem instances. We also provide the problem statement and main contribution of our work, which is followed by the organization of the thesis.

Chapter 2: The second chapter entitled " Lattice Based Tractability Analysis of RCC-8 ", provides the details of our work on the RCC-8 framework. The idea is inspired by the partial order based lattice structure of the sister framework- interval algebra. We introduce a lattice based interpretation of RCC-8 relations. We provide an analytical interpretation of the relations that are not allowed in any of the maximal tractable classes known so far. The least element based interpretation of the relations help us to identify a tractable set that is larger than all the maximal tractable classes. We identify convex and tractable classes for RCC-8 and prove that the convex class indeed satisfies the minimality condition. We devise a single step solution technique for general RCC-8 path consistent networks with least element.

Chapter 3: The third chapter entitled "Characterization of Hard TCSP", provides the details of our work on the existing weighted interval algebra paradigm. The idea is inspired by an in-depth analysis of the highest weight heuristic that fails to compute a solution for general interval algebra networks. We introduce the concept of nasty constraints for weighted IA constraints. We provide an analytical proof for tractability of convex and preconvex classes of interval algebra based on a new concept of highest. We provide an analytical proof that the fluctuation in entropy of constraints in the network is due to the presence of nasty constraints. We introduce different strategies to relax the nasty constraints to resolve the conflict between the edge constraint and the path constraint. We devise a strategy to compute an approximate solution of general IA networks. We show that the same algorithm can also handle the inconsistent networks.

Chapter 4: The fourth chapter entitled " Application: Prediction of Next Road Traffic Scene ", provides the details of our work on the application of the least element instantiation identified in this study. The idea is inspired by the continuity of the movement of objects that is interpreted in terms of the lattice-based navigation of the relations in the network that represents a scene of objects. The domain we choose to demonstrate is the road traffic scene of moving vehicles.

Chapter 5: In this chapter entitled "Conclusions", we summarize our work, presented in the thesis. The conclusions are drawn based on the observations and empirical results. We also provide some future directions, in which the research, described in the thesis, can be extended.

A detailed bibliography on qualitative reasoning research and relevant areas is included at the end of the thesis.

CHAPTER 2
LATTICE BASED TRACTABILITY ANALYSIS OF RCC-8

2.1 Introduction

The main result in this chapter is the least element-based interpretation of RCC-8 relations. In this study, we are concerned with the algorithmic properties of CT based consistency. Our results provide a good tool for reasoning in RCC-8 and valid for the domain for which the CT is extensional and complete. We assume strong composition in this chapter. We propose a domain of objects for which RCC-8 CT is complete and extensional. In this chapter, we propose lattice-theory based representation of RCC-8 to provide a better insight of its expressive power. Basic properties of partially ordered sets are studied to give a new description of the known tractable classes. The disjunctions of RCC-8 relations correspond to partially order sets in the lattice and the existence of the least element plays a very crucial role in analyzing the tractability properties of the set of relations. We show here that the known tractable classes exhibit this property and hence, path consistency becomes sufficient condition for existence of a solution. We also establish here that for these classes, we can get a solution by a process called '*least element instantiation'*. We identify a new set J_8, a superset of the three maximal tractable classes of RCC-8. We prove that reasoning in J_8 is a polynomial-time problem. We analytically support the result that path consistency is sufficient for deciding satisfiability in J_8. Tractability proofs for the three existing maximal tractable classes become special cases. The property of lattice intervals in RCC-8 relations is studied to identify two new tractable classes namely $Conv_8$ and $PConv_8$. Interestingly, the path-consistency of the convex class implies minimal network. This study identifies a subset of RCC-8 satisfying the minimality property. The formalism proposed in this thesis can be extended to other frameworks of qualitative reasoning.

Main contributions of this chapter are five-fold. First and foremost, we investigate and identify a domain for which RCC-8 CT is complete and extensional. Secondly, the novel method proposed here lays a foundation for a new direction of reasoning for any qualitative binary CSP. Thirdly, this method helps in providing a simple analytical proof for all the tractability results in RCC-8 reported so far by other researchers using a number of complex methods. Fourthly, we demonstrate the applicability of this method by identifying three new tractable classes of RCC-8. We extend this method to show that the convex class indeed satisfies the minimality property as well. Lastly, we propose an efficient single-step algorithm for constructing a solution for any general RCC-8 constraint network that is path consistent without any computational overhead of enforcing path consistency intermittently.

2.2 Domain of objects for which RCC-8 CT is Extensional and Complete

In this section, we propose a domain of spatial objects for which RCC-8 CT is extensional and complete. There are two approaches to tackle this research problem. First one but more difficult is to modify the set of axioms that underlay a given composition table. The other option, which is more meaningful in actual practice, is to identify the domain of spatial objects for which notions of extensionality and completeness hold. A motivating example is the interval algebra composition table with the domain of one-dimensional time intervals on the real line. It is possible to realize any size atomic path consistent interval algebra network. The consistency of such a network directly depends on the local consistency of singleton triplets. We investigate these properties with the help of a domain as defined below.

Definition 1: A domain of spatial regions $\boldsymbol{D_T}$ is defined as the set of triangular regions where each region is bounded by three lines in the Euclidean plane satisfying the following properties:

a. Universal triangular region not allowed.
b. Hole relation between any pair of triangular regions not allowed.
c. Triangular regions with discrete components not allowed.

The three properties mentioned above are required for any domain as necessary conditions for extensionality (Chapter 1). We have verified that all the 193 triads of RCC-8 CT are extensional. The details are included in Appendix A. We formalize this result as the following theorem.

Theorem 1: RCC-8 CT is extensional for the domain $\boldsymbol{D}_{\mathrm{T}}$.

Proof:

Consider a singleton path consistent RCC8 network *N* consisting *n* nodes.

Let us assume that RCC-8 CT is not extensional for the $\boldsymbol{D_T}$. This means that there will exist at least one triplet in *N* that violates the condition for extensionality. Since *N* is consistent with respect to CT, this implies that this triplet is consistent but we are not able to place a third object with respect to a given pair of objects in this domain but satisfy the constraint.

This violates the basic observation that each triad of RCC-8 CT is extensional for this domain. This contradiction to our assumption proves that the extensional interpretation for any given *n*-node network *N* is possible.

■

The two properties of completeness and extensionality of RCC-8 CT are considered as unrelated [Düntsch, 1999]. These properties of RCC-8CT have been investigated by numerous researchers[Bennett, 1999], [Düntsch, 1999], [Sanjiang, 2003-2004-2005]. These attempts have been successful in identifying necessary conditions that help in verifying whether a given domain is extensional. But none of the researchers have identified sufficient condition(s) for extensionality that could result in the identification of domain for which this CT exhibits extensional interpretation. Moreover no domain is identified so far for which RCC-8CT is extensional and complete.

Extensionality addresses the realization of CT-consistent triplets. This indirectly makes it clear that it is meaningless to deal with realization of triplets that are CT-inconsistent. There are some triplets, which are CT-consistent but not realizable. Specific properties of objects in an extensional domain help us in realizing every CT-consistent triplet. The property of extensionality prevents the difficulty of realizing a CT-consistent triplet.

On the other hand, completeness deals with realization of CT consistent networks of larger size. The completeness of a CT lies in identifying the difficulty of extending a partial instantiation of a network to an instantiation of larger size. If a given CT is complete for a domain, this possibility should arise only due to the inconsistency with respect to CT and not due to the specific properties of the domain. Thus at the level of networks with three nodes, extensionality implies completeness and vice versa. For networks of size three, if RCC-8CT is extensional in a given domain then it is complete and vice versa.

However, for networks of larger size, same analogy cannot be drawn. If a given CT is extensional with respect to a domain, it may not be always possible to extend the given instantiation infinitely so that the CT is complete as well. However, if a given CT is complete it is possible to conclude extensionality. Consider a hypothetical case of a singleton consistent network *N* with infinitely many objects in the domain such that *N* consists of all the possible consistent triplets of the RCC-8 CT. The fact that the CT is complete, *N* is realizable. This means that a realization is possible for every triplet in the network *N* that in turn implies extensional interpretation of every triad in the CT. The case when the realization is not possible arises only when the triplet is inconsistent with respect to CT. The proof for extensionality of RCC-8CT is not sufficient to conclude the completeness. Extensionality only assures the existence of a third object for each triad in the CT. The two notions of consistency namely CT-based and instantiation-based, coincide when the RCC8CT is complete and extensional for a domain. The property of completeness of a CT implies that it is possible to realize any size consistent base network in the given domain. The CT is said to be incomplete for the domain when it is not possible to extend the realization beyond a certain node size. For instance, the realization of an all-EC network of 7 nodes cannot be realized in the domain of equal sized discs. The key to conclude the completeness of the CT lies in the fact that it is should be possible to realize any given network infinitely.

The basic intuition for an infinite number of objcets to touch each other is that a point can be common to any number of triangles irrespective of the size. An all-EC network of any size can be realized in the domain of triangular objects $\boldsymbol{D_T}$ as shown in the Figure 16. All the fifteen triangles touch each other at the centre point.

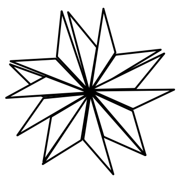


Figure 16. Realization of a 15-node all EC RCC-8 network in domain D_T.

The above example is a much studied counter example for RCC-8 CT to be complete for any domain so far as discussed in chapter1. The domain $\boldsymbol{D_T}$ identified in this study does not suffer from any such counter example. If we are not able to generate a realization for a network of any size in this domain, implies that the network is CT-inconsistent. We have the following theorem in this direction.

Theorem 2: RCC-8 CT is complete for the domain $\boldsymbol{D_T}$.

Proof:

Let us assume that RCC-8 CT is not complete for the domain $\boldsymbol{D_T}$. By the definition of completeness of a CT(chapter 1), if a given singleton network *N* is consistent with respect to the CT, it should be possible to generate a realization of *N*. If we are not able to realize *N* implies that *N* is not CT-consistent.

As per our assumption, though *N* is consistent but realization in $\boldsymbol{D_T}$ is not possible. This implies that there is a sub network in *N*, which is CT-consistent and realizable, but it is not possible to extend this realization from *k* to *k*+1 triangles in the domain $\boldsymbol{D_T}$. This violates the above-mentioned property of $\boldsymbol{D_T}$ that it is possible to extend an instantiation to infinitely many triangles. There are three possibilities for this extended realization not being possible:

a. *N* consists of at least one inconsistent triplet, a violation to assumption: *N* is CT-consistent.
b. Extensional interpretation is violated for at least one triplet in *N*, which violates theorem 1.

This proves that RCC-8 CT is complete for the domain $\boldsymbol{D_T}$.

■

This motivates us to conclude that completeness of RCC-8 CT impacts the extensionality property, where instead of a third object, it is possible to place a *n*th object in the presence of (*n*-1) objects. We have the following theorem in this direction.

Theorem 3: Completeness of RCC-8 CT with respect to a domain is a sufficient condition for RCC-8 CT to be extensional.

Proof:

Suppose *N* is a singleton path consistent RCC-8 network. Let us assume that RCC-8CT is complete for the domain $\boldsymbol{D_T}$, but not extensional. Two possibilities arise in this case:

a. Atleast one triplet is inconsistent in the network *N*. This violates the property of completeness given that *N* is consistent.
b. A triplet is consistent but there exists a sub network in *N* for which it is not possible to extend a realization. This means that there exists at least one triad such that extensional interpretation is not possible. This violates the completeness property for the triplet.
c. Violation of definition of extensional model, i.e. $v(R) \text{ o } v(S) = \cup_{T\in \text{ CT}(R,S)} v(T)$. This means that there exists one triplet such that a direct relation *T* exists between objects *a* and *c*, but the indirect relation does not exist. We are not able to place a third object *b* with respect to *a* and *c*. This violates the completeness property that any difficulty in extending a realization can arise only due to inconsistency of the triplet.

■

For any given domain, if the RCC-8CT is complete, the only possibility for a domain to fail for extensionality arises only when the CT is a weak CT for the domain. Testing whether a network of atomic relations among *n* objects is consistent with a CT is tractable. It amounts to checking all the sub networks of size three in $O(n^3)$ time. There has been considerable interest in generalizing the use of CTs to deal with reasoning about relations, which are disjunctions of those appearing explicitly in the table. From the point of view of the completeness of a CT the case of disjunctive compositional reasoning is not significantly different from that of simple non-disjunctive reasoning. As long as reasoning for atomic relations is complete then checking consistency of a network containing disjunctive relations is only a matter of showing that by picking one disjuncts from each arc of the network, we can obtain a non-disjunctive network which is consistent with the CT, and if such a network exists it can always be found by any solution technique.

In this study, we are able to propose a domain for which RCC-8 CT is complete as well as extensional. We address these two issues in an integrated fashion. The analogy from RCC-8 to interval algebra like objects has helped us to formalize that completeness property for RCC-8CT is sufficient enough a condition for extensionality. In this study, we are concerned with the algorithmic properties of CT based consistency. Our results provide a good tool for reasoning in RCC-8 and valid for the domain for which the CT is extensional and complete.

2.3 Semi-Lattice based interpretation of RCC-8 relations

In [Freksa 1991], conceptual neighborhood for temporal and spatial relations is introduced. Two relations are neighbours if they can be directly transformed into one another by continuous deformations of the entities on which the relations are defined. If the conceptual neighborhood is taken into consideration, a partial order $\preccurlyeq$ on **U** can be naturally induced. Following this, we represent the elements of **U** as shown in Figure 17. The ordering $\preccurlyeq$ is partial as there are atomic relations that are not related with respect to $\preccurlyeq$. For instance there is no ordering between TPP and TPPI. We list below all the pairs of atomic relations for which there is no ordering.

TPP and TPPI,	NTPP and EQ,	NTPPI and EQ,
NTPP and NTPPI,	TPP and NTPPI,	NTPP and TPPI

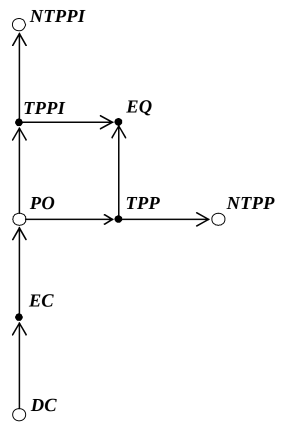


Figure 17. Partial ordering of atomic relations in RCC-8

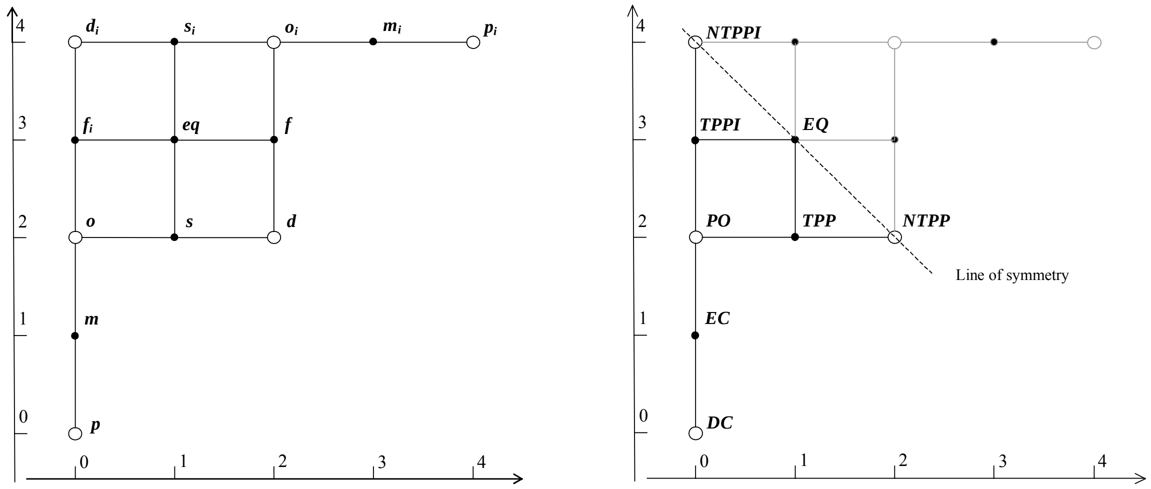


Figure 18. Symmetry between interval algebra lattice and RCC-8 semi lattice.

Each non-atomic relation in RCC-8 forms a partially ordered set with respect to the precedence relation $\preccurlyeq$. For $r \preccurlyeq s$, we say r precedes s. It is interesting to note that if the spatial entities are intervals, the ordering by $\preccurlyeq$ conforms to that of IA lattice as proposed in [Ligozat 1994].

Definition 2**:** An atomic relation m in a RCC-8 relation R is a *minimal element* of R if for any $s \in R$, $s \preccurlyeq m$ then $s = m$.

A partially ordered set may consist of more than one minimal element. The above definition ignores the atomic relations that are not related to the minimal element. Any atomic relation if related to the minimal element can only succeed but not precede it. For instance, the relation R = {TPP, TPPI, NTPPI} can be divided into two subsets namely R_1 = {TPP}, R_2 =

{TPPI, NTPPI} such that none of the atomic relations in either of the two are related to those in the other. TPP is a self-minimal element in R_1. TPPI is the minimal element of R_2 since it precedes every element in R_2. R consists of two minimal elements namely TPP and TPPI that are not related to each other.

***Definition 3*:** An atomic relation l in R is called a *lower bound* of R if $l \preccurlyeq s, \forall s \in R$.

If every other lower bound of R precedes the lower bound l of R, then l is called the *greatest lower bound* or *infimum* of R, i.e., $inf(R)$. Infimum of a partially ordered set is unique if it exists. Infimum of a set of atomic relations need not necessarily be in it. For instance, the candidate lower bounds for the relation R = {TPP, NTPP, TPPI} are the atomic relations DC, EC, PO. TPP and TPPI are the minimal elements of R. R does not contain any of the possible lower bounds. PO is the greatest lower bound of R denoted as $inf(R)$ = PO.

***Definition 4*:** An atomic relation s is the *least element* of R, if s is a greatest lower bound of R and is an element of R. We denote for a relation R, the least element as $\sigma(R)$, if it exists.

For instance, the infimum of the relation R_1 = {DC, TPP, NTPP, TPPI, NTPPI, EQ} is DC, and DC$\in R_1$. Hence $\sigma(R_1)$ exists. On the other hand, for R_2 = {TPP, NTPP, TPPI, NTPPI, EQ}, $inf(R_2)$ = PO. Since PO $\notin R_2$, $\sigma(R_2)$ is not defined.

Definition 5: An element b of a partially ordered set is called a *maximal* element if every other atomic relation in R precedes b.

For instance R = {DC, EC, PO, TPPI, NTPPI} has NTPPI as a maximal element and DC as minimal element. For R_3 = {TPP, TPPI, NTPP, NTPPI, EQ}, TPP is a minimal element since the atomic relations that are related to TPP are NTTP, EQ. TPP precedes both the atomic relations. Similarly, TPPI is another minimal element. NTPP, NTPPI and EQ are three maximal elements in R_3.

Any given relation in RCC-8 can be divided into a group of subset relations such that each subset contains at least one minimal element. In the worst case, when none of the elements in a relation are related, each subset contains an atomic relation. The same holds true for the maximal element(s) for each relation in RCC-8. We conclude that every relation in RCC-8 has at least one minimal as well as at least one maximal element.

Definition 6**:** A totally ordered subset of a partially ordered set R is a *chain* in R.

For instance, for R = {DC, EC, PO, TPP, TPPI} some chains are C_1 = {DC, EC, PO, TPP}, C_2 = {DC, EC, PO, TPPI}, C_3 = {DC, EC, PO}. The subset {PO, TPP, TPPI} is not a chain since TPP and TPPI are not comparable with respect to $\preccurlyeq$. The complete set of chain relations in RCC-8 contains 33 relations (Appendix - D).

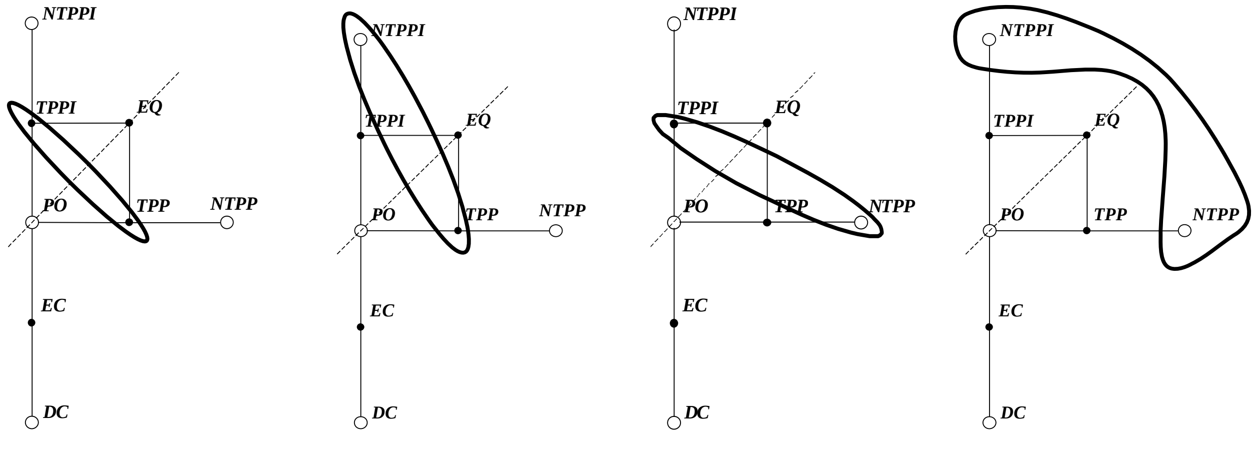
Definition 7**:** An *antichain* is a set of pair wise incomparable elements.

All possible pairs of incomparable atomic relations listed above are antichains denoted as namely AC_1 = {TPP, TPPI} AC_2 = {TPP, NTPPI} AC_3 = {NTPP, TPPI} AC_4 = {NTPP, NTPPI} AC_5 = {NTPP, EQ} AC_6 = {NTPPI, EQ} and AC_7 = {NTPP, NTPPI, EQ}. For instance, the relation {DC, NTPP, NTPPI, EQ} is not an antichain since DC $\preccurlyeq$ NTPP(I) and DC $\preccurlyeq$ EQ. An antichain A is a *maximal antichain* if no other superset of A in the partially ordered set is an antichain. AC_7 is a maximal antichain since any superset obtained by inserting any other atomic relation is not an antichain. Similarly, AC_1, AC_2, AC_3 , AC_4, AC_5, AC_6 are minimal antichains since the subsets are singleton atomic relations.

Definition 8**:** A partially ordered set is a join *semi lattice* if infimum exists for every binary subset.

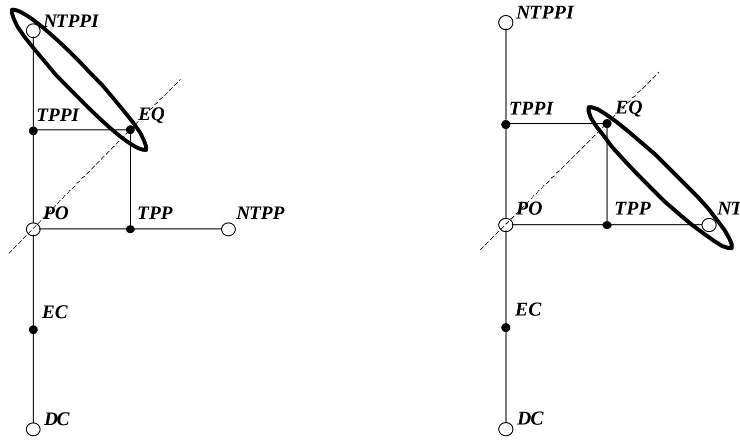
The partial order makes RCC-8 a *join semi lattice* since infimum is defined for every pair of atomic relations whereas supremum is not defined for every pair of atomic relations. For instance, the infimum for R = {TPP, TPPI} is PO, whereas the supremum for R = {NTPP, EQ} is not defined in the RCC-8 partial order.

We denote the dual of a relation R as the relation D_R that contains the mirror image of each atomic relation m in R with respect to the line joining PO and EQ in RCC-8 semi lattice. For instance, the dual of the relation R = {TPP, NTPP, EQ} is relation D_R = {TPPI, NTPPI, EQ}. The dual of any antichain in RCC-8 is the relation itself of another antichain that belongs to the set of 7 antichains in RCC-8.



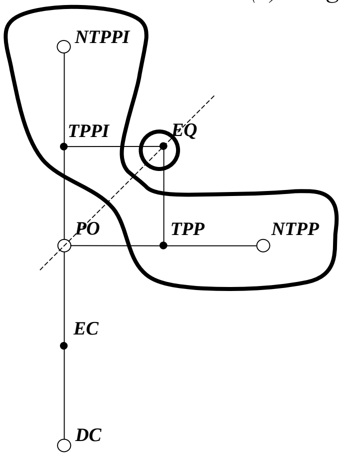
(i) (ii) (iii) (iv)

(a) Diagrams for condition depicting four antichains containing a pair of non-comparable atomic relations from the opposite sides of the PO-EQ line

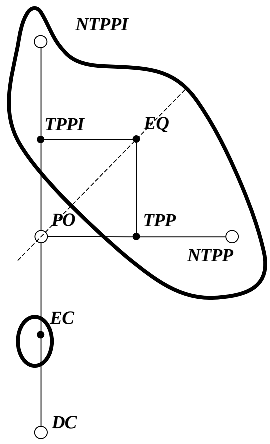


(i) (ii)

(b) Diagrams for $\hat{H}_8$



(c) Diagram for condition of Q_8



(d) Diagram for C_8

Figure 19. Pictorial representation of the properties of $\hat{H}_8$, C_8, Q_8

Three maximal tractable classes for RCC-8 namely $\hat{H}_8$, C_8 and Q_8 are empirically identified by researchers so far as discussed in chapter 1. We interpret the definitions of relations of $\hat{H}_8$, C_8 and Q_8 in terms of the existence of the least element. This interpretation gives a better insight into the structure of these classes. We consider two groups of relations for which the least element is not defined.

The first group (G_1) consists of relations R such that $R \cap \{DC, EC, PO\} = \emptyset$ and R contains at least one element from each side of the line joining PO and EQ in RCC-8 lattice in Figure 17. These relations are different unions of the four antichains namely AC_1 = {TPP, TPPI} AC_2 = {TPP, NTPPI} AC_3 = {NTPP, TPPI} AC_4 = {NTPP, NTPPI}.
The second group (G_2) of relations are the three antichains, namely AC_5 = {NTPP, EQ} AC_6 = {NTPPI, EQ} and AC_7 = {NTPP, NTPPI, EQ}.

Definition 9: Ĥ$_8$ [Renz 1999b] is the set of all subsets of **U** for which none of the following condition holds.

a. PO $\notin R$ and {(N)TPP,(N)TPPI} $\subseteq R$

b. TPP $\notin R$ {EQ,NTPP}$\subseteq R$

c. TPPI $\notin R$ {EQ,NTPPI} $\subseteq R$

Each of these conditions corresponds to the existence of an antichain. {(N)TPP, (N)TPPI} defines four antichains, namely AC_1, AC_2,… AC_4 and the infimum of each of these is PO. If R contains any of these and does not contain PO, EC or DC then R does not have a least element. Similarly, {EQ, NTPP} is an antichain and its infimum is TPP. Thus, by definition, Ĥ$_8$ does not include relations R for which *σ(R)* is not defined. For each element of Ĥ$_8$ the corresponding least element exists. There are relations outside Ĥ$_8$, for which the least element exists. For example, R = {DC, TPP, TPPI} is not in Ĥ$_8$ but *σ(R)* is DC. Another example of such relation is R = {PO, EQ, NTPP} and this is in Q$_8$ as shown below.

Definition 10: Q$_8$ [Renz 1999b] is the set of all subsets of **U** for which none of the following condition holds.

a. PO $\notin R$ and {(N)TPP,(N)TPPI} $\subseteq R$

b. EQ$\in R$ and PO $\notin R$ and $R \cap$ {TPP, NTPP, TPPI, NTPPI} $\neq \emptyset$

Unlike Ĥ$_8$, Q$_8$ allows antichain {EQ, NTPP} in the presence of PO, even when TPP is absent. Thus, Q$_8$ contains relations for which the least elements exist. We summarize the above-mentioned observations as following theorems.

Theorem 4: For any RCC-8 relation R, if $R \in$ Ĥ$_8$, then σ(R) exists.

Proof:

For a given RCC-8 relation, suppose σ(*R*) does not exist and $R \in \hat{H}_8$. As per the notation introduced above, either $R \in G_1$ or $R \in G_2$, which implies that *R* satisfies one of the following conditions:

a. *R* contains at least one of the four antichains AC_1, AC_2, AC_3, AC_4.

   $\Rightarrow R \cap \{PO\} = \emptyset$ and $R \cap \{AC_1, AC_2, AC_3, AC_4\} \neq \emptyset$.

   This violates the property (a) in the definition of $\hat{H}_8$ (definition 9).

b. *R* is among the antichain relations AC_5, AC_6, AC_7.

   $\Rightarrow$ either $\{NTPP, EQ\} \subseteq R$ and $TPP \notin R$ or $\{NTPPI, EQ\} \subseteq R$ and $TPPI \notin R$.

   This violates the properties (b) and (c) in the definition of $\hat{H}_8$ (definition 9).

This contradicts our assumption that σ(*R*) does not exist and $R \in \hat{H}_8$. This proves that *σ(R)* exists for all the relations in $\hat{H}_8$. ■

Theorem 5: For any RCC-8 relation *R,* if $R \in Q_8$, then σ(R) exists.

Proof:

For a given RCC-8 relation, suppose σ(*R*) does not exist and $R \in Q_8$.

As per the notation introduced above, either $R \in G_1$ or $R \in G_2$, which implies that *R* satisfies one of the following conditions:

a. *R* contains at least one of the four antichains AC_1, AC_2, AC_3, AC_4.

   $\Rightarrow R \cap \{PO\} = \emptyset$ and $R \cap \{AC_1, AC_2, AC_3, AC_4\} \neq \emptyset$.

   This violates the property (a) in the definition of Q_8 (definition 10).

b. *R* is among the antichain relations AC_5, AC_6, AC_7.

   $\Rightarrow$ if $EQ \in R$ and $R \cap \{PO\} = \emptyset$, $R \cap \{AC_1, AC_2, AC_3, AC_4, AC_5, AC_6, AC_7\}$.

   This violates the property (c) in the definition of Q_8 (definition 10).

This contradicts our assumption that σ(*R*) does not exist and $R \in Q_8$. This proves that *σ(R)* exists for all the relations in Q_8. ■

Definition 11: C_8 [Renz 1999b] is the set of all subsets of **U** for which none of the following condition holds.

a. $PO \notin R$ and $\{(N)TPP,(N)TPPI\} \subseteq R$

b. $EC \in R$ and $PO \notin R$ and $R \cap \{TPP, NTPP, TPPI, NTPPI, EQ\} \neq \emptyset$

It is not true that for $R \in C_8$, σ(*R*) exists. The antichains AC_5 and AC_6 are in C_8.

We extend the novel approach of least element-based interpretation of the tractable classes of RCC-8 proposed in this study to prove the closure of these classes under converse, intersection and composition.

Theorem 6: $\hat{H}_8$ and is closed with respect to converse, intersection and composition.

Proof:

There are two possibilities for a relation $R \notin \hat{H}_8$.

a. $\sigma(R)$ does not exist.

$\{DC, EC, PO\} \cap R = \emptyset$ and $\exists\, i$ such that $R \cap AC_i = AC_i$ and $\inf(AC_i) \notin R$ where $i \in [1,7]$.

b. $\sigma(R)$ exists.

$\{DC, EC\} \cap R \neq \emptyset$, $PO \notin R$ and $\exists\, i$ such that $R \cap AC_i = AC_i$ and $\inf(AC_i) \notin R$ where $i \in [1,7]$.

For a given relation $R \in \hat{H}_8$, suppose $R^{-1} \notin \hat{H}_8$. Due to symmetry with respect to PO-EQ line, if R^{-1} satisfies (a) or (b) then so does R. This proves that $\hat{H}_8$ is closed under converse.

Let R_1 and $R_2 \in \hat{H}_8$, such that $R = R_1 \cap R_2$, suppose $R \notin \hat{H}_8$. If $\sigma(R)$ does not exist then either the least element does exist for at least one of R_1, R_2 or are not equal if it exists. In either case, a contradiction to assumption. If $\sigma(R)$ exists then atleast one of R_1 and $R_2 \notin \hat{H}_8$. This proves that $\hat{H}_8$ is closed under intersection.

Let R_1 and $R_2 \in \hat{H}_8$ such that $R = R_1$ o R_2. Suppose $R \notin \hat{H}_8$.

For (a) to be satistfied, we observe that there are 16 CT entries where $R \cap \{DC, EC, PO\} = \emptyset$:

{TPP,NTPP}o{TPP,NTPP},{TPP,NTPP}oEQ, EQo{TPP,NTPP}

{TPPI,NTPPI}o{TPPI,NTPPI},{TPPI,NTPPI}oEQ, EQo{TPPI,NTPPI}

This implies that both R_1 and $R_2 \notin \hat{H}_8$.

For (b) to be satisfied, we observe that there are 12 more entries that satisfy this condition.

{DC,EC}o{TPPI, NTPPI,EQ},{TPP, NTPP,EQ}o{ DC, EC }.

In either case, both R_1 and $R_2 \notin \hat{H}_8$. This proves that $\hat{H}_8$ is closed under composition. ■

Theorem 7: Q_8 is closed with respect to converse, intersection and composition.

Proof:

For a given relation $R \in Q_8$, suppose $R^{-1} \notin Q_8$. There are two possibilities for R^{-1}.

Case I. $\sigma(R^{-1})$ does not exist. This means that R^{-1} contains at least one of the four antichains: AC_1, AC_2, AC_3, AC_4 with an additional property that $\{DC, EC, PO\} \cap R^{-1} = \emptyset$. This implies that the dual of the corresponding antichain(s) is present in the relation R. This violates the basic assumption that R is a Q_8 relation.

Case II. $\sigma(R^{-1})$ exists. This means R^{-1} satisfies either of the following conditions:

a. $\{DC, EC\} \cap R^{-1} \neq \emptyset$, $PO \notin R^{-1}$, R^{-1} contains at least one of the four antichains AC_1, AC_2, AC_3, AC_4.

b. $\{DC, EC\} \cap R^{-1} \neq \emptyset$, $EQ \in R^{-1}$ and $PO \notin R^{-1}$ and $R^{-1} \cap \{TPP, NTPP, TPPI, NTPPI\} \neq \emptyset$.

R also maintains the same structure as that of R^{-1} due to the dual behavior of relations involved in a and b above. This structure of R violates the basic assumption that $R \in Q_8$. This proves that Q_8 is closed under converse.

Let R_1 and R_2 be two non-atomic Q_8 relations, such that $R_3 = R_1 \cap R_2$. Suppose R_3 is not a Q_8 relation. This means that R_3 satisfies at least one of the following two conditions:

Case I. $\sigma(R_3)$ does not exist. There is only one possibility for the structure of R_3.

$\{DC, EC, PO\} \cap R_3 = \emptyset$ and R_3 contains at least one of the four antichains AC_1, AC_2, AC_3, AC_4. The absence of least element of the common part of R_1 and R_2 means that $\sigma(R_1) \neq \sigma(R_2)$. Let us denote R as a relation in the set $\{R_1, R_2\}$. If $\sigma(R)$ does not exist in at least one case, violates the basic assumption for R. If $\sigma(R) \neq PO$ implies that R contains at least one antichain without PO, which again violates the basic assumption. If PO is present in both, this implies that $\sigma(R_3) = PO$, i.e. $R_3 \in Q_8$, a violation to our assumption.

Case II. $\sigma(R_3)$ exists but $R_3 \notin Q_8$. Two possibilities arise:

a. $\{DC, EC\} \cap R \neq \emptyset$, PO is not present in at least one of R_1, R_2 and R_3 is a union of at least one of the antichains AC1, AC2, AC3, AC4. This violates the assumption that R_1, R_2 are Q_8 relations.

b. $\{DC, EC\} \cap R \neq \emptyset$, PO is absent in at least one of R_1 and R_2, R_1 and $R_2 \cap \{TPP, NTPP, TPPI, NTPPI, EQ\} \neq \emptyset$, R_1 and $R_2 \cap \{DC, EC\} \neq \emptyset$. This violates second condition in the definition of Q_8 for at least one of the two relations R_1, R_2.

This proves that Q_8 is closed under intersection.

Let R_1 and R_2 be two non-atomic Q_8 relations, such that $R_3 = R_1 \circ R_2$. Suppose R_3 is not a Q_8 relation. This means that R_3 satisfies at least one of the following two conditions:

Case I. $\sigma(R_3)$ does not exist: This violates theorem 4, that the least element is defined for any relation resulting from the composition of two relations with least element.

Case II. $\sigma(R_3)$ exists. This means that PO $\notin R_3$, EQ $\in R_3$, {TPP, NTPP, TPPI, NTPPI} $\cap R_3 \neq \varnothing$. For this type of relation to result due to composition, we summarize the possibilities for the structure of the relations R_1 and R_2:

a. {TPP, NTPP, EQ} o {DC, EC, TPP, NTPP, EQ}]
b. {DC, EC, TPPI, NTPPI, EQ} o { TPPI, NTPPI, EQ}

In the above two possibilities, the relations {TPP, NTPP, EQ} and {TPPI, NTPPI, EQ} are not present in Q_8 but a least element is defined. We conclude that either of the two relations R_1, R_2 do not belong to Q_8. This violates the basic assumption that R_1 and $R_2 \in Q_8$. This proves that Q_8 is closed under composition. ■

This result will be helpful in the next Section for providing analytical proof for tractability of $\hat{H}_8$, C_8 and Q_8. The infimum property of RCC-8 lattice has motivated us to classify the set of RCC-8 relations based on the important observation – the existence of least element.

2.4 Analytical proof for tractability

In the foregoing discussion, we provide a novel interpretation of RCC-8 based on the existence of the least lattice-elements. We intend to propose an entirely new formalism on the basis of observations in the previous two Sections. For any framework, one needs to define the operations of converse and composition to answer basic reasoning task of satisfiability. In this Section, we formalize the unary operation of converse and binary operation of composition for relations having the least element. We outline some of the important characteristics of $\sigma(R)$.

Theorem 8: For any RCC-8 relation *R*, if the least element for *R* exists, then the least element for R^{-1} also exists and defined as $\sigma(R^{-1}) = (\sigma(R))^{-1}$.

*Proof***:**

We divide the set of RCC-8 relations with $\sigma(R)$ defined into two groups based on the symmetry with respect to an imaginary line passing through PO and EQ as shown in the lattice structure in Figure 17 and provide the proof for this theorem in following two steps:

Case I: Suppose a relation R contains any of DC, EC or PO. It is a clear observation from the RCC-8 lattice that $\sigma(R)$ is defined for this R. Converse of the relation R will also contain DC, EC, or PO as the infimum element since DC, EC, PO basic relations are self-inverse base relations. Hence for relations containing self-inverse atomic relations as $\sigma(R)$, the infimum of the converse of the relation also belongs to the converse relation.

Case II: Suppose a relation does not contain DC, EC or PO. The candidate atomic relations for $\sigma(R)$ are TPP or NTPP(TPPI or NTPPI). Converse of such a relation will contain TPPI or NTPPI(TPP or NTPP) as possible $\sigma(R)$. Hence for relations containing non-self inverse atomic relations as $\sigma(R)$, the inverse of $\sigma(R)$ is same as infimum of the converse of the relation. This means that converse of any relation corresponds to a symmetry with respect to PO-EQ line. Moreover, the converse operation preserves the precedence relation, that is, for atomic relations s_1 and s_2 in RCC-8 if $s_1 \preccurlyeq s_2$ then $s_1^{-1} \preccurlyeq s_2^{-1}$. The proof directly follows from this.

■

Let R = { DC, EC, TPP, NTPP } such that R contains both the types of atomic relations, namely self-inverse and non-self-inverse.

R^{-1} = {DC, EC, TPPI, NTPPI }

$\sigma(R^{-1})$ = DC

= $(\sigma(R))^{-1}$

Consider a relation R = {TPP, TPPI} for which $\sigma(R)$ is not defined.

R^{-1} = { TPP, TPPI }, $\sigma(R^{-1})$ is not defined.

By inspecting the entries in the composition table, we can conclude the following. The proofs for these lemmas are straightforward and can be verified from the composition table.

Lemma 1: For two atomic relations s_1 and s_2, the least element of s_1 o s_2 always exists.

Proof:

Each cell entry in the composition table corresponds to an atomic relation s_1 along a row and s_2 along a column. It is observed that each cell entry among all the 64 entries is a relation with $\sigma(R)$ defined.

■

Let s_1 = EC, s_2 = EC, s_1 o s_2 = { DC, EC, PO, TPP, TPPI, EQ }, σ(s_1 o s_2) = DC. It is observed that straight examples of RCC-8 relations with *σ(R)* not defined, namely the set of antichains are not among any of the cell entries. For example, no cell entry is the relation { TPP, TPPI }.

Lemma 2: For three atomic relations s_1, s_2 and s_3, if $s_2 \preccurlyeq s_3$ then $\sigma(s_1 \circ s_2) \preccurlyeq \sigma(s_1 \circ s_3)$.

Proof:

RCC-8 CT consists of eight atomic relations arranged as neighbours along rows and columns. Composing neighboring relations gives neighboring sets of results. It is observed that each cell entry is a relation with *σ(R)* defined. Composing an atomic relation along a row with an atomic relation in first column gives a relation at the corresponding cell entry. If same row relation is composed with that in next column, the resulting relation will start either at *σ(R)* of previous cell entry relation or it will start at an atomic relation that succeeds *σ(R)* of the previous cell entry relation by few steps. Hence as the atomic relation along a row is successively composed with relations along the columns from left to right, the starting relation of resulting relations will either remain same or move right to a succeeding atomic relation but will never move left to a preceding relation.

■

Let s_1 = PO, s_2 = DC and s_3 = TPP such that DC $\preccurlyeq$ TPP.

σ(s_1 o s_2) = σ(PO o DC)

= σ{DC, EC, PO, TPPI, NTPPI}

= DC

σ(s_1 o s_3) = σ(PO o TPP)

= σ(PO, TPP, NTPP)

= PO

DC $\preccurlyeq$ PO clearly shows that $\sigma(s_1 \circ s_2) \preccurlyeq \sigma(s_1 \circ s_3)$.

Lemma 3: For any triad (s_1, s_2, s_3) of three atomic relations s_1, s_2 and s_3 such that $s_3 \notin s_1 \circ s_2$, the following holds true except for two triads, namely (TPP, TPP, EQ) and (TPPI, TPPI, EQ).

if $\sigma(s_1 \circ s_2) \preccurlyeq s_3$ then either $s_1 \prec \sigma(s_3 \circ s_2^{-1})$ or $s_2 \prec \sigma(s_1^{-1} \circ s_3)$

(We denote $r \prec s$ for the case where $r \preccurlyeq s$ and $r \neq s$. In other words, r and s are distinct atomic relations and r precedes s in the semi lattice.)

Proof:

Consider the relation R in an entry in the composition table corresponding to an atomic relation s_1 along the row and the atomic relation s_2 along a column. Any atomic relation s_3 that is not present in R forms a singleton triplet with s_1 and s_2 that is not path consistent.

■

Let s_1 = EC, s_2 = TPP. From the composition table, s_1 o s_2 = {EC, PO, TPP, NTPP}. If s_3 is to be selected such that $s_3 \notin s_1$ o s_2 and σ $(s_1$ o $s_2) \preccurlyeq s_3$ then s_3 can be either TPPI, NTPPI or EQ.

Let s_3 = TPPI, σ(s_3 o s_2^{-1}) = σ(TPPI o TPPI)

= σ(TPPI, NTPPI)

= TPPI.

Clearly, $s_1 \prec$ σ(s_3 o s_2^{-1}) is satisfied since EC $\preccurlyeq$ TPPI.

Similarly for s_3 = NTPPI or EQ, $s_1 \prec$ σ(s_3 o s_2^{-1}) also holds.

Whereas if σ $(s_1$ o $s_2) \preccurlyeq s_3$ is not considered i.e. if s_3 = DC then both the conditions -

$s_1 \prec$ σ(s_3 o s_2^{-1}) and $s_2 \prec$ σ(s_1^{-1} o s_3) are violated.

We use the above observations in proving the following results.

Theorem 9: If R_1 and R_2 are two RCC-8 relations, such that σ(R_1) and σ(R_2) exist, then the least element of (R_1 o R_2) exists and is given by the following.

$$\sigma(R_1 \text{ o } R_2) = \sigma(\sigma(R_1) \text{ o } \sigma(R_2))$$

Proof: By Lemma 2,

$$\sigma(R_1) \text{ o } \sigma(R_2) \preccurlyeq s \text{ o } \sigma(R_2), \forall s \in R_1.$$

By Lemma 1, σ(σ(R_1) o σ(R_2)) exists. Thus σ(σ(R_1) o σ(R_2)) precedes every element of s_1 o σ(R_2), $\forall s_1 \in R_1$ and similarly, of σ(R_1) o s_2, $\forall s_2 \in R_2$.

Hence σ(σ(R_1) o σ(R_2)) precedes elements of $(s_1$ o $s_2)$, $\forall s_1 \in R_1$ and $\forall s_2 \in R_2$.

So σ(σ(R_1) o σ(R_2)) $\preccurlyeq s$, $\forall s \in (R_1$ o $R_2)$. By definitions of least element and composition, σ(σ(R_1) o σ(R_2)) $\in R_1$ o R_2. Hence σ(σ(R_1) o σ(R_2)) is the least element of R_1 o R_2.

■

For instance, suppose R_1 = {EC, PO} and R_2 = {TPP, NTPP}.

R_1 o R_2 = {EC, PO, TPP, NTPP}

σ$(R_1$ o $R_2)$ = EC

$\sigma(R_1)$ o $\sigma(R_2)$ = {EC, PO, TPP, NTPP}

$\sigma(\sigma(R_1)$ o $\sigma(R_2))$ = EC

= $\sigma(R_1$ o $R_2)$

Theorem 10: For any three RCC-8 relations R_1, R_2 and R_2', such that the least elements of R_1, R_2 and R_2' exist. If $\sigma(R_2) \preccurlyeq \sigma(R_2')$ then $\sigma(R_1$ o $R_2) \preccurlyeq \sigma(R_1$ o $R_2')$ and $\sigma(R_2$ o $R_1) \preccurlyeq \sigma(R_2'$ o $R_1)$

Proof: From Theorem 9, we have

$\sigma(R_1$ o $R_2) = \sigma(\sigma(R_1)$ o $\sigma(R_2))$ and

$\sigma(R_1$ o $R_2') = \sigma(\sigma(R_1)$ o $\sigma(R_2'))$.

From Lemma 2, and from $\sigma(R_2) \preccurlyeq \sigma(R_2')$ we have

$\sigma(\sigma(R_1)$ o $\sigma(R_2)) \preccurlyeq \sigma(\sigma(R_1)$ o $\sigma(R_2'))$

So $\sigma(R_1$ o $R_2) \preccurlyeq \sigma(R_1$ o $R_2')$ ■

For instance, suppose R_1 = { TPPI, NTPPI } so that $\sigma(R_1)$ = TPPI

and R_2 = { EC, PO } so that $\sigma(R_2)$ = EC

and R_2' = { TPP, EQ } so that $\sigma(R_2')$ = TPP

It is obvious that $\sigma(R_2) \preccurlyeq \sigma(R_2')$, since EC $\preccurlyeq$ TPP.

R_1 o R_2 = { EC, PO, TPPI, NTPPI } and R_1 o R_2' = { PO, TPP, TPPI, NTPPI, EQ }

Hence $\sigma(R_1$ o $R_2) \preccurlyeq \sigma(R_1$ o $R_2')$.

Theorem 11: For relations R_1, R_2 and R_3 with least element defined, if (R_1, R_2, R_3) forms a path consistent triad then ($\sigma(R_1)$, $\sigma(R_2)$, $\sigma(R_3)$) forms a path consistent triad.

Proof:

Let (R_1, R_2, R_3) be a path consistent triad. By Definition of path consistency (chapter 1), $R_3 \subseteq R_1$ o R_2. Hence, $\sigma(R_3) \in R_1$ o R_2.

Let us assume that the triad ($\sigma(R_1)$, $\sigma(R_2)$, $\sigma(R_3)$) is not path consistent. Then $\sigma(R_3) \notin \sigma(R_1)$ o $\sigma(R_2)$. Since the least element of R_1 o R_2 is $\sigma(\sigma(R_1)$ o $\sigma(R_2))$ and $\sigma(R_3) \in R_1$ o R_2, we have $\sigma((\sigma(R_1)$ o $\sigma(R_2)) \preccurlyeq \sigma(R_3)$. By Lemma 3, $\sigma(R_1) \prec \sigma(\sigma(R_3)$ o $\sigma(R_2^{-1}))$ or, $\sigma(R_2) \prec \sigma(\sigma(R_1^{-1})$ o $\sigma(R_3))$, except when $\sigma(R_1) = \sigma(R_2)$ = TPP(I) and $\sigma(R_3)$= EQ.

This implies that except of these two cases either $R_1 \not\subset R_3$ o R_2^{-1} or $R_2 \not\subset R_1^{-1}$ o R_3, which contradicts that (R_1, R_2, R_3) is a path consistent triad. For the exceptional cases referred above, it is not possible to have a path consistent triad (R_1, R_2, R_3) with $\sigma(R_1) = \sigma(R_2) =$ TPP(I) and $\sigma(R_3)$= EQ.

`

■

Above-mentioned properties for a triplet of relations can be extended to a general *n*-node network with relations for which least element is defined. A consistency-checking method only determines whether a given instance of a constraint network has a solution or not, without determining one. In other words, consistency-check and determination of a solution are two separate steps in qualitative reasoning frameworks. In the case of constraint networks with atomic relations as the constraint labels, these steps merge together. In case the singleton network is path consistent, it is in turn the solution. On the contrary, for disjunctive constraint networks, the general strategy followed is that a certain level of local consistency is sufficient to decide consistency for a network with relations from a known tractable class. After the step of consistency check, backtrack search is the de facto standard method to determine the solution(s). Consistency-check method for RCC-8 can be casted as satisfaction of two conditions namely the local consistency at level of triads (path consistency) and existence of the least element of every constraint in the network. As a byproduct of checking consistency, the least element-based interpretation is constructing a solution as well. However, this method is not applicable when the least element is not defined for any of the constraints in the network. We now state our main result.

Theorem 12: Consider a RCC-8 network satisfying the following:

a. For every constraint R_{ij}, $\sigma(R_{ij})$ exists.
b. The network is path consistent.

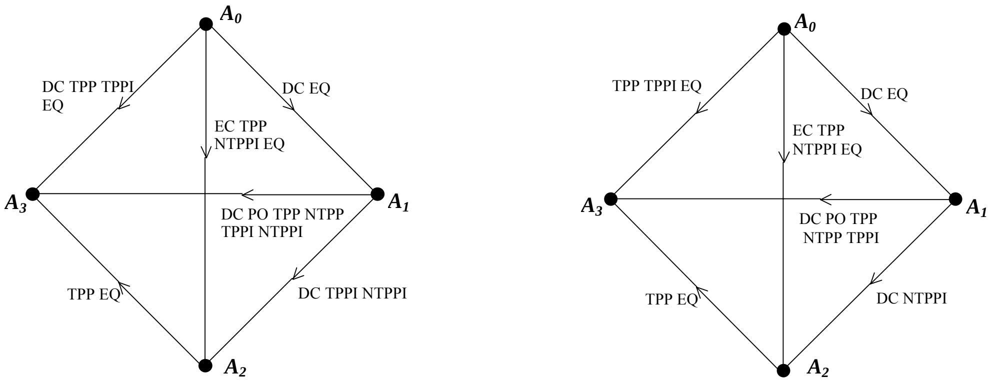
The singleton labeling of this network obtained by replacing R_{ij} by $\sigma(R_{ij})$ is a consistent singleton labeling. Thus a solution can be obtained in polynomial time.

Proof: The proof is a direct consequence of Theorem 11. Since the network is path consistent, so is every triplet. We can replace every constraint by its least element to get a singleton labeling. We know that for singleton constraints path consistency is necessary and sufficient condition for consistency. Hence the network is consistent.

Checking condition a. takes constant time, checking condition b. takes $O(n^3)$ time, and replacing every edge by its least element takes $O(n^2)$. Hence the method requires $O(n^3)$ time.

■

It may happen that while enforcing path consistency, the network may have a constraint for which least element may not exist. Consider the example shown in Figure 20.



(a) **(b)**

(a) An example network with relations for which the least elements exist.
(b) Enforcing path-consistency, A_0A_3 changes to a relation for which least element does not exist.

Figure 20. A J_8 network for which least element is not preserved by path consistency.

Enforcing path consistency on a network with constraints from a class of relations does not bring the relations that do not belong to the class. Closure with respect to the three operations individually is a very stringent condition. Closure under path consistency is much more relaxed condition. Any set of relations that is closed under the three operations will be closed under path consistency but vice versa may not always hold. The closure under path consistency can be ignored if the class is closed with respect to three operations. The main idea is to preserve the condition a. of Theorem 12 for every constraint of the network. It is possible either when the class is closed or when we do not have to enforce path consistency further.

Theorem 13: For any RCC-8 network with relations in $\hat{H}_8$ (or, Q_8) path consistency decides consistency.

Proof: If RCC-8 network with relations in $\hat{H}_8$ is path consistent, then replacing each R_i by $\sigma(R_i)$ is a consistent singleton labeling by theorem 12. This is a solution of the original network and hence it can be concluded that original network is consistent.

■

Same argument of providing analytical proof of tractability cannot be extended to C_8, since there are two relations in C_8 namely {EQ, NTPP} and {EQ, NTPPI} for which $\sigma(R)$ is not defined. We give the analytical proof of tractability in a different manner in Section 2.9.

2.5 The largest tractable set in RCC-8 - J_8

It is interesting to study the effect of the properties of $\sigma(R)$ in tractability analysis of relations in RCC-8 beyond the known set of maximal tractable classes. In Section 3, it is seen that there exist relations outside $\hat{H}_8$, C_8 and Q_8 such that $\sigma(R)$ for these relations is defined. From theorem 11, we know that existence of $\sigma(R)$ and path consistency of the network are two critical conditions for establishing tractability. Thus these observations prompt us to define a new set, namely J_8, for which some tractability result can be derived.

Definition 12: The set $J_8 \subseteq P$ is defined as follows:

$$J_8 = \{ R \mid \text{the least element of } R,\ \sigma(R), \text{ exists}\}$$

It is found that J_8 has 235 relations and so there are only twenty relations for which $\sigma(R)$ is not defined. These relations are listed in Appendix -A. Let us denote NJ_8 as the complement of J_8 in the complete set of RCC-8 relations. The following results directly hold.

Proposition 1: $\hat{H}_8$ and Q_8 are subsets of J_8. C_8 is not a subset of J_8.

Except for two relations namely {NTPP, EQ} and {NTPPI, EQ} all other relations of C_8 are members of J_8.

J_8 is not closed with respect to intersection. The following is the counterexample.

{DC, TPP, TPPI} $\cap$ {EC, TPP, TPPI} = {TPP, TPPI}

Hence, the set J_8 is not a class in the sense of being closed with respect to the three operations.

Proposition 2: Given a path-consistent J_8 network, a solution can be obtained in polynomial time by replacing each edge relation R_i by $\sigma(R_i)$.

Though J_8 is not closed with respect to intersection, the path consistency condition ensures that we do not need any of these operations. However, the example in Figure 20 shows that during the process of enforcing path consistency of J_8 network, we may generate relations that is not in J_8. Nevertheless, we can still claim that satisfiability can be determined in polynomial time if the network is in J_8 after it is made path consistent. This extends the tractability beyond the maximal classes identified earlier. The network in Figure 21 is one such path consistent network with relations in J_8 but not from any of the three maximal tractable classes. The method is not complete in the sense that path consistency and being in J_8 suffice to establish satisfiability but unsatisfiability cannot be established if these conditions do not hold.

We enlist some observations on the sets J_8 and NJ_8:

- $J_8 \cup NJ_8$ = RCC-8
- NJ_8 contains 20 relations
- J_8 contains 235 relations: with 177 relations of the union set of $\hat{H}_8$, C_8 and Q_8 except for two relations of C_8, which are in NJ_8.
- J_8 closed under composition and converse but not under intersection.
- 91% of the triplets (not in any of the three maximal tractable classes) are path-consistent, 9% are not path-consistent.
- J_8 is not a tractable class but most of the triplets are path-consistent.
- C_8 relations {NTPP,EQ} and {NTPPI,EQ} $\in NJ_8$.
- Renz[1999] shows that $\hat{H}_8 \cup$ {EC,NTPP,EQ} contains {NTPP,EQ}. It can be seen that {EC,NTPP,EQ} $\in J_8$, {NTPP,EQ} $\in NJ_8$.

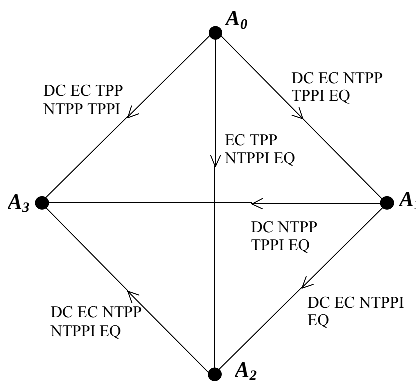


Figure 21. Example network with relations in J_8 but not in any of $\hat{H}_8$, C_8 and Q_8.

2.6 Least Element Instantiation

In [Gerevini and Renz 1998], an algorithm is proposed to find a specific solution for tractable classes when the network is path consistent. The constraints are instantiated to singletons in a sequence of refinements with intermediate steps of enforcing path consistency. Later, in

[Renz, 1999], the second author introduces the concept of refinement matrix and proposes an improvement of this algorithm by replacing the computationally expensive step of path consistency with a table lookup. The table lookup requires *a priori* computation of the refinement matrix. This matrix stores all possible path consistent triplets of refinements for a set of relations. Assuming that the lookup takes constant time, the computational complexity of this improvement is $O(n^2)$ whereas that of the original algorithm is $O(n^3)$. These algorithms converge to singleton instantiations of the given network. In this Section, we propose a simple solution technique that avoids table lookup as well as repeated enforcement of path consistency. A systematic analysis of the existing method highlights the nicety of the proposed algorithm.

***Definition 13*: ** A *refinement* of a RCC-8 relation β is a relation β' such that $\beta' \subseteq \beta$.
For instance, all possible refinements R' for the relation R = {DC, EC, PO, TPP, NTPP} are the possible subsets namely {DC}, {EC}, {PO}, {TPP}, {NTPP}, {DC,EC}, {DC,PO}, {DC, TPP}, {DC,NTPP}, {EC,PO}, {EC,TPP}, {EC,NTPP}, {PO,TPP}, {PO,NTPP}, {TPP,NTPP}, {DC,EC,PO}, {DC,EC,TPP}, {DC,EC,NTPP}, {DC,PO,TPP}, {DC,PO,NTPP}, {EC,PO,TPP}, {DC,TPP,NTPP}, {EC,PO,NTPP}, {PO,TPP,NTPP}, {DC,ECPO,TPP}, {DC,ECPO,NTPP}, {DC,EC,TPP,NTPP}, {DC,PO,TPP,NTPP}, {EC,PO,TPP,NTPP}, {DC,EC,PO,TPP,NTPP}. The set of refinements excludes the empty set.

***Definition 14*: ** *α-refinement* of a relation β is a refinement β' obtained by the following rule.

if $\beta \cap \alpha = \varnothing$, then $\beta' = \beta$;

otherwise, $\beta' = \beta \cap \alpha$.

Suppose α = DC, then α-refinement of the subset {DC, EC, TPP} is {DC}. DC-refinement of a relation that does not contain DC, such as {PO,TPP,NTPP} is the relation itself.

***Definition 15*: ** *α-elimination* of a relation β is a refinement β' obtained by the following rule.

if $\beta = \alpha$, $\beta' = \beta$;

if $\beta \cap \alpha = \varnothing$, then $\beta' = \beta$;

otherwise $\beta' = \beta \setminus (\beta \cap \alpha)$.

Suppose α = DC, then α-elimination of the subset {DC, EC, TPP} is the relation {EC, TPP}. DC-elimination of the relation that does not contain DC, {PO, TPP, NTPP} is the relation itself. We observe that the two concepts of α-refinement and α-elimination have the same meaning for a singleton relation or for the relations that do not contain α. These concepts hold for constraint as well as the set of constraints. The two sets of relations obtained by α-refinement and α-elimination are same, namely {DC}, {EC}, {PO}, {TPP}, {NTPP}, {EC,PO}, {EC,TPP}, {EC,NTPP}, {PO,TPP}, {PO,NTPP}, {TPP,NTPP}, {EC,PO,NTPP}, {PO,TPP,NTPP}, {EC,TPP,NTPP}, {EC,PO,TPP,NTPP}. The same concepts are applicable to the cases when α is a non-singleton relation.

The algorithm proposed in [Gerevini and Renz 1998] reduces the constraints to atomic relations by five steps of consistency preserving refinements in a sequence of DC-refinement, EC-refinement, EQ-elimination, PO-refinement and NTPP-refinement. It may be noted that a relation having NTPPI or TPPI can be equivalently converted to a relation with NTPP or TPP by reversing the orientation of the edge. This strategy of reversing the orientation is valid only when we do not have combinations of the sort ((N)TPP, (N)TPPI). As discussed earlier, such combinations are ruled out in $\hat{H}_8$, C_8, and Q_8. Moreover, it also ensures that the five steps of refinements are sufficient to get singleton labeling for the network of these classes. It is proved [Gerevini and Renz 1998] that if the network is path consistent this process guarantees a consistent singleton labeling. We describe pseudocode of the algorithm (Table 2.)

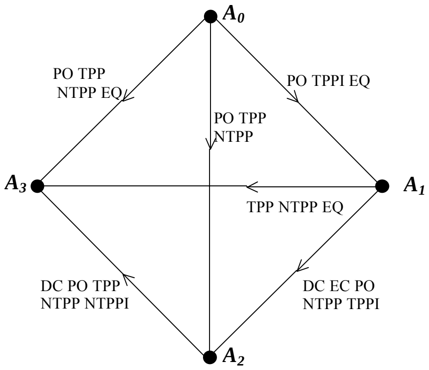
```
Algorithm: Scenario(N)
Input:      N- A n-node path consistent RCC-8 network with A_ij either in Ĥ8, C8, or Q8.
Output:     S- A consistent scenario for the constraint network N
RefinementSet = {DC, EC, EQ, PO, NTPP}
while (RefinementSet is not empty} do
        r = first element of RefinementSet
        reset ModifiedFlag
        for each edge constraint A_ij do
             if ( r ≠ EQ )
               do r-refinement of A_ij
            else
               do r-elimination of A_ij
               set ModifiedFlag
            endif
        enddo
        if  ModifiedFlag
            enforce path_consistency(N)
```

if (N is path_consistent) and (each A_{ij} becomes a singleton)
$S = N$, output S, exit
Delete *r* from *RefinementSet*
enddo
$S = N$, output S

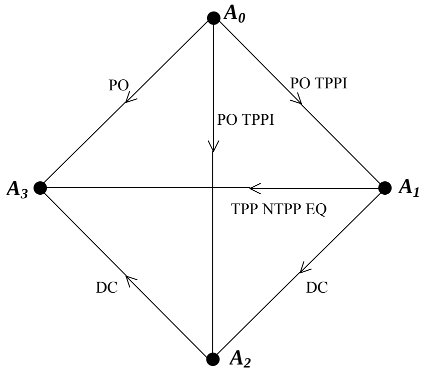
Table 2. Refinement based algorithm to compute a solution.

We illustrate the algorithm *Scenario* for an example C_8 network in Figure 22 where (b) is the result of DC-refinement on the edges A_{23} and A_{12} followed by path-consistency step. Since EC is not present in any edge, the EC-refinement is not applicable. EQ-eliminating EQ of A_{13} followed by enforcing path consistency and similarly using PO-refinement and NTTP-refinement steps, we get network as shown in figure 22(c).

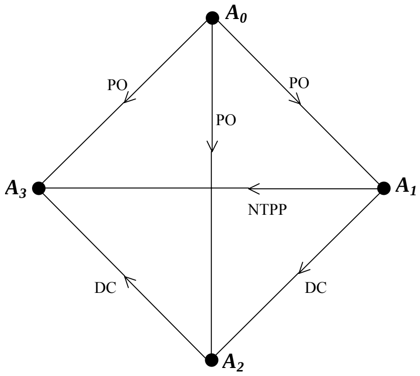
One can see that each refinement step requires making one pass over the set of edges and there are 5 such passes over the entire network. Although EC-refinement is not necessary, the algorithm has to go through one pass to realize that EC-refinement does not change network. The algorithm requires 4 path consistency steps. This algorithm is of $O(n^3)$ time since each of the refinement steps takes $O(n^2)$ time and the path consistency step takes $O(n^3)$ time. The algorithm requires path consistency to be enforced at six steps in the worst case.



(a) Starting network



(b) After DC-refinement and EC-refinement



(c) Final network

Figure 22. Output of *Scenario( )* for an example C_8 constraint network.

Above algorithm is improved by replacing the path consistency enforcement steps with table-lookups [Renz 1999].

Definition 16[Renz 1999]***:*** A *refinement matrix* M of $S \subseteq 2^A$ has $|S| \times 2^{|A|}$ Boolean entries such that $\forall s \in S$, $R \in 2^A$, M[S][R] = true only if $R \subseteq S$.

For example, if the relation {DC, EC, PO, TPP} is allowed to be refined only to the relations {DC, TPP} and {DC}, then M[{DC,EC,PO,TPP}][*R*] is true only for *R* = {DC,TPP} and for *R* = {DC} and false for all other relations $R \in 2^A$. M is called the basic refinement matrix if M[*S*][*R*] = true if and only if *S* = *R*.

A refinement matrix is computed in advance and is used for the lookup step. This approach is an eager method where in all possible refinements of every path consistent triplet are precomputed and stored as a lookup table. For a given set of relations *S*, all possible path-consistent refinements of triplets (R_{12}, R_{23}, R_{13}) of *S* are computed and compiled in a tabular form. Once such a table is ready, for any given triplet, the path consistent refinement can be obtained by a table lookup and hence enforcing path consistency is avoided. For example, the refinement matrix for the example constraint shown in Figure 22 is shown below.

Constraints	**Entries in the Refinement matrix, M**
A_{01}	(PO TPPI EQ) (**PO**) (**TPPI**) (PO TPPI) (**EQ**) (PO EQ) (TPPI EQ) (PO TPPI EQ)
A_{02}	(PO TPP NTPP) (**PO**) (**TPP**) (PO TPP) (**NTPP**) (PO NTPP) (TPP NTPP) (PO TPP NTPP)
A_{03}	(PO TPP NTPP EQ) (**PO**) (**TPP**) (PO TPP) (**NTPP**) (PO NTPP) (TPP NTPP) (PO TPP NTPP) (**EQ**) (PO EQ) (TPP EQ) (PO TPP EQ) (NTPP EQ) (PO NTPP EQ) (TPP NTPP EQ) (PO TPP NTPP EQ)
A_{12}	(DC EC PO NTPP TPPI) (**DC**) (**EC**) (DC EC) (**PO**) (DC PO) (EC PO) (DC EC PO) (**NTPP**) (DC NTPP) (EC NTPP) (DC EC NTPP) (PO NTPP) (DC PO NTPP) (EC PO NTPP) (DC EC PO NTPP) (TPPI) (DC TPPI) (EC TPPI) (DC EC TPPI) (PO TPPI) (DC PO TPPI) (EC PO TPPI) (DC EC PO TPPI) (NTPP TPPI) (DC NTPP TPPI) (EC NTPP TPPI) (DC EC NTPP TPPI) (PO NTPP TPPI) (DC PO NTPP TPPI) (EC PO NTPP TPPI) (DC EC PO NTPP TPPI)
A_{13}	(TPP NTPP EQ) (**TPP**) (**NTPP**) (TPP NTPP) (**EQ**) (TPP EQ) (NTPP EQ) (TPP NTPP EQ)
A_{23}	(DC PO TPP NTPP NTPPI) (**DC**) (**PO**) (DC PO) (**TPP**) (DC TPP) (PO TPP) (DC PO TPP) (**NTPP**) (DC NTPP) (PO NTPP) (DC PO NTPP) (TPP NTPP) (DC TPP NTPP) (PO TPP NTPP) (DC PO TPP NTPP) (NTPPI) (DC NTPPI) (PO NTPPI) (DC PO NTPPI) (TPP NTPPI) (DC TPP NTPPI) (PO TPP NTPPI) (DC PO TPP NTPPI) (NTPP NTPPI) (DC NTPP NTPPI) (PO NTPP NTPPI) (DC PO NTPP NTPPI) (TPP NTPP NTPPI) (DC TPP NTPP NTPPI) (PO TPP NTPP NTPPI) (DC PO TPP NTPP NTPPI)

Figure 23. Refinement matrix for example in Figure 22.

The refinement matrix is computed on the same lines as the algorithm [Renz 1999]. The refinement matrix is a sparse matrix; hence we have reproduced only the relations for which each given relation in the example has non-zero entries. Analyzing the entries in M, we find that for almost all relations at least one of the entries in M is a base relation that is related to the original relation according to a particular refinement scheme. The singleton entries are shown as bold relations. For the constraint A_{13}, three singletons are possible labels, namely TPP, NTPP and EQ. The singleton label NTPP is chosen as per the refinement order. Finding the consistent scenario is a trivial table lookup here, since all relations have a base relation as an entry in M. If the relations that do not have a base relation as an entry in M, a base relation is assigned according to the refinement scheme. If we consider all the relations, then there are at most n^3 possible triples of relations, at most n^3 possible refinements of each triple and a total of n^2 edges in a network. Thus, a rough estimation of the worst-case running time leads to $O(n^8)$. This is exponential in the number of constraints.

This method requires substantial computational effort for the preparation of the refinement matrix and the refinement matrix is going to be very large and a sparse matrix. Unless the lookup is handled judiciously, the lookup step may still require nontrivial computational effort. Correctness of algorithm *Scenario* is established in Renz [Renz 1999], wherein it is claimed that this algorithm when applied to a path consistent set of constraints over $\hat{H}_8$, Q_8 or C_8 guarantees a consistent scenario. Correctness of the modified algorithm is obvious as the refinement matrix M precomputes all possible consistent refinements of all triplets.

The analysis of the structures of J_8, $\hat{H}_8$, Q_8 and C_8 presented in the previous section reveals that neither a path consistency algorithm nor a table lookup is necessary between different stages of refinements. Once this intermediate step is avoided, even sequentially refining is no longer relevant. As a result, one can have a very efficient algorithm to get a solution. We describe the new algorithm for $\hat{H}_8$ and Q_8 below.

Algorithm: *Construct_Solution(N)*
Input: *N*- A n-node path-consistent network with relations R_{ij}
Output: *S*- A consistent scenario for *N* with relations S_{ij}
do for every constraint $R_{ij} \in N$
    $S_{ij} = \sigma(R_{ij})$
enddo

Table 3. Least element instantiation based algorithm to compute solution.

By theorem 12, it is clear that least element instantiation preserves path consistency. Path consistent singleton label is a solution for the network and hence *Construct_Solution* algorithm correctly determines a solution for a network in $\hat{H}_8$, or in Q_8 by the process of least element instantiation. The algorithm works in $O(n^2)$ time as the instantiation is done for every edge and the number of edges in the network is $O(n^2)$. It is still much more efficient than the previous two algorithms as it avoids stepwise refinement and intermediate path consistency or table lookup. This algorithm directly constructs a solution for example in Figure 22(a) by instantiating each edge constraint by the least element of the constraint on the edge, to give the solution in Figure 22(c). The similar algorithm for C_8 is presented in a later Section.

This algorithm differs from the earlier algorithms (*Scenario* and variant) with regard to relative order of instantiations of TPP and NTPP. It can be seen that swapping the order of refinement of TPP and NTPP does not affect the path-consistency. This is evident from the following lemma.

Lemma 4: Let *N* be a path consistent RCC-8 constraint network with relations in $\hat{H}_8$, or in Q_8 such that the constraints are of the form {DC}, {EC}, {PO}, and {TPP, NTPP}. By instantiating either all the non-singleton constraints to {TPP} or instantiating all to {NTTP}, we get a consistent singleton labeling.

Proof: Consider a triplet in *N* such that one edge has {TPP, NTPP} and other two edges have any of DC, EC or PO. From the composition table it is clear that each of TPP and NTPP give a consistent singleton triplet. A triplet is not path-consistent when {TPP, NTPP} is present on two edges. When the triplet contains {TPP, NTPP} on all three edges, it can be checked from the composition table that the triplet is path-consistent and allows both the relations in the two edge constraints. ■

In [Gerevini and Renz 1998], the proof of correctness of the algorithm scenario guarantees that NTTP instantiation gives a valid consistent singleton labeling. But above lemma gives a more general result. The same result holds also for labels {TPPI, NTPPI}. It is interesting to note here that labels like {TPP, TPPI} are not possible after the stages of DC- EC- PO-refinements for relations in $\hat{H}_8$ or Q_8 (also for C_8).

2.7 Convex Relations in RCC-8

The study of RCC-8 network in terms of the least element helps in many other ways to provide better insight to RCC-8. In this Section, we introduce the convex class of RCC-8 and show some interesting properties using the concept of existence of the least element.

The concept of convex relation plays a very important role in devising solution techniques. The convex constraints of IA ensure that path consistency is sufficient for consistency. Convex class has another important characteristic- path consistency implies that the convex network is minimal. Convexity is important not only due to its algorithmic properties but also for its physical significance. In real-life, the constraints are mostly convex. Another interesting aspect of convexity in qualitative reasoning research is that the rich theories of geometry can be made use of to derive interesting conclusions. We are deprived of similar benefits when we resort to formalisms other than geometry. Prominent evidence is the machine-assisted proof of tractability for ORD-Horn in [Nebel and Bürckert 1995]. This result can only be analytically derived by an equivalent geometric interpretation as preconvex class [Ligozat 1996]. In this light, it can also be noted that the maximal tractable classes of RCC-8 are also generated by machine-assisted proofs. We do not have yet a valid explanation for the absence of any result in the direction of convexity in RCC-8. One can only guess that due to lack of a valid diagrammatic interpretation of RCC-8, researchers have not given adequate attention to this aspect. Nevertheless, the significance of convexity cannot be undermined. The indefiniteness in the shape, connectedness and dimension of the spatial variables may be another reason for the absence of a diagrammatic interpretation. The importance of the general notion of convexity is clearly acknowledged in [Bennett and Cohn 1999]. We make use the properties of convex sets of the lattice to overcome the deficiency. Conventionally, a convex relation is interpreted as an interval in a complete lattice for interval algebra [Ligozat 1994]. In order to get a lattice from RCC-8 semi lattice, we introduce three virtual atomic relations namely TPP', TPPI' and PO' as shown below in Figure 24. We call this extended set as **L**. The virtual atomic relations are neither used for any computation nor included as constituents in any RCC-8 relation. Though the naming of these virtual nodes as TPP', TPPI' or PO' is not relevant in the present context, it can be seen as the reflection of TPP, TPPI or PO (respectively) on the line joining NTPPI, EQ and NTTP. If one takes the reflection of all the eight elements then the relationship of RCC-8 and IA emerges.

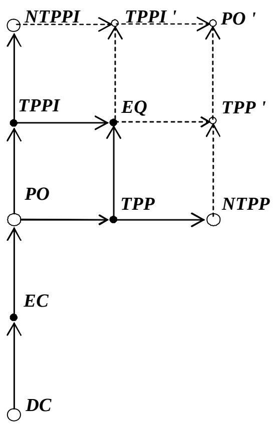


Figure 24. RCC-8 semi lattice with virtual atomic relations.

If r and s are two elements in a lattice such that $r \preccurlyeq s$, an interval $[r, s]$ is the set of all elements t, such that $r \preccurlyeq t$ and $t \preccurlyeq s$ ($[r, s]$ is known as the convex set of the lattice [Liu 1985].

Definition 17: A relation $R \in$ **P** is *convex* relation in RCC-8 if $R = [r, s] \cap$ **U**, for some interval $[r, s]$ in **L**.

For example, R = {DC, EC, PO, TPP, TPPI, EQ} is a convex relation defined by the interval [DC, EQ]. But the relation R = {TPP, TPPI, EQ} is not a convex relation because it is not an interval in the lattice. The convex relation R_1' = {PO, TPP, NTPP, TPPI, EQ, NTPPI} is obtained from the interval [PO, PO'].

***Definition 18**:* For a relation $R \in$ **P**, the *convex closure* of *R*, *I(R)*, is the smallest convex relation containing it.
Convex closure of a convex relation is the relation itself, i.e. $I(R) = R$.

***Definition 19**:* $Conv_8$ is the set of all convex relations of RCC-8.
It is interesting to note that all the convex relations belong to the set J_8. It is easy to enumerate all the convex RCC-8 relations and we observe that there are 41 convex relations in RCC-8. These relations are listed in Appendix -E. $Conv_8$ is a class common to the classes C_8 and $\hat{H}_8$. We observe that four relations of $Conv_8$ namely {TPP, EQ}, {TPP, NTPP, EQ}, {TPPI, EQ} and {TPPI, NTPPI, EQ} are not present in Q_8. We observe that $Conv_8$ is closed under composition, intersection and converse. Since the number of relations in this class in only 41, we verified exhaustively for every pair of convex relations. We extend the least element-based interpretation to the analytical proof of the set of convex relations in RCC-8.

Theorem 14: $Conv_8$ is closed under composition, intersection and inverse, and it contains 41 relations.
Proof:
For a given convex relation $R = [r, s]$, suppose $R^{-1} = [r^{-1}, s^{-1}]$ is not a convex relation. This means that $\exists\, t \in R^{-1}$ such that either $r^{-1} \nprec t$ or $t \nprec s^{-1}$. This implies that $\exists\, t^{-1} \in R$ such that either $r \nprec t^{-1}$ or $t^{-1} \nprec s$, which violates the definition of convex relation. This proves that $Conv_8$ is closed under inverse.
Let R_1 and R_2 be two non-atomic convex relations, such that $R_3 = R_1 \cap R_2$. Suppose R_3 is not a convex relation. This means that the virtual bounds of R_3, r and s exist, such that $\forall\, t \in R_3$, $r \preccurlyeq t$ and $t \preccurlyeq s$ and $r \notin R_3$, $s \notin R_3$. Since R_1 and R_2 are convex relations, $r \in R_1$, $s \in R_1$, $r \in R_2$, $s \in R_2$. By the definition of R_3, any set of atomic relations common to R_1 and R_2 are present in R_3. Hence proved that R_3 contains r and s. This proves that $Conv_8$ is closed under intersection.
Let R_1 and R_2 be two non-atomic convex relations, such that $R_3 = R_1 \circ R_2$. Suppose R_3 is not a convex relation. This means that $\exists\, t \notin R_3$ such that $r \preccurlyeq t$ and $t \preccurlyeq s$, where $\forall\, m \in R_3$, $r \preccurlyeq m$

and $m \preccurlyeq s$. This means that the there is a gap in the relation obtained by composition of an atomic relation (r_i) in a row with two atomic relations(c_i, c_{i+1}) in the consecutive columns of RCC-8 CT. This can be denoted as (r_i o c_i) ∪ (r_i o c_{i+1}) ∉ $conv_8$. This violates the continuity property of RCC-8 CT. This proves that $Conv_8$ is closed under composition.

■

On the same lines as the tractability of the proof for the existing tractable classes, we prove the tractability of convex class of relations as the following theorem.

Theorem 15: $Conv_8$ is a tractable class of relations for consistency checking.
Proof:
Tractability of $Conv_8$ is established by showing that path consistency is sufficient for consistency for $Conv_8$ network. $Conv_8$ satisfies two properties – closed class and a subset of J_8. We can construct a consistent singleton labeling for a $Conv_8$ network by least element instantiation method. Since the $Conv_8$ class is closed, path consistency is sufficient to ensure that the singleton labeling is consistent.

■

2.8 Minimal Class in RCC-8

Besides identifying tractable classes of a CSP, researchers have also attempted to determine if path consistency is sufficient to determine minimal network. Determining consistency by itself does not give any solution of a CSP; it only ensures the existence of at least one solution. The next task is naturally to determine a solution or all solutions. In [Ligozat, 1994], a direct method to construct a feasible scenario for a path consistent preconvex IA network is proposed. The main idea is to choose an atomic relation among the allowed ones that is of highest possible dimension. The solution constructed in such a way is indirectly a solution for an IA network with convex relations, since the atomic relations present in any preconvex relation are also present in a convex relation. This solution contains mostly the two-dimensional elements i.e. b, o, d, di, oi, bi or else the lower-dimension ones in the absence of the higher-dimension ones. This indirectly proposes a method to build a solution of a path consistent convex IA network without backtrack.

In the context of RCC-8, our tractability proofs are constructive in nature in the sense that we provide a scheme of instantiating every relation by its least element that preserves path consistency. Thus a solution can be generated for any consistent network. But in many applications, one needs to determine other solutions too for several reasons and it may be necessary to generate all the solutions. It may often be necessary to get a solution based on some criteria such that the desired solution is better than the solution obtained by the least element instantiation. In this context, it is important to study the minimality property of the constraint network. In [Renz and Nebel 1999], it is shown that for the known tractable classes of RCC-8, namely Q_8, C_8 and $\hat{H}_8$, path consistency does not ensure minimality. And hence neither does path consistency ensure consistency. We show here that a path consistent network with relations in $Conv_8$ class is minimal. The following lemma can be established by simple inspection of the composition table.

Lemma 5: For the atomic relations r, t, s_1 and s_2, such that $s_1 \preccurlyeq s_2$

if $r \in s_1 \text{ o } t$ and $r \in s_2 \text{ o } t$ then $r \in s \text{ o } t$, $\forall\ s \in [s_1, s_2]$.

Similarly, if $r \in t \text{ o } s_1$ and $r \in t \text{ o } s_2$ then $r \in t \text{ o } s$, $\forall\ s \in [s_1, s_2]$.

For instance suppose s_1 = EC and s_2 = NTPP so that $s_1 \preccurlyeq s_2$.

Let t = TPP, $s_1 \text{ o } t$ = {EC, PO, TPP, NTPP} $s_2 \text{ o } t$ = {NTPP}

For r = NTPP, $r \in s_1 \text{ o } t$ and $r \in s_2 \text{ o } t$.

Consider $[s_1, s_2]$ i.e. [EC, NTPP]. If we compute $s\ o\ t$ for every s in $[s_1, s_2]$ we have, s = EC, $s \text{ o } t$ = {PO, TPP, NTPP}; s = PO, $s \text{ o } t$ = {PO, TPP, NTPP}; s = TPP, $s \text{ o } t$ = {TPP, NTPP} and s = NTPP, $s \text{ o } t$ = {NTPP }. We see that NTPP $\in s \text{ o } t\ \forall\ s \in$ [EC, NTPP].

Lemma 6: For convex relation R and S and atomic relations r and s such that $r \preccurlyeq s$ such that (r, R, S) and (s, R, S) are path-consistent triplets then so is ([r, s], R, S).

Proof:

Consider the following pair of path-consistent triplet of relations $T_1(r, R, S)$ and $T_2(s, R, S)$ where r and s are atomic relations, R and S are $Conv_8$ relations.

By theorem 11, the triplets (r, σ(R), σ(S)) and (s, σ(R), σ(S)) are path-consistent. Let r_1 = σ(R) and r_2 = σ(S). Hence *(r, r_1, r_2) and (s, r_1, r_2)* are path-consistent triplets.

So, $r \in r_1 \text{ o } r_2,\ r_1 \in r \text{ o } r_2^{-1},\ r_2 \in r_1^{-1} \text{ o } r$.

Also, $s \in r_1 \text{ o } r_2,\ r_1 \in s \text{ o } r_2^{-1},\ r_2 \in r_1^{-1} \text{ o } s$.

By Lemma 5, we have $[r, s] \in r_1 \text{ o } r_2$, $r_1 \in [r, s] \text{ o } r_2^{-1}$ and $r_2 \in r_1^{-1} \text{o } [r, s]$.

Hence ([r, s], R, S) is a path consistent triplet. ■

The generalized version of lemma 6 can be stated as the following theorem.

Theorem 16: If (r_1, r_2, r_3) and (s_1, s_2, s_3) are two path consistent triads of atomic relations such that $r_i \preccurlyeq s_i$ $\forall$ i, then ([r_1, s_1], [r_2, s_2], [r_3, s_3]) is also path consistent.

Proof:

Suppose ([r_1, s_1], [r_2, s_2], [r_3, s_3]) is not a path consistent triad.

$\Rightarrow$ ([r_2, s_2] o [r_3, s_3]) $\cap$ [r_1, s_1] $\neq$ [r_1, s_1]

$\Rightarrow$ $\exists$ $t \in$ [r_1, s_1] such that $t \notin$ ([r_2, s_2] o [r_3, s_3])

i.e. (t, [r_2, s_2], [r_3, s_3]) is not a path consistent triad.

But from Lemma 6 above (t, [r_2, s_2], [r_3, s_3]) is path consistent. ■

Theorem 17: Any path consistent $Conv_8$ network is a minimal label network.

Proof:

Suppose N is a path consistent convex network. In order to show that N is minimal, it is necessary to show that every atomic relation participates in a solution. Let t be an atomic relation in R_{12}. R_{12} is convex and is $[r, s] \cap \mathbf{U}$. $\sigma(R_{12}) = r$. So $r \preccurlyeq t$ and $t \preccurlyeq s$. Without loss of generality, we assume that there is no other t' such that $r \preccurlyeq t' \preccurlyeq t$. It suffices to show that t is a feasible relation. We replace R_{12} by R'_{12} (that is, $R'_{12} = [t, s] \cap \boldsymbol{U}$) in N. We enforce path consistency. If the modified network is path consistent then the least elements of the edges are feasible relations and the least element of the modified R'_{12} is either t or a relation t' that is preceded by t. By convexity property of generalized lemma 6, since r and t' are feasible, so is t. If the modified network is not path consistent then for some triad (i, j, k), R'_{ij} o $R'_{jk} \cap R'_{ik} = \emptyset$. We know that R_{ij} o $R_{jk} \cap R_{ik} = R_{ik}$ and $R'_{ij} \subseteq R_{ij}$, $\forall$ i, j.

So for any $t \in R'_{ik}$,

$t \in R_{ij}$ o R'_{jk} or $t \in R'_{ij}$ o R_{jk} and $t \notin R'_{ij}$ o R'_{jk}.

In such a case

either t o $(R'_{jk})^{-1} \cap R'_{ij} = \emptyset$ or $(R'_{ij})^{-1}$ o $t \cap R'_{jk} = \emptyset$

This implies that R_{ij} o $R_{jk} \cap R_{ik} \neq R_{ik}$, which is a contradiction. ■

In this study, we provide a direct proof for backtrack-free solution for RCC-8 convex without any preference of dimension of the constituent atomic relations. The proof purely depends on the closure property of the convex class of relations. We prove that minimality and convexity properties are sufficient conditions to find a solution in backtrack-free manner.

Theorem 18: Solution of a minimal path consistent RCC-8 convex network can be found in a backtrack-free manner.

Proof:

Suppose *N* is a minimal path consistent RCC-8 convex network. By the definition of minimality, every atomic relation *r*, in every constraint R_{ij} of *N* participates in a solution *S*. *S* is a singleton path consistent network. In order to show that it is possible to find *S* without backtracking, it is necessary to show that any partial instantiation of *N* retains the path consistency of the network.

Let *r* be an atomic relation in a relation R_{ij} for a given *i* and *j*. We replace R_{ij} by *r* and enforce path consistency. If the modified network *N* is inconsistent, then for some triad (*i*,*j*,*k*),

$$(R_{ij}' \circ R_{jk}) \cap R_{ik} = \emptyset$$

where $(R_{ij} \circ R_{jk}) \supseteq R_{ik}$ and $R_{ij}' \subseteq R_{ij}$.

This implies that $(R_{ij}' \circ R_{jk}) \cap (R_{ij} \backslash R_{ij}' \circ R_{jk}) = \emptyset$, i.e. $(R_{ij} \circ R_{jk})$ is not a convex relation. This is a violation of the closure property of the convex relations of RCC-8. Hence the incremental process of singleton instantiation of an edge followed by enforcing path consistency for all the edges retains the path consistency of the network till the complete solution is found. ■

We have shown that for a convex and minimal RCC-8 network, it is guaranteed to find a solution without backtracking. Convexity is only a sufficient condition for minimality but not necessary. It is possible to find a network that is path consistent and minimal but not in the convex class. For such a network if all the steps of intermediate instantiations, as mentioned above retain the path consistency of the network, then also a solution is assured without backtrack. It may be noted here that [Renz 1999] proposes a $O(n^2)$ procedure to derive a specific consistent scenario in a backtrack-free manner. But this procedure generates only a specific type of consistent scenario. For instance, if each edge in the network has DC as a label, then the trivial solution of mutually disconnected set of spatial entities is generated. However, knowledge of minimality would help in obtaining all types of consistent scenarios in a backtrack-free manner.

2.9 Preconvex relations in RCC-8

In this Section, we introduce preconvex relations in RCC-8. We can notionally term DC, PO, NTTP and NTTPI as relations of one category (say, *dimension* 2 or 2-cell), the relations EC, TPP and TPPI as of lattice *dimension* 1 or 1-cell and EQ as the relation with *lattice dimension* 0 or 0-cell. This distinction of the roles of these atomic relations is with respect to the intersection of the boundaries of the spatial entities. When two spatial entities are related by relations of the first category the relationship determined by sharing interior regions, the second category corresponds to the sharing of boundaries. When they are related by the third category the boundaries coincide. The dimension of a relation *R* is the highest dimension of the atomic relations that constitute the relation *R*. Preconvex relations are the relations that are obtained from the convex relations by removing some atomic relations of lower dimensions.

Definition 20: A relation *R* of RCC-8 is said to be a *preconvex* relation if it is obtained from a convex relation by removing some lower dimension relations denoted as

$$dimension(I(R) \setminus R) < dimension(R).$$

For example, the relation R = {EC, PO, EQ} is preconvex, since the atomic relations TPP and TPPI of dimension 1 are removed from the convex relation I(R) = {EC, PO, TPP, TPPI, EQ} of dimension 2. The relation R_1 = {EC, TPP} is not preconvex because the atomic relation PO of dimension 2 is removed from the convex relation I(R_1) = {EC, PO, TPP}.

Definition 21: $PConv_8$ is the set of all preconvex relations of RCC-8.

By enumerating we find that there are 144 preconvex relations (Appendix -C). It is verified that this class is closed under three operations.

Theorem 19*:* $PConv_8$ is closed under converse, intersection and composition.

Proof:

For a given $PConv_8$ relation R, suppose R^{-1} is not a $PConv_8$ relation.

$$\Rightarrow \dim(I(R^{-1})\setminus R^{-1}) \geq \dim(R^{-1})$$

$$\Rightarrow \exists\, t \in I(R^{-1}) \text{ such that } \dim(t) \geq \dim(R^{-1})$$

$$\Rightarrow \exists\, \dim(t^{-1}) \geq \dim(R)$$

This is violates that $R \in PConv_8$. This proves that $PConv_8$ is closed under inverse.

Let R_1 and $R_2 \in PConv_8$ relations, such that $R_3 = R_1 \cap R_2$ where $R_3 \notin PConv_8$.

$$\Rightarrow \dim(I(R_3)\setminus R_3) \geq \dim(R_3)$$

$$\Rightarrow \exists\, t \in I(R_3) \text{ such that } \dim(t) \geq \dim(R_3)$$

Since $R_3 \subseteq R_1$, $I(R_3) \subseteq I(R_1)$, above-mentioned $t \in I(R_1)$. Absence of t in R_3 implies that t is absent (i.e. a gap of at least same dimension exists) in at least one of the two relations R_1 and

R_2.This contradicts the assumption that R_1 and $R_2 \in$ PConv$_8$. This proves that R_3 is a PConv$_8$ relation. This proves that PConv$_8$ is closed under intersection.

Let R_1 and $R_2 \in$ PConv$_8$, such that $R_3 = R_1$ o R_2. Suppose $R_3 \notin$ PConv$_8$.

$\Rightarrow$ dim(I(R_3)\R_3) $\geq$ dim(R_3)

$\Rightarrow \exists\, t \in$ I(R_3) such that dim(t) $\geq$ dim(R_3) but $t \notin R_1$ o R_2.

In other words, $t \in$ I(R_1 o R_2). If there exists a gap of at least the same dimension in R_1 o R_2, this can happen only if such a gap exists in at least one of R_1 or R_2. This means that dim(t) $\geq$ dim(R_1) and/or dim(t) $\geq$ dim(R_2). This violates the assumption that both R_1 and R_2 are PConv$_8$ relations. This proves that PConv$_8$ is closed under composition. ■

PConv$_8$ properly contains Conv$_8$. PConv$_8$ is a proper subset of C$_8$ but it is not contained in J$_8$ as {NTPP, EQ} is preconvex but not in J$_8$. Thus, this class contains relations for which the corresponding least elements do not exist. Hence our earlier analysis of tractability as well as the algorithm to determine consistent scenario are not valid for the preconvex class due to this reason. One tends to think that though the existence of the least element plays a crucial role in *least element instantiation* scheme, the notion of dimension also plays some nontrivial role for certain classes. The maximal instantiation in interval algebra is also an evidence of role of dimension in tractability analysis. Moreover we observe that majority of relations in PConv$_8$ are two-dimensional (102 among 144) and composition of any pair of two-dimensional atomic relations has the least element that is two-dimensional. These observations motivate us to attribute dimension with the property of existence of the least element.

In this Section, we introduce a modification of the partial order $\preccurlyeq$ as $\preccurlyeq'$, which gives higher preference to the higher dimensional atomic elements. The precedence order with respect to $\preccurlyeq$ differs from that of $\preccurlyeq'$ as EC $\preccurlyeq$ PO, TPP(I) $\preccurlyeq$ NTPP(I) and NTPP(I) $\npreceq$ EQ become PO $\preccurlyeq'$ EC, NTPP(I) $\preccurlyeq'$ TPP(I) and NTPP(I) $\preccurlyeq'$ EQ respectively. The least element of a relation R with respect to $\preccurlyeq'$ is said to be the *maximal dimension least element* of R with respect to $\preccurlyeq$. It is interesting to note that due to the proposed modifications the set of antichains described Section 3, change to {AC$_1$, AC$_2$, AC$_3$, AC$_4$, AC$_7$}.

Definition 22: A basic relation s is the *maximal dimension least element* of R, if s is the least element using the partial order $\preccurlyeq'$. We denote for a relation R, the maximal dimension least element as $\mu(R)$, if it exists.

Let R_1 be {EC, PO, TPP, NTPP, TPPI, NTPPI, EQ}. We know that $\sigma(R_1)$ is EC and $\mu(R_1)$ is PO. We have the following observation.

Lemma 7: For any RCC-8 relation R, if $\sigma(R)$ exists then $\mu(R)$ also exists.

Proof: The proof is trivial as the modified precedence relation introduces precedence for additional pairs of atomic relations and retains the precedence of $\preccurlyeq$.

There are RCC-8 relations for which $\mu(R)$ exists but $\sigma(R)$ does not exist. Let R_2 = {NTPP, EQ}, $\sigma(R_2)$ does not exist but $\mu(R_2)$ = NTPP. On the other hand, if R_3 = {TPP, TPPI}, we can see that $\mu(R_3)$ is not defined. Hence except for the two relations least element is defined for all other relations in the preconvex class. ■

Lemma 8: For each $PConv_8$ relation the *maximal dimension least element* $\mu(R)$ exists.

Proof: We know from Lemma 7 that if $\sigma(R)$ exists for any R then $\mu(R)$ also exists. For $PConv_8$ class there are only two relations namely {NTPP, EQ} and {NTPPI, EQ} for which $\sigma(R)$ does not exist. The modified precedence order $\preccurlyeq'$ introduces the new ordering, NTPP $\preccurlyeq'$ EQ and NTPPI $\preccurlyeq'$ EQ with NTPP and NTPPI as $\mu(R)$ respectively and for these relations $\mu(R)$ exist. ■

The following result follows from the proof of Lemma 8.

Corollary 1: For each C_8 relation the *maximal dimension least element* $\mu(R)$ exists.

Theorem 20: Let R_1, R_2 and R_3 be C_8 relations such that (R_1, R_2, R_3) is a path consistent triad then ($\mu(R_1)$, $\mu(R_2)$, $\mu(R_3)$) is also a path consistent triad.

Proof: By theorem 11, for a path consistent triplet (R_1, R_2, R_3), ($\sigma(R_1)$, $\sigma(R_2)$, $\sigma(R_3)$) is a consistent singleton triplet. For relations with $\mu(R) = \sigma(R)$ the result holds directly. Let us identify the cases when $\mu(R) \neq \sigma(R)$. $\sigma(R)$ differs from $\mu(R)$ in the following cases.

Case I: R has EC and PO but does not contain DC.

Suppose the triplet has a relation along one edge *R* that contains {EC, PO} as a subset and DC is not in *R*. Since the triplet is path consistent, the composition of the relations in other two edges contains {EC, PO}. We also know that by instantiating this relation to {EC} the triplet remains path consistent (theorem 11). From direct inspection of the composition table we see that for those composition table cell entries, which do not contain DC, if EC is contained in composition of any pair of atomic relations, so is PO. By instantiating the relation as PO the triplet still remains path consistent.

Suppose the triplet has relations along two edges *R* that contains {EC, PO}as a subset without DC. It is clear from the composition table that every atomic relation in EC o EC is in PO o PO. If we get (*s*, EC, EC) as the solution of any consistent triplet with two relations containing {EC, PO}, then *s* is in EC o EC and hence in PO o PO. This means that the singleton triplet (*q*, PO, PO) is also a consistent triplet.

Suppose the triplet has all the relations of type *R* such that *R* contains {EC, PO}as a subset and DC is not in *R*. When all the three relations in a triplet contain {EC, PO}, it can be checked from the composition table that both the singleton triplets (EC, EC, EC) and (PO, PO, PO) are consistent triplets. Hence swapping the order of replacement of EC and PO does not affect the path consistency of a triplet.

Case II: *R* contains TPP and NTPP but does not contain DC, EC or PO.

By theorem 11, the least element instantiation gives path consistent triplet. The least element in this case is TPP. By lemma 4, swapping the order of instantiation of TPP and NTPP does not affect the path consistency of a triplet. Hence instantiating with NTPP yields a path consistent triplet.

Case III: *R* contains TPPI and NTPPI but does not contain DC, EC or PO.

The similar argument as that of Case II holds.

Case IV: *R* contains EQ and NTPP but does not contain DC, EC, PO or TPP.

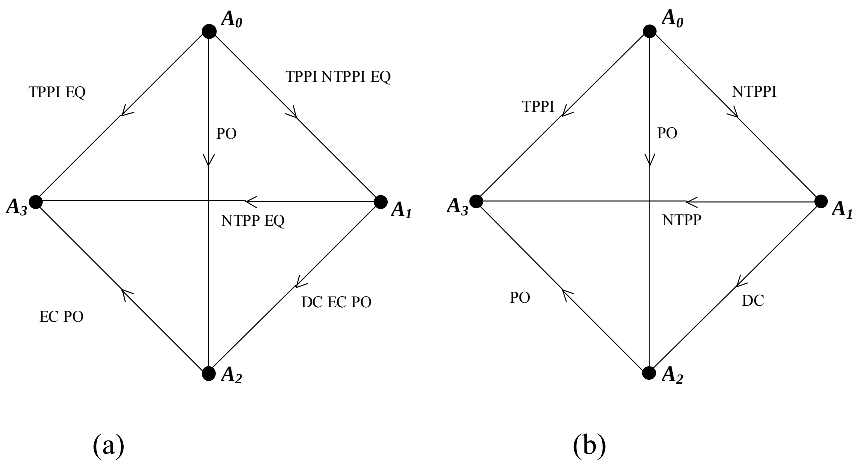
This is trivially proved due to the fact that EQ-elimination yields path consistent triplets.

Case V: R_i contains {NTPPI, EQ}

The similar argument as that of Case IV holds.

Besides the above argument, it is exhaustively checked that the theorem holds for all possible cases. ■

The above theorem gives us scope for defining a new instantiation scheme, namely *maximal dimension least element instantiation.* Given a $PConv_8$ network, the method of least element instantiation may not always yield a solution as $\sigma(R)$ is not defined for every preconvex relation. In such cases, maximal dimension least element instantiation helps in finding a solution. Consider the $PConv_8$ path consistent network in Figure 25.



(a) Example RCC-8 constraint network with constraints in $PConv_8$
(b) Solution obtained by maximal dimension least element instantiation.
Figure 25. Example preconvex network with a consistent instantiation.

The least element instantiation cannot give a solution since $\sigma(R)$ does not exist for edge relation A_1A_3. We get a solution by *maximal dimension least element instantiation* (Figure 25b). We have exhaustively checked for all possible path consistent triplets in $PConv_8$ class that the method of maximal dimension least element instantiation gives a solution. We describe below the *maximal dimension least element instantiation process* in Table 4.

Algorithm: *Construct_Solution_Maximal(N)*

Input: *N*- A n-node path-consistent network with relations R_{ij}

Output: *S*- A consistent scenario for *N* with relations S_{ij}

$S_{ij} = \mu(R_{ij})$

Table 4. Maximal dimension least element instantiation based algorithm.

Theorem 21: For any $PConv_8$ network path consistency decides consistency and $PConv_8$ is a tractable class.

Proof: By theorem 20, it is clear that maximal dimension least element instantiation preserves path consistency for C_8 relations. Path consistent singleton label is a solution for the network. The *Construct_Solution_Maximal* algorithm correctly computes a solution of any path consistent network when for every C_8 relation. The *Construct_Solution_Maximal* algorithm works in $O(n^2)$ time. Hence this algorithm gives solution for a path consistent $PConv_8$ relation in polynomial time. ■

Though path consistency is sufficient for determining consistency for preconvex relations, any path consistent $PConv_8$ network is not necessarily a minimal network. Consider a four node preconvex network relations on edges as {DC, EC}, {DC}, {DC, EC}, {TPPI, EQ}, {NTPPI, EQ}, {DC EC, PO, EQ}, respectively. This network is path consistent but not minimal because the relation EC on edge v_{03} is not a feasible relation.

2.10 Investigation of all possible partial orders for RCC-8

An idea that has been brought to the fore by Freksa (1992) is that of conceptual neighbourhood. It was originally applied to temporal intervals as described by Allen (1983).

***Definition* 23**[Freksa 1992]**:** The graph of all possible direct transitions from one relation to another is termed as a *transition graph*.

Freksa actually introduces the notion of conceptual neighbourhood in the context of a discussion of states of partial or incomplete knowledge. The example he uses is related to Allen's work on representing temporal relations.

***Definition* 24**[Freksa 1992]**:** Any connected sub graph of the transition graph (including single nodes and full graph itself) is termed as a *conceptual neighbourhood*.

Two relations drawn from a set of JEPD relations are immediate neighbours if one can be transformed into the other by a process of gradual, continuous change which does not involve passage through any third relation. Freksa presents conceptual neighbourhoods primarily as reflecting general properties of human knowledge and as a means of simplifying transitivity tables and the reasoning that depends upon them. The temporal relations between intervals, with which he is particularly concerned, cannot themselves change over time. There are actually 1255 different conceptual neighbourhoods implicit in the transition graph for interval algebra. Freksa picks out as of particular significance those neighborhoods that appear as entries in the transitivity table for the set of base relations, using these neighbourhoods to construct new and simpler transitivity tables without loss of information. Out of 169 entries in the IA CT, only 29 are different. Similarly, as per the corresponding neighbourhood diagram for the RCC-8 relations (Figure 26), only 21 entries are different in the RCC-8 CT. For instance, no entry in the RCC-8 CT contains the three base relations {TPP, NTPP, EQ}.

The entries in a composition table depend on a set of axioms that represent a particular set of restrictions on the kinds of change we want to permit. Freksa distinguishes three kinds of neighbourhood among Allen's temporal relations. Two relations are 'A-neighbours' if one can be reached directly from the other by moving one of the boundary points(start or end) of one of the two related temporal intervals. If one relation can be reached from the other by leaving the durations of the two related intervals fixed, and moving one forward or backward in time relative to the other, they are 'B-neighbours'. Finally, if one relation can be transformed directly into another by leaving the midpoints of two related intervals fixed, but varying their durations, they are 'C-neighbours'. Similar distinctions can be drawn in the case of RCC-8 relations, although here it is natural to interpret the different kinds of neighbourhood as corresponding to different sets of physically possible or permitted continuous change over time.

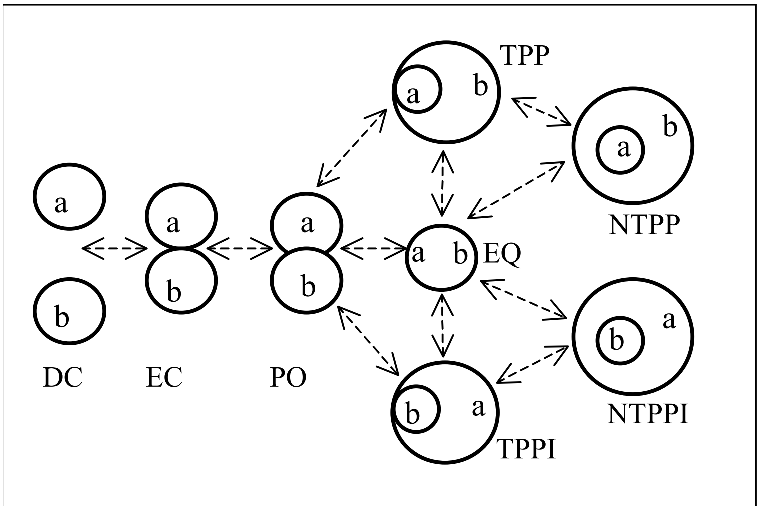


Figure 26. Conceptual neighbourhood diagram for RCC-8.

To summarize, conceptual neighborhoods give 'next' relation. The uncertainty of relation gives connected sub-graphs namely, the entries in the composition table. We study the kinds of spatial change that are allowed in the transition graph for RCC-8. In general, translation or rotation of one region relative to another and expansion or contraction of one region while the other remains unchanged, bring about topological changes that follow the links in the graph.

The links in the diagram indicate what are the candidate next relations if the spatial entities transform or translate. We can delete links given certain constraints. For instance a gradual expansion of a region x can in some circumstances lead directly from NTPP(x,y) to EQ(x,y) and then to NTPPI(x,y), bypassing TPP and TPPI. The NTPP-EQ-NTPPI path requires a kind of spatiotemporal coincidence in order to occur: the boundary of the region that begins as the smaller must reach that of the initially larger region everywhere at once, if it does not do so, the transition will be from NTPP to TPP.

1. The partial order introduced so far in this chapter, considers only two links namely TPP-EQ and TPPI-EQ. The partial order so defined makes it a join semi lattice as shown in Figure 27(a). The set of convex relations contains 41 relations and the set is closed under converse, intersection and composition. $\sigma(R)$ is defined for 235 relations. Least element instantiation computes a solution for path consistent J_8 networks.

2. Consider the partial order with two links namely NTPP-EQ and NTPPI-EQ. This partial order makes the lattice a complete lattice as shown in Figure 27(b). There are 41 convex relations and the set is closed under converse and intersection. This set is not closed under composition since it does not contain the relations {DC, EC, PO, TPP, TPPI, EQ}, {DC, EC, PO, TPP, TPPI, NTPPI, EQ} and {PO, TPP, TPPI, NTPPI, EQ}. The least element instantiation method does not hold for this partial order. In [Bittner and Stell 2000], a specific style of defining RCC relations is introduced based on the outcome of *meet* operation between two spatial regions. Each RCC-8 relation is represented as a triple where the three entries take a value from a set of three truth values {F, M, T}. The set of triples is partially ordered by setting $(a_1, a_2, a_3) \leq (b_1, b_2, b_3)$ iff $a_i \leq b_i \ \forall \ i = 1,2,3$ where the truth values are ordered by F < M < T. Unlike the ordering presented in figure 17, their lattice diagram shows the ordering NTPP ≤ EQ and NTPPI ≤ EQ but {TPP, EQ} and {TPPI, EQ} are indirectly comparable pairs of atomic relations in the absence of direct links. This partial ordering makes the set a complete lattice, which is same here.

3. Consider the partial order if we introduce the PO-EQ link only, this link is implied with the relation that PO precedes EQ. All the other four links from EQ are dropped such that EQ is not related to any of TPP, NTPP, TPPI, NTPPI. This ordering makes it a join semi lattice as shown in Figure 27(c). The convex set contains 54 relations. This set is closed under the three operations. The least element instantiation method holds for this partial order.

4. When all the five links of EQ are dropped, EQ is not related to any of the five atomic relations, namely PO, TPP, NTPP, TPPI, NTPPI, EQ. This is a join semi lattice as shown in

Figure 27(d). This lattice differs from that in (3) above in that the atomic relation is not at all related to EQ (not even indirect ordering). The set of convex relations contains 49 relations. This set is closed under intersection. The set is not closed under converse. {TPPI, NTPPI, EQ} is not an interval in the lattice but {TPP, NTPP, EQ} is an interval. This set is not closed under composition, since the relation {DC, EC, PO, TPP, TPPI, NTPPI, EQ} is not a convex relation. The least element instantiation method does not hold for this partial order.

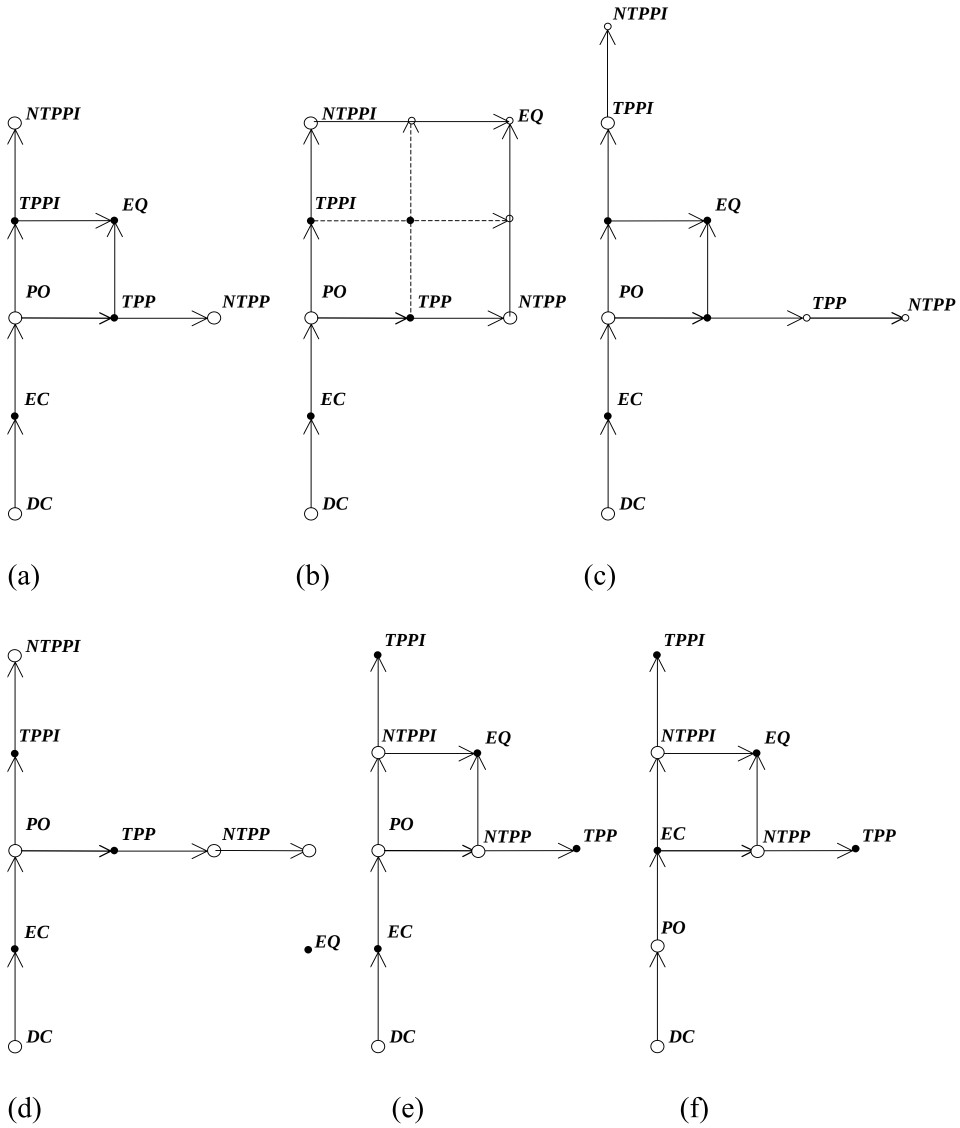


Figure 27. Lattice diagrams for the partial orders in RCC-8.

5. For three links to be considered, namely PO-EQ, NTPP-EQ, NTPPI-EQ it becomes a join-semi lattice as shown in Figure 27(e). The higher dimensional elements NTPP and NTPPI are placed before the corresponding lower dimensional ones, TPP and TPPI. The set of convex relations contains 41 relations. This set is closed under intersection and converse, but not

under composition. The least element instantiation method does not hold for this partial order.

6. The join semi lattice in Figure 27(e) is modified to that shown in Figure 27(f), where in addition to NTPP, NTPPI the other two-dimensional element PO is given preference than EC. The set of convex relations contains 41 relations. This set is closed under intersection and converse, but not under composition. The least element instantiation method does not hold for this partial order.

There are domains in which preference for continuous over discontinuous change in spatial relations is unjustified. The partial order in 5 and 6 are applicable for reasoning in such domains. For instance, consider the regions based on the amount of rainfall they receive annually. The regions *R* and *S* correspond to the regions receiving high and low rainfalls respectively. Initially, high rainfall happens at two different patches r_1 and r_2, such that NTPP(S,r_1) and DC(S,r_2). This describes a spatial relation of PO between *R* and *S*. The subsequent year, suppose high rainfall is received only at r_2 only. This means that region patch r_1 does not exist at all. This changes the spatial relation between *R* and *S* to DC, without an intermediate relation of EC. Hence, in such a domain the relation discontinuously changes from PO to DC without EC in between. These are the situations where in higher dimension based instantiations will help in reasoning tasks.

We observe that the set of convex relations is closed only for those partial orders for RCC-8, for which the relation { DC, EC, PO, TPP, TPPI, EQ } is a convex relation for the given RCC-8 composition table.

2.11 Coarsening of a tractable class

In the previous Section we provide analytical proofs for tractability of $PConv_8$. It is seen that the method of least element instantiation provides a solution for a path consistent RCC-8 network with convex relations. When the property of convexity is lost from $Conv_8$ to $PConv_8$ class, least element instantiation method no longer gives the solution and is replaced by maximal dimension least element instantiation. We explore the possibility of a method for preconvex relations such that least element instantiation method still gives a solution. An obvious method is to artificially induct atomic relation DC as the least element to every $PConv_8$ relation.

Definition 25: $DPConv_8 = \{R_{ij} \mid R_{ij} = R_{ij}' \cup \{DC\}$ for some $R_{ij}' \in PConv_8\}$

It is interesting to note that all relations in $DPConv_8$ belong to the set J_8 and the least element is always DC. For instance the preconvex relation R_1 = {NTPP, EQ} for which least element is not defined becomes $R_2 = R_1 \cup \{DC\}$ = {DC, NTPP, EQ} such that $R_2 \in DPConv_8$ with $\sigma(R)$ defined. Hence the method of including DC in every $PConv_8$ relation has converted pair of relations {NTPP, EQ} and {NTPPI, EQ} to relations {DC, NTPP, EQ} and {DC, NTPPI, EQ}, respectively which belong to the set J_8. We observe that there are 79 relations in $DPConv_8$ and that the following result holds for this set.

Theorem 22: $DPConv_8$ is closed under converse, intersection and composition.

Proof:

For a given $DPConv_8$ relation R, suppose $R^{-1}$is not a $DPConv_8$ relation. There are two possibilities for the structure of R^{-1}:

a. DC $\notin R^{-1}$ irrespective of the fact whether R^{-1} is a $PConv_8$ relation or not. This means that DC is not present in R as well. This violates that $R \in DPConv_8$.

b. DC $\in R^{-1}$ but $R^{-1} \notin PConv_8$.

$\Rightarrow \dim(I(R^{-1})\backslash R^{-1}) \geq \dim(R^{-1})$

$\Rightarrow \exists\, t \in I(R^{-1})$ such that $\dim(t) \geq \dim(R^{-1})$

$\Rightarrow \exists\, t^{-1} \geq \dim(R)$

Since $I(R^{-1}) = (I(R))^{-1} \Rightarrow t \in (I(R))^{-1}$

$\Rightarrow t^{-1} \in I(R)$.

This violates the assumption that R is a preconvex relation.

Let R_1 and R_2 be two non-atomic $DPConv_8$ relations, such that $R_3 = R_1 \cap R_2$. Suppose $R_3 \notin DPConv_8$. There are two possibilities for the structure of R_3:

a. DC $\notin R_3$ irrespective of the fact whether R_3 is a $PConv_8$ relation or not. This means that DC is not present in the common part of R_1 and R_2. This implies that DC $\notin R_1$ and DC $\notin R_2$. This violates that R_1 and R_2 are $DPConv_8$ relations.

b. DC $\in R_3$ but $(R_3\backslash DC) \notin PConv_8$. Let $R_4 = R_3\backslash DC$.

$\Rightarrow \dim(I(R_4)\backslash R_4) \geq \dim(R_4)$

$\Rightarrow \exists\, t \in I(R_4)$ such that $\dim(t) \geq \dim(R_4)$

Since $R_4 \subset (R_1\backslash DC)$, $R_4 \subset (R_2\backslash DC)$, $I(R_4) \subseteq I(R_1\backslash DC)$, above-mentioned $t \in I(R_1\backslash DC)$. The absence of t in R_3 implies that t is absent (i.e. a gap of at least the same dimension exists) in at least one of the two relations R_1 and R_2.This contradicts the assumption that R_1 and R_2 are preconvex relations. This proves that R_3 is a $PConv_8$ relation.

Let R_1 and R_2 be two non-atomic DPConv$_8$ relations, such that $R_3 = R_1$ o R_2. Suppose R_3 is not a DPConv$_8$ relation. This means that R_3 satisfies at least one of the following two conditions:

a. DC $\notin R_3$ irrespective of the fact whether R_3 is a PConv$_8$ relation or not. We observe from the composition table, that for a relation R_3 resulting due to composition not to contain DC is possible only when DC is not present in at least one of R_1 or R_2.

b. DC $\in R_3$ but (R_3\DC) $\notin$ PConv$_8$. This means that (R_3\DC) = (R_1\DC) o (R_2\DC). This implies that at least one of R_1 or $R_2 \notin$ PConv$_8$. ■

Theorem 23: For any DPConv$_8$ network path consistency decides consistency.

Proof:

Method of least element instantiation for DPConv$_8$ gives the same solution as that given by least maximal dimension element instantiation due to the fact that atomic relation DC is the least element as well as the maximal dimension least element. ■

The class DPConv$_8$ is a proper subset of C$_8$. Interestingly, the above two classes jointly define the class C$_8$. That is, C$_8$ = (PConv$_8$ $\cup$ DPConv$_8$). The following theorem is known in [Renz 1999] and analytical proof based on lattice interpretation is presented here.

Theorem 24: The class C$_8$ is closed under converse, intersection and composition.

Proof:

Possibilities for a relation $R \notin$ C$_8$ are the following:

a. $\sigma(R)$ does not exist: {DC, EC, PO}$\cap R=\emptyset$ and $\exists$ i such that $R \cap AC_i = AC_i$ and inf(AC_i)$\notin R$ where $i \in [1,4]$.

b. $\sigma(R)$ exists: {DC, EC}$\cap R \supseteq$ {EC}, PO$\notin R$ and R {TPP, NTPP,TPPI,NTPPI,EQ} $\neq \emptyset$

Let $R \in$ C$_8$, $R^{-1} \notin$ C$_8$. Due to symmetry w.r.t PO-EQ line, R^{-1} satisfies (a) or (b) then so does R. This proves that C$_8$ is closed under converse.

Let R_1 and $R_2 \in$ C$_8$, such that $R = R_1 \cap R_2$, suppose $R \notin$ C$_8$. For (a) to hold at least one of R_1, R_2 satisfy (a). For (b) to hold at least one of R_1 and R_2 satisfy (b). This proves that C$_8$ is closed under intersection.

Let R_1 and $R_2 \in C_8$ such that $R = R_1$ o R_2. Suppose $R \notin C_8$. For (a) to hold we observe that there are 16 CT entries where $R \cap$ {DC, EC, PO} = ∅:

{TPP,NTPP}o{TPP,NTPP},{TPP,NTPP}oEQ, EQo{TPP,NTPP}

{TPPI,NTPPI}o{TPPI,NTPPI},{TPPI,NTPPI}oEQ, EQo{TPPI,NTPPI}

⇒ both R_1 and $R_2 \notin C_8$.

For (b) to hold such that R to result due to composition, we observe that at least one of following 4 entries are required in addition to atleast one from above 16 entries.

{EC}o{TPPI,EQ},{TPP}o{EC,EQ}.

⇒ both R_1 and $R_2 \notin C_8$.

This proves that C_8 is closed under composition. ■

The tractability result for $PConv_8$ can be extended to C_8 class in a similar manner.

Theorem 25: For any C_8 network, path consistency decides consistency and C_8 is a tractable class.

2.12 Maximality of $\hat{H}_8$

So far, present framework of existence of the least element has helped us in providing analytical proof of tractability of the classes $\hat{H}_8$, C_8 and Q_8. It is known that these are maximal tractable classes. In this Section we show that the present framework helps in providing analytical proof for maximality. It may be noted here that the earlier proof of maximality is partly computer-assisted. The proof of maximality consists of two parts. The first part is to compute the closure of a superset and the second part is to prove the NP-completeness. It is shown that by adding an extra element to the class $\hat{H}_8$ (or C_8, Q_8) and by computing exhaustively the closure of this augmented set, the closure contains the relation {NTPP, EQ}. It is proved that {NTPP, EQ} with all basic relations is intractable [Renz and Nebel 1999]. We show here that by using the lattice representation, the first part of the proof can also be given analytically.

Since the number of relations in any of these classes is small, (around 160) it is not hard to compute exhaustively the closure when another element is added to it. Hence we realize that there is no real advantage of replacing computer-assisted closure computation by an analysis. Nevertheless, we provide the analytical proof of maximality of RCC-8 for two main reasons. First reason is to complete the analysis of all stages of tractability of RCC-8 and the second is to demonstrate that our present framework provides a theoretical basis, general enough for all spatial and temporal CSPs.

Theorem 26: $\hat{H}_8$ is a maximal tractable class.

Proof:

By analyzing the relations of $\hat{H}_8$, we know that for a relation in $\hat{H}_8$, the following two conditions hold:

Condition I. If PO is not present in the relation, then it doesn't contain relations from both sides of the PO-EQ line.

Condition II. If TPP (TPPI, respectively) is not present in the relation, then the relation doesn't contain {NTPP, EQ} ({NTPPI, EQ}).

Any relation that is not in $\hat{H}_8$ violates one or both of the above conditions.

Let *R* be a relation that violates condition II, then *R* contains one of EC, DC, PO and contains {NTPP, EQ} but doesn't contain TPP. Let *T* = {TPP, NTPP, EQ} $T \cap R$ = {NTPP, EQ} and $T \in \hat{H}_8$. It is known that the set of all basic relations together with {NTPP, EQ} is intractable. Closure of $\hat{H}_8 \cup \{R\}$ contains {NTPP, EQ}.

Let *R* be a relation that violates condition I but doesn't violate condition II and $R \in J_8$, then *R* contains one or more of EC, DC and contains relations from both sides of the PO-EQ line. Let $S = R \setminus$ {EC, DC}, T = {PO} $\cup$ *S*. It is clear that $T \in \hat{H}_8$. So $\hat{H}_8 \cup \{R\}$ is not closed under intersection. Moreover $T \cap R$ contains at least one of {(N)TPP, (N)TPPI}. It is already known that set of all basic relations together with one of {(N)TPP, (N)TPPI} leads to a network for which satisfiability is NP-Complete [Renz and Nebel 1999]. Closure of $\hat{H}_8 \cup \{R\}$ contains one such relation as $T \cap R$ is in the closure of $\hat{H}_8 \cup \{R\}$.

This establishes that whatever *R* we add to $\hat{H}_8$ leads to intractability. ■

2.13 Conclusions

This chapter provides an in-depth study of tractable property of networks outside the maximal tractable classes. We show that the lattice theoretic interpretation of RCC-8 helps us to study the structural properties of the classes.

In earlier research, the known classes are shown to be closed under the three basic operations by computing the closure explicitly. We provide here an analytical proof for the closure of these classes. It is shown earlier that path consistency decides consistency for $\hat{H}_8$, using the intricate analysis of modal logic. We prove the same result by a simple technique of least element instantiation. For the tractability proof of C_8 and Q_8, it is shown [Renz 1999] that by computation that these classes can be refined consistently to $\hat{H}_8$. Our proof of tractability for $\hat{H}_8$ naturally extends to Q_8. We prove the tractability of C_8 by using a method of maximal dimension least element instantiation. In order to determine a solution of the tractable network, earlier algorithms follow a sequence of refinements with intermediate steps of path consistency or table look-ups. We propose here the best possible algorithm of one stroke instantiation for finding a solution.

Besides improving the earlier results, the present formalism is potentially useful in arriving at many new results. We introduce two new tractable classes namely, convex and preconvex classes for RCC-8 based on the lattice intervals. We show that for the convex class, path consistency implies minimal network. The set of relations J_8 for which the least elements exist is introduced here and it is shown that any path consistent J_8 network is consistent. Our method of least element instantiation finds a solution for a path consistent network in polynomial time. It is interesting to see that unlike Interval Algebra, the preconvex class is not a maximal tractable class, but this class and its coarsening can together define a maximal class.

We believe that the present study is going to open a new direction of research in the tractability analysis of qualitative CSPs, in general. We intend to extend this study to other formalisms in future. It is interesting to note that though lattice representation for Interval Algebra is proposed earlier, no attempt has been yet made in linking the existence of the least element with tractability.

So far tractability results are based on machine-assisted proofs. The formalism proposed in this chapter has given an analytical interpretation of these results. The interpretation is based purely on exploiting structure of a partial ordering and the composition table that limits the enumeration to only 64 entries of the table. The least element method has demonstrated that in the case of RCC-8, existence of a least element for every constraint in a J_8 path consistent network is sufficient to identify a solution. This method has also helped to explain why a maximal tractable class is so.

The present study reveals another interesting aspect with regard to the equivalence of complexity of a set and that of its closure. In this chapter we have put forth that while enforcing path consistency to a J_8 network brings in relations, which are not in this set. In such a case, the proposed formalism cannot decide the satisfiability of the network. Traditional approach for a class to be tractable requires it to be closed under each of the three operations separately. It is possible that a set of relations is not closed under some of these three relations individually, but while these are used in path-consistency algorithm, they are closed (the set is such that the algorithm does not induct any relation outside the class). One good example of such set is the set of chains (totally ordered sets) of **L**. Thus an interesting direction may emerge if we restrict the tractability study to classes closed under path-consistency rather than closed under each of the three relations separately.

This study clearly demonstrates that tractability of a set of relations can be addressed even if it is not closed with respect to the three operations individually. Similarly, the least element method together with path consistency closure can help in grouping relations with same least element as tractable sets. This direction of research can be extended to identify both necessary as well as sufficient condition for minimality instead of just restricting the study of feasible relations to convex relations. We plan to investigate this direction of research in future.

The described work in the chapter is published as:

Chandra, P. Pujari, A K 2005. Minimality and Convexity Properties in Spatial CSPs. 17^{th} *International Conference on Tools with Artificial Intelligence (ICTAI)*, Hong Kong, China.

CHAPTER 3

CHARACTERIZATION OF HARD TCSP

3.1 Introduction

The contributions of this chapter can be divided into two halves. In the first half, the formalism proposed in the previous chapter is tailored to another framework in qualitative binary CSP – interval algebra. This method helps in providing a simple analytical proof for all the tractability results in interval algebra reported so far by other researchers using a number of complex methods. This helps us to analytically answer the question "What makes an IA instance easier?". Basic properties of composition table based on the frequency of atomic relations are studied to give a new description of the known tractable classes. The disjunctive relations in IA contain a set of atomic relations that occur maximum number of times due to composition. The property of high set plays a crucial role in analyzing the tractability properties of a class. We show here that the known tractable classes exhibit this property and hence path consistency becomes sufficient condition for existence of a solution. We also establish here that for these classes, we can get a solution by a method called '*highset instantiation*'. We propose an efficient algorithm for constructing a solution for path consistent tractable networks without any computational overhead of enforcing path consistency intermittently.

In the second half of the chapter, we are able to answer the question " What makes an IA instance hard to solve? " for general IA networks. We extend the existing entropy-based interpretation of the weighted IA network paradigm. We investigate the reasons for the incompleteness of the existing highest weight heuristic. In this chapter, we propose a measure of hardness of problem instances in interval algebra. Our investigations reveal that the constraints with entropy fluctuations, termed as *nasty constraints* are the sole reason for hardness of a given instance. The conflict between the highest weight relation in the constraint and the most supported relation along paths is the reason for entropy fluctuations in IA networks. We show here that the known tractable classes of interval algebra cannot exhibit the property of nastiness i.e. entropy fluctuation. We provide different techniques to relax nasty constraints so that number of conflicts reduce as the entropy of each constraint stabilizes over the search trajectory. This controls the path that the search takes resulting into an approximate solution of any instance of general IA networks. This study differs from the existing solution techniques of upper approximation of IA constraints that adopt a sort of a blind approximation of each constraint without addressing whether any particular relaxation is responsible for hardness. This study identifies the nasty constraints in an intelligent manner with the help of the conflict-based analysis. The main contributions of this study on the weighted IA networks are five fold. The novel method of conflicts proposed here lays a foundation for a new direction towards addressing hardness of any qualitative binary constraint satisfaction problem. Secondly, this method provides simple proof for ease and difficulty in computing a solution instance wise. Thirdly, we systematically investigate number of methods to relax the identified nasty constraints that has finally helped us to propose an algorithm to compute an approximate solution on completion. Fourthly, we handle enormous amount of numerical errors that arise due to operations on a weighted IA constraint. This approach is very handy and can be used by anybody in practice. Lastly, we extend this method to show that constraints in inconsistent IA networks are routine cases of nasty constraints with conflicts.

3.2 Tractability analysis of Tractable Classes in Interval Algebra

The disjunctive relation computed due to the conventional composition operation gives equal preference to the all constituent atomic relations. Frequency based interpretation of composition operation between the operand constraints can capture more information than IA-composition does. We rank the constituent atomic relations based on this quantitative measure. The researchers though exploited this aspect of ranked relations indirectly but there has not been any attempt to analyze an impact of highest ranked atomic relation(s) on the tractability of instances in interval algebra. In this Section, we provide a novel approach to analyze tractability of convex and preconvex classes of interval algebra using the concept of atomic relations with highest rank.

In IA CT, the frequency of 2-D atomic relations is 41 whereas that of 1-D relations is 25. This observation of higher frequency for higher dimension atomic relations motivates us to study characteristics of atomic relations along with frequency of occurrence.

Definition 1: The *high set* (H_T) for a pair of IA relations R and S is defined as a set of atomic relations in $T = R \circ S$ with highest value of frequency.

Consider for instance, R = { d, fi, eq }, S = { o, s, d, f, si, oi, mi }, T = **U** with set containing frequency of occurrence of each disjunct in IA lattice order, F_T = { 1, 1, 5, 3, **8**, 1, 1, 4, 3, 3, 5, 3, 3 }. Atomic relation *d* occurs in *T* with highest frequency of 8, denoted as high set, H_T = {d }. We list following properties of high set H_T:

a. $d = \dim(H_T)$, $d \in \{1, 2\}$ where dim(R) denotes dimension of relation R

b. $H_T \subseteq T$

The high set is like any other IA relation that may not always be disjunctive. For instance, R = { o, s, fi, eq }, S = { s, d, eq, f, si, oi }, T = { o, s, d, fi, eq, f, di, si, oi } with F_T = { 8, 8, 8, 4, 4, 4, 4, 4, 4 } and H_T = { o, s, d }.

3.2.1 High set based interpretation of IA convex and preconvex relations

In this subsection, we study the behavior of convex IA networks based on the properties of the high set for convex class of relations. We extend this interpretation to provide theoretical analysis for tractability of convex and preconvex classes.

Lemma 1: For atomic relations r, s_1, s_2 such that $T = (r \text{ o } s_1) \cup (r \text{ o } s_2)$ is a convex relation, where s_1, s_2 are neighboring atomic relations in IA lattice.

Proof: This is an observation from the composition table. We select a pair of neighboring entries s_1, s_2 along a row corresponding to atomic relation r such that $\{s_1, s_2\}$ is smallest possible convex relation. The disjunctive relation resulting due to this pair of neighboring cell entries in composition table is a convex relation. ■

Lemma 2: For atomic relation r and a convex IA relation $S = [\, s_1, s_2 \,]$, $T = r \text{ o } S$ is a convex IA relation.

Proof: Let us assume that $T = r \text{ o } S$ is not a convex relation.

$\Rightarrow \exists\, s \in S$ such that $(r \text{ o } s) \cap T = \emptyset$.

$\Rightarrow S$ contains a gap in the lattice interval.

$\Rightarrow S$ is not a convex IA relation that violates Lemma 1. ■

Lemma 3: For convex IA relations R and S, $\dim(H_T) = 2$ except for the following cases:

both $R, S \subseteq \{\, s, eq, s_i \,\}$ or $\{\, f, eq, f_i \,\}$.

Proof:

This is an observation from IA composition table. We select a pair of convex IA relations R and S as the lattice intervals $[r_1, r_2]$, $[s_1, s_2]$ along row and column respectively.

$\forall\ r \in R$, lemma 2 holds and $\dim(r \text{ o } S) = 2$ - 1

except for $r \in \{ s, eq, s_i \} \cup \{ f, eq, f_i \}$ and $S \subseteq \{ s, eq, s_i \} \cup \{ f, eq, f_i \}$.

Similarly

$\forall\ s \in S$, lemma 2 holds and $\dim(r \text{ o } S) = 2$ - 2

except for $s \in \{ s, eq, s_i \} \cup \{ f, eq, f_i \}$ and $R \subseteq \{ s, eq, s_i \} \cup \{ f, eq, f_i \}$.

Hence proved for the disjunctive relation resulting by combining 1 and 2 above. ■

Every cell entry in the CT contains a disjunctive relation that is a trivial case of a high set for the corresponding atomic relations r and s along the row and column. Path consistency of the triplet (r, s, T) allows us to decide the third relation T that can be a non-singleton relation. $T = R \text{ o } S$ with $H_T \subseteq T$. Enforcing path consistency on a disjunctive triplet (R, S, T) may eliminate the set H_T partially or completely. For of a path consistent triplet, the high set(H_T) is constrained to the atomic relations in T that satisfy the basic conditions of path consistency and have the highest frequency relative to other atomic relations present in T. This can be extended to a network with n nodes in the following manner.

Definition 2: For a given path consistent network N with n nodes, the high set (H_{ij}) for the constraint R_{ij} between the pair of nodes i and j is defined as the set of atomic relations with highest frequency that are present in the constraint resulting due to composition along every path of length two denoted as

$$H_{ij} = Highset\left(\bigcap_{k}^{n-2} \left(\left(R_{ik}\ o\ R_{kj} \right) \cap R_{ij} \right) \right)$$

For every constraint R in the network N, we obtain the corresponding high set H_T with the help of above definition. We denote the network obtained by replacing every constraint(R) by the corresponding high set(H_R) as the *high set network*(N_H).

Lemma 4: For atomic relations r, and neighbouring atomic relations s_1, s_2 such that $T = (r \text{ o } s_1) \cup (r \text{ o } s_2)$ with High sets H_T, H_S the following condition holds true

$\forall\ t \in H_T \quad r \in t \text{ o } s^{-1}, s \in r^{-1} \text{ o } t \qquad$ where $s \in H_S$

Proof: Consider the relation T in two neighbouring entries in the composition table corresponding to the atomic relations s_1, s_2 along the columns in a row r. The high sets for the relations S and T are H_S, H_T.. Every singleton t in H_T forms a path consistent triplet with atleast one s in H_S and r. ■

We extend this observation to a path consistent triplet where only one edge is a singleton.

The generalized version of Lemma 4 can be stated as the following theorem.

Theorem 1: For a path consistent convex triplet (r, S, T) with the high set triplet (r, H_S, H_T) following condition holds

$\forall\, s \in H_S$ (r, s, t) is path consistent where $t \in H_T$

Proof:

Let us assume that this condition does not hold true.

$\Rightarrow \exists\, s \in H_S$ such that $(r \text{ o } s) \cap H_T = \varnothing$

This violates Lemma 4. ■

We extend this result to a path consistent triplet where each constraint is a disjunctive IA relation.

Theorem 2: For a path consistent convex triplet (R, S, T) the high set triplet (H_R, H_S, H_T) satisfies the following condition

$\forall\, r \in H_R\, \exists$ at least one $s \in H_S$ and at least one $t \in H_T$ such that (r, s, t) is path consistent.

Proof: Let us assume this condition does not hold true.

$\Rightarrow \exists\, r \in H_R$ such that $\forall s \in H_S$ such that $(r \text{ o } s) \cap H_T = \varnothing$

This violates Theorem 1. ■

Theorem 3: All the paths of length two in a path consistent convex network agree with respect to the high set.

Proof:

Consider a path consistent convex network N. Consider an edge T with the (n-2) paths P_1, P_2, …, P_{n-2}. Let us assume that the high sets along the paths do not agree for T.

$\Rightarrow H_1 \cap H_2 \cap \ldots \cap H_{n-2} = \varnothing$

The set of paths may intersect pair-wise but this does not imply that all (n-2) paths have not empty intersection. This means that there exists a path P for T such that

$$H_P \cap \left(\bigcap_{i}^{n-3} H_i \right) = \phi$$

This implies that there exists at least one triplet in N for which theorem 2 is violated. ■

We extend the high set-based analysis of convex networks to the IA networks containing preconvex relations. Preconvex class of relations is largest known tractable class for interval algebra. This class contains 868 preconvex relations of which 853 are 2-D relations.

Lemma 5: For atomic relations r and neighbouring 2-D atomic relations s_1, s_3 such that $T = (r \circ s_1) \cup (r \circ s_3)$ with High sets H_T, H_S the following condition holds true

$$\forall\, t \in H_T \quad r \in t \circ s^{-1}, s \in r^{-1} \circ t \qquad \text{where } s \in H_S$$

Proof: Consider the relation T in two alternate entries in the composition table corresponding to the atomic relations s_1, s_3 along the columns in a row r. The high sets for the relations S and T are H_S, H_T. Every singleton t in H_T forms a path consistent triplet with atleast one s in H_S and r. ■

The same set of results hold true for preconvex networks as well.

Corollary 1: High sets along all the paths of length two agree in path consistent preconvex networks.

We conclude that this study helps us to give a simple proof that when all the relations in a path consistent preconvex network are 2-D, the 2-D solution obtained is same as that for the corresponding convex network. We can appreciate nicety of this approach in another way.

We compare this algorithm with the two counterpart results for interval algebra namely [Ligozat, 1994] and [van Beek, 1990]. The counterpart result by [Ligozat, 1994] proposes a maximal instantiation strategy. We conclude that this results due to the convexity of 2-D atomic relations in CT entries along the neighbouring 2-D relations without the lower dimension relation. The high set instantiation method automatically selects a 1-D element in high set as the candidate relation even in the presence of 2-D element. Moreover, not every 2-D relation belongs to the high set. Thus the maximal dimension instantiation strategy and the high set instantiation strategy do not always compute same solution. The latter result [van Beek, 1990] is applicable for only the convex relations that cannot be extended for the preconvex networks.

The high set instantiation method opens a new direction for the identification of sets of relations that may satisfy the high set property but may not be a class in the conventional sense of closure under the three operations. In future we will continue the investigation of least element and high set based tractability analysis for other qualitative frameworks as well.

3.2.2 High Set Instantiation Method

For any given path consistent convex network, we have to compute the high set for each constraint. This need not be an additional step and the existing algorithm for path consistency can be modified to capture the frequency of atomic relations during each composition operation. We describe a very efficient algorithm to construct a solution based on the high network($\boldsymbol{N_H}$) thus computed by replacing each constraint by the corresponding high set. The pseudocode for this algorithm is shown in Table 5.

```
Algorithm: construct_IA_convex_sol(N)
Input :        N – A n-node path consistent convex network with relations Rij
Output:        S – A consistent scenario for N with relations Sij
//compute the high set network NH with relations Hij in the following manner
∀ Rij do
        Hij = Highset( ∩_k^{n-2} ((Rik o Rkj) ∩ Rij) )

        mark Hij as notFixed
enddo

//select the atomic relation from Hij in the following manner
EdgeSet = set of any (n-2) edges from NH
while EdgeSet ≠ ∅ do
        Hij = first element of EdgeSet
        ∀k such that k ≠ i and k ≠ j do
                if atleast one of the constraint Hij, Hik, Hkj is not fixed
                        if Hij is notFixed
                                Sij = σ(Hij), Set Hij as Fixed
                                Select δik, δkj such that       σ(Hij) ∈ δik ⊗ δkj
                                mark Hik, Hkj as fixed
                        endif
                        if Hij is Fixed and Hik is notFixed and Hkj is notFixed
                                Select δik , δkj such that       Sij ∈ δik ⊗ δkj
                                mark Hik, Hkj as fixed
                        endif
                        if Hij is Fixed and Hik is Fixed and Hkj is notFixed
                                Select δkj such that             δkj ∈ Sik ⊗ Sij
                                mark Hkj as fixed
                        endif
                endif
        enddo
        Remove Hij from EdgeSet
enddo
```

Table 5. Pseudocode to compute solution of IA convex network.

By theorem 3, it is clear that *construct_IA_convex_sol* preserves path consistency. The path consistent label is a solution for the network and hence *construct_IA_convex_sol* algorithm correctly determines a solution for a convex IA network by the instantiation of the edges from the high set. In this section, we have proposed a method to instantiate every edge with one among the high set atomic relations.

Example 1. Consider a path consistent convex IA network *N* as shown below:

$R_{01} = \{ s, d, eq, f, s_i, o_i \}$

$R_{02} = \{ o_i , m_i \}$ $\quad$ $R_{12} = \{ o_i \}$

$R_{03} = \{ o, s, f_i, eq \}$ $\quad$ $R_{13} = \{ o, s, f_i, eq, d_i, s_i \}$ $\quad$ $R_{23} = \{ m, o \}$

Frequency of constituent atomic relations in lattice order for path wise indirect constraints of R_{01} namely $R_{01}(3)$ = { -, -, 8, ***7***, ***7***, 4, ***3***, ***3***, 4, ***3***, ***3***, -, - }

and $R_{01}(2)$ = { -, -, 1, ***1***, ***2***, 1, ***1***, ***2***, 1, ***1***, ***2***, -, -}

contain the highest $H_T = \{ d \}$. The highest frequency relation in $R_{01}(3) - o$ is eliminated due to the path consistency condition of the triplet (R_{01}, R_{03}, R_{31}).

The highest for $R_{02}(1)$ = { -, -, -, -, 2, -, -, 2, -, -, **6**, -, - }

and $R_{02}(3)$ = { -, -, 1, 1, 2, 1, 1, 2, 4, 4, **6**, 2, - }

is $H_T = \{ o_i \}$ that contains the highest frequency relation o_i along both the paths.

The highest frequency relation along $R_{03}(1)$ is $\{ o, s, d, d_i, s_i, o_i \}$ and [o, oi] for $R_{03}(2)$ that results into a highest H_T = { o, s } after elimination of { d, di, si, oi } due to intersection with R_{03}.

The high network N_H obtained by replacing the constraints by the corresponding highset is given below:

$R_{01} = \{ d \}$

$R_{02} = \{ o_i \}$ $\quad$ $R_{12} = \{ o_i \}$

$R_{03} = \{ o, s \}$ $\quad$ $R_{13} = \{ o \}$ $\quad$ $R_{23} = \{ o \}$

The solution computed by the highest instantiation method is given below:

$S_{01} = \{ d \}$

$S_{02} = \{ o_i \}$ $\quad$ $S_{12} = \{ o_i \}$

$S_{03} = \{ o \}$ $\quad$ $S_{13} = \{ o \}$ $\quad$ $S_{23} = \{ o \}$

This method of computing can help in computation of multiple number of solutions for given convex network. In other words, this helps us to prove that all atomic relations in high set network are feasible relations. This method holds true for IA preconvex networks as well.

3.3 Improved Weighted Path Consistency Algorithm

Existing work on weighted path consistency is successful in opening up a new direction for solution techniques for IA networks based on the highest weight heuristic. This work however suffers from serious drawback in terms of inefficiency and incompleteness. The complete set of ${}^{n}C_2 \times 2 \times (n\text{-}2)$ operations is enforced during an iteration irrespective of the fact whether the underlying disjunctive IA constraint requires modification or not. Each edge is asynchronously updated without waiting for that along subsequent paths. We propose following modifications in order to reduce number of operations.

Definition 2: The *composition of a pair of weighted IA constraints U_{ik}, V_{kj}* is defined as

$$W_{ij}^{m} = U_{ik} \otimes V_{kj} = \sum_{u}\sum_{v} M_{uvm} U_{ik}^{u} V_{kj}^{v} \qquad \text{where } 1 \leq m \leq 13$$

The vector resulting due to this modification in composition operation is not a weighted IA constraint as per the conventional definition (chapter 1) since it is not normalized.

***Definition 3*:** An *atomic relation present in a set of indirect IA constraints* is defined as an atomic relation obtained by taking element-wise intersection in the set of path-wise disjunctive IA constraints denoted as

$$\gamma = \bigcap^{m} \left\{ \left(R_{ik} \circ S_{kj} \right) \textit{where}\, k = 1, 2, ..., n-2 \right\}$$

Definition 4: An *indirect weighted constraint* for complete set of paths (*n-2*) of length two for a constraint W_{ij} is an aggregated vector resulting due to non-zero sum of composition vector along each path. Each component of indirect weighted constraint is given by expression:

$$W^{m}(k) = \sum_{k}^{\gamma \neq \phi} U_{ik} \otimes V_{\mathrm{kj}}$$

where ⊗ is weighted composition operator in definition 4 above.

γ represents presence of an atomic IA relation in a disjunctive relation.

The indirect constraint contains weight values for only those atomic relations that are present in every path. An atomic relation is ignored if it is not present along at least one path. The indirect aggregated constraint is converted into a weighted IA constraint by normalization. We formalize these modifications in wt_pc algorithm as a pseudocode shown in Table 6.

_Weighted_path_consistency_ **(W(N))**
Input: A weighted IA network W(N)
Output: A modified weighted network W(N)

do for T *iterations*

Weighted_path_consistency_iteration (W(N))

enddo

_Weighted_path_consistency_iteration_ (W(N))

```
do ∀Wij
        do ∀k = 1 to n, such that k ≠ I and k ≠ j
            W(k) = Wik ⊗ Wkj
        W ← normalized non-zero average over W(k)
        if W = 0 then exit with status = inconsistent
        Wavg[ij] ← W ∩ Wij
        if Wavg[ij] = 0 then exit with status = inconsistent
        { where ⊗ and ∩ are weighted composition and intersection
        operators}
enddo
∀Wij Wij ← Wavg[ij]
```

Table 6. Weighted path consistency algorithm.

This algorithm works in $O(n^3T)$ time where n is network size. There will be no occasion when weights in vectors will stop changing unless it is a network only with singleton labels (trivial case). For sake of simplicity, we assume that we prefer terminating the algorithm after a pre-specified number of iterations T. The functionality of modified wt_pc algorithm is summarized below.

$$\forall\, wt\_pc\_itn \left(\forall W_{ij} \left(W_{ij} \cap \frac{\sum_{k}^{\gamma \neq \phi} \sum_{u} \sum_{v} M_{uvm}\, U_{ik}^{u}\, V_{kj}^{v}}{\sum_{k}^{\gamma \neq \phi} \sum_{m} \sum_{u} \sum_{v} M_{uvm}\, U_{ik}^{u}\, V_{kj}^{v}} \right) \right)$$

The advantages of non-zero aggregation are many fold.

a. Synchronous updation of weights captures the commonality across all the paths during weighted composition.
b. Reduces the number of floating point operations to ${}^{n}C_2 \times (n - 2 + 1 + 1) = {}^{n}C_2 \times n$.
c. IA_pc is not a mandatory pre-processing step before wt_pc.

We illustrate computation of averaged constraint(W_{avg}) for the direct constraint W_{ij} based on indirect constraint(W) obtained by aggregation of all paths as given below.

Example 2: Consider an example 7 node network with ${}^7C_2$ constraints R_{ij}, $0 \le i<j \le 6$.

$\boldsymbol{R_{01}}$={ < d di mi s si f fi }
$\boldsymbol{R_{02}}$={ < > d di mi s si }
$\boldsymbol{R_{03}}$= *U*
$\boldsymbol{R_{04}}$={ > d o fi }
$\boldsymbol{R_{05}}$={ eq < oi mi si f fi }
$\boldsymbol{R_{06}}$= *U*

$\boldsymbol{R_{12}}$={ d m si fi }
$\boldsymbol{R_{13}}$={ > o oi s f fi }
$\boldsymbol{R_{14}}$= *U*
$\boldsymbol{R_{15}}$={ d o m s si f fi }
$\boldsymbol{R_{16}}$={ eq > di o m s }

$\boldsymbol{R_{23}}$={ > d di s si fi }
$\boldsymbol{R_{24}}$={ eq m mi s f fi }
$\boldsymbol{R_{25}}$={ eq > d di o oi s si f }
$\boldsymbol{R_{26}}$={ eq di oi mi s si }

$\boldsymbol{R_{34}}$={ > d o m s }
$\boldsymbol{R_{35}}$= *U*
$\boldsymbol{R_{36}}$={ eq > d oi m mi si f fi }

$\boldsymbol{R_{45}}$={ d di o si f fi }
$\boldsymbol{R_{46}}$={ = > d m mi f }

$\boldsymbol{R_{56}}$={ = d oi m mi s fi }

Each constraint R_{ij} is converted into a weighted IA constraint with equal weight for each disjunct in order of {eq, b, bi, d, di, o, oi, m, mi, s, si, f, fi} is adhered.

W_{01} = (-, 1/8, 0 1/8, 1/8, -, -, -, 1/8, 1/8, 1/8, 1/8, 1/8)
W_{02} = (-, 1/7, 1/7, 1/7, 1/7, -, -, -, 1/7, 1/7, 1/7, -, -)
W_{03} = (1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13)
W_{04} = (-, -, ¼ , ¼ , -, ¼ , -, -, -, -, -, -, ¼)
W_{05} = (1/7, 1/7, -, -, -, -, 1/7, -, 1/7, -, 1/7, 1/7, 1/7)
W_{06} = (1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13)
W_{12} = (-, -, -, 0.25, -, -, -, 0.25, -, -, 0.25, -, 0.25)
W_{13} = (-, -, 1/6, -, -, 1/6, 1/6, -, -, 1/6, -, 1/6, 1/6)
W_{14} = (1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13)
W_{15} = (-, -, -, 1/7, -, 1/7, -, 1/7, -, 1/7, 1/7, 1/7, 1/7)
W_{16} = (1/6, -, 1/6, -, 1/6, 1/6, -, 1/6, -, 1/6, -, -, -)
W_{23} = (-, -, 1/6, 1/6, 1/6, -, -, -, -, 1/6, 1/6, -, 1/6)
W_{24} = (1/6, -, -, -, -, -, -, 1/6, 1/6, 1/6, -, 1/6, 1/6)
W_{25} = (1/9, -, 1/9, 1/9, 1/9, 1/9, 1/9, -, -, 1/9, 1/9, 1/9, -)
W_{26} = (1/6, -, -, -, 1/6, -, 1/6, -, 1/6, 1/6, 1/6, -, -)
W_{34} = (-, -, 1/5, 1/5, -, 1/5, -, 1/5, -, 1/5, -, -, 0-)
W_{35} = (1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13, 1/13)
W_{36} = (1/9, -, 1/9, 1/9, -, -, 1/9, 1/9, 1/9, -, 1/9, 1/9, 1/9)
W_{45} = (-, -, -, 1/6, 1/6, 1/6, -, -, -, -, 1/6, 1/6, 1/6)
W_{46} = (1/6, -, 1/6, 1/6, -, -, -, 1/6, 1/6, -, -, 1/6, -)
W_{56} = (1/7, -, -, 1/7, -, -, 1/7, 1/7, 1/7, 1/7, -, -, 1/7)

Consider for instance constraint $W_{24,}$ vectors $\boldsymbol{Q_{24}}$(k), k = 0,1,3,5,6 along individual paths and the aggregated constraint along these paths are given below.

$\boldsymbol{Q_{24}}$(0) = (7.14E-2, 3.57E-1, 3.57E-1, 3.92E-1, 2.85E-1, 3.57E-1, 2.50E-1, 2.14E-1, 1.78E-1, 2.14E-1, 1.42E-1, 1.78E-1, 1.42E-1)
$\boldsymbol{Q_{24}}$(1) = (7.69E-2, 1.53E-1, 2.30E-1, 1.92E-1, 3.07E-1, 1.92E-1, 2.30E-1, 1.15E-1, 1.53E-1, 1.15E-1, 1.53E-1, 1.53E-1, 1.53E-1)
$\boldsymbol{Q_{24}}$(3) = (6.66E-2, 1.33E-1, 3.33E-1, 3.66E-1, 2.66E-1, 3.66E-1, 2.66E-1, 1.00E-1, 2.00E-1, 1.66E-1, 1.33E-1, 2.00E-1, 2.00E-1)
$\boldsymbol{Q_{24}}$(5) = (9.25E-2, 1.11E-1, 2.22E-1, 3.88E-1, 2.96E-1, 2.22E-1, 3.88E-1, 1.11E-1, 1.48E-1, 1.66E-1, 1.66E-1, 2.40E-1, 1.48E-1)
$\boldsymbol{Q_{24}}$(6) = (5.55E-2, 2.50E-1, 1.11E-1, - , 4.72E-1, 2.50E-1, 1.11E-1, 1.94E-1, 1.66E-1, 5.55E-2, 1.38E-1, - , 2.50E-1)

$\boldsymbol{Q_{24}}$ = (3.63E-1, 1.00E+0, 1.25E+0, -, 1.62E+0, 1.38E+0, 1.24E+0, 7.35E-1, 8.47E-1, 7.18E-1, 7.35E-1, -, 8.94E-1)

Atomic relations *d* and *f* obtain a value of zero along path *k*=6 since these are the only ones that are not supported by every path in W_{24}. The indirect weighted IA constraint is computed after normalizing vector Q_{24}.

W = (3.35E-2, 9.29E-2, 1.15E-1, -, 1.50E-1, 1.28E-1, 1.15E-1, 6.79E-2, 7.83E-2, 6.64E-2, 6.79E-2, - , 8.27E-2)

The averaged constraint is obtained by intersecting W with Wij followed by normalization.

$\boldsymbol{W \cap W_{24}}$ = (5.59E-3, -, -, -, -, -, -, 1.132E-2, 1.30E-2, 1.10E-2, -, -, 1.37E-2)

$\boldsymbol{W_{avg}}$ = (1.02E-1, -, -, -, -, -, -, 2.06E-1, 2.38E-1, 2.01E-1, -, -, 2.51E-1)

The weighted constraints at the end of 1st iteration are given below.

Iteration 1.

$\boldsymbol{W_{01}}$ = (- 1.58E-1 - 1.74E-1 1.54E-1 - - - 1.18E-1 1.02E-1 1.02E-1 1.00E-1 8.95E-2)
$\boldsymbol{W_{02}}$ = (- 1.76E-1 1.38E-1 2.29E-1 1.61E-1 - - - 8.46E-2 1.17E-1 9.20E-2 - -)
$\boldsymbol{W_{03}}$ = (9.60E-2 1.06E-1 9.37E-2 1.09E-1 8.78E-2 1.00E-1 5.58E-2 5.79E-2 5.71E-2 6.54E-2 6.81E-2 6.32E-2)
$\boldsymbol{W_{04}}$ = (-, -, 2.471784E-1 3.158807E-1 -, 2.671287E-1 -, -, -, -, -, -, 1.698122E-1)
$\boldsymbol{W_{05}}$ = (7.693077E-2 2.069828E-1 -, -, -, -, 1.893758E-1 -, 1.248499E-1 -, 1.086435E-1 1.532613E-1 1.399560E-1)
$\boldsymbol{W_{06}}$ = (3.26E-2 7.47E-2 1.27E-1 1.15E-1 9.32E-2 8.09E-2 1.04E-1 4.75E-2 6.84E-2 6.87E-2 7.22E-2 6.07E-2 5.30E-2)
$\boldsymbol{W_{12}}$ = (-, -, -, 3.71E-1, -, -, -, 2.16E-1, -, -, 1.95E-1, -, 2.16E-1)
$\boldsymbol{W_{13}}$ = (-, -, 2.14E-1, -, -, 2.17E-1, 1.91E-1, -, -, 1.32E-1, -, 1.12E-1, 1.30E-1)
$\boldsymbol{W_{14}}$ = (3.60E-2, 9.33E-2, 8.35E-2, 1.29E-1, 8.76E-2, 1.21E-1, 8.10E-2, 6.93E-2, 5.15E-2, 6.79E-2, 4.70E-2, 6.36E-2, 6.75E-2)
$\boldsymbol{W_{15}}$ = (-, -, -, 2.01E-1, -, 1.986E-1, -, 1.17E-1, -, 1.16E-1, 1.24E-1, 1.24E-1, 1.16E-1)
$\boldsymbol{W_{16}}$ = (8.92E-2, -, 2.40E-1, -, 2.06E-1, 1.90E-1, -, 1.27E-1, -, 1.45E-1 -, -, -)
$\boldsymbol{W_{23}}$ = (-, -, 1.99E-1, 1.69E-1, 2.59E-1, -, -, -, -, 1.02E-1, 1.32E-1, -, 1.36E-1)
$\boldsymbol{W_{24}}$ = (1.02E-1, -, -, -, -, -, -, 2.06E-1, 2.38E-1, 2.01E-1, -, -, 2.51E-1)
$\boldsymbol{W_{25}}$ = (5.69E-2, -, 1.00E-1, 1.47E-1, 1.59E-1, 1.42E-1, 1.21E-1, -, -, 8.97E-2, 9.45E-2, 8.77E-2 -)
$\boldsymbol{W_{26}}$ = (7.67E-2, -, -, -, 2.42E-1, -, 2.28E-1, -, 1.52E-1, 1.49E-1, 1.49E-1, -, -)
$\boldsymbol{W_{34}}$ = (-, -, 1.93E-1, 2.54E-1, -, 2.46E-1, -, 1.49E-1, -, 1.56E-1, -, -, -)
$\boldsymbol{W_{35}}$ = (3.48E-2, 1.00E-1, 8.54E-2, 1.19E-1, 8.17E-2, 1.17E-1, 8.80E-2, 7.17E-2, 5.40E-2, 7.31E-2, 5.31E-2, 6.22E-2, 5.90E-2)
$\boldsymbol{W_{36}}$ = (4.75E-2, -, 1.63E-1, 1.74E-1, -, -, 1.56E-1, 7.42E-2, 1.07E-1, -, 1.09E-1, 8.91E-2, 7.80E-2)
$\boldsymbol{W_{45}}$ = (-, -, -, 2.10E-1, 2.10E-1, 1.93E-1, -, -, -, -, 1.29E-1, 1.50E-1, 1.04E-1)
$\boldsymbol{W_{46}}$ = (9.06E-2, -, 2.44E-1, 2.22E-1, -, -, -, 1.35E-1, 1.64E-1, -, -, 1.40E-1 -)
$\boldsymbol{W_{56}}$ = (7.17E-2, -, -, 1.98E-1, -, -, 2.36E-1, 9.55E-2, 1.29E-1, 1.30E-1, -, -, 1.37E-1)

Range of values in indirect constraint is very small as compared to that in direct constraint. The average operation of weight values over a large number of paths prevents a larger difference between highest and lowest (non-zero) supported relation. A relatively larger range of values in direct constraint makes it highly skewed in nature. The wt_pc algorithm plays around only with the weight values without affecting label size of the constraint that remains same till the last iteration. The enormous amount of floating point arithmetic is highly prone to numerical errors. The multiplication of very small floating point numbers result into underflow error condition. In this study we provide a robust method to handle numerical errors as given in Appendix – F.

3.4 Characterization of Nasty constraints for IA networks

In this section, we propose a new measure to identify hardness of general IA network instances. During this analysis, we introduce a new concept of nasty constraints to show that a property such as entropy can play an important role in solving the TCSP instances. We prove that *nastiness* property is responsible for a difficulty in computing solution of a problem instance. We provide a theoretical basis to characterize the nasty constraints based on the dynamics of entropy.

We summarize some observations on existing highest weight heuristic (chapter 1) since the work on this heuristic is a starting point for this study. Highest weight heuristic [Pujari and Adilakshmi, 2004] does not work when general IA networks are converted into convex networks by taking convex cover of each constraint. Reasons for incompleteness of this heuristic are many fold. Firstly, approach of simple addition during composition for all paths prevents wt_pc to subsume IA_pc without affecting ranks of individual atomic relations. Secondly, it does not address an important necessity of numerical errors to be handled. Number of non-zero terms cannot be assumed to remain constant in each constraint over iterations. A decrease in number of disjuncts tightens a constraint further and makes instance more difficult to solve with worst-case possibility of not able to maintain consistency. Lastly, existing study ignores a possibility that paths for a constraint may disagree with highest weight relation in the constraint.

Conventionally weighted vectors can be compared in several ways. Component values can be sorted and a pair of vectors can be compared lexicographically. Disadvantage of this comparison is that one ignores importance of position of components. We compare direct and indirect weighted constraints using inner product value since lexicographic comparison is not valid in this study.

Definition 5: Inner product of a pair of non-zero weighted vectors *A* and *B* is defined as

$\lambda = \sum_{i}^{13} a_i b_i$ where a_i and b_i are individual components.

Inner product for a pair of weighted vectors captures an extent of similarity. A value close to 1 indicates more similarity whereas that closer to 0 means lesser. One gets different values of λ with same set of values if we ignore positions. This framework is expressive to capture the higher ranked values in the given pair of vectors.

Lemma 6: For a pair of Weighted Vectors *A* and *B*, $\lambda \in [a_{min}, a_{max}]$ and $\lambda \in [b_{min}, b_{max}]$ where (a_{min}, b_{min}), (a_{max}, b_{max}) are non-zero minimum, maximum values in *A*, *B* respectively.

Proof: By definition 7 above, $\lambda = \sum_{i}^{13} a_i b_i$ (1)

Consider the vector *A* $\forall\, i \quad a_i \leq a_{max}$ (2)

Combining (1) and (2) we get

$$\lambda \leq \sum_{i}^{13} a_{\max} b_i$$

$$\Rightarrow \lambda \leq a_{\max} \sum_{i}^{13} b_i \qquad \text{since} \sum_{i}^{13} b_i = 1$$

$$\Rightarrow \lambda \leq a_{\max}$$

Similarly $\lambda \geq a_{\min}$ ■

Lemma 7: Given Weighted Vectors A, B and C such that $c_i = (a_i b_i)/\lambda$ where $\lambda = \Sigma\, a_i b_i$

$\forall$ i $\in B_{high}$, $c_i \geq a_i$ and $\forall ip \in B_{low}$, $c_i < a_i$

Proof:

Consider three normalized vectors A, B and C such that $c_i = (a_i b_i)/\lambda$ where $\lambda = \Sigma\, a_i b_i$

Suppose $c_i \geq a_i$, $\Rightarrow (a_i b_i)/\lambda \geq a_i$

$\Rightarrow a_i\, b_i \geq \lambda\, a_i$

$\Rightarrow b_i \geq \lambda$ ■

Definition 6: Weighted high set(Whigh) for a pair of weighted IA constraints A and B is a set of atomic relations that contains the higher ranked relations of A and B.

This notion of weighted high set is a weighted counterpart of high set introduced in section 3.2. However highest weight value in W_{avg} does not depend on how many path vectors contain this relation as highest ranked relation. Rank along individual paths does not matter but sum total along all paths renders a relation to be most supported one. The λ value divides vector W into two halves – Whigh and Wlow. Relations with weights greater than normalization factor increase if W_{avg} replaces W_{ij}. In other words, if highest weight relation is among higher weight relations in indirect constraint, then its weight increases in the next iteration. Contrary to this, when the highest weight relation is not highest weight relation in indirect constraint, weight of this component decreases.

U = (3.35E-2, 9.29E-2, 1.15E-1, -, 1.50E-1, 1.28E-1, 1.15E-1, 6.79E-2, 7.83E-2, 6.64E-2, 6.79E-2, - , 8.27E-2)
V = (1.02E-1, -, -, -, -, -, -, 2.06E-1, 2.38E-1, 2.01E-1, -, -, 2.51E-1)

u_{max} = 1.50E-1 u_{min} = 3.35E-2
v_{max}= 2.51E-1 v_{min} = 1.02E-1
λ = 5.48E-02
Clearly, for vector ***U***: 3.35E-2 ≤ ***λ*** ≤ 1.50E-1 and for vector ***V***: 1.02E-1≤ ***λ*** ≤ 2.51E-1

Consider the vectors W_{24}, W and W_{avg} in example 2.

W_{24} = (1/6, 0, 0, 0, 0, 0, 0, 1/6, 1/6, 1/6, 0, 1/6, 1/6)
W = (3.35E-2, 9.29E-2, 1.15E-1, -, 1.50E-1, 1.28E-1, 1.15E-1, 6.79E-2, 7.83E-2, 6.64E-2, 6.79E-2, - , 8.27E-2)
W_{avg} = (1.02E-1, 0, 0, 0, 0, 0, 0, 2.06E-1, 2.38E-1, 2.01E-1, 0, 0, 2.51E-1)

λ = 5.48E-2
Rij = { = m mi s f fi }
W_{high} = { 2 3 5 6 7 8 9 10 11 13 0 0 0 }
W_{low} = { 1 4 12 }

The value of λ divides vector W into two halves namely W_{high} with 10 atomic relations and W_{low} with 3 atomic relations respectively. The weights for *b*, *bi*, *di*, *o*, *oi*, *m*, *mi*, *s*, *si* and *fi* increase in value from W_{24} to W_{avg} whereas weights for *eq*, *d* and *f* decrease. We formalize the mismatch of maxima between direct and indirect constraint in the following manner.

***Definition 7*:** For a weighted IA constraint W_{ij}, a *conflict* is said to occur if highest weight relation in W_{ij} is not among higher ranked relations in indirect path constraint W, denoted as

$$\text{argmax}(W_{ij}) \notin W_{high}$$

We discuss the impact of a conflict due to the replacement of the constraint by the averaged constraint. As wt_pc algorithm iterates, weights in networks are adjusted based on influences of the weights in indirect constraint. Absence of conflict for a constraint over a pair of iterations implies that there is a net increase in support for highest weight relation. This net increase in value may result due to an increase in support along some paths as well as a decrease in support along remaining path(s). When this decrease in support along paths dominates in value the increase, net weight decreases and shows up as a conflict. From the high set based analysis in the previous section, this conflict arises when the paths- do not agree with respect to the high set for a constraint. We have observed that for smaller instances of general IA networks (up to 10 nodes), more than one candidate highest weight relations may exist. The instances with multiple highest weight relations contain a very low number of conflicts when we match highest weight set with W_{high} set. We formalize the impact of this conflict on replacement of W_{ij} by W_{avg} as following results based on Lemma 9 and 9.

Corollary 2: $W_{avg}[\text{argmax}(W_{ij})] \geq W_{ij}[\text{argmax}(W_{ij})]$ iff $\text{argmax}(W_{ij}) \in W_{high}$.

Corollary 3: If $\text{argmax}(W_{ij}) \notin W_{high}$, then $W_{avg}[\text{argmax}(W_{ij})] < W_{ij}[\text{argmax}(W_{ij})]$.

Range of values in every weighted IA constraint increases over iterations of wt_pc algorithm in the absence of conflicts. In an ideal situation, a conflict should not take place at all. There are two possibilities here, the highest weight relation may exist in the averaged constraint with a lower weight or may be absent i.e. a weight of zero. We study relative distribution of values in W and W_{ij} based on mismatch of maxima.

Theorem 4: For weighted vectors A, B and C with following properties

a. $c_i = (a_i b_i)/\lambda$ where $\lambda = \Sigma\, a_i b_i$

b. $\forall\, a_i \geq \lambda$, $|a_i - \lambda| > |b_i - \lambda|$

c. maxima in A are high, $\forall\, a_i \geq \lambda$ and $b_i \geq \lambda$, $a_i > b_i$

if $\text{argmax}(A) \in B_{high}$ then $E_c \geq E_a$.

Proof:
By definition of entropy(chapter 1), entropies of vectors *A* and *C* are given as

$E_a = \Sigma a_i^2 \qquad E_c = \Sigma c_i^2$

$\Rightarrow \quad E_c - E_a = \Sigma c_i^2 - \Sigma a_i^2$

$\Rightarrow \quad E_c - E_a = \Sigma (a_ib_i)^2 / \lambda^2 - \Sigma a_i^2$

$\Rightarrow \quad E_c - E_a = \Sigma a_i^2 / \lambda (b_i^2 - \lambda^2)$

$E_c - E_a = \Delta E_{high} + \Delta E_{low}$ - (1)

$= \Sigma a_i^2 / \lambda (b_i^2 - \lambda^2) + \Sigma a_j^2 / \lambda (b_j^2 - \lambda^2)$

For $i \in B_{high}$, $b_i^2 - \lambda^2 > 0$, $\Delta E_{high} > 0$ and for $i \in B_{low}$, $b_j^2 - \lambda^2 < 0$, $\Delta E_{low} < 0$.

For $E_c > E_a$, the r.h.s in the expression(1) above should be greater than 0.

$\Rightarrow |\Delta E_{high}| > |\Delta E_{low}|$

By property (b) of *B* above, the factors ($b_i^2 - \lambda^2$) in the l.h.s and r.h.s are negligible as compared to the factor a_i^2 in each term.

$\Rightarrow |a_i^2 / \lambda (b_i^2 - \lambda^2)| > |\Sigma a_j^2 / \lambda (b_j^2 - \lambda^2)|$

$\Rightarrow c_1 \Sigma a_i^2 > c_2 \Sigma a_j^2$, where we assume $b_i^2 - \lambda^2$ as constant terms - (2)

if argmax(*A*) $\in B_{low}$ then $c_1 \Sigma a_i^2 < c_2 \Sigma a_j^2$ which contradicts (2) above. ■

In other words, there may exist other entry(s) in Ahigh which do not exist in Bhigh that decrease in value from *A* to *C* but not sufficient enough for entropy to decrease.

Theorem 5: For weighted vectors *A*, *B* and *C* with following properties

a. $c_i = (a_ib_i)/\lambda$ where $\lambda = \Sigma a_ib_i$

b. $\forall a_i \geq \lambda$, $|a_i-\lambda| > |b_i-\lambda|$

c. maxima in *A* are high, $\forall a_i \geq \lambda$ and $b_i \geq \lambda$, $a_i > b_i$

if argmax(*A*) $\notin B_{high}$ is necessary but not sufficient condition for $E_c < E_a$.

Proof:

Necessary : $E_c < E_a \Rightarrow$ argmax(*A*) $\notin$ Bhigh

For $E_c < E_a$, the r.h.s in expression(1) above should be less than 0.

$\Rightarrow c_{max} < a_{max}$ from lemma 6

Sufficient: argmax(*A*) $\notin$ Bhigh does not always implies $E_c < E_a$

Suppose argmax(*A*) $\notin B_{high} \Rightarrow c_{max} < a_{max}$

But for $E_c > E_a$ in expression (1) above, $|\Delta E_{high}| > |\Delta E_{low}|$

$\Rightarrow \exists\, m \in B_{high}$ such that (2) holds true. ■

In simple words, when weights for two components are very close by small fractional value, entropy increases from *A* to *C* despite a fall in maximum. This means that there exists another peak in *A* with a value more than threshold of λ. These results on entropy of weighted constraints help us to conclude that presence of conflict in a constraint does not always implies a fluctuation in entropy. We formalize these observations to introduce a concept of nasty constraints for IA networks.

***Definition 8*:** For a network *N*, a weighted IA constraint W_{ij} is said to be a *nasty constraint* if it contains a conflict and entropy decreases during atleast one iteration of wt_pc algorithm.

The analysis is based on the relative distribution of maxima between direct and indirect constraints. The whole difficulty is due to maximum shifting from one atomic relation to another. The concept of conflict definitely implies presence of competing atomic relations. A question arises as to at what stage highest weight value starts to decrease. When it dominates all other components and is closer to 1.0 or when *k* relations compete in value. In former case, a decrease in highest weight value implies a decrease in entropy. However, in latter case, it may happen that when value of highest weight relation decreases, value of atleast one more relation is very nearby. This prevents entropy to decrease despite a decrease in highest weight value as formalized above. We summarize these observations with the help of Theorems 4 and 5 in the following manner.

Corollary 5: Conflict is a necessary but not sufficient condition for entropy of a weighted IA constraint to decrease over iterations of wt_pc if the constraint contains multiple peaks.

Corollary 6: Conflict is a necessary and sufficient condition for entropy of a weighted IA constraint to decrease over iterations of wt_pc algorithm if the constraint does not contain any another competing atomic relation.

Corollary 7: Entropy of a weighted IA constraint may not always decrease in the presence of conflict.

Corollary 8: Decrease in highest weight value in a weighted IA constraint is not a sufficient condition for entropy of constraint to decrease during an iteration of wt_pc algorithm.

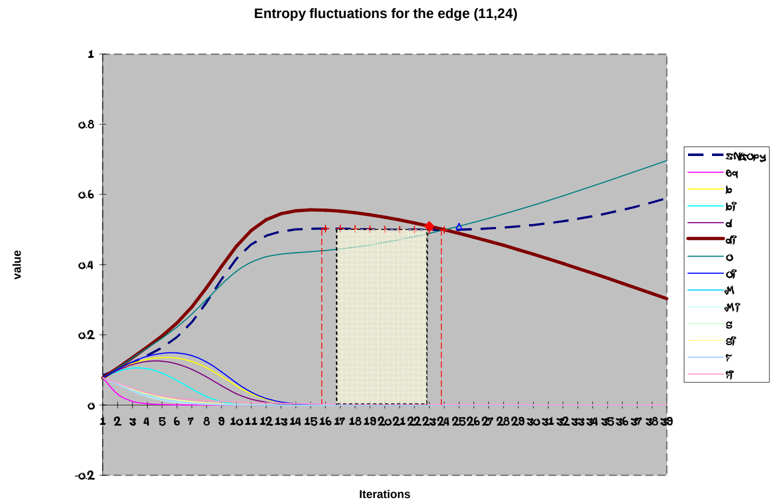
Corollary 9: The highest weight value may not always decrease due to a conflict.

Corollary 10: Entropy of a constraint can decrease only due to a conflict.

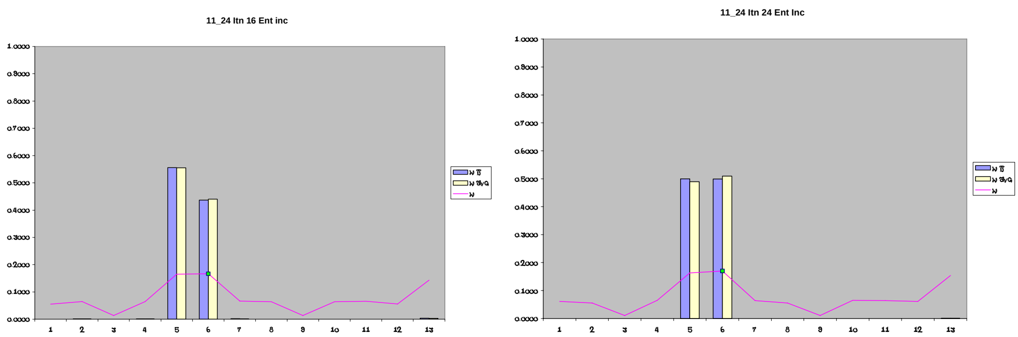
We illustrate this new concept of nasty constraints with the help of three examples.

Example 3: A relation that contains a conflict and is also a nasty constraint.

Consider for instance constraint (11,24) in a 40 node problem instance. The conflict starts at 16th iteration and continues to occur till 24th iteration.



(a). Entropy and weights for edge 11,24 from 1st to 39th iteration. Entropy continuously decreases from 17th to 23rd iteration (cream coloured rectangular area). Conflict starts from 16th till the 24th iteration (pair of dotted red lines). di is highest weight relation from 1st till 23rd iteration. Weight for di starts decreasing from 16th iteration till the last iteration shown and o becomes the highest weight relation at 24th iteration.



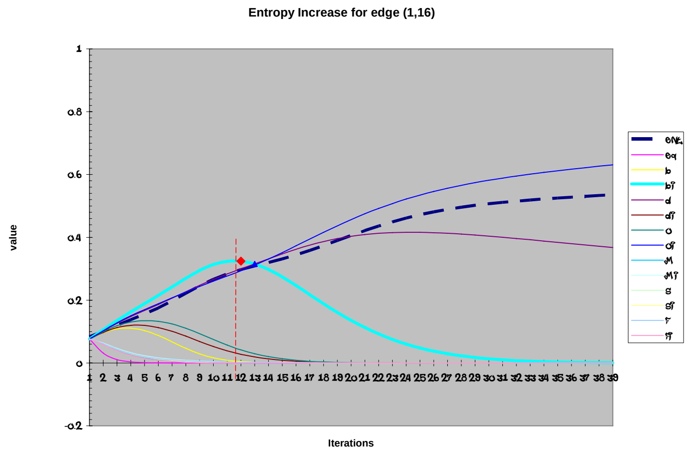
(b)-(c). Direct (W_{ij}), indirect (W) and average constraint(W_{avg}) for edge 11,24 16th and 24th iterations. di is highest weight relation till 16th iteration but looses weight thereafter due to less support than o in W (a green spot in W). o becomes highest weight relation at 24th iteration.

Figure 29. A constraint that contains a conflict and is a nasty constraint.

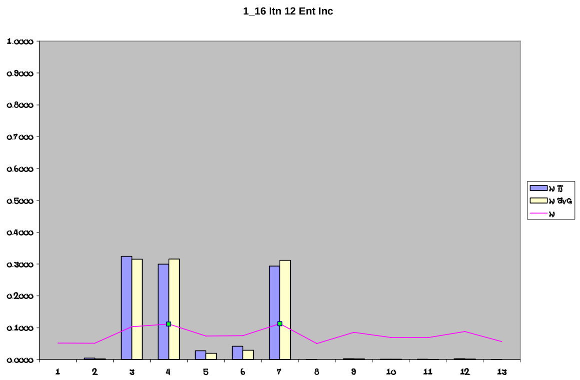
Entropy increases from 1st iteration till 16th iteration. Conflict at 16th iteration decreases the entropy from 17th till the 23rd iteration. Subsequently entropy increases from 24th iteration despite the conflict continuing till 24th iteration. The entropy does not decrease during the first and last iteration of the conflict. We observe that entropy fluctuates in iterations only when there is a conflict. However, entropy does not always decrease on conflict. The entropy of a constraint with conflict decreases only when same atomic relation continues as highest weight relation but with decreasing value.

Example 4: A constraint that contains a conflict but is not a nasty constraint.

Consider for instance constraint (1,16) in the same problem instance of 40 nodes. This constraint contains the sole conflict at 12th iteration conflict as shown in Figure 30. The entropy increases from 1st iteration till last iteration.



(a). Entropy and weights for (1,16) for 1st-39th iteration. Entropy continuously increases from 17th - 23rd iteration (thick dotted blue line). Conflict at 12th iteration (dotted red line). bi is highest weight relation from 1st - 12th iteration. Weight value for bi starts decreasing from 12th iteration till last iteration. d becomes highest weight relation at 13th iteration.



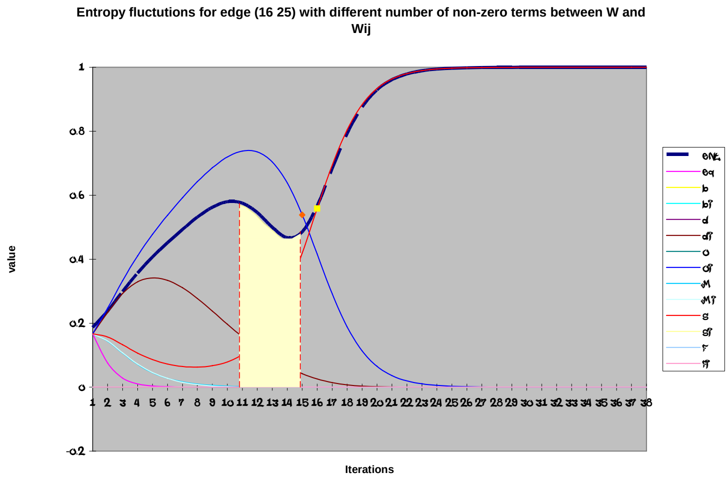
(b). Direct (W_{ij}), indirect (W) and average constraint(W_{avg}) for edge (1, 16) for 12th iteration. bi is highest weight relation till 12th iteration but looses weight thereafter due to less support than d in the indirect constraint (a green spot in W). d and oi are higher supported relation in W. d becomes the highest weight relation at 13th iteration (yellow bar).

Figure 30. A constraint that contains a conflict but is not a nasty constraint.

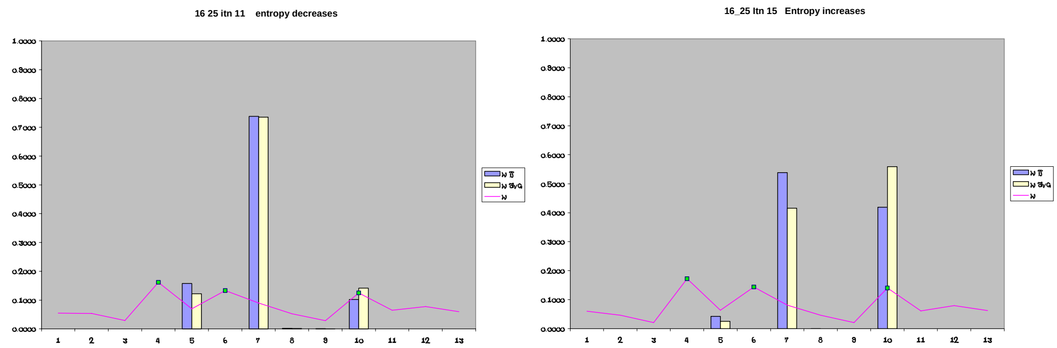
The conflict at 12th iteration does not effect the increase in the entropy at any of the subsequent iterations. This happens because the maximum shift from bi to d immediately in 13th iteration, thereby preventing entropy to decrease. Entropy of a constraint does not decrease when another atomic relation picks up as the highest weight relation within a couple of iterations.

Example 5: A Nasty constraint with highest supported relation not present in direct constraint.

Consider a constraint (16, 25) in the same instance. The conflict starts at 11th iteration and continues to occur till the 15th iteration as shown in Figure 31. The entropy increases from 1st iteration till 10th iteration. Conflict at 11th iteration decreases entropy from 11th till 15th iteration shown as cream colored area. Entropy decreases during each iteration of conflict.



(a). Entropy and weights for edge (16, 25)) for 1st-38th iteration. Entropy continuously increases from 1st -11th iteration (thick dotted blue line). Conflict takes place only at 11th iteration (dotted red line). oi is highest weight relation for 1st -18th iteration. s becomes the highest weight relation at 18th iteration.



(b)-(c). Plot for direct (W_{ij}), indirect (W) and average constraint(W_{avg}) for edge (16,25) 12th - 16th iterations.

Figure 31. A constraint that contains a conflict and is a nasty constraint.

Relations *d, di, o, oi* in *W* are in W_{high} from 1st iteration. Relations *d, o, oi* (in the order of value) continue to be higher supported relations in *W* from 5th iteration. The highest weight relation is oi. W_{ij} does not contain – *d, o*. The support of *d* and *o* along *W* increases and that for *oi* decreases. This results into a conflict at 11th iteration as shown in Figure 31(a). This conflict results into a decrease in entropy till 14th iteration. Weight for s slowely increases from 8th iteration to become highest weight relation at 16th iteration resulting into an increase in entropy. Such constraints which contain different number of non-zero entries between *W* and W_{ij} with highest supported relation not present on the edge are more prone to conflicts and larger fluctuations in entropy.

Absence of nasty constraints in Convex IA networks

The highest weight heuristic [Pujari and Adilakshmi, 2004] solves the convex IA networks with the help of the weighted paradigm. The solution is computed in the early iterations of the weighted path consistency algorithm. It is possible that there may exist a number of competing 2-D relations in the high set. In this subsection, we provide theoretical justification for tractability of weighted IA convex networks due to the absence of nasty constraints.

Theorem 6: Path consistent convex IA networks do not contain nasty constraints.

Proof:

Consider a path consistent convex IA network *N*. Let us assume that it consists of a conflict for atleast one constraint W_{ij} in the nth iteration of wt_pc algorithm. By the definition of a conflict, argmax(W_{ij}) $\notin W_{high}$. This means that there is a net decrease in the support for argmax(W_{ij}) along (n-2) paths. Thus there exists atleast one path *P*, along which the weight value of argmax(W_{ij}) decreases.

By lemma 7 for path *P*: argmax(W_{ij}) $\in W_{low}$

$\Rightarrow$ argmax(W_{ij}) $\notin H_{ij}$

This contradicts result of theorem 3 that high sets for all the paths agree in a convex network. This contradicts our basic assumption that W_{ij} is a nasty constraint. ■

We conclude that due to the absence of conflicts, entropy does not fluctuate for any constraint in the weighted IA convex network. There is a possibility that some high sets may contain a pair of competing relations but this does not hamper the computation of solution. A difficulty in this computation does not arise since the high sets agree for all the paths in convex networks. Thus the algorithm terminates with a solution much before the network approaches stability. This is an ideal situation where it is always desirable to get a solution before stabilization. Thus a need for stability of the network is not mandatory condition for a solution.

The above theoretical analysis justifies the reason for the highest weight heuristic [Pujari and Adilakshmi, 2004] tp solve the complete set of convex IA networks.

3.5 Complete Instantiation Strategies to solve general IA networks

In this section, we investigate four variants of highest weight heuristic to solve general IA instances. These strategies are based on different policies to handle conflicts. We demonstrate performance of proposed strategies experimentally.

3.5.1 Conflict handling complete instantiation strategies

We divide the set of solution strategies based on solution type that is computed in worst case namely exact and approximate. A given problem instance may be solved by any number of strategies. We propose following three exact solution strategies

a. The default strategy (τ_{HW}) is to pick up highest weight relation for every constraint.
b. For constraint with conflict, we select highest weight relation among highest supported relation that is also present in direct constraint ($\tau_{HW\_C\_WhighP}$).
c. For every constraint, irrespective of conflict we select highest supported relation that is also present in direct constraint ($\tau_{HW\_X\_WhighP}$).

We cannot ensure an exact solution, when we select a singleton based on only the support due to paths in indirect constraint as given below.

a. For every constraint, select highest supported relation (τ_{WhighP}).
b. For every constraint, select highest weight relation in the absence of conflict otherwise highest supported relation ($\tau_{HW\_C\_WhighP}$).

The pseudocode for these algorithms is shown in Table 7.

Construct_IA_sol (W(N))
Input: W(N) – A n-node path consistent weighted IA network with relation Wij.
Output: S - A n-node singleton network that is a solution with relations Sij.

do for 1000 iterations
 weighted_path_consistency_iteration(W(N))
enddo

weighted_path_consistency_iteration(W(N))
 $\forall W_{ij}$ $\forall k = 1$ to n, such that $k \neq i$ and $k \neq j$
  $W(k) = W_{ik} \otimes W_{kj}$
 W ← normalized non-zero average over W(k)
 $\lambda = \sum w[p]w_{ij}[p]$, p = 1 to 13
 $W_{avg}[i,j] \leftarrow W \cap W_{ij}$
 partition W such that $W = W_{high} \cup W_{low}$, $W_{high} \cap W_{low} = \emptyset$,
  $\forall p \in W_{high}$, $W[p] \geq \lambda$, $\forall q \in W_{low}$, $W[q] < \lambda$

 Replace $\forall(i,j)$ $W_{ij} \leftarrow W_{avg}(ij)$
 $\tau_{hw}(i,j)$ = argmax(Wij)

τ_{whigh}(i,j) = argmax(Whigh)
if (argmax(Wij) $\notin$ Whigh
{
$\tau_{hw\_c\_whigh}$(i,j) = argmax(Whigh)
$\tau_{hw\_c\_whigh\_p}$(i,j) = arg(Whigh) such that arg(Whigh) $\in$ Rij
}
else
{
$\tau_{hw\_c\_whigh}$(I,j) = argmax(Wij)
$\tau_{hw\_c\_whigh\_p}$(I,j) = argmax(Wij)
}
$\tau_{whigh\_x\_p}$(I,j) = arg(Whigh) such that arg(Whigh) $\in$ Rij
enddo

Table 7. construct_IA solution algorithm.

The algorithm works in $O(n^3\ T)$ time since we prefer terminating it after a pre-specified number of iterations(T).

3.5.2 Experimental Setup

For our experimental study through out this chapter, we generate random instances in a similar method as described in [Nebel, 1997]. A model **M**(n, d, t) [Adilakshmi and Pujari, 2004] generates networks with n number of nodes, d percentage of edges of network representing constraint density of graph, and t, constraint tightness, as average number of relations per edge. For instance, **M**(*30*, *40*, *6*) is an instance of 30 nodes, 40% of edges have 6 atomic relations and remaining 60% edges have universal relation ***U***.

In [Nebel, 1997], it is empirically shown that constraint density, d' of IA constraint graph is a critical order parameter that can lead to generate arbitrarily hard instances. A phase transition is identified at d' = 9.5 for the label size t = 6.5. We can determine critical value of d as $(100 \times d')/(n-1)$ for a given value of t from corresponding value d' given in [Nebel, 1997]. For instance, when t' = 6.5, critical value of d' is 9.5 and hence, critical value of d is 950/(n-1). It is observed that with larger values of t, phase transition region shifts to higher values of d and runtime requirements grow exponentially. Hard instances for networks of different sizes for t = 7 are given by following values of d.

n	10	20	30	40	50	60
d	100%	50%	32.8%	24.5%	19.4%	16%

Value of *d* is rounded to nearest interval step of 10. Experiments are conducted on 1.7GHz 512MB RAM Pentium IV with Windows 2000 operating system. The algorithms in this study are implemented in C-language using Visual C++ 6.0 environment.

We generate instances that completely cover both easy as well as hard problem regions. We have experimented with networks of size *n* in range [10,60] in steps of 10. For each value of *n*, we vary *d* and *t* in respective ranges [10%,50%], [5,12]. For each combination of *d* and *t* values within a given network size, two instances are generated. Complete set of instances contains 480 problems with 80 problems for each node size.

3.5.3 Experimental Analysis

During this study, numerical errors are successfully eliminated to a very large extent. The advantage of handling numerical errors is that it makes vast computations based on floating point operations robust(Appendix -F). We study experimentally behaviour of complete instantiation strategies for exact solution of general IA networks. Conventional partial instantiation strategy of backtrack and variants is known to take exponential time to solve general IA instances contrary to known easy instances in tractable classes [Nebel and Burckert, 1995]. We cannot compare performance with backtrack due to known exponential behaviour of the latter. Objective is to demonstrate improvements of existing highest weight strategy based on insight gained due to conflict introduced in previous section.

3.5.3.1 Exact Solution Complete Instantiation Strategies

Comparison based on percentage of problems solved

We compute percentage of problems solved (*p*) for every value of *n*. HW_C_Whigh_P and HW_X_Whigh_P solve nearly same number of problems as those solved by HW. This percentage falls from 100% for *n*=10 nodes to 10% for *n*=60 nodes (Figure 32). We conclude that performance of exact solution strategies drastically degrades with increase in network size. Due to this reason we do not study effect of other factors such as *d*, *t* and solution iteration number for complete set of instances

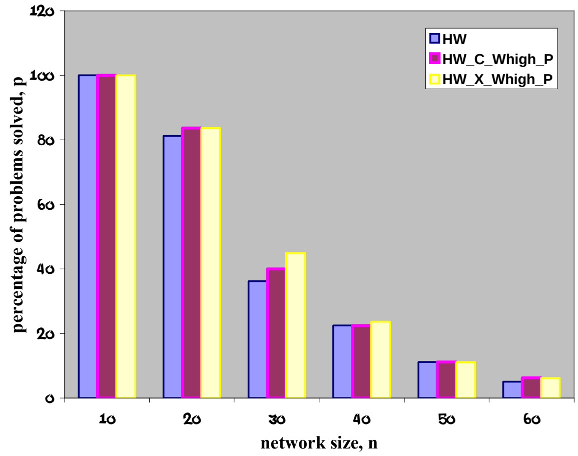
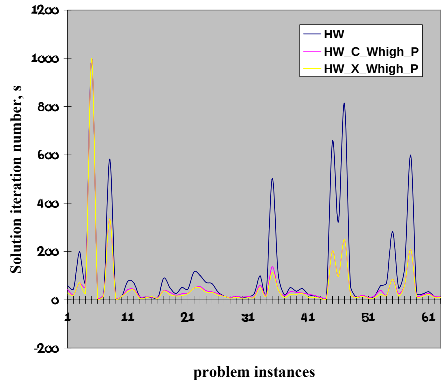


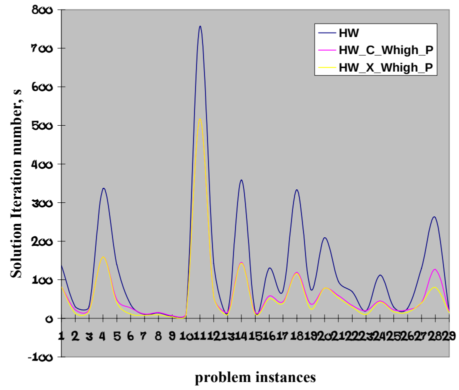
Figure 32. % of problems solved by exact solution complete instantiation strategies.

Comparison based on number of iterations required to solve

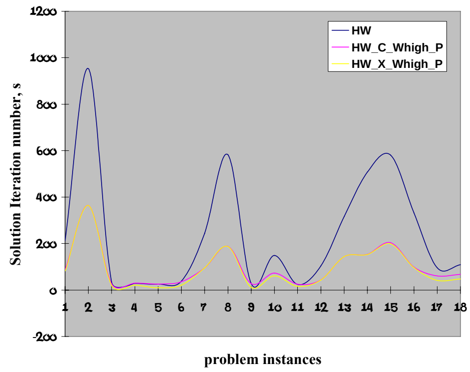
The observations for subset of problems solved by all three strategies based on solution iteration number (s) (Figure 33) are given below.

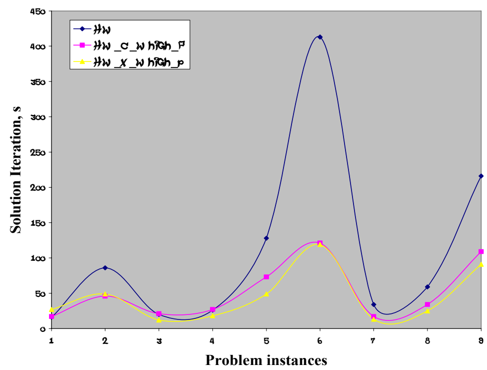
20 Nodes **30 nodes**





40 nodes **50 nodes**





60 nodes

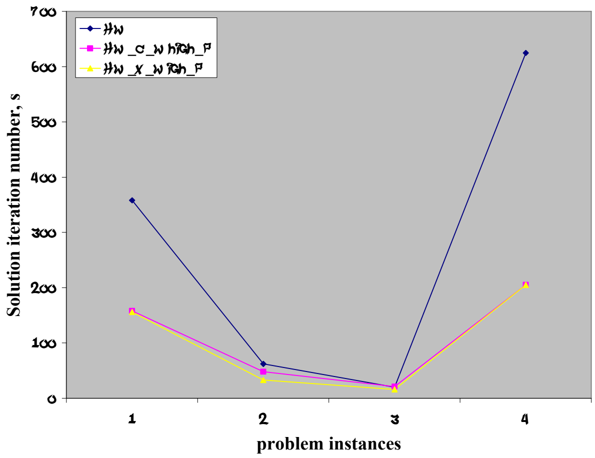


Figure 33. Solution iterations for problems solved by exact solution strategies.

a. Number of instances(Probs) solved by three strategies forms a strict hierarchy i.e $Probs_{HW} \subset Probs_{HW\_C\_Whigh\_P} \subset Probs_{HW\_X\_Whigh\_P}$. Number of such problems drastically reduce with *n*. This means that if none of the relations with higher support is feasible, then highest weight relation cannot be a feasible relation. Hence for highest weight relation to be feasible, it has to be among higher supported relations.

b. HW strategy computes a late solution as compared to other two.

c.There is not much difference in solution iteration number for the pair of conflict handling strategies. This difference increases drastically with *n*. This means that atomic relation in constraint that is higher supported in indirect constraint is more suitable for feasibility irrespective of conflict.

d. There are two possibilities for maxima in graphs for three strategies to agree.

i. Maxima match with not a very late HW solution as compared to other two. The highest weight relation has highest support. But this does not imply that this atomic relation has highest weight value in indirect constraint. This happens when there are very less number of conflicts.

ii. Maxima match with a very late HW solution as compared to other two. The highest weight relation becomes feasible very late due to conflicts. Before this happens, another relation in constraint becomes higher supported one. This happens when there are more number of conflicts. This clues that when highest weight relation is not highest supported one as compared to others on edge, pick up highest supported one on edge, even if it is not highest weight relation.

3.5.3.2 Approximate Solution Complete Instantiation Strategies

Comparison based on percentage of problems solved

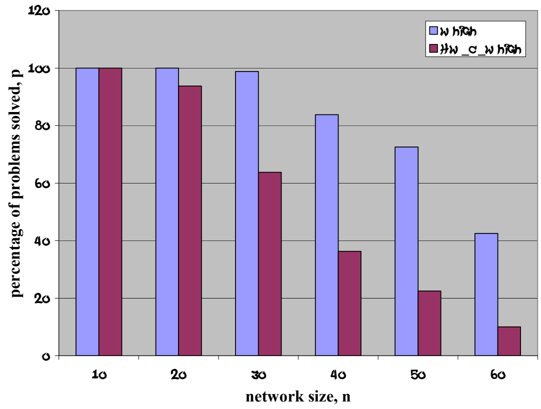
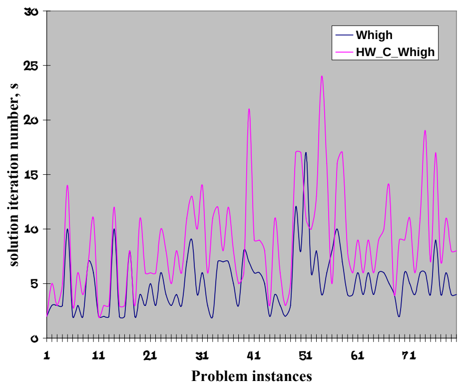


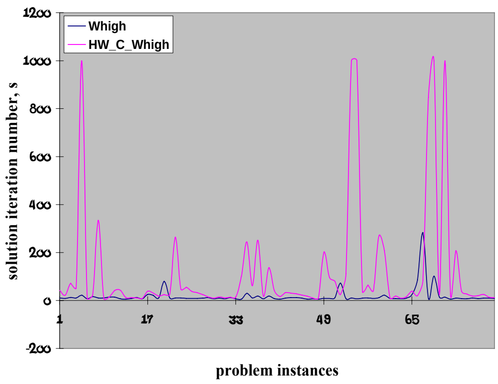
Figure 34. % of problems solved by approximate solution strategies.

We observe that there is an improvement in percentage of problems solved by this pair of approximate solution strategies as against those described so far. The Whigh strategy solves nearly 50% problems of size 60.The graph in Figure 34 shows clearly that if we do not bother about solution quality, highest supported relation in indirect constraint is the suitable candidate for a feasible relation.

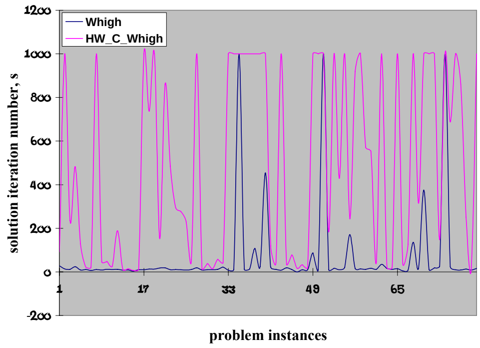
Comparison based on solution iteration and constraint density



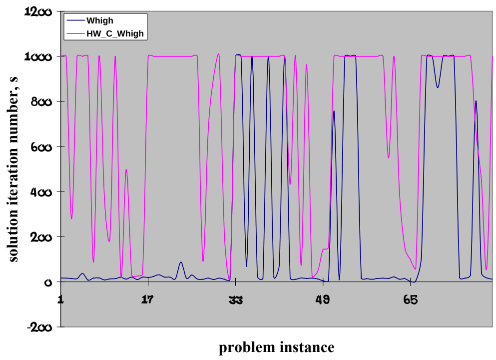
n = 10



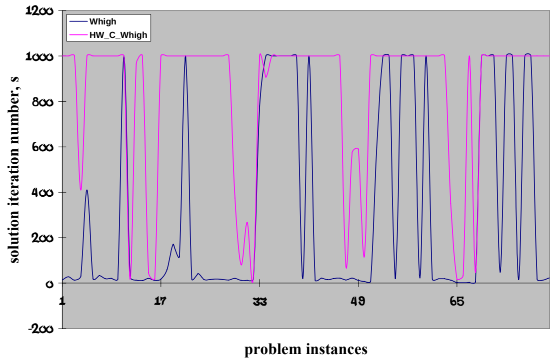
n = 20



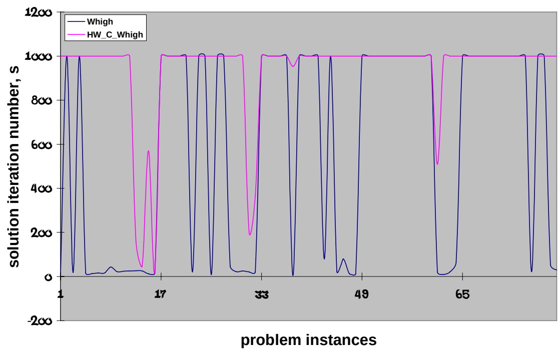
n = 30



n = 40



n = 50



n = 60

Instances are ordered for values of d [10,50] and in increasing value of t in range [5,12] for each d value.

Figure 35. Graphs for solution number, s vs network size n[10,60].

We observe that Whigh strategy outperforms HW_C_Whigh strategy in terms of number of problem solved as well as by computing an early solution. However, the former will always compromise on solution quality. The Whigh strategy graph shows maxima for t = 5 for networks upto 30 nodes for d [40,50]. Number and height of such maxima increases with n[10,30]. Same pattern continues for larger size networks, but number of these maxima increase in number fast as n increases implying that more problems are left unsolved for higher d values. This difficulty increases drastically from nearby maxima in 40 node graph to broader maxima in 60 node graph. We conclude that this strategy fails to solve networks with larger number of non-trivial constraints.

General observations for conflict handling strategies

We observe that exact solution strategies are able to solve networks only upto 20 node size. On the other hand, best possible performance for approximate solution is 50% for 60 node size. In general, despite strategies to handle conflicts, entropy fluctuations alongwith conflicts continue to take place in large numbers. We list following reasons for not able to solve problems:

a. Absence of conflicts for every constraint for many iterations continuously also fails. We support this observation with the fact that a monotonic increase in entropy towards a non-feasible relation does not help.

b. argmax(W) singleton alone does not give a solution because direct and indirect constraints disagree.

c. Conflicts as well as entropy fluctuations continue to occur due to characteristic of general IA instances. Number of nasty constraints does not increase once network is near stabilization.

3.5.4 Restart Strategies to Avoid Conflicts

When highest supported relation in Whigh is not present in a constraint, we introduce a small non-zero weight value for this atomic relation followed by renormalization and wt_pc iterations continue. Due to this forced modification, this weight value increases in subsequent iterations. But the increase is very slow and remains among lower weighted ones. This relation never picks up as highest weight relation.

Restart Policy 1: Due to a very slow increase in the weight value, it is better to introduce this atomic relation right in the beginning of weighted path consistency instead of inserting it as and when this situation arises. This will give some dwell time for the relation to become the highest weight relation. In other words, we insert this atomic relation in the starting IA network N, convert it to a weighted network W(N) and restart weighted path consistency. This algorithm solved many networks of smaller size namely, 20 nodes but the higher nodes networks could not be solved.

Restart Policy 2: For higher node networks, the restart strategy 1 forced us to restart the problem for large number of times. In order to reduce the frequency of restarts, we collect such conflicting constraints till a fixed number of iterations (say k = 2*n, 4*n etc). These atomic relations are then inserted in the initial IA network followed by a restart of the iterations. This outer loop of collecting the conflicting constraints continues for many times. We experience that this way of collecting the constraints may not suffice since there may still be conflicting relations after these k iterations.

Restart Policy 3: To capture these remaining conflicting relations that may arise in later iterations, we tried with restarting the particular constraint instead of the complete problem. This modification still does not reduce the number of conflicts.

Restart policy 4: A slight variation of restart policy 3 policy is to force the maximum supported atomic relation along the path to the highest weight of 1.0 (relax the constraint by U) and renormalizing the constraint. With this variation, small percentage of problems could not be solved (nearly 30%).

We experience large number of conflicts are present in the general IA instances. Despite the variants to handle conflicts and the restart policies, we are not able to solve majority of the instances. There is a need to modify the characteristic of the given instance such that the edge constraint and the indirect constraints are forced to agree. This will help us to control the search trajectory such that the highest supported relation which is the best possible candidate feasible relation also attains the highest weight on the edge.

3.6 Approximate Algorithm to Solve general IA networks

In this section, we propose a method to determine an approximation of the given problem. The formulation for computing this approximation of the given instance is based on our foregoing analyses that highest weight relations is a suitable candidate for feasible relation only if it is highly supported.

3.6.1 Conflict Resolution Strategy

When most supported relation in indirect constraint is not highest weight relation in direct constraint, we are forcing it to become highest weight relation for next iteration. Otherwise, constraints for which such a conflict does not take place, highest weight relation is chosen as the candidate. This modification falls in line with our intuition that constraints with conflict are forced to agree with indirect constraint by raising required atomic relation. There are many possible ways to resolve conflicts in a given problem instance. We choose to propose a strategy based on an unary operator defined below.

***Definition* 9:** A unary operation, *Setmax(m)* on a weighted vector *A* results into another weighted vector *B* denoted as $b_i = a_i / \gamma$ where $\gamma = (2-a_i)$

This operation is helpful in forcing a particular component to attain highest value. When a new relation is inserted, the constraint is approximated. In such a case, number of terms increase from *n* to *n*+1, when component value is raised from initial value of zero. When any of existing components is raised, number of non-zero terms remains the same. This operation affects entropy of vector. If existing maximum is raised, entropy increases otherwise it decreases. The latter cases arise when an existing maximum component is devoid to be raised but one among other components is raised. We formalize this observation below.

Corollary 10: Increase in maximum value in a vector *A* due to *setmax(i)* operation does not always increases entropy of weighted constraint.

Pseudocode of algorithm to compute a solution to the modified problem based on proposed conflict resolution strategy here is shown in Table 8.

---

compute_approx_solution(W(N))
Output: A singleton network τ that is a solution
while no solution *weighted_path_consistency_iteration*(W(N)) *enddo*
weighted_path_consistency_iteration(W(N))
$\forall W_{ij}$ $\forall k = 1$ to *n*, such that $k \neq i$ and $k \neq j$
    $W(k) = W_{ik} \otimes W_{kj}$
    $W \leftarrow$ normalized non-zero average over W(k)
        $\lambda = \sum w[p]w_{ij}[p]$, p = 1 to 13
    $W_{avg}[i,j] \leftarrow W \cap W_{ij}$
    partition W such that $W=W_{high} \cup W_{low}$, $W_{high} \cap W_{low} = \emptyset$,
        $\forall p \in W_{high}$, $W[p] \geq \lambda$, $\forall q \in W_{low}$, $W[q] < \lambda$
    if ($\text{argmax}(W_{ij}) \notin W_{high}$)
        $W_{avg}[\text{argmax}(W)] = 1.0$
        renormalize W_{avg}
    endif
    if Entropy(Wij) > Entropy (Wavg) then mark Rij as nasty constraint
    replace $\forall(i,j)$ $W_{ij} \leftarrow W_{avg}(ij)$
    $\forall(i,j)$ $\tau_{ij} \leftarrow$ atomic relation at argmax(W)
    if τ is path consistent then solution found
    $\forall(i,j)$ if $\tau_{ij} \notin R_{ij}$ then constraint is violated *endif*
        where R_{ij} is the disjunctive constraint in the IA network N
    endif

Table 8. Pseudocode for approximate solution algorithm.

Clearly *compute_approx_solution* is of $O(n^3T)$ complexity. As per our foregoing analyses this algorithm captures those constraints as nasty constraints for which entropy fluctuates. It is observed that there are some more constraints that are not nasty constraints, but still the highest weight relation along the paths is forced to become highest one. We term all constraints (including nasty constraints) where any time this type of adjustment of weights takes place as *approximated constraints*(AC). The algorithm starts with state of highest entropy for all constraints, that corresponds to starting point when all atomic relations in a constraint are assigned equal weights. As the algorithm iterates, *compute_approx_solution* ensures entropy of every constraint to increase non-monotonically. In later iterations, weight of an atomic relation dominates others, leading to state of maximum entropy beyond which a bounded variable like entropy (with a minimum value of -1) cannot increase.

Example 6: Consider a 60 node instance that is not solved by conflict handling strategies, contains 300 constraints with total number of 3407 conflicts till 56th iteration. This instance is solved by conflict resolution strategy in 56 iterations, 352 constraints contain 372 conflicts till 56th iteration, 13 constraints contain repeated conflicts out of which only three repeat more than twice with respective repetitions as 3,4 and 6. Entropy fluctuations are only due to conflict resolution operation. There is not even a single instance of entropy fluctuation resulting due to conflict.

The conflict resolution strategy shares some edges with conflicts. For instance, edge (0,4) contains conflict in both the strategies- with and without conflict resolution. There is a remarkable difference in behaviour of entropy of this edge as shown in Figure 36.

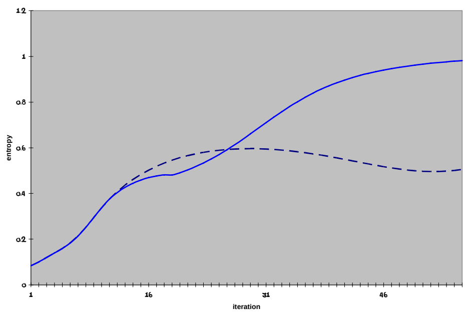


Figure 36. Entropy graph for a common constraint (0,4) for 56 iterations.

Entropy continuously decreases from 32nd iteration due to default behaviour (blue dotted line). The conflict resolution strategy prevents entropy from decreasing by resolving conflict in 19th iteration (smooth blue line). Some new constraints show up conflicts and hence affect entropy. For instance consider edge (44,54), entropy with and without the conflict resolution is shown in Figure 37.

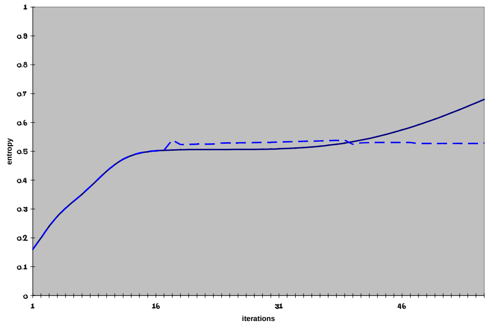
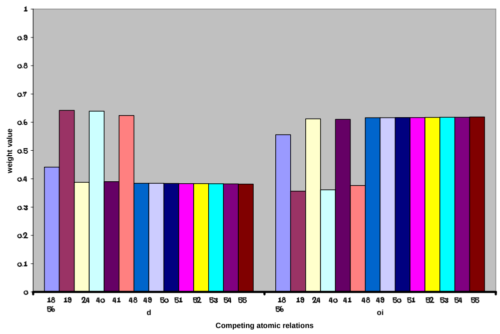


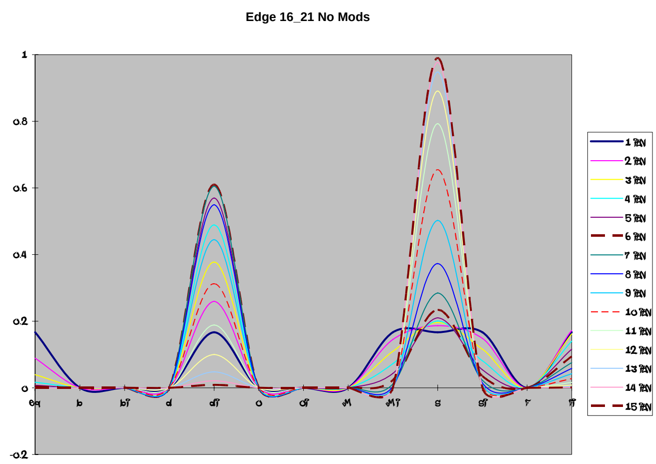
Figure 37. Entropy graph for a new-conflict constraint (44, 54) for 56 iterations. Entropy in this case depicts a monotonic increase by default from 1st till the end (smooth blue line). Conflicts repeat in this case for six iterations namely – 18, 19, 24, 40, 41, 48. We observe abrupt peak at 18th and an abrupt small dip at 40th iteration in entropy graph (dotted blue line). Relation *d* is highest weight relation till 17th iteration. Due to conflict, oi weight is raised up to bring it among high support relations. There are repeated conflicts due to competing weights of these two relations. Finally oi is chosen as feasible relation due to continuously increasing support in indirect constraint from 49th iteration (Figure 38).



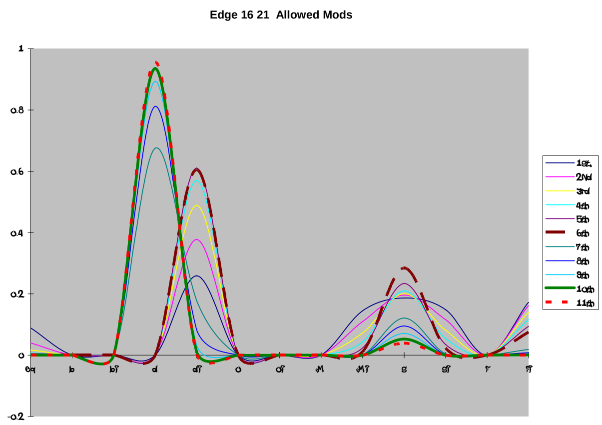
d and oi compete in a constraint with repeated conflicts at 18th, 19th, 24th, 40th, 41st and 48th iterations.

Figure 38. Constraint with competing relations during compute_approx_sol.

This is tune with our intuition that conflict resolution prevents constraints to stabilize towards a non-feasible relation and select among a number of candidates. The weight adjustments due to conflict resolution operation result in abrupt change in entropy. This is analogous to keep moving along a search trajectory till there is no conflict. We introduce some disturbance in path of search by directing it to a different point so that search trajectory leaves older path and continues from this new point during iterations to come. In this way, we are artificially altering search trajectory by removing conflict, instead of allowing it to end at a non-feasible relation. An added advantage of this policy is that during this process, we are inserting new relation(s) in constraint if required. Consider for instance, weights for constraint (16,21) without and with conflict resolution strategy as shown in Figure 12(a) and (b).



(a) Weights in constraint (16, 21) due to default behaviour of wt_pc algorithm.



(b) Weights in constraint (16,21) due to conflict resolution strategy.

Figure 39. Example constraint with a new relation inserted by *compute_approx_sol.*

This constraint contains relations {eq, di, mi, s, si, fi } in parent disjunctive network *N*. Due to default functionality, d_i is highest weight relation from 1st iteration till 10th iteration (smooth blue-dotted brown) lines in Figure 39a. A conflict takes place in 10th iteration and di continues to loose weight. Relation s steadily gains weight from 1st iteration itself but becomes highest weight relation only at 11th iteration. Entropy of constraint decreases during this shift of maximum from di to s. The constraint stabilizes at15th iteration. The behaviour of this constraint when relaxed due to conflict resolution is shown in Figure 39b. Due to conflict at 6th iteration, highest supported d is inserted and maximum shifts from di to d instead of s. The entropy of constraint does not fluctuate and it stabilizes early at 11th iteration itself. This helps us to compute a cover of constraint so that it agrees with indirect constraint. It makes it easy for search trajectory to end at a feasible relation. We interprete relations di and s as local maxima. The algorithm has prevented search trajectory from getting stuck at any of these local maxima by directing it to a global maximum relation – d.

The number of constraints with conflicts increase in number but number of conflicts drastically reduce during subsequent iterations of wt_pc. As a result number of inconsistent triplets in singleton network at every iteration decrease in number. This implies that algorithm helps us to approach the solution slowly in a stage-wise fashion. Multiple candidates in constraint collected in this fashion are a set of local maxima. When most of the constraints are near stabilization towards their respective global maxima, one of these probable local maxima is selected as the global one. We experience that in later iterations, even for higher node size networks, most supported atomic relation does not really change and Whigh network is the solution network. Nearly 98% problem instances are solved by this algorithm. However, for same set of parameters (*n*, *d*, *t*), sometimes solution is computed very early whereas it is late for some instances. We summarize some observations for conflict resolution strategy.

1. A prolonged conflict on an edge leads to decrease in entropy.
2. It cannot be predicted as to when in subsequent iterations – the entropy will decrease or how long a conflict is going to continue.
3. Decrease in entropy can be avoided by resolving conflict as and when it occurs.
4. Resolving a conflict may lead to conflicts in other constraints that did not show any conflict so far.
5. Mostly, conflict does not repeat for same constraint unless there is another atomic relation that is also highly supported by indirect constraint to compete with current set highest.
6. Number of conflicts reduce from nearly 2000 to 20 in a 60 node problem.
7. Though number of constraints with conflicts (new or old) do show an increase but entropy of all constraints is prevented from decreasing due to a conflict.
8. When operation of resolving a conflict reduces weight of an already dominating relation, entropy decreases drastically. These are those constraints that try to stabilize early towards a non-feasible relation.
9. For general IA networks, entropy does stabilize along a default path but not towards feasible relations. We are successful in diverting this search path towards feasible relation in along with stabilizing entropy of constraints.

The algorithm proposed in this study handles the case of networks that are not path consistent as one of the cases of conflicting constraint. We summarize the impact of the resolution of conflicts on the basic characterstic of the underlying network and the entropy.

Case I: Suppose N is not path consistent

$\Rightarrow \exists W_{ij}$ such that $W \cap W_{ij} = \varnothing$

During the algorithm, argmax(W) is inserted into W_{ij}.

$\Rightarrow$ This case reduces to that of a path consistent network (Case II) as given below.

Case II: Suppose N is path consistent

a. Let us assume that $\exists$ conflict for an edge W_{ij} in n^{th} iteration. This implies that by default, the entropy may decrease continuously in subsequent iterations till this conflict lasts (Theorem 5). During the algorithm, insert argmax(W) in W_{ij} converts the original network to N'. This implies that entropy increases from $(n+1)^{th}$ iteration onwards (Theorem 4).
b. Let us assume that $\exists$ another conflict for same constraint in later iteration. Repeat step (a) above to prevent entropy from decreasing due to conflict.

This algorithm computes an approximation of the problem followed by computing a solution for the same. There are three bounded variables-

a. Decrease in entropy due to conflicts is prevented. This means that entropy is forced to increase monotonically.
b. Number of constraints that agree with the indirect constraint increase in number due to the decrease in the number of conflicts.
c. The parent network N is modified to N' such that $\forall\ R_{ij}' \supseteq R_{ij}$. Thus the algorithm maintains the consistency of N. Possibilities for the upper bound of approximation induced by conflict resolution strategy are
    i. Best case – all paths to agree with respect to highest for all constraints.
    ii. Worst case –edge becomes a universal constraint.

Multiple conflicts for same constraint result into multiple relations with competing weights. When all constraints agree i.e highest weight relation is highest supported relation, the network nears the stable state. Stability of most of the constraints forces a choice among multiple candidates.

For the networks that are consistent, the global maxima for every constraint is the set of feasible atomic relations in the constraint. An ideal algorithm would be the one which terminates at one such maximum that belongs to the instance. The algorithm proposed in this study modifies the basic problem by successfully pin pointing the reason for hardness. The search terminates at global maximum that may not belong to the problem instance.

3.6.2 Diagrammatic Interpretation of Nasty Constraints

In this section, we illustrate the behaviour of the search during the iteration of wt_pc algorithm diagrammatically. A simplex triangle represents a weighted constraint with three atomic relations in 3-D with four vertices. The simplex triangle results due to intersection of x + y + z = 1 plane with the three axes. The contours represent the states with equal entropy. The contours result due to the intersection of isentropic spheres with radius of the difference between the three intercepts. This interpretation can be generalized to 13 dimensions but not possible to visualize.

The point A corresponds to the lowest entropy state with co-ordinates (1/3,1/3,1/3) analogous to the equal starting point of the wt_pc algorithm. The vertices C, D, E are highest entropy states. The point B is the lowest entropy state on line CD. In conventional path consistency, we move from A to C in one step or from A to B and then possibly to C. Ideally, any search technique should choose a ascending path from A to one of the vertices, say D. A way of achieving this is to use a PC-like algorithm for weighted network that manipulates weights in same manner as PC manipulates relations.

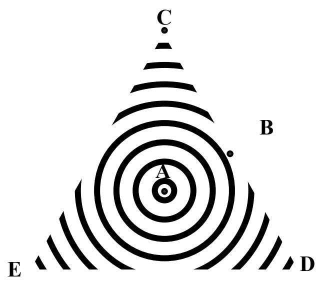


Figure 40. Weighted IA constraint as a simplex.

Convex IA networks

In the event of a solution, the search trajectory terminates at a point in this simplex. The search starts at A in the first iteration. When the problem is solved in the beginning few iterations, it stops near A in the absence of conflict. When the constraint contains competing relations, the search terminates near B as shown in Figure41a. When a single relation dominates in value, it ascends towards a vertex. The constraint stabilizes when the search is very near an axis. The search trajectory never touches any axis, which would otherwise mean a decrease in the number of non-zero terms.

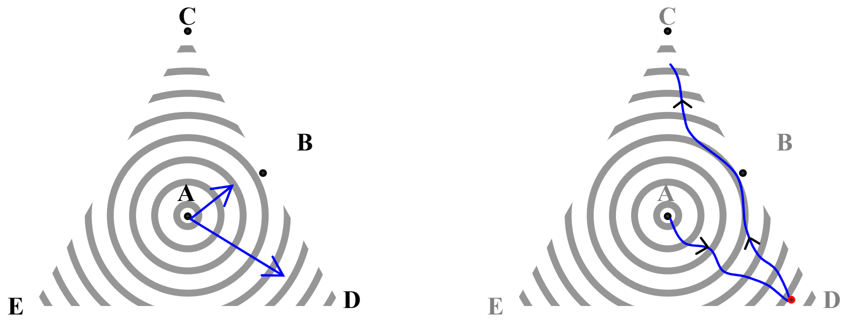


Figure 41. Search trajectory for (a) convex IA constraint, (b) nasty constraint

Nasty Constraints

An increase in highest weight value ascends search towards a vertex. This ascent continues so far as direct and indirect constraints agree i.e. no conflict. In event of a conflict, highest weight value decreases and search towards a vertex recedes back in direction towards another vertex. Before ascending towards another vertex, it passes through B, where weight for two relations is same. Entropy of a nasty constraint decreases due to this shift in maximum. When conflict takes place very near B, entropy does not decrease due to multidimensional nature of search. On the other hand, when conflict takes place when path is near a vertex, entropy decreases as search descends towards B as shown in Figure 41b. Thus a prolonged conflict results in a decrease in entropy of the constraint rendering it as a nasty constraint.

Relaxation due to Conflict Resolution

Search trajectory path becomes longer due to multiple ascents and descents between local maxima. During this, overall characteristic of network is perturbed in the sense that it disturbs trajectory of other constraints. The weight adjustments or new relations inserted by algorithm proposed here shortens this trajectory towards a global maximum(Figure 42).

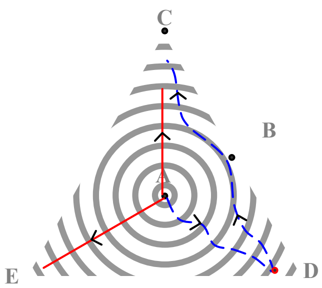


Figure 42. Search trajectory for a nasty constraint.

Entropy of such constraints is prevented to increase to maximum value and forced to decrease to a value near an edge instead of a vertex in the simplex. This is done in an iterative manner till all constraints agree with indirect constraint. A stage will come, beyond which entropy cannot decrease further and hence stabilizes. Stability of constraints helps to choose one among the relations with multiple local maxima.

3.6.3 Experimental Analysis

The objective of first part of experimental analysis is two fold here. Firstly we justify our definition of nasty constraints by showing that known cases of easy instances – convex IA networks are free of nasty constraints. Secondly we show that instances in hard regions have large number of nasty constraints.

Number of Nasty Constraints

We observe by experimenting with instances of convex networks that for every constraint for not even a single instance, entropy fluctuates. We repeat same experiment for general IA instances. We find that unlike the convex case, entropy of some constraints increase after an initial decrease and again continue to increase until stabilization. It is interesting to note that number of nasty constraints are very high for instances in hard region. The number of nasty constraints for different node size for t = 7 are shown in Figure 43.

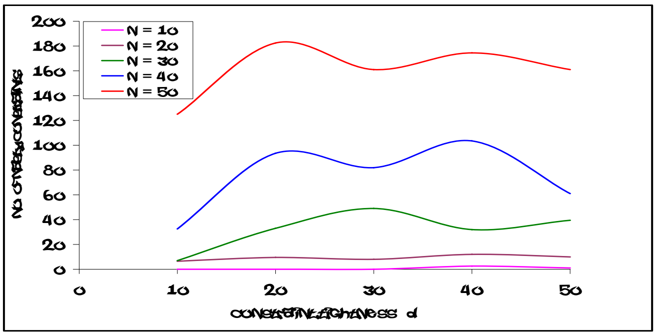


Figure 43. Number of nasty constraints for M(n,d,7) for general IA problems.

It can be seen that hard region instances contain an exceptionally high number of these constraints. For instance, for n = 50, maximum number of nasty constraints are captured when d is around 20% which corresponds to d' = 10. The maxima in this graph are exactly in the region of hard instances. These two observations namely (i) convex networks do not have nasty constraints, (ii) hard problem instances consist of very large number of nasty constraints, corroborates our hypothesis that the nasty constraints as defined here are indeed the root cause of problem hardness. Backtrack algorithm is known to take exponential time in hard region. This fact further helps us to conclude that observed nastiness in constraints is independent of constraint type as well as algorithm type used to solve an instance. Thus, larger the number of nasty constraints in a problem, harder the instance.

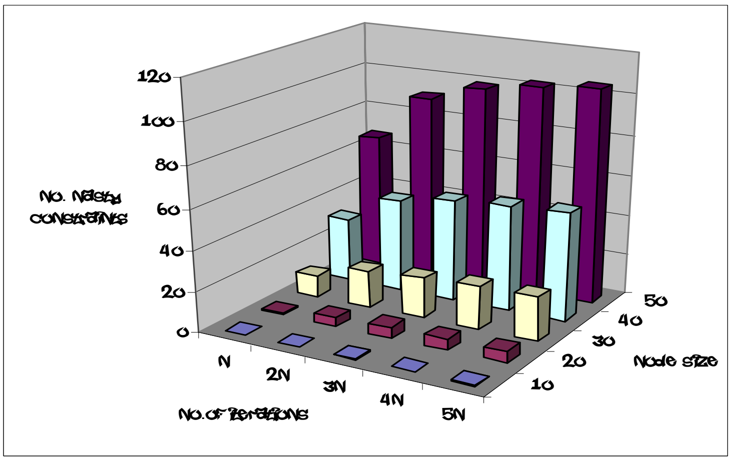


Figure 44. No. of nasty constraints vs *5n* iterations for *n* in the range[10,50]

The proposed method of characterizing hardness in terms of nasty constraints provides us an insight to repair constraints in order to find an approximate solution. This is possible if number of nasty constraints in a problem is bounded above. We observe that for every *n* iterations, number of nasty constraints keep increasing. Figure 44 shows that when T exceeds

5n we do not get any additional nasty constraints. The majority of nasty constraints are detected in linear time and hard instances do not contain an infinite number.

Approximation Algorithm

Objective of experimental analysis is essentially to confirm that proposed paradigm of conflicts based analysis for nasty constraints is valid for general IA networks. We report performance of algorithm proposed in previous section in terms of important parameters – solution iteration number, constraint density, constraint tightness, number constraints required to be approximated to compute an approximate solution and lastly the solution quality in terms of number of constraints that are violated in this process.

Percentage of general IA problems solved

Our method is able to solve 100% of problems for *n*[10,40]. Two instances for 40 nodes and five problems in 60 nodes set of problems are left without a solution with a time out of 1000 iterations.

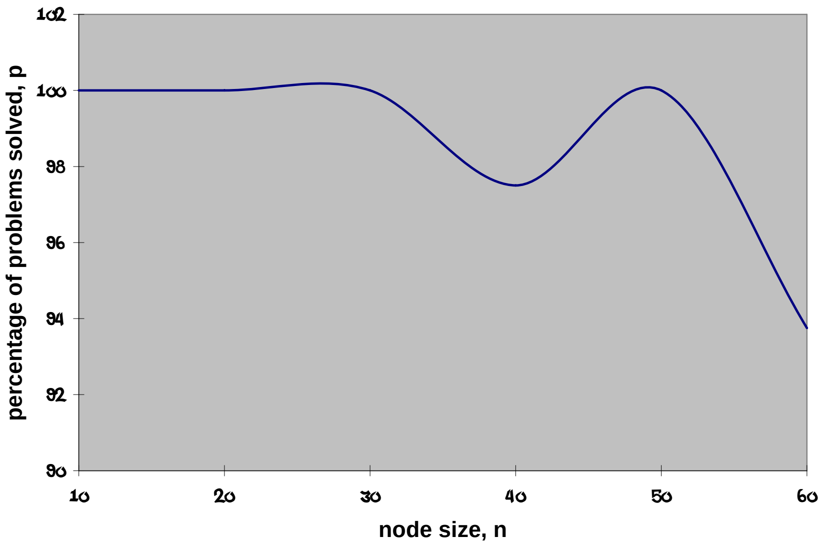


Figure 45. Percentage of problem solved by *compute_approx_solution* for n [10,60]

The graph in Figure 45 shows a dip for percentage of problems of size 40 nodes. One can notice that for 60 node size, we are able to deliver a success rate of 93.75%, a definite improvement as compared to 50% of the best possible conflict handling strategy(Whigh) in section 3.9.2. An average success rate of 98.5% in the range of n[10,60] is definitely a noticeable success for general IA instances . With experience, we admit that this 5% of failure is due to numerical errors that are still left over despite the decimal scaling procedure. It is interesting to note that our method involves a huge amount of numerical computations involving floating point numbers. Therefore, this method is prone to numerical errors at every iteration of wt_pc. We have experienced main difficulties with respect to multiplication of a pair of very small number that is known to be theoretically non-zero but results in a zero value due to underflow in the implementation. To avoid these problems, we have to introduce approximations in floating point arithmetic operations. This adjustment requires renormalization of weighted constraint in order to maintain network consistency as explained in Appendix-F.

Number of iterations required to get a solution

Iteration at which a consistent scenario is obtained(*s*) is noted and average of these is taken for each combination of *n*, *d*. This approach gives an empirical estimate of average number of iterations required to solve a network with size *n* and d% of non-trivial constraints(Figure 46).

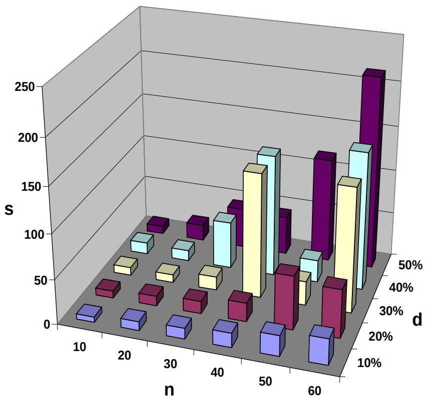


Figure 46. Graph for n vs d.

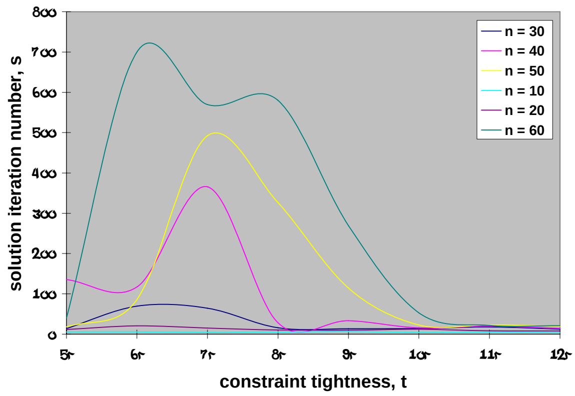


Figure 47. Graph for s vs t

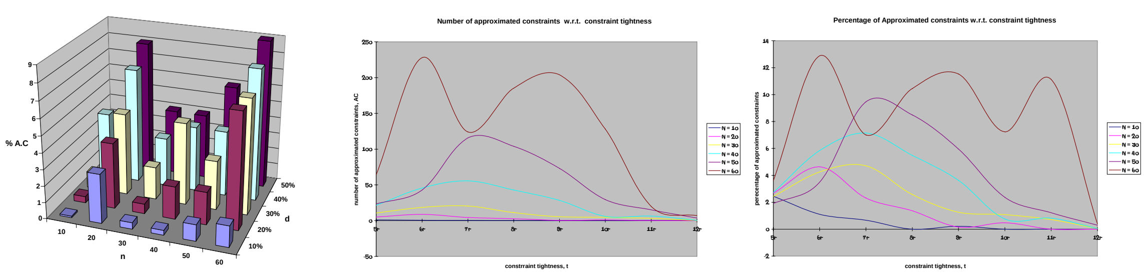
For networks of size in the range [10,30] relatively more iterations are required for higher constraint density. For 10 nodes, this difficulty for the solution increases from 10% to 50%. For 20 and 30 nodes, this difficulty shifts to 40% and 50%. For 40 nodes, this difficulty dominates in the hard region for t in the range[7,10] i.e d in the range [30%,40%] . For 50 and 60 size networks, iteration for late solution gradually increases from d = 10% to 50% with a very late solution at 50% constraint density. We observe that problems in the hard region are among the difficult to solve problems by this algorithm.

In order to study effect of *t*, we take an average of *s* for each value of in range [5,12] for each *n* [10,60] (Figure 47). For small values of *t*, algorithm requires more number of iterations to compute a solution and *s* increases fast as *n* increases for small value of *t*. A late solution is computed in the range of *t* [6,9]. This difficulty in computation increases fast as *n* increases. This shows that difficult region for this algorithm is *t*[6,9]. When *t* is more than 9 algorithm finds a quick solution irrespective of *n* value. This show that region of easy problem instances for this algorithm is beyond 9.

Efficiency in terms of constraint density and constraint tightness

Any comparison with backtrack algorithm will not be in place. We feel that comparison of two methods that give different types of solutions does not helpus. However, an outright advantage of our method can be simply seen by the fact that for 50 and 60 node problems, backtrack is known to take exponentially high computation time, where as our method gives solution in a maximum computation time of 80 minutes, which is equivalent to a maximum of 1000 iterations of wt_pc algorithm for a 60 node problem. This method is able to solve even hard problems in reasonable time despite a large number of nasty constraints.

Efficiency in terms of approximation required to compute a solution

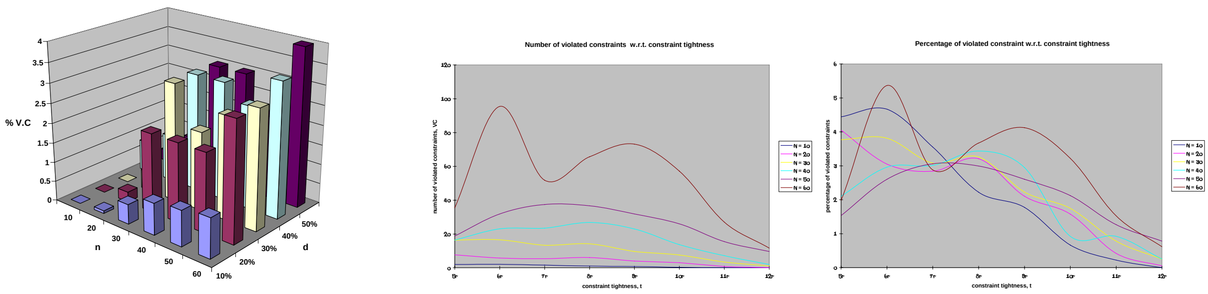


(a) Percentage of constraints approximated for different values of n [10,60] and d [10,50].
(b)-(c) Constraints approximated for different values of n [10,60] and t [5,12].

Figure 48. Dependence between % and no. AC required to solve *N* on *d* and *t*.

The approximated constraints are constraints that contain a conflict (may be repeated). This implies either weight of an existing atomic relation is adjusted or a new one is inserted irrespective of constraint entropy fluctuation. This set of constraints is a subset of nasty constraints. For smaller values for the constraint density, relatively lesser number of constraints are approximated and this percentage of constraints increases very slowly with *n* (Figure 48a). For small values of t (Figure 21b), extent of approximation required to handle conflicts is very less and this number increases very fast when *n* increases for small *t*. On the other hand, when t is in the range [6,8], larger approximation is required irrespective of *n*, this being the hard region. This show that approximation required is maximum for *t* in [6,8] where as less for *t* in [5, 9-12].

Efficiency in terms of quality of the solution computed



(a) Percentage of constraints violated for different values of n [10,60] and d [10,50].
(b)-(c) Constraints violated for different values of n [10,60] and t [5,12].

Figure 49. Dependence between % and no.of violated constraints on *d* and *t*.

For smaller values for constraint density, relatively lesser number of constraints are violated and this percentage of constraints increases very slowly when *n* increases (Figure 49a). For small values of *t* (figure 49b), very few constraints are violated and this number increases very slowly when *n* increases for small *t*. On the other hand, when *t* is in range [6,8], the algorithm violates maximum number of constraints irrespective of *n*. Thus we conclude that the algorithm computes a solution that is an approximate one for instances in the hard region. The quality of the solution improves for values of *t* as we move away from the range [6,8]. This shows that the quality of the solution is worst for *t* in [6,8] where as the quality improves for *t* in [5, 9-12].

The extent of approximation required is 14% and that for violation is only 6%. This number is negligible in comparison to the number of nasty constraints. This clearly shows the positive impact of the conflict resolution strategy that prevents the entropy of the constraints from

fluctuating in the later iterations. This control in turn reduces the number of nasty constraints in a given network that appear more in number during the later iterations.
We conclude that the method has diverted the search outside the given problem instance. The advantage of the approach is clearly evident since we are able to pin point the atomic relation in the constraint that is responsible to the difficulty in the solving the instance.

3.7 Conclusions

The present work introduces a new method to identify easy instances and hard instances in interval algebra. We show that fluctuation in constraint entropy is an important observation to study the structure of instances that are difficult to solve.

We show that the least element based interpretation in the previous chapter can be extended to introduce a new concept high set in an IA instance. We prove that the high sets in convex and preconvex IA networks agree.
In earlier research, the tractability of the known classes rests on the closure property. We provide here an analytical proof for the tractability of these classes. It is shown earlier that path consistency decides consistency for preconvex class using the intricate analysis of modal logic. We prove the same result by a simple technique of high set based instantiation. In order to determine a solution of tractable network, earlier algorithms follow conventional backtrack search with intermediate steps of path consistency. We propose here the best possible algorithm for constructing a solution.

We believe that the present study has opened a new direction of research for investigation of the reasons of tractability. We intend to extend this study to investigate the structure of sets of relations that is sufficient to satisfy the property of non-empty high set. This analysis reveals another interesting aspect that tractability needs to be studied at the network level instead of the stringent requirement of closure at the level of constraints.

In earlier research on the existing weighted paradigm for IA networks, a highest weight heuristic is proposed to solve general IA networks. We show that nasty constraints based interpretation of IA networks helps us study the structural properties that make an instance difficult to solve.
Besides improving the existing results, the framework based on nasty constraints is potentially useful in arriving at many new results. We analytically prove that fluctuations in

constraint entropy is the reason for difficulty. We provide a clear understanding that a constraint entropy can fluctuate only due to the presence of nasty constraints.

In earlier research, it is shown that highest weight heuristic solves all the instances of convex IA networks. We provide an insight as to why it is so. We not only prove that this is due to the absence of nasty constraints in convex networks but also extend the same for preconvex networks.

We propose an algorithm that computes an approximation of general IA problems and then solves it. This is a linear time algorithm that captures the hardness of the problem in terms of nasty constraints. Our method is able to exploit structure of individual problems. Exact solution computation is a special case of this method i.e. no nasty constraint is present which in turn implies that absence of fluctuations in the constraint entropy. In the process of handling the conflicts, the link with the original problem is not lost. User intervention is possible for an interactive choice of which nasty constraints to be settled, that may be crucial to the problem. It is possible to keep track of which new atomic relations are resolved iteration-wise. User can analyze the impact of avoiding or choosing a new atomic relation. We propose to extend this study for networks of higher size.

We believe that the present study is going to open a new direction of research for tractability in qualitative CSPs. We intend to study the extent of approximation required based on the structure of any given general IA instance. In future, we intend to devise a single-step algorithm to construct an $\in$-approximate solution for general IA instances for other frameworks.

The described work in the chapter is published as:

1. Chandra, P. Pujari, A K 2005. Complete Instantiation Strategy for Approximate Solution of TCSP. In Alexander Gelbukh and Raúl Lonroy (eds.): *Advances in Artificial Intelligence Theory: Proceedings of the 4th Mexican International Conference on Artificial Intelligence*, pp. 55-64, Monterrey, Mexico, Instituto Politecnico National, Mexico.
2. Pujari, A K, Adilakshmi, T. Chandra, P 2005. Constrainedness Analysis of TCSP using Entropy. *Proc. Symposium on AI-Methods*, Nov 16-18, Gliwice, Poland.

CHAPTER 4

APPLICATION: PREDICTION OF NEXT ROAD TRAFFIC SCENE

4.1 Introduction

In this chapter, we propose a framework to study structure of spatial and temporal networks that represent a scene in the domain of road traffic. We predict the next scene based on the current scene using IA and RCC-8 networks. We investgate the benefit of complete instantiation strategy based interpretation of networks in this application. This approach is a step away from the conventional backtrack-based approach for an application based on a constraint satisfaction problem. We start with an instantiation of the variables. The instantiation strategy exploits the neighborhood property to help us to predict the state of the variables in the next scene. This approach helps us to identify different ways in which occlusion can take place. We propose a strategy to distinguish between the case of occlusion and an object leaving a scene.

4.2 Problem Definition

Consider a road traffic scene at a given section of road over an interval of time. Any number of vehicles can enter/leave a scene during the period of observation. The vehicles move in a continuous manner resulting into a motion trajectory starting from the moment it enters the scene till the exit. Any given road traffic scene can consist of either types of vehicles – stationary or moving. Depending on the activity at the road, a vehicle may not always be visible during the entire course of movement trajectory. In such cases, a vehicle is said to be occluded by other vehicle(s) in the scene. The number of vehicles in a given scene changes due to three reasons, namely occluding vehicles, new vehicles entering or the existing ones leaving.

We capture road traffic activity using a fixed camera as a digital video movie film with a capture rate of 25 frames per second. This movie file is converted into a series of frames using a freely available software – XVideoConverter(http://www.aoamedia.com). Each frame is manually edited to represent each vehicle by a two-dimensional minimum bounding rectangle termed as a *blob*. Each frame contains a group of blobs arranged in some fashion within the frame rectangular limits.

Definition 1: A *snapshot instant* is defined as the instant when a spatial relation changes between atleast one pair of objects in a scene.

There are three semantic possibilities for the projection of movement trajectory namely time, frame and snapshots instant axes (Figure 50). A mapping between the three is not necessarily a one-to-one mapping. Due to the frame capture rate, 25 frames on the frame-axis are equivalent to one second on time-axis. However, a similar mapping between snapshot instant axis and the other two cannot be fixed. If the scene is a fast changing one, say consisting of small fast moving vehicles(two wheelers), the snapshot instants will be very close and dense. For slow changing scene that involve slower moving vehicles(four wheelers), these instants are far apart in time.

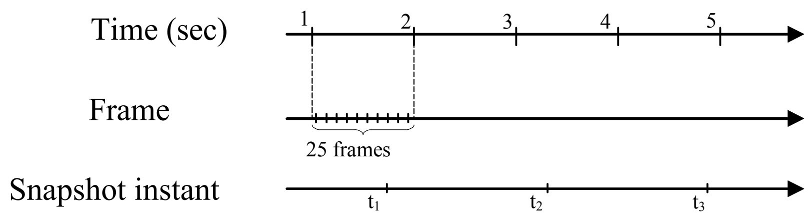


Figure 50. Mapping of the time, frame and snapshot instant axes.

Frames corresponding to the snapshot instants are again an ordered but discontinuous sequence of time instants. In other words, the frames along the snapshot axis are a group of unique spatial arrangements. Given a spatial arrangement at a snapshot instant, the problem is to predict the spatial arrangement of blobs that is possible at the next snapshot instant.

4.3 Candidate Network based on RCC-8 Semi Lattice

In this section, we exploit the neighborhood property of RCC-8 semi lattice to predict the scene at the next snapshot instant. We explore a new approach that is a step away from from the conventional way where the starting state is a disjunctive network. The given scene is a complete instantiation i.e. a singleton RCC-8 network. The neighbours for each atomic relation varies from one to a maximum of four as shown in the Table 9.

r	***Neighbour*(*r*)**
DC	EC
EC	DC, PO
PO	EC, TPP, TPPI
TPP	PO, NTPP, EQ
NTPP	TPP

TPPI	PO, NTPPI, EQ
NTPPI	TPPI
EQ	TPP, TPPI

Table 9. Neighbourhood for RCC-8 atomic relations

This neighbourhood is formalized as a disjunctive network that is parent to all the possibilities.

Definition 2: *Candidate network set C(N)* for a given RCC-8 singleton network *N* is defined as the set of singleton networks generated from the disjunctive network *P(N)* denoted as:

$$P_{ij} = \vee\ neighbour(S_{ij})$$

where neighbor(S_{ij}) is an immediate succeeding or preceding relation of atomic relation S_{ij}.

For instance, consider an example scene(N_1) with parent and candidate set of networks in Figure 51. Each constraint network is denoted as an upper triangular matrix.

$$N: \begin{matrix} PO & DC & DC & DC \\ & PO & DC & DC \\ & & DC & DC \\ & & & PO \end{matrix}$$

$$P(N): \begin{matrix} EC,PO,TPP,TPPI & DC,EC & DC,EC & DC,EC \\ & EC,PO,TPP,TPPI & DC,EC & DC,EC \\ & & DC,EC & DC,EC \\ & & & EC,PO,TPP,TPPI \end{matrix}$$

Figure 51. Example network and disjunctive parent network for the candidate set.

The candidate set is computed by exhaustively listing all the possible singleton combinations. The candidate set for this example contains 8192 possible networks with only 2692 are path consistent networks. The drawback is clearly the exponentially high number. This method helps us to synthesize all candidate scenes that can follow a given scene but in actual practice it may not be possible to realize every scene. A strategy to reduce this number would be helpful to directly pinpoint the pair of blobs where the spatial relation will change first. The objective is to match the next scene with a reasonable number of possibilities before predicting the next one in the sequel. In this study we investigate this prediction and matching problem with the help of qualitative temporal and spatial networks.

4.4 Temporal model to reduce number of candidate networks

In this section, we model the scene activity using interval algebra constraint networks. The starting point is a rudimentary interval algebra network model that is incrementally updated as the snapshot instants cross by. The rich reasoning techniques from conventional interval algebra and weighted paradigm are investigated to predict the spatial relation that will change in the current scene.

4.4.1 Temporal interpretation of the road traffic scene

We can visualize movement trajector of each blob from the side view in space and time dimensions along y, x axes respectively as shown in Figure 52. We ignore other dimensions of space – width and depth of the cuboid that contains the blob. The size of a blob may vary from frame to frame when the vehicle approach/recede the camera. This change can be ignored in this study since we are trying to qualitatively model the scene activity. The concept of toplogical relation remains the same despite the varying size.

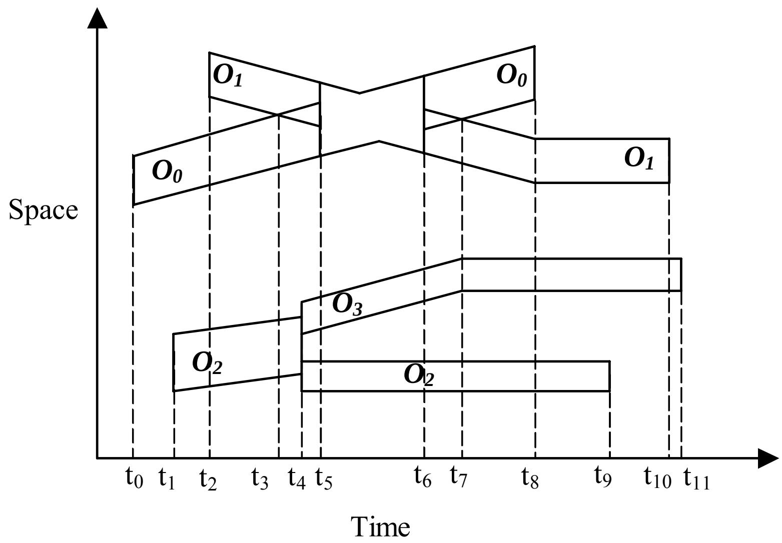


Figure 52. Spatio-temporal interpretation of blob movement trajectories in a scene.

The example scene contains trajectories of four objects - O_0, O_1, O_2 and O_3. When a trajectory spans over a sequence of frames, this does not always implies that the object is moving. It merely indicates presence of an object in a frame, whether stationary or moving. Such an interpretation helps us to study the change in relative arrangement of the blobs during the observation time. The scene starts with only one object O_0 followed by O_2 entering at t_1 and O_1 at t_2. O_0 and O_1 start approaching each other as shown by the acute angle θ_{01} between their trajectories as shown in Figure 53. They remain very close to each other as a single blob from t_3 to t_7. They start separating out at t_7 shown by diverging angle α_{01} between their trajectories at s_3. The trajectory of O_0 ends at t_8, which implies that O_0 leaves scene at t_8. The angle of approach and angle of recede may not always be same in general. A simple picture reading of this diagram helps in summarizing pair-wise angular change in directions.

The change in direction of trajectories of O_1 and O_2 at s_5 has in a way avoided a possible collision at point of intersection C of trajectories. This interpretation helps in detecting converging or diverging nature of any pair of objects in the scene.

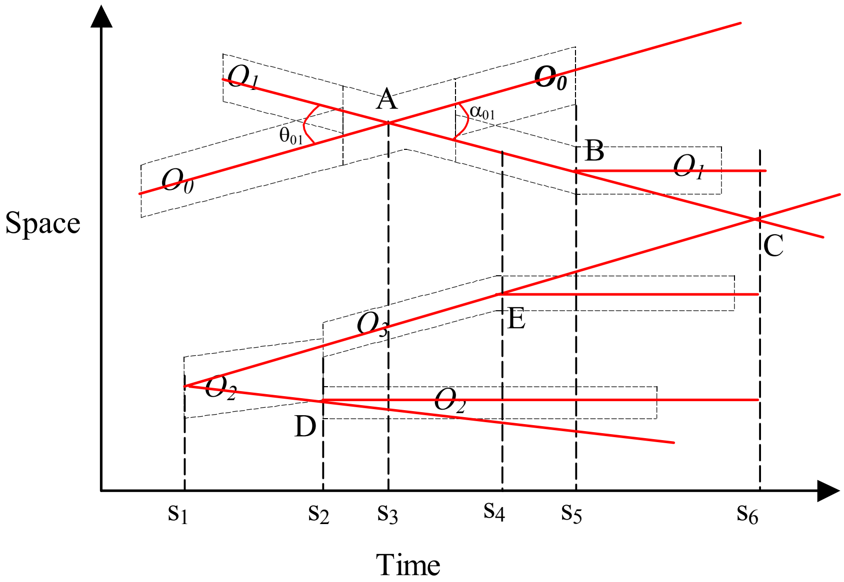


Figure 53. Geometric interpretation of movement trajectories of blobs.

An occlusion may take place between objects in various ways. A pair of objects occlude each other during motion. A stationary object may be occluded by a moving object as it crosses the view. A pair of objects may enter the scene together in an occluded fashion such that they seem to be a single blob. The objects participating in occlusion may separate out in various fashions thereby resulting in splitting of a blob into a number of blobs. This split may be vertical or horizontal. When the objects separate sideways, the blob splits vertically. For instance, O_0 and O_1 combine into a single blob at t_5 and separate out again at t_6 by splitting vertically. Horizontal splits are possible in two ways. First, when the objects move down a flyover and an object merges into another from the upper margin. Similar is the case for split when one moves down the flyover faster than another. Secondly, on a flat road when an object merges into another from the upper margin and later slows down and left behind. An instance of a vertical split is shown in Figure 53 between O_2 and O_3. The vertical split appears as a discontinuity at t_4, since we ignore second dimension of space.

4.3.2 Interval Algebra based Temporal Model of a Road Traffic Scene

We formally represent the temporal interpretation discussed above as interval algebra constraint networks. The pair-wise spatial relation between any pair of objects changes continuously in a sequential manner.

Definition 3: An *event* is defined as the time interval for which a particular spatial relation holds between a pair of blobs.

Relationship between any pair of blobs during the complete life of the scene is represented as a sequence of events. The subsequent event starts immediately after the current one terminates. For instance, O_0 DC O_1 holds during [t_2,t_3] and O_0 EC O_1 spans for [t_3,t_5].

Definition 4: A *temporal track* is defined as a temporally ordered sequence of events that are observed for a pair of objects.

A given scene with n objects can have a maximum of $^{n}C_2$ temporal tracks that can be active during the complete scene. A temporal track is created when either a new blob enters or an existing parent blob splits into a number of children blobs. We study the behaviour of movement of objects with the help of a sequence diagram. For instance, the scene activity in Figure 52 is depicted as a set of temporal tracks in Figure 54. There are six temporal tracks namely $R_{01}, R_{02}, R_{12}, R_{03}, R_{13}, R_{23}$ numbered from 1 to 6. The first track that becomes active is the track number 2 since O_0 and O_2 is the first pair of objects in the scene.

Each temporal track contains the spatio-temporal history for a pair of objects in the scene. For instance, the blobs for the object pair O_0, O_1 have remained disconnected for [t_2,t_3], touch for [t_3, t_5], overlap for [t_5, t_6] followed by a touch relation for [t_6, t_7] and subsequently remain disconnected till the remaining active period for the track.

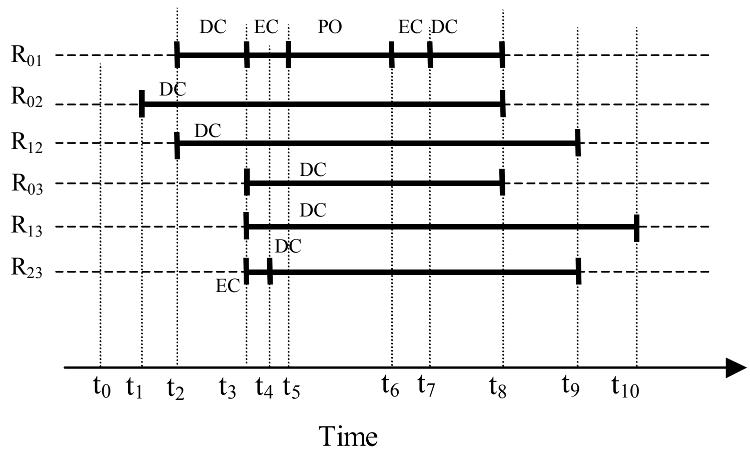
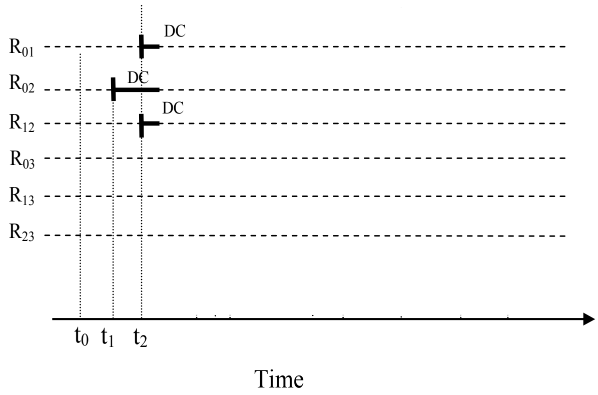


Figure 54. Temporal sequence diagram for the example scene of Figure 52.

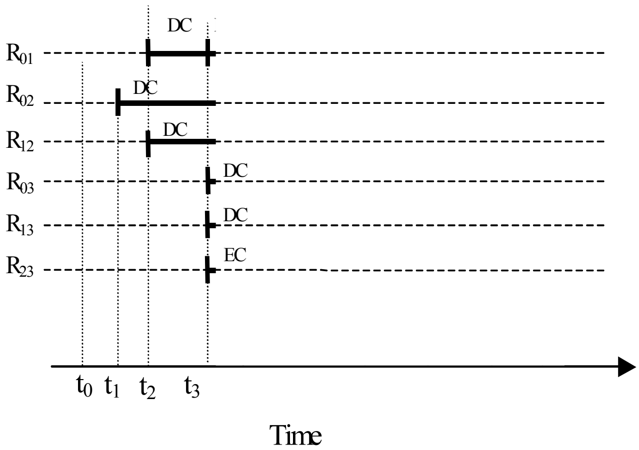
The objective is to predict the temporal track that will change before others. The temporal networks for the first four snapshot instants are shown in Figure 55.

	1a	2a	3a
1a		d f oi	s eq si
2a			o di fi
3a			



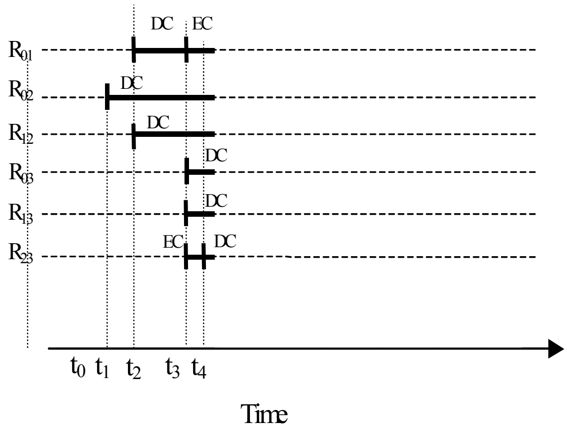
a. Sequence diagram and IA network at second snapshot instant t_2.

	1a	2a	3a	4a	5a	6a	1b
1a		d	s	m	m	m	m
2a			o di fi	o di fi	o di fi	o di fi	o di fi
3a				o di fi	o di fi	o di fi	o di fi
4a					s eq si	s eq si	s eq si
5a						s eq si	s eq si
6a							s eq si
1b							



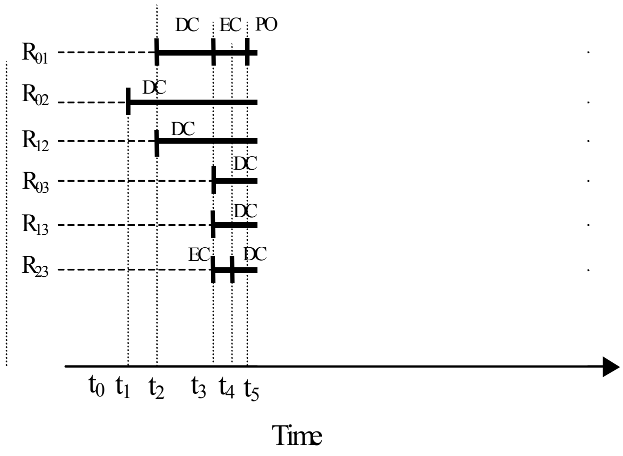
b. Sequence diagram and IA network at third snapshot instant t_3.

	1a	2a	3a	4a	5a	6a	1b	6b
1a		d	s	m	m	m	m	b
2a			o di fi	o di fi	o di fi	di	o di fi	o di fi
3a				o di fi	o di fi	di	o di fi	o di fi
4a					s eq si	si	s eq si	o di fi
5a						si	s eq si	o di fi
6a							s	m
1b								o di fi
6b								



c. Sequence diagram and IA network at fourth snapshot instant t_4.

	1a	2a	3a	4a	5a	6a	1b	6b	1c
1a		d	s	m	m	m	m	b	b
2a			o di fi	o di fi	o di fi	di	di	di	o di fi
3a				o di fi	o di fi	di	di	di	o di fi
4a					s eq si	si	si	di	o di fi
5a						si	si	di	o di fi
6a							s	m	b
1b								o di fi	m
6b									o di fi
1c									



d. Sequence diagram and IA network at fifth snapshot instant t_5.

Figure 55. IA networks with sequence diagrams at 4 snapshots for scene in Figure 52.

4.4.2 Disjunctive IA network based reasoning to reduce candidate networks

The sequence diagrams discussed above contains the spatio-temporal history of the complete scene. Events along each track are labeled alphabetically. Event along some of these tracks may not terminate at an immediate subsequent snapshot instant. This results into longer events in some tracks. For instance, at the beginning instant, an event is initiated along each track but none of them gets terminated till second snapshot. Edges in the network contain interval algebra relations between both inter-track and intra-track nodes(events). The constraint that involves atleast one non-terminated event is a disjunctive IA constraint. Over a passage of time, as the snapshot instants are crossed, the set of completed events increase in number. The termination of any single event results in conversion of disjunctive constraint into a singleton relation for those relations in which this event participates. The insertion of a new node in the network results into a set of disjunctions. The number of disjuncts will always reduce in the existing constraints and a number of new constraints are always introduced as a disjunction

The interval algebra based temporal model of a traffic scene can help us in to predict the next possible change along the temporal track(s). The robustness of this prediction is key to sharply reduce the search space for candidate network possibilities for the subsequent spatial network. In order to utilize the IA model for prediction, we need to insert one additional node for each active temporal track. The IA relation of this additional node with the existing ones will help us in accurately predicting as to which of the ongoing events on these tracks is going to complete first. These next event nodes bring in more nodes and more disjunctive constraints in the consolidated IA network.

Example Consolidated IA networks for three snapshot instants are shown in Figure 56 (a-c).

	1a	**2a**	**3a**	**1b**	**2b**	**3b**
1a		d f oi	s eq si	m	U	U
2a			o di fi	U	m	U
3a				U	U	M
1b					U	U
2b						U
3b						

a. Consolidated IA network at t2.

	1a	**2a**	**3a**	**4a**	**5a**	**6a**	**1b**	**2b**	**3b**	**4b**	**5b**	**6b**	**1c**
1a		**d**	**s**	**m**	**m**	**m**	**m**	**U**	**U**	**U**	**U**	**U**	**b**
2a			**o di fi**	**o di fi**	**o di fi**	**o di fi**	**o di fi**	**m**	**U**	**U**	**U**	**U**	**U**
3a				**o di fi**	**o di fi**	**o di fi**	**o di fi**	**U**	**m**	**U**	**U**	**U**	**U**
4a					**s eq si**	**s eq si**	**s eq si**	**U**	**U**	**m**	**U**	**U**	**U**
5a						**s eq si**	**s eq si**	**U**	**U**	**U**	**m**	**U**	**U**
6a							**s eq si**	**U**	**U**	**U**	**U**	**m**	**U**
1b								**U**	**U**	**U**	**U**	**U**	**m**
2b									**U**	**U**	**U**	**U**	**U**
3b										**U**	**U**	**U**	**U**
4b											**U**	**U**	**U**
5b												**U**	**U**
6b													**U**
1c													

b. Consolidated IA network at t_3.

	1a	**2a**	**3a**	**4a**	**5a**	**6a**	**1b**	**2b**	**3b**	**4b**	**5b**	**6b**	**1c**	**6c**
1a		**d**	**s**	**m**	**m**	**m**	**m**	**U**	**U**	**U**	**U**	**U**	**b**	**U**
2a			**o di fi**	**o di fi**	**o di fi**	**di**	**o di fi**	**m**	**U**	**U**	**U**	**U**	**o di fi**	**U**
3a				**o di fi**	**o di fi**	**di**	**o di fi**	**U**	**m**	**U**	**U**	**U**	**o di fi**	**U**
4a					**s eq si**	**si**	**s eq si**	**U**	**U**	**m**	**U**	**U**	**o di fi**	**U**
5a						**si**	**s eq si**	**U**	**U**	**U**	**m**	**U**	**o di fi**	**U**

6a							**s**	**U**	**U**	**U**	**U**	**m**	**m**	**b**
1b								**U**	**U**	**U**	**U**	**U**	**o di fi**	**U**
2b									**U**	**U**	**U**	**U**	**U**	**U**
3b										**U**	**U**	**U**	**U**	**U**
4b											**U**	**U**	**U**	**U**
5b												**U**	**U**	**U**
6b													**U**	**m**
1c														**U**
6c														

c. Consolidated IA network at t_4.

Figure 56. Consolidated IA nets with additional node along each active temporal track.

From the given set of consolidated IA networks till i^{th} snapshot instant t_i, we require to compute the next possible change in the event list. In other words, given a scenario consisting of completed and ongoing events, we have to predict which ongoing event will complete first. We enforce path consistency on the consolidated network at t_i to restrict the other disjunctive relations involving the new nodes. It is interesting to note that the relations in the networks in this domain are IA convex relations. The number of nodes increase drastically with respect to an increase in the snapshot instants. The traditional solution technique namely backtrack search with forward checking can help us in reducing the search space if the number of nodes do not increase stage wise. The backtrack method with forward checking gives very large number of solutions for this network.

4.4.4 Weighted IA network based model to reduce candidate networks

We observe that traditional solution technique provide us with enormous number of possibilities for the next possible scene. We enhance this model with the help of a more expressive formalism that provides a tool to incorporate some bias that helps in reducing the search space.

We categorize events in the temporal tracks into three categories namely *completed*, *incomplete* and *future* events. An event is said to be a completed event when the spatial relation between a pair of blobs has changed to another, thereby marking the start and end points of the event. An incomplete event is one that indicates a spatial relation is still holding, thereby only the start point is known but the end-point is still not known. A future event is one that is to be predicted to start immediately after the current incomplete event finishes. Both the start and end points of a future event are not known.

To begin with, every active temporal track consists of an incomplete event. The consolidated IA network(N) at the first snapshot(t_1) is converted into the weighted network W(N) by assigning equal weights to disjuncts in all the constraints. At each subsequent snapshot instant, there are three possibilities for tracks – number of active tracks either remain same, increase or decrease in number.

In the first case, an incomplete event of atleast one track completes thereby converting the future event for the track into an incomplete event and initiating another future event. During this process, the existing disjunctive constraint either reduces to a subset or directly to a singleton. The weight values for this constraint are renormalized to reflect the change. In the second case, atleast one new track starts with an incomplete event. During this process, a new node is inserted into the weighted network. Lastly, an active track finishes with an incomplete event completing and droping the future event. During this process, an existing node is deleted from the weighted network.

The temporal model is incrementally updated at each subsequent snapshot instant. We compute the solution of this weighted network based on highest weight algorithm. The constraints where only the weights are modified carry forward the knowledge as trained weights that help in evolving better subsequent predictions. The pseudocode of the algorithm for predicting the next change based on the complete spatio-temporal history of the scene captured right from the beginning till the current snapshot instant is shown in Table 10.

```
at  t1 :  for each active Temporal Track create an Incomplete Event and a Future Event
          number of nodes = no. of active tracks × 2
          //construct the IA network N
          for each Rij do
                    if j is an incomplete event Rij = IAdisjunction(i,j)
                    if j is a future event and i,j belong to same track       Rij = "meets"
                    if j is a future event and i,j belong to different track  Rij = U
          enddo

          //process for each snapshot instant
          enforce path consistency on N
          convert N into W(N)
at ti:    for each snapshot instant ti do
```

case I: same number of active temporal tracks
    for each modified track T do
        // modifyTrack(T)
        convert the incomplete event into a complete event
        update W(N)
        create a future event
    enddo
case II: number of active tracks increase in number
    for each new track: insert an incomplete and a future event into W(N)
case III: number of active tracks decrease in number
    for each terminated track do
        convert the incomplete event into completed event
        remove the future event
    enddo
enddo

Table 10. Pseudocode to predict temporal scenario at next snapshot.

By the property of highest weight, it introduces the bias for 2-D temporal relations b, bi, d, di, o, oi. In an application domain, it is less likely that the end points of any two events will match. Hence choice of an algorithm that is biased towards these relations is justified. This solution technique helps in computing a single solution that is the single possibility for the next change in the scene. The solution networks for IA networks in Figure 56 are shown in Figure 57 along with corresponding scenarios for subsequent snapshot instants in Figure 58.

	1a	2a	3a	1b	2b	3b
1a		d	si	m	b	o
2a			di	o	m	o
3a				b	b	m
1b					di	oi
2b						d
3b						

a. Solution network for t_2 snapshot computed at 157^{th} iteration.

	1a	2a	3a	1b	2b	3b
1a		d	s	m	b	b
2a			di	o	m	o
3a				o	b	m
1b					di	di
2b						d
3b						

b. Solution network for t_2 snapshot computed at 159^{th} iteration.

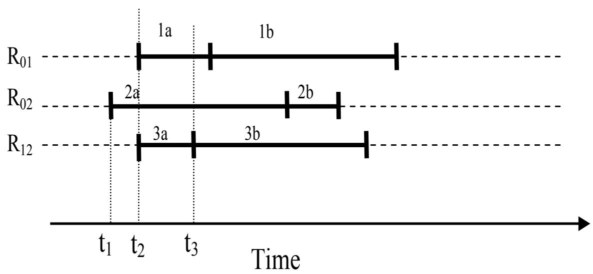
	1a	2a	3a	4a	5a	6a	1b	2b	3b	4b	5b	6b	1c
1a		d	s	m	b	b	m	m	m	b	b	b	b
2a			di	o	m	o	di	di	di	di	di	di	b
3a				o	b	m	o	o	o	b	b	b	b
1b					di	o	si	si	si	di	di	di	m
2b						d	bi	bi	bi	bi	bi	bi	b
3b							oi	oi	oi	di	di	di	di
4a								s	s	m	b	o	b
5a									si	o	m	o	b
6a										b	b	m	b
4b											di	d	b
5b												d	b
6b													b
1c													

c. Solution network for t_3 snapshot computed at 297^{th} iteration.

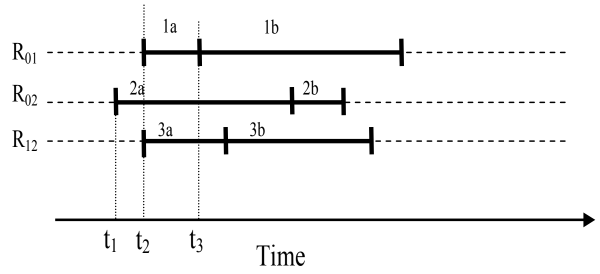
	1a	2a	3a	4a	5a	6a	1b	2b	3b	4b	5b	6b	1c	6c
1a		d	s	m	b	b	m	m	m	b	b	b	b	b
2a			di	o	m	di	di	di	di	di	di	di	b	di
3a				o	b	m	o	o	di	b	b	o	b	b
1b					di	di	si	si	si	di	di	di	m	di
2b						bi	bi	bi	bi	bi	bi	bi	b	bi
3b							oi	oi	bi	di	di	d	b	b
4a								s	si	m	b	o	b	b
5a									si	o	m	o	b	b
6a										b	b	m	b	b
4b											di	d	b	b
5b												d	b	b
6b													b	m
1c														bi
6c														

d. Solution IA network for t_4 snapshot computed at 1423^{rd} iteration.

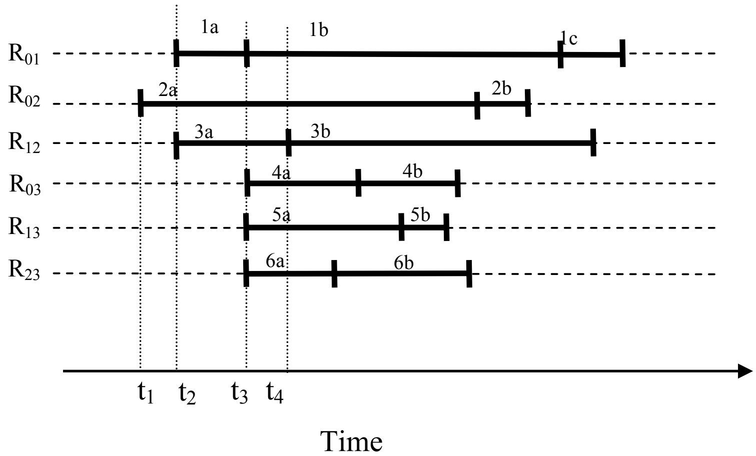
Figure 57. Solution based on highest weight solution technique.



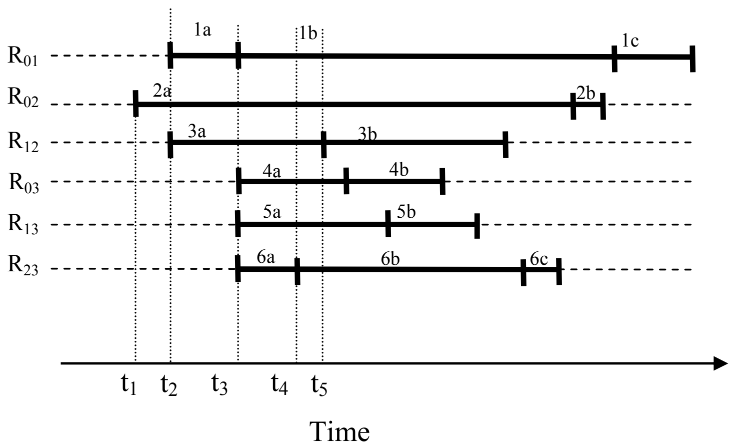
a. First possible scenario for t_3 predicted at t_2.



b. Second possible scenario for t_3 predicted at t_2.



c. Possible scenario for t_4 predicted at t_3.



d. Possible scenario for t_5 predicted at t_4.

Figure 58. Predictions based on highest weight solution.

We can compare these stage wise predictions with actual scenarios in Figure 55. We are able to predict two possible scenarios for snapshot t_3 at 157 and 159^{th} iterations of algorithm. Out of the two, later one is a correct prediction. The rest of the predictions are incorrect. By experimentation with similar such cases, we are facing a poor success rate of 25% for prediction based on temporal model. Moreover the highest weight technique reduces the search space narrowly to a single solution. We require a relatively better approach that gradually narrows down the search space to a smaller group of possible solutions.

4.5 Spatial analysis to reduce number of candidate networks

In this section, we improve the spatial model the captures activity in a scene. The basic motivation to reduce the number of networks in the candidate set is based on the fact that for any given scene, it is not possible that every relation between every pair of objects changes in the subsequent scene that follows. We simplify this problem by introducing the assumption of a single change in any given network at any moment. The prediction of next scene is based on the graph matching problem.

Definition 5: An *episode* is defined as a continuous stream of frames starting from the instant t_s till the ending instant t_e.

Any given episode begins with a scene containing a set of objects and the pair-wise relations. This network of relations (N_1) continues for some frames till it transforms to another network (N_2). Although the frame capture rate is as fine as 25fps, there always exists atleast few number of frames between any two different networks. Thus a given episode contains a set of such distinct pairs of networks N_1 and N_2. The pseudocode to compute the set of candidate networks is shown below in Table 11.

```
Generate candidateSet(N1)
        Input: a singleton RCC-8 network N1
        Output: The candidate set of networks C(N1) for N1
Initialize C(N1) as a null set
∀ Rij ∈ N1 do
        NeighbourhoodSet = ∨ neighbour(Rij)
        N ← N1
        ∀ r ∈ NeighbourhoodSet do
                Rij(N) ← r
        enddo
        If N is path consistent insert N into C(N1)
        N ← N1
enddo
```

Table 11. Algorithm for generation of candidate networks.

This reduces the search space i.e. the candidate networks for example in Figure 50 from 8192 to only 24 networks. This simplifies the problem of networks matching tremendously. Each member network in this candidate set is matched with the network N_2 that represents the scene that immediately follows N_1.

4.5.1 Graph Matching

In the present context, the blobs are not labeled. In the absence of labeling, it is not possible to trace the movement of any particular object qualitatively. We have to match the pair of graphs based on the qualitative relation between the objects. We opt the conventional way of graph matching based on isomorphism of graphs. The problem of graph isomorphism helps in determining whether two graphs can be drawn with identical graph drawings.

Definition 6: Given two graphs G_1 and G_2, G_1 is said to an *isomorphic form* of the graph G_2 if it is possible to permute (or *relabel*) the vertices of G_1 so that it is equal to G_2.

A given graph of n-nodes can be permuted in $n!$ ways. For instance any 4-node graph will contain 24 isomorphic forms as shown in the figure 59. The six edges in the four-node network are labeled as a,b,c,d,e,f. We pinpoint to the edge 01 and place it on each of the six edges in the graph that includes four sides of the rectangle and the two diagonals. We obtain the six rows in the diagram. In each row, two graphs correspond to the clockwise position of the vertices 0 and 1. For each position of this pair of vertices, we can place the other two vertices in the clockwise and counterclockwise position. Each row in the diagram contains four graphs. Any of these forms can be converted into the other form by a mere rearrangement of the vertices.

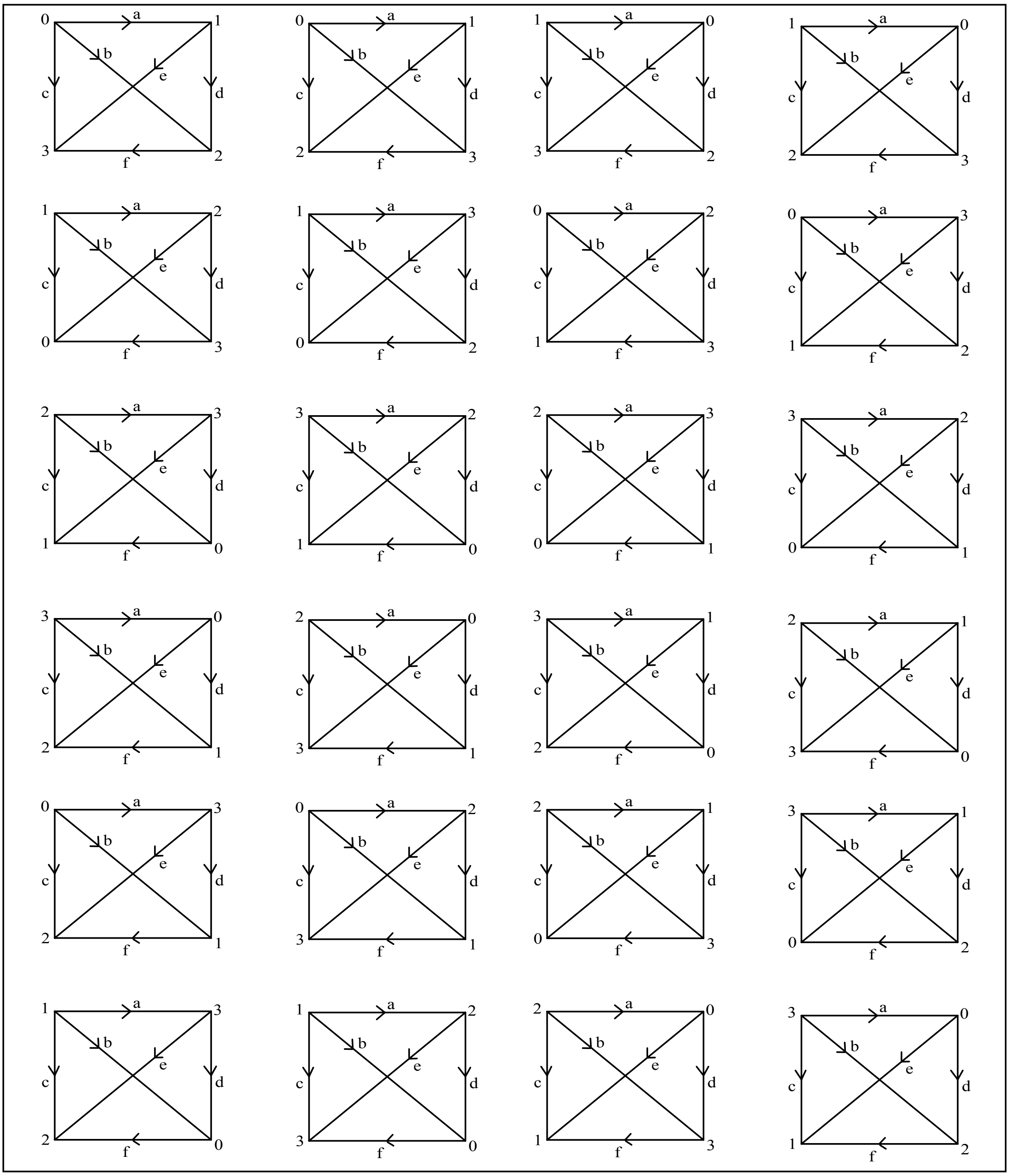
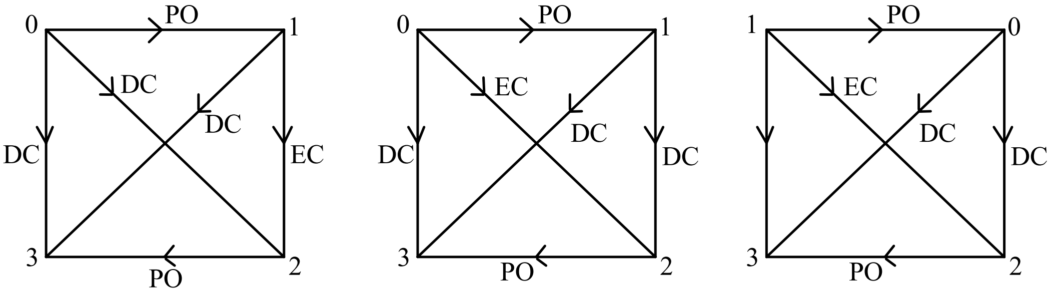


Figure 59. 24-isomorphic forms for a 4-node network.

Consider for instance the two example RCC-8 networks shown in the figure 60a,b. The graph can be obtained from figure 60b if we relabel the vertices in figure 60b by interchanging the labels of the vertices 0 and 1 as shown in figure 60c.



a, b Two isomorphic forms. c. graph a relabeled to graph b.
Figure 60. Three isomorphic forms of a four-node RCC-8 network.

Similarly, the isomorphic forms of any given *n*-node network can be calculated. For 3-node network, there are six isomorphic forms. This definition holds true for the case when the two graphs consist of the same number of nodes. The difficulty arises when the two do not conatin the same number of nodes. In such a case, we look for a *n*-node network embedded into a larger *m*-node network. In the present context $m = n+1$, which means that this study is limited to only one object leaving / entering at any given moment.

Definition 7: A *true n-node network* is defined as a *n*-node RCC-8 singleton network such that each relation in the network satisfies the condition

$$\forall R_{ij}, R_{ij} \cap \{U/DC, EC, PO\} = \varnothing$$

This condition is required to be checked only when the number of nodes is not the same. It is possible to generate *n* no. of (*n*-1) node subnetworks from any given *n*-node network. The pseudocode for matching the subsequent network with candidate set is given in Table 12.

MatchNetwork(N_2, $C(N_1)$)
  Input: Pair of networks N_2 and N_1, Set of candidate networks $C(N_1)$
  Output: True – if N_2 matches with atleast one of the networks in $C(N_1)$
  P_{next} - The set of networks that matched with N_2
  P_{dec} – No.of candidate networks that contain an occluded object
Case I: **nodes(N_1) = nodes(N_2)**
    return true if N_2 is an isomorphic form of atleast one network in $C(N_1)$
Case II: **nodes(N_1) > nodes(N_2)**
  $\forall N \in C(N_1)$ do
    $\forall S \in {}^{n1}C_{n1-1}$ subnetworks of N such that $\forall R_{ij}, R_{ij} \cap \{U/DC, EC, PO\} = \varnothing$ do
      if S is an isomorphicform of N_2
        insert S into $\mathbf{P_{next}}$

```
if ∃ R_ij ∈ (N\S)
    such that (R_ij(N_1) = PO) ∧ (R_ij(N) = TPP(I))
    increment nMatches
    insert S into P_dec
endif
        endif
    enddo
enddo
```

Table 12. Algorithm to match a given network with the candidate networks.

The algorithm for matching the network N_2 with $C(N_1)$ can be logically divided into three parts. The first possibility arises when the number of objects in the pair of scenes is the same. In this case, only one of the relation between a pair of blobs has changed. This reduces the problem to an isomorphic match of the network with each of the candidate networks. The second possibility arises when the number of blobs has reduced in number from N_1 to N_2. Such a case arises either when one of the existing pair of blobs have merged into a parent blob or an existing blob has left the scene. In either case, it is reduced to a problem of matching a smaller network with a subnetwork of the candidate networks. Every candidate network is converted into a set of subnetworks by eliminating a node. Each of these subnetworks is matched with the network N_2.

Consider the example pair of N_1 and N_2 networks shown in Figure 61.

N_1 :

$$\begin{array}{cccc} PO & DC & DC & DC \\ & PO & DC & DC \\ & & EC & DC \\ & & & PO \end{array}$$

N_2 :

$$\begin{array}{ccc} PO & DC & DC \\ & EC & DC \\ & & PO \end{array}$$

$P(N_1)$:

$$\begin{array}{cccc} EC,PO,TPP,TPPI & DC,EC & DC,EC & DC,EC \\ & EC,PO,TPP,TPPI & DC,EC & DC,EC \\ & & DC,EC,PO & DC,EC \\ & & & EC,PO,TPP,TPPI \end{array}$$

Figure 61. Example pair of networks where number of nodes decreases.

The number of networks in the candidate set are given by the expression

$$4+2+2+2+4+2+2+3+2+4 = 25$$

The number of nodes decrease from five to four. For each candidate network, we generate a subnetwork of four nodes. For instance, consider a sample candidate network *N* shown in figure 62. The subnetwork *S* obtained by eliminating the vertex 2 (assuming the vertices are numbered from 0) contains TPP in one of the relations. This means that *S* is not a true 4-node network.

$$N: \begin{array}{cccc} TPP & DC & DC & DC \\ & PO & DC & DC \\ & & EC & DC \\ & & & PO \end{array} \qquad S: \begin{array}{ccc} TPP & DC & DC \\ & DC & DC \\ & & PO \end{array}$$

Figure 62. Example pair of networks with conventional meaning of occlusion.

Candidate networks that are not true *n*-node networks are not considered for a match with N_2.

4.5.2 Experimentation

Description of the input: We have captured an actual road traffic scenario as a digital video movie film over a period of 25 minutes at the rate of 25 frames per sec(fps). This video movie is converted into a continuous stream of frames at the rate of 10 frames per second. We observed that the frame splitting rate of 10 is not sufficient to seggregate the input for single change. We regenerated with the full capacity of 25 frames for a second. This stream of frames is logically divided into a set of episodes. We have selected eight episodes at different instants of time during this film.

Experimental Analysis: During the extensive experimentation(Tables 13a-g), for each episode we predict the (*i*+1)th scene from the ith scene. We continue prediction in this pair-wise manner till the end of each episode. Consider for instance, the Table 13a which describes the first episode. This episode contains nearly 250 frames with 21 distinct scenes. The first column contains the frame number with the number of nodes in the second column. The third column indicates whether the number of nodes remain same or not in the pair N_1 and N_2. The last column contains the number of matches computed as a result of the algorithm. The non-zero values in this column mean that the prediction from one row to the next row in the table is successful. The no. of matches value is most of the times more than 1 which means that the predicted result is one among the candidate networks. This value contains the redundant matches with the same candidate network.

Frame No.	**No. of nodes**	**Relation between the no. of nodes**			$P_{next}(P_{dec})$
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
3259	5		eq		*
3262	5			>	19(1)
3268	4		eq		*
3270	4		eq		*
3272	4		eq		*
3273	4			>	5(0)
3304	3		eq		*
3308	3	<			6
3323	4			>	20(2)
3324	3	<			9
3326	4	<			12
3327	5		eq		*
3328	5			>	32(4)
3336	4		eq		*
3360	4		eq		*
3370	4			>	3(0)
3375	3	<			8
3427	4			>	13(1)
3458	3		eq		*
3466	3			>	3(0)
3472	2		eq		*
3507	2				

a. Episode 1 from frames 3259-3507

Frame No.	**No. of nodes**	**Relation between the no. of nodes**			P_{next}
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
3607	3	<			12
3617	4		eq		*
3620	4		eq		*
3624	4	<			12
3628	5		eq		*
3630	5				
3631	5		eq		*
3632	5		eq		*
3633	5		eq	>	64(8)
3648	4		eq		*
3659	4		eq		*
3662	4		eq		*
3666	4		eq		*
3670	4				

b. Episode 2 from frames 3607-3670

Frame No.	No. of nodes	Relation between the no. of nodes			P_{next}
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
3700	3			>	27(12)
3804	2	<			2
3886	3			>	12(2)
3922	2		eq		*
3937	2	<			3
3942	3		eq		*
3950	3		eq		*
3952	3		eq		*
3965	3	<			3
3979	4			>	5(0)
4005	3		eq		*
4013	3				

c. Episode 3 from frames 3700-4013

Frame No.	No. of nodes	Relation between the no. of nodes			P_{next}
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
4251	4		eq		*
4278	4		eq		*
4321	4			>	4(0)
4380	3		eq		*
4385	3			>	3(0)
4407	2				
4417	2		eq		*
4424	3		eq		*
4433	3		eq		*
4436	3	<			6
4444	4				

d. Episode 4 from frames 4251-4444

Frame No.	No. of nodes	Relation between the no. of nodes			P_{next}
		N1< n2	**N1= n2**	**N1> n2**	
5055	3		eq		*
5060	3		eq		
5070	3			>	14(4)
5080	2		eq		*
5090	2		eq		*
5099	2	<			2
5102	3		eq		*
5109	3		eq		*
5112	3			>	14(4)
5139	2	<			2

5165	3				

e. Episode 5 from frames 5055-5165

Frame No.	**No. of nodes**	**Relation between the no. of nodes**			P_{next}
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
5227	4		eq		*
5232	4		eq		*
5236	4			>	20(2)
5248	3		eq		*
5251	3		eq		8
5253	3	<			6
5264	4		eq		*
5278	4		eq		*
5289	4			>	36(0)
5291	3				

f. Episode 6 from frames 5227-5291

Frame No.	**No. of nodes**	**Relation between the no. of nodes**			P_{next}
		$n_1 < n_2$	$n_1 = n_2$	$n_1 > n_2$	
5332	2		eq		*
5335	2	<			3
5340	3		eq		*
5349	3		eq		*
5351	3		eq		*
5362	3		eq		*
5370	3	<			12
5376	4		eq		*
5291	4			>	22(4)
5309	3		eq		*
5313	3		eq		*
5315	3		eq		*
5319	3		eq		*
5324	3		eq		*
5409	4		eq		*
5417	4			>	22(4)
5450	3		eq		*
5500	3	<			3
5543	4		eq		*
5566	4		eq		*
5569	4			>	22(4)
5628	3		eq		*
5650	3				

g. Episode 7 from frames 5332-5650

Table 13. Results for prediction based on spatial model for seven episodes.

When the number of nodes remains same between any given pair of networks, the number of matches are not relevant. In this case, it is sufficient for N_2 to belong to the candidate set as inidicated by "*". We observe during the experimentation for the set of seven episodes, the performance of the proposed method for prediction based on spatial network is 100%. The method does not fail for any of the instance pairs in all the three cases for the number of variables remain same, increase or decrease.

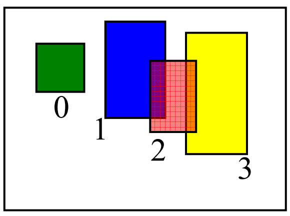
4.6 Instance Based Analysis to distinguish between Occlusion and Exit

We extend the framework proposed in the previous section from prediction to analysis for cases when number of objects reduce in number. Two possibilities arise for this decrease in the number of objects. Firstly, an existing object may leave the scene. Second possibility arises when a pair of blobs merge to form a single blob. From the basic understanding of movement of objects, an object A is said to be occluded by or occludes another object B if the relations between A and B is among the RCC-8 relations – TPP, NTPP, TPPI, NTPPI, EQ. In the event of any of these relations, a pair of blobs reduces to a single one. On the lines of this qualitative description of occlusion, analyze the positive and negative results to propose the possibilities of occlusion and that for an object leaving the scene.
The algorithm in the previous section predicts the set of candidate networks P_{next} that contain the next possible network in the sequence of frames. Each network in this set represents the network N_1 that contains an embedded N_2 network. This set contains P_{dec} the set of networks that contain less number of blobs when arranged.

Consider for instance an example pair of networks in episode 1 and the corresponding scenes as shown below in Figure 63.

N_1:

DC	*DC*	*DC*
	PO	*DC*
		PO



N_2:

DC	*DC*
	EC

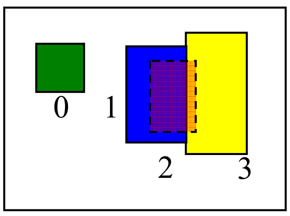


Figure 63. Example occlusion in episode 1 with Pdec = ∅.

For the pair of networks in Figure 63,

n_I = no. of nodes in N_I = 4

n = no. of candidate networks = 2 + 2 + 2 + 4 + 2 + 4 = 16

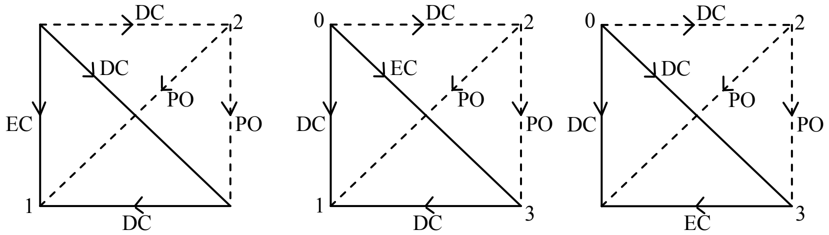
$P_{next} = P_{dec} \cup P_{same}$

$P_{dec} = \emptyset$. This implies that occlusion not due to PO changing to TPP(I)

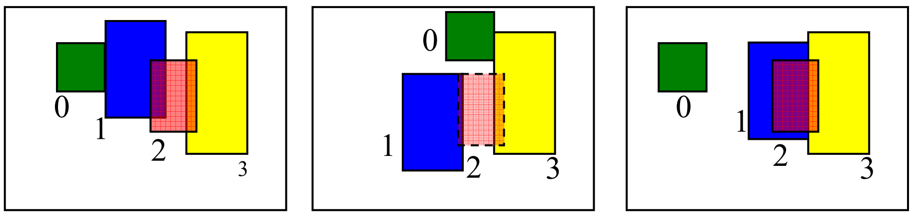
The set P_{same} contains following networks categorized on the vertex removed.

EC	*DC*	*DC*		*DC*	*DC*	*EC*		*DC*	*DC*	*DC*
	PO	*DC*			*PO*	*DC*			*PO*	*EC*
		PO				*PO*				*PO*

a. **Candidate networks with vertex 2 removed.**



b. Graphs for the candidate networks with vertex 2 removed.

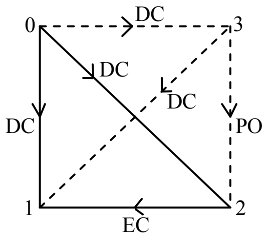


c. Possible scenes.

Figure 64. Example candidate networks with the predicted scenes.

The dotted edges in the network in Figure 64 indicate the three edges that are removed due to vertex 2 being droped. The possible scenes for the three candidate networks after removing the vertex 2 are shown above in Figure 64. Presence of object 2 is shown as dotted box. We observe from the figure that for the object 2 to be occluded, the relation between 1 and 3 cannot remain DC. Thus the right candidate predicted for occlusion of object 2 is the third network since in none of the other two possibilities, the occlusion of 2 is possible.

DC	*DC*	*DC*
	EC	*DC*
		PO



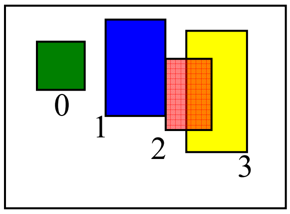
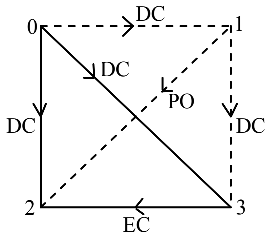


Figure 65. Candidate network, graph and predicted scene with vertex 3 removed

DC	*DC*	*DC*
	PO	*DC*
		EC



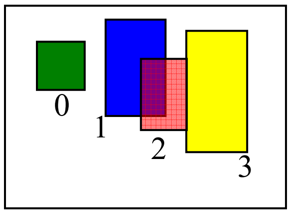


Figure 66. Candidate network, graph and predicted scene with vertex 1 removed

Similarly the scenes that are possible by removing the objects 1 and 3 do not indicate any occlusion as shown in Figures 65 and 66. We observe from the second set of the tables that the reason of occlusion are two fold. Firstly, the P_{dec} set is a null set. This implies that unlike the conventional meaning of occlusion, an event of occlusion takes place due to an object touching another object. Secondly, the P_{dec} set is a non-null set. This falls in line with the conventional meaning of occlusion.

N_1: Frame no.= 3979

DC	*DC*	*PO*
	DC	*DC*
		PO

N_2: Frame no. = 4005

DC	*EC*
	PO

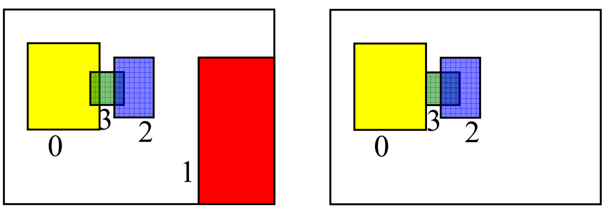


Figure 67. Example networks in episode 3 for a case of an object exit.

Consider another pair of N_1 and N_2 in the episode 3 as shown in figure 67,

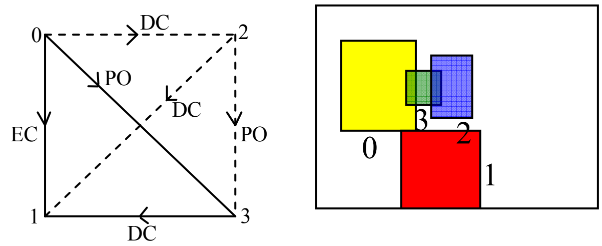
n_1 = no. of nodes in N_1 = 4

n = no. of candidate networks = 2 + 2 + 4 + 2 + 2 + 4 = 16

$P_{dec} = \varnothing$ This implies that occlusion not due to PO changing to TPP(I).

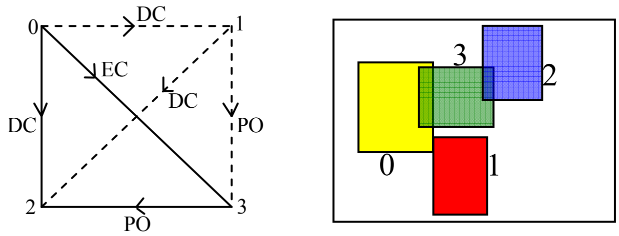
The set P_{next} contains following 5 networks as shown in figure 68.

$$\begin{array}{ccc} EC & DC & PO \\ & DC & DC \\ & & PO \end{array}$$



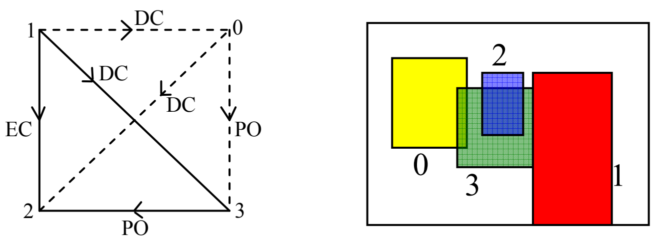
a. Candidate network with vertex 2 removed.

$$\begin{array}{ccc} EC & DC & PO \\ & DC & DC \\ & & PO \end{array}$$



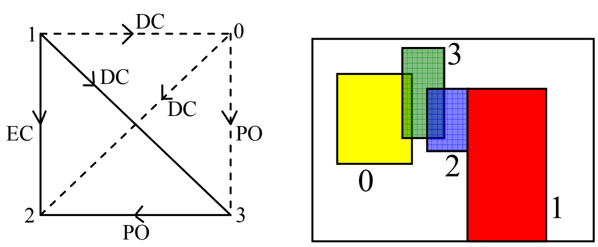
b. Candidate network with vertex 1 removed.

$$\begin{array}{ccc} DC & DC & PO \\ & EC & DC \\ & & PO \end{array}$$



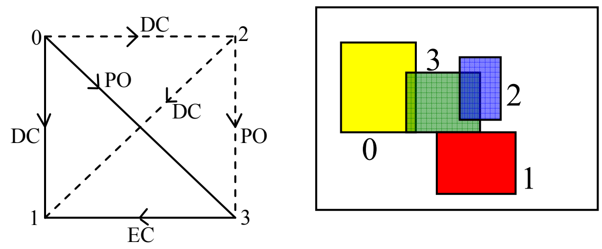
c. Candidate network with vertex 0 removed.

$$\begin{array}{ccc} DC & DC & PO \\ & DC & EC \\ & & PO \end{array}$$



d. Candidate network with vertex 0 removed.

DC	*DC*	*PO*
	DC	*EC*
		PO



e. Candidate network with vertex 2 removed.

Figure 68. Possible scenes for matched candidate networks.

We observe that occlusion is not possible by any of the five possible scenes. Hence it is a case of an object leaving a scene as shown in the figure 68 based on the data collected from the frame sequence.

4.7 Conclusion

The framework proposed in this study is able to correctly predict the next scene in a road traffic scenario. The scenario in this domain is dynamic in the sense that number of vehicles may change over a period of time and relation between any pair of vehicles can change. We are able to extend this framework beyond prediction to detect occlusion. The algorithm proposed here is able to predict the set of next scene in case of an occlusion or an object leaving the scene. We are further able to distinguish between the case of occlusion and an object leaving the scene.

Scene activity in the domain of road traffic can be qualitatively described as RCC-8, IA networks. The neighbourhood property in RCC-8 semi lattice helps to correctly predict next scene. The conventional backtrack search does not help in prediction based on IA networks. The set of completed events help to train weights for incomplete, future disjunctive events. Weighted paradigm based prediction has a limitation of a very narrow search space due to a bias for 2-D relations but helps in partially correct predictions. Assumption of a single change at a time in spatial model is suitable for simplification of prediction in this domain. Occlusion can be qualitatively described in two ways. The conventional meaning is that relation between a pair of blobs changes from PO to TPP(I). The analysis here reveals that occlusion does have another meaning where a pair of blobs touching each other without satisfying the conventional meaning. This study helps us to conclude that a case of an occlusion and an object exit can be qualitatively distinguished based on neighbourhood property of RCC-8 semi lattice.

CHAPTER 5
CONCLUSIONS

In this chapter, we provide the summary of the thesis and discuss the possible extensions as future work.

5.1 Summary

The thesis tries to answer the problem, stated earlier as "How to analyze- the tractability beyond known tractable class to find reasons for hardness of instances in qualitative CSPs and to solve any given instance whether consistent or inconsistent." Consequently, we propose some methods, which individually constitute the answer of the problem.

We study the existing algorithms to understand their strength and weakness. This has motivated us to identify the gaps in the existing research. As a result, we provide a literature survey on the qualitative binary constraint satisfaction problems. In this thesis, our focus has been on the two popular qualitative frameworks namely RCC-8 and interval algebra. We theoretically analyze the reasons of tractability beyond the existing tractable classes in RCC-8. We theoretically analyze the reasons for hardness based on individual instances in interval algebra. We propose a number of complete instantiation strategies ranging from incomplete exact to complete approximate for the two frameworks.

We study the basic properties of partially ordered sets to propose a lattice-theory based representation of RCC-8 relations. This interpretation helps us to identify new tractable classes and to analytically prove the closure, tractability of the known classes. We prove that path consistency preserves the least element in the existing tractable classes. We are able to extend this property to a larger set that contains known tractable classes. The idea is inspired by an existing refinement-based solution technique. During this study, we are able to prove minimality of the convex class of RCC-8. We are able to extend this analysis to coarsening a known class to find new tractable class.

The least element based formalism is extended to the interval algebra framework. The concept does not hold as it is since the least element is not preserved by path consistency for the convex class itself. We introduce new concepts – dimension based convexity and high sets of constraints. A combination of these two concepts helps us to analytically prove tractability of convex and preconvex classes.

This helps us to analytically answer the question “What makes an IA instance easier?” Basic properties of composition table based on high set are studied to give a new description of the known tractable classes. The property of convexity in a given dimension plays a crucial role in analyzing the tractability properties of a class. We show here that the known tractable classes exhibit this property and hence path consistency becomes sufficient condition for existence of a solution. We also establish here that for these classes, we can get a solution by a method called ‘highset instantiation’. We propose an efficient single-step algorithm for constructing a solution for path consistent tractable networks without any computational overhead of enforcing path consistency intermittently.

For interval algebra, we are able to answer the question “What makes an IA instance hard to solve? “ for general IA networks. We extend the existing entropy-based interpretation of the weighted IA network paradigm. We investigate the reasons for the incompleteness of the existing highest weight heuristic. We propose a measure of hardness of problem instances for interval algebra networks. Our investigations reveal that the fluctuations in the entropy of the IA constraints, termed as nasty constraints is the reason for hardness of the instance. The conflict between the highest weight relation on the edge and the most supported relation along the paths is the reason for entropy fluctuations in the IA networks. We show here that the known tractable classes of interval algebra do not exhibit the property of entropy fluctuations. We provide different techniques to relax the nasty constraints so that number of conflicts reduce as the entropy of each constraint stabilizes over the search trajectory. This controls the path that the search trajectory takes resulting into an approximate solution of any instance of general IA networks. This study differs from the existing solution techniques of upper approximation of the IA constraints that adopt a sort of a blind approximation of each constraint without addressing whether any particular relaxation is responsible for hardness. This study identifies the nasty constraints in an intelligent manner with the help of the conflicts. The main contributions of this study on the weighted IA networks are five fold. The novel method of conflicts proposed here lays a foundation for a new direction towards addressing hardness of any qualitative binary constraint satisfaction problem. Secondly, this method provides simple analytical proof for ease and difficulty in computing a solution instance wise. Thirdly, we systematically investigate number of methods to relax the identified nasty constraints that has finally helped us to propose an algorithm that computes an approximate solution on completion. Fourthly, we handle the enormous amount of numerical errors that arise due to each and every operation on a weighted IA constraint. This approach is very handy and can be used by anybody in practice. Lastly, we extend this method to show that the inconsistent IA networks can be handled by this algorithm as routine cases of nasty constraints with trivial conflicts.

We investigate the application of the frameworks proposed for RCC-8 and IA in the domain of road traffic. We propose a framework to predict the subsequent possible scene. The scenario in this domain is dynamic in the sense that number of vehicles may change over a period of time and relation between any pair of vehicles can change. The instantiation strategies for interval algebra are not helpful in the correct prediction. The neighbouhood property of the RCC-8 lattice is helpful in correct prediction based on the spatial networks. The algorithm proposed here is able to predict the set of next scenes. We are further able to qualitatively distinguish between the case of occlusion and an object leaving the scene.

5.2 Future Work

In this thesis, we present a detailed theoretical analysis for RCC-8 and interval algebra frameworks along with a purely qualitative formulation of the well known problem of occlusion detection. There are several interesting future directions, out of which few are mentioned below.

Tractability analysis:

- The least element based instantiation strategies are though exact but not complete. We need to identify some more reason which when combined with existence of least element holds true only for inconsistent RCC-8 networks.
- For computing a solution, the need to enforce path consistency may be completely redundant. This will be possible if we are able to identify some more structural criteria for tractability in addition to the least element existence, path consistency preserving the least element, non-empty highsets, and entropy fluctuation based nasty constraints.
- The property of high sets is studied for convex and preconvex classes. This concept can be combined with split based search strategy for general instances of any framework. This will help to identify sets of relations that may satisfy non-conflicting high sets but are not classes.
- The high set based instantiation can be extended to general instances to propose the extent of approximation required to solve a given instance by exploiting the structure. This will help us to propose $\in$-approximate PTAS for qualitative CSPs.

Prediction of Next Road Traffic Scene:

- The prediction based on temporal networks has failed to predict correctly. The reason is that the strategies used narrow down the search to a single possibility. We need to work out some assumption in the temporal context as well, so that the search space is reduced to a set of candidate temporal networks.
- The set of predictions contains more than one match. This set can be analyzed to study the structure of disjunctive networks with/without occlusion.
- Though I am able to distinguish between an occlusion and exit of an object from the scene, an extensive experimentation is desirable in this direction.
- A full fledged system needs to be integrated that should contain the much assumed functionality of software-based computing the bounding boxes of the vehicles in the scene.

5.3 Concluding remarks

During the years, spent on the work reported in this thesis, I experienced mostly moments of sorrow. Each of the failures forced me to think more and work hard. I will remember the lessons that I have learnt due to these failures throughout my life. Though this is the end of this thesis, but I find it is the beginning of my journey as an independent researcher to contribute and serve my organization – DRDO, known for serving the nation.

****

APPENDIX A

Extensional interpretation triplets in the domain D_T of Triangular Objects

RCC-8 CT contains 193 unique triads. A given triad <*R*, *T*, *S*> denotes that the relation *T* is an entry in the cell specified by the ordered pair <*R*, *S*> in the table. The task is to verify for each triad <*R*, *T*, *S*> in the domain $\boldsymbol{D_T}$ whether or not following extensionality condition is satisfied:

$$(\forall\ a, c \in \boldsymbol{D_T})\ [T(a,c) \rightarrow (\exists\ b \in \boldsymbol{D_T})\ [\ R(a, b) \wedge S(b,c)]] \qquad - (1)$$

For a triad, if this condition is true, we say that this triad is extensional. Some observations about the nature of the possible triads are given below:

a. If either *R* or *S* or *T* is the identity relation EQ, then the triad is extensional, since the condition (1) holds true. There are 22 such triads; we are left with 171 triads.
b. Self-dual triads are the triads for which the inverse triad $<S^{-1}, T^{-1}, R^{-1}>$ is the triad itself. There are 25 self-dual triads, in 8 of these, EQ relation occurs.

Among 171 triads, where EQ does not occur and which are unique, there are (25 – 8) + (171 – (25 – 8))/2 = 94 triads. To show that a triad is extensional, by definition we should show, for two arbitrary triangular objects a, c ∈ $\boldsymbol{D_T}$, related by the RCC-8 relation *T*, there exists another simple region b ∈ $\boldsymbol{D_T}$, such that *R*(a, b) and *S*(b, c). To simplify the analysis, we can fix one of the regions a as a unit triangle. To formulate this result as well as other general observations we need to fix the following notations.

a. Closer side of a with respect to c is denoted as A_{close}
b. Farthest side of a with respect to c is denoted as A_{far}
c. $line(C_{far})$ denotes the line passing through the side C_{far}
d. $line\ (C_{far}+\in)$ denotes the line parallel to the side C_{far}, such that the distance of the line($C_{far}+\in$) from the object a is more than the line (C_{far}).
e. Plane described by the lines of support of a with respect to. c is T_{AC}
f. H_{AC}: A half plane that contains the object c completely and passes through A_{close}. In case A_{close} intersects the interior of c, this denotes the half plane that contains c and passes through closest vertex of a and parallel to C_{close}.
g. *B(p, q, r)* denotes the region b described by the appropriate choice of three points *p, q* and *r*. We choose point *p* outside a, interior of a or boundary of a depending on the relation between a and b.

h. Similarly, we choose point *q* with respect to. c depending on the relation between b and c. The point *r* is chosen and joined with *p* and *q*, such that it either lies inside or outside an object.

i. $Contact_{AC}$ denotes the line segment which common between the boundaries of the regions a and c. This arises in the case of RCC-8 relations where a pair of objects touch each other externally(EC), touch internally(TPP or TPPI).

j. *Interior(a* $\cup$ *c)* denotes the region that is common to both the objects a and c without the boundary points of A and C.

We discuss 94 triads separately according to whether *T* is DC, EC, PO, TPP or NTPP. Cases where *T* is TPPI or NTPPI, are the inverse cases of the cases where *T* is TPP or NTPP respectively.

1. The cases of < -, DC->: There are altogether 17 different pairs of triads where *T* = DC.

   a. <DC, DC, DC>: *p*, *q* and *r* in $H_{AC} \cap H_{CA}$ such that none of the points lie on A_{close}, C_{close} as shown in Figure A.1(a).

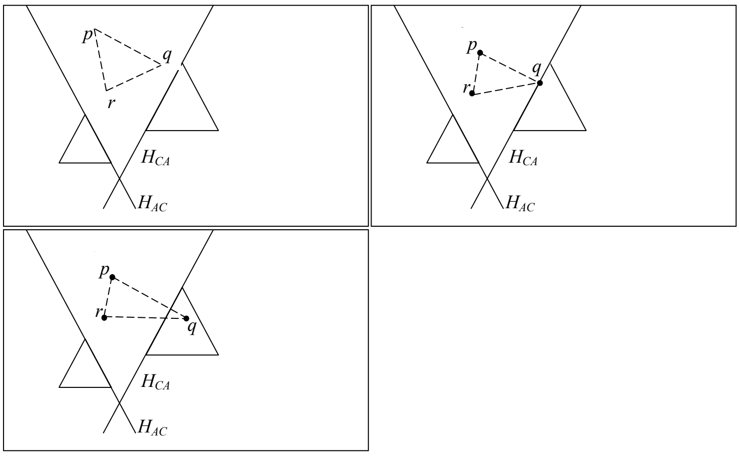
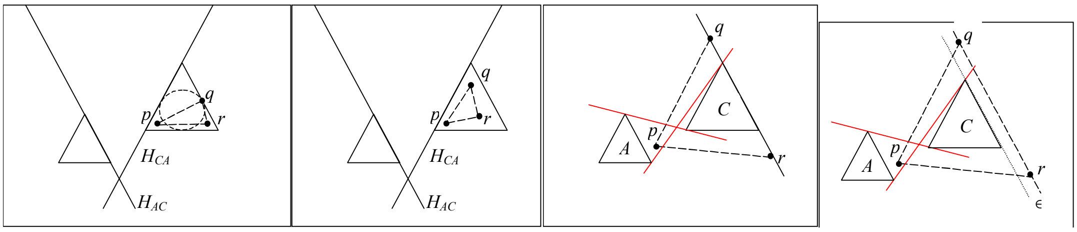


Figure A.1 (a). <DC, DC, DC> (b). <DC, DC, EC> (c). <DC, DC, PO>

   b. **<DC, DC, EC>:** *p*, *q* and *r* in $H_{AC} \cap H_{CA}$ such that $p \notin A_{close}$, C_{close}, $q \in C_{close}$, *r* is chosen such that the line segments *pr* and *qr* do not intersect a and c other than the point *q* as shown in Figure A.1(b).

   c. **<DC, DC, PO>:** *p, r* in $H_{AC} \cap H_{CA}$ such that $p \notin A_{close}$, C_{close}, $q \in$ interior of c, *r* is chosen such that the line segment *pr* does not intersect a, c as shown in Figure A.1(c).

   d. **<DC, DC, TPP>:** *p, q and r* $\in$ c such that at least one of the points lie on a side as shown in Figure A.1(d).



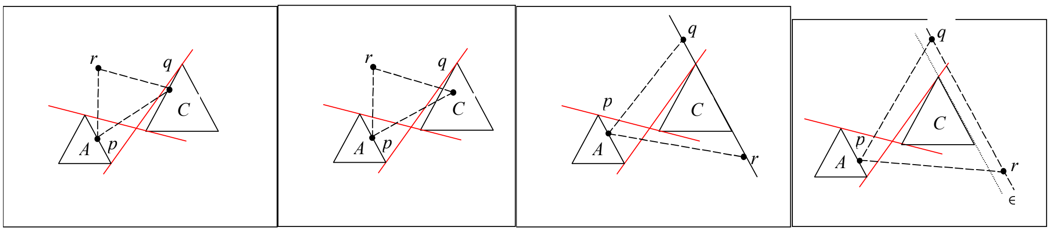
(d). <DC, DC, TPP> (e). <DC, DC, NTPP> (f). <DC, DC, TPPI> (g) <DC,DC,NTPPI>

Figure A.1

e. **<DC, DC, NTPP>:** *p, q and r* c such that none of the points lie on any of the sides as shown in Figure A.1(e).

f. **<DC, DC, TPPI>:** $p \in T_{AC}$, *q and r* $\in$ line(C_{far}) such that none of the line segments pq and pr intersect c as shown in Figure A.1(f).

g. **<DC, DC, NTPPI>:** $p \in T_{AC}$, *q and r* $\in$ line($C_{far}+\in$) such that none of the line segments pq and pr intersect c as shown in Figure A.1(g).



(h). <EC, DC, EC> (i). <EC, DC, PO> (j). <EC, DC, TPPI> (k). <EC, DC, NTPPI>

Figure A.1

h. **<EC, DC, EC**>: $p \in T_{AC}$ and $p \in A_{close}$, $q \in T_{CA}$ and $q \in C_{close}$ and $r \in H_{AC} \cap H_{CA}$ such that none of the line segments pq and pr intersect interior of c as shown in Figure A.1(h).

i. **<EC, DC, PO>:** $p \in T_{AC}$ and $p \in A_{close}$, $q \in$ c and $r \in H_{AC} \cap H_{CA}$ such that the line segment pq does not intersect interior of c as shown in Figure A.1(i).

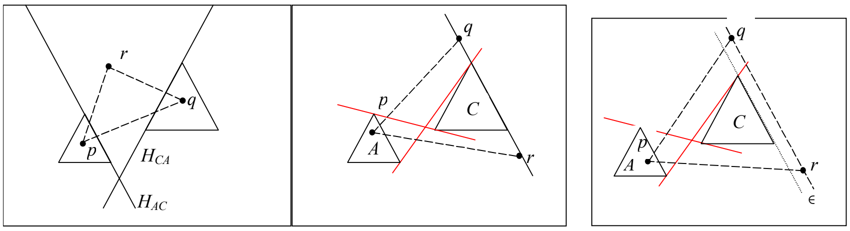
j. **<EC, DC, TPPI>:** $p \in T_{AC}$ and $p \in A_{close}$, *q and r* $\in$ line(C_{far}) such that none of the line segments pq and pr intersect c as shown in Figure A.1(j).

k. **<EC, DC, NTPPI>:** $p \in T_{AC}$ and $p \in A_{close}$ *q and r* $\in$ line($C_{far}+\in$) such that none of the line segments pq and pr intersect c as shown in Figure A.1(k).

l. **<PO, DC, PO>:** p $\in$ interior(a), q $\in$ interior(c) and r $\notin$ a, r$\notin$ c as shown in Figure A.1(l).

m. **<PO, DC, TPPI**>: p $\in$ interior of A, *q* and *r* $\in$ line(C_{far}) such that none of the line segments pq and pr intersect c as shown in Figure A.1(m).

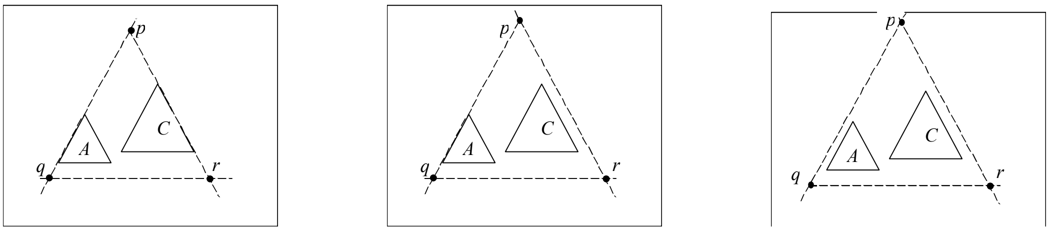
n. **<PO, DC, NTPPI>:** p ∈ interior of a, *q* and *r* ∈ line($C_{far}+\in$) such that none of the line segments pq and pr intersect c as shown in Figure A.1(n).



(l). <PO, DC, PO> (m). <PO, DC, TPPI> (n). <PO, DC, NTPPI>

Figure A.1

o. **<TPP, DC, TPPI>:** *p* is the point of intersection of the two lines – line(A_{far}) and line(C_{far}), *q* ∈ line(A_{far}), *r* ∈ line(C_{far}) such that *qr* ∩ (A∪C) = ∅ and the region b(*p*, *q*, *r*) ⊃ (a ∪ c) as shown in Figure A.1(o).



(o). <TPP, DC, TPPI> (p). <TPP, DC, TPPI> (q). <NTPP, DC, NTPPI>

Figure A.1

p. **<TPP, DC, TPPI>:** *p* is the point of intersection of the two lines – line(A_{far}) and line($C_{far} + \in$), *q* ∈ line(A_{far}), *r* ∈ line($C_{far} + \in$) such that *qr* ∩ (A∪C) = ∅ and the region b(*p*, *q*, *r*) ⊃ (a ∪ c) as shown in Figure A.1(p).

q. **<NTPP, DC, NTPPI>:** *p* is the point of intersection of the two lines – line($A_{far} + \in$) and line($C_{far} + \in$), *q* ∈ line($A_{far} + \in$), *r* ∈ line($C_{far} + \in$) such that *qr* ∩ (a∪c) = ∅ and the region b(*p*, *q*, *r*) ⊃ (a ∪ c) as shown in Figure A.1(q).

2. The cases of <-, EC, ->: There are altogether 15 essentially different pairs of triads where *T* = EC.

   a. **<DC, EC, DC**>: p, q, r ∈ ~(a ∪ c) such that none of the line segments pq, pr, qr intersect (a ∪ c).

   b. **<DC, EC, EC>:** p, r ∈ ~(a ∪ c), q ∈ boundary of c such that none of the line segments pq, pr, qr intersect neither the interior of (a ∪ c) nor the interior of a.

c. **<DC, EC, PO>:** p, r ∈ ~(A ∪ C), q ∈ interior of c such that none of the line segments pq, pr, qr intersect the interior of a.

d. **<DC, EC, TPP>:** p, q ∈ c such that p ∉ (a ∩ c) and q ∈ boundary of c. Choose r in c such that the line segments pq, pr, qr lie completely inside c.

e. **<DC, EC, NTPP>:** p, q ∈ c such that p ∉ (a ∩ c) and q ∉ boundary of c. Choose r in c such that the line segments pq, pr, qr lie completely inside c.

f. **<EC, EC, EC>:** p ∈ A_{close}, q ∈ C_{close}. Choose r in ~(a ∪ c) such that the line segments pq, pr, qr do not intersect with the interior of (a ∪ c).

g. **<EC, EC, PO>:** p ∈ A_{close}, q ∈ interior of c. Choose r in ~(a ∪ c) such that the line segments pq, pr, qr do not intersect with the interior of a.

h. **<EC, EC, TPP>:** p ∈ A_{close}, q ∈ boundary of c, such that p ∈ (a ∩ c) Choose r in c such that the region b(p, q, r) completely lies inside c with at least one of the points lie on the boundary of c.

i. **<EC, EC, TPP^{-1}>:** p, q ∈ line(A_{close}) such that this line does not cross the interior of both a and c. Choose r in ~(a ∪ c) such that the two line segments pr and qr do not intersect c and the region b(p, q, r) ⊃ c.

j. **<PO, EC, PO>:** p ∈ interior of a, q ∈ interior of c and r ∈ ~(a ∪ c).

k. **<PO, EC, TPPI**>: p ∈ interior of a, q, r ∈ ~(a ∪ c) such that the region b(p, q, r) ⊃ c with at least one of the line segments pq, qr contains an edge of c.

l. **<PO, EC, NTPPI>:** p ∈ interior of a, q, r ∈ ~(a ∪ c) such that the region b(p, q, r) ⊃ c with none of the line segments pq, qr contain any edge of c.

m. **< TPP, EC, TPPI>:** p, q ∈ line (A_{far}) such that this line does not cross the interior of c. Choose r such that r ∈ ~(a ∪ c) and the line segments pr, qr do not intersect c.

n. **< TPP, EC, NTPPI>:** p, q ∈ line (A_{far}) such that this line does not cross the interior of c. Choose r such that r ∈ ~(a ∪ c) and the line segments pr, qr do not intersect c.

o. **< NTPP, EC, NTPPI>:** p, q ∈ line (A_{far}+ ∈) such that this line does not cross the interior of c. Choose r such that r ∈ ~(a ∪ c) and the line segments pr, qr do not intersect c.

3. The cases of <-, PO, ->: There are 20 different pairs of triads where *T* = PO.

a. <DC, PO, DC>: p, q, r ∈ ~(a ∪ c) such that none of the line segments pq, qr, pr intersect a or c.

b. <DC, PO, EC>: $p, r \in \sim(a \cup c)$, $q \in$ Boundary of c such that the line segment $pq \cap a = \emptyset$, $pq \cap c = q$, $pr \cap (a \cup c) = \emptyset$, $qr \cap c = q$.

c. <DC, PO, PO>: $p, r \in \sim(a \cup c)$, $q \in$ Interior of c such that the line segment pr, $pq \cap a = \emptyset$.

d. <DC, PO, TPP>: $p, q, r \in c$ such that none of the line segments pq, qr, $pr \cap a$ and at least one of the points p, q, r is a boundary point of c.

e. <DC, PO, NTPP>: $p, q, r \in c$ such that none of the line segments pq, qr, $pr \cap a$ and none of the points p, q, r is a boundary point of c.

f. <EC, PO, EC>: $p \in$ Boundary of a, $q \in$ Boundary of c, $r \in \sim(a \cup c)$ such that the region $b(p, q, r) \cap a = p$ and $b(p, q, r) \cap c = q$.

g. <EC, PO, PO>: $p \in$ Boundary of a, $q \in$ Boundary of c, $r \in \sim(a \cup c)$ such that the region $b(p, q, r) \cap a = p$.

h. <EC, PO, TPP>: $p, q, r \in c$ where $p \in$ Boundary of a such that the region $b(p, q, r) \cap a = p$ and at least one of the points is a boundary point of c.

g. <EC, PO, NTPP>: $p, q, r \in c$ where $p \in$ Boundary of a such that the region $b(p, q, r) \cap a = p$ and none of the points is a boundary point of c.

h. <PO, PO, PO>: $p \in$ Interior of a, $q \in$ Interior of c, $r \in \sim(a \cup c)$

i. <PO, PO, TPP>: $p, q, r \in c$, $p \in$ Interior of a such that at least one of the points is a boundary point of c.

j. <PO, PO, NTPP>: $p, q, r \in c$, $p \in$ Interior of a such that none of the points is a boundary point of c.

k. <PO, PO, TPPI>: $p \in$ Interior of a, $q,r \in$ line(A_{far}) such that this line does not pass through $(a \cup c)$ other than A_{far}, pq, $qr \cap c = \emptyset$.

l. <PO, PO, NTPPI>: $p \in$ Interior of a, $q,r \in$ line($A_{far} + \in$) such that this line does not pass through $(a \cup c)$ other than A_{far}, pq, $qr \cap c = \emptyset$.

m. <TPP, PO, TPPI>: $p, q \in$ lineA_{far} such that this line does not intersect c where r is the farthest vertex of c such that $pr \cap c = r$, $qr \cap c = r$.

n. <NTPP, PO, NTPPI>: $p, q \in$ line($A_{far} + \in$) such that $pq \cap c = \emptyset$, $pq \cap a = \emptyset$. The point r is at a $\in$ distance behind the farthest vertex of c such that pr, $qr \cap c = \emptyset$.

o. <TPPI, PO, TPP>: $p, q \in (a \cup c)$, $p \in$ Boundary of a, $q \in$ Boundary of c. Choose $r \in a \cap c$ such that the region $b(p, q, r) \subset (a \cap c)$.

p. <TPPI, PO, NTPP>: $p, q, r \in (a \cup c)$, $p \in$ Boundary of a, $q \in$ Interior of $(a \cap c)$. Choose $r \in a \cap c$ such that the region $b(p, q, r) \subset (a \cap c)$.

q. <NTPPI, PO, NTPP>: $p, q, r \in$ Interior of $(a \cap c)$ such that the region $b(p, q, r) \subset (a \cap c)$.

4. The cases of <-, TPP, ->: There are 21 essentially different pairs of triads where T = DC.
   a. **<DC, TPP, DC>:** $p, q, r \in \sim(a \cup c)$ such that the region $b(p, q, r) \cap (a \cup c) = \varnothing$.
   b. **<DC, TPP, EC>:** $p, r \in \sim(a \cup c)$, $q \in$ Boundary of c such that the region $b(p, q, r) \cap (a \cup c) = q$.
   c. **<DC, TPP, PO>:** $p, r \in \sim(a \cup c)$, $q \in$ Interior of c such that the line segments pq, $rq \cap a = \varnothing$.
   d. **<DC, TPP, TPP>:** $p, q, r \in c$, $p, q, r \notin a$ such that the region $b(p, q, r) \cap a = \varnothing$ such that at least one of the points p, q is a boundary point of c.
   e. **<DC, TPP, NTPP>:** $p, q, r \in C$, $p, q, r \notin a$ such that the region $b(p, q, r) \cap a = \varnothing$ such that none of the points p, q is a boundary point of c.
   f. **<EC, TPP, EC>:** $r \in \sim(a \cup c)$, choose $p, q \in line(Contact_{AC})$ such that this line does not cross through the interior of a and the region $b(p,q,r) \cap Interior(a \cup c) = \varnothing$.
   g. **<EC, TPP, PO>:** $r \in \sim(a \cup c)$, $q \in$ Interior of c, $p \in$ Boundary of a such that $pq \cap a = p$.
   h. **<EC, TPP, TPP>:** $p \in$ Boundary of a, $q \in$ Boundary of c, $r \in$ Interior of c such that the region $b(p, q, r) \cap a = p$.
   i. **<EC, TPP, NTPP>:** $p \in$ Boundary of a, $q, r \in$ Interior of c, $pq \cap a = p$ such that none of the points p, q, are the boundary points of c.
   j. **<NTPP, TPP, PO>:** $r \in \sim(a \cup c)$ and $p, q \in$ Line $(A_{far} + \in)$ with respect to r such that $p, q \in$ Interior of c and pr, $qr \cap a = \varnothing$.
   k. **<NTPP, TPP, TPPI>:** $r \in \sim(a \cup c)$ and $p, q \in$ Line (C_{far}) with respect to r such that $pr \cap a \cup c = \varnothing$, $qr \cap a \cup c = \varnothing$.
   l. **<NTPP, TPP, NTPPI>:** $r \in \sim(a \cup c)$ and $p, q \in$ Line $(C_{far} + \in)$ with respect to r such that $pr \cap a \cup c = \varnothing$, $qr \cap a \cup c = \varnothing$.
   m. **<TPPI, TPP, TPP>:** $r \in Contact_{AC}$. $p, q \in$ Interior of a such that the region $b(p, q, r) \subset a$.
   n. **<TPPI, TPP, NTPP>:** $p, q, r \in a$ such that the region $(p, q, r) \cap Contact_{AC} = \varnothing$ and at least one of the points p, q, r is a boundary point at a.

o. **<NTPPI, TPP, NTPP>:** p, q, r ∈ a such that the region (p, q, r) ∩ $Contact_{AC}$ = ∅ and none of the points p, q, r is a boundary point at a.

5. The cases of <-, NTPP, ->: There are 21 essentially different pairs of triads where *T* = DC.

a. **<DC, NTPP, DC>:** p, q, r ∈ ~(a ∪ c) such that pq, qr, pr ∩ (a ∪ c) = ∅.

b. **<DC, NTPP, EC>:** p ∈ ~(a ∪ c), q ∈ C_{close} with respect to p. Choose r ∈ ~(a ∪ c) such that pr ∩ (a ∪ c) = ∅, pq ∩ (a ∪ c) = q, qr ∩ (a ∪ c) = q.

c. **<DC, NTPP, PO>:** p ∈ ~(a ∪ c), q ∈ Interior of c such that pq ∩ a = ∅, r ∈ ~(a ∪ c) such that pr ∩ (a ∪ c) = ∅, qr ∩ c = ∅

d. **<DC, NTPP, TPP>:** p, q, r ∈ Incircle(c), such that at least one of these points are the boundary points of c. Triangle(p, q, r) ∩ a =∅.

e. **<DC, NTPP, NTPP>:** p, q, r ∈ Incircle(c), such that none of these points are the boundary points of c. Triangle(p, q, r) ∩ a =∅.

f. **<EC, NTPP, PO>:** q ∈ Interior of c, q ∉ a, p ∈ A_{close} with respect to q, r ∈ ~(a ∪ c) such that qr ∩ a = ∅, pr ∩ a = p.

g. **<EC, NTPP, TPP>:** p ∈ Boundary of a, q, r ∈ C_{close} with respect to p such that pr, pq ∩ a is a subset of the boundary of a and qr ∩ a = ∅, qr ∈ c.

h. **<EC, NTPP, NTPP>:** p ∈ Boundary of a, q, r ∈ line(C_{close} - ∈) with respect to p such that pr, pq ∩ a is a subset of the boundary of a and qr ∩ a = ∅, qr ∈ c.

i. **<PO, NTPP, PO>:** p ∈ Interior of a, q, r ∈ ~(a ∪ c).

j. **<PO, NTPP, TPP>:** p ∈ Interior of a, q ∈ boundary of c, r ∈ c.

k. **<PO, NTPP, NTPP>:** p ∈ Interior of a, q, r ∈ Interior of c.

l. **<TPP, NTPP, PO>:** p, q ∈ line(Boundary of a), choose r ∈ ~(a ∪ c) such that pq, qr do not cross a.

m. **<TPP, NTPP, TPP>:** p, q ∈ line(Boundary of a), choose r ∈ c such that pr, qr do not cross a and at least one of the points p, q lies on the boundary of c.

n. **<TPP, NTPP, NTPP>:** p, q ∈ line(Boundary of a), choose r ∈ c such that pr, qr do not cross a and none of the points p, q lies on the boundary of c.

o. **<NTPP, NTPP, PO>:** Choose r ∈ ~(a ∪ c) and p, q ∈ line(A_{far} + ∈) with respect to r such that pr, qr ∩ a =∅.

p. **<NTPP, NTPP, TPP>:** Choose $r \in c$, $r \notin a$ and p, q $\in$ line($A_{far} + \in$) with respect to r such that pr, qr $\cap$ a = $\emptyset$ and at least one of the points p, q is a boundary point of c.

q. **<NTPP, NTPP, NTPP>:** Choose $r \in c$, $r \notin a$ and p, q $\in$ line($A_{far} + \in$) with respect to r such that pr, qr $\cap$ a = $\emptyset$ and none of the points p, q, r are the boundary points of c.

r. **<NTPP, NTPP, NTPPI>:** $r \in \sim(a \cup c)$, p, q $\in$ line($A_{far} + \in$) with respect to r such that the region b(p, q, r) $\supset$ ($a \cup c$).

s. **<NTPP, NTPP, TPPI>:** $r \in \sim(a \cup c)$, p, q $\in$ line(A_{far}) with respect to r such that the region b(p, q, r) $\supset$ ($a \cup c$).

t. **<TPPI, NTPP, NTPP>:** p, q, r $\in$ a such that at least one of these points is a boundary point in a.

u. **<NTPPI, NTPP, NTPP>:** p, q, r $\in$ a such that none of these points is a boundary point in a.

APPENDIX B

	b	**m**	**o**	**s**	**d**	**f_i**	**eq**	**f**	**d_i**	**s_i**	**o_i**	**m_i**	**b_i**
b	b	b	b	b	b,m,o,s,d	b	b	b,m,o,s,d	b	b	b,m,o,s,d	b,m,o,s,d	**U**
m	b	b	b	m	o,s,d	b	m	o,s,d	b	m	o,s,d	f_i,eq,f	d_i,s_i,o_i,m_i,b_i
o	b	b	b,m,o	o	o,s,d	b,m,o	o	o,s,d	b,m,o,f_i,d_i	o,f_i,d_i	o,s,d,f_i,eq,f,d_i,s_i,o_i	d_i,s_i,o_i	d_i,s_i,o_i,m_i,b_i
s	b	b	b,m,o	s	d	b,m,o	s	d	b,m,o,f_i,d_i	s,eq,s_i	d,f,o_i	m_i	b_i
d	b	b	b,m,o,s,d	d	d	b,m,o,s,d	d	d	**U**	d,f,o_i,m_i,b_i	d,f,o_i,m_i,b_i	b_i	b_i
f_i	b	m	o	o	o,s,d	f_i	f_i	f_i,eq,f	d_i	d_i	o_i,s_i,d_i	d_i,s_i,o_i	d_i,s_i,o_i,m_i,b_i
eq	b	m	o	s	d	f_i	eq	f	d_i	s_i	o_i	m_i	b_i
F	b	m	o,s,d	d	d	f_i,eq,f	f	f	d_i,s_i,o_i,m_i,b_i	o_i,m_i,b_i	o_i,m_i,b_i	b_i	b_i
d_i	b,m,o,f_i,d_i	o,f_i,d_i	o,f_i,d_i	o,f_i,d_i	o,s,d,f_i,eq,f,d_i,s_i,o_i	d_i	d_i	d_i,s_i,o_i	d_i	d_i	d_i,s_i,o_i	d_i,s_i,o_i	d_i,s_i,o_i,m_i,b_i
s_i	b,m,o,f_i,d_i	o,f_i,d_i	o,f_i,d_i	s,eq,s_i	d,f,o_i	d_i	s_i	o_i	d_i	s_i	o_i	m_i	b_i
o_i	b,m,o,f_i,d_i	o,f_i,d_i	o,s,d,f_i,eq,f,d_i,s_i,o_i	d,f,o_i	d,f,o_i	d_i,s_i,o_i	o_i	o_i	d_i,s_i,o_i,m_i,b_i	o_i,m_i,b_i	o_i,m_i,b_i	b_i	b_i
m_i	b,m,o,f_i,d_i	s,eq,s_i	d,f,o_i	d,f,o_i	d,f,o_i	m_i	m_i	m_i	b_i	b_i	b_i	b_i	b_i
b_i	**U**	d,f,o_i,m_i,b_i	d,f,o_i,m_i,b_i	d,f,o_i,m_i,b_i	d,f,o_i,m_i,b_i	b_i	b_i	b_i	b_i	b_i	b_i	b_i	b_i

Table B.1 Interval algebra Composition Table in IA lattice order

	DC	**EC**	**PO**	**TPP**	**NTPP**	**TPPI**	**NTPPI**	**EQ**
DC	**U**	(DC EC PO TPP NTPP)	(DC EC PO TPP NTPP)	(DC EC PO TPP NTPP)	(DC EC PO TPP NTPP)	DC	DC	DC
EC	(DC EC PO TPPI NTPPI)	(DC EC PO TPP TPPI EQ)	(DC EC PO TPP NTPP)	(EC PO TPP NTPP)	(PO TPP NTPP)	(DC EC)	DC	EC
PO	(DC EC PO TPPI NTPPI)	(DC EC PO TPPI NTPPI)	**U**	(PO TPP NTPP)	(PO TPP NTPP)	(DC EC PO TPPI NTPPI)	(DC EC PO TPPI NTPPI)	PO
TPP	DC	(DC EC)	(DC EC PO TPP NTPP)	(TPP NTPP)	NTPP	(DC EC PO TPP TPPI EQ)	(DC EC PO TPPI NTPPI)	TPP
NTPP	DC	DC	(DC EC PO TPP NTPP)	NTPP	NTPP	(DC EC PO TPP NTPP)	**U**	NTPP
TPPI	(DC EC PO TPPI NTPPI)	(EC PO TPPI NTPPI)	(PO TPPI NTPPI)	(PO TPP TPPI EQ)	(PO TPP NTPP)	(TPPI NTPPI)	NTPPI	TPPI
NTPPI	(DC EC PO TPPI NTPPI)	(PO TPPI NTPPI)	(PO TPPI NTPPI)	(PO TPPI NTPPI)	(PO TPP TPPI NTPP NTPPI EQ)	NTPPI	NTPPI	NTPPI
EQ	DC	EC	PO	TPP	NTPP	TPPI	NTPPI	EQ

Table B.2 RCC-8 Composition Table in RCC-8 Semi-lattice order.

APPENDIX C

J_8

1. (DC)	2. (EC)
3. (DC EC)	4. (PO)
5. (DC PO)	6. (EC PO)
7. (DC EC PO)	8. (TPP)
9. (DC TPP)	10. (EC TPP)
11. (DC EC TPP)	12. (PO TPP)
13. (DC PO TPP)	14. (EC PO TPP)
15. (DC EC PO TPP)	16. (NTPP)
17. (DC NTPP)	18. (EC NTPP)
19. (DC EC NTPP)	20. (PO NTPP)
21. (DC PO NTPP)	22. (EC PO NTPP)
23. (DC EC PO NTPP)	24. (TPP NTPP)
25. (DC TPP NTPP)	26. (EC TPP NTPP)
27. (DC EC TPP NTPP)	28. (PO TPP NTPP)
29. (DC PO TPP NTPP)	30. (EC PO TPP NTPP)
31. (DC EC PO TPP NTPP)	32. (TPPI)
33. (DC TPPI)	34. (EC TPPI)
35. (DC EC TPPI)	36. (PO TPPI)
37. (DC PO TPPI)	38. (EC PO TPPI)
39. (DC EC PO TPPI)	40. (DC TPP TPPI)
41. (EC TPP TPPI)	42. (DC EC TPP TPPI)
43. (PO TPP TPPI)	44. (DC PO TPP TPPI)
45. (EC PO TPP TPPI)	46. (DC EC PO TPP TPPI)
47. (DC NTPP TPPI)	48. (EC NTPP TPPI)
49. (DC EC NTPP TPPI)	50. (PO NTPP TPPI)
51. (DC PO NTPP TPPI)	52. (EC PO NTPP TPPI)
53. (DC EC PO NTPP TPPI)	54. (DC TPP NTPP TPPI)
55. (EC TPP NTPP TPPI)	56. (DC EC TPP NTPP TPPI)

57. (PO TPP NTPP TPPI)	58. (DC PO TPP NTPP TPPI)
59. (EC PO TPP NTPP TPPI)	60. (DC EC PO TPP NTPP TPPI)
61. (NTPPI)	62. (DC NTPPI)
63. (EC NTPPI)	64. (DC EC NTPPI)
65. (PO NTPPI)	66. (DC PO NTPPI)
67. (EC PO NTPPI)	68. (DC EC PO NTPPI)
69. (DC TPP NTPPI)	70. (EC TPP NTPPI)
71. (DC EC TPP NTPPI)	72. (PO TPP NTPPI)
73. (DC PO TPP NTPPI)	74. (EC PO TPP NTPPI)
75. (DC EC PO TPP NTPPI)	76. (DC NTPP NTPPI)
77. (EC NTPP NTPPI)	78. (DC EC NTPP NTPPI)
79. (PO NTPP NTPPI)	80. (DC PO NTPP NTPPI)
81. (EC PO NTPP NTPPI)	82. (DC EC PO NTPP NTPPI)
83. (DC TPP NTPP NTPPI)	84. (EC TPP NTPP NTPPI)
85. (DC EC TPP NTPP NTPPI)	86. (PO TPP NTPP NTPPI)
87. (DC PO TPP NTPP NTPPI)	88. (EC PO TPP NTPP NTPPI)
89. (DC EC PO TPP NTPP NTPPI)	90. (TPPI NTPPI)
91. (DC TPPI NTPPI)	92. (EC TPPI NTPPI)
93. (DC EC TPPI NTPPI)	94. (PO TPPI NTPPI)
95. (DC PO TPPI NTPPI)	96. (EC PO TPPI NTPPI)
97. (DC EC PO TPPI NTPPI)	98. (DC TPP TPPI NTPPI)
99. (EC TPP TPPI NTPPI)	100.(DC EC TPP TPPI NTPPI)
101.(PO TPP TPPI NTPPI)	102.(DC PO TPP TPPI NTPPI)
103.(EC PO TPP TPPI NTPPI)	104.(DC EC PO TPP TPPI NTPPI)
105.(DC NTPP TPPI NTPPI)	106.(EC NTPP TPPI NTPPI)
107.(DC EC NTPP TPPI NTPPI)	108.(PO NTPP TPPI NTPPI)
109.(DC PO NTPP TPPI NTPPI)	110.(EC PO NTPP TPPI NTPPI)
111.(DC EC PO NTPP TPPI NTPPI)	112.(DC TPP NTPP TPPI NTPPI)
113.(EC TPP NTPP TPPI NTPPI)	114.(DC EC TPP NTPP TPPI NTPPI)
115.(PO TPP NTPP TPPI NTPPI)	116.(DC PO TPP NTPP TPPI NTPPI)
117.(EC PO TPP NTPP TPPI NTPPI)	118.(DC EC PO TPP NTPP TPPI NTPPI)
119.(EQ)	120.(DC EQ)
121.(EC EQ)	122.(DC EC EQ)

123.(PO EQ)	124.(DC PO EQ)
125.(EC PO EQ)	126.(DC EC PO EQ)
127.(TPP EQ)	128.(DC TPP EQ)
129.(EC TPP EQ)	130.(DC EC TPP EQ)
131.(PO TPP EQ)	132.(DC PO TPP EQ)
133.(EC PO TPP EQ)	134.(DC EC PO TPP EQ)
135.(DC NTPP EQ)	136.(EC NTPP EQ)
137.(DC EC NTPP EQ)	138.(PO NTPP EQ)
139.(DC PO NTPP EQ)	140.(EC PO NTPP EQ)
141.(DC EC PO NTPP EQ)	142.(TPP NTPP EQ)
143.(DC TPP NTPP EQ)	144.(EC TPP NTPP EQ)
145.(DC EC TPP NTPP EQ)	146.(PO TPP NTPP EQ)
147.(DC PO TPP NTPP EQ)	148.(EC PO TPP NTPP EQ)
149.(DC EC PO TPP NTPP EQ)	150.(TPPI EQ)
151.(DC TPPI EQ)	152.(EC TPPI EQ)
153.(DC EC TPPI EQ)	154.(PO TPPI EQ)
155.(DC PO TPPI EQ)	156.(EC PO TPPI EQ)
157.(DC EC PO TPPI EQ)	158.(DC TPP TPPI EQ)
159.(EC TPP TPPI EQ)	160.(DC EC TPP TPPI EQ)
161.(PO TPP TPPI EQ)	162.(DC PO TPP TPPI EQ)
163.(EC PO TPP TPPI EQ)	164.(DC EC PO TPP TPPI EQ)
165.(DC NTPP TPPI EQ)	166.(EC NTPP TPPI EQ)
167.(DC EC NTPP TPPI EQ)	168.(PO NTPP TPPI EQ)
169.(DC PO NTPP TPPI EQ)	170.(EC PO NTPP TPPI EQ)
171.(DC EC PO NTPP TPPI EQ)	172.(DC TPP NTPP TPPI EQ)
173.(EC TPP NTPP TPPI EQ)	174.(DC EC TPP NTPP TPPI EQ)
175.(PO TPP NTPP TPPI EQ)	176.(DC PO TPP NTPP TPPI EQ)
177.(EC PO TPP NTPP TPPI EQ)	178.(DC EC PO TPP NTPP TPPI EQ)
179.(DC NTPPI EQ)	180.(EC NTPPI EQ)
181.(DC EC NTPPI EQ)	182.(PO NTPPI EQ)
183.(DC PO NTPPI EQ)	184.(EC PO NTPPI EQ)
185.(DC EC PO NTPPI EQ)	186.(DC TPP NTPPI EQ)
187.(EC TPP NTPPI EQ)	188.(DC EC TPP NTPPI EQ)

189.(PO TPP NTPPI EQ)	190.(DC PO TPP NTPPI EQ)
191.(EC PO TPP NTPPI EQ)	192.(DC EC PO TPP NTPPI EQ)
193.(DC NTPP NTPPI EQ)	194.(EC NTPP NTPPI EQ)
195.(DC EC NTPP NTPPI EQ)	196.(PO NTPP NTPPI EQ)
197.(DC PO NTPP NTPPI EQ)	198.(EC PO NTPP NTPPI EQ)
199.(DC EC PO NTPP NTPPI EQ)	200.(DC TPP NTPP NTPPI EQ)
201.(EC TPP NTPP NTPPI EQ)	202.(DC EC TPP NTPP NTPPI EQ)
203.(PO TPP NTPP NTPPI EQ)	204.(DC PO TPP NTPP NTPPI EQ)
205.(EC PO TPP NTPP NTPPI EQ)	206.(DC EC PO TPP NTPP NTPPI EQ)
207.(TPPI NTPPI EQ)	208.(DC TPPI NTPPI EQ)
209.(EC TPPI NTPPI EQ)	210.(DC EC TPPI NTPPI EQ)
211.(PO TPPI NTPPI EQ)	212.(DC PO TPPI NTPPI EQ)
213.(EC PO TPPI NTPPI EQ)	214.(DC EC PO TPPI NTPPI EQ)
215.(DC TPP TPPI NTPPI EQ)	216.(EC TPP TPPI NTPPI EQ)
217.(DC EC TPP TPPI NTPPI EQ)	218.(PO TPP TPPI NTPPI EQ)
219.(DC PO TPP TPPI NTPPI EQ)	220.(EC PO TPP TPPI NTPPI EQ)
221.(DC EC PO TPP TPPI NTPPI EQ)	222.(DC NTPP TPPI NTPPI EQ)
223.(EC NTPP TPPI NTPPI EQ)	224.(DC EC NTPP TPPI NTPPI EQ)
225.(PO NTPP TPPI NTPPI EQ)	226.(DC PO NTPP TPPI NTPPI EQ)
227.(EC PO NTPP TPPI NTPPI EQ)	228.(DC EC PO NTPP TPPI NTPPI EQ)
229.(DC TPP NTPP TPPI NTPPI EQ)	230.(EC TPP NTPP TPPI NTPPI EQ)
231.(DC EC TPP NTPP TPPI NTPPI EQ)	232.(PO TPP NTPP TPPI NTPPI EQ)
233.(DC PO TPP NTPP TPPI NTPPI EQ)	234.(EC PO TPP NTPP TPPI NTPPI EQ)
235.(DC EC PO TPP NTPP TPPI NTPPI EQ)	

APPENDIX D
RCC-8 Chain Relations

1. (DC)	2. (EC)
3. (DC EC)	4. (PO)
5. (EC PO)	6. (DC EC PO)
7. (TPP)	8. (PO TPP)
9. (EC PO TPP)	10. (DC EC PO TPP)
11. (NTPP)	12. (TPP NTPP)
13. (PO TPP NTPP)	14. (EC PO TPP NTPP)
15. (DC EC PO TPP NTPP)	16. (TPPI)
17. (PO TPPI)	18. (EC PO TPPI)
19. (DC EC PO TPPI)	20. (NTPPI)
21. (TPPI NTPPI)	22. (PO TPPI NTPPI)
23. (EC PO TPPI NTPPI)	24. (DC EC PO TPPI NTPPI)
25. (EQ)	26. (TPP EQ)
27. (PO TPP EQ)	28. (EC PO TPP EQ)
29. (DC EC PO TPP EQ)	30. (TPPI EQ)
31. (EC PO TPPI EQ)	32. (PO TPPI EQ)
33. (DC EC PO TPPI EQ)	

APPENDIX E
Convex and Preconvex relations in RCC-8

A. Conv$_8$

1. (DC)	2. (EC)
3. (DC EC)	4. (PO)
5. (EC PO)	6. (DC EC PO)
7. (TPP)	8. (PO TPP)
9. (EC PO TPP)	10. (DC EC PO TPP)
11. (NTPP)	12. (TPP NTPP)
13. (PO TPP NTPP)	14. (EC PO TPP NTPP)
15. (DC EC PO TPP NTPP)	16. (TPPI)
17. (PO TPPI)	18. (EC PO TPPI)
19. (DC EC PO TPPI)	20. (NTPPI)
21. (TPPI NTPPI)	22. (PO TPPI NTPPI)
23. (EC PO TPPI NTPPI)	24. (DC EC PO TPPI NTPPI)
25. (EQ)	26. (TPP EQ)
27. (TPP NTPP EQ)	28. (TPPI EQ)
29. (PO TPP TPPI EQ)	30. (EC PO TPP TPPI EQ)
31. (DC EC PO TPP TPPI EQ)	32. (PO TPP NTPP TPPI EQ)
33. (EC PO TPP NTPP TPPI EQ)	34. (DC EC PO TPP NTPP TPPI EQ)
35. (TPPI NTPPI EQ)	36. (PO TPP TPPI NTPPI EQ)
37. (EC PO TPP TPPI NTPPI EQ)	38. (DC EC PO TPP TPPI NTPPI EQ)
39. (PO TPP NTPP TPPI NTPPI EQ)	40. (EC PO TPP NTPP TPPI NTPPI EQ)
41. (DC EC PO TPP NTPP TPPI NTPPI EQ)	

B. PConv$_8$

1. (DC)	2. (EC)
3. (DC EC)	4. (PO)
5. (DC PO)	6. (EC PO)
7. (DC EC PO)	8. (TPP)
9. (PO TPP)	10. (DC PO TPP)
11. (EC PO TPP)	12. (DC EC PO TPP)
13. (NTPP)	14. (PO NTPP)
15. (DC PO NTPP)	16. (EC PO NTPP)
17. (DC EC PO NTPP)	18. (TPP NTPP)
19. (PO TPP NTPP)	20. (DC PO TPP NTPP)
21. (EC PO TPP NTPP)	22. (DC EC PO TPP NTPP)

23. (TPPI)	24. (PO TPPI)
25. (DC PO TPPI)	26. (EC PO TPPI)
27. (DC EC PO TPPI)	28. (PO TPP TPPI)
29. (DC PO TPP TPPI)	30. (EC PO TPP TPPI)
31. (DC EC PO TPP TPPI)	32. (PO NTPP TPPI)
33. (DC PO NTPP TPPI)	34. (EC PO NTPP TPPI)
35. (DC EC PO NTPP TPPI)	36. (PO TPP NTPP TPPI)
37. (DC PO TPP NTPP TPPI)	38. (EC PO TPP NTPP TPPI)
39. (DC EC PO TPP NTPP TPPI)	40. (NTPPI)
41. (PO NTPPI)	42. (DC PO NTPPI)
43. (EC PO NTPPI)	44. (DC EC PO NTPPI)
45. (PO TPP NTPPI)	46. (DC PO TPP NTPPI)
47. (EC PO TPP NTPPI)	48. (DC EC PO TPP NTPPI)
49. (PO NTPP NTPPI)	50. (DC PO NTPP NTPPI)
51. (EC PO NTPP NTPPI)	52. (DC EC PO NTPP NTPPI)
53. (PO TPP NTPP NTPPI)	54. (DC PO TPP NTPP NTPPI)
55. (EC PO TPP NTPP NTPPI)	56. (DC EC PO TPP NTPP NTPPI)
57. (TPPI NTPPI)	58. (PO TPPI NTPPI)
59. (DC PO TPPI NTPPI)	60. (EC PO TPPI NTPPI)
61. (DC EC PO TPPI NTPPI)	62. (PO TPP TPPI NTPPI)
63. (DC PO TPP TPPI NTPPI)	64. (EC PO TPP TPPI NTPPI)
65. (DC EC PO TPP TPPI NTPPI)	66. (PO NTPP TPPI NTPPI)
67. (DC PO NTPP TPPI NTPPI)	68. (EC PO NTPP TPPI NTPPI)
69. (DC EC PO NTPP TPPI NTPPI)	70. (PO TPP NTPP TPPI NTPPI)
71. (DC PO TPP NTPP TPPI NTPPI)	72. (EC PO TPP NTPP TPPI NTPPI)
73. (DC EC PO TPP NTPP TPPI NTPPI)	74. (EQ)
75. (PO EQ)	76. (DC PO EQ)
77. (EC PO EQ)	78. (DC EC PO EQ)
79. (TPP EQ)	80. (PO TPP EQ)
81. (DC PO TPP EQ)	82. (EC PO TPP EQ)
83. (DC EC PO TPP EQ)	84. (NTPP EQ)
85. (PO NTPP EQ)	86. (DC PO NTPP EQ)
87. (EC PO NTPP EQ)	88. (DC EC PO NTPP EQ)
89. (TPP NTPP EQ)	90. (PO TPP NTPP EQ)
91. (DC PO TPP NTPP EQ)	92. (EC PO TPP NTPP EQ)
93. (DC EC PO TPP NTPP EQ)	94. (TPPI EQ)
95. (PO TPPI EQ)	96. (DC PO TPPI EQ)
97. (EC PO TPPI EQ)	98. (DC EC PO TPPI EQ)
99. (PO TPP TPPI EQ)	100.(DC PO TPP TPPI EQ)

101.(EC PO TPP TPPI EQ)	102.(DC EC PO TPP TPPI EQ)
103.(PO NTPP TPPI EQ)	104.(DC PO NTPP TPPI EQ)
105.(EC PO NTPP TPPI EQ)	106.(DC EC PO NTPP TPPI EQ)
107.(PO TPP NTPP TPPI EQ)	108.(DC PO TPP NTPP TPPI EQ)
109.(EC PO TPP NTPP TPPI EQ)	110.(DC EC PO TPP NTPP TPPI EQ)
111.(NTPPI EQ)	112.(PO NTPPI EQ)
113.(DC PO NTPPI EQ)	114.(EC PO NTPPI EQ)
115.(DC EC PO NTPPI EQ)	116.(PO TPP NTPPI EQ)
117.(DC PO TPP NTPPI EQ)	118.(EC PO TPP NTPPI EQ)
119.(DC EC PO TPP NTPPI EQ)	120.(PO NTPP NTPPI EQ)
121.(DC PO NTPP NTPPI EQ)	122.(EC PO NTPP NTPPI EQ)
123.(DC EC PO NTPP NTPPI EQ)	124.(PO TPP NTPP NTPPI EQ)
125.(DC PO TPP NTPP NTPPI EQ)	126.(EC PO TPP NTPP NTPPI EQ)
127.(DC EC PO TPP NTPP NTPPI EQ)	128.(TPPI NTPPI EQ)
129.(PO TPPI NTPPI EQ)	130.(DC PO TPPI NTPPI EQ)
131.(EC PO TPPI NTPPI EQ)	132.(DC EC PO TPPI NTPPI EQ)
133.(PO TPP TPPI NTPPI EQ)	134.(DC PO TPP TPPI NTPPI EQ)
135.(EC PO TPP TPPI NTPPI EQ)	136.(DC EC PO TPP TPPI NTPPI EQ)
137.(PO NTPP TPPI NTPPI EQ)	138.(DC PO NTPP TPPI NTPPI EQ)
139.(EC PO NTPP TPPI NTPPI EQ)	140.(DC EC PO TPPI NTPP NTPPI EQ)
141.(PO TPP NTPP TPPI NTPPI EQ)	142.(DC PO TPP NTPP TPPI NTPPI EQ)
143.(EC PO TPP NTPP TPPI NTPPI EQ)	144.(DC EC PO TPP NTPP TPPI NTPPI EQ)

APPENDIX F

Handling numerical errors due to floating point operations

In this appendix, we report the details of difficulties along with work around for computer-based implementation of floating-point number arithmetic. The floating-point numbers use the IEEE (institute of Electrical and Electronics Engineers) format. We can declare variables as float or double, depending on the needs of the application. The principal differences between the two types are the significance they can represent, the storage they require and their range. The floating-point variables are represented by a mantissa - which contains the value of the number, and an exponent - which contains the order of magnitude of the number. For a better precision, it is advisable to use the double-precision double data type with 8 bytes. The minimum and maximum values one can store in variables are shown below.

Data type	**Minimum value**	**Maximum value**
Float	FLT_MIN = 1.175494351e-38	FLT_MAX = 3.402823466e+38
Double	DBL_MIN = 2.2250738585072014e-308	DBL_MAX=1.7976931348623158e+308

Floating-point decimal values generally do not have an exact binary representation. This is a side effect of how the CPU represents floating point data. For this reason, you may experience some loss of precision, and some floating-point operations may produce unexpected results. Secondly, there is a type mismatch between numbers used (for example, mixing float and double). To resolve this behavior, most programmers either ensure that value is greater or less than what is needed, or they get and use a Binary Coded Decimal (BCD) library that will maintain the precision.

These constants are defined as the smallest positive number x, such that x+1.0 is not equal to 1.0. Because this is a very small number it is advisable that we employ user-defined tolerance for calculations involving very large numbers. For EPSILON, one may use the constants FLT_EPSILON defined for float as 1.192092896e-07F or DBL_EPSILON defined for double as 2.2204460492503131e-016.

A Numerical error results due to discrepancy of the result of a floating-point arithmetic operation as computed by a machine based environment and that expected theoretically due to a mathematical formulation.

There are many possible factors responsible for numerical errors. The two well-known categories of these errors are - underflow and overflow. In this context, we are dealing with very small floating-point numbers. We focus on the underflow numerical errors. The arithmetic operations used are addition, multiplication and division between a pair of floating point numbers. An underflow numerical error resulting due to multiplication of the two numbers, denoted as ab = 0, given that a > 0 and b>0 is said to occur if and only if | exp(a) + exp(b) | > | exp(DBL_MIN)|. Due to underflow condition, result of computation is zero. This happens due to the limitation of the precision of the double data type, which cannot store a number smaller than DBL_MIN. The impact of these incorrect values is two fold.

For a given pair of non-zero floating point numbers a and b such that a > 0 and b>0, if product of two numbers(c) results into underflow, then we assign product an approximate value of min(a, b). This assignment of smaller is an approximation of actual result that should be assigned theoretically. But this ensures that multiplication of two small numbers is not greater than any of the two. This avoids possibility of a zero value as a wrong result.

The vector multiplication is a group of such pair-wise multiplications taken together. A common experience in normalization of the vectors results into the sum total of all the elements in the vector never precisely to 1.0. Such errors propagate more and more during the iterative calculations. There are a number of normalization techniques in literature - min-max scaling, decimal scaling and z-score scaling. We employ a variant of decimal-scaling scheme here but the final bottleneck is the machine representation of the floating-point numbers. Instead of fixing the number of significant digits after the decimal point, we use a greedy method to choose the number of these digits. The algorithm is shown in Table F.1.

```
Normalize_weighted_vector
        Input:          A weighted vector V.
        Output:         Success- the normalized vector V
                        Failure - error condition
        sum = ∑vi
        if ( sum = = 0)
                Return Failure - null vector
        else
                if (sum = = 1.0)
                        Return success - V is a normalized vector.
                else
                        compute the smallest non-zero weight value in V
                        Exp = abs(exponent of smallest non-zero)
                        LoopCount = 0
                        do while( sum > 1.0 ) and (exp <= 306)
```

```
                ProdFactor = 10^(exp+2)
                ∀i Xi = floor(Vi × ProdFactor)
                IntegerSum = ∑Xi
                ∀i tempXi = Xi / IntegerSum
                sum = ∑tempXi
                ∀i Vi = tempXi
                ++exp, ++LoopCount
            enddo
            if ( exp > 306)     and ( sum != 1.0)
                return success - V is approximately normalized
            endif
            if ( sum = = 1.0 )
                return success - V is precisely normalized
            endif
        endif
    endif
```

Table F.1. Normalization of a weighted vector based on a variant of decimal scaling approach.

The basic principle of this algorithm is that we suppress number of significant digits in mantissa portion of value. We decide on number of digits to be allowed depending on power of 10 with which smallest non-zero value is to be multiplied to become more than 1.0. This scales all the values in vector to be large numbers more than 1.0. We suppress remaining digits after this decimal point by selecting nearest smallest real value without decimal portion i.e. floor of a real number. We normalize this scaled vector(X_i) consisting of large values to a normalized vector($tempX_i$). We iterate on this logic to renormalize vector V by increasing exponent value by one. This increased exponent value shifts decimal point to right by one more digit. We continue shifting this decimal point in every iteration(LoopCount) till we reach maximum number of digits i.e. 306 or until the normalization is done correctly (which ever is earlier). The complete operation of decimal point shifting and renormalizing the values takes constant time of execution.

REFERENCES

1. Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of ACM*, 26(11): 832-843.
2. Allen, J.F. and Koomen, J.A. 1983. Planning using a temporal world model. In *Proceedings of the 8th IJCAI*, 741-747, Karlsruhe, West Germany.
3. Almeida, M. J. 1987. Reasoning about the Temporal structure of Narrative. Ph.D. thesis State University of New York at Buffalo Technical Report 87-10, Buffalo, N.Y.
4. Balbiani, P., Condotta, J. F. and Cerro, L. F. D. 1999a. A tractable subclass of the block algebra: constraint propagation and preconvex relations. In P. Barahona and J. Alferes (eds.): *Progress in Artificial Intelligence, 9th PCAI, EPIA '99*, Evora, Portugal, LNCS 1695: 75-89. Springer.
5. Balbiani, P., Condotta, J. F. and Cerro, L. F. D. 1999b. A new tractable subclass of the rectangle algebra. In T. Dean (ed.): *Proceedings of the 16th IJCAI-99*, Vol.1: 442-447. Morgan Kaufmann, Stockholm, Sweden.
6. Bennett, B. 1994a. Spatial reasoning with propositional logic. In J. Doyle, E. Sandewall, and P. Torasso, (eds.): *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference*, pp. 51-62, Bonn, Germany, Morgan Kaufmann.
7. Bennett, B. 1994b. Some observations and puzzles about composing spatial and temporal relations. In R. Rodriguez (ed.): *Proceedings ECAI-94 Workshop on Spatial and Temporal Reasoning*, Amsterdam, The Netherlands.
8. Bennett, B. 1996. An application of qualitative spatial reasoning to GIS. In R.J. Abrahart (ed.): *Proceedings of I st Intl. Conf. on GeoComputation.* Vol. I: 44-47*,* University of Leeds, United Kingdom.
9. Bennett, B., Isli, A. and Cohn, A. G. 1997. When does a composition table provide a complete and tractable proof procedure for a relational constraint language? In *Proceedings of IJCAI-97 workshop on Spatial and Temporal Reasoning*, Nagoya, Japan.
10. Bennett, B. 1998b. Logical representations for automated reasoning about spatial relationships. PhD thesis, University of Leeds, UK.

11. Bennett, B. and Cohn, A.G. 1999. Consistency of topological relations in the presence of convexity constraints. In *Proceedings of IJCAI-99 Workshop on Hot Topics in Temporal and Spatial Reasoning*, Stockholm, Sweden.

12. Bhavani, S. D. and Pujari, A. K.: EVIA-Evidential Interval Algebra and Heuristic backtrack-free algorithm, *Constraints*, 9 (3), 2004, 193.

13. Bittner, T. and Stell, J.G. 2000. Rough sets in approximate spatial Reasoning. In *Proceedings of RSCTC-2000, Lecture Notes in AI*, 2005, pp. 445-453, Berlin-Heidelberg, Springer-Verlag.

14. Cheeseman, P., Kanefsky, B. and Taylor, W.M.1991. Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 331-337, Sydney, Australia, August 1991. Morgan Kaufmann.

15. Cormen, T. H., Leiserson, C.E. and Rivest, R.L 1990. *Introduction to Algorithms.* MIT Press.

16. Cui, Z., Cohn, A. G. and Randell, D. A. 1993. Qualitative and topological relationships in spatial databases. In D. Abel and B. C. Ooi (eds.): *Advances in Spatial Databases*, LNCS 692, pp. 293-315, Springer-Verlag, Berlin.

17. Dechter, R.: *Constraint Processing*. Morgan Kaufmann Publishers, San Francisco, USA(2003).

18. Düntsch, I. 1999a. A tutorial on relation algebras and their application in spatial reasoning. Given at *COSIT-99*. Available via http://www.infj.ulst.ac.uk/cccz23/.

19. Düntsch, I. Wang, H. and McCloskey, S. 1999b. Relation algebras in qualitative spatial reasoning. *Fundamenta Informaticae*, Vol 39 pp. 229-248.

20. Düntsch, I., Schmidt, G. and Winter, M. 2001b. A necessary relation algebra for mereotopology. *Studia Logica* Vol 69, pp. 381-409.

21. Düntsch, I., Wang, H. and McCloskey, S. 2001c. A relation algebraic approach to the region connection calculus. *Theoretical Computer Science*, Vol 255, pp. 229-248.

22. Egenhofer, M.J. 1991. Reasoning about binary topological relations. In O. Gunther and H. J. Schek (eds.): *2nd Symposium, SSD '91: Advances in Spatial databases,* pp. 143-160. Springer-Verlag, New York, USA.

23. Feigenbaum, E.A. and J. Feldman, eds., 1963, Computers and Thought, McGraw-Hill, p.6.

24. Freksa, C. 1991. Conceptual neighborhood and its role in temporal and spatial reasoning. In M. G. Singh and L. T. Massuys (eds.): *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pp. 181-187, North-Holland, Amsterdam. Elsevier.

25. Freuder, E.C. Sythesizing constraint expressions. Communications ACM 21 (1978) 958-966.

26. Freuder, E.C. and Wallace, R. Partial constraint satisfaction. *Artificial Intelligence*, 58(1):21-70, 1992.

27. Gent, I.P. and Walsh, T. 1996. The satisfiability constraint gap. *Artificial Intelligence*, 81(1-2): 59-80, 1996.

28. Gerevini, A. and Renz, J. 1998. Combining topological and qualitative size constraints for spatial reasoning. In M. J. Maher and J.F. Puget (eds.): *Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming*, Pisa, Italy, pp. 26-30, LNCS 1522, Springer.

29. Golumbic, M.C. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

30. Golumbic, M.C and Shamir, R. 1993. Complexity and algorithms for reasoning about time: A graph-theoretic approach. Journal of ACM, 40, 1108-1133.

31. Grigni, M., Papadias, D., Papadimitriou, C. 1995. Topological inference. In C. Mellish (ed.): *Proceedings of the 14th IJCAI* , pp. 901-906. Morgan Kaufmann Publishers, San Mateo, CA, USA.

32. Hirsch, R. A finite relation algebra with undecidable network of satisfaction problem. Bulletin of IGPL, 1999.

33. Hogg, T. Huberman, B.A. and Williams, C.P.1996. Special Volume on frontiers in problem solving: Phase transitions and complexity. *Artificial Intelligence,* 81(1-2), 1996.

34. Jonsson, D.S. A catalog of complexity classes. In J. Leeuwen, editor, *Handbook of Theoretical Computer Science,* Vol A, pages 67-161. MIT Press.

35. Koubarakis, M., Mylopoulos, J., Stanley, M. and Borgida, M. 1989. Telos: Features and Formalization. Knowledge Representation and Reasoning Technical Report KRR-TR-89-4, Department of Computer Science, University of Toronto.

36. Ladkin, P.B. and Maddux, R. 1994. On binary constraint problems. *Journal of the Association for Computing Machinery,* 41(3): 435-469.

37. Ladkin, P.B. and Reinefeld, A. Effective solution of qualitative interval constraint problems. *Artificial Intelligence,* 57(1): 105-124, September 1992.

38. Ladkin, P.B. and Reinefeld, A. A symbolic approach to interval constraint problems. In J. Calmet and J.A. Campbell, editors, *Artificial Intelligence and Symbolic Mathematical Computing,* volume 737 of *Lecture Notes in Computer Science*, pages 65-84. Springer-Verlag, Berlin, Heidelberg, New York, 1993.

39. Ligozat, G. 1994. Tractable relations in temporal reasoning: pre-convex relations. In *Proceedings of the ECAI-94 Workshop W12 on Spatial and Temporal Reasoning*, Amsterdam, The Netherlands.

40. Ligozat, G. 1996. A new proof of tractability for ORD-Horn relations. In *Proceedings of AAAI*, 395-401. AAAI Press.

41. Ligozat, G. 1998. Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, 9(1): 23-44.

42. Ligozat, G., Mitra, D., and Condotta. 2004. J.F. Spatial and Temporal Reasoning: Beyond Allen's Calculus. AI Communications 17(2004), 223-233, IOS Press.

43. Liu, C. L. 1985. *Elements of Discrete Mathematics*. 2nd edition, McGraw-Hill.

44. Lu, R., Sadiq, S., Padmanabhan, V., Governatori, G. Using a Temporal Constraint Network for Business Process Execution. *Proceedings of 17th Australasian Database Conference*, 2006, Hobart, Australia.

45. Mackworth. Consistency in networks of relations. AI, 1977.

46. Mackworth and Freuder. The complexity of some polynomial network consistency algorithms for CSP. AI 25:65-73, 1985.

47. Montanari, U. 1974a. Networks of constraints: fundamental properties and applications to picture processing. Information Sciences, 7:95-132, 1974.

48. Montanari, U. 1974b. Networks of constraints: fundamental properties and applications to picture processing, Inf. Sci. 7(1974) 95-132.

49. Munkres, J. Toplogy (2nd Edition), Prentice Hall, 1999.

50. Nebel, B. and Bürckert, H. 1995a. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM,* 42(1): 43-66.

51. Nebel, B. 1995b. Computational properties of qualitative spatial reasoning: First results. In I. Wachsmuth, C.-R. Rollinger, and W. Brauer (eds.): *Proceedings of KI-95: Advances*

in Artificial Intelligence, volume 981 of Lecture Notes in Computer Science, pp 233-244. Bielefeld, Germany. Springer-Verlag.

52. Nebel, B.1997. Solving hard qualitative temporal reasoning problems: evaluating the efficiency of using the ORD-Horn class. *Constraints*, 3(1): 175-190.
53. Nökel, K. 1989. Convex relations between time intervals. Artificial Intelligence -Tagung, pp. 298-302, Springer-Verlag, Berlin.
54. Papadimitriou, C.H. Computational Complexity. Addison-Wesley, Reading, MA, 1994.
55. Pujari, A. K., Kumari, G.V. and Sattar, A. 1999. INDU- Interval duration network. In N. Foo, (ed.): Proceedings of 16th Australian joint conference on AI, pp. 291-303. SpringerVerlag.
56. Adilakshmi, T. Pujari, A K 2002. A New Heuristics to Solve Temporal Constraint Satisfaction Problem Using Entropy and Convexity. *International Conference on Knowledge Based Computer Systems (KBCS)*, Mumbai, India.
57. Pujari, A. K. and Adilakshmi, T. 2004. A novel Heuristic to solve IA network by Convex Approximations and Weights. *In Proceedings of 8th PRICAI* (Auckland, New Zealand, August, 2004), 154-163.
58. Randell, D.A., Cui, Z. and Cohn, A.G. 1992. A spatial logic based on regions and connection. In B. Nebel, C. Rich, W.R. Swartout (eds.): *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*. pp. 165-176. Cambridge, MA, Morgan Kaufmann.
59. Renz, J. 1998. A canonical model of the region connection calculus. In A.G. Cohn, L. K. Schubert, S.C. Shapiro (eds.), *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 330-341. Trento, Italy, Morgan Kaufmann.
60. Renz, J. and Nebel, B. 1999a. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of RCC. *Artificial Intelligence*, 108(1-2): 69-123.
61. Renz, J. 1999b. Maximal tractable fragments of the region connection calculus: A complete analysis. In T. Dean (ed.), *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 358-363. Stockholm, Sweden, Morgan Kaufmann.
62. Renz, J., and Nebel, B. 2001. Efficient methods for qualitative spatial reasoning. *Journal of Artificial Intelligence Research*, 15: 289-318.

63. Rossi, F., Venable K. B., Khatib, L., Morris, P. and Morris, R.: Two solvers for tractable temporal constraints with preferences. *Proc. AAAI 2002 workshop on preferences in AI and CP Edmonton, Canada.*

64. Sanjiang, Li. and Mingsheng, Y. 2003a. Extensionality of the RCC8 composition table. *Fundamenta Informaticae*, 55: 363-385.

65. Sanjiang, Li. and Mingsheng, Y. 2003b. Region connection calculus: its models and composition. *Artificial Intelligence,* 145(1-2): 121-146.

66. Seidel, R. A new method for solving constraint satisfaction problems. *In Proceedings IJCAI-81* Vancouver, BC (1981) 338-442.

67. Song, F., and Cohen. R. 1988. The Interpretation of Temporal Relations in Narrative. *Proceedings of the Seventh National Conference on Artificial Intelligence*, Saint Paul, Minn., 745-750.

68. Tarski, A. 1941. On the calculus of relations. *Journal of Symbolic Logic*, 6:73-89.

69. Thornton, J., Beaumont, A., Sattar, M., Maher, J. Applying Local Search to Temporal Reasoning. TIME 2002: 94-99.

70. Tsang, E. Foundations of Constraint Satisfaction. Academic Press Limited, 1993.

71. van Beek, P. 1992. Reasoning about qualitative temporal information. *Artificial Intelligence*, Special volume on constraint-based reasoning, 58(1-3): 297-326.

72. van Beek, P., and Manchak, D. W. 1996. The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research*, 4: 1-18.

73. van Beek, P., and Dechter, R. 1995. On the minimality and global consistency of row-convex constraint networks. *Journal of the ACM*, 42(3): 543-561.

74. Vilain, M.B., Kautz, H.A. and van Beek, P. 1989. Constraint propagation algorithms for Temporal reasoning: A revised report, Eds. Readings in Qualitative Reasoning about Physical Systems, pp. 373-381. Morgan Kaufmann, San Mateo, CA.

75. Vilain, M.B. and Kautz, H.A. Constraint propagation algorithms for temporal reasoning. *In Proceedings AAAI-86*, Philadelphia, PA (1986) 132-144.

76. Willard, S. General Topology. Addison-Wesley, 1970.

77. Zitnick III, C. L., 2003. Computing Conditional Probabilities in Large Domains by Maximizing R´enyi's Quadratic Entropy. PhD Thesis: CMU-RI-TR-03-20, Carnegie Mellon University.