CONSTRAINT TIGHTNESS versus GLOBAL CONSISTENCY

Peter van Beek
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada

Rina Dechter
Department of Information and Computer Science
University of California, Irvine

Technical Report 94-48
November 22, 1994

Abstract

Constraint networks are a simple representation and reasoning framework with diverse applications. In this paper, we present a new property called *constraint tightness* that can be used for characterizing the difficulty of problems formulated as constraint networks. Specifically, we show that when the constraints are tight they may require less preprocessing in order to guarantee a backtrack-free solution. This suggests, for example, that many instances of crossword puzzles are relatively easy while scheduling problems involving resource constraints are quite hard. Formally, we present a relationship between the tightness or restrictiveness of the constraints, and the level of local consistency sufficient to ensure global consistency, thus ensuring backtrack-freeness. Two definitions of local consistency are employed. The traditional variable-based notion leads to a condition involving the tightness of the constraints, the level of local consistency, and the arity of the constraints, while a new definition of *relational* consistency leads to a condition expressed in terms of tightness and local-consistency level, alone. New algorithms for enforcing relational consistency are introduced and analyzed.

To be published in KR-94

Constraint Tightness versus Global Consistency

Peter van Beek
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2H1
vanbeek@cs.ualberta.ca

Rina Dechter
Department of Computer and Information Science
University of California, Irvine
Irvine, California, USA 92717
dechter@ics.uci.edu

Abstract

Constraint networks are a simple representation and reasoning framework with diverse applications. In this paper, we present a new property called *constraint tightness* that can be used for characterizing the difficulty of problems formulated as constraint networks. Specifically, we show that when the constraints are tight they may require less preprocessing in order to guarantee a backtrack-free solution. This suggests, for example, that many instances of crossword puzzles are relatively easy while scheduling problems involving resource constraints are quite hard. Formally, we present a relationship between the tightness or restrictiveness of the constraints, and the level of local consistency sufficient to ensure global consistency, thus ensuring backtrack-freeness. Two definitions of local consistency are employed. The traditional variable-based notion leads to a condition involving the tightness of the constraints, the level of local consistency, and the arity of the constraints, while a new definition of *relational* consistency leads to a condition expressed in terms of tightness and local-consistency level, alone. New algorithms for enforcing relational consistency are introduced and analyzed.

1 Introduction

Constraint networks are a simple representation and reasoning framework. A problem is represented as a set of variables, a domain of values for each variable, and a set of constraints between the variables, and the reasoning task is to find an instantiation of the variables that satisfies the constraints. In spite of the simplicity of the framework, many interesting problems can be formulated as constraint networks, including graph coloring [Montanari, 1974], scene labeling [Waltz, 1975], natural language parsing [Maruyama, 1990], and temporal reasoning [Allen, 1983; Dechter *et al.*, 1991; Meiri, 1991; van Beek, 1992].

Constraint networks are often solved using a backtracking algorithm. However, backtracking algorithms are susceptible to "thrashing:" discovering over and over again the same reason for reaching a dead end in the search for a solution. To ameliorate this thrashing behavior, algorithms for preprocessing a constraint network by removing local inconsistencies have been proposed and studied (e.g., [Dechter and Meiri, 1989; Mackworth, 1977; Montanari, 1974]). Sometimes a certain level of local consistency is enough to guarantee that the network is globally consistent. A network is globally consistent if any solution for a subnetwork can always be extended to a solution for the entire network. Hence, if a network is globally consistent, a solution can be found in a backtrack-free manner.

In this paper, we present a relationship between the tightness or restrictiveness of the constraints, the arity of the constraints, and the level of local consistency sufficient to ensure global consistency. Specifically, in any constraint network where the constraints have arity r or less and the constraints have tightness of m or less, if the network is strongly $((m+1)(r-1)+1)$-consistent, then the network is globally consistent. Informally, a network is strongly k-consistent if any consistent instantiation of any $k-1$ or fewer variables can be extended consistently to any additional variable. Also informally, given an r-ary constraint and an instantiation of $r-1$ of the variables that participate in the constraint, the parameter m is an upper bound on the number of instantiations of the rth variable that satisfy the constraint.

We also present a new definition of local consistency called *relational m-consistency*. The virtue of this definition is that, firstly, it allows expressing the relationship between tightness and local consistency in a way that avoids an explicit reference to the arity of the constraints. Secondly, it is operational, thus generalizing the concept of the composition operation defined for binary constraints, and can be incorporated natuother. However, the networks are not 3-consistent. For example, for the confused 4-queens problem shown in Fig. 1a, there is no way to place a queen in the last column that is consistent with the previously placed queens. Similarly the networks are not 4-consistent (see Fig. 1b). Finally, every row and every column of the (0,1)-matrices that define the constraints has at most 3 ones. Hence, the networks are 3-tight.

2.1 Related work

Much work has been done on identifying relationships between properties of constraint networks and the level of local consistency sufficient to ensure global consistency. This work falls into two classes: identifying topological properties of the underlying graph of the network and identifying properties of the constraints. Here we review only the literature for constraint networks with finite domains.

For work that falls into the class of identifying topological properties, Freuder [1982; 1985] identifies a relationship between the *width* of a constraint graph and the level of local consistency needed to ensure a solution can be found without backtracking. As a special case, if the constraint graph is a tree, arc consistency is sufficient to ensure a solution can be found without backtracking. Dechter and Pearl [1988] provide an adaptive scheme where the level of local consistency is adjusted on a node-by-node basis. Dechter and Pearl [1989] generalize the results on trees to hyper-trees which are called acyclic databases in the database community [Beeri *et al.*, 1983].

For work that falls into the class of identifying properties of the constraints (the class into which the present work falls), Montanari [1974] shows that path consistency is sufficient to guarantee that a binary network is globally consistent if the relations are monotone. Van Beek and Dechter [1994] show that path consistency is sufficient if the relations are row convex. Dechter [1992b] identifies a relationship between the size of the domains of the variables, the arity of the constraints, and the level of local consistency sufficient to ensure the network is globally consistent. She proves the following result.

Theorem 1 (Dechter [1992b]) *Any $|D|$-valued r-ary constraint network that is strongly $(|D|(r-1)+1)$-consistent is globally consistent. In particular, any $|D|$-valued binary constraint network that is strongly $(|D|+1)$-consistent is globally consistent.*

For some networks, Dechter's theorem is tight in that the level of local consistency specified by the theorem is really required (graph coloring problems formulated as constraint networks are an example). For other networks, Dechter's theorem overestimates. Our results should be viewed as an improvement on Dechter's theorem. In particular, our main theorem, by taking into account the tightness of the constraints, always specifies a level of strong consistency that is less than or equal to the level of strong consistency required by Dechter's theorem.

3 Binary constraint networks

In this section we restrict our attention to binary constraint networks and present a relationship between the tightness of the constraints and the level of local consistency sufficient to ensure a network is globally consistent. The results are generalized to constraint networks with constraints of arbitrary arity in the next section.

The following lemma is needed in the proof of the main result for constraint networks with binary constraints and in a later proof of the result generalized to constraint networks with constraints of arbitrary arity. The lemma is really about the "tightness" of constraints and the sufficiency of a certain level of consistency. We state the lemma in more colloquial terms to make the proof more understandable.

Lemma 1 *Suppose there are fan clubs that like to meet and talk about famous people, and the following conditions.*

1. *There are n fan clubs and d famous people.*
2. *Each fan club meets and talks about at most m, $m < d$, famous people.*
3. *For every set of $m+1$ or fewer fan clubs, there exists at least one famous person that every club in the set talks about.*

Then, there must exist at least one famous person that every fan club talks about.

Proof. The proof is by contradiction and uses a proof technique discovered by Dechter for Theorem 1. Assume to the contrary that no such famous person exists. Then, for each famous person, f_i, there must exist at least one fan club that does not talk about f_i. Let c_i denote one of the fan clubs that does not talk about f_i. By construction, the set $c = \{c_1, c_2, \ldots, c_d\}$ is a set of fan clubs for which there does not exist a famous person that every club in the set talks about (every candidate f_i is ruled out since c_i does not talk about f_i). For every possible value of m, this leads to a contradiction.

Case 1 ($m = d-1$): The contradiction is immediate as $c = \{c_1, c_2, \ldots, c_d\}$ is a set of fan clubs of size $m+1$ for which there does not exist a famous person that every club in the set talks about. This contradicts condition (3).

Case 2 ($m = d-2$): The nominal size of the set $c = \{c_1, c_2, \ldots, c_d\}$ is $m+2$. We claim, however, that

2. Applying a path consistency algorithm does tighten the constraints between the variables. Once the network is made path consistent, each row has $\leq$ 2 ones. Now the theorem guarantees that if the constraint network is strongly 4-consistent, the network is globally consistent.

3. Applying a 4-consistency algorithm results in no changes as the network is already 4-consistent. Thus, the network is strongly 4-consistent and therefore also globally consistent.

Second, suppose that n is odd. This time, after applying path consistency, the networks are still 3-tight and it can be verified that the networks are not 4-consistent. Enforcing 4-consistency would require non-binary constraints, hence Theorem 2 no longer applies. We take this example up again in the next section where the results are generalized to non-binary constraints. There we show that recording 3-ary constraints is sufficient.

Recall that Nadel [1989] uses confused n-queens problems to empirically compare backtracking algorithms for finding all solutions to constraint networks. Nadel states that these problems provide a "non-trivial test-bed" [1989, p.190]. We believe the above analysis indicates that these problems are quite easy and that any empirical results on these problems should be interpreted in this light. Easy problems potentially make even naive algorithms for solving constraint networks look promising. To avoid this potential pitfall, backtracking algorithms should be tested on problems that range from easy to hard. In general, hard problems are those that require a high level of local consistency to ensure global consistency. Note also that these problems are trivially satisfiable.

Example 3. The graph k-colorability problem can be viewed as a problem on constraint networks: there is a variable for each node in the graph; the domains of the variables are the possible colors, $D = \{1, \ldots, k\}$; and the binary constraints are that two adjacent nodes must be assigned different colors. Graph k-colorability provides examples of networks where both Theorems 1 and 2 give the same bound on the sufficient level of local consistency (since $|D| = k$ and $m = |D| - 1$). Further, as Dechter [1992b] shows, the bound is tight. For example, consider coloring a complete graph on five nodes with four colors. The network is 3-tight and strongly 4-consistent, but not strongly 5-consistent and not globally consistent. Hence, when $m = |D| - 1$, the level of local consistency specified by Theorem 2 is as strong as possible and cannot be lowered.

We can also construct examples to show that Theorem 2 is as strong as possible for all $m < |D| - 1$. This can be done by "embedding" graph coloring constraints into the constraints for the new network. For example, consider the network where the domains are $D = \{1, \ldots, 5\}$ and the constraints between all variables is given by,

$$R_{ij} = \begin{bmatrix} 1\,0\,0\,0\,1 \\ 0\,0\,1\,1\,0 \\ 0\,1\,0\,1\,0 \\ 0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1 \end{bmatrix}.$$

The inner 3×3 matrix is the 3-coloring constraint. The network is 2-tight and strongly 3-consistent, but not strongly 4-consistent and not globally consistent.

4 R-ary constraint networks

In this section we generalize the results of the previous section to networks with constraints of arbitrary arity. We will define m-tightness of r-ary relations, namely relations having r variables. We use the following notations and definitions.

Definition 4 (Relations)
Given a set of variables $X = \{x_1, \ldots, x_n\}$, each associated with a domain of discrete values $D_1, \ldots, D_n$, respectively, a relation *(or, alternatively, a* constraint*) ρ over X is any subset*

$$\rho \subseteq D_1 \times D_2 \times \cdots \times D_n.$$

Given a relation ρ on a set X of variables and a subset $Y \subseteq X$, we denote by $Y = y$ or by y an instantiation of the variables in Y, called a subtuple *and by $\sigma_{Y=y}(\rho)$ the selection of those tuples in ρ that agree with $Y = y$. We denote by $\Pi_Y(\rho)$ the projection of relation ρ on the subset Y. Namely, a tuple over Y appears in $\Pi_Y(\rho)$ if and only if it can be extended to a full tuple in ρ. If Y is not a subset of ρ's variables the projection is over the subset of variables that appear both in Y and in X. The operator $\bowtie$ is the join operator in relational databases.*

Definition 5 (Constraint networks)
A constraint network *R over a set X of variables $\{x_1, x_2, \ldots, x_n\}$, is a set of relations $R_1, \ldots, R_t$, each defined on a subset of variables $S_1, \ldots, S_t$ respectively. A relation in R specified over $Y \subseteq X$ is also denoted R_Y. The set of subsets $S = \{S_1, \ldots, S_t\}$ on which constraints are specified is called the* scheme *of R. The network R represents its set of all consistent solutions over X, denoted $\rho(R)$ or $\rho(X)$, namely,*

$$\rho(R) = \{x = (X_1, \ldots, X_n) \mid \forall S_i \in S, \Pi_{S_i}(x) \in R_i\}.$$

For non-binary networks the notion of *consistency of a subtuple* can be defined in several ways. We will use the following definition. A subtuple over Y is consistent if it satisfies all the constraints defined over Y including all R's constraints obtained by projection over Y.

Definition 6 (Consistency of a subtuple)
A subtuple $Y = y$ is consistent relative to R iff, for all $S_i \in S$,

$$\Pi_{S_i \cap Y}(y) \in \Pi_{S_i \cap Y}(R_i).$$

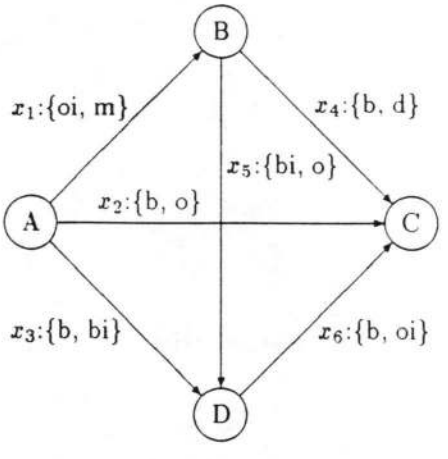

Figure 3: Example temporal network

variables and their associated domains for our example. The ternary constraints for our example are given by,

R_{124} = {(oi,b,b), (oi,o,b), (m,b,b), (m,o,d)},
R_{135} = {(oi,bi,bi), (m,bi,bi), (m,b,o)},
R_{236} = {(b,b,b), (b,b,oi), (b,bi,b), (o,b,oi), (o,bi,b)},
R_{456} = {(b,bi,b), (b,o,b), (d,bi,b), (d,o,oi)}.

It can be shown that the network is 1-tight. Therefore, by Theorem 3, if the network is strongly 5-consistent, then the network is globally consistent. Suppose that we attempt to either verify or achieve this level of strong consistency. The network is strongly 3-consistent, but not 4-consistent. For example, (b,b,oi) is a consistent instantiation of (x_2, x_3, x_6), since it satisfies the constraint R_{236} as well as all the constraints obtained by projection. However, there is no way to extend the instantiation to x_4: (i) $x_4 \leftarrow$ b is inconsistent by the constraint R_{46} obtained by projecting R_{456} on $\{x_4, x_6\}$, and (ii) $x_4 \leftarrow$ d is inconsistent by the constraint R_{24} obtained by projecting R_{124} on $\{x_2, x_4\}$. The modified constraint R'_{236} is given by,

R'_{236} = {(b,b,b), (b,bi,b), (o,b,oi), (o,bi,b)}.

As well, some 3-ary constraints between previously unconstrained triples of variables need to be introduced. For example, (oi,o,oi) is a consistent instantiation of (x_1, x_2, x_6), since it satisfies all the constraints obtained by projection. However, there is no way to extend the instantiation to x_3: (i) $x_3 \leftarrow$ b is inconsistent by the constraint R_{13} obtained by projecting R_{135} on $\{x_1, x_3\}$, and (ii) $x_3 \leftarrow$ bi is inconsistent by the constraint R'_{236}. Once the following 3-ary relations are added, the network is strongly 4-consistent:

R_{126} = {(oi,b,b), (oi,o,b), (m,b,b), (m,o,b), (m,o,oi)},
R_{234} = {(b,b,b), (b,bi,b), (o,b,d), (o,bi,b), (o,bi,d)},
R_{256} = {(b,bi,b), (b,o,b), (o,bi,b), (o,o,oi)},
R_{346} = {(b,b,b), (b,d,oi), (bi,b,b), (bi,d,b)}.

It can now be verified that the network is also strongly 5-consistent. Therefore, by Theorem 3, the network is globally consistent. The network is also minimal. A network of r-ary relations is minimal if each tuple in the relations participates in at least one consistent instantiation of the network. These two properties, global consistency and minimality, ensure that we can efficiently answer some important classes of temporal queries.

4.1 Relational local consistency

In [van Beek and Dechter, 1994] we extended the notion of path-consistency to non-binary relations, and used it to specify an alternative condition under which row-convex non-binary networks of relations are globally consistent. This definition, since it considers the relations rather than the variables as the primitive entities, does not mention the arity of the constraint explicitly. We now extend this definition even further and show how it can be used to alternatively describe Theorem 3.

Definition 8 (Relational m-consistency) *Let R be a network of relations over a set of variables X, let $R_{S_1}, \ldots, R_{S_{m-1}}$ be $m-1$, $m \geq 3$, relations in R, where $S_i \subseteq X$. We say that $R_{S_1}, \ldots, R_{S_{m-1}}$ are relational m-consistent relative to variable x iff any consistent instantiation of the variables in A, where $A = \bigcup_{i=1}^{m-1} S_i - \{x\}$, has an extension to x that satisfies $R_{S_1}, \ldots, R_{S_{m-1}}$ simultaneously. Namely, if and only if*

$$\rho(A) \subseteq \Pi_A(\bowtie_{i=1}^{m-1} R_{S_i}).$$

(Recall that $\rho(A)$ is the set of all consistent instantiations of the variables in A). A set of relations $R_{S_1}, \ldots, R_{S_{m-1}}$ are relational m-consistent iff they are relational m-consistent relative to each variable in $\bigcap_{i=1}^{m-1} S_i$. A network of relations is said to be relational m-consistent iff every set of $m-1$ relations is relational m-consistent. Relational 3-consistency is also called relational path-consistency. A network is strongly relational m-consistent if it is relational i-consistent for every $i \leq m$.

Note that we do not need to define relational 2-consistency since our definition of consistency of a subtuple, which takes into account all the networks' projections, guarantees that any notion of relational 2-consistency is redundant.

Example 7. Consider the following network of relations. The domains of the variables are all $D = \{0, 1, 2\}$ and the relations are given by,

$$(1)\ R_{fxyz} = \{0000, 1000, 0100, 0010, 0001\},$$

$$(2)\ R_{fzs} = \{011, 122, 021\}.$$

The constraints are not relational path-consistent. For example, the instantiation $f = 0, x = 1, y = 0$ satisfies all the constraints, (namely all the projections of

As with variable-based local-consistency, we can improve the efficiency of enforcing relational consistency by enforcing it only along a certain direction. Below we present algorithm *Directional Relational m-Consistency (DRC_m)* that enforces strong relational m-consistency on a network R, relative to a given ordering, d, of the variables $x_1, x_2, \ldots, x_n$. We denote as $DRC_m(R,d)$, a network that is strongly relational m-consistent relative to an ordering d.

$DRC_m(R,d)$

1. *Initialize*: generate an ordered partition of the constraints, $bucket_1, \ldots, bucket_n$, where $bucket_i$ contains all the constraints whose highest variable is x_i.
2. **for** $i \leftarrow n$ **downto** 1
3. **do for** every set of $m-1$ relations $R_{S_1}, \ldots, R_{S_{m-1}}$ in $bucket_i$ (if $bucket_i$ contains fewer than $m-1$ relations, then take all the relations in the bucket).
4. **do** $A \leftarrow \bigcup_{i=1}^{m-1} S_i - \{x_i\}$
5. $R_A \leftarrow R_A \cap \Pi_A(\bowtie_{i=1}^{m-1} R_{S_i})$
6. Add R_A to its appropriate bucket.

While the algorithm is incomplete for deciding consistency in general, it is complete for $(m-2)$-tight relations that are closed under extended m-composition. In fact, it is sufficient to require directional $(m-2)$-tightness relative to the ordering used. Namely, requiring that if x_i appears before x_j in the ordering then any value of x_i will be $(m-2)$-tight relative to x_j but not vice-versa. For example, functional relations are always 1-tight from input to outputs but not for any ordering.

Definition 9 (directionally m-tight)
A binary constraint, R_{ij}, is directionally m-tight *with respect to an ordering of the variables, $d = (x_1, \ldots, x_n)$, if x_i appears before x_j in the ordering and every row of the (0,1)-matrix that defines the constraint has at most m ones. An r-ary relation is directionally m-tight with respect to an ordering of the variables if and only if all of its binary projections are directionally m-tight with respect to the ordering.*

The following theorems will be stated without proofs. Their correctness can be verified using similar theorems on directional consistency algorithms reported earlier [Dechter and Pearl, 1989].

Theorem 6 (Completeness)
If a network $DRC_m(R,d)$ is directionally $(m-2)$-tight relative to d, then $DRC_m(R,d)$ is backtrack-free along d.

Like similar algorithms for imposing directional consistency, DRC_m's worst-case complexity can be bounded as a function of the topological structure of the problem via parameters like the *induced width* of the graph [Dechter and Pearl, 1988].

A network of constraints R can be associated with a constraint graph, where each node is a variable and two variables that appear in one constraint are connected. A general graph can be embedded in a *clique-tree* namely, in a graph whose cliques form a tree-structure. The induced width, $W*$, of such an embedding is its maximal clique size and the induced width $W*$ of an arbitrary graph is the minimum induced width over all its tree-embeddings. For more details see [Dechter and Pearl, 1989]. The complexity of DRC_m can be bounded as a function of the $W*$ of its constraint graph.

Theorem 7 (Complexity) *Given a network of relations R, the complexity of algorithm DRC_m along ordering d is $O(exp(mW^*(d)))$ where $W^*(d)$ is the induced width of the constraint graph of R along d.*

Example 9. Crossword puzzles have been used in experimentally evaluating backtracking algorithms for solving constraint networks [Ginsberg *et al.*, 1990]. We use an example puzzle (taken from [Dechter, 1992a]) to illustrate algorithm DRC_m (see Figure 4).

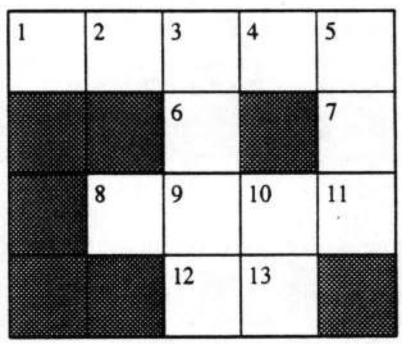

Figure 4: A crossword puzzle

We can formulate this problem as a constraint problem as follows, each possible slot holding a character will be a variable, and the possible words are relations over the variables. Therefore, we have $x_1, \ldots, x_{13}$ variables as marked in the figure. Their domains are the alphabet letters and the constraints are the following relations:

$R_{1,2,3,4,5}$ = {(H,O,S,E,S), (L,A,S,E,R), (S,H,E,E,T), (S,N,A,I,L), (S,T,E,E,R)}

$R_{3,6,9,12}$ = {(H,I,K,E), (A,R,O,N), (K,E,E,T), (E,A,R,N), (S,A,M,E)}

$R_{8,9,10,11} = R_{3,6,9,12}$

$R_{5,7,11}$ = {(R,U,N), (S,U,N), (L,E,T), (Y,E,S), (E,A,T), (T,E,N)}

$R_{10,13}$ = {(N,O), (B,E), (U,S), (I,T)}

$R_{12,13} = R_{10,13}$

Acknowledgements

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, by the NSF under grant IRI-9157636, by the Air Force Office of Scientific Research under grant AFOSR 900136, by Toshiba of America, and by a Xerox grant.

References

[Allen, 1983] J. F. Allen. Maintaining knowledge about temporal intervals. *Comm. ACM*, 26:832–843, 1983.

[Beeri *et al.*, 1983] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30:479–513, 1983.

[Dechter and Meiri, 1989] R. Dechter and I. Meiri. Experimental evaluation of preprocessing techniques in constraint satisfaction problems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 271–277, Detroit, Mich., 1989.

[Dechter and Pearl, 1988] R. Dechter and J. Pearl. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence*, 34:1–38, 1988.

[Dechter and Pearl, 1989] R. Dechter and J. Pearl. Tree clustering for constraint networks. *Artificial Intelligence*, 38:353–366, 1989.

[Dechter and Rish, 1994] R. Dechter and I. Rish. Directional resolution: The Davis-Putnam procedure, revisited. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning*, Bonn, Germany, 1994.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[Dechter, 1992a] R. Dechter. Constraint networks. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence, Second Edition*, pages 276–285. John Wiley & Sons, 1992.

[Dechter, 1992b] R. Dechter. From local to global consistency. *Artificial Intelligence*, 55:87–107, 1992.

[Freuder, 1978] E. C. Freuder. Synthesizing constraint expressions. *Comm. ACM*, 21:958–966, 1978.

[Freuder, 1982] E. C. Freuder. A sufficient condition for backtrack-free search. *J. ACM*, 29:24–32, 1982.

[Freuder, 1985] E. C. Freuder. A sufficient condition for backtrack-bounded search. *J. ACM*, 32:755–761, 1985.

[Ginsberg *et al.*, 1990] M. L. Ginsberg, M. Frank, M. P. Halpin, and M. C. Torrance. Search lessons learned from crossword puzzles. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 210–215, Boston, Mass., 1990.

[Kondrak, 1993] G. Kondrak, 1993. Personal Communication.

[Mackworth, 1977] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[Maruyama, 1990] H. Maruyama. Structural disambiguation with constraint propagation. In *Proceedings of the 28th Conference of the Association for Computational Linguistics*, pages 31–38, Pittsburgh, Pennsylvania, 1990.

[Meiri, 1991] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 260–267, Anaheim, Calif., 1991. An expanded version is available as: Department of Computer Science Technical Report R-160, University of California, Los Angeles.

[Montanari, 1974] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Inform. Sci.*, 7:95–132, 1974.

[Nadel, 1989] B. A. Nadel. Constraint satisfaction algorithms. *Computational Intelligence*, 5:188–224, 1989.

[van Beek and Dechter, 1994]
P. van Beek and R. Dechter. On the minimality and decomposability of row-convex constraint networks. *Accepted for publication in J. ACM*, 1994.

[van Beek, 1992] P. van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 58:297–326, 1992.

[Waltz, 1975] D. Waltz. Understanding line drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975.