From Local to Global Consistency in Temporal Constraint Networks ⋆

Manolis Koubarakis

Dept. of Computation
UMIST
P.O. Box 88
Manchester M60 1QD
U.K.
manolis@sna.co.umist.ac.uk

Abstract

We study the problem of global consistency for several classes of quantitative temporal constraints which include inequalities, inequations and disjunctions of inequations. In all cases that we consider we identify the level of local consistency that is necessary and sufficient for achieving global consistency and present an algorithm which achieves this level. As a byproduct of our analysis, we also develop an interesting minimal network algorithm.

1 Introduction

One of the most important notions found in the constraint satisfaction literature is global consistency [Fre78]. In a *globally consistent* constraint set all interesting constraints are explicitly represented and the projection of the solution set on any subset of the variables can be computed by simply collecting the constraints involving these variables. An important consequence of this property is that a solution can be found by *backtrack-free* search [Fre82]. Enforcing global consistency can take an exponential amount of time in the worst case [Fre78,Coo90]. As a result it is very important to identify cases in which *local consistency*, which presumably can be enforced in polynomial time, implies global consistency [Dec92].

⋆ Invited submission to the special issue of Theoretical Computer Science dedicated to the 1st International Conference on Principles and Practice of Constraint Programming (CP95), Editors: U. Montanari and F. Rossi. Forthcoming.

In this paper we study the problem of enforcing global consistency for sets of quantitative temporal constraints over the rational (or real) numbers. The class of constraints that we consider includes:

- equalities of the form $x - y = r$,
- inequalities of the form $x - y \leq r$,
- inequations of the form $x - y \neq r$, and
- disjunctions of inequations of the form

$$x_1 - y_1 \neq r_1 \ \vee \ \cdots \ \vee \ x_n - y_n \neq r_n$$

where $x, y, x_1, y_1, \ldots, x_n, y_n$ are variables ranging over the rational numbers and $r, r_1, \ldots, r_n$ are rational constants. For the representation of equalities, inequalities and inequations, we utilize *binary temporal constraint networks*. Disjunctions of inequations are represented separately.

Disjunctions of inequations have been introduced in [Kou92] following the observation that in the process of eliminating variables from a set of temporal constraints, an inequation can give rise to a disjunction of inequations.[1] In related temporal reasoning research [VK86,vB90a,GS93,GSS93] have considered inequations of the form $t_1 \neq t_2$ in the context of point algebra (PA) networks. Also, [Mei91a] has studied inequations of the form $t \neq r$ (r a real constant) in the context of point networks with *almost-single-interval domains*. In a more general context, researchers in constraint logic programming (originally [LM89] and later [IvH93,Imb93,Imb94]) have studied disjunctions of arbitrary linear inequations (e.g., $2x_1 + 3x_2 - 4x_3 \neq 4 \ \vee \ x_2 + x_3 + x_5 \neq 7$). [LM89,IvH93] concentrate on deciding consistency and computing canonical forms while [Imb93,Imb94] deal mostly with variable elimination. It is interesting to notice that the basic algorithm for variable elimination in this case has been discovered independently in [Kou92] and [Imb93] although [Kou92] has used the result only in the context of temporal constraints.

The contributions of this paper can be summarized as follows.

(i) We show that strong 5-consistency is necessary and sufficient for achieving global consistency in temporal constraint networks for inequalities and inequations (Corollary 3.1).[2] This result (and all subsequent ones) rely heavily on an observation of [LM89,Kou92,Imb93]: *(disjunctions of) inequations can be treated* independently *of one another for the purposes of deciding consistency or performing variable elimination.*

We give an algorithm which achieves global consistency in $O(Hn^4)$

[1] Elimination of variables is a very important operation in temporal constraint databases [Kou94c,Kou94a,Kou94b].

[2] As shown in [DMP91] if only inequalities are considered path consistency is necessary and sufficient for achieving global consistency.

where n is the number of nodes in the network and H is the number of inequations (Theorems 3.1 and 3.2). The analysis of this algorithm demonstrates that there are situations where it is *impossible* to enforce global consistency without introducing disjunctions of inequations.

A detailed analysis of the global consistency algorithm also gives us an algorithm for computing the minimal temporal constraint network in this case. The complexity of this algorithm is $O(\max(Hn^2, n^3))$ (Theorem 4.1).

(ii) We also consider global consistency of point algebra networks [VK86]. In this case strong 5-consistency is also necessary and sufficient for achieving global consistency (Theorem 5.2). This result, which answers an open problem of [vB90a], also follows from [Kou92] but the bounds of the algorithms given there were not the tightest possible.

(iii) Finally we consider global consistency when disjunctions of inequations are also allowed in the given constraint set. This case is mostly of theoretical interest and is presented here for completeness. In this case, strong $(2V+1)$-consistency is necessary and sufficient for achieving global consistency (Corollary 6.1). The parameter V is the maximum number of variables in any disjunction of inequations.

Most of the above results come from the author's Ph.D. thesis [Kou94c] or are refinements of ideas presented there.

The paper is organized as follows. The next section presents definitions and preliminaries. Section 3 discusses global consistency of temporal constraint networks while Section 4 presents an algorithm for computing the minimal network. Section 5 considers the case of point algebra networks. Section 6 considers the case of arbitrary temporal constraints. Finally Section 7 summarizes our results. Appendix A contains two long proofs.

2 Definitions and Preliminaries

We consider time to be linear, dense and unbounded. *Points* will be our only time entities. Points are identified with the rational numbers but our results still hold if points are identified with the reals. The set of rational numbers will be denoted by $\mathcal{Q}$.

Definition 2.1 A *temporal constraint* is a formula $t-t' \leq r,\ t-t' < r,\ t-t' = r$ or $t_1 - t'_1 \neq r_1 \ \vee \ \cdots \ \vee \ t_n - t'_n \neq r_n$ where $t, t', t_1, \cdots, t_n, t'_1, \cdots, t'_n$ are variables and $r, r_1, \cdots, r_n$ are rational constants.

The rationale for studying disjunctions of inequations has been given in [Kou92].

Definition 2.2 Let C be a set of temporal constraints in variables $t_1, \ldots, t_n$. The *solution set* of C, denoted by $Sol(C)$, is:

$$\{(\tau_1, \ldots, \tau_n) : \ (\tau_1, \ldots, \tau_n) \in \mathcal{Q}^n \text{ and for every } c \in C, \ (\tau_1, \ldots, \tau_n) \text{ satisfies } c\}$$

Each member of $Sol(C)$ is called a *solution* of C. A set of temporal constraints is called *consistent* if and only if its solution set is nonempty.

If c is a disjunction of inequations then $\overline{c}$ denotes the *complement* of c i.e., the conjunction of equations obtained by negating c. If C is a set of equalities in n variables, the solution set of C is an affine subset of $\mathcal{Q}^n$. If C is a set of inequalities in n variables, the solution set of C is a convex polyhedron in $\mathcal{Q}^n$. If C is a set of disjunctions of inequations, the solution set of C is $\mathcal{Q}^n \setminus Sol(\{\overline{c} : \ c \in C\})$. The interested reader can find background material on affine spaces and convex polyhedra in [Sch86].

Let C be a set of temporal constraints in variables $x_1, \ldots, x_n$ which contains only equations, inequalities and inequations (but not disjunctions of inequations). The *temporal constraint network* (TCN) associated with C is a labeled directed graph $G = (V, E)$ where $V = \{1, \ldots, n\}$. Node i represents variable x_i and edge (i, j) represents the binary constraints involving x_i and x_j. As usual unary constraints will be represented as binary constraints with the introduction of a special variable $x_0 = 0$. The set of constraints associated with a TCN N will be denoted by $Constraints(N)$.

Definition 2.3 Let I be a set of rational numbers. I will be called an *almost convex interval* if it is of the form

$$[l, r_1) \cup (r_1, r_2) \cup \cdots \cup (r_{k-1}, r_k) \cup (r_k, u]$$

where $l, r_1, \ldots, r_{k-1}, r_k, u$ are rational numbers such that $l < r_1 < \cdots < r_{k-1} < r_k < u$, and $k \geq 0$. An almost convex interval is also allowed to be open from the right or left.

The k values $r_1, \ldots, r_k$ will be called the *"holes"* of interval I. We define a function *holes* such that, for each almost convex interval I as above,

$$holes(I) = \{r_1, \ldots, r_k\}.$$

Let us assume that the set of constraints c_{ij} on $x_j - x_i$ is

$$\{x_j - x_i \leq d_{ij}, \ x_j - x_i \geq -d_{ji}, \ x_j - x_i \neq r_{ji}^1, \ldots, x_j - x_i \neq r_{ji}^{h_{ji}}\}$$

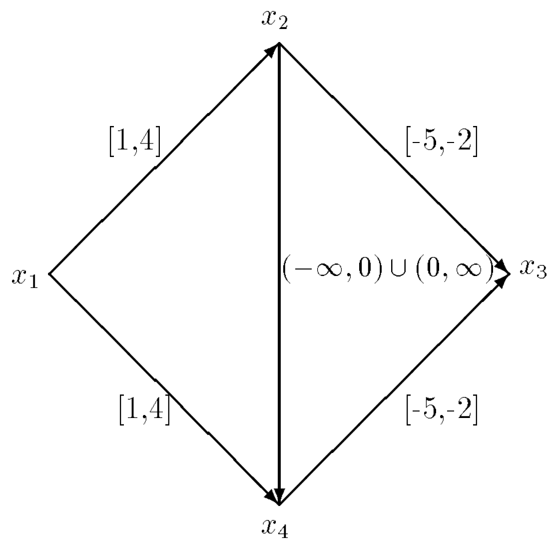

Fig. 1. A temporal constraint network

where $-d_{ji} < r_{ji}^1 < \cdots < r_{ji}^{h_{ji}} < d_{ij}$. Then the corresponding TCN N will have an edge $i \to j$ labeled by the almost-convex interval

$$N_{ij} = [-d_{ji}, r_{ji}^1) \cup (r_{ji}^1, r_{ji}^2) \cup \cdots \cup (r_{ji}^{h_{ji}-1}, r_{ji}^{h_{ji}}) \cup (r_{ji}^{h_{ji}}, d_{ij}].$$

Example 2.1 The TCN of Figure 1 represents the constraints

$$1 \le x_2 - x_1 \le 4,\ 2 \le x_2 - x_3 \le 5,\ 1 \le x_4 - x_1 \le 4,$$
$$2 \le x_4 - x_3 \le 5,\ x_4 - x_2 \ne 0.$$

Given an interval I, $conv(I)$ will denote the *convex hull* of I i.e., the minimal (in the set-theoretic sense) convex interval which includes I. Formally,

$$conv([l, r_1) \cup (r_1, r_2) \cup \cdots \cup (r_{k-1}, r_k) \cup (r_k, u]) = [l, u]$$

and $conv(I) = I$ if I is convex. If N is a TCN then $conv(N)$ denotes the TCN which is obtained from N by substituting each interval N_{ij} by $conv(N_{ij})$.

If N is a TCN then its *solution set* is $Sol(N) = Sol(Constraints(N))$. A TCN is called *consistent* iff its solution set is nonempty. Two TCN are called *equivalent* iff their solution sets are equal [DMP91,Mei91b].

For the case of TCN, the operations of composition and intersection of almost-convex intervals are defined as usual [Mei91b].

Definition 2.4 Let I_1, I_2 be almost convex intervals. The *composition* of I_1 and I_2, denoted by $I_1 \otimes I_2$, is defined as follows:

$$I_1 \otimes I_2 = \{z :\ \exists x \in I_1,\ \exists y \in I_2 \text{ and } x + y = z\}.$$

The intersection operation $\oplus$ has the usual set-theoretic semantics.

The following proposition is straightforward.

Proposition 1 *The class of almost-convex intervals over $\mathcal{Q}$ is closed under composition and intersection.*

3 Global Consistency of a TCN

We will first consider enforcing global consistency in a TCN.

Notation 3.1 Let C be a set of constraints in variables $x_1, \ldots, x_n$. For any i such that $1 \leq i \leq n$, $C(x_1, \ldots, x_i)$ will denote the set of constraints in C involving only variables $x_1, \ldots, x_i$.

The following definition is from [Dec92].

Definition 3.1 Let C be a set of constraints in variables $x_1, \ldots, x_n$ and $1 \leq i \leq n$. C is called *i-consistent* iff for every $i-1$ distinct variables $x_1, \ldots, x_{i-1}$, every valuation $u = \{x_1 \leftarrow x_1^0, \ldots, x_{i-1} \leftarrow x_{i-1}^0\}$ such that u satisfies the constraints $C(x_1, \ldots, x_{i-1})$ and every variable x_i different from $x_1, \ldots, x_{i-1}$, there exists a rational number x_i^0 such that u can be extended to a valuation $u' = u \cup \{x_i \leftarrow x_i^0\}$ which satisfies the constraints $C(x_1, \ldots, x_{i-1}, x_i)$. C is called *strong i-consistent* if it is j-consistent for every j, $1 \leq j \leq i$. C is called *globally consistent* iff it is i-consistent for every i, $1 \leq i \leq n$.

Let us present some examples illustrating the above definitions.

Example 3.1 The constraint set $C = \{x_2 - x_1 \leq 5,\ x_1 - x_3 \leq 2,\ x_5 - x_4 \leq 1,\ x_4 - x_6 \leq 3\}$ is 1- and 2-consistent but not 3-consistent. For example, the valuation $v = \{x_2 \leftarrow 10,\ x_3 \leftarrow 2\}$ satisfies $C(x_2, x_3) = \emptyset$ but it cannot be extended to a valuation which satisfies C.

We can enforce 3-consistency by adding the constraints $x_2 - x_3 \leq 7$ and $x_5 - x_6 \leq 4$ to C. The resulting set is 3-consistent and also globally consistent.

Example 3.2 The constraint set $C = \{x_2 - x_1 = 5,\ x_1 - x_4 \neq 1\}$ is 1- and 2-consistent but not 3-consistent. For example, the valuation $v = \{x_2 \leftarrow 6,\ x_4 \leftarrow 0\}$ satisfies $C(x_2, x_4) = \emptyset$ but it cannot be extended to a valuation which satisfies C.

We can enforce 3-consistency by adding the constraint $x_2 - x_4 \neq 6$ to C. The resulting set is 3-consistent and also globally consistent.

Example 3.3 The constraint set $C = \{x_2 - x_1 \leq 5,\ x_1 - x_3 \leq 2,\ x_2 - x_3 \leq 7,\ x_1 - x_4 \neq 1\}$ is strong 3-consistent but not 4-consistent. For example, the valuation $v = \{x_2 \leftarrow 7,\ x_3 \leftarrow 0,\ x_4 \leftarrow 1\}$ satisfies $C(x_2, x_3, x_4) = \{x_2 - x_3 \leq 7\}$ but it cannot be extended to a valuation which satisfies C.

Enforcing 4-consistency amounts to adding the *disjunction*

$$x_2 - x_4 \neq 6 \ \vee \ x_3 - x_4 \neq -1.$$

The resulting set is 4-consistent and also globally consistent.

Example 3.4 The constraint set $C = \{x_2 - x_1 \leq 5,\ x_1 - x_3 \leq 2,\ x_2 - x_3 \leq 7,\ x_5 - x_4 \leq 1,\ x_4 - x_6 \leq 3,\ x_5 - x_6 \leq 4,\ x_1 - x_4 \neq 1\}$ is strong 3-consistent but not 4-consistent. Adding the constraint $x_2 - x_4 \neq 6 \ \vee \ x_3 - x_4 \neq -1$ (as in the previous example) is not enough. For example, the valuation $v = \{x_5 \leftarrow 2,\ x_6 \leftarrow -2,\ x_1 \leftarrow 2\}$ satisfies $C(x_5, x_6, x_1) = \{x_5 - x_6 \leq 4\}$ but it cannot be extended to a valuation which satisfies $C(x_5, x_6, x_1, x_4)$.

We can enforce 4-consistency by also adding the constraint $x_5 - x_1 \neq 0 \ \vee \ x_6 - x_1 \neq -4$ to C. Let the resulting set be C'. C' is strong 4-consistent but not 5-consistent. For example, the valuation $v = \{x_2 \leftarrow 7,\ x_3 \leftarrow 0,\ x_5 \leftarrow 2,\ x_6 \leftarrow -2\}$ satisfies $C(x_2, x_3, x_5, x_6) = \{x_2 - x_3 \leq 7,\ x_5 - x_6 \leq 4\}$ but it cannot be extended to a valuation which satisfies $C(x_2, x_3, x_5, x_6, x_1)$ (or $C(x_2, x_3, x_5, x_6, x_4)$).

We can enforce 5-consistency by adding the constraint

$$x_2 - x_3 \neq 7 \ \vee \ x_5 - x_6 \neq 4 \ \vee \ x_2 - x_5 \neq 5$$

to C'. The resulting constraint set is strong 5-consistent and also globally consistent.

Figure 2 presents algorithm TCN-GConsistency which enforces global consistency on its input TCN. TCN-GConsistency takes as input a TCN and returns an equivalent set of temporal constraints which is globally consistent. TCN-GConsistency's output is not a TCN because, as the above examples indicate, enforcing global consistency might result in the introduction of disjunctions of inequations which cannot be represented by a TCN. TCN-GConsistency takes advantage of an observation of [Kou92,Imb93]: inequations can be treated *independently* of one another for performing variable elimination.

The algorithm TCN-GConsistency essentially enforces strong 5-consistency on its input network N. As we will show shortly, this level of local consistency is enough for achieving global consistency. In step 1, TCN-GConsistency

Algorithm TCN-GConsistency
Input: A consistent TCN N.
Output: A globally consistent set of constraints equivalent to N.

Method:
Step 1: Enforce path consistency on conv(N).
For $k, i, j = 1$ to n do
$N_{ij} := N_{ij} \oplus (conv(N_{ik}) \otimes conv(N_{kj}))$
EndFor

Step 2: Enforce global consistency.
$C := \emptyset$
For $i, k = 1$ to n do
For $g = 1$ to h_{ik} do

Step 2.1
For $m, l = 1$ to n do
If N_{im}, N_{li} are closed from the right then
$C := C \cup \{x_m - x_k \neq d_{im} + r^g_{ik} \ \vee \ x_l - x_k \neq -d_{li} + r^g_{ik}\}$
EndIf
EndFor

Step 2.2
For $m, l, s, t = 1$ to n do
If $N_{im}, N_{li}, N_{ks}, N_{tk}$ are closed from the right then
$C := C \cup \{x_m - x_l \neq d_{im} + d_{li} \ \vee \ x_s - x_t \neq d_{ks} + d_{tk} \ \vee$
$x_m - x_s \neq r^g_{ik} + d_{im} - d_{ks}\}$
Endif
EndFor

EndFor
EndFor

Return $Constraints(N) \cup C$

Fig. 2. Enforcing global consistency

enforces strong 3-consistency on $conv(N)$. This is achieved by running the modified Floyd-Warshall algorithm of [DMP91] on $conv(N)$. Let N' denote the resulting TCN and $A' = Constraints(N')$. Then $conv(N')$ is minimal and globally consistent [DMP91].

In step 2, TCN-GConsistency completes its job. For each $r^g_{ik} \in holes(N_{ki})$ or equivalently for each inequation $x_i - x_k \neq r^g_{ik}$ of $A = Constraints(N)$, TCN-GConsistency explores the inequalities of A involving x_i and x_k in the following systematic way. Figure 3 illustrates the structure of the subnetworks

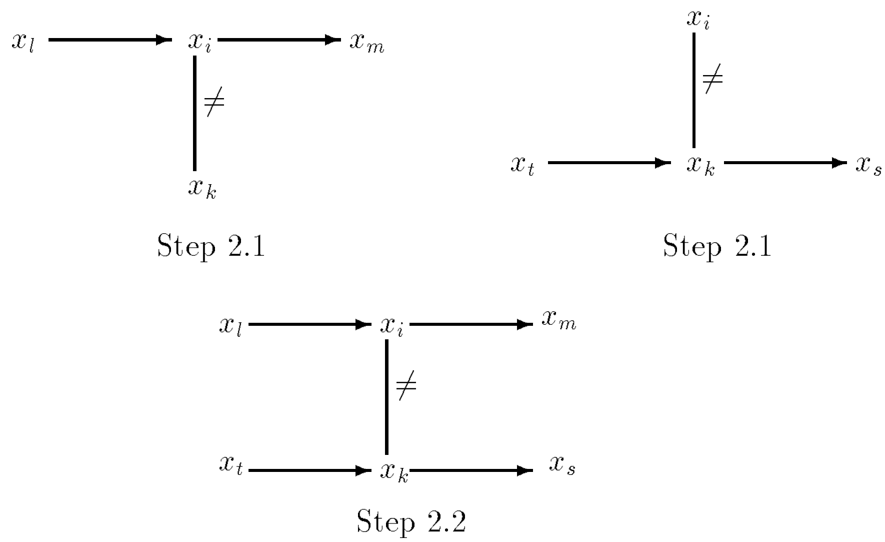

Fig. 3. The subnetworks examined by step 2 of TCN-GConsistency

of N explored in this step. Edges labeled with $\neq$ denote non-convex intervals.

(i) If there are inequalities $x_m - x_i \leq d_{im}$ and $x_i - x_l \leq d_{li}$ then step 2.1 ensures that any valuation $v = \{x_l \leftarrow x_l^0, x_m \leftarrow x_m^0, x_k \leftarrow x_k^0\}$, which satisfies $A(x_l, x_m, x_k)$, can be extended to a valuation $v' = v \cup \{x_i \leftarrow x_i^0\}$ which satisfies $A(x_l, x_m, x_k, x_i)$. This is achieved with the introduction of the inequation constraint

$$x_m - x_k \neq d_{im} + r_{ik}^g \ \vee \ x_l - x_k \neq -d_{li} + r_{ik}^g.$$

If there are inequalities $x_s - x_k \leq d_{ks}$ and $x_k - x_t \leq d_{tk}$ then step 2.1 also ensures that any valuation $v = \{x_s \leftarrow x_s^0, x_t \leftarrow x_t^0, x_i \leftarrow x_i^0\}$, which satisfies $A(x_s, x_t, x_i)$, can be extended to a valuation $v' = v \cup \{x_k \leftarrow x_k^0\}$ which satisfies $A(x_s, x_t, x_i, x_k)$. This is achieved with the introduction of the inequation constraint

$$x_s - x_i \neq d_{ks} - r_{ik}^g \ \vee \ x_t - x_i \neq -d_{tk} - r_{ik}^g.$$

(ii) If there are inequalities $x_m - x_i \leq d_{im}$, $x_i - x_l \leq d_{li}$, $x_s - x_k \leq d_{ks}$ and $x_k - x_t \leq d_{tk}$ then step 2.2 ensures that any valuation $v = \{x_l \leftarrow x_l^0, x_m \leftarrow x_m^0, x_s \leftarrow x_s^0, x_t \leftarrow x_t^0\}$, which satisfies $A(x_l, x_m, x_s, x_t)$, can be extended to a valuation $v' = v \cup \{x_i \leftarrow x_i^0, x_k \leftarrow x_k^0\}$ which satisfies $A(x_l, x_m, x_s, x_t, x_i, x_k)$. This is achieved with the introduction of the inequation constraint

$$x_m - x_l \neq d_{im} + d_{li} \ \vee \ x_s - x_t \neq d_{ks} + d_{tk} \ \vee \ x_m - x_s \neq r_{ik}^g + d_{im} - d_{ks}.$$

Discussion. It is possible that step 2 of algorithm TCN-GConsistency introduces constraints that are not strictly necessary for enforcing global consistency. This happens when a generated constraint is equivalent to *true* or when it is implied by another constraint. TCN-GCONSISTENCY can also introduce disjunctions of inequations that are equivalent to inequations (e.g., $x_1 - x_5 \neq 2 \ \vee \ x_1 - x_5 \neq 2$). We tolerate this inefficiency because it allow us to present our ideas clearly and minimizes the case analysis in the forthcoming proofs. The reader can consult [Kou94c] for an improved but complicated version of TCN-GCONSISTENCY.

The following theorem demonstrates the correctness of algorithm TCN-GCONSISTENCY. Its proof, presented in Appendix A, is rather long but easy to follow.

Theorem 3.1 *The algorithm* TCN-GCONSISTENCY *is correct i.e., it returns a globally consistent set of constraints equivalent to the input network.*

Corollary 3.1 *Strong 5-consistency is necessary and sufficient for achieving global consistency of a TCN.*

Proof: Example 3.4 shows the necessity of achieving strong 5-consistency. The sufficiency follows from the previous theorem; the algorithm TCN-GCONSISTENCY essentially achieves strong 5-consistency. □■

The following theorem gives the complexity of TCN-GCONSISTENCY.

Theorem 3.2 *The running time of* TCN-GCONSISTENCY *is* $O(Hn^4)$ *where* H *is the number of inequations and* n *is the number of variables in the input TCN.*

Proof: Step 1 takes $O(n^3)$ time, step 2.1 takes $O(Hn^2)$ time and step 2.2 takes $O(Hn^4)$ time. □■

4 Computing Minimal TCN

In this section we present an algorithm for computing the minimal network equivalent to a given TCN. Minimal networks are important representations because they make explicit all binary constraints implied by a given network. In the words of Montanari, a minimal network M " ... is perfectly explicit: as far as the pair of variables x_i and x_j is concerned, the rest of the network does not add any further constraint to the direct constraint M_{ij}" [Mon74]. Minimal networks have been studied extensively in temporal reasoning as important tools for answering queries concerning given temporal information (see [vB90b,vB92,DMP91], and especially [vB91] for examples). For example, let C be a set of temporal constraints of the form $x_i - x_j \leq r$ where x_i, x_j are variables ranging over the rational (or real) numbers and r is a rational (or real) constant. The minimal network corresponding to C can be computed in $O(n^3)$ time and $O(n^2)$ space [DMP91]. Then the minimal network can be used to answer in *constant* time all "interesting" queries of the form "Does $x_i - x_j \sim r$ follow from the constraints in C?" (where r is a rational constant and $\sim$ is $\leq$ or $=$).

We will also consider a network to be minimal if it makes explicit all "interesting" binary constraints. In our case "interesting" binary constraints are all constraints of the form $x_i - x_j \sim r$ where x_i, x_j are variables ranging over the rational numbers, r is a rational constant, and $\sim$ is $\leq, =$ or $\neq$. The following definition will suffice for our purpose [DMP91,Mei91b].

Definition 4.1 A TCN M is *tighter* than a TCN N if for every i, j, $M_{ij} \subseteq N_{ij}$. A TCN N is called *minimal* if there is no tighter network equivalent to it.

For our class of constraints the above definition of minimality slightly deviates from the standard intuitions behind minimal networks (as stated by Montanari [Mon74]). To see this consider the constraint set

$$C = \{x_1 \leq x_2,\ x_2 \leq 5,\ x_2 \neq x_3\}.$$

If we adopt our definition, the minimal TCN N for C has $N_{13} = (-\infty, +\infty)$. But C also implies the *disjunctive* binary constraint $x_3 \neq x_1 \ \vee \ x_3 \neq 5$ which *cannot* be represented by N. Thus if one is interested in answering queries involving disjunctive binary constraints then one has to discard the above definition and adopt the one in [Dec92]. In this case a *set of constraints* will be called minimal if and only if any instantiation of two variables which satisfies the constraints involving these variables, can be extended to a solution of the full network [Dec92].

The minimal network algorithm TCN-Minimal, shown in Figure 4, is essentially a by-product of algorithm TCN-GConsistency. As we discussed above, the constraints in the minimal TCN will be only inequalities and inequations. Therefore an algorithm for computing the minimal TCN can be constructed if we start with TCN-GConsistency and omit any part that generates a disjunction of inequations. This can be achieved by a detailed analysis of Step 2 of TCN-GConsistency. If we want to adopt the second definition of the minimal network and take into account disjunctive binary constraints then we have to modify TCN-Minimal accordingly.

TCN-Minimal computes the minimal TCN in four steps. In the first step, we enforce path-consistency on the convex part $conv(N)$ of the input network N. Steps 2, 3 and 4 are illustrated in Figure 5. In Step 2, TCN-Minimal performs constraint propagation involving equalities from $conv(N)$ and inequations from L. More precisely, for every inequation $x_i - x_j \neq r \in L$ and every equality $x_k - x_i = d_{ki} \in conv(N)$ Step 2.1 adds inequation $x_k - x_j \neq r + d_{ki}$ to N. Similarly, for every inequation $x_i - x_j \neq r \in L$ and every equality

Algorithm TCN-MINIMAL
Input: A consistent TCN N.
Output: A minimal TCN equivalent to N.
Method:
Step 1: Enforce path consistency on $conv(N)$ (as in Step 1 of TCN-GCONSISTENCY).

Step 2:
Let L be the list of inequations in N.
<u>For</u> every (i, j, r) in L <u>do</u>
    Step 2.1:
    <u>For</u> $k = 1$ to n <u>do</u>
        <u>If</u> $-d_{ik} = d_{ki}$ <u>then</u>
            $N_{kj} := N_{kj} \oplus ((-\infty, r + d_{ki}) \cup (r + d_{ki}, \infty))$
        <u>EndIf</u>
    <u>EndFor</u>

    Step 2.2:
    <u>For</u> $k = 1$ to n <u>do</u>
        <u>If</u> $-d_{kj} = d_{jk}$ <u>then</u>
            $N_{ik} := N_{ik} \oplus ((-\infty, r + d_{jk}) \cup (r + d_{jk}, \infty))$
        <u>EndIf</u>
    <u>EndFor</u>
<u>EndFor</u>

Step 3:
<u>For</u> every (i, k, r) in L <u>do</u>
    <u>For</u> $m, l = 1$ <u>to</u> n <u>do</u>
        <u>If</u> $N_{im}, N_{li}, N_{km}, N_{lk}$ are closed from the right and $m \neq l$ and
            $r + d_{im} - d_{km} = 0$ and $d_{li} + d_{im} = d_{lk} + d_{km}$ <u>then</u>
                $N_{lm} := N_{lm} \oplus ((-\infty, d_{li} + d_{im}) \cup (d_{li} + d_{im}, \infty))$
        <u>EndIf</u>
    <u>EndFor</u>
<u>EndFor</u>

Step 4:
<u>For</u> every (i, k, r) in L <u>do</u>
    <u>For</u> $m, t = 1$ <u>to</u> n <u>do</u>
        <u>If</u> $-d_{mi} = d_{im}$ and $-d_{kt} = d_{tk}$ <u>then</u>
            $N_{tm} := N_{tm} \oplus ((-\infty, d_{im} + d_{tk} + r) \cup (d_{im} + d_{tk} + r, \infty))$
        <u>EndIf</u>
    <u>EndFor</u>
<u>EndFor</u>

Return N

Fig. 4. A minimal TCN algorithm

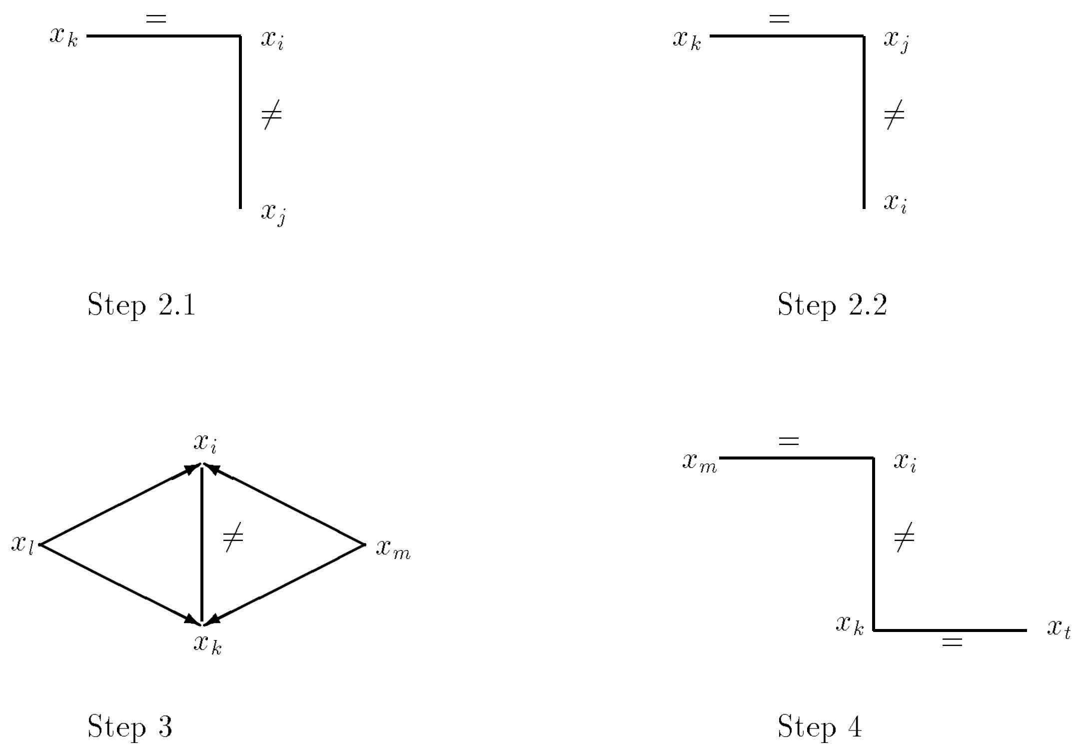

Fig. 5. The networks examined by algorithm TCN-Minimal

$x_j - x_k = d_{jk} \in conv(N)$ Step 2.2 adds inequation $x_i - x_k \neq r + d_{jk}$ to N.

In Step 3, TCN-Minimal considers subnetworks of N like the ones considered by Step 2.2 of TCN-GConsistency (see Figure 3) when $l = t$ and $m = s$.[3] In this case the constraint generated by TCN-GConsistency is equivalent to a binary inequation thus it should be reflected in the minimal TCN. This can be shown as follows. If $l = t$ and $m = s$ then Step 2.2 of TCN-GConsistency examines the constraint set

$$\{x_m - x_i \leq d_{im},\ x_i - x_l \leq d_{li},\ x_m - x_k \leq d_{km},\ x_k - x_l \leq d_{lk},\ x_i - x_k \neq r\}$$

and generates the constraint

$$x_m - x_l \neq d_{im} + d_{li}\ \vee\ x_m - x_l \neq d_{km} + d_{lk}\ \vee\ 0 \neq r + d_{im} - d_{km}.$$

If $r + d_{im} - d_{km} = 0$ and $d_{im} + d_{li} = d_{km} + d_{lk}$ then the above constraint becomes $x_m - x_l \neq d_{im} + d_{li}$ otherwise it evaluates to *true*.

Finally, in Step 4 TCN-Minimal considers subnetworks of N like the ones

[3] The case where $l = s$ and $m = t$ does not need to be considered because it leads to disjunctions of inequations that are equivalent to *true*.

considered by Step 2.2 of TCN-GConsistency when $l = m$ and $t = s$. In this case the constraint generated by TCN-GConsistency is also equivalent to a binary inequation. This can be shown as follows. If $l = m$ and $t = s$ then Step 2.2 of TCN-GConsistency considers the constraint set

$$\{x_m - x_i \leq d_{im},\ x_i - x_m \leq d_{mi},\ x_t - x_k \leq d_{kt},\ x_k - x_t \leq d_{kt},\ x_i - x_k \neq r\}$$

and generates the constraint

$$-d_{im} \neq d_{mi} \ \vee\ -d_{kt} \neq d_{tk} \ \vee\ x_m - x_t \neq d_{im} + d_{tk} + r.$$

If $-d_{im} = d_{mi}$ and $-d_{kt} = d_{tk}$ then this constraint becomes $x_m - x_t \neq d_{im} + d_{tk} + r$ otherwise it evaluates to *true*.

The following lemma summarizes the above discussion.

Lemma 4.1 *If* TCN-GConsistency *computes a binary inequation c and N is the output of* TCN-Minimal *then $c \in Constraints(N)$.*

The following theorem shows that the algorithm TCN-Minimal is correct and gives its complexity.

Theorem 4.1 *The algorithm* TCN-Minimal *computes the minimal TCN equivalent to its input in $O(\max(Hn^2, n^3))$ time where H is the number of inequations and n is the number of variables.*

Proof: The correctness part follows from the previous lemma. The complexity bound is achieved by either maintaining L explicitly or by having an adjacency list recording the inequations for every node of N. □■

An algorithm with the same complexity has also been discovered independently by Gerevini and Cristani without prior analysis of the global consistency problem [GC95]. A careful comparison of the two algorithms shows that Step 2 of TCN-Minimal computes *3-path implicit inequations*, Step 3 deals with *forbidden subgraphs* and Step 4 deals with *4-path implicit inequations* (this new terminology comes from [GC95] and the reader is referred there for more details).

Independently, Isli has studied a subclass of the class of temporal constraints that we consider in this section [Isl94]. Isli does not consider inequations of the form $x - y \neq r$ where $r \neq 0$, and achieves the same complexity bound for computing the minimal network.

5 Global Consistency of Point Algebra Networks

We will now turn our attention to an important subset of TCN: the point algebra networks introduced in [VKvB89]. A *point algebra network (PAN)* is a labeled directed graph where nodes represent variables and edges represent PA constraints. The labels of the edges are chosen from the set of relations $\{<, \leq, >, \geq, =, \neq, ?\}$. The symbol ? is used to label an edge $i \rightarrow j$ whenever there is no constraint between variables x_i and x_j.

Van Beek and Cohen have studied PAN in detail [vBC90,vB92]. Theorem 4.1 and the following results of [vB92] show that the complexity of computing the minimal network does not change when we go from PAN to TCN.

Theorem 5.1 *The minimal network equivalent to a PAN can be computed in $O(\max(Hn^2, n^3))$ time where H is the number of edges labeled with $\neq$ and n is the number of nodes.*

In [vBC90] the minimal network is computed by algorithm AAC. However, in the proof of correctness of AAC (Theorem 4 of [vBC90]), Van Beek and Cohen suggest that the algorithm for computing the minimal network of a given PAN also achieves global consistency. This is not true and has been corrected in [vB90a]. As the following example demonstrates, the introduction of disjunctions of inequations is necessary for achieving global consistency in this case. But algorithm AAC of [vBC90] does not introduce such disjunctions so it cannot achieve global consistency.

Example 5.1 For the PAN with constraints

$$x_1 \leq x_2,\ x_2 \leq x_3,\ x_4 \leq x_5,\ x_5 \leq x_6,\ x_2 \neq x_5$$

AAC will also introduce constraints $x_1 \leq x_3,\ x_4 \leq x_6$. The resulting PAN is strong 3-consistent but *not* globally consistent. This can be demonstrated via an argument similar to the one for Example 3.4. If we enforce strong 5-consistency with the addition of constraints $x_1 \neq x_5 \ \vee\ x_3 \neq x_5,\ x_4 \neq x_2 \ \vee\ x_6 \neq x_2$ and $x_1 \neq x_3 \ \vee\ x_1 \neq x_4 \ \vee\ x_1 \neq x_6$, then the resulting set is globally consistent.

Global consistency of PAN can be enforced by TCN-GConsistency if PAN are represented by their equivalent TCN. The following theorem summarizes the result of Section 3 as it applies to PAN.

Theorem 5.2 *Strong 5-consistency is necessary and sufficient for achieving global consistency in PAN. Strong 5-consistency can be enforced in $O(Hn^4)$ time where H is the number of edges labeled with $\neq$ and n is the number of nodes.*

Global consistency of PAN has also been discussed (under the name decomposability) in Section 5 of [Kou92] and algorithm Decompose has been proposed for achieving this task. The algorithm is correct but it adopts a representation which is rather inappropriate for the task at hand and leads to a complexity bound which is not the tightest. The results of this section subsume the results of Section 5 (only!) of [Kou92].

Let us now comment on some observations of Dechter [Dec92] on the problem of enforcing global consistency in PAN. [Dec92] discusses global consistency in general constraint networks with finite variable domains. The most important result of [Dec92] is the following. If N is a constraint network with constraints of arity r or less and domains of size k or less which is strongly $(k(r-1)+1)$-consistent, then N is globally consistent.

The above result can be applied to PAN if PAN are redefined as "traditional" constraint networks where variables represent relations between two points and constraints are defined by the transitivity table of [VKvB89]. This representation yields a constraint network with $k = 3$ and $r = 3$. Dechter's result now gives us the following. If strong 7-consistency in PAN can be enforced with ternary constraints then strong 7-consistency implies global consistency. Dechter uses the aforementioned incorrect assertion of [vBC90] to conclude that strong 7-consistency in the traditional formulation of PAN can be enforced with ternary constraints. Thus she also concludes that in the traditional formulation strong 7-consistency implies global consistency [Dec92, page 100]. In the light of Theorem 5.2, Dechter's conclusion remains unjustified.

6 The General Case

Let us now consider enforcing global consistency when disjunctions of inequations are allowed in the given constraint set.

Example 6.1 The constraint set

$$C = \{x_5 \leq x_1,\ x_1 \leq x_6,\ x_5 \leq x_6,\ x_7 \leq x_3,\ x_3 \leq x_8,\ x_7 \leq x_8,\ x_9 \leq x_2,\ x_2 \leq x_{10},\ x_9 \leq x_{10},\ x_1 \neq y\ \vee\ x_2 \neq z\ \vee\ x_3 \neq w\}$$

is strong 7-consistent but not 8-consistent. For example, the valuation

$$v = \{y \leftarrow 0,\ z \leftarrow 0,\ w \leftarrow 0,\ x_2 \leftarrow 0,\ x_3 \leftarrow 0,\ x_5 \leftarrow 0,\ x_6 \leftarrow 0\}$$

satisfies $C(y, z, w, x_2, x_3, x_5, x_6) = \{x_5 \leq x_6\}$ but it cannot be extended to a valuation which satisfies $C(y, z, w, x_2, x_3, x_5, x_6, x_1)$. We can enforce 8-consistency by adding the constraints

$$\begin{array}{c} x_5 \neq y \ \vee \ x_6 \neq y \ \vee \ x_2 \neq z \ \vee \ x_3 \neq w \\ x_1 \neq y \ \vee \ x_9 \neq z \ \vee \ x_{10} \neq z \ \vee \ x_3 \neq w \\ x_1 \neq y \ \vee \ x_2 \neq z \ \vee \ x_7 \neq w \ \vee \ x_8 \neq w. \end{array}$$

The resulting set is strong 8-consistent but not 9-consistent. We can enforce 9-consistency by adding the constraints

$$\begin{array}{c} x_5 \neq y \ \vee \ x_6 \neq y \ \vee \ x_9 \neq z \ \vee \ x_{10} \neq z \ \vee \ x_3 \neq w \\ x_1 \neq y \ \vee \ x_9 \neq z \ \vee \ x_{10} \neq z \ \vee \ x_7 \neq w \ \vee \ x_8 \neq w \\ x_5 \neq y \ \vee \ x_6 \neq y \ \vee \ x_2 \neq z \ \vee \ x_7 \neq w \ \vee \ x_8 \neq w. \end{array}$$

The resulting set is strong 9-consistent but not 10-consistent. We can enforce 10-consistency by adding the constraint

$$x_5 \neq y \ \vee \ x_6 \neq y \ \vee \ x_9 \neq z \ \vee \ x_{10} \neq z \ \vee \ x_7 \neq w \ \vee \ x_8 \neq w.$$

The resulting set is strong 10-consistent and also globally consistent.

Figure 6 presents algorithm GConsistency which enforces global consistency on its input constraint set. The reader should have no problem understanding the details of GConsistency since it is a straightforward generalization of algorithm TCN-GConsistency.

The following theorem demonstrates the correctness of GConsistency. The proof is given in Appendix A.

Theorem 6.1 *The algorithm* GConsistency *is correct i.e., it returns a globally consistent set of constraints equivalent to the input one.*

In essence, algorithm GConsistency achieves strong $2V + 1$-consistency where V is the maximum number of variables in any disjunction of inequations. Thus we have the following corollary.

Corollary 6.1 *Let C be a set of temporal constraints. If C is $2V+1$-consistent, where V is the maximum number of variables in any disjunction of inequations, then C is globally consistent.*

The time complexity of GConsistency is exponential in V. However, if V is *fixed* then the time complexity of GConsistency is polynomial in the number of variables and the number of constraints in C. This has an interesting consequence for variable elimination due to its relation to global consistency.

Corollary 6.2 *Let C be a set of temporal constraints such that the number of variables in every disjunction of inequations is fixed. Eliminating any number*

Algorithm GCONSISTENCY
Input: A set of temporal constraints $C = C_i \cup C_d$ where C_i is a set of inequalities and C_d is a set of disjunctions of inequations.
Output: A globally consistent set of constraints equivalent to C.

Method:
Step 1: Enforce strong 3-consistency on C_i.
Let N be the TCN corresponding to C_i.
For $k, i, j = 1$ to n do
    $N_{ij} := N_{ij} \oplus (N_{ik} \otimes N_{kj})$
EndFor

Step 2: Enforce global consistency
$C'_d := \emptyset$
For each $c \in C_d$ do
    For all subsets $\{k_1, \ldots, k_i\}$ of the set of variables of c do
        For $m_1, \ldots, m_i, l_1, \ldots, l_i = 1$ to n do
            If $N_{k_1 m_1}, \ldots, N_{k_i m_i}, N_{l_1 k_1}, \ldots, N_{l_i k_i}$ are closed from the right then
                Eliminate variables $x_{k_1}, \ldots, x_{k_i}$ from
                $\overline{c},\ x_{m_1} - x_{k_1} = d_{k_1 m_1},\ x_{k_1} - x_{l_1} = d_{l_1 k_1}, \ldots, x_{m_i} - x_{k_i} = d_{i m_i},$
                $x_{k_i} - x_{l_i} = d_{l_i k_i}$
                to obtain c'
                $C'_d := C'_d \cup \{\overline{c'}\}$
            Endif
        EndFor
    EndFor
EndFor

Return $Constraints(N) \cup C_d \cup C'_d$

Fig. 6. Enforcing global consistency

of variables from C can be done in time polynomial in the number of variables and the number of constraints. In addition, the resulting constraint set has size polynomial in the same parameters.

Proof: Let $x_1, \ldots, x_n$ be all the variables of C. When V is fixed, the size of the constraint set generated by algorithm GCONSISTENCY is polynomial in the number of variables and the number of constraints. If C is globally consistent then for any i such that $1 \leq i \leq n$, $C(x_1, \ldots, x_i)$ is the projection of $Sol(C)$ on $\{x_1, \ldots, x_i\}$. Thus we can eliminate variables $x_1, \ldots, x_i$ from C by running GCONSISTENCY on C and returning $C(x_{i+1}, \ldots, x_n)$. This algorithm takes time polynomial in the number of variables and the number of constraints. ■

The above corollary complements Theorem 4.4 of [Kou92] which states that variable elimination can result in constraint sets with an exponential number of disjunctions of inequations.

7 Conclusions

We discussed the problem of enforcing global consistency in sets of quantitative temporal constraints which include inequalities, inequations and disjunctions of inequations. In future research it would be interesting to consider directional consistency algorithms for this class of temporal constraints [DP88]. It would also be interesting to combine our results with the results of [Mei91b] in order to identify classes of qualitative and quantitative point/interval constraints where global consistency is tractable.

8 Acknowledgements

I would like to thank Peter van Beek, Rina Dechter, Amar Isli, the reviewers of CP95 and the reviewers of this paper for interesting comments. I would also like to thank Alfonso Gerevini for pointing out to me that an earlier version of Lemma 4.1 had been stated inaccurately.

References

[Coo90] M.C. Cooper. An optimal k-consistency algorithm. *Artificial Intelligence*, 41(1):89–95, 1990.

[Dec92] R. Dechter. From local to global consistency. *Artificial Intelligence*, 55:87–107, 1992.

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal Constraint Networks. *Artificial Intelligence*, 49(1-3):61–95, 1991. Special Volume on Knowledge Representation.

[DP88] R. Dechter and J. Pearl. Network-Based Heuristics for Constraint Satisfaction Problems. *Artificial Intelligence*, 34(1):1–38, 1988.

[Fre78] E. Freuder. Synthesizing Constraint Expressions. *Communications of ACM*, 21(11):958–966, November 1978.

[Fre82] E. Freuder. A Sufficient Condition For Backtrack-Free Search. *Journal of ACM*, 29(1):24–32, 1982.

[GC95] A. Gerevini and M. Cristani. Reasoning with Inequations in Temporal Constraint Networks. Technical report, IRST - Instituto per la Ricerca

Scientifica e Tecnologica, Povo TN, Italy, 1995. A shorter version appears in the Proceedings of the Workshop on Spatial and Temporal Reasoning, IJCAI-95.

[GS93] A. Gerevini and L. Schubert. Efficient Temporal Reasoning Through Timegraphs. In *Proceedings of IJCAI-93*, pages 648–654, 1993.

[GSS93] A. Gerevini, L. Schubert, and S. Schaeffer. Temporal Reasoning in Timegraph I-II. *SIGART Bulletin*, 4(3):21–25, 1993.

[Imb93] J.-L. Imbert. Variable Elimination for Generalized Linear Constraints. In *Proceedings of the 10th International Conference on Logic Programming*, 1993.

[Imb94] J.-L. Imbert. Redundancy, Variable Elimination and Linear Disequations. In *Proceedings of the International Symposium on Logic Programming*, pages 139–153, 1994.

[Isl94] A. Isli. Constraint-Based Temporal Reasoning: a Tractable Point Algebra Combining Qualitative, Metric and Holed Constraints. Technical Report 94-06, LIPN-CNRS URA 1507, Inst. Galilée, Université Paris-Nord, 1994.

[IvH93] J.-L. Imbert and P. van Hentenryck. On the Handling of Disequations in CLP over Linear Rational Arithmetic. In F. Benhamou and A. Colmerauer, editors, *Constraint Logic Programming: Selected Research*, Logic Programming Series, pages 49–71. MIT Press, 1993.

[Kou92] M. Koubarakis. Dense Time and Temporal Constraints with $\neq$. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR'92)*, pages 24–35. Morgan Kaufmann, San Mateo, CA, October 1992.

[Kou94a] M. Koubarakis. Complexity Results for First-Order Theories of Temporal Constraints. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR'94)*, pages 379–390. Morgan Kaufmann, San Francisco, CA, May 1994.

[Kou94b] M. Koubarakis. Foundations of Indefinite Constraint Databases. In A. Borning, editor, *Proceedings of the 2nd International Workshop on the Principles and Practice of Constraint Programming (PPCP'94)*, volume 874 of *Lecture Notes in Computer Science*, pages 266–280. Springer Verlag, 1994.

[Kou94c] M. Koubarakis. *Foundations of Temporal Constraint Databases.* PhD thesis, Computer Science Division, Dept. of Electrical and Computer Engineering, National Technical University of Athens, February 1994. Available electronically from `http://www.co.umist.ac.uk/~manolis/MK/M-Koubarakis.html`.

[LM89] Jean-Louis Lassez and Ken McAloon. A Canonical Form for Generalized Linear Constraints. Technical Report RC15004 (#67009), IBM Research Division, T.J. Watson Research Center, 1989.

[Mei91a] I. Meiri. Combining Qualitative and Quantitative Constraints in Temporal Reasoning. Technical Report R-160, Cognitive Systems Laboratory, University of California, Los Angeles, 1991.

[Mei91b] I. Meiri. Combining Qualitative and Quantitative Constraints in Temporal Reasoning. In *Proceedings of AAAI-91*, pages 260–267, 1991.

[Mon74] U. Montanari. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Information Sciences*, 7:95–132, 1974.

[Sch86] A. Schrijver, editor. *Theory of Integer and Linear Programming*. Wiley, 1986.

[vB90a] Peter van Beek. Exact and Approximate Reasoning About Qualitative Temporal Relations. Technical Report TR 90-29, Department of Computing Science, University of Alberta, August 1990.

[vB90b] Peter van Beek. Reasoning About Qualitative Temporal Information. In *Proceedings of AAAI-90*, pages 728–734, 1990.

[vB91] Peter van Beek. Temporal Query Processing with Indefinite Information. *Artificial Intelligence in Medicine*, 3:325–339, 1991.

[vB92] Peter van Beek. Reasoning About Qualitative Temporal Information. *Artificial Intelligence*, 58:297–326, 1992.

[vBC90] Peter van Beek and Robin Cohen. Exact and Approximate Reasoning about Temporal Relations. *Computational Intelligence*, 6:132–144, 1990.

[VK86] Marc Vilain and Henry Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *Proceedings of AAAI-86*, pages 377–382, 1986.

[VKvB89] Marc Vilain, Henry Kautz, and Peter van Beek. Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report. In D.S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, 1989.

A Proofs

Proof of Theorem 3.1: Let C' denote $Constraints(N) \cup C$. The set C' is consistent therefore 1-consistency holds trivially. We will show that C' is ν-consistent for every ν, $2 \leq \nu \leq n$.

Let us take an arbitrary valuation $v = \{x_1 \leftarrow x_1^0, \ldots, x_{\nu-1} \leftarrow x_{\nu-1}^0\}$ such that $C'(x_1^0, \ldots, x_{\nu-1}^0)$ is satisfiable. We will show that for every variable x_ν, v can be extended to a valuation $v' = v \cup \{x_\nu \leftarrow x_\nu^0\}$ such that $C'(x_1^0, \ldots, x_\nu^0)$ is satisfiable.

If all constraints involving x_ν and any of $x_1, \ldots, x_{\nu-1}$ are inequalities, our result is immediate since $Constraints(N)$ is globally consistent. Let us then assume that $C'(x_1, \ldots, x_\nu)$ contains inequations, and consider $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu)$.

Let D_{ji} denote the number of inequation constraints involving $x_j - x_i$ in C'. Let I_i be the set of natural numbers j such that $x_j - x_i \neq r \vee \phi$ or $x_i - x_j \neq r \vee \phi$ is an inequation constraint in C'. Then $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu)$ can be written as

$$\{x_\mu^0 - d_{\nu\mu} \prec_1 x_\nu,\ x_\nu \prec_2 x_\lambda^0 + d_{\lambda\nu}\} \cup \bigcup_{\zeta \in I_\nu} \{x_\nu \neq x_\zeta^0 + r_{\nu\zeta}^1, \ldots, x_\nu \neq x_\zeta^0 + r_{\nu\zeta}^{D_{\nu\zeta}}\} \quad (A.1)$$

where $\mu, \lambda, \zeta \in \{1, \ldots, \nu-1\}$ and $\prec_1, \prec_2 \in \{<, \leq\}$. Since the rational numbers are dense, there is only one case which would not allow us to find a value x_ν^0 such that $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu^0)$ is satisfiable. This is the case when $\prec_1$ is $\leq$, $\prec_2$ is $\leq$ and there exists $\rho \in I_\nu$ and $\eta \in \{1, \ldots, D_{\nu\rho}\}$ such that

$$x_\mu^0 - d_{\nu\mu} = x_\lambda^0 + d_{\lambda\nu} = x_\rho^0 + r_{\nu\rho}^\eta. \quad (A.2)$$

We will show that this case *cannot* arise.

Depending on the form of the inequation constraint c from which inequation $x_\nu \neq x_\rho^0 + r_{\nu\rho}^\eta$ was generated, the following cases must be considered. Figure A.1 illustrates the analysis by depicting the subnetworks involved in each case.

(i) c is $x_\nu - x_\rho \neq r_{\nu\rho}^\eta \in Constraints(N)$ or equivalently $r_{\nu\rho}^\eta \in holes(N_{\rho\nu})$. In this case, the constraint $x_\mu - x_\rho \neq d_{\nu\mu} + r_{\nu\rho}^\eta \ \vee\ x_\lambda - x_\rho \neq r_{\nu\rho}^\eta - d_{\lambda\nu}$ is added to C in Step 2.1 of algorithm TCN-GConsistency with $g = \eta, m = \mu, l = \lambda$ and $k = \rho$. Then

$$x_\mu^0 - x_\rho^0 \neq d_{\nu\mu} + r_{\nu\rho}^\eta \ \vee\ x_\lambda^0 - x_\rho^0 \neq r_{\nu\rho}^\eta - d_{\lambda\nu} \in C'(x_1^0, \ldots, x_{\nu-1}^0)$$

thus we have a contradiction.

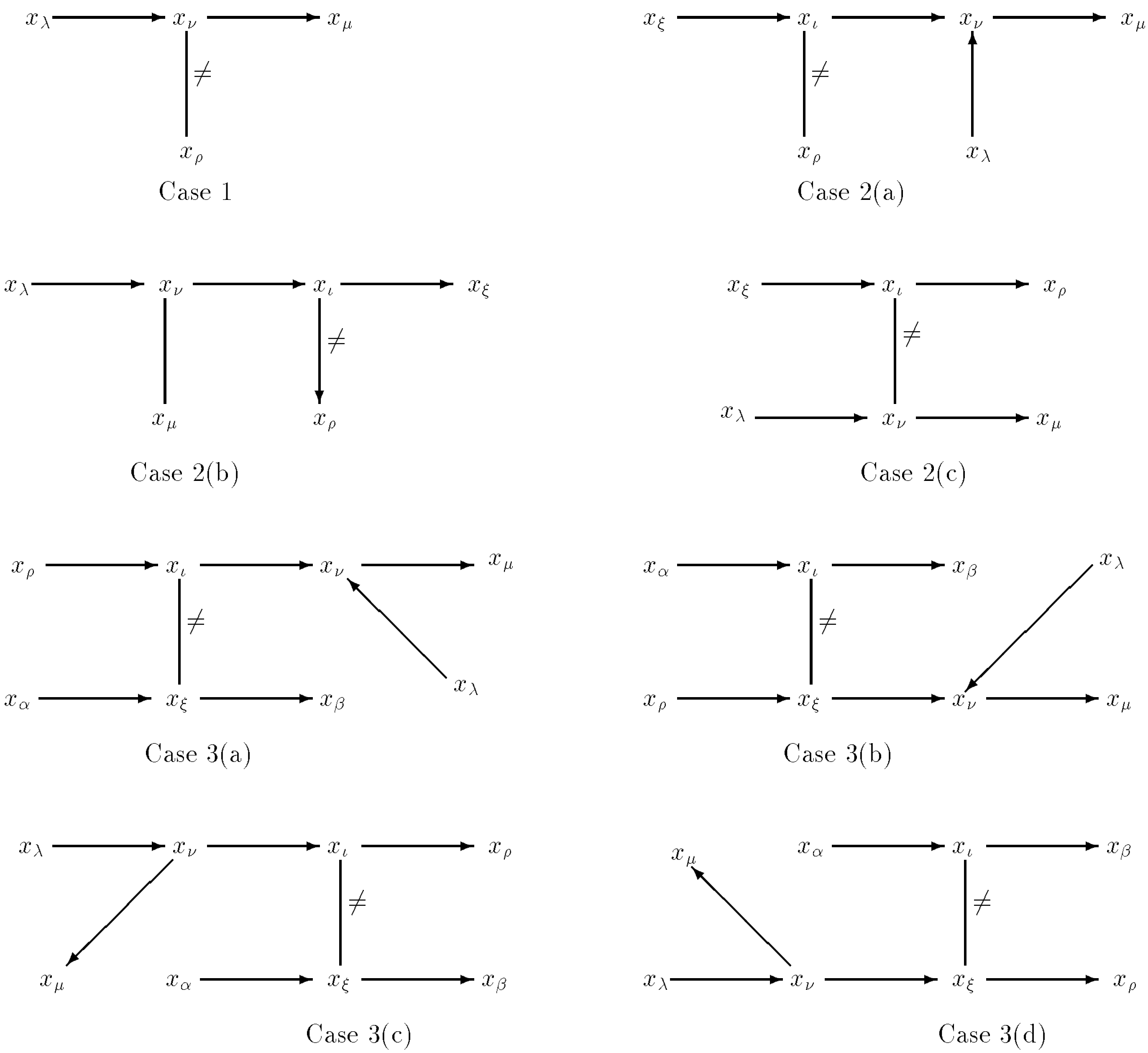

Fig. A.1. The cases examined in Theorem 3.1

(ii) c is added to C in Step 2.1 of TCN-GCONSISTENCY. Depending on the values of g, l, i, m and k we can consider the following subcases.

(a) c is added to C in Step 2.1 of TCN-GCONSISTENCY with $g = \eta, l = \xi, i = \iota, m = \nu$ and $k = \rho$. Thus c is $x_\nu - x_\rho \neq r^\eta_{\iota\rho} + d_{\iota\nu} \ \vee \ x_\xi - x_\rho \neq -d_{\xi\iota} + r^\eta_{\iota\rho}$. The constraints (A.1), (A.2) and c imply

$$x^0_\xi - x^0_\rho = -d_{\xi\iota} + r^\eta_{\iota\rho}, \quad x_\mu - x_\rho = d_{\nu\mu} + r^\eta_{\nu\rho} = d_{\nu\mu} + r^\eta_{\iota\rho} + d_{\iota\nu} \quad \text{(A.3)}$$

Now we have the following subcases:
$d_{\iota\mu} = d_{\iota\nu} + d_{\nu\mu}$. Then (A.3) contradicts the constraint

$$x_\mu - x_\rho \neq d_{\iota\mu} + r^\eta_{\iota\rho} \ \vee \ x_\xi - x_\rho \neq -d_{\xi\iota} + r^\eta_{\iota\rho}$$

of $C'(x_1, \ldots, x_{\nu-1})$ which is introduced in Step 2.1 of TCN-GConsistency with $g = \eta, l = \xi, i = \iota, m = \mu$ and $k = \rho$.
$d_{\iota\mu} < d_{\iota\nu} + d_{\nu\mu}$. Then $x^0_\mu - x^0_\xi \leq d_{\xi\mu} \leq d_{\xi\iota} + d_{\iota\mu} < d_{\xi\iota} + d_{\iota\nu} + d_{\nu\mu}$. This contradicts $x^0_\mu - x^0_\xi = d_{\xi\iota} + d_{\iota\nu} + d_{\nu\mu}$ which is implied by (A.3).

(b) c is added to C in Step 2.1 of TCN-GConsistency with $g = \eta, l = \nu, i = \iota, m = \xi$ and $k = \rho$. Thus c is $x_\xi - x_\rho \neq r^\eta_{\iota\rho} + d_{\iota\xi} \ \vee \ x_\nu - x_\rho \neq -d_{\nu\iota} + r^\eta_{\iota\rho}$. This case is symmetric to 2(a).

(c) c is added to C in Step 2.1 TCN-GConsistency with $g = \eta, l = \xi, i = \iota, m = \rho$ and $k = \nu$. Thus c is $x_\rho - x_\nu \neq r^\eta_{\iota\nu} + d_{\iota\rho} \ \vee \ x_\xi - x_\nu \neq -d_{\xi\iota} + r^\eta_{\iota\nu}$ or equivalently $x_\nu \neq x_\rho - r^\eta_{\iota\nu} - d_{\iota\rho} \ \vee \ x_\nu \neq x_\xi + d_{\xi\iota} - r^\eta_{\iota\nu}$. The constraints (A.1) and c imply $x^0_\rho - r^\eta_{\iota\nu} - d_{\iota\rho} = x^0_\xi + d_{\xi\iota} - r^\eta_{\iota\nu} = x^0_\rho + r^\eta_{\nu\rho}$. These equalities together with (A.2) imply

$$x^0_\rho - x^0_\xi = d_{\xi\iota} + d_{\iota\rho}, \quad x^0_\mu - x^0_\lambda = d_{\nu\mu} + d_{\lambda\nu}, \quad x^0_\rho - r^\eta_{\iota\nu} - d_{\iota\rho} = x^0_\mu - d_{\nu\mu}.$$

But for $g = \eta, l = \xi, m = \rho, i = \iota, k = \nu, t = \lambda$ and $s = \mu$ the constraint

$$x_\rho - x_\xi \neq d_{\iota\rho} + d_{\xi\iota} \ \vee \ x_\mu - x_\lambda \neq d_{\nu\mu} + d_{\lambda\nu} \ \vee \ x_\rho - x_\mu \neq r^\eta_{\iota\nu} + d_{\iota\rho} - d_{\nu\mu}$$

is added to C in Step 2.2 of TCN-GConsistency. This constraint also belongs to $C'(x_1, \ldots, x_{\nu-1})$ thus we have a contradiction.

(iii) c is added to C in Step 2.2 of TCN-GConsistency. Depending on the values of g, l, i, m, t, k and s we can consider the following subcases.

(a) c is added to C in Step 2.2 of TCN-GConsistency with $g = \eta, l = \rho, i = \iota, m = \nu, t = \alpha, k = \xi$ and $s = \beta$. Thus c is

$$x_\nu - x_\rho \neq d_{\iota\nu} + d_{\rho\iota} \ \vee \ x_\beta - x_\alpha \neq d_{\xi\beta} + d_{\alpha\xi} \ \vee \ x_\nu - x_\beta \neq r^\eta_{\iota\xi} - d_{\xi\beta} + d_{\iota\nu}.$$

The constraints (A.1) and c imply

$$x^0_\rho + d_{\iota\nu} + d_{\rho\iota} = x^0_\beta + r^\eta_{\iota\xi} - d_{\xi\beta} + d_{\iota\nu} = x^0_\rho + r^\eta_{\nu\rho}, \quad x^0_\beta - x^0_\alpha = d_{\xi\beta} + d_{\alpha\xi}.$$

These equations together with (A.2) imply

$$\begin{aligned} &x^0_\mu - x^0_\rho = d_{\iota\nu} + d_{\rho\iota} + d_{\nu\mu}, \quad x^0_\beta - x^0_\alpha = d_{\xi\beta} + d_{\alpha\xi}, \\ &x^0_\mu - x^0_\beta = d_{\nu\mu} + r^\eta_{\iota\xi} - d_{\xi\beta} + d_{\iota\nu} \end{aligned} \qquad \text{(A.4)}$$

Now we have to consider the following subcases:
$d_{\iota\mu} = d_{\iota\nu} + d_{\nu\mu}$. Then (A.4) contradicts the constraint

$$x_\mu - x_\rho \neq d_{\iota\mu} + d_{\rho\iota} \ \vee \ x_\beta - x_\alpha \neq d_{\xi\beta} + d_{\alpha\xi} \ \vee \ x_\mu - x_\beta \neq r^\eta_{\iota\xi} + d_{\iota\mu} - d_{\xi\beta}$$

of $C'(x_1,\ldots,x_{\nu-1})$ which is added to C in Step 2.2 of TCN-GConsistency with $g=\eta, l=\rho, i=\iota, m=\mu, t=\alpha, k=\xi$ and $s=\beta$.
$d_{\iota\mu} < d_{\iota\nu} + d_{\nu\mu}$. Then

$$x^0_\mu - x^0_\rho \le d_{\rho\mu} \le d_{\rho\iota} + d_{\iota\mu} < d_{\rho\iota} + d_{\iota\nu} + d_{\nu\mu}.$$

This contradicts the first equation of (A.4).

(b) c is introduced in Step 2.2 of TCN-GConsistency with $g=\eta, l=\alpha, i=\iota, m=\beta, t=\rho, k=\xi$ and $s=\nu$. Thus c is

$$x_\beta - x_\alpha \ne d_{\iota\beta} + d_{\alpha\iota} \;\vee\; x_\nu - x_\rho \ne d_{\xi\nu} + d_{\rho\xi} \;\vee\; x_\beta - x_\nu \ne r^\eta_{\iota\xi} + d_{\iota\beta} - d_{\xi\nu}.$$

This case is symmetric to case 3(a).

(c) c is introduced in Step 2.2 of TCN-GConsistency with $g=\eta, l=\nu, i=\iota, m=\rho, t=\alpha, k=\xi$ and $s=\beta$. Thus c is

$$x_\rho - x_\nu \ne d_{\iota\rho} + d_{\nu\iota} \;\vee\; x_\beta - x_\alpha \ne d_{\xi\beta} + d_{\alpha\xi} \;\vee\; x_\rho - x_\beta \ne r^\eta_{\iota\xi} + d_{\iota\rho} - d_{\xi\beta}$$

or $x_\nu - x_\rho \ne -d_{\iota\rho} - d_{\nu\iota} \vee x_\beta - x_\alpha \ne d_{\xi\beta} + d_{\alpha\xi} \vee x_\rho - x_\beta \ne r^\eta_{\iota\xi} + d_{\iota\rho} - d_{\xi\beta}$.
The constraints (A.1) and c imply

$$x^0_\beta - x^0_\alpha = d_{\xi\beta} + d_{\alpha\xi}, \quad x^0_\rho - x^0_\beta = r^\eta_{\iota\xi} + d_{\iota\rho} - d_{\xi\beta}, \quad r^\eta_{\nu\rho} = -d_{\iota\rho} - d_{\nu\iota}$$

The above equations together with (A.2) imply

$$\begin{aligned} &x^0_\beta - x^0_\alpha = d_{\xi\beta} + d_{\alpha\xi}, \quad x^0_\rho - x^0_\beta = r^\eta_{\iota\xi} + d_{\iota\rho} - d_{\xi\beta}, \\ &x^0_\rho - x^0_\lambda = d_{\lambda\nu} + d_{\iota\rho} + d_{\nu\iota} \end{aligned} \tag{A.5}$$

Now we have to consider the following cases:
$d_{\lambda\iota} = d_{\lambda\nu} + d_{\nu\iota}$. Then (A.5) contradicts the constraint

$$x_\rho - x_\lambda \ne d_{\iota\rho} + d_{\lambda\iota} \;\vee\; x_\beta - x_\alpha \ne d_{\xi\beta} + d_{\alpha\xi} \;\vee\; x_\rho - x_\beta \ne r^\eta_{\iota\xi} + d_{\iota\rho} - d_{\xi\beta}$$

of $C'(x_1,\ldots,x_{\nu-1})$ which is introduced in Step 2.2 of of TCN-GConsistency with $g=\eta, l=\lambda, i=\iota, m=\rho, t=\alpha, k=\xi$ and $s=\beta$.
$d_{\lambda\iota} < d_{\lambda\nu} + d_{\nu\iota}$. Then

$$x^0_\rho - x^0_\lambda \le d_{\lambda\rho} \le d_{\lambda\iota} + d_{\iota\rho} < d_{\lambda\nu} + d_{\nu\iota} + d_{\iota\rho}$$

which contradicts the last equation of (A.5).

(d) c is introduced in Step 2.2 of TCN-GConsistency with $g=\eta, l=\alpha, i=\iota, m=\beta, t=\nu, k=\xi$ and $s=\rho$. Thus c is

$$x_\beta - x_\alpha \ne d_{\iota\xi} + d_{\alpha\iota} \;\vee\; x_\rho - x_\nu \ne d_{\xi\rho} + d_{\nu\xi} \;\vee\; x_\beta - x_\rho \ne r^\eta_{\iota\xi} + d_{\iota\beta} - d_{\xi\rho}.$$

This case is symmetric to case 3(c). □

Proof of Theorem 6.1: The proof will have the same structure as the proof of theorem 3.1. Let C' be the set returned by GConsistency. Let us take an arbitrary valuation $v = \{x_1 \leftarrow x_1^0, \ldots, x_{\nu-1} \leftarrow x_{\nu-1}^0\}$ such that $C'(x_1^0, \ldots, x_{\nu-1}^0)$ is satisfiable. We will show that for every variable x_ν, v can be extended to a valuation $v' = v \cup \{x_\nu \leftarrow x_\nu^0\}$ such that $C'(x_1^0, \ldots, x_\nu^0)$ is satisfiable.

If all constraints involving x_ν and any of $x_1, \ldots, x_{\nu-1}$ are inequalities, our result is immediate since $Constraints(N)$ is globally consistent. Let us then assume that $C'(x_1, \ldots, x_\nu)$ contains inequations, and consider $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu)$.

Let D_{ji} denote the number of inequation constraints involving $x_j - x_i$ in C'. Let I_i be the set of natural numbers j such that $x_j - x_i \neq r \ \vee \ \phi$ or $x_i - x_j \neq r \ \vee \ \phi$ is an inequation constraint in C'. Then $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu)$ can be written as

$$\{x_\mu^0 - d_{\nu\mu} \prec_1 x_\nu, \ x_\nu \prec_2 x_\lambda^0 + d_{\lambda\nu}\} \cup \bigcup_{\zeta \in I_\nu} \{x_\nu \neq x_\zeta^0 + r_{\nu\zeta}^1, \ldots, x_\nu \neq x_\zeta^0 + r_{\nu\zeta}^{D_{\nu\zeta}}\} \quad \text{(A.6)}$$

where $\mu, \lambda, \zeta \in \{1, \ldots, \nu-1\}$ and $\prec_1, \prec_2 \in \{<, \leq\}$. Since the rational numbers are dense, there is only one case which would not allow us to find a value x_ν^0 such that $C'(x_1^0, \ldots, x_{\nu-1}^0, x_\nu^0)$ is satisfiable. This is the case when $\prec_1$ is $\leq$, $\prec_2$ is $\leq$ and there exists $\rho \in I_\nu$ and $\eta \in \{1, \ldots, D_{\nu\rho}\}$ such that

$$x_\mu^0 - d_{\nu\mu} = x_\lambda^0 + d_{\lambda\nu} = x_\rho^0 + r_{\nu\rho}^\eta. \quad \text{(A.7)}$$

We will show that this case *cannot* arise.

Depending on the form of the inequation constraint c_1 from which inequation $x_\nu \neq x_\rho^0 + r_{\nu\rho}^\eta$ was generated, the following cases must be considered.

(i) $c_1 \in C_d$. Then c_1 can be written as

$$x_\nu - x_\rho \neq r_{\nu\rho}^\eta \ \vee \ \overline{\phi}$$

where ϕ does not contain x_ν. When the set $\{\nu\}$ is considered by Step 2 of GConsistency and $m_1 = \mu, \ l_1 = \lambda$, the variable x_ν is eliminated from

$$\overline{c_1}, \ x_\mu - x_\nu = d_{\nu\mu}, \ x_\nu - x_\lambda = d_{\lambda\nu}$$

to obtain the following constraint c_2:

$$\overline{\phi} \ \vee \ x_\mu - x_\rho \neq d_{\nu\mu} + r_{\nu\rho}^\eta \ \vee \ x_\rho - x_\lambda \neq d_{\lambda\nu} - r_{\rho\nu}^\eta.$$

We have arrived at a contradiction since $c_2 \in C'(x_1^0, \ldots, x_{\nu-1}^0)$ and the equalities A.7 hold.

(ii) c_1 is added to C' in Step 2 of GConsistency. Depending on the values of $c, i, m_1, \ldots, m_i, l_1, \ldots, l_i$ we consider the following subcases.

(a) $c = c_3, i = \iota, m_1 = \beta_1, \ldots, m_j = \nu, \ldots, m_\iota = \beta_\iota,\ l_1 = \alpha_1, \ldots, l_j = \rho, \ldots, l_\iota = \alpha_\iota$.

Thus c_1 is obtained after variables $x_{k_1}, \ldots, x_{k_\iota}$ are eliminated from $\overline{c_3},\ x_{\beta_1} - x_{k_1} = d_{k_1\beta_1},\ x_{k_1} - x_{\alpha_1} = d_{\alpha_1 k_1}, \ldots, x_\nu - x_{k_j} = d_{k_j\nu},$
$x_{k_j} - x_\rho = d_{\rho k_j}, \ldots, x_{\beta_\iota} - x_{k_\iota} = d_{k_\iota\beta_\iota},\ x_{k_\iota} - x_{\alpha_\iota} = d_{\alpha_\iota k_\iota}$.

Therefore c_1 is
$\overline{c_3[x_{k_1}/x_{\alpha_1} + d_{\alpha_1 k_1}, \ldots, x_{k_j}/x_\rho + d_{\rho k_j}, \ldots, x_{k_\iota}/x_{\alpha_\iota} + d_{\alpha_\iota k_\iota}]} \vee$
$x_{\beta_1} - x_{\alpha_1} \neq d_{\alpha_1 k_1} + d_{k_1\beta_1}\ \vee\ \ldots\ \vee\ x_\nu - x_\rho \neq d_{\rho k_j} + d_{k_j\nu}\ \vee\ \ldots$
$\vee\ x_{\beta_\iota} - x_{\alpha_\iota} \neq d_{\alpha_\iota k_\iota} + d_{k_\iota\beta_\iota}$.

Let us recall that $x_\nu \neq x^0_\rho + r^\eta_{\nu\rho}$ has been generated by c_1. This implies that $x^0_\rho + r^\eta_{\nu\rho} = x^0_\rho + d_{\rho k_j} + d_{k_j\nu}$. We can now conclude, using A.7, that

$$x^0_\mu - x^0_\rho = d_{\nu\mu} + r^\eta_{\nu\rho} = d_{\rho k_j} + d_{k_j\nu} + d_{\nu\mu} \qquad \text{(A.8)}$$

Now we have to consider the following cases:

$d_{k_j\mu} = d_{k_j\nu} + d_{\nu\mu}$. If

$$c = c_3, i = \iota, m_1 = \beta_1, \ldots, m_j = \mu, \ldots, m_\iota = \beta_\iota,\ l_1 = \alpha_1, \ldots, l_j = \rho, \ldots, l_\iota = \alpha_\iota$$

then Step 2 of GConsistency adds the following constraint c_4 to C_d:
$\overline{c_3[x_{k_1}/x_{\alpha_1} + d_{\alpha_1 k_1}, \ldots, x_{k_j}/x_\rho + d_{\rho k_j}, \ldots, x_{k_\iota}/x_{\alpha_\iota} + d_{\alpha_\iota k_\iota}]} \vee$
$x_{\beta_1} - x_{\alpha_1} \neq d_{\alpha_1 k_1} + d_{k_1\beta_1}\ \vee\ \ldots\ \vee\ x_\mu - x_\rho \neq d_{\rho k_j} + d_{k_j\mu}\ \vee\ \ldots$
$x_{\beta_\iota} - x_{\alpha_\iota} \neq d_{\alpha_\iota k_\iota} + d_{k_\iota\beta_\iota}$.

The equalities A.8 and the form of c_1 and c_4 imply that we have arrived at a contradiction.

$d_{k_j\mu} < d_{k_j\nu} + d_{\nu\mu}$. In this case

$$x^0_\mu - x^0_\rho \leq d_{\rho\mu} \leq d_{\rho k_j} + d_{k_j\mu} < d_{\rho k_j} + d_{k_j\nu} + d_{\nu\mu}.$$

Thus we have a contradiction with A.8.

The symmetric cases where ν is one of $m_1, \ldots, m_{j-1}, m_{j+1}, \ldots, m_\iota$ can be treated similarly.

(b) $i = \iota, m_1 = \beta_1, \ldots, m_j = \rho, \ldots, m_\iota = \beta_\iota,\ l_1 = \alpha_1, \ldots, l_j = \nu, \ldots, l_\iota = \alpha_\iota$.

This case and the symmetric ones where ν is one of $l_1, \ldots, l_{j-1}, l_{j+1}, \ldots, l_\iota$ are analogous to (a). □