FI MU

Faculty of Informatics
Masaryk University

Constraints with Variables' Annotations

by

Hana Rudová

FI MU Report Series **FIMU-RS-98-04**

 May 1998

Constraints with Variables' Annotations

Hana Rudová
e-mail: hanka@fi.muni.cz

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic

May 11, 1998

Abstract

Variables' annotations in over-constrained problems are proposed and described in order to express preferences for optimal solution selection using preferences on variables. The basic interpretation of variables' annotations is presented and correspondence with hierarchical CSP and possibilistic CSP is described. An algorithm for solving systems of constraints with variables' annotations using mappings to hierarchy is designed. The potential application areas are also mentioned.

1 Introduction

Over-constrained problems are usually solved by giving some preferences or weights to individual constraints and defining the solution as such a valuation which minimizes the violations of constraints. There are, however, over-constrained problems with partially or even completely ordered variables. Assigning preferences to variables could be more natural than defining preferences for constraints artificially. We propose a new constraint solving environment where preferences (or annotations) are assigned to individual variables instead of to the constraints themselves [Rud98]. Moreover, the annotations are local to variable occurrences, i.e., any variable may have different annotations in different constraints (in fact, even different occurrences in the same constraint are allowed).

The most prominent example is the timetabling problem. The problem is stated in variables which represent teachers (dean, professors, assistants...), rooms (more and less occupied) and different groups of students. All these variables have their own preferences, which could be applied directly instead of creating unnatural preferences over individual constraints.

Another example is the problem of the search for the optimal sequence of aircraft departures from a runway. Because practically all the constraints are safety regulations, we can only change aircraft allocation to different time slots. In an over-constrained situation the only allowed action is the removal of an aircraft from further consideration. We can assign preferences to individual variables (planes) and define such a comparator which simply abandons the aircraft with the smallest preferences.

The small timetabling example illustrates a possible interpretation of variables' annotations in constraints over natural numbers. Also, we show by this example that assigning preferences to variables could be more natural than defining preferences for constraints. There is a lecture `L` and its practice `P`. The practice should be preferably taught at least one day after the lecture. We would like to express by the following constraint that the lecture (taught by a professor) is more preferred than the practice (taught by an assistant):

```
L@strong + 1 #=< P@medium  % c1
```

There are two weaker constraints: the lecture has to be taught on Thursday or Friday and the practice from Monday to Thursday:

```
L@weak in 4..5             % c2
P@weak in 1..4             % c3
```

These constraints form a kind of hierarchy: the constraint `c1` with the highest preferences must be satisfied first and then we may try to satisfy constraints `c2` and `c3`. It is possible to satisfy `c1` but not `c2` and `c3` taken together. The constraint `c2` influences the variable with higher annotations (look at `c1`), so this constraint is also satisfied. Then, trying to minimize the overall constraint violation, we get (a kind of) optimal solution `L=4, P=5`. By classical hierarchy where `c1` is annotated by `strong` or `medium`, the solution `L=3, P=4` is also obtained. But this solution is not optimal from our point of view. The different requirements towards the lecture and practice must be stated by assigning different preferences to `c2` and `c3` and so these constraints must be ordered. But this could be wrong with respect to other constraints in a more complex problem. Also, the exact location of the two appropriate constraints need not be easy to find in this context.

2 The Annotations' Properties

A constraint system with variables' annotations is composed from a set of variables V, a set of constraints C, and from the finite domain of variables D. A solution of this system is a valuation $\theta : V \to D$. An error function $e(c\theta)$ indicates how nearly constraint c is satisfied for a valuation θ. This error function can be trivial ($e(c\theta) = 0/1$ means c is satisfied / unsatisfied) or we can define the error function by using the domain's metric. A function a determines the variable annotation in constraint $a : C \times V \to \mathcal{A}$. The system is defined by:

- $\mathcal{A}$ as a set of annotations;
- $\preceq$ as an ordering on $\mathcal{A}$,
  if $a, b \in \mathcal{A}$ then $a \preceq b$ means a is more preferred annotation than b;
- $\underset{a_i \in A}{\circledast} a_i$ (finite $A \subset \mathcal{A}$) as a function computing global annotation,
  it is defined by applying either $\circledast : \mathcal{A}^k \to \mathcal{A}$ or a commutative and associative closed binary operation $*$ on $\mathcal{A}$, which has an extension $\circledast$.

These primary definitions induce:

- $\leq_c$ as an ordering of constraints, [1]
  if $c, d \in C$ then $c \leq_c d$ means c is more preferred than d;
- a method for the selection of optimal solution θ.

And now we describe some auxiliary definitions:

- $var(c)$ is a set of variables of constraint c, $var(c) \subseteq V$;
- global variable annotation $av : V \to \mathcal{A}$, $av(v) = \underset{\{c \in C \mid v \in var(c)\}}{\circledast} a(c, v)$;
- constraint annotation $ac : C \to \mathcal{A}$, $ac(c) = \underset{\{v \in var(c)\}}{\circledast} a(c, v)$;
- global constraint annotation $acv : C \to \mathcal{A}$, $acv(c) = \underset{\{v \in var(c)\}}{\circledast} av(v)$.

[1] Sect. 3.2 gives an example of how the primary definitions induce an ordering of constraints.

3 Instances of the Framework

Different instances of the framework may be obtained by determining the mentioned properties in a similar way to Semiring-based CSP [BMR97a, BMR97b] and Valued CSP [SFV95] (both of these frameworks are introduced and compared in [BFM+96]). In the following, we describe a mapping of annotations to the possibilistic CSP [DFP96, Sch92] and to the hierarchical CSP [BFBW92, WB93]. These mappings may be used as examples of possible semantics of variables' annotations.

3.1 Possibilistic System

The interpretation of annotations through possibilistic CSP emphasizes global variables' annotations whereas the importance of particular constraint's annotation is deferred.

The specifications of the general definitions for this mapping are $\mathcal{A} = \langle 0, 1\rangle$, ordering $\geq$ over real numbers as $\preceq$, and the geometric average over real numbers as $\circledast$. The value 0 is not a member of $\mathcal{A}$ because a variable with such annotation plays no role in the constraint system.

Every solution θ has the error $E(C\theta) = \max_{\{c \in C\}} acv(c)e(c\theta)$. The optimal solution is the solution with minimal error. Preferences for the trivial error function $e(c\theta)$ are expressed only by the global constraint annotation. When the metric error function is used the situation is very different. Then, the preferences of constraints are changed with respect to the chosen valuation θ and a sufficiently great value of error function can change the solution drastically. This combination of the metric error function and global constraint annotation can be used only when the metric error function is normalized.

Let us consider the example with constraints $C = \{c_1, c_2\}$ and two valuations θ_0, θ_1 and describe the selection of a better valuation with a metric comparator. We suppose global constraints' annotations:

$$acv(c_1) = 0.9,\ acv(c_2) = 0.1$$

and value for error function (in this case, the normalization is done by dividing by maximal expectable value of the error function):

$$e(c_1\theta_0) = 0/1000 \quad e(c_2\theta_0) = 10/1000$$
$$e(c_1\theta_1) = 1/1000 \quad e(c_2\theta_1) = 9/1000$$

The error $E(C\theta_i)$ for both valuations are

$$\begin{aligned} E(C\theta_0) &= \max\{acv(c_1) \times e(c_1\theta_0),\ acv(c_2) \times e(c_2\theta_0)\} = \\ &\quad \max\{0.9 \times 0,\ 0.1 \times 0.01\} = 0.001 \\ E(C\theta_1) &= \max\{acv(c_1) \times e(c_1\theta_1),\ acv(c_2) \times e(c_2\theta_1)\} = \\ &\quad \max\{0.9 \times 0.001,\ 0.1 \times 0.009\} = 0.0009 \end{aligned}$$

and the better valuation is θ_1 with minimal error 0.0009. This valuation violates the constraint c_1 with the highest global constraint's annotation but the great value of $e(c_2\theta_0)$ causes this selection. But the trivial comparator chooses as a solution the valuation θ_0, because the valuation θ_0 does not violate the strong constraint c_1 and the combination with a value of the error function is trivial.

The great differences was seen between the metric and trivial approach. The metric error function should be used when the decrease of large values of the error function is desirable. The trivial error function is advantageous when the meaning of annotations is not to be influenced by the shape of the error function so the size of error (any greater than zero) is not too important.

3.2 Hierarchy

An opposite view of variables' annotation demonstrates the interpretation using hierarchical CSP. The hierarchy is constructed over constraint annotations av, with additional order imposed by global constraint annotations acv within each level.

The global definitions are specialized to $\mathcal{A} = \langle 0, 1\rangle$, ordering $\geq$ over real numbers ($\preceq$), and the geometric average over real numbers ($\circledast$). In this case, the basic properties are the same as in the possibilistic approach. The main difference between instances appears in the definition of the method for selecting optimal solution. In the hierarchical approach this method applies the definition of constraint annotation: for $c, d \in C$ such that $c \leq_c d$ holds, the proposition $ac(c) \geq ac(d)$ is implied. The next part describes this method for global and local comparators, respectively.

The hierarchy of constraints C is a union of disjunctive sets C_i where preferences of constraints decrease with increasing value of i:

$$C = C_0 \cup C_1 \cup \ldots \cup C_n \text{ and for } i \in 0 \ldots n$$
$$C_i = \{c \in C | \, (\forall d \in C_j, j < i : d <_c c) \wedge (\forall e \in C_k, i < k : c <_c e)\} \text{ holds.}$$

The level C_0 is a set of required constraints. The annotation of every variable in constraint at this level is equal to 1. The valuation θ has an error

$$E(C\theta) = [E(C_0\theta), E(C_1\theta), \ldots, E(C_n\theta)]$$
$$\text{where } E(C_i\theta) = \sum_{\{c \in C_i\}} acv(c)\, e(c\theta).$$

Therefore the value $acv(c)$ is understood as a weight of constraint c. The optimal solution θ has minimal error $E(C\theta)$ compared by weighted-sum-better comparator as in classical hierarchical CSP. The normalization of the error function had to be used in possibilistic approach with respect to main properties of possibility theory. There is no reason for the normalization in the hierarchical approach.

In a similar way, worst-case-better and least-squares-better comparators can be applied:

$$\begin{array}{rcl} \text{worst-case-better:} \quad E(C_i\theta) & = & \max_{\{c \in C_i\}} acv(c) e(c\theta) \\ \text{least-squares-better:} \quad E(C_i\theta) & = & \sum_{\{c \in C_i\}} acv(c)\, e(c\theta)^2. \end{array}$$

The described mapping applies global comparators for the selection of a better solution. The complexity of global comparators is very high and standard local comparators are not very suitable for efficient mapping of variables' annotations to constraint hierarchy. Therefore a new local comparator is proposed, which uses an ordering of constraints at every level. This ordering is then defined through global constraint annotation.

A valuation θ is ordered-better than another valuation δ if, for each of the constraints through some level $k-1$, the error after applying θ is equal to that after applying δ, and at the level k the errors are compared with respect to an ordering $\leq_w$ of a set W given by a function $w : C \to W$ (proposition $w(c) \leq_w w(d)$ means c is preferred constraint over d):

$$\begin{array}{l} \textit{ordered-better}(\theta, \delta, C) \equiv \\ \quad \exists k \in 1 \ldots n \text{ such that} \\ \quad\quad \forall l \in 1 \ldots k-1\ \forall c \in C_l :\ e(c\theta) = e(c\delta) \\ \quad\quad \wedge\ \exists c \in C_k :\ e(c\theta) < e(c\delta) \\ \quad\quad \wedge\ \forall d \in C_k \text{ such that } w(d) \leq_w w(c) :\ e(d\theta) \leq e(d\delta). \end{array}$$

All constraints at level C_0 have to be satisfied and therefore we may restrict ourselves to levels $1 \ldots n$ only. We can choose trivial error function e or metric function, and then we get ordered-predicate-better or ordered-metric-better comparators, respectively. The valuation θ is *ordered-better* if no valuation ω ordered-better than θ exists.

Now we can define the mapping of constraints with variables' annotations to constraint hierarchy with local comparator. The hierarchy is constructed using constraints' annotations as above and the ordered-better comparator chooses a better solution. The function w, the set W, and the ordering $\leq_w$ correspond to acv, $\mathcal{A}$, and $\preceq$, respectively.

3.2.1 Theory

We clarify the relations between ordered-better and locally-better [2] comparators and also we study some special properties of ordered-better comparator.

Next we suppose that $C = \{c_1, c_2, \ldots, c_m\}$ is constraint hierarchy with levels $C_0, C_1, \ldots, C_n$, an ordering $\leq_w$, and function w.

Lemma 3.1 *Every ordered-better solution θ of hierarchy C is locally-better.*

Proof. Assume that θ is not a locally-better solution and let ω be locally-better than θ. Next let $c_\omega \in C_k$ be the first constraint where ω and θ have different errors $e(c_\omega\theta), e(c_\omega\omega)$ and because the valuation ω is locally-better than valuation θ:

$$e(c_\omega\omega) < e(c_\omega\theta) \tag{1}$$

C_k is the first level where any error functions differ, and so the next proposition follows from the ordered-better comparator definition:

$$\begin{aligned} &\exists c_\theta \in C_k \text{ such that } \forall d \in C_k \text{ such that} \\ &\quad w(d) \leq_w w(c_\theta) : \; e(d\theta) \leq e(d\omega) \wedge e(c_\theta\theta) < e(c_\theta\omega). \end{aligned} \tag{2}$$

This proposition holds for every d and so it holds for c_ω too. There are two possibilities:

1. $w(c_\omega) \leq_w w(c_\theta)$ implies $e(c_\omega\theta) \leq e(c_\omega\omega)$ (2) but this conflicts with proposition (1);

2. $w(c_\omega) >_w w(c_\theta)$ and at the same time we know $e(c_\theta\theta) < e(c_\theta\omega)$ (2) which is contradictory to locally-better property of ω solution.

[2] Locally-better comparator does not consider any ordering. We recall its definition briefly (in detail [BFBW92, WB93]): *locally-better*$(\theta, \delta, C) \equiv \exists k \in 1 \ldots n$ such that $(\forall l \in 1 \ldots k-1 : \; (\forall c \in C_l : \; e(c\theta) = e(c\delta))) \wedge \; (\exists c \in C_k : \; e(c\theta) < e(c\delta)) \wedge \; (\forall d \in C_k : \; e(d\theta) \leq e(d\delta))$.

This means that no locally-better solution ω exists and valuation θ has to be the locally-better solution. $\square$

There are locally-better solutions which are not ordered-better. For example, let us consider hierarchy $C = C_1 = \{c, d\}$ where $w(c) <_w w(d)$ holds. Let there exist solutions ω and θ such that $e(c\omega) > e(c\theta)$, $e(d\omega) < e(d\theta)$. Both solutions could be locally-better but only θ could be ordered-better because it is ordered-better than ω.

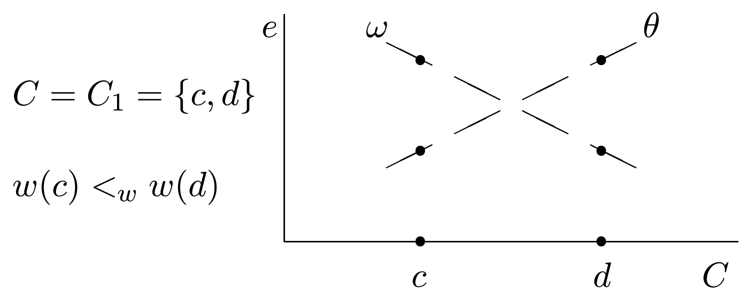

Figure 1: The valuation ω could be locally-better but not ordered-better.

Definition 3.2 A sequence $SC = \langle c_1, \ldots, c_m \rangle$ is *hierarchy-ordering* of hierarchy $C = \{c_1, \ldots, c_m\}$ if all constraints of SC are sorted by the level of hierarchy ($c_i \in C_k$, $c_j \in C_l$, $k < l$ implies $i < j$) and by the ordering $\leq_w$ (for $c_i, c_j \in C_k$ such that $w(c_i) <_w w(c_j)$ implies $i < j$). A sequence $\langle c_1, c_2, \ldots, c_i \rangle$ is denoted SC_i for $i \leq m$.

Definition 3.3 Let $SC = \langle c_1, c_2, \ldots, c_m \rangle$ be a hierarchy-ordering of hierarchy C. Recursively defined set $S = S_m$ is denoted *ordering-solution-set* of hierarchy-ordering SC if

$$
\begin{aligned}
S_0 &= \{\theta \,|\, \theta \text{ is a valuation of } SC\} \\
S_i &= \{\theta \,|\, \theta \in S_{i-1} \wedge e(c_i\theta) = min_{\omega \in S_{i-1}} e(c_i\omega)\} \quad \text{for } i \in 1 \ldots m
\end{aligned}
$$

holds.

Lemma 3.4 *Let us consider constraint hierarchy C. If $w(d) \neq_w w(f)$ holds for every two constraints $d, f \in C_k$ for all $k \in 1 \ldots n$, then a value of error function $e(c\theta)$ is determined for every constraint $c \in C$ and for every ordered-better solution θ uniquely.*

Proof. There is only one hierarchy-ordering SC of such hierarchy C. We show that the set S_i from definition 3.3 is the set of all ordered-better solution of SC_i for every $i \in 1 \dots m$. So the value of $e(c_i\theta)$ is uniquely determined for every i.

The proof is by induction on i. The base case $i = 0$ is trivial because SC_0 is empty and S_0 is the set of all hierarchy's valuation.

Suppose that the proposition holds for SC_{i-1} and now consider the solution of SC_i. Constraint c_i belongs to a higher level of hierarchy than $c_j (j < i)$ or $w(c_j) <_w w(c_i)$ holds which entails $S_i \subseteq S_{i-1}$. Let the value $e(c_i\omega), \omega \in S_i$ be not minimal, then $e(c_i\omega) > e(c_i\theta)$ holds for some ordered-better valuation θ. Every ordered-better solution of SC_{i-1} is the member of S_{i-1} and so $e(c_j\omega) = e(c_j\theta)$ holds for every $j < i$. We obtain the result that the valuation θ is ordered-better than ω. So ω is not ordered-better and can not be a member of S_i. □

Theorem 3.5 *Let SC be a hierarchy-ordering of C and S be the ordering-solution-set of SC. Then S is the set of ordered-better solutions.*

Proof. The proof is by induction on the number of constraints m. The base case is for $m = 1$. The hierarchy is $C = \{c_1\}$ and only one $SC = \langle c_1 \rangle$ exists. We obtain $S = S_1 = \{\theta \,|\, \forall\omega : e(c_1\theta) \leq e(c_1\omega)\}$ and so every valuation $\theta \in S$ is an ordered-better solution.

Suppose that the proposition holds for a hierarchy with m constraints and now show the case with $m + 1$ constraint. Let us suppose $\theta \in S_{m+1}$, $SC = SC_{m+1} = \langle c_1, \dots, c_{m+1} \rangle$ and show for every valuation δ that either θ is ordered-better than δ or δ is not ordered-better than θ.

1. $\delta \notin S_{m+1} \wedge \delta \in S_m$:
   The error function for every c_i $(i \in 1 \dots m)$ is defined uniquely which follows from the assumption $\delta \in S_m$ and the definition of ordering-solution-set. Inequality $e(c_{m+1}\theta) < e(c_{m+1}\delta)$ is implied from the assumptions $\delta \notin S_{m+1}$ and minimal value for c_{m+1}'s error function. Together both these properties induce that θ is ordered-better than δ.

2. $\delta \in S_{m+1}$:
   The error function for every constraint is the same again, so no constraint c_i $(i \in 1 \dots m + 1)$ exists such that $e(c\theta) > e(c\delta)$ (or <) and neither δ nor θ is ordered-better than second valuation for SC_{m+1}.

3. $\delta \notin S_m$:
   $\theta \in S_m$ and so δ can not be ordered-better than θ for SC_m from induction's assumptions. We show that the adding of c_{m+1} does not

change this situation for SC_{m+1}. The value of error function for some $i \in 1 \dots m$ differs for θ and δ (from $\delta \notin S_m$). Let i be the first of them and suppose $c_i \in C_k$ and $e(c_i\theta) < e(c_i\delta)$ (by analogy for >). θ and δ are not comparable for SC_m and so some $c_j \in C_k$ has to exist such that $w(c_j) \leq_w w(c_i)$ and $e(c_j\theta) > e(c_j\delta)$. The proposition $w(c_j) =_w w(c_i)$ holds because i is the smallest index ($j > i$) and SC_m is hierarchy-ordering. These differences induce incomparability for SC_{m+1} too.

Therefore every $\theta \in S$ is an ordered-better solution. □

There are ordered-better solutions which are not obtained using any hierarchy-ordering as its ordering-solution-set. Let us consider the example

```
c1: B >= 10
c2: B =< 8
```

where $C = C_1 = \{\texttt{c1}, \texttt{c2}\}$ and $w(\texttt{c1}) = w(\texttt{c2})$. The valuation $\{\texttt{B} = 10\}$ is obtained for hierarchy-ordering $\langle\texttt{c1}, \texttt{c2}\rangle$ and $\{\texttt{B} = 8\}$ for $\langle\texttt{c2}, \texttt{c1}\rangle$. Both valuations are ordered-better but for example a valuation $\{\texttt{B} = 9\}$ is ordered-better, too.

4 Example

Let us consider that the example from the Introduction is solved by a hierarchy (see Sect. 3.2) with weighted-sum-metric-better comparator.

```
L@3/4 + 1 #=< P@1/2   % c1
L@1/4 in 4..5         % c2
P@1/4 in 1..4         % c3
```

Constraints' annotations are

$$ac(\texttt{c1}) = \sqrt[2]{3/4 \times 1/2}, \quad ac(\texttt{c2}) = \sqrt[1]{1/4}, \quad ac(\texttt{c3}) = \sqrt[1]{1/4}$$

and so the hierarchy is $C = [\{\texttt{c1}\}, \{\texttt{c2}, \texttt{c3}\}]$. Global variables' annotations, which we need for computing global constraints' annotations, are

$$av(\texttt{L}) = \sqrt[2]{3/4 \times 1/4} \doteq 0.43 \quad av(\texttt{P}) = \sqrt[2]{1/2 \times 1/4} \doteq 0.35$$

Global constraints' annotations represent the weight of constraint

$$acv(\texttt{c1}) = \sqrt[2]{av(\mathsf{L}) \times av(\mathsf{P})} \doteq 0.39,$$
$$acv(\texttt{c2}) = \sqrt[1]{av(\mathsf{L})} \doteq 0.43,$$
$$acv(\texttt{c3}) = \sqrt[1]{av(\mathsf{P})} \doteq 0.35.$$

Constraint c1 from the first level can be satisfied but c2 and c3 taken together can not. With respect to the smaller weight of c3, we get optimal solution θ=$\{[\mathsf{L}, 4], [\mathsf{P}, 5]\}$ with error

$$\begin{aligned} E(C\theta) &= [0, (acv(\texttt{c2}) \times e(\texttt{c2}\,\theta) + acv(\texttt{c3}) \times e(\texttt{c3}\,\theta))] = \\ &\quad [0, (acv(\texttt{c2}) \times 0 + acv(\texttt{c3}) \times 1)] \doteq [0, 0.35]. \end{aligned}$$

Let us consider a change in P annotation in c3 to $1/2$. Three levels of hierarchy $C = [\{\texttt{c1}\}, \{\texttt{c3}\}, \{\texttt{c2}\}]$ arise as a consequence of $ac(\texttt{c3})$ increase. The values $av(\mathsf{P}) = \sqrt[2]{1/2 \times 1/2} = 1/2$ and $acv(\texttt{c3})$=$1/2$ increase, too. Optimal solution $\theta = \{[\mathsf{L}, 3], [\mathsf{P}, 4]\}$ has error $E(C\theta) \doteq [0, 0, 0.43]$. This value of error reflects that we violate a more important constraint than in the example above.

Possibilistic system with trivial error function $e(c\theta)$ gives for the original example solution $\theta = \{[\mathsf{L}, 4], [\mathsf{P}, 5]\}$ with unsatisfied constraint c3 and error

$$\begin{aligned} E(C\theta) &= \max\{acv(\texttt{c1}) \times e(\texttt{c1}\theta),\ acv(\texttt{c2}) \times e(\texttt{c2}\theta),\ acv(\texttt{c3}) \times e(\texttt{c3}\theta)\} \\ &\doteq \max\{0.39 \times 0,\ 0.43 \times 0,\ 0.35 \times 1\} = 0.35. \end{aligned}$$

In possibilistic system with metric error function, the normalization can be done by dividing (count_of_days - 1), which is maximal possible value of error function. The solution is the same as above.

5 Algorithm for Solving the Hierarchy

The mappings from variables' annotations to the hierarchy of constraint was described in Sect. 3.2. The ordered-better comparator was proposed as a method to solve the hierarchy efficiently using local comparator only. So, we can apply a local propagation algorithm for solving the hierarchy of constraints which could be easily adapted to ordered-better comparator. These requirements satisfy the Indigo algorithm [BAFB96a, BAFB96b] which efficiently manipulates the acyclic set of inequality constraints.

The key idea in Indigo is that lower and upper bounds on variables (i.e. intervals) are propagated, and the constraints are processed from strongest to weakest, tightening the bounds on variables using interval arithmetic [Ben95] step by step.

The whole solution is divided into three parts:

1. the splitting set of constraints C with variables' annotations to constraint hierarchy $\{C_0, C_1, \ldots, C_n\}$ using constraint annotation ac and ordering $\leq_c$;

2. sorting constraints in every level C_i of hierarchy using global constraint annotation acv to an output sequence of constraints OC_i;

3. the application of the Indigo algorithm with sorted input constraints by the sequence $\langle OC_0, OC_1, \ldots, OC_n \rangle$.

Theorem 5.1 *Given an acyclic set of constraints, the algorithm computes ordered-metric-better solution.*

Proof. Input constraints for the Indigo algorithm define hierarchy-ordering SC using $OC_0, OC_1, \ldots, OC_n$. The Indigo algorithm minimizes error function in the order given by hierarchy-ordering SC. Those are requirements of Theorem 3.5 and so we obtain an ordered-metric-better solution as a result of the algorithm. □

This algorithm for solving inequality constraints with variables' annotation mapped to the hierarchy with the ordered-better comparator was implemented in Prolog with attributed variables and mutable terms.

6 Conclusions and Future Work

We proposed a new approach for solving over-constrained problems using variables' annotations. This approach could be suitable for application areas like planning or scheduling. We defined the mappings from variables' annotations to existing over-constrained systems which allows us to express and make sense of the semantics of variables' annotations in the background of existing systems. We presented mappings of variables' annotations to possibilistic CSP and hierarchical CSP and we proposed a new local comparator for the efficient solving of problems with variables' annotations. Also, the algorithm for solving constraints with variables' annotation was designed and implemented in Prolog.

The future work will consists of the precise interpretation of constraints with variables' annotations which would give us a better understanding of the potential for expressivity of variables' preferences. We will also consider the properties and scope of such interpretation with respect to real problems.

Acknowledgements

This work is supported by the Universities Development Fund of the Czech Republic under the contract # 0748. I would like to thank my supervisor, Luděk Matyska, for exhaustive discussion on topic and reading drafts of this paper.

References

[BAFB96a] Alan Borning, Richard Anderson, and Bjorn Freeman-Benson. Indigo: A local propagation algorithm for inequality constraints. In *Proceedings of the 1996 ACM Symposium on User Interface Software and Technology*, pages 129–136, 1996.

[BAFB96b] Alan Borning, Richard Anderson, and Bjorn Freeman-Benson. The Indigo algorithm. Technical Report TR-96-05-01, Department of Computer Science and Engineering University of Washington, July 1996.

[Ben95] Frédéric Benhamou. Interval constraint logic programming. In Andreas Podelski, editor, *Constraints Programming: Basics and Trends*, pages 1–21. Springer-Verlag LNCS 910, 1995.

[BFBW92] Alan Borning, Bjorn Freeman-Benson, and Molly Wilson. Constraint hierarchies. *Lisp and Symbolic Computation*, 5(3):223–270, 1992.

[BFM+96] Stefano Bistarelli, Hélène Fargier, Ugo Montanari, Francesca Rossi, Thomas Schiex, and Gérard Verfaillie. Semiring-based CSPs and valued CSPs: Basic properties and comparison. In Michael Jampel, Eugene Freuder, and Michael Maher, editors, *Over-Constrained Systems*, pages 111–150. Springer-Verlag LNCS 1106, August 1996.

[BMR97a] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint solving and optimization. *Journal of ACM*, 44(2):201–236, March 1997.

[BMR97b] Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint logic programming. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1997.

[DFP96] Didier Dubois, Hélène Fargier, and Henri Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996.

[Rud98] Hana Rudová. Constraints with variables' annotations. In Henri Prade, editor, *13th European Conference on Artificial Intelligence*, pages 261–262. John Wiley & Sons, Ltd., 1998. In press.

[Sch92] Thomas Schiex. Possibilistic constraint satisfaction problems or "How to handle soft constraints ?". In *8^{th} International Conference on Uncertainty in Artificial Intelligence*, pages 268–275, Stanford, CA, July 1992.

[SFV95] Thomas Schiex, Hélène Fargier, and Gérard Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 631–639, San Mateo, August 1995. Morgan Kaufmann.

[WB93] Molly Wilson and Alan Borning. Hierarchical constraint logic programming. *Journal of Logic Programming*, 16(3,4):227–318, 1993.

Publications in the FI MU Report Series are in general accessible via WWW and anonymous FTP:

`http://www.fi.muni.cz/informatics/reports/`
`ftp  ftp.fi.muni.cz (cd pub/reports)`

Copies may be also obtained by contacting:

Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic