Learning Preferences on Temporal Constraints: A Preliminary Report

F. Rossi
University of Padova, Italy
frossi@math.unipd.it

A. Sperduti
University of Pisa, Italy
perso@di.unipi.it

L. Khatib(*), P. Morris, R. Morris
(*) Kestrel Technology
NASA Ames Research Center, Moffett Field, CA, USA
{lina,pmorris,morris}@ptolemy.arc.nasa.gov

Abstract

A number of reasoning problems involving the manipulation of temporal information can naturally be viewed as implicitly inducing an ordering of potential local decisions involving time (specifically, associated with durations or orderings of events) on the basis of preferences. For example, a pair of events might be constrained to occur in a certain order, and, in addition, it might be preferable that the delay between the start times of each of them be as large, or as small, as possible.

Sometimes, however, it is more natural to view preferences as something initially ascribed to complete solutions to temporal reasoning problems, rather than to local decisions. For example, in classical scheduling problems, the preference for solutions which minimize makespan is a global, rather than a local, condition. In such cases, it might be useful to learn *the local preferences that contribute to globally preferred solutions. This information could be used in heuristics to guide the solver to more promising solutions.*

To address the potential requirement for information about local preferences, we propose to apply learning techniques to infer local preferences from global ones. The preliminary work in this paper proposes an approach based on the notion of learning a set of soft temporal constraints, given a training set of solutions to a Temporal CSP, and an objective function for evaluating each solution in the set.

1. Introduction and motivation

Several real world problems involving the manipulation of temporal information in order to find an assignment of times to a set of activities or events can naturally be viewed as having preferences associated with local temporal decisions, where by a local temporal decision we mean one associated with how long a single activity should last, when it should occur, or how it should be ordered with respect to other activities. For example, an antenna on an earth orbiting satellite such as Landsat 7 must be slewed so that it is pointing at a ground station in order for recorded science or telemetry data to be downlinked to earth. Assume that as part of the daily Landsat 7 scheduling activity a window $W = [s, e]$ is identified within which a slewing activity to one of the ground stations for one of the antennae can begin, and thus there are choices for assigning the start time for this activity. Antenna slewing on Landsat 7 has been shown to occasionally cause a slight vibration to the satellite, which in turn might affect the quality of the image taken by the scanning instrument if the scanner is in use during slewing. Consequently, it is preferable for the slewing activity not to overlap any scanning activity, although because the detrimental effect on image quality occurs only intermittently, this disjointness is best not expressed as a hard constraint. Thus if there are any start times t within W such that no scanning activity occurs during the slewing activity starting at t, then t is to be preferred. Of course, the cascading effects of the decision to assign t on the scheduling of other satellite activities must be taken into account as well. For example, the selection of t, rather than some earlier start time within W, might result in a smaller overall contact period between the ground station and satellite, which in turn might limit the amount of data that can be downlinked during this period. This may conflict with the preference for attaining maximal contact times with ground stations, if possible.

Reasoning simultaneously with hard temporal constraints and preferences, as illustrated in the example just given, is crucial in many situations. However, in many temporal reasoning problems it is difficult or impossible to specify a local preference on durations. In real world scheduling problems, for example, it is common to impose a global constraint on solutions, for example, to minimize makespan or tardiness, but virtually impossible to specify how specific ordering choices between pairs of events contribute to meeting the overall objective. If such knowledge were at hand, it could be used as heuristics to guide the scheduler to prefer local assignments that were found to yield better solutions.

This paper explores the problem of automatically generating local temporal preference information. We propose to do it via a machine learning approach (for example, based on gradient descent), using a representation of local preferences in terms of soft constraints. This approach has already been successfully used in the context of general soft constraint problems, and here we propose to adapt it to the temporal constraint framework with preferences. This paper is preliminary in nature, proposing a technique and offering some theoretical justification for its adoption, but as yet no empirical evidence for its viability.

The paper is organized as follows. In Section 2 we describe the framework for temporal constraints with preferences, and define the temporal learning problem. In Section 3 we formulate a general solution to the problem, and in Section 4 we discuss a variant of the learning approach which maintains tractability, and in Section 5 we summarize the main issues raised by the paper and we point to directions for future work.

2. Temporal constraint problems with preferences

Temporal constraint reasoning. The Temporal CSP framework (TCSP) [4] has been used widely to solve temporal reasoning problems. This framework is based on knowledge represented as constraints on distances or durations of events. More precisely, variables represent events happening over time, and each constraint gives an allowed range for the distances or durations, expressed as a set of intervals over the time line. For example, a constraint over X and Y could say that $5 \leq Y - X \leq 10$ or $15 \leq Y - X \leq 30$, which is formally represented by the two intervals $[5, 10]$ and $[15, 30]$. Informally, this constraint states that the distance between the two events should be either between 5 and 10, or between 15 and 30. Satisfying such a constraint means choosing any of the allowed distances. A solution for a TCSP consisting of a set of temporal constraints is an assignment of values to its variables such that all constraints are satisfied.

Complexity issues for TCSPs. As expected, general TCSPs are NP-hard. However, TCSPs with just one interval for each constraint, called STPs, are polynomially solvable. In fact, one can transform the given STP into a distance graph, apply to this graph a shortest path algorithm, and then assign to each variable the value corresponding to the shortest distance thus found (see [4] for details).

Hard and soft temporal constraints. Although very expressive, TCSPs are able to model just *hard* temporal constraints. This means that all constraints have to be satisfied, and that the solutions of a constraint are all equally satisfying. As noted in the introduction, in many real-life scenarios these two assumptions do not hold. In particular, sometimes some solutions are preferred with respect to others. Therefore the global problem is not to find a way to satisfy all constraints, but to find a way to satisfy them optimally, according to the preferences specified.

Soft temporal constraint problems. To address such problems, recently [6] a new framework has been proposed, where each temporal constraint is associated with a preference function, which specifies the preference for each distance. This framework is based on a simple merger of TCSPs and soft constraints, where for soft constraints we have taken a general framework based on semirings [2]. The result is a class of problems called Temporal Constraint Satisfaction problems with preferences (TCSPPs).

Preference functions. A *soft temporal constraint* in a TCSPP is represented by a pair consisting of a set of disjoint intervals and a preference function: $\langle I = \{[a_1, b_1], \ldots, [a_n, b_n]\}, f \rangle$, where $f : I^1 \rightarrow A$, is a mapping of the elements of I into preference values, taken from a set A. Examples of common preference functions involving time are:

- **min-delay**: any function in which smaller distances are preferred, that is, the delay of the second event w.r.t. the first one is minimized.
- **max-delay**: any function which assigns higher preference values to larger distances;
- **close to k**: any function which assigns higher values to distances which are closer to k; in this way, we specify that the distance between the events must be as close as possible to k.

Global preference value. A *solution* to a TCSPP is a complete assignment to all the variables that satisfies the distance constraints. Each solution has a *global preference value*, obtained by combining the local preference values

[1] Here by I we mean the set of all elements appearing in the intervals of I.

found in the constraints. To formalize the process of combining local preferences into a global preference, and comparing solutions, we impose a semiring structure onto the TCSPP framework.

Semirings. A *semiring* is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ such that

- A is a set and $\mathbf{0}, \mathbf{1} \in A$;
- $+$, the additive operation, is commutative, associative and $\mathbf{0}$ is its unit element;
- $\times$, the multiplicative operation, is associative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element.

A *c-semiring* is a semiring in which $+$ is idempotent (i.e., $a + a = a, a \in A$), $\mathbf{1}$ is its absorbing element, and $\times$ is commutative. These additional properties (w.r.t. usual semirings) are required to cope with the usual nature of constraints.

C-semirings allow for a partial order relation $\leq_S$ over A to be defined as $a \leq_S b$ iff $a + b = b$. Informally, $\leq_S$ gives us a way to compare tuples of values and constraints, and $a \leq_S b$ can be read *b is better than a*. Moreover, one can prove that:

- $+$ and $\times$ are monotone on $\leq_S$;
- $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum;
- $\langle A, \leq_S\rangle$ is a complete lattice where:
  - for all $a, b \in A$, $a + b = lub(a, b)$ (where *lub*=least upper bound);
  - if $\times$ is idempotent, then $\langle A, \leq_S\rangle$ is a complete distributive lattice and $\times$ is its greatest lower bound (*glb*).

Given a semiring[2] with a set of values A, each preference function f associated with a soft constraint $\langle I, f\rangle$ of a TCSPP takes an element from I and returns an element of A, where A is the carrier of a semiring. This allows us to associate a preference with a duration or distance.

From local to global preferences. The two semiring operations allow for complete solutions to be evaluated in terms of the preference values assigned locally. More precisely, given a solution t in a TCSPP with associated semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$, let $T_{ij} = \langle I_{i,j}, f_{i,j}\rangle$ be a soft constraint over variables X_i, X_j and (v_i, v_j) be the projection of t over the values assigned to variables X_i and X_j (abbreviated as $(v_i, v_j) = t_{\downarrow X_i, X_j}$). Then, the corresponding preference value given by f_{ij} is $f_{ij}(v_j - v_i)$, where $v_j - v_i \in I_{i,j}$. Finally, where $F = \{x_1, \ldots, x_k\}$ is a set, and $\times$ is the multiplicative operator on the semiring, let $\times F$ abbreviate $x_1 \times \ldots \times x_k$. Then the global preference value of t, $val(t)$, is defined to be $val(t) = \times\{f_{ij}(v_j - v_i) \mid (v_i, v_j) = t_{\downarrow X_i, X_j}\}$. The optimal solutions of a TCSPP are those solutions which have the best global preference value, where "best" is determined by the ordering $\leq_S$ of the values in the semiring.

Example: fuzzy temporal constraints. For example, consider the semiring $S_{fuzzy} = \langle [0, 1], max, min, 0, 1\rangle$, used for fuzzy constraint solving [9]. The global preference value of a solution will be the minimum of all the preference values associated with the distances selected by this solution in all constraints, and the best solutions will be those with the maximal value.

Example: hard temporal constraints. Another example is the semiring $S_{csp} = \langle \{false, true\}, \vee, \wedge, false, true\rangle$, which allows to describe hard constraint problems [7]. Here there are only two preference values: *true* and *false*, the preference value of a complete solution will be determined by the logical *and* of all the local preferences, and the best solutions will be those with preference value *true* (since *true* is better than *false* in the order induced by using the logical or as the $+$ operator of the semiring). This semiring thus recasts the classical TCSP framework into a TCSPP.

Simple soft temporal constraints. A special case occurs when each constraint of a TCSPP contains a single interval. We call such problems *Simple Temporal Problems with Preferences* (STPPs), due to the fact that they generalize Simple Temporal Problems (STPs) [4]. This case is interesting because, as noted above, STPs are polynomially solvable, while general TCSPs are NP-hard, and the computational effect of adding preferences to STPs is not immediately obvious. In [6] it has been shown that, while in general TCSPPs are NP-hard, under certain restrictions on the "shape" of the preference functions and on the semiring, STPPs are tractable.

Linear preference functions. For example, when the preference functions of an STPP are linear, and the semiring chosen is such that its two operations maintain such linearity when applied to the initial preference functions, it can be seen that the given STPP can be written as a linear programming problem, which is tractable [3].

Convex preference functions. Linear preference functions are expressive enough for many cases, but there are

[2]For simplicity, from now on we will write *semiring* meaning *c-semiring*.

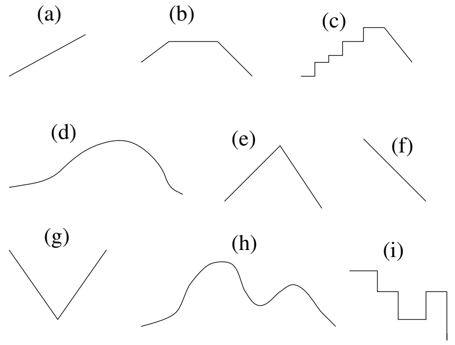

Figure 1. Examples of semi-convex functions (a)-(f) and non-semi-convex functions (g)-(i)

also several situations in which we need preference functions which are not linear. A typical example arises when we want to state that the distance between two variables must be as close as possible to a single value. Then, unless this value is one of the extremes of the interval, the preference function is convex, but not linear.

Step preference functions. Another case is one in which preferred values are as close as possible to a single distance value, but in which there are some subintervals where all values have the same preference. In this case, the preference criteria define a *step function*, which is not convex.

Semi-convex preference functions. A class of functions which includes linear, convex, and also some step functions has been called in [6] *semi-convex*. Semi-convex functions are such that, if one draws a horizontal line anywhere in the Cartesian plane defined by the function, the set of X such that $f(X)$ is not below the line forms an interval. More formally, a *semi-convex* function f is one such that, for all Y, the set $\{X$ such that $f(X) \geq Y\}$ forms an interval. It is easy to see that semi-convex functions include linear ones, as well as convex and some step functions. For example, the *close to k* criteria cannot be coded into a linear preference function, but it can be specified by a semi-convex preference function, which could be $f(x) = x$ for $x \leq k$ and $f(x) = 2k - x$ for $x > k$.

Figure 1 shows some examples of semi-convex and non-semi-convex functions.

Tractability results for STPPs. It is proven in [6] that STPPs with semi-convex preference functions and a semiring with a total order of preference values and an idempotent multiplicative operation can be solved in polynomial time. This result can be interpreted as possibly defining a "maximal" subset of preference functions for which soft temporal reasoning can be performed effectively.

3. Learning soft temporal constraints

As noted in the introduction, it is not always easy to specify the preference functions in each temporal constraint in a way that the real-life problem at hand is faithfully modelled. This happens because sometimes it is easier, or possible, to specify only global preference functions, to be associated to entire solutions, rather than local preference functions to be attached to the constraints. For this reason, and since the whole TCSPP machinery is based on local preference functions, we propose here a method to induce local preferences from global ones. Since STPPs are, with some restrictions, tractable problems, and since the whole point of this paper is to make soft temporal reasoning more practical, we will for now restrict our attention to such problems.

Inductive learning. More precisely, the problem to be considered here can now be formally described as an inductive learning problem [8]. Inductive learning can be defined as the ability of a system to induce the correct structure of a map d which is known only for particular inputs. More formally, defining an example as a pair $(x, d(x))$, the computational task is as follows: *given a collection of examples of d, i.e., the training set, return a function h that approximates d.* Function h is called a hypothesis.

A common approach to inductive learning, especially in the context of neural networks, is to evaluate the quality of a hypothesis h (on the training set) through an *error function* [5]. An example of popular error function, that can be used over the reals, is the sum of squares error [5]: $E = \frac{1}{2}\sum_{i=1}^{n}(d(x_i) - h(x_i))^2$, where $(x_i, d(x_i))$ is the i-th example of the training set.

Given a starting hypothesis h_0, the goal of learning is to minimize the error function E by modifying h_0. This can be done by using a definition of h which depends on a set of internal parameters W, i.e., $h \equiv h_W$, and then adjusting these parameters. This adjustment can be formulated in different ways, depending on whether the domain is isomorphic to the reals or not. The usual way to be used over the reals, and if h_W is continuous and derivable, is to follow the negative of the *gradient* of E with respect to W. This technique is called *gradient descent* [5]. Specifically, the set of parameters W is initialized to small random values at time $\tau = 0$ and updated at time $\tau + 1$ according to the following equation: $W(\tau + 1) = W(\tau) + \Delta W(\tau)$, where $\Delta W(\tau) = -\eta \frac{\partial E}{\partial W(\tau)}$, and η is the step size used for the gradient descent. Learning is stopped when a minimum of E is reached. Note that, in general, there is no guarantee that the found minimum is global.

From an STP to an STPP. Learning in our context can be used to find suitable preference functions to be associated to the constraints of a given STP. More precisely, let $P = (V, C)$ be an STP where

- V is a set of variables with domains consisting of moments of time, and
- C is a set of distance constraints of the form $l \leq X - Y \leq u$, where $X, Y, \in V$ and l, u are time points.

Let f be a function $f : S \rightarrow U$, where

- S is the set of solutions to P and
- U is a set of values indicating the "quality" of the solution.

The learning task consists of transforming the STP into an STPP, with each constraint $c_{i,j} \in C$ replaced by a soft constraint $\langle c_{i,j}, f_{i,j} \rangle$, where $f_{i,j}$ is the local preference function for $c_{i,j}$.

The examples to be used in the learning task consist of pairs $(s, f(s))$, where s is a solution to the original STP and $f(s)$ is its "score". In the following, we use P to denote an STP and P' to denote a corresponding STPP. Also, $val_{P'}(t)$ is used to indicate the value of a solution t over P'.

Semiring choice. Let P and f be as defined above, and suppose a set of examples

$$TR = \{(t_1, r(t_1)), \ldots, (t_m, r(t_m))\}$$

is given. To infer the local preferences, we must also be given the following: a semiring whose element set A contains the values $r(t_i)$ in the examples; and a distance function over such a semiring. For example, if the score values are positive real numbers, we could choose the semiring $\langle \Re^+, min, +, +\infty, 0 \rangle$ and, as distance function, the usual one over reals: $dist(val_{P'}(t), r(t)) = \mid val_{P'}(t) - r(t) \mid$.

Given the above, the goal is to define a corresponding STPP P' over the same semiring such that

1. P and P' have the same set of variables, domains and interval constraints, and
2. for each t such that $(t, r(t))$ is an example, $dist(val_{P'}(t), r(t)) < \epsilon$, where $\epsilon > 0$ and small.

Parameters. If the first condition is true, the only free parameters that can be arbitrarily chosen in order to satisfy the other conditions are the values to be associated to each distance. For each constraint $c_{ij} = \{[a_1, b_1], \ldots, [a_n, b_n]\}$ in P, the idea is to associate, in P', a free parameter w_d, where $d = X_j - X_i$ (note that such a parameter must belong to the set of the chosen semiring), to each element d in $I = \bigcup_i [a_i, b_i]$. This parameter will represent the preference over that specific distance. With the other distances, those outside I, we associate the constant **0**, (the lowest value of the semiring (w.r.t. $\leq_S$)).

If I contains an infinite number of distances, we would need an infinite number of parameters, which would make learning impossible. To avoid this problem, we can restrict the class of preference functions to a subset which can be described via a finite number of parameters. For example, if we use only linear functions, we just need two parameters a and b, since every linear function can be expressed as $a \cdot (X_j - X_i) + b$. In general, we will have a function which depends on a set of parameters W, thus we will denote it as $f_W : (W \times I) \rightarrow A$.

Notice also that the choice of associating parameters only to allowed distances (that is, those in I) implies that we cannot change the preference of non-allowed distances. This means that global instantiations which are not solutions of the original problem will always have the worst rating in the resulting soft problem. To make the framework more flexible, we could also associate parameters to non-allowed distances. In that case, we would need to use parametric preference functions that also range outside I, that is, $f_W : (W \times D) \rightarrow A$, where D is the variable domain.

The value assigned to each solution t in P' is

$$val_{P'}(t) = \prod_{c_{ij} \in P'} [\sum_{d \in \bigcup_{D \in I_{ij}} D} \text{check}(d, t, i, j) \times f_W(d)] \quad (1)$$

where $\prod$ generalizes the $\times$ operation, $\sum$ generalizes +, I_{ij} is the set of intervals associated to constraint c_{ij}, and $\text{check}(d, t, i, j) = \mathbf{1}$ if $d = t \downarrow_{X_j} - t \downarrow_{X_i}$ and **0** otherwise. Note that, for each constraint c_{ij}, there is exactly one distance d such that $\text{check}(d, t, i, j) = \mathbf{1}$, namely $d = t \downarrow_{X_j} - t \downarrow_{X_i}$. Thus, $val_{P'}(t) = \prod_{c_{ij} \in P'} f_W(t \downarrow_{X_j} - t \downarrow_{X_i})$. The values of the free parameters in W may be obtained via a minimization of the error function, which will be defined according to the distance function of the semiring.

Example: over the reals. For example, assume that the desired ratings are real numbers and that the chosen c-semiring is $\langle \Re^+, min, +, +\infty, 0 \rangle$. Here the semiring $\times$ operator, that is, the real number sum operator, is continuous and differentiable. Thus we can use the gradient descent technique. In this case equation (1) becomes

$$val_{P'}(t) = \sum_{c_{ij} \in P'} min_{d \in \bigcup_{D \in I_{ij}} D} (\text{check}(d, t, i, j) + f_W(d))$$
$$= \sum_{c_{ij} \in P'} f_W(t \downarrow_{X_j} - t \downarrow_{X_i})$$

and the error function could be $E = \frac{1}{2} \sum_{i=1}^{m} (r(t_i) - val_{P'}(t_i))^2$. By deriving the error function w.r.t. one parameter $w \in W$, we get $\Delta w(\tau) = -\eta \frac{\partial E}{\partial w(\tau)}$, where τ is time and η is the step size used for the gradient descent. The derivative can be computed by using the chain derivation rule.

Incremental learning. In this learning framework, a user should provide the system with all the examples (that is, rated solutions). Of course, the higher the number of examples, the higher the probability of a successful learning phase. However, it is not feasible to ask the user to provide too many examples. For this reason, in [1] we proposed an incremental strategy which aims at reducing the number of the examples the user has to provide. Using this strategy, the user just gives an initial small set of examples, over which the system performs the first learning phase. Then, the user checks the resulting system on a set of new solutions, and collects those that are mis-rated by the system, which will be given as new examples for the next learning phase. This process iterates until the user is satisfied with the current state of the system. This approach can also be used in the temporal framework we have described in this paper.

4. Learning semi-convex preference functions

Suppose we are given a class of STPs to be "softened" via the learning approach defined above. As we know, STPs are tractable [4]. However, in general we may end up with STPPs which are not tractable, since there is no guarantee that our learning approach returns preference functions which are semi-convex. Moreover, one needs to choose a semiring which preserves semi-convexivity.

To force the learning framework to produce semi-convex functions, we can specialize it for a specific class of functions with this property. For example, we could choose convex quadratic functions of the form $f(d) = a \cdot d^2 + b \cdot d + c$, where $a < 0$. In this case we just have three parameters to consider: $W = \{a, b, c\}$. Another possibility is to incorporate additional constraints that determine semi-convexivity (e.g. for a step-function, these could be linear inequalities) into the objective as in a Lagrange multipliers approach.

Of course, by choosing a specific class of semi-convex functions f_W, not all local preference shapes will be representable. Therefore, there will be cases in which the desired solution ratings, as specified in the training set, cannot be matched. For example, the user could have specified a set of examples which is not consistent with any soft temporal constraint problem using that class of semi-convex preference functions. Even if one chooses f_W to cover any semi-convex function, there is no guarantee that the desired solution ratings will be matched.

In general, the learning process will return a soft temporal problem which will approximate the desired rating as much as possible, considering the chosen class of functions and the error function. But we will gain tractability for the solution process of the resulting problems: starting from the class of STPs, via the learning approach we will obtain a class of STPPs which is tractable as well.

5. Conclusions and future work

This paper has presented an abstract general strategy for inductive learning of local preferences for values representing distances between events in a temporal reasoning problem such as scheduling. This strategy is based on a formalism which merges TCSPs with a calculus of soft constraints based on a semiring structure of preference values. The potential value of such a system would reside in the ability of a reasoning system to learn to detect temporal distance relations that tend to result in good solutions, which in turn could assist a heuristic temporal reasoner to incrementally improve its performance over time as it samples the space of solutions.

We are building a solver for STPPs, which will concentrate, in its first version, on those classes of STPPs which are tractable. We are also developing a learning module, which will be tailored on some classes of semi-convex functions and where learning will be performed via gradient descent and on a real-valued semiring. We plan to experiment the overall system using both randomly generated STPPs and also real-life problem instances such as satellite event scheduling.

References

[1] A. Biso, F. Rossi, and A. Sperduti. Experimental results on learning soft constraints. In *Proc. KR 2000*. Morgan Kaufmann, 2000.

[2] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based Constraint Solving and Optimization. *Journal of the ACM*, 44(2):201–236, March 1997.

[3] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

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

[5] S. Haykin. *Neural Networks: a comprehensive Foundation*. Piscataway, NJ: IEEE Press, 1994.

[6] L. Khatib, P. Morris, R. Morris, and F. Rossi. Temporal constraint reasoning with preferences. In *Proc. IJCAI-01, to appear*, 2001.

[7] A. Mackworth. Constraint satisfaction. In S. C. Shapiro, editor, *Encyclopedia of AI (second edition)*, volume 1, pages 285–293. John Wiley & Sons, 1992.

[8] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[9] T. Schiex. Possibilistic constraint satisfaction problems, or "how to handle soft constraints?". In *Proc. 8th Conf. of Uncertainty in AI*, pages 269–275, 1992.