Temporal Constraint Reasoning With Preferences

Lina Khatib[1,2] **Paul Morris**[2] **Robert Morris**[2]
1. Kestrel Technology
2. Computational Sciences Division
NASA Ames Research Center, MS 269-2
Moffett Field, CA 94035

Francesca Rossi
Dipartimento di Matematica Pura ed Applicata
Universita' di Padova
Via Belzoni 7, 35131 Padova, Italy

Abstract

A number of reasoning problems involving the manipulation of temporal information can be viewed as implicitly inducing an ordering of decisions involving time (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 them be as large, or as small, as possible. This paper explores problems in which a set of temporal constraints is specified, each with preference criteria for making local decisions about the events involved in the constraint. A reasoner must infer a complete solution to the problem such that, to the extent possible, these local preferences are met in the best way. Constraint-based temporal reasoning is generalized to allow for reasoning about temporal preferences, and the complexity of the resulting formalism is examined. While in general such problems are NP-complete, some restrictions on the shape of the preference functions, and on the structure of the set of preference values, can be enforced to achieve tractability. In these cases, a generalization of a single-source shortest path algorithm can be used to compute a globally preferred solution in polynomial time.

1 Introduction and Motivation

Some real world temporal reasoning problems can naturally be viewed as involving preferences associated with decisions such as 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 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 cause a vibration to the satellite, which in turn affects the quality of the observation taken by the imaging instrument if the instrument 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. Rather, the constraint is better expressed as follows: 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 sequencing 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 maintaining maximal contact times with ground stations.

Reasoning simultaneously with hard temporal constraints and preferences, as illustrated in the example just given, is the subject of this paper. The overall objective is to develop a system that will generate solutions to temporal reasoning problems that are *globally preferred* in the sense that the solutions simultaneously meet, to the best extent possible, all the local preference criteria expressed in the problem.

In what follows a formalism is described for reasoning about temporal preferences. This formalism is based on a generalization of the Temporal Constraint Satisfaction Problem (TCSP) framework [Dechter *et al*, 1991], with the addition of a mechanism for specifying preferences, based on the semiring-based soft constraint formalism [Bistarelli *et. al.*, 1997] . The result is a framework for defining problems involving *soft temporal constraints*. The resulting formulation, called Temporal Constraint Satisfaction Problems with Preferences (TCSPPs) is introduced in Section 2. A sub-class of TCSPPs in which each constraint involves only a single interval, called Simple Temporal Problems with Preferences (STPPs), is also defined. In Section 3, we demonstrate the hardness of solving general TCSPPs and STPPs, and pinpoint one source of the hardness to preference functions whose "better" values may form a non-convex set. Restricting the class of admissible preference functions to those with convex intervals of "better" values is consequently shown to result in a tractable framework for solving STPPs. In section 4, an algorithm is introduced, based on a simple generalization of the single source shortest path algorithm, for finding globally best solutions to STPPs with restricted preference functions. In section 5, the work presented here is compared to other approaches and results.

2 Temporal Constraint Problems with Preferences

The proposed framework is based on a simple merger of two existing formalisms: Temporal Constraint Satisfaction Problems (TCSPs) [Dechter *et. al.*, 1991], and soft constraints based on semirings [Bistarelli *et. al.*, 1997] [1]. The result of the merger is a class of problems called Temporal Constraint Satisfaction problems with preferences (TCSPPs). In a TCSPP, a *soft temporal constraint* 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 $f : I \to A$, and A is a set of preference values.

Examples of 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**: assigning higher preference values to larger distances;
- **close to k**: assign higher values to distances which are closer to k; in this way, we specify that the distance between the two events must be as close as possible to k.

As with classical TCSPs, the interval component of a soft temporal constraint depicts restrictions either on the start times of events (in which case they are unary), or on the distance between pairs of distinct events (in which case they are binary). For example, a unary constraint over a variable X representing an event, restricts the domain of X, representing its possible times of occurrence; then the interval constraint is shorthand for $(a_1 \le X \le b_1) \vee \ldots \vee (a_n \le X \le b_n)$. A binary constraint over X and Y, restricts the values of the distance $Y - X$, in which case the constraint can be expressed as $(a_1 \le Y - X \le b_1) \vee \ldots \vee (a_n \le Y - X \le b_n)$. A uniform, binary representation of all the constraints results from introducing a variable X_0 for the *beginning of time*, and recasting unary constraints as binary constraints involving the distance $X - X_0$.

An interesting 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 STPs [Dechter *et. al.*, 1991]. This case is interesting because STPs are polynomially solvable, while general TCSPs are NP-complete, and the effect of adding preferences to STPs is not immediately obvious. The next section discusses these issues in more depth.

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 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.

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.

C-semirings allow for a partial order relation $\le_S$ over A to be defined as $a \le_S b$ iff $a + b = b$. Informally, $\le_S$ gives us a way to compare tuples of values and constraints, and $a \le_S b$ can be read *b is better than a*. Moreover: $+$ and $\times$ are monotone on $\le_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum; $\langle A, \le_S\rangle$ is a complete lattice and, for all $a, b \in A$, $a + b = lub(a, b)$ (where *lub*=least upper bound). If $\times$ is idempotent, then $\langle A, \le_S\rangle$ is a complete distributive lattice and $\times$ is its greatest lower bound (*glb*). In our main results, we will assume $\times$ is idempotent and also restrict $\le_S$ to be a total order on the elements of A. In this case $a + b = \max(a, b)$ and $a \times b = \min(a, b)$.

Given a choice of semiring with a set of values A, each preference function f associated with a soft constraint $\langle I, f\rangle$ takes an element from I and returns an element of A. The 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 preference value, where "best" is determined by the ordering of the values in the semiring. For example, consider the semiring $S_{fuzzy} = \langle [0, 1], max, min, 0, 1\rangle$, used for fuzzy constraint solving [Schiex, 1995]. The 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. Another example is the semiring $S_{csp} = \langle\{false, true\}, \vee, \wedge, false, true\rangle$, which is related to solving classical constraint problems [Mackworth, 1992]. 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

[1] Semiring-based soft constraints is one of a number of formalisms for soft constraints, but it has been shown to generalize many of the others, e.g., [Freuder and Wallace, 1992] and [Schiex *et. al.*, 1995].

than *false* in the order induced by logical or). This semiring thus recasts the classical TCSP framework into a TCSPP.

Given a constraint network, it is often useful to find the corresponding minimal network in which the constraints are as explicit as possible. This task is normally performed by enforcing various levels of local consistency. For TCSPPs, in particular, we can define a notion of *path consistency*. Given two soft constraints, $\langle I_1, f_1\rangle$ and $\langle I_2, f_2\rangle$, and a semiring S, we define:

- the *intersection* of two soft constraints $T_1 = \langle I_1, f_1\rangle$ and $T_2 = \langle I_2, f_2\rangle$, written $T_1 \oplus_S T_2$, as the soft constraint $\langle I_1 \oplus I_2, f\rangle$, where
  - $I_1 \oplus I_2$ returns the pairwise intersection of intervals in I_1 and I_2, and
  - $f(a) = f_1(a) \times f_2(a)$ for all $a \in I_1 \oplus I_2$;
- the *composition* of two soft constraints $T_1 = \langle I_1, f_1\rangle$ and $T_2 = \langle I_2, f_2\rangle$, written $T_1 \otimes_S T_2$, is the soft constraint $T = \langle I_1 \otimes I_2, f\rangle$, where
  - $r \in I_1 \otimes I_2$ if and only if there exists a value $t_1 \in I_1$ and $t_2 \in I_2$ such that $r = t_1 + t_2$, and
  - $f(a) = \sum\{f_1(a_1) \times f_2(a_2) \mid a = a_1 + a_2, a_1 \in I_1, a_2 \in I_2\}$, where $\sum$ is the generalization of $+$ over sets.

A *path-induced* constraint on variables X_i and X_j is $R_{ij}^{path} = \oplus_S \forall k (T_{ik} \otimes T_{kj})$, i.e., the result of performing $\oplus_S$ on each way of composing paths of size two between i and j. A constraint T_{ij} is *path-consistent* if and only if $T_{ij} \subseteq R_{ij}^{path}$, i.e., T_{ij} is at least as strict as R_{ij}^{path}. A TCSPP is path-consistent if and only if all its constraints are path-consistent.

If the multiplicative operation of the semiring is idempotent, then it is easy to prove that applying the operation $T_{ij} := T_{ij} \oplus_S (T_{ik} \otimes_S T_{kj})$ to any constraint T_{ij} of a TCSPP returns an equivalent TCSPP. Moreover, under the same condition, applying this operation to a set of constraints returns a final TCSPP which is always the same independently of the order of application[2]. Thus any TCSPP can be transformed into an equivalent path-consistent TCSPP by applying the operation $T_{ij} := T_{ij} \oplus (T_{ik} \otimes T_{kj})$ to all constraints T_{ij} until no change occurs in any constraint. This algorithm, which we call Path, is proven to be polynomial for TCSPs (that is, TCSPPs with the semiring S_{csp}): its complexity is $O(n^3 R^3)$, where n is the number of variables and R is the range of the constraints [Dechter *et. al.*, 1991].

General TCSPPs over the semiring S_{csp} are NP-complete; thus applying Path is insufficient to solve them. On the other hand, with STPPs over the same semiring that coincide with STPs, applying Path is sufficient to solve them. In the remaining sections, we prove complexity results for both general TCSPPs and STPPs, and also of some subclasses of problems identified by specific semirings, or preference functions with a certain shape.

[2]These properties are trivial extensions of corresponding properties for classical CSPs, proved in [Bistarellli, *et. al.*, 1997.]

3 Solving TCSPPs and STPPs is NP-Complete

As noted above, solving TCSPs is NP-Complete. Since the addition of preference functions can only make the problem of finding the optimal solutions more complex, it is obvious that TCSPPs are at least NP-Complete as well.

We turn our attention to the complexity of general STPPs. We recall that STPs are polynomially solvable, thus one might speculate that the same is true for STPPs. However, it is possible to show that in general, STPPs fall into the class of NP-Complete problems.

Theorem 1 (complexity of STPPs) *General STPPs are NP-complete problems.*

Proof:

First, we prove that STPPs belong to NP. Given an instance of the feasibility version of the problem, in which we wish to determine whether there is a solution to the STTP with global preference value $\geq k$, for some k, we use as a certificate the set of times assigned to each event. The verification algorithm "chops" the set of preference values of each local preference function at k. The result of a chop, for each constraint, is a set of intervals of temporal values whose preference values are greater than k. The remainder of the verification process reduces to the problem of verifying General Temporal CSPs (TCSP), which is done by non-deterministically choosing an interval on each edge of the TCSP, and solving the resulting STP, which can be done in polynomial time. Therefore, STTPs belong to NP.

To prove hardness we reduce an arbitrary TCSP to an STPP. Thus, consider any TCSP, and take any of its constraints, say $I = \{[a_1, b_1], \ldots, [a_n, b_n]\}$. We will now obtain a corresponding soft temporal constraint containing just one interval (thus belonging to an STPP). The semiring that we will use for the resulting STPP is the classical one: $S_{csp} = \langle\{false, true\}, \vee, \wedge, false, true\rangle$. Thus the only two allowed preference values are false and true (or 0 and 1). Assuming that the intervals in I are ordered such that $a_i \leq a_{i+1}$ for $i \in \{1, \ldots, n-1\}$, the interval of the soft constraint is just $[a_1, b_n]$. The preference function will give value 1 to values in I and 0 to the others. Thus we have obtained an STPP whose set of solutions with value 1 (which are the optimal solutions, since $0 \leq_S 1$ in the chosen semiring) coincides with the set of solutions of the given TCSP. Since finding the set of solutions of a TCSP is NP-hard, it follows that the problem of finding the set of optimal solutions to an STPP is NP-hard. □

4 Linear and Semi-Convex Preference Functions

The hardness result for STPPs derives either from the nature of the semiring or the shape of the preference functions. In this section, we identify classes of preference functions which define tractable subclasses of STPPs.

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, the given STPP can be written as a linear programming problem, solving which is tractable [Cormen *et. al.*, 1990]. Thus,

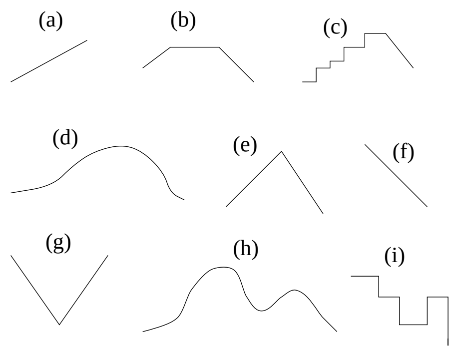

Figure 1: Examples of semi-convex functions (a)-(f) and non-semi-convex functions (g)-(i)

consider any given TCSPP. For any pair of variables X and Y, take each interval for the constraint over X and Y, say $[a, b]$, with associated linear preference function f. The information given by each of such intervals can be represented by the following inequalities and equation: $X - Y \leq b$, $Y - X \leq -a$, and $f_{X,Y} = c_1(X - Y) + c_2$. Then if we choose the fuzzy semiring $\langle [0, 1], max, min, 0, 1\rangle$, the global preference value V will satisfy the inequality $V \leq f_{X,Y}$ for each preference function $f_{X,Y}$ defined in the problem, and the objective is $max(V)$. If instead we choose the semiring $\langle \mathcal{R}, min, +, \infty, 0\rangle$, where the objective is to minimize the sum of the preference levels, we have $V = f_1 + \ldots + f_n$ and the objective is $min(V)$ [3]. In both cases the resulting set of formulas constitutes a linear programming problem.

Linear preference functions are expressive enough for many cases, but there are 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. Unless this value is one of the extremes of the interval, the preference function is convex, but not linear. 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.

A class of functions which includes linear, convex, and also some step functions will be called *semi-convex functions*. Semi-Convex functions have the property 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. Figure 1 shows examples of semi-convex and non-semi-convex functions.

More formally, a *semi-convex function* 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 $x > k$.

[3] In this context, the "+" is to be interpreted as the arithmetic operation, not the additive operation of the semiring.

Semi-Convex functions are closed under the operations of intersection and composition defined in Section 2, when certain semirings are chosen. For example, this happens with the fuzzy semiring, where the intersection performs the *min*, and composition performs the *max* operation. The closure proofs follow.

Theorem 2 (closure under intersection) *The property of functions being semi-convex is preserved under intersection. That is, given a totally-ordered semiring with an idempotent multiplicative operation $\times$ and binary additive operation $+$ (or $\sum$ over an arbitrary set of elements), let f_1 and f_2 be semi-convex functions which return values over the semiring. Let f be defined as $f(a) = f_1(a) \times f_2(a)$, where $\times$ is the multiplicative operation of the semiring. Then f is a semi-convex function as well.*

Proof: From the definition of semi-convex functions, it suffices to prove that, for any given y, the set $S = \{x : f(x) \geq y\}$ identifies an interval. If S is empty, then it identifies the empty interval. In the following we assume S to be not empty.

$$\{x : f(x) \geq y\} = \{x : f_1(x) \times f_2(x) \geq y\}$$
$$= \{x : min(f_1(x), f_2(x)) \geq y\}$$
($\times$ is a lower bound operator since it is assumed to be idempotent)
$$= \{x : f_1(x) \geq y \wedge f_2(x) \geq y\}$$
$$= \{x : x \in [a_1, b_1] \wedge x \in [a_2, b_2]\}$$
(since each of f_1 and f_2 is semi-convex)
$$= [max(a_1, a_2), min(b_1, b_2)]$$
□

Theorem 3 (closure under composition) *The property of functions being semi-convex is preserved under composition. That is, given a totally-ordered semiring with an idempotent multiplicative operation $\times$ and binary additive operation $+$ (or $\sum$ over an arbitrary set of elements), let f_1 and f_2 be semi-convex functions which return values over the semiring. Define f as $f(a) = \sum_{b+c=a}(f_1(b) \times f_2(c))$. Then f is a semi-convex function as well.*

Proof: Again, from the definition of semi-convex functions, it suffices to prove that, for any given y, the set $S = \{x : f(x) \geq y\}$ identifies an interval. If S is empty, then it identifies the empty interval. In the following we assume S to be not empty.

$$\{x : f(x) \geq y\}$$
$$= \{x : \textstyle\sum_{u+v=x}(f_1(u) \times f_2(v)) \geq y\}$$
$$= \{x : \max_{u+v=x}(f_1(u) \times f_2(v)) \geq y\}$$
(since $+$ is an upper bound operator)
$$= \{x : f_1(u) \times f_2(v) \geq y \text{ for some } u \text{ and } v \text{ such that } x = u + v\}$$
$$= \{x : \min(f_1(u), f_2(v)) \geq y \text{ for some } u \text{ and } v \text{ such that } x = u + v\}$$
($\times$ is a lower bound operator since it is assumed to be idempotent)
$$= \{x : f_1(u) \geq y \wedge f_2(v) \geq y,$$

$$\begin{aligned}&\qquad\qquad \text{for some } u+v=x\} \\ &= \{x : u \in [a_1,b_1] \wedge v \in [a_2,b_2], \\ &\qquad\qquad \text{for some } u+v=x \text{ and some } a_1,b_1,a_2,b_2\}\end{aligned}$$

(since each of f_1 and f_2 is semi-convex)

$$\begin{aligned}&= \{x : x \in [a_1+a_2, b_1+b_2]\} \\ &= [a_1+a_2, b_1+b_2]\end{aligned}$$

□

That closure of the set of semi-convex functions requires a total order and idempotence of the $\times$ operator is demonstrated by the following example. In what follows we assume monotonicity of the $\times$ operator. Let $\mathbf{a}$ and $\mathbf{b}$ be preference values with $\mathbf{a} \not< \mathbf{b}$, $\mathbf{b} \not< \mathbf{a}$, $\mathbf{a} \times \mathbf{b} < \mathbf{a}$, and $\mathbf{a} \times \mathbf{b} < \mathbf{b}$. Suppose x_1 and x_2 are real numbers with $x_1 < x_2$. Define $g(x) = \mathbf{1}$ for $x < x_1$ and $g(x) = \mathbf{a}$ otherwise. Also define $h(x) = \mathbf{b}$ for $x < x_2$ and $h(x) = \mathbf{1}$ otherwise. Clearly, g and h are semi-convex functions. Define $f = g \times h$. Note that $f(x) = \mathbf{b}$ for $x < x_1$, $f(x) = \mathbf{a} \times \mathbf{b}$ for $x_1 \le x < x_2$ and $f(x) = \mathbf{a}$ for $x \ge x_2$. Since $\{x | f(x) \not< a\}$ includes all values except where $x_1 \le x < x_2$, f is not semi-convex.

Now consider the situation where the partial order is not total. Then there are distinct incomparable values $\mathbf{a}$ and $\mathbf{b}$ that satisfy the condition of the example. We conclude the order must be total. Next consider the case in which idempotence is not satisfied. Then there is a preference value $\mathbf{c}$ such that $\mathbf{c} \times \mathbf{c} \neq \mathbf{c}$. It follows that $\mathbf{c} \times \mathbf{c} < \mathbf{c}$. In this case, setting $\mathbf{a} = \mathbf{b} = \mathbf{c}$ satisfies the condition of the example. We conclude that idempotence is also required.

The results in this section imply that applying the Path algorithm to an STPP with only semi-convex preference functions, and whose underlying semiring contains a multiplicative operation that is idempotent, and whose values are totally ordered, will result in a network whose induced soft constraints also contain semi-convex preference functions. These results will be applied in the next section.

5 Solving STPPs with Semi-Convex Functions is Tractable

We will now prove that STPPs with semi-convex preference functions and an underlying semiring with an idempotent multiplicative operation can be solved tractably.

First, we describe a way of transforming an arbitrary STPP with semi-convex preference functions into a STP. Given an STPP and an underlying semiring with A the set of preference values, let $y \in A$ and $\langle I, f\rangle$ be a soft constraint defined on variables X_i, X_j in the STPP, where f is semi-convex. Consider the interval defined by $\{x : x \in I \wedge f(x) \ge y\}$ (because f is semi-convex, this set defines an interval for any choice of y). Let this interval define a constraint on the same pair X_i, X_j. Performing this transformation on each soft constraint in the original STPP results in an STP, which we refer to as STP_y. (Notice that not every choice of y will yield an STP that is solvable.) Let opt be the highest preference value (in the ordering induced by the semiring) such that STP_{opt} has a solution. We will now prove that the solutions of STP_{opt} are the optimal solutions of the given STPP.

Theorem 4 *Consider any STPP with semi-convex preference functions over a totally-ordered semiring with $\times$ idempotent. Take opt as the highest y such that STP_y has a solution. Then the solutions of STP_{opt} are the optimal solutions of the STPP.*

Proof: First we prove that every solution of STP_{opt} is an optimal solution of STPP. Take any solution of STP_{opt}, say t. This instantiation t, in the original STPP, has value $val(t) = f_1(t_1) \times ... \times f_n(t_n)$, where t_i is the distance $v_j - v_i$ for an assignment to the variables X_i, X_j, $(v_i, v_j) = t \downarrow_{X_i, X_j}$, and f_i is the preference function associated with the soft constraint $\langle I_i, f_i\rangle$, with $v_j - v_i \in I_i$.

Now assume for the purpose of contradiction that t is not optimal in STPP. That is, there is another instantiation t' such that $val(t') > val(t)$. Since $val(t') = f_1(t'_1) \times ... \times f_n(t'_n)$, by monotonicity of the $\times$, we can have $val(t') > val(t)$ only if each of the $f_i(t'_i)$ is greater than the corresponding $f_i(t_i)$. But this means that we can take the smallest such value $f_i(t'_i)$, call it w', and construct $STP_{w'}$. It is easy to see that $STP_{w'}$ has at least one solution, t', therefore opt is not the highest value of y, contradicting our assumption.

Next we prove that every optimal solution of the STPP is a solution of STP_{opt}. Take any t optimal for STPP, and assume it is not a solution of STP_{opt}. This means that, for some constraint, $f(t_i) < opt$. Therefore, if we compute $val(t)$ in STPP, we have that $val(t) < opt$. Then take any solution t' of STP_{opt} (there are some, by construction of STP_{opt}). If we compute $val(t')$ in STPP, since $\times$ = glb (we assume $\times$ idempotent), we have that $val(t') \ge opt$, thus t was not optimal as initially assumed. □

This result implies that finding an optimal solution of the given STPP with semi-convex preference functions reduces to a two-step search process consisting of iteratively choosing a w, then solving STP_w, until STP_{opt} is found. Under certain conditions, both phases can be performed in polynomial time, and hence the entire process can be tractable.

The first phase can be conducted naively by trying every possible "chop" point y and checking whether STP_y has a solution. A binary search is also possible. Under certain conditions, it is possible to see that the number of chop points is also polynomial, namely:

- if the semiring has a finite number of elements, which is at most exponential in the number n of variables of the given STPP, then a polynomial number of checks is enough using binary search.
- if the semiring has a countably infinite number of elements, and the preference functions never go to infinity, then let l be the highest preference level given by the functions. If the number of values not above l is at most exponential in n, then again we can find opt in a polynomial number of steps.

The second phase, solving the induced STP_y, can be performed by transforming the graph associated with this STP into a distance graph, then solving two single-source shortest path problems on the distance graph [Dechter *et. al.*, 1991]. If the problem has a solution, then for each event it is possible to arbitrarily pick a time within its time bounds, and find corresponding times for the other events such that the set of times for all the events satisfy the interval constraints. The

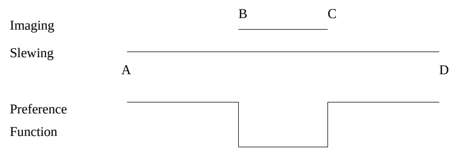

Figure 2: Non-semi-convex Preference Function for the Landsat problem

complexity of this phase is $O(en)$ (using the Bellman-Ford algorithm [Cormen *et. al.*, 1990]).

The main result of this discussion is that, while general TCSPPs are NP-Complete, there are sub-classes of TCSPP problems which are polynomially solvable. Important sources of tractability include the shape of the temporal preference functions, and the choice of the underlying semiring for constructing and comparing preference values.

Despite this encouraging theoretical result, the extent to which real world preferences conform to the conditions necessary to utilize the result is not clear. To illustrate this, consider again the motivating example at the outset. As illustrated in Figure 2, suppose an imaging event is constrained to occur during $[B, C]$, and that the interval $[A, D]$ is the interval during which a slewing event can start to occur. Assuming the soft constraint that prefers no overlap between the two occurrences, the preference values for the slewing can be visualized as the function pictured below the interval, a function that is not semi-convex. A semi-convex preference function would result by squeezing one or the other of the endpoints of the possible slewing times far enough that the interval would no longer contain the imaging time. For example, removing the initial segment $[A, B]$ from the interval of slewing times would result in a semi-convex preference function. Dealing with the general case in which preference functions are not semi-convex is a topic of future work.

6 Related work

The merging of temporal CSPs with soft constraints was first proposed in [Morris and Khatib, 2000], where it was used within a framework for reasoning about recurring events. The framework proposed in [Rabideau *et. al.*, 2000] contains a representation of local preferences that is similar to the one proposed here, but uses local search, rather than constraint propagation, as the primary mechanism for finding good complete solutions, and no guarantee of optimality can be demonstrated.

Finally, the property that characterizes semi-convex preference functions, viz., the convexity of the interval above any horizontal line drawn in the Cartesian plane around the function, is reminiscent of the notion of row-convexity, used in characterizing constraint networks whose global consistency, and hence tractability in solving, can be determined by applying local (path) consistency [Van Beek and Dechter, 1995]. There are a number of ways to view this connection. One way is to note that the row convex condition for the 0-1 matrix representation of binary constraints prohibits a row in which a sequence of ones is interrupted by one or more zeros. Replacing the ones in the matrix by the preference value for that pair of domain elements, one can generalize the definition of row convexity to prohibit rows in which the preference values decrease then increase. This is the intuitive idea underlying the behavior of semi-convex preference functions.

7 Summary

We have defined a formalism for characterizing problems involving temporal constraints over the distances and duration of certain events, as well as preferences over such distances. This formalism merges two existing frameworks, temporal CSPs and soft constraints, and inherits from them their generality, and also allows for a rigorous examination of computational properties that result from the merger.

Acknowledgments

Thanks to David McAllister, Rina Dechter and Alessandro Sperduti for helpful comments in earlier stages of the development of this work. This work was partially funded by the Research Institute for Advanced Computer Science, NASA Ames Research Center, and by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems).

References

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

[Cormen *et al.*, 1990] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT press, Cambridge, MA, 1990.

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

[Freuder and Wallace , 1992] E. C. Freuder and R. J. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58, 1992.

[Mackworth , 1977] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.

[Schiex *et al.*, 1995] T. Schiex, H. Fargier, and G. Verfaille. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *Proceedings of IJCAI95*, pages 631–637. Morgan Kaufmann, 1995.

[Morris and Khatib , 2000] R. Morris, and L. Khatib. Reasoning about Recurring Events: Satisfaction and Optimization. In *Computational Intelligence*, 16(2), 2000.

[Rabideau *et al.*, 2000] G. Rabideau, B. Engelhardt and S. Chien. Using Generic Preferences to Incrementally Improve Plan Quality. In *Proceedings of the 2nd NASA International Workshop on Planning and Scheduling for Space*, 11–16, March, 2000.

[Van Beek and Dechter , 1995] P. Van Beek, and R. Dechter. On the Minimality and Global Consistency of Row-Convex Constraint Networks. In *Journal of the ACM*, 42, 543–561, 1995.