Efficient and Expressive Extensions of Constraint-Based Temporal Reasoning

Michael D. Moffitt
Department of Electrical Engineering and Computer Science
University of Michigan
mmoffitt@eecs.umich.edu

Abstract

In this extended abstract, I present a brief overview of several proposed extensions to the field of constraint-based temporal reasoning. Combined, these extensions allow one to reason efficiently and simultaneously about overconstrained problems, preferences, finite-domain constraints, and uncertain situations within the context of Temporal CSPs. I also describe a particularly exciting application of these techniques to an area of research outside the usual scope of temporal reasoning.

Introduction

In the field of artificial intelligence, a great deal of effort has been extended toward improving existing methods for *temporal constraint satisfaction*. Temporal Constraint Satisfaction Problems (TCSPs) (Dechter *et al.* 1991) are especially suited to express constraints regarding the time, order, and duration of events, and as a result, it is common to find TCSPs applied in problems relating to planning and scheduling. Recent work has begun to extend the TCSP to handle uncontrollable events (Vidal & Fargier 1999; Morris & Muscettola 2005; Venable & Yorke-Smith 2005) and preferences (Khatib *et al.* 2001; Peintner & Pollack 2004).

However, there are certain situations where existing temporal representations and reasoning systems remain inadequate. First, it may be the case that a given TCSP is overconstrained, and thus admits no solution. If one instead desires a partial solution, where as many constraints can be satisfied as possible, traditional DTP solving algorithms are insufficient. Second, there are some scenarios in which the constraints of a given CSP contain a mixture of both finite-domain and temporal constraints. The problem of constructing such hybrid representations and algorithms has, until recently, been largely overlooked. Finally, there may be cases where the constraints of the problem are themselves uncertain. When such decisions lie outside the control of the constraint engine, it may be valuable to model the manner in which this information becomes known in an online environment, or to efficiently precompute a set of potential solutions in advance.

The objective of my thesis is to extend the particularly expressive Disjunctive Temporal Problem (DTP) along with traditional meta-CSP algorithms in order to cope with overconstrained problems, preferences, finite-domain constraints, and uncertain situations. To achieve these goals while maintaining efficiency requires both the creation of novel methods as well as the integration of well-established techniques that have proven effective in prior literature. An additional goal is to expand the range of applications to which TCSPs can be applied, demonstrating their usefulness outside the typical planning and scheduling domains that have been the focus of previous work.

Background

A *Disjunctive Temporal Problem* (DTP) (Stergiou & Koubarakis 1998) is a type of TCSP defined by a pair $\langle X, C \rangle$, where each element $X_i \in X$ designates a time point, and each element $C_i \in C$ is a constraint of the form:

$$c_{i1} \vee c_{i2} \vee ... \vee c_{in}$$

where in turn, each c_{ij} is of the form:

$$a_{ij} \leq x_{ij} - y_{ij} \leq b_{ij}$$

with $x_{ij}, y_{ij} \in X$ and $a_{ij}, b_{ij} \in \Re$. A solution to a DTP can be defined in one of two ways. The first of these is as an object-level assignment of a numeric value to each of the time points in X, such that all the constraints in C are satisfied. A second type of solution is a *meta-CSP* assignment. Here, instead of directly considering assignments to the time points in X, a meta-variable C_i is created for each constraint in the DTP. The domain $D(C_i)$ is simply the set $\{c_{i1}, c_{i2}, ..., c_{in}\}$, representing the various disjuncts one can choose to satisfy that disjunctive constraint. A meta-CSP solution is thus a selection of a single disjunct for each meta-variable such that the resulting set of inequalities is consistent.

Temporal Constraint Relaxation

A significant portion of my thesis deals with the problem of constraint relaxation in Disjunctive Temporal Problems. In this section I describe both systematic and approximate methods for handling overconstrained problems, and suggest how these can be applied to the more interesting issue of temporal preference optimization.

```
Partially-Solve-DTP(A, U, cost, upperbound)
If (cost ≥ upperbound) return
If (U = ∅)
      best-solution-so-far ← A
      upperbound ← cost
      return
EndIf
C_i ← select-variable(U), U′ ← U − {C_i}
For each disjunct c_ij of D(C_i)
      A′ ← A ∪ {C_i ← c_ij}
      If (consistent(A′))
            Partially-Solve-DTP(A′, U′, cost, upperbound)
      EndIf
EndFor
A′ ← A ∪ {C_i ← ε}
Partially-Solve-DTP(A′, U′, cost + 1, upperbound)
```

Figure 1: A PCS algorithm for DTPs

Partial Constraint Satisfaction of DTPs

Existing packages for solving DTPs, such as Epilitis (Tsamardinos & Pollack 2003) and TSAT++ (Armando *et al.* 2004), are sufficient for problems that admit one or more consistent solutions. However, in the event that a given DTP is overconstrained, these solvers are unable to provide anything other than a notice of failure. In some situations, one may instead desire a partial solution, in which as many constraints are satisfied as possible.

My thesis work extends traditional meta-CSP-based search algorithms in order to achieve *partial constraint satisfaction* of DTPs (Moffitt & Pollack 2005a). To accomplish this, the domain of each meta-variable (or constraint) in the DTP must be implicitly augmented with an empty disjunct, labeled 'ϵ'. This mechanism allows constraints to be violated explicitly during the meta-CSP search, a nuance that sets the algorithm apart from previous applications of partial constraint satisfaction to classical CSPs (Freuder & Wallace 1992). A solver, named Maxilitis, applies a branch-and-bound search (outlined in Figure 1) to minimize the total number of so-called *ϵ-relaxations*.

Applying Local Search to DTPs

One drawback to the systematic algorithm is that it can become rather expensive for extremely overconstrained problems that require a large number of constraint violations. Although Maxilitis has the anytime property (meaning that it can be interrupted at any time to extract a suboptimal solution), one may wonder whether there are more efficient ways of obtaining such solutions.

To address this question, my thesis work includes an application of local search to overconstrained temporal problems (Moffitt & Pollack 2005b). In contrast to previous work on DTPs, the approach works within a total assignment space at the object-level, and thus abandons the meta-CSP and corresponding graph-based consistency algorithms that have been employed in prior DTP literature. This particular search space presents several interesting challenges, such as the presence of infinitely many neighbors at each search node.

Revisiting Temporal Preference Optimization

One of the more active subjects in recent TCSP literature is the problem of *preferential optimization*. In this line of research, traditional temporal constraints (Dechter *et al.* 1991) are augmented with local preference functions that express how well a particular assignment satisfies the corresponding constraint. Early versions of this research focused on the problem of maximizing the minimum such preference value (Khatib *et al.* 2003; Peintner & Pollack 2004), although later developments have begun to address the more challenging problem of utilitarian optimization (Morris *et al.* 2004), where the sum of the individual preference values is maximized. Unfortunately, existing CSP-based methods for this objective (Peintner & Pollack 2005) have been shown to suffer in performance compared to more general SAT-based approaches (Sheini *et al.* 2005).

My thesis work explores a new means of obtaining utilitarian optimal solutions to Disjunctive Temporal Problems with Preferences (DTPPs) (Moffitt & Pollack 2006a). I depart from the SAT encoding and instead introduce the Valued DTP (VDTP). In contrast to the traditional semiring-based formalism (Bistarelli, Montanari, & Rossi 1997) that annotates legal object-level tuples of a constraint with preferences, the framework I develop instead assigns elementary costs to the constraints themselves, as is commonly done in finite-domain Valued CSP literature (Schiex *et al.* 1995). While this reformulation provides no increase in expressive power, it simplifies some of the computational difficulties related to temporal optimization, since (as mentioned earlier) search strategies for DTP solving rarely invoke object-level assignments directly. After proving that the VDTP can express the same set of utilitarian optimal solutions as the DTPP with piecewise-constant preference functions, I develop a method for achieving *weighted constraint satisfaction* within the meta-CSP search space that has traditionally been used to solve DTPs without preferences. This allows the application of well-established strategies that have proven effective in previous literature on both temporal reasoning and constraint optimization. As shown in Figure 2, empirical results suggest that an implementation of this approach (named WEIGHTWATCHER) consistently outperforms prior DTPP solvers – including GAPD (Peintner 2005) and the SAT-based solver ARIO (Sheini *et al.* 2005) – by several orders of magnitude.

Temporal/Finite-Domain Hybrid CSPs

There are some cases where the constraints of a given problem contain a mixture of both finite-domain and temporal components. For instance, consider the task of scheduling a set of meetings, where each meeting must be held in one of finitely-many locations. Temporal CSPs can quite easily capture temporal aspects of the problem such as start and end times, but a finite-domain network may be needed to reason about the locations. If these separate constraint networks exhibit any degree of interaction (e.g., if the physical locations of two meetings have an effect on their pairwise temporal relationship), then some kind of hybrid approach is required.

My thesis work considers the problem of constructing a

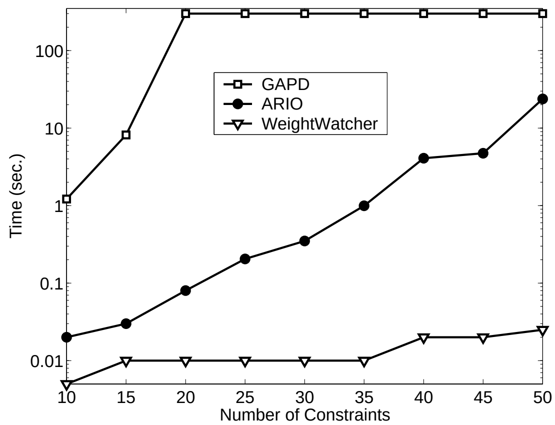

Figure 2: Median running times for GAPD, ARIO, and WEIGHTWATCHER for DTPPs of varying sizes (Timeout set at 300s)

hybrid constraint system capable of managing both finite-domain CSPs and temporal constraints (Moffitt *et al.* 2005), an endeavor that poses two formidable challenges. The first of these is that a more flexible representation is required to express both the individual constraint systems and their interaction. The second is that a new algorithm for establishing consistency of the hybrid problem is needed. In particular, I am in the process of developing a *least-commitment* algorithm especially suited for cases in which the finite-domain network is large but relatively underconstrained.

Dealing with Uncertainty

Uncertainty is a common element in many real-world scenarios. Within the context of temporal reasoning, prior work on uncertainty has focused on the presence of uncontrollable events (Vidal & Fargier 1999; Morris & Muscettola 2005; Venable & Yorke-Smith 2005), where the values of some subset of time points are decided on by nature. The problem is then no longer one of consistency, but rather one of controllability.

My thesis work examines a different dimension of uncertainty; specifically, how to deal with situations in which the constraints of the problem are themselves uncertain. For instance, if it is unknown whether a pair of activities must share the same resource, there may or may not exist a non-overlap constraint between them. Such uncertainty could exist even if the the object-level temporal variables are themselves fully controllable. In the presence of such uncertainty, several options are available. First, one can attempt to model the manner in which these constraints become known in an online environment, allowing the various notions of controllability to be generalized. Second, one can alternatively reason about the possible individual *realizations* of the problem, where each of these corresponds to a single DTP whose constraints are fixed. As an example, a precomputed set of potential solutions to the original problem, known as a *covering set closure* in classical CSP literature (Yorke-Smith & Gervet 2003), could be constructed. This might be done in a sequential enumerative fashion; or, since these realizations will likely share a significant amount of structure, some kind of parallelized approach may prove more efficient. Optimization variants exist as well, where a single solution is generated that maximizes the likelihood of feasibility. This work is largely in development, and I am still in the process of comparing my approaches to a wide body of related literature.

An Application to Optimal Rectangle Packing

So far, we have explored ways in which temporal representations and reasoning methods can be extended in order to handle overconstrained problems, preferences, finite-domain constraints, and uncertainty with respect to the constraints of the problem. While no single application has been used exclusively to motivate these extensions, one can imagine how the domains of planning and scheduling would benefit most directly, as they are popular areas to which temporal reasoning has traditionally been applied.

However, there are other problems that have attracted recent interest where TCSP techniques have yet to be considered. For instance, consider the topic of rectangle packing, a problem that has drawn attention from several diverse fields of computer science (e.g., VLSI/CAD) in addition to some areas of operations research. The current state-of-the-art (Korf 2003; 2004) has cast optimal rectangle packing as a CSP in which a variable is created for each rectangle, whose legal values are the positions that rectangle could occupy without exceeding the boundaries of the enclosing space. In addition, there is a binary constraint between each pair of rectangles, requiring that they do not overlap. To solve this CSP, Korf developed a backtracking algorithm, where each partial assignment is defined to be the fixed placement of a subset of rectangles. By obtaining lower bounds on the amount of wasted space at each node in the search, an algorithm was constructed that is the fastest known for optimal rectangle packing.

My thesis work addresses the problem of optimal rectangle packing (Moffitt & Pollack 2006b) in a way that departs from the aforementioned search space. Specifically, I cast the problem of optimally packing a set of rectangles with fixed orientations as a meta-CSP, in which a meta-variable is created for each pair of rectangles, whose values are the four pairwise relationships (i.e., *above*, *below*, *left of*, *right of*) that prevent that pair from overlapping. As such, commitment to the exact placement of any rectangle is not established until a consistent solution has been generated. I show how to apply several powerful DTP-solving techniques to this problem, and also develop a suite of new methods that exploit both the symmetry and geometry present in this particular domain. Despite its many differences with the fixed-placement formulation, the meta-CSP algorithm is shown to be quite competitive in performance, as evidenced in Figure 3 on a set of benchmarks fully explained in (Moffitt & Pollack 2006b).

Motivation for Other Extensions

The domain of rectangle packing proves to be an extremely interesting application of not only existing TCSP methods,

N	Opt. Dimen.	Korf '04	Moffitt '06
14	23×45	0	0
15	23×55	1	1
16	27×56	2	3
17	39×46	10	10
18	31×69	1:08	1:29
19	47×53	8:15	4:11
20	34×85	13:32	15:03
21	38×88	1:35:08	1:32:01
22	39×98	6:46:15	4:51:23
23	64×68	36:54:50	29:03:49
24	56×88	213:33:00	146:38:48

Figure 3: Experimental results for minimum-area rectangles than contain all consecutive squares from 1×1 up to $N \times N$. Runtime is reported in hours, minutes, and seconds.

but also of the other extensions proposed in this thesis. For instance, although the current formulation cannot represent rotatable rectangles, these could be handled by encoding the rectangles' orientations as finite-domain variables, and exploiting the hybrid representation discussed earlier. In addition, I have collaborated with researchers in VLSI to develop the *floorplan repair* problem, and proposed a means to solve it using a variation of the ϵ-relaxation (Moffitt *et al.* 2006).

Integration of Techniques

Within each of the extensions described in this paper, there are a number of issues that remain to be addressed. However, one of the more challenging tasks is to combine these extensions into a single, unified framework that elegantly integrates all techniques. The development of a unified framework is crucial for handling complicated real-world scenarios, such as calendar management and meeting scheduling, that require each of these extensions to some degree. A system based on this framework will be implemented and compared in both design and performance to other related recent developments.

References

Armando, A.; Castellini, C.; Giunchiglia, E.; and Maratea, M. 2004. A SAT-based decision procedure for the boolean combination of difference constraints. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT-2004)*.

Bistarelli, S.; Montanari, U.; and Rossi, F. 1997. Semiring-based constraint satisfaction and optimization. *Journal of the ACM* 44(2):201–236.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.

Freuder, E. C., and Wallace, R. J. 1992. Partial constraint satisfaction. *Artificial Intelligence* 58(1-3):21–70.

Khatib, L.; Morris, P.; Morris, R.; and Rossi, F. 2001. Temporal constraint reasoning with preferences. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, 322–327.

Khatib, L.; Morris, P.; Morris, R.; and Venable, K. B. 2003. Tractable Pareto optimal optimization of temporal preferences. In *Proceedings of the 18th International Joint Conference on Artificial Intelligece (IJCAI-2003)*, 1289–1294.

Korf, R. E. 2003. Optimal rectangle packing: Initial results. In *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS-2003)*, 287–295.

Korf, R. E. 2004. Optimal rectangle packing: New results. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS-2004)*, 142–149.

Moffitt, M. D.; Ng, A. N.; Markov, I. L.; and Pollack, M. E. 2006. Constraint-Driven Floorplan Repair. To appear in *Proceedings of the 43rd Design Automation Conference (DAC-2006)*.

Moffitt, M. D.; Peintner, B.; and Pollack, M. E. 2005. Augmenting disjunctive temporal problems with finite-domain constraints. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*, 1187–1192.

Moffitt, M. D., and Pollack, M. E. 2005a. Partial constraint satisfaction of disjunctive temporal problems. In *Proceedings of the 18th International Florida Artificial Intelligence Research Society Conference (FLAIRS-2005)*, 715–720.

Moffitt, M. D., and Pollack, M. E. 2005b. Applying local search to disjunctive temporal problems. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, 242–247.

Moffitt, M. D., and Pollack, M. E. 2006a. Temporal preference optimization as weighted constraint satisfaction. To appear in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-2006)*.

Moffitt, M. D., and Pollack, M. E. 2006b. Optimal rectangle packing: a meta-CSP approach. To appear in *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS-2006)*.

Morris, P. H., and Muscettola, N. 2005. Temporal dynamic controllability revisited. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*, 1193–1198.

Morris, P.; Morris, R.; Khatib, L.; Ramakrishnan, S.; and Bachmann, A. 2004. Strategies for global optimization of temporal preferences. In *Proceedings of the 10th International Conference on Principles and Practices of Constraint Programming*, 408–422.

Peintner, B., and Pollack, M. E. 2004. Low-cost addition of preferences to DTPs and TCSPs. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-2004)*, 723–728.

Peintner, B., and Pollack, M. E. 2005. Anytime, complete algorithm for finding utilitarian optimal solutions to STPPs. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-2005)*, 443–448.

Peintner, B. M. 2005. *Algorithms For Constraint-Based Temporal Reasoning With Preferences*. Ph.D. Dissertation, University of Michigan.

Schiex, T.; Fargier, H.; and Verfaillie, G. 1995. Valued constraint satisfaction problems: hard and easy problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-1995)*, 631–639.

Sheini, H. M.; Peintner, B.; Sakallah, K. A.; and Pollack, M. E. 2005. On solving soft temporal constraints using SAT techniques. In *Proceedings of the 11th International Conference on Principles and Practice of Constraint Programming (CP-2005)*, 607–621.

Stergiou, K., and Koubarakis, M. 1998. Backtracking algorithms for disjunctions of temporal constraints. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, 248–253.

Tsamardinos, I., and Pollack, M. E. 2003. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence* 151(1-2):43–90.

Venable, K. B., and Yorke-Smith, N. 2005. Disjunctive temporal planning with uncertainty. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2005)*, 1721–1722.

Vidal, T., and Fargier, H. 1999. Handling contingency in temporal constraint networks: From consistency to controllabilities. *Journal of Experimental and Theoretical Artificial Intelligence* 11(1):23–45.

Yorke-Smith, N., and Gervet, C. 2003. Certainty closure: A framework for reliable constraint reasoning with uncertainty. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP-2003)*, 769–783.