# A Reduction-based Approach for Solving Disjunctive Temporal Problems with Preferences

Jean-Rémi Bourguet[1], Marco Maratea[2], and Luca Pulina[1]

[1] POLCOMING - University of Sassari, Viale Mancini 5, 07100 Sassari, Italy.
boremi@uniss.it,lpulina@uniss.it
[2] DIBRIS - University of Genova, Viale F. Causa 15, Genova, Italy.
marco@dibris.unige.it

**Abstract.** Disjunctive Temporal Problems with Preferences (DTPPs) extend DTPs with piece-wise constant preference functions associated to each constraint of the form $l \leq x - y \leq u$, where $x, y$ are (real or integer) variables, and $l, u$ are numeric constants. The goal is to find an assignment to the variables of the problem that maximizes the sum of the preference values of satisfied DTP constraints, where such values are obtained by aggregating the preference functions of the satisfied constraints in it under a "max" semantic. The state-of-the-art approach in the field, implemented in the DTPP solver Maxilitis, extends the approach of the DTP solver Epilitis.
In this paper we present an alternative approach that reduces DTPPs to Maximum Satisfiability of a set of Boolean combination of constraints of the form $l \bowtie x - y \bowtie u$, $\bowtie \in \{<, \leq\}$, that extends previous work that dealt with constant preference functions only. Results obtained with the Satisfiability Modulo Theories (SMT) solver Yices on randomly generated DTPPs show that our approach is competitive to, and can be faster than, Maxilitis.

## 1 Introduction

Temporal constraint networks [1] provide a convenient formal framework for representing and processing temporal knowledge. Along the years, a number of extensions to the framework have been presented to deal with, e.g. more expressive preferences. Disjunctive Temporal Problems with Preferences (DTPPs) is one of such extensions. DTPPs extend DTPs, i.e. conjunctions of disjunctions of constraints of the form $l \leq x - y \leq u$, where $x, y$ are (real or integer) variables, and $l, u$ are numeric constants, with piece-wise constant preference functions associated to each constraint. The goal is to find an assignment to the variables of the problem that maximizes the sum of the preference values of satisfied disjunctions of constraints (called DTP constraints), where such values are obtained by aggregating the preference functions of the satisfied constraints in it. We consider an utilitarian aggregation of such DTP constraints values, and a "max" semantic for aggregating preference values within DTP constraints: given a (candidate) solution of a DTPP, the preference value of the DTP constraint is defined to

be the maximum value achieved by any of its satisfied disjuncts (see, e.g. [2]). The actual state-of-the-art approach that considers such aggregation methods is implemented in the DTPP solver MAXILITIS, and is based on an extension of the DTP approach of the solver EPILITIS [3] to deal with piece-wise constant preference functions. Various other approaches have been designed in the literature to deal with DTPPs [4–6, 2], possibly relying on alternative preference aggregation methods (see, e.g. [7, 8]).

In this paper we present an alternative approach that reduces DTPPs to Maximum Satisfiability of a set of Boolean combination of constraints of the form $l \bowtie x - y \bowtie u$, where $\bowtie \in \{<, \leq\}$. At first, we have considered a very natural modeling of the problem where the generated constraints are mutually exclusive, and each is weighted by a preference value: the set is constructed in order to maximize the degree of satisfaction of the DTP constraint. Preliminary experiments report that this solution is impractical. A second solution we propose is, instead, obtained by extending previous work that dealt with constant preference functions only [9], and reduces each DTP constraint to a set of disjunction of constraints, and a non-trivial interplay among their preference values to maximize, as before, the preference value of the DTP constraint. In order to test the effectiveness of our proposal, we have randomly generated DTPPs, following the method originally developed in [7] and then employed in all other papers on DTPPs. In our framework, each problem is then represented as a Satisfiability Modulo Theory (SMT) formula, and the YICES SMT solver, that is able to deal with optimization issue, is employed[3]. An experimental analysis conducted on a wide set of benchmarks, using the same benchmarks setting already employed in past papers, shows that our approach is competitive to, and can be faster than, MAXILITIS.

The rest of the paper is structured as follows. Section 2 introduces preliminaries about DTPs, DTPPs and Maximum Satisfiability. Then, in Section 3 we present our reduction from DTPPs to Maximum Satisfiability of Boolean combination of constraints, while the experimental analysis is presented in Section 4. The paper ends by providing a discussion about the related work in Section 5 and some conclusions in Section 6.

## 2 Formal Background

Problems involving disjunction of temporal constraints have been introduced in [10], as an extension of the Simple Temporal Problem (STP) [1], which consists of conjunction of different constraints. The problem was referred for the first time as Disjunctive Temporal Problem (DTP) in [11], and is presented in the first subsection. The remaining subsections introduce Maximum Satisfiability of DTPs and DTPPs.

[3] YICES showed the best performance in [9] among of number of alternatives, and it is the only SMT solver able to cope with (Partial Weighted) Maximum Satisfiability problems.

### 2.1 DTP

Let $\mathcal{V}$ be a set of symbols, called *variables*. A *constraint* is an expression of the form $l \bowtie x - y \bowtie u$, where $\bowtie \in \{<, \leq\}$, $x, y \in \mathcal{V}$, and $l, u$ are numeric constants. A *DTP constraint* is a disjunction of constraints having $\bowtie = \leq$ (equivalently seen as a disjunctively intended set of constraints). A *DTP formula*, or simply *formula*, is a conjunction of DTP constraints. A DTP constraint can be either *hard*, i.e. its satisfaction is mandatory, or *soft*, i.e. its satisfaction is not necessary but preferred, and in case of satisfaction it contributes to the generation of high quality solutions according to the aggregation methods employed and defined later. A *DTP$^A$ constraint* is a Boolean combination of constraints.

About the semantics, let the set $\mathcal{D}$ (*domain of interpretation*) be either the set of the real numbers $\mathcal{R}$ or the set of integers $\mathcal{Z}$. An *assignment* is a total function mapping variables to $\mathcal{D}$. Let $\sigma$ be an assignment and $\phi$ be a formula composed by hard DTP constraints only. Then, $\sigma \models \phi$ ($\sigma$ satisfies *a formula* $\phi$) is defined as follows

- $\sigma \models l \leq x - y \leq u$ if and only if $l \leq \sigma(x) - \sigma(y) \leq u$;
- $\sigma \models \neg\phi$ if and only if it is not the case that $\sigma \models \phi$;
- $\sigma \models (\wedge_{i=1}^{n} \phi_i)$ if and only if for each $i \in [1, n]$, $\sigma \models \phi_i$; and
- $\sigma \models (\vee_{i=1}^{n} \phi_i)$ if and only if for some $i \in [1, n]$, $\sigma \models \phi_i$.

If $\sigma \models \phi$ then $\sigma$ is also called a *model* of $\phi$. We also say that a formula $\phi$ is *satisfiable* if and only if there exists a model for $\phi$. The DTP is the problem of deciding whether a formula $\phi$ is satisfiable or not in the given domain of interpretation $\mathcal{D}$. Notice that the satisfiability of a formula depends on $\mathcal{D}$, e.g. the formula

$$x - y > 0 \wedge x - y < 1$$

is satisfiable if $\mathcal{D}$ is $\mathcal{R}$ but unsatisfiable if $\mathcal{D}$ is $\mathcal{Z}$. However, the problems of checking satisfiability in $\mathcal{Z}$ and $\mathcal{R}$ are closely related and will be treated uniformly.

### 2.2 Max-DTP

Consider now a DTP$^A$ formula $\phi$ consisting of hard DTP constraints and soft DTP$^A$ constraints. Intuitively, in this case the goal is to find an assignment to the variables in $\phi$ that satisfies all hard DTP constraints and maximizes the sum of the weights associated to satisfied soft DTP$^A$ constraints. The problem is called Partial Weighted Maximum Satisfiability of DTP$^A$, and is formally defined as a pair $\langle \phi, w \rangle$, where

1. $\phi$ is a DTP$^A$ formula consisting of both hard DTP and soft DTP$^A$ constraints, and
2. $w$ is a function that maps DTP$^A$ constraints to positive integer numbers.

More precisely, the goal is to find an assignment $\sigma'$ for $\phi$ that satisfies all hard DTP constraints and maximizes the following linear objective function $f$

$$f = \sum_{d \in \phi, \sigma' \models d} w(d) \tag{1}$$

where $d$ is a soft DTP$^A$ constraint. In the following, for simplicity we will use Max-DTP to refer to the Partial Weighted Maximum Satisfiability problem of mixed DTP and DTP$^A$ constraints as defined above.

### 2.3 DTPP

DTPP is an extension of DTP, and it is defined as a pair $\langle \phi, w' \rangle$, where

1. $\phi$ is a DTP formula consisting of both hard and soft DTP constraints, and
2. $w'$ is a (possibly partial) function that maps constraints in soft DTP constraints to piece-wise constant preference functions.

We consider, as before, an utilitarian method for aggregating soft DTP constraints weights: the goal is now to find an assignment $\sigma'$ for $\phi$ that $(i)$ satisfies all hard DTP constraints, and $(ii)$ maximizes the sum of weights associated to satisfied soft DTP constraints, i.e. maximizes the linear objective function (1).

It is left to define how weights, corresponding to preference values, are aggregated within soft DTP constraints to "define" their weights $w(d)$ in (1). In our work we consider a prominent semantic for this purpose: the *max* semantic.

Given a constraint $dc := l \leq x - y \leq u$, its preference function $w'(dc)$ is in general defined as:

$$w'(dc) : t \subseteq [l, u] \rightarrow [0, R^+]$$

mapping every feasible temporal interval $t$ to a preference value expressing its weight. The *max* semantic [5, 2] defines the weight $w(d)$ of a satisfied soft DTP constraint $d$ as the maximum among the possible preference values of satisfied constraints in $d$, i.e. given an assignment $\sigma'$

$$w(d) := max\{w'(\sigma'(x) - \sigma'(y)) : dc \in d, \sigma' \models dc\}$$

## 3 Reducing DTPPs to Max-DTPs

As we said before, our main idea is to reduce the problem of solving DTPPs to solving Max-DTPs. Hard DTP constraints remain unchanged in our reduction, while soft DTP constraints need special treatment. Given a soft DTP constraint $d$, for each constraint $dc$ in $d$, let $L_{dc}$ be a set of pairs, each pair $\langle DC, v \rangle$ being composed by $(i)$ a set $DC$ of pairs $(\overline{l}, \overline{u})$, representing the end points of intervals, such that $[\overline{l}, \overline{u}] \subseteq [l, u]$, and $(ii)$ the preference value $v$ of the constraints of the type $\overline{l} \bowtie x - y \bowtie \overline{u}$, $\bowtie \in \{\leq, <\}$, extracted from $DC$, where the variables $x, y$ are obtained from the constraint name. If the preference function is a constant $v'$,

$L_{dc}$ is composed by only one pair $\langle\{(l,u)\},v'\rangle$, i.e. the interval $[l,u]$, representing the constraint $l \leq x - y \leq u$, and its preference value $v'$.

We need now to "aggregate" the preference values corresponding to different levels of the piece-wise constant functions in the various constraints in order to implement our reduction. The idea is to "merge" the pairs $\langle DC, v\rangle$, representing preference function of constraints, in the same soft DTP constraint; intuitively, this means that, if the candidate solution satisfies at least one of the constraints obtained from $DC$ at preference value $v$, then a possible preference value for $d$ is $v$.

More formally, consider aggregating $L_{dc_1}$ and $L_{dc_2}$, coming from two constraints $dc_1$ and $dc_2$ in $d$, respectively. $L_{dc_1 \vee dc_2}$ :=MERGE$(L_{dc_1}, L_{dc_2})$ is an operator that

- contains the preference values that are in the preference functions of $dc_1$ or $dc_2$; and
- if the preference functions of $dc_1$ and $dc_2$ have a common preference value, i.e. $L_{dc_1}$ contains a pair $\langle DC_i, v_i\rangle$, $L_{dc_2}$ contains a pair $\langle DC_j, v_j\rangle$ and $v_i = v_j$, these pairs are merged and $L_{dc_1 \vee dc_2}$ contains a pair $\langle DC_i \cup DC_j, v_i\rangle$.

Moreover, during MERGE pairs $(\bar{l}, \bar{u})$ are attached a subscript, from which we deduce the ordered pair of variables involved in the constraint it represents.

The operator MERGE can be easily generalized to an arbitrary finite number of constraints.

Consider a soft DTP constraint

$$d := dc_1 \vee ... \vee dc_k \tag{2}$$

where $\{dc_1, \ldots, dc_k\}$ is the set of constraints in $d$.

The first attempt we considered for our reduction is to express a soft DTP constraint $d$ using soft DTP$^A$ constraints that force the highest preference value associated to satisfied constraints in $d$ to be assigned as weight for $d$. First, we apply the operator MERGE to all the constraints in $d$, and related piece-wise constant preference functions, i.e. $L_d$ :=MERGE$(L_{dc_1}, \ldots, L_{dc_k})$.

Further, consider an ordering on the $k$ pairs in $L_d$ of a $dc$ in $d$ induced by the preference values, i.e. an ordering $\prec$ is which $\langle DC_i, v_i\rangle \prec \langle DC_j, v_j\rangle$ iff $v_i < v_j, 1 \leq i, j \leq k, i \neq j$. For simplicity, from now on we consider the pairs in $L_d$ to be re-ordered according to $\prec$, i.e. $DC_1$ is the set whose $v_1$ is maximum among the weights in $d$, i.e. $v_1 > v_i, 2 \leq i \leq k$, while the set $DC_k$ is such that $v_k < v_i, 1 \leq i \leq k-1$.

Then, starting from $L_d$, $d$ and its preference value are expressed by the following $|L_d|$ soft DTP$^A$ constraints: for each $z = 1 \ldots |L_d|$

$$c_z := \wedge_{i=1}^{z-1} \neg(\vee_{p \in DC_i} dc_p) \wedge (\vee_{p \in DC_z} dc_p), w(d) = w(c_z) = v_z \tag{3}$$

where $dc_p$ is a constraint built from the pair $p$ (we recall that the subscript identifies the variables involved in the constraint, and in which order). The set

of constraints is mutually exclusive: considering an assignment, at most one of the constraints in (3) can be satisfied, and the relative value is assigned to $d$. If a constraint in (3) is satisfied, this is the constraint leading to the maximum value (according to the candidate solution considered).

This is done for each soft DTP constraint in the formula.

*Example 1.* Consider a soft DTP constraint $dc_1 \vee dc_2$, where $dc_1 : 1 \leq x - y \leq 10$ and $dc_2 : 5 \leq z - q \leq 15$. The piece-wise constant preference function associated to $dc_1$ is

$$f(dc_1) = \begin{cases} 1 & 1 \leq x - y \leq 3 \\ 2 & 3 < x - y \leq 7 \\ 1 & 7 < x - y \leq 10 \end{cases} \tag{4}$$

and can be represented with $L_{dc_1} = \{\langle\{(1,3),(7,10)\},1\rangle, \langle\{(3,7)\},2\rangle\}$.

Regarding $dc_2$, its preference function is

$$f(dc_2) = \begin{cases} 2 & 5 \leq z - q \leq 8 \\ 4 & 8 < z - q \leq 10 \\ 2 & 10 < z - q \leq 15 \end{cases} \tag{5}$$

represented with $L_{dc_2} = \{\langle\{(5,8),(10,15)\},2\rangle, \langle\{(8,10)\},3\rangle\}$. We now "merge" $L_{dc_1}$ and $L_{dc_2}$ into $L_{dc_1 \vee dc_2} :=\text{MERGE}(L_{dc_1}, L_{dc_2})$ whose result is

$$\{\langle\{(1,3)_1,(7,10)_1\},1\rangle, \langle\{(3,7)_1,(5,8)_2,(10,15)_2\},2\rangle, \langle\{(8,10)_2\},4\rangle\}. \tag{6}$$

Following (3), the reduction is

$$c_1 : (8 < z - q \leq 10),\ w(c_1) = 4$$

$$c_2 : \neg c_1 \wedge ((3 < x - y \leq 7) \vee (5 \leq z - q \leq 8) \vee (10 < z - q \leq 15)),\ w(c_2) = 2$$

$$c_3 : \neg c_1 \wedge \neg c_2 \wedge (1 \leq x - y \leq 3 \vee 7 < x - y \leq 10),\ w(c_3) = 1$$

Further note that the preference functions we have considered are characterized by having the left-most sub-interval with both bounds included, while the remaining sub-intervals have only the right bound included: to correctly reproduce the reduction from the set $L$, we have further assumed that with the subscript we can recognize the left-most sub-interval of each constraint.

This first reduction corresponds to a very natural way of expressing soft DTP constraints; unfortunately, preliminary experiments show that it is inefficient.

A second reduction transforms each soft DTP constraint $d$ to $|L_d|$ soft DTP$^A$ constraints as follows: for each $z = 1 \ldots |L_d|$

$$c'_z := \vee_{i=1}^{z} \vee_{p \in DC_i} dc_p \tag{7}$$

The problem is now to define what are the weights associated to each newly defined soft DTP$^A$ constraint, in order to reflect the semantic of our problem.

In the previous reduction (2), the constraints occurred positively only once; now there can be many occurrences in the corresponding soft DTP$^A$ constraints in (7) that influence constraints weights adaptation and definition. Our solution starts from the following fact: if the constraint $c'_{|L_d|}$ (i.e. the one that contains all constraints generated with out method) is satisfied, it is safe to consider that it contributes for at least the minimal preference value $v_{|L_d|}$, i.e. the one associated to the set $DC_{|L_d|}$, from which $c'_{|L_d|}$ is constructed. Satisfying the constraint $c'_{|L_d|-1}$ contributes for $v_{|L_d|-1} - v_{|L_d|}$, and given that a constraint $c'_z$ implies all constraints $c'_{z'}$, $z' > z$, these two soft DTP$^A$ constraints together contribute for $v_{|L_d|-1}$. This method is recursively applied up to the set of constraints constructable from $DC_1$, i.e. $c'_1$, whose preference value is $v_1 - v_2$ and, given that $c'_1$ implies all other introduced soft DTP$^A$ constraints, satisfying $c'_1$ correctly corresponds to assign a weight $v_1$ to $d$.

More formally, for each $z = 1 \dots |L_d|$

$$w(c'_z) = \begin{cases} v_{|L_d|} & z = |L_d| \\ v_z - v_{z+1} & 1 \leq z < |L_d| \end{cases} \tag{8}$$

and, given an assignment $\sigma'$, $w(d) = \sum_{z \in \{1,\dots,|L_d|\}, \sigma' \models c'_z} v_z$.

*Example 2.* Concerning the second reduction, the soft DTP$^A$ constraints that express the constraint $d$ with the preference functions in Example 1 are

$$c'_1 := 8 < z - q \leq 10,\ w(c'_1) = 2$$

$$c'_2 := c'_1 \vee (3 < x - y \leq 7 \vee 5 \leq z - q \leq 8 \vee 10 < z - q \leq 15),\ w(c'_2) = 1$$

$$c'_3 := c'_1 \vee c'_2 \vee (1 \leq x - y \leq 3 \vee 7 < x - y \leq 10),\ w(c'_3) = 1$$

Such reduction works correctly if we consider a single soft DTP constraint. However, considering a formula $\phi$, given our reduction, it is possible to have repeated DTP$^A$ constraints in the reduced formula $\phi'$. In this case, intuitively, we want each single occurrence in $\phi'$ to count "separately", given that they take into account different contributions from different soft DTP constraints in $\phi$. A solution is to consider a *single occurrence* of the resulting soft DTP$^A$ constraint in $\phi'$ whose weight is the sum of the weights of the various occurrences. The same applies to the first reduction.

## 4 Experimental Analysis

We have implemented both reductions, and expressed the resulting formulas as SMT formulas with optimization, then solved with Yices ver. 1.0.38. A preliminary analysis showed that the first reduction is not competitive, thus our experimental analysis compares the performance of our second reduction, called dtppYices, with two versions of the Maxilitis solver, namely Maxilitis-IW and Maxilitis-BB. Maxilitis-IW (IW standing for Iterative Weakening)

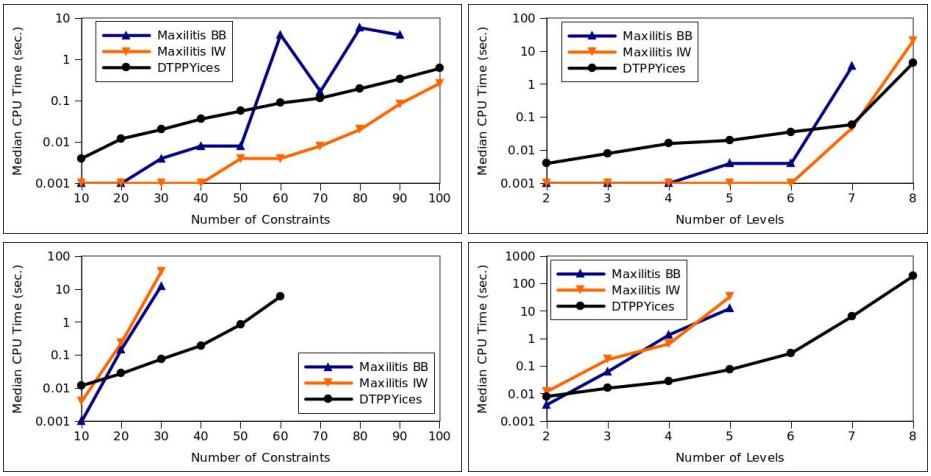


**Fig. 1.** Results of the evaluated solvers on random DTPPs.

searches for solutions with a progressively increasing number of violated constraints; MAXILITIS-BB uses a branch and bound approach for reaching the optimal solution. Our experiments aim at comparing the considered solvers on two dimensions, namely the size of the benchmarks and the number of preference levels in the piece-wise constant preference function, as used in past papers on DTPPs, with the same parameter settings. Moreover, we also investigated the performance of the solvers in the case where the preference values of the levels of the piece-wise preference functions are randomly generated. For randomly generating the benchmarks the main parameters considered are: $(i)$ the number $k$ of disjuncts per DTP constraint; $(ii)$ the number $n$ of arithmetic variables; $(iii)$ the number $m$ of DTP constraints; and $(iv)$ number $l$ of levels in the preference functions.[4] For each tuple of values of the parameters, 25 instances have been generated.

The experiments reported in the following ran on PCs equipped with a processor Intel Core i5 running at 3.20 GHz, with 4 GB of RAM, and running GNU Linux Ubuntu 12.04. The timeout for each instance has been set to 300s.

As a first experiment, we randomly generated benchmarks by varying the total amount of constraints, with the following parameters: $k$=2, $m \in \{10, \ldots, 100\}$, $n$=$0.8 \times m$, $l$=5, lower and upper bounds of each constraint taken in $[-50, 100]$. In this setting, the preference value of the i-th levels is $i$.[5]

[4] The preference functions considered are, as in previous papers, semi-convex piecewise constant: starting from the lower and upper bounds of the constraints, intervals corresponding to higher preference levels are randomly put within the interval of the immediate lower level, with a reduction factor, up to an highest level. For details see, e.g. [2].

[5] These benchmarks have been generated using the program provided by Michael D. Moffitt, author of MAXILITIS.

The results obtained in the experiment are shown in Figure 1, which is organized as follows. Concerning the left-most plots, in the $x$ axis we show the total amount of constraints, while in the right-most plots the total amount of levels of the piece-wise constant preference function is reported. In the $y$ axis (in log scale), it is shown the related median CPU time (in seconds). MAXILITIS-BB's performance is depicted by blue triangles, MAXILITIS-IW's by using orange upside down triangles, and DTPPYICES performance is denoted by black circles. Plots in the top row have a preference value corresponding to $i$ for the i-th preference level, while plots in the bottom row are related to random DTPPs whose preference values are randomly generated in $\{1, \ldots, 100\}$ (still ensuring to maintain the same shape for preference functions),

Looking at Figure 1, and considering the top-left plot, we can see that the median time of MAXILITIS-BB on benchmarks with 100 constraints runs into timeout. We can also see the up to $m = 80$, MAXILITIS-IW is one order of magnitude of CPU time faster that DTPPYICES, while for $m > 80$ the performance of the solvers are in the same ballpark. Now, considering the same analysis in the case where values of the preference levels are randomly generated, we can see (bottom-left plot) that the picture changes in a noticeable way. Benchmarks are harder than previously: MAXILITIS-BB and MAXILITIS-IW are not able to efficiently cope with benchmarks with $m > 30$. In this case, DTPPYICES is the best solver, and we report that it is able to deal with benchmarks up to $m = 60$.

Detailed results for these benchmarks containing, for each solver and number of constraints, the number of solved instances, and the sum of their solving CPU times, are reported in Table 1.

Our next experiment aims to evaluate the solvers by varying the number of levels in the preference functions, with the following parameters: $k$=2, $n$=24, $m$=30, $l \in \{2, \ldots, 8\}$, lower and upper bounds of each constraint taken in $[-50, 100]$. Top-right and bottom-right plots have the same meaning as before w.r.t. the preference functions. Looking at the top-right plot of Figure 1, we can see that MAXILITIS-IW is the best solver up to $l = 7$, while for $l = 8$, we report that DTPPYICES is faster. Also in this case MAXILITIS-BB does not efficiently deal with the most difficult benchmarks in the suite. Looking now at the plot in the bottom-right, we can see the same picture related to the bottom-left plot: the performance of both versions of MAXILITIS are very similar, while DTPPYICES is the fastest solver: the median CPU time of both MAXILITIS-BB and MAXILITIS-IW runs in timeout for $l > 5$, while DTPPYICES solves all set of benchmarks within the time limit. Along with the previous results, this reveals that MAXILITIS may have specialized techniques to deal with DTPPs whose preference values are of the first type we have analyzed.

Finally, detailed results for each number of levels are reported in Table 2.

| N | M | Maxilitis-BB | | | | Maxilitis-IW | | | | dtppYices | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Fixed | | Rand | | Fixed | | Rand | | Fixed | | Rand | |
| | | # | Time | # | Time | # | Time | # | Time | # | Time | # | Time |
| 8 | 10 | 25 | 0.01 | 25 | 0.04 | 25 | 0.01 | 25 | 0.33 | 25 | 0.12 | 25 | 0.27 |
| 16 | 20 | 25 | 0.16 | 25 | 16.79 | 25 | 0.01 | 25 | 62.97 | 25 | 0.33 | 25 | 0.92 |
| 24 | 30 | 25 | 5.75 | 18 | 593.82 | 25 | 0.02 | 21 | 922.82 | 25 | 0.55 | 25 | 4.78 |
| 32 | 40 | 24 | 70.12 | 3 | 85.98 | 25 | 0.05 | 9 | 946.78 | 25 | 1.06 | 25 | 27.92 |
| 40 | 50 | 22 | 27.58 | 1 | 108.71 | 25 | 0.29 | 3 | 334.98 | 25 | 1.68 | 24 | 278.40 |
| 48 | 60 | 17 | 254.24 | – | – | 25 | 4.26 | – | – | 25 | 2.80 | 20 | 225.48 |
| 56 | 70 | 21 | 59.28 | – | – | 25 | 0.70 | – | – | 25 | 4.04 | 10 | 900.48 |
| 64 | 80 | 16 | 155.92 | – | – | 25 | 7.16 | – | – | 25 | 5.91 | 5 | 110.92 |
| 72 | 90 | 17 | 400.46 | – | – | 25 | 15.38 | – | – | 25 | 9.01 | 3 | 225.93 |
| 80 | 100 | 12 | 790.15 | – | – | 25 | 58.37 | – | – | 25 | 12.31 | 3 | 144.44 |

**Table 1.** Performance of the selected solvers on random DTPPs with different sizes. The first columns ("N") reports the total amount of variables for each pool of DTPPs, while the second one ("M") reports the total amount of constraints. It is followed by three groups of columns, and the label is the solver name. Each group is composed of four columns, reporting the total amount of instances solved within the time limit ("#") and the total CPU time in seconds ("Time") spent, both in the case of fixed preference value corresponding to the level, and randomly generated (groups "Fixed" and "Rand", respectively). In case a solver does not solve any instance, "–" is reported.

| L | Maxilitis-BB | | | | Maxilitis-IW | | | | dtppYices | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fixed | | Rand | | Fixed | | Rand | | Fixed | | Rand | |
| | # | Time | # | Time | # | Time | # | Time | # | Time | # | Time |
| 2 | 25 | 0.01 | 25 | 1.70 | 25 | 0.01 | 25 | 2.86 | 25 | 0.10 | 25 | 0.20 |
| 3 | 25 | 0.01 | 25 | 7.10 | 25 | 0.01 | 25 | 47.69 | 25 | 0.21 | 25 | 0.41 |
| 4 | 25 | 0.01 | 21 | 396.39 | 25 | 0.01 | 25 | 205.43 | 25 | 0.36 | 25 | 0.93 |
| 5 | 25 | 5.81 | 18 | 593.82 | 25 | 0.02 | 21 | 922.82 | 25 | 0.55 | 25 | 4.78 |
| 6 | 24 | 33.63 | 10 | 679.97 | 25 | 4.83 | 10 | 614.07 | 25 | 1.82 | 25 | 22.12 |
| 7 | 21 | 235.20 | 2 | 68.38 | 23 | 130.73 | 2 | 59.78 | 25 | 80.57 | 21 | 270.73 |
| 8 | 12 | 450.63 | 2 | 218.46 | 17 | 602.64 | 2 | 306.20 | 25 | 195.52 | 13 | 493.60 |

**Table 2.** Performance of the selected solvers on random DTPPs with different levels. In column "L" we report the total amount of levels, while the rest of the table is organized similarly to Table 1.

## 5 Related Work

Maxilitis [2, 5], WeightWatcher [6] and ARIO [4] implement different approaches for solving DTPPs as defined in [7]. Maxilitis is a direct extension of the DTP solver Epilitis [3], while WeightWatcher uses an approach based on Weighted Constraints Satisfaction problems, even if the two methods are similar (as mentioned in, e.g., [6]). ARIO, instead, relies on an approach based

on Mixed Logical Linear Programming (MLLP) problems. In our analysis we have used MAXILITIS because the results in, e.g. [2] clearly indicate its superior performance.

About the comparison to MAXILITIS, our solution is easy, yet efficient, and has a number of advantages w.r.t. the approach of MAXILITIS. On the modeling side, it allows to consider (with no modifications) both integer and real variables, while MAXILITIS can deal with integer variables only. Moreover, our implementation provides an unique framework for solving DTPPs, while the techniques proposed in [2] are implemented in two separate versions of MAXILITIS. Finally, our solution is modular, i.e. it is easy to rely on different back-end solvers (or, on a new version of YICES), thus taking advantages on new algorithms and tools for solving our formulas of interest.

## 6 Conclusions

In this paper we have introduced a general reduction-based approach for solving DTPPs, that reduces these problems to Maximum Satisfiability of DTPs as defined in the paper. An experimental analysis performed with the YICES SMT solver on randomly generated DTPPs shows that our approach is competitive to, and sometimes faster than, the specific implementations of the MAXILITIS solver. The executable of our solver can be found at

`http://www.star.dist.unige.it/~marco/DTPPYices/`.

**Acknowledgment.** The authors would like to thank Michael D. Moffitt for providing his solvers and the program for generating random benchmarks, and Bruno Dutertre for his support about YICES.

## References

1. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. Artificial Intelligence **49**(1-3) (1991) 61–95
2. Moffitt, M.D.: On the modelling and optimization of preferences in constraint-based temporal reasoning. Artificial Intelligence **175**(7-8) (2011) 1390–1409
3. Tsamardinos, I., Pollack, M.: Efficient solution techniques for disjunctive temporal reasoning problems. Artificial Intelligence **151** (2003) 43–89
4. Sheini, H.M., Peintner, B., Sakallah, K.A., Pollack, M.E.: On solving soft temporal constraints using SAT techniques. In van Beek, P., ed.: Proc. of the 11th International Conference on Principles and Practice of Constraint Programming (CP 2005). Volume 3709 of Lecture Notes in Computer Science., Springer (2005) 607–621
5. Moffitt, M.D., Pollack, M.E.: Partial constraint satisfaction of disjunctive temporal problems. In Russell, I., Markov, Z., eds.: Proc. of the 18th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS 2005), AAAI Press (2005) 715–720

6. Moffitt, M.D., Pollack, M.E.: Temporal preference optimization as weighted constraint satisfaction. In: Proc. of the 21st National Conference on Artificial Intelligence (AAAI 2006), AAAI Press (2006)
7. Peintner, B., Pollack, M.E.: Low-cost addition of preferences to DTPs and TCSPs. In McGuinness, D.L., Ferguson, G., eds.: Proc. of the 19th National Conference on Artificial Intelligence (AAAI 2004), AAAI Press / The MIT Press (2004) 723–728
8. Peintner, B., Moffitt, M.D., Pollack, M.E.: Solving over-constrained disjunctive temporal problems with preferences. In Biundo, S., Myers, K.L., Rajan, K., eds.: Proc. of the 15th International Conference on Automated Planning and Scheduling (ICAPS 2005), AAAI (2005) 202–211
9. Maratea, M., Pulina, L.: Solving disjunctive temporal problems with preferences using maximum satisfiability. AI Commununications **25**(2) (2012) 137–156
10. Stergiou, K., Koubarakis, M.: Backtracking algorithms for disjunctions of temporal constraints. In Shrobe, H.E., Mitchell, T.M., Smith, R.G., eds.: Proc. of the 15th National Conference on Artificial Intelligence (AAAI 1998), AAAI Press / The MIT Press (1998) 248–253
11. Armando, A., Castellini, C., Giunchiglia, E.: SAT-based procedures for temporal reasoning. In Biundo, S., Fox, M., eds.: Proc. of the 5th European Conference on Planning (ICAPS 1999). Volume 1809 of Lecture Notes in Computer Science., Springer (1999) 97–108