Preference-Based Improvements on Solutions of Multi-Agent Temporal Problems by Automated Negotiation

Johan Los

September 2016

Master's thesis
Artificial Intelligence
University of Groningen

Supervisors:
Prof. dr. L.C. Verbrugge, University of Groningen
Prof. dr. C. Witteveen, Delft University of Technology

university of groningen

TU Delft

Preference-Based Improvements on Solutions of Multi-Agent Temporal Problems by Automated Negotiation

Johan Los

September 2016

Master's thesis: MSc Artificial Intelligence
Faculty of Mathematics and Natural Sciences
University of Groningen

European credits: 45 ECTS

First supervisor: Prof. dr. L.C. Verbrugge
University of Groningen
Faculty of Mathematics and Natural Sciences
Artificial Intelligence & Cognitive Engineering
Nijenborgh 9, 9747 AG Groningen, The Netherlands
Room: 5161.0355
L.C.Verbrugge@rug.nl
+31 50 363 6334

Second supervisor: Prof. dr. C. Witteveen
Delft University of Technology
Faculty of Engineering, Mathematics and Computer Science
Algorithmics
Mekelweg 4, 2628 CD Delft, The Netherlands
Room: HB 07.110
C.Witteveen@tudelft.nl
+31 15 278 2521

university of groningen

TUDelft

Abstract

Scheduling problems arise in numerous real-world situations. In many of them, different agents are each responsible for their own part of a scheduling problem, although they are dependent on the schedules of other agents. Hence, the agents have to adjust their schedules to certain external constraints.

General approaches for solving multi-agent scheduling problems return an arbitrary solution that satisfies all constraints, regardless of the quality of the schedule. However, in many cases, agents have certain preferences with respect to the schedule, for example, they want to do tasks in a certain order, to complete all tasks as early as possible, or to have a break between two tasks.

In the current literature, scheduling frameworks with preferences are only defined for single-agent problems. We therefore extend the work on Simple Temporal Problems and Disjunctive Temporal Problems, and develop the frameworks of Multi-agent Simple Temporal Problems with Preferences and Multi-agent Disjunctive Temporal Problems with Preferences.

Having defined frameworks for these problems, we develop methods for solving them. Existing approaches for Multi-agent Temporal Problems search for a decoupling, i.e. a collection of local subproblems, one for each agent, such that any combination of solutions to the local subproblems is a solution to the original shared problem. By this, the agents can solve the subproblems in parallel, and their privacy is respected.

We propose to improve existing decoupling solutions, with the preferences of the different agents as performance criterion, by applying automated negotiation. Automated negotiation is in general a computationally efficient distributed process in which the agents do not have to reveal all their preference information, that deals with both cooperation and competition, and hence fits well into the multi-agent scheduling context. Two types of negotiation protocols are developed: First, we propose post-decoupling protocols for finding Pareto-improvements on a given decoupling. Next, we develop pre-decoupling negotiation protocols that can be applied in existing decoupling algorithms to improve the decoupling.

Our results show that both the post-decoupling and the pre-decoupling negotiation methods improve existing decoupling algorithms with respect to the preferences of the agents. For a larger number of agents, the pre-decoupling negotiation algorithms turn out to be better. Furthermore, we compare our solutions with methods that have the flexibility of a solution as performance metric and show that our solutions are more flexible. Finally, the possibilities for further increasing social welfare by letting the agents also make non-Pareto improvements are discussed.

List of Abbreviations

△DPC	Triangulating Directional Path Consistency
CSP	Constraint Satisfaction Problem
D△DPC	Distributed Triangulating Directional Path Consistency
D△PPC	Distributed Triangulating Partial Path Consistency
DPC	Directional Path Consistency
DTP	Disjunctive Temporal Problem
DTPP	Disjunctive Temporal Problem with Preferences
FPC	Full Path Consistency
MaDTP	Multi-agent Disjunctive Temporal Problem
MaDTPP	Multi-agent Disjunctive Temporal Problem with Preferences
MaDTP-TD	MaDTP Temporal Decoupling
MaSTP	Multi-agent Simple Temporal Problem
MaSTPP	Multi-agent Simple Temporal Problem with Preferences
MaTD	Multi-agent Temporal Decoupling
MaTDR	Multi-agent Temporal Decoupling with Relaxation
OMT	Optimization Modulo Theories
PPC	Partial Path Consistency
SMT	Satisfiability Modulo Theories
STN	Simple Temporal Network
STP	Simple Temporal Problem
STPP	Simple Temporal Problem with Preferences
VDTP	Valued Disjunctive Temporal Problem

Contents

Abstract **5**

List of Abbreviations **7**

1 Introduction **11**
1.1 Research goals . . . 12
1.2 Thesis outline . . . 13

2 Theoretical Background **14**
2.1 Temporal problems . . . 15
2.1.1 Simple Temporal Problems . . . 15
2.1.2 Disjunctive Temporal Problems . . . 17
2.1.3 Temporal Problems with Preferences . . . 18
2.1.4 Multi-agent Temporal Problems . . . 19
2.2 Automated negotiation . . . 25
2.2.1 General overview . . . 25
2.2.2 Strategy components in automated negotiation . . . 26
2.2.3 Sophisticated negotiation approaches . . . 27
2.3 Formalized research goals . . . 27

3 The Multi-Agent Simple Temporal Problem with Preferences **30**
3.1 Example . . . 30
3.2 Framework definition . . . 30
3.3 Analysis . . . 34

4 The Multi-Agent Disjunctive Temporal Problem with Preferences **35**
4.1 Example . . . 35
4.2 Adding preferences . . . 37
4.2.1 Preferences at time point level and at disjunct level . . . 37
4.2.2 Using Disjunctive Normal Form to handle dependencies between constraints 38
4.3 Framework definition . . . 39
4.4 Analysis . . . 45

5 Solving the Multi-Agent Simple Temporal Problem with Preferences **46**
5.1 Post-decoupling negotiation approach . . . 46
5.1.1 Protocol . . . 47
5.1.2 Expected utility . . . 47
5.1.3 Obtaining local consistency . . . 48
5.1.4 Example . . . 50
5.2 Pre-decoupling negotiation approach . . . 51
5.2.1 Protocol . . . 52
5.2.2 Example . . . 52

6 Solving the Multi-Agent Disjunctive Temporal Problem with Preferences **54**
6.1 Post-decoupling negotiation approach . . . 54
6.1.1 Protocol . . . 55
6.1.2 Example . . . 55
6.2 Pre-decoupling negotiation approach . . . 56
6.2.1 Optimization modulo theories . . . 57
6.2.2 Protocol . . . 58
6.2.3 Example . . . 58

7 Experiments and Results **61**
7.1 Problem set . . . 61
7.2 Experiments . . . 63
7.2.1 General performance . . . 63
7.2.2 Allowing non-Pareto improvements . . . 63
7.2.3 Comparing flexibility . . . 64
7.3 Results . . . 65
7.3.1 No profit from non-Pareto negotiations . . . 68
7.3.2 Larger flexibility . . . 68

8 Discussion **72**
8.1 Algorithms . . . 74
8.2 Implications . . . 75
8.3 Future work . . . 75

Bibliography **76**

Chapter 1

Introduction

Scheduling problems arise in many different real-world situations. In manufacturing processes, for example, different production steps are dependent on one another, and need to be planned taking several factors into account. Airplane and railway traffic has to deal with limited space at airports and stations, but needs to meet several requirements with respect to arrival and departure times. Maintenance workshops have to handle strict deadlines for their reparations, using a restricted set of resources and work force. Or consider the work schedules for people in different domains, e.g. in the educational sector: Lecturers have to give lectures on their specialization topics, beside their other tasks. Furthermore, tutorials and practical sessions have to be planned, but there needs to be enough preparation time (both for the lecturer and for the students) in between the different class hours. On the other hand, there are different groups of students, each with their own study program. Each group needs to be able to attend the appropriate lectures, making the scheduling problem quite complex.

In many of such scheduling problems, different individuals or agencies (in general: agents) are each responsible for their own part of a scheduling problem, although they are dependent on the schedules of other agents. Consider, for example, a lecturer who needs to plan a meeting with a student. Such a meeting will not be incorporated in the global university schedules, but needs to be planned by the lecturer and the student themselves. However, there are certain restrictions with respect to, for example, the class hours: The lecturer cannot plan the meeting at a time when he needs to give a lecture, and the student cannot meet the lecturer when he needs to attend a lecture. Hence, in general, the agents in such multi-agent scheduling problems have to adjust their schedules to certain external constraints.

Although the autonomy of the agents for scheduling their own tasks can be strongly restricted by external constraints, it is possible for them to partially obtain freedom in scheduling their tasks when the external requirements have been fixed. When a student on a morning needs both to study a topic himself, and to discuss the topic with his professor, an individual plan cannot be made by the student, since he is dependent on the scheduling requirements of the professor. However, when the meeting has been planned already, the student can independently plan at which hours he will study the topic. The other activities in the schedule of the professor do not matter for the student's schedule. Similarly, the professor can make his own planning for the morning at this point, irrespective of the rest of the schedule of the student. Hence, decoupling a multi-agent scheduling problem into different local problems for the different agents can be done by first solving the constraints that describe dependencies between agents. An advantage of this approach is that agents can thereafter solve their local problems individually, irrespective of new choices of the other agents.

In some scheduling approaches, the goal is to find a schedule that satisfies all requirements. However, in many situations, only finding a schedule that matches the requirements is too restricted: The agents can prefer certain schedules to other ones, and hence, the goal becomes to find a solution that is preferred by the agents as much as possible.

Consider, for example, again the professor and the student who have to plan a meeting on a certain morning (between 8:00 and 12:00). Assume that the student also has to study the topic of the meeting himself. If the study takes place before the meeting, the meeting can be very efficient (15 minutes). On the other hand, if the study takes place after the meeting, the meeting will require slightly more time (30 minutes), but the study can be more effective and hence lasts much shorter (2 hours instead of 3 hours). Furthermore, assume that the professor has to give a lecture from 9:00 to 11:00.

We can imagine that the student prefers the longer meeting and subsequently the shorter study time to the longer study time and the shorter meeting, since the total working time of the first solution is less than the total working time of the second solution. On the other hand, the professor can have a preference for the shorter meeting, since that is less time-consuming for him. Furthermore, when the professor does not like to start working before 9:00, that can be an extra argument for choosing the shorter meeting, since the shorter meeting can be chosen to take place after the lecture, and the student can study the topic before the meeting, when the professor gives the lecture.

For this scheduling problem, an arbitrary solution that satisfies all constraints can be one in which a meeting of 30 minutes is planned from 8:00 to 8:30. However, the professor, not wanting to start early, can propose an improvement by suggesting to have the meeting from 8:35 to 8:55, such that he can start the lecture directly after the meeting at 9:00. This solution is not worse for the student, and hence is acceptable for both agents. Another option is for the professor to propose to have a short meeting after 11:00. This will be preferred by the professor, but results in the dispreferred longer study time for the student. However, the student can accept this option since it is much better for the professor and not that bad for the student himself. Furthermore, if the student makes sure that the meeting will be scheduled from 11:45 to 12:00, the student has some flexibility for planning the three hours of study before it. In any case, it is clear that taking the preferences of the agents into account for this problem is better than generating an arbitrary solution.

1.1 Research goals

In this thesis, we consider multi-agent scheduling problems for which the different agents can prefer certain schedules to other ones. In the literature, both single-agent scheduling problems with preferences and multi-agent scheduling problems without preferences have been reported, but, as far as we know, multi-agent scheduling problems with preferences have been given little attention. Since a general framework for such problems does not exist, our first goal is to develop a framework in which agents can specify preferences on their schedules.

Next, our goal is to develop methods for solving the temporal problems, taking the preferences of the agents into account. Due to the expected complexity of the problems, we propose to apply automated negotiation methods. An approach based on automated negotiation has three main advantages: First, automated negotiation encompasses both competitive and cooperative aspects, which is necessary for multi-agent scheduling: If the agents only stick to their own local optimal solutions, it is likely that no solution for all agents together will be found, due to interfering constraints. Hence, agents need to cooperate to find a solution. On the other hand, some competition is present since the agents try to get a benefit as high as possible for themselves. A second advantage of automated negotiation is that it more or less respects the privacy of the agents. Thirdly, automated negotiation can act as a computationally efficient distributed approach to efficiently explore large search spaces.

We propose to work with current decoupling solutions for multi-agent temporal problems, and improve them by automated negotiation of the agents. Our goal is to find modifications to original solutions that result in higher preference values for at least one agent, while not obtaining a lower preference value for one of the other agents. Hence, agents are cooperative if it does not disadvantage themselves.

A next goal is searching for solutions that improve maximal welfare: Agents can then accept a lower preference value than the one of the original solution, if the sum of the preference values of all agents increases with respect to the original solution. In this case, the agents are more cooperative and abandon some of their own utility for the collective.

1.2 Thesis outline

In Chapter 2, we describe the two relevant research fields of temporal problems and automated negotiation. First, we consider the different frameworks for temporal problems, to be able to indicate the lack of a framework for multi-agent temporal problems with preferences and to become familiar with the representations that we need for our new framework. Furthermore, current approaches on which we will build our approach will be considered here. Second, we give a brief overview of the field of automated negotiation, hereby describing the methods that we will use in our approach. We conclude the chapter with a specification of our research questions.

In Chapters 3 and 4, we develop and define the new frameworks for multi-agent temporal problems with preferences. Chapter 3 deals with an extension to the Multi-agent Simple Temporal Problem (MaSTP, defined in Section 2.1.4), where Chapter 4 extends the more powerful Multi-agent Disjunctive Temporal Problem (MaDTP, defined in Section 2.1.4). At the end of these chapters, we will be able to represent scheduling problems with preferences in an appropriate framework.

Chapters 5 and 6 describe the negotiation protocols that have been developed for solving the problems. Chapter 5 describes two negotiation protocols for solving the Multi-agent Simple Temporal Problem with Preferences (MaSTPP, defined in Chapter 3) and Chapter 6 continues with negotiation approaches for the Multi-agent Disjunctive Temporal Problem with Preferences (MaDTPP, defined in Chapter 4).

In Chapter 7, we describe the construction of a problem set on which the algorithms are tested, and we present the performance results. Finally, Chapter 8 gives an interpretation of the results, draws some conclusions, and suggests ideas for future research.

Chapter 2

Theoretical Background

A recent overview of multi-agent scheduling has been given by Agnetis, Billaut, Gawiejnowicz, Pacciarelli & Soukhal (2014). They characterize a multi-agent scheduling problem as a scheduling problem in which each agent has its own objective function, only based on its own tasks. This seems to correspond well with our theme of focus, but Agnetis et al. (2014) propose a mathematical approach and only give solutions for a specific set of problems in which the objective functions are regular, i.e., the objective functions are such that an earlier completion of a task is always preferred to a later completion. This might be suitable for optimization in completely automated situations such as factory processes, but is not applicable in all scheduling situations in which humans are involved: A person with three tasks to be finished in a morning, each requiring half an hour, of which the last can only start after 11:00, may prefer to arrive at 10:00 and do all tasks right after each other instead of arriving at 8:00 and have idle time in between the tasks. Hence, we focus on general objective functions instead of on regular ones. Another disadvantage of the approach of Agnetis et al. (2014) is that they use centralized methods to find solutions. When the number of agents increases, it would be advantageous, both from a computational and a privacy perspective, to use a distributed approach in which each agent computes its own part of a schedule, while coordinating with other agents if needed.

Another line of scheduling research, to which we will contribute, is based on the framework of Simple Temporal Problems (STPs) (Dechter, Meiri & Pearl, 1991). An STP (defined in Section 2.1.1) consists of temporal variables and constraints on them that put a bound on the temporal difference of two variables. Extensions of the STP with preference functions (Peintner & Pollack, 2004; Khatib et al., 2007) (see Section 2.1.3), and STPs for multiple agents (Boerkoel & Durfee, 2013b) (see Section 2.1.4) have been developed. For Multi-agent Simple Temporal Problems (MaSTPs), an important part of solving them is to find a temporal decoupling, i.e. a division of the MaSTP into local STPs, one for each agent, such that each solution to a local STP fits well within a global solution to the MaSTP and does not disturb any local STP of another agent (Boerkoel & Durfee, 2013b; Wilson, Klos, Witteveen & Huisman, 2014). Hence, distributed approaches are possible for MaSTPs. However, MaSTPs with different objective or preference functions for the different agents have not been studied before, as far as we know.

We therefore focus on developing frameworks of multi-agent temporal problems with preferences. In Section 2.1, we describe the existing frameworks for temporal problems related to the STP, to be able to base our frameworks on them.

Since solving the MaSTP and its generalized pendant (Boerkoel & Durfee, 2013a) is itself already complex, we do not focus on finding optimal solutions with respect to the preferences of the agents, but on improving already existing solutions by automated negotiation. Automated negotiation is an actual and promising research field (Baarslag, 2016; Fatima, Kraus & Wooldridge, 2015), that is chosen to be applied here for three reasons. Automated negotiation encompasses both competitive and cooperative aspects, it respects the privacy of the agents more or less, and it is a computationally efficient distributed approach.

We focus in more detail on automated negotiation in Section 2.2. Different negotiation approaches that can be useful in our research will be described.

Our approach will consist of a combination of the fields of temporal problems and automated negotiation. Earlier approaches that combine multi-agent scheduling and automated negotiation exist, but they use specific frameworks: Cohen, Cordeiro & Trystram (2015) apply algorithmic game theory to study the relation between cooperation and competition in multi-agent scheduling, but they only consider regular objective functions as Agnetis et al. (2014) do. Also Lang, Fink & Brandt (2016) propose a negotiation approach on scheduling problems with regular objective functions. They apply a negotiation protocol that is based on a simulated annealing heuristic. A comparable study is the one of Ramacher & Mönch (2016). They apply automated negotiation with a mediator on a very specific scheduling problem, in which two agents each have a predefined objective function.

Adhau, Mittal & Mittal (2012) apply an auction-based negotiation to a resource-constrained multi-project scheduling problem, in which agents can do bids for obtaining resources at certain time slots. Duan, Doğru, Özen & Beck (2012) describe an extensive automated negotiation approach for combinatorial optimization problems, but focus only on delivery schedules of two agents, a supplier and a manufacturer. However, they handle more complex objective functions than only regular ones.

The approaches described above combine the fields of multi-agent scheduling and automated negotiation, but work on rather limited problems. Contrary to previous research, we assume no specification of the preference functions beforehand, and we focus not only on systems with two agents, but also on problems with more than two agents. Hence, we propose a less restricted approach, and conclude this chapter with a presentation of our research questions regarding the combination of the two fields in Section 2.3.

2.1 Temporal problems

In this section, we consider the different frameworks of temporal problems that have been developed in recent years. We describe the representational capabilities of the frameworks, and the characteristics of the methods that have been developed to solve them will be discussed.

2.1.1 Simple Temporal Problems

A Simple Temporal Problem (STP) consists of a set $V = \{v_0, v_1, \cdots, v_{n-1}\}$ of temporal variables (where $z = v_0$ is a reference time point which will be assigned the value 0) and a set $C \subseteq \{c_{ij} \mid 0 \leq i, j \leq n-1, i \neq j\}$, where each c_{ij} is of the form $v_j - v_i \leq b_{ij}$ ($b_{ij} \in \mathbb{R} \cup \{\infty\}$), representing the temporal constraints on the variables in V (Dechter et al., 1991). As a notational convenience, c_{ij} and c_{ji} are combined into the constraint $-b_{ji} \leq v_j - v_i \leq b_{ij}$, or equivalently $v_j - v_i \in [-b_{ji}, b_{ij}]$. Giving such an interval is, for any two variables, always possible, since a constraint c_{ij} with bound $b_{ij} = \infty$ can be added to C without changing the problem.

Example 2.1. *Consider a student who on a morning has to study a certain topic for approximately three hours, and subsequently has to talk with a professor for approximately a quarter of an hour. We can model this as an STP with a set of five time points,* $V = \{z, S_S, S_E, M_S, M_E\}$*, which are the reference time point, the start and end time of studying, and the start and end time of the meeting, respectively.*

The constraints are defined as follows: since all activities need to be done in the morning, let the reference time point represent 8:00 and set bounds $0 \leq S_S - z \leq 240$, $0 \leq S_E - z \leq 240$, $0 \leq M_S - z \leq 240$*, and* $0 \leq M_E - z \leq 240$ *(the values represent minutes in our examples), such that all the activities will be restricted to occur between 8:00 and 12:00. Furthermore, since the studying will require approximately three hours (let between 175 and 185 minutes be allowed), we add the constraint* $175 \leq S_E - S_S \leq 185$*, and equivalently, for the meeting time we add* $10 \leq M_E - M_S \leq 20$*. To represent the requirement that the meeting takes place after the student's study, we add the constraint* $S_E - M_S \leq 0$*, which can also be represented as* $0 \leq M_S - S_E \leq \infty$*.*

An STP $\mathcal{S} = \langle V, C \rangle$ can be represented as a weighted, directed graph $\mathcal{G}_\mathcal{S} = \langle N, E, w \rangle$, where $N = V$, i.e. the temporal variables of $\mathcal{S}$ are represented as nodes, and $E = \{\langle v_i, v_j \rangle \mid c_{ij} \in C\}$ with $w(\langle v_i, v_j \rangle) = b_{ij}$, i.e. the constraints are represented as directed weighted edges, such that an edge from v_i to v_j has weight b_{ij}, representing the constraint $v_j - v_i \leq b_{ij}$.[1] The graph $\mathcal{G}_\mathcal{S}$ for the STP $\mathcal{S}$ is called a Simple Temporal Network (STN).

Example 2.2. *The STN for the STP in Example 2.1 is shown in Figure 2.1. As a notational shorthand, an edge from v_i to v_j with interval label $[-b_{ji}, b_{ij}]$ represents both the edge from v_i to v_j with bound b_{ij}, and the edge from v_j to v_i with bound b_{ji}. E.g. the constraint $175 \leq S_E - S_S \leq 185$ is represented by the edge from S_S to S_E with interval label $[175, 185]$. Furthermore, the time point $v_0 = z$ is not explicitly shown, but the edges from z to v_i and v_i to z are represented by a self-loop on v_i with interval label $[-b_{i0}, b_{0i}]$. For clarity, we also use concrete times instead of bounds in minutes for these edges. The interval label* [8:00, 12:00] *is in this example a more readable notation for* $[0, 240]$.

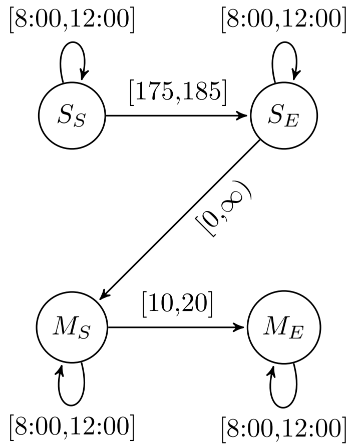


Figure 2.1: STP for two consecutive tasks.

A solution σ to an STP $\mathcal{S} = \langle V, C \rangle$, called a *schedule*, is an assignment of real values to the temporal variables $V \setminus \{z\}$ such that all constraints in C are satisfied. In many cases, however, a flexible solution, consisting of intervals instead of values for the temporal variables, is more desirable because it is more resistant to disturbances (Wilson et al., 2014). An *interval schedule* is an assignment of intervals to the temporal variables in such a way that choosing any value in an interval does not violate any constraint in C. If an STP does not have any solution, it is said to be *inconsistent*.

Example 2.3. *Consider the example in Figure 2.1. Two possible solutions σ_1 and σ_2 for this problem are the assignments given by $\sigma_1(S_S) =$ 8:10, $\sigma_1(S_E) =$ 11:10, $\sigma_1(M_S) =$ 11:10, $\sigma_1(M_E) =$ 11:25 and $\sigma_2(S_S) =$ 8:51, $\sigma_2(S_E) =$ 11:47, $\sigma_2(M_S) =$ 11:48, $\sigma_2(M_E) =$ 11:58.*

Solving the STP can be done in $\mathcal{O}(n^3)$ time by applying an all-pairs-shortest-path algorithm (Floyd, 1962) to its STN, to obtain Full Path Consistency (FPC). FPC on an STN means that for all pairs of nodes, an edge between the nodes exists, with the tightest possible weight. The algorithm is as follows: For each set of three temporal variables, the weight w_{ij} between v_i and v_j (initially b_{ij}) will be updated by assigning the minimum of w_{ij} (or ∞ if not defined) and $w_{ik} + w_{kj}$ to it, such that shortest paths are established between each pair of vertices. Since the constraints that form a path in the corresponding graph can be combined to form a new constraint with the sum of the original bounds as its bound ($c_{ij} : v_j - v_i \leq b_{ij}$ and $c_{jk} : v_k - v_j \leq b_{jk}$ can be summed to get $c_{ik} : v_k - v_i \leq b_{ij} + b_{jk}$), there is a direct relationship between shortest paths in the graph and tightest constraints of the STP.

[1]In the remainder of this thesis, we write e_{ij} for an edge $\langle v_i, v_j \rangle \in E$ and w_{ij} for its weight $w(\langle v_i, v_j \rangle)$.

A computationally more efficient way to solve the STP uses a triangulated network instead of a fully connected network. A triangulated graph is a graph without non-bisected cycles of which the length is more than 3. (The graph in Figure 2.1 is, for example, a triangulated graph, but the graph in Figure 2.3 is not triangulated, since it has a cycle of length 4.) A graph can be triangulated by handling the vertices in a certain *elimination order*; for each vertex, before eliminating it, a *fill edge* will be added between each pair of its not yet eliminated neighbors if no such an edge exists.

On a triangulated graph, Partial Path Consistency (PPC), giving the tightest bounds only for the edges present in the triangulated graph, can be established in less than $\mathcal{O}(n^3)$ time. For this, the nodes will first be handled in order of elimination and the edges of their neighbors will be tightened to establish Directional Path Consistency (DPC);[2] subsequently, updates of the edges will be done in reverse elimination order. Planken, De Weerdt & Van der Krogt (2008) innovated the $\mathrm{P}^3\mathrm{C}$ algorithm that establishes PPC on an STP in $\mathcal{O}(n \cdot {\omega_o^*}^2)$ time, where ω_o^* is the *induced width* relative to elimination order o and is defined to be the maximum (over all vertices v_i) of the number of neighbors of v_i that appear after v_i in o, which is generally much less than n.

From Full Path Consistent and Partial Path Consistent STPs, a solution to an STP can easily be derived.

2.1.2 Disjunctive Temporal Problems

For many scheduling cases, the STP is not sufficient to represent the problem. For example, consider two tasks that can be done in any order, but not at the same time. To represent this problem as an STP, an ordering of the tasks should be chosen beforehand, or the STP will allow the tasks to be done simultaneously. To overcome this problem, the Disjunctive Temporal Problem (DTP) is used.

The DTP, introduced by Stergiou & Koubarakis (2000), consists of a set V of time points, as in the STP, and a set C of disjunctive constraints. DTP constraints are of the form $d_1 \vee d_2 \vee \cdots \vee d_k$ ($k \geq 1$), where each disjunct d_x is an STP constraint, i.e. a temporal constraint of the form $-b_{ji} \leq v_j - v_i \leq b_{ij}$. Given a DTP $\mathcal{D} = \langle V, C\rangle$, a *component STP* is an STP $\mathcal{S} = \langle V, \hat{C}\rangle$ that results from selecting one disjunct from each of the disjunctive constraints, i.e. $\forall c \in C\ \exists x$ s.t. $d_x \in c \cap \hat{C}$.[3] An assignment σ of values to the temporal variables $V \setminus \{z\}$ is a solution to the DTP if and only if it is a solution to a component STP of the DTP.

Example 2.4. *Consider the situation of Example 2.1. If we want to allow the student to have the meeting before studying, we can use the DTP framework, and replace the single constraint* $0 \leq M_S - S_E \leq \infty$ *by the disjunctive constraint* $0 \leq M_S - S_E \leq \infty \vee 0 \leq S_S - M_E \leq \infty$*. See Figure 2.2.*

Solving the DTP is much more complex than solving the STP: Finding a solution to the DTP is known to be an NP-complete problem (Boerkoel & Durfee, 2012). In general, the number of possible component STPs resulting from choosing disjuncts can be $k^{|C|}$, for k the maximum number of disjuncts in a constraint. Approaches for DTP solving generally search for a consistent component STP, rather than for a specific solution to the DTP. (Subsequently, a consistent component STP can be solved in polynomial time, as described in Section 2.1.1.) With respect to the STP as a Constraint Satisfaction Problem (CSP), finding component STPs for a DTP can be seen as a meta-CSP, where the constraints are not explicit, but are given in the temporal constraints of the selected disjuncts (Tsamardinos & Pollack, 2003). Different techniques are used to improve general backtracking algorithms for DTP solving, among others backjumping (going back directly to the last assigned variable (disjunct) that directly relates to an inconsistency), removal of subsumed variables (disjunctions that are inherently satisfied by the choice of some other disjuncts) and (incremental) forward checking (Tsamardinos & Pollack, 2003; Oddi, 2014).

[2] A DPC algortihm for an STN checks for consistency of the STN and assigns weights in such a way that a solution can be obtained by handling the nodes in reverse elimination order.

[3] Note that a disjunctive constraint c is represented in formal descriptions in this thesis as the set of its disjuncts $\{d_1, \cdots, d_k\}$. In this chapter, sometimes a double index is used for a disjunct to represent over which nodes the constraint is specified: d_{ij} then represents the constraint $-b_{ji} \leq v_j - v_i \leq b_{ij}$.

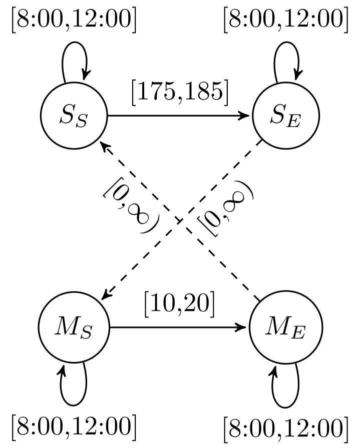


Figure 2.2: DTP for two serial tasks without prescribed order.

Note that the DTP belongs to the category of *satisfiability modulo theories* (SMT) problems (Barrett, Sebastiani, Seshia & Tinelli, 2009; Barrett & Tinelli, 2017), since it can be rewritten as a formula in first-order logic with predicates that are linear inequalities for which a consistent assignment has to be determined. DTPs therefore can also be solved by general SMT solvers (e.g. Yices (Dutertre, 2014), MathSAT5 (Cimatti, Griggio, Schaafsma & Sebastiani, 2013), Z3 (De Moura & Bjørner, 2008)), which are still improved. We will use a state-of-the-art SMT-solver for solving DTPs in our process of solving MaDTPPs (see Chapter 6).

2.1.3 Temporal Problems with Preferences

Solving DTPs is generally searching for one satisfying schedule. However, in many cases, it is reasonable to assume that some disjuncts are preferred to other disjuncts. To handle this, the Disjunctive Temporal Problem with Preferences (DTPP) has been developed (Peintner & Pollack, 2004), in which some preference values are added to the possible temporal difference values. To each disjunct $-b_{ji} \leq v_j - v_i \leq b_{ij}$, a preference function $f_{ij} : [-b_{ji}, b_{ij}] \to [0, \infty)$ can be added that maps each value for the difference to a preference value. The preference value for a disjunctive constraint will be the maximum of the preference values that are obtained by its disjuncts.[4]

Moffitt (2011) uses a Valued Disjunctive Temporal Problem (VDTP) instead of the DTPP, in which there are both hard constraints that need to be satisfied, and soft constraints that allow violations to a certain cost. In this case, there are no preference functions added to the separate disjuncts of a constraint, but cost values are added to each soft constraint as a whole. This can be modeled by a cost function $f : C \to [0, \infty)$ which maps each constraint to a cost value for violating the constraint. Hard constraints are mapped to ∞. Moffitt (2011) shows that for each VDTP for which the preference functions are piecewise constant, there exists an equivalent DTPP and vice versa.

Example 2.5. *Consider the situation of Example 2.4. Assume that the student prefers to have the meeting after the study, and that he wants to study as long as possible. The DTPP can be constructed as follows: for the constraint* $175 \leq S_E - S_S \leq 185$, *we can add the preference function* $f(S_S, S_E) = (S_E - S_S) - 175$ *such that a value of 185 for* $S_E - S_S$ *has a preference value of 10 and a value of 175 for* $S_E - S_S$ *has a preference value of 0; for the disjunctive constraint* $0 \leq M_S - S_E \leq \infty \vee 0 \leq S_S - M_E \leq \infty$ *we can take the preference functions* $f(S_E, M_S) = 2$ *and* $f(M_E, S_S) = 1$, *such that the first disjunct is preferred to the second, regardless of the exact values of the temporal differences.*

[4]Note that the Simple Temporal Problem with Preferences (STPP) is a special case of the DTPP in which each constraint consists of only one disjunct. See also Khatib et al. (2007).

The VDTP case can be constructed by letting the disjunctive constraint $c_1 : 0 \leq M_S - S_E \leq \infty \vee 0 \leq S_S - M_E \leq \infty$ *be a hard constraint* ($f(c_1) = \infty$) *and adding two soft constraints* $c_2 : 0 \leq M_S - S_E \leq \infty$ *and* $c_3 : 0 \leq S_S - M_E \leq \infty$ *with cost values* $f(c_2) = 2$ *and* $f(c_3) = 1$*, respectively. Also, beside the hard constraint* $c_4 : 175 \leq S_E - S_S \leq 185$*, we can create for example the soft constraints* $c_5 : 175 \leq S_E - S_S < 180$ *and* $c_6 : 180 \leq S_E - S_S \leq 185$ *with cost values* $f(c_5) = 1$ *and* $f(c_6) = 2$*, such that violating* c_6 *will be less appreciated.*

Different solution techniques for DTPPs and VDTPs have been proposed (e.g. Peintner, 2005; Moffitt, 2011; Maratea & Pulina, 2014). Moffitt (2011) describes that most DTP approaches start by selecting disjuncts such that a feasible component STP remains, which in turn can be solved. With the VDTP approach, this is still possible, since the preference values for a feasible component STP are known. In contrast, for the DTPP approach the preference values are not known until exact values for the time points are established. Note that some state-of-the-art SMT solvers are able to deal with optimization of an objective function, which makes them appropriate DTPP solvers (Bjørner, Phan & Fleckenstein, 2015; Sebastiani & Trentin, 2015).

Micalizio & Torta (2015) notice that in these approaches, preferences on different constraints are independent of one another. They describe an approach in which preferences on constraints are dependent on the values of other constraints.

2.1.4 Multi-agent Temporal Problems

When different agents are responsible for different sets of temporal variables that have to be adjusted to one another, distributed versions of the STP and DTP frameworks can be used. In general, each agent has its own local STP or DTP subproblem, and the different subproblems are linked by inter-agent constraints. We give an overview of the frameworks of the Multi-agent Simple Temporal Problem (MaSTP) (Boerkoel & Durfee, 2010) and the Multi-agent Disjunctive Temporal Problem (MaDTP) (Boerkoel & Durfee, 2012). Subsequently, we describe in detail the solution techniques of Boerkoel & Durfee (2013b,a) for these problems, since we build on them in Chapters 5 and 6.

Multi-agent Simple Temporal Problems

In the MaSTP, a set $G = \{A, B, C, \cdots\}$ describes m different agents that are responsible for their particular time points. Furthermore, V and C are the set of temporal variables and the set of constraints as in the STP, but each time point (except z) is assigned to exactly one of the m agents. Formally, $\{V_a\}_{a=1}^m$ is a partition of $V \setminus \{z\}$, and each set V_a corresponds to the temporal variables agent a is responsible for. Constraints are only known to agent a if at least one of the two temporal variables mentioned in it belongs to agent a. Formally, we define the following sets, slightly adapted from Boerkoel & Durfee (2013b):

- $V_L^a = V_a \cup \{z\}$ is the set of agent a's *local variables*, which are (except z) assignable by agent a.
- $C_L^a = \{c_{ij} \in C \mid v_i, v_j \in V_L^a\}$ is the set of agent a's *local constraints*, defined over local variables.
- $C_X^a = \{c_{ij} \in C \mid \exists b \neq a \in G \ ((v_i \in V_L^a \setminus \{z\} \wedge v_j \in V_L^b \setminus \{z\}) \vee (v_i \in V_L^b \setminus \{z\} \wedge v_j \in V_L^a \setminus \{z\}))\}$ is the set of agent a's *external constraints*, in which exactly one of the two involved time points belongs to agent a, and the other to another agent b.
- $V_X^a = \{v_j \in V \mid \exists v_i \in V_L^a \ (c_{ij} \in C_X^a)\}$ is the set of agent a's *external variables*, that are local to some other agent, but connected with an external constraint to a local variable of agent a.

We define the *local STP subproblem* of agent a as $\mathcal{S}_L^a = \langle V_L^a, C_L^a \rangle$. Some more definitions that we need for reasoning about the MaSTP are the following:

- $V_S = \bigcup_{i=1}^{m} V_X^i \cup \{z\}$ is the set of *shared variables* in the MaSTP, i.e. the set of variables that are involved in external constraints (added with z).
- $V_I^a = V_L^a \cap V_S$ is the set of agent a's *interface variables*, i.e. the local variables of the agent that are external variables for some other agent (added with z).
- $V_P^a = V_L^a \setminus V_S$ is the set of agent a's *private variables*, i.e. the local variables of the agent that are not external variables for some other agent.

An agent a is aware of its own local variables V_L^a (including the reference time point z) and its external variables V_X^a. No other time points of other agents are known to agent a. It follows that the private time points of an agent are not known by any other agent.

Example 2.6. *In our running example, we gave the student the freedom to construct his schedule (including the start and end time of the meeting) autonomously, regardless of the availablity of the professor. It is obvious that this is too simplistic. We expand the situation of Example 2.1 to an MaSTP by modeling also the schedule of the professor.*

Assume that the professor has to give a lecture from 9:00 to 11:00 before the meeting with the student, and will be present from 8:30. Then, our MaSTP will consist of the set $G = \{A, B\}$ *of agents (where agent A represents the student and agent B the professor), and the set* $V = \{z, S_S^A, S_E^A, M_S^A, M_E^A, M_S^B, M_E^B, L_S^B, L_E^B\}$ *of time point variables, where* S_S^A*,* S_E^A*,* M_S^A*, and* M_E^A *are defined as before but are now explicitly ascribed to agent A,* M_S^B *and* M_E^B *are the meeting start and end time in the schedule of agent B, and* L_S^B *and* L_E^B *represent the begin and end of the professor's lecture. The set C contains the original constraints* $0 \leq S_S^A - z \leq 240$*,* $0 \leq S_E^A - z \leq 240$*,* $0 \leq M_S^A - z \leq 240$*,* $0 \leq M_E^A - z \leq 240$*,* $175 \leq S_E^A - S_S^A \leq 185$*,* $10 \leq M_E^A - M_S^A \leq 20$*, and* $0 \leq M_S^A - S_E^A \leq \infty$*. Furthermore, we add to C the constraints* $30 \leq M_S^B - z \leq 240$*,* $30 \leq M_E^B - z \leq 240$*, and* $10 \leq M_E^B - M_S^B \leq 20$ *for B's meeting restrictions,* $60 \leq L_S^B - z \leq 60$*,* $180 \leq L_E^B - z \leq 180$*, and* $120 \leq L_E^B - L_S^B \leq 120$ *for B's lecture,*[5] *and* $0 \leq M_S^B - L_E^B \leq \infty$ *to represent the fact that the meeting has to take place after the lecture. Also, we add* $0 \leq M_S^B - M_S^A \leq 0$ *and* $0 \leq M_E^B - M_E^A \leq 0$ *to ensure that the meeting times for the professor and the student are synchronized. See the graph representation in Figure 2.3.*

Note that the local variables of agents A and B are the sets $V_L^A = \{S_S^A, S_E^A, M_S^A, M_E^A, z\}$ *and* $V_L^B = \{M_S^B, M_E^B, L_S^B, L_E^B, z\}$ *respectively, that the external constraints* $C_X^A = C_X^B = \{0 \leq M_S^B - M_S^A \leq 0,\ 0 \leq M_E^B - M_E^A \leq 0\}$ *are the same for both agents, and that for both agents, the external variables are only the meeting time points of the other agent:* $V_X^A = \{M_S^B, M_E^B\}$ *and* $V_X^B = \{M_S^A, M_E^A\}$*. The set of shared variables is given by* $V_S = \{M_S^A, M_E^A, M_S^B, M_E^B, z\}$*, and interface and private variables are given by the sets* $V_I^A = \{M_S^A, M_E^A, z\}$*,* $V_P^A = \{S_S^A, S_E^A\}$*,* $V_I^B = \{M_S^B, M_E^B, z\}$*, and* $V_P^B = \{L_S^B, L_E^B\}$*. Hence, from the MaSTP,* S_S^A *and* S_E^A *are not known to agent B, and* L_S^B *and* L_E^B *are not known to agent A.*

A solution σ to an MaSTP is, similar to a solution to an STP, an assignment of real values to the temporal variables $V \setminus \{z\}$ such that all constraints in C are satisfied, regardless of the agents owning the variables and knowing the constraints. However, a much more interesting problem is to find a *temporal decoupling* for an MaSTP. Intuitively, the goal of a temporal decoupling is to obtain a local STP for each agent, such that these local STPs can be solved by the agents independently, irrespective of what choices the other agents make. Hence, the STPs resulting from a temporal decoupling can be solved in parallel.

Definition 2.1. A temporal decoupling of an MaSTP (Hunsberger, 2002a; Boerkoel & Durfee, 2013b, p. 111) is defined to be a set of local STP subproblems $\{\mathcal{S}_{L+\Delta}^1, \cdots, \mathcal{S}_{L+\Delta}^m\}$ such that

- $\mathcal{S}_{L+\Delta}^1, \cdots, \mathcal{S}_{L+\Delta}^m$ are all consistent; and
- aggregating any combination of locally consistent solutions to $\mathcal{S}_{L+\Delta}^1, \cdots, \mathcal{S}_{L+\Delta}^m$ gives a solution to the original MaSTP.

[5]Note that the constraint $120 \leq L_E^B - L_S^B \leq 120$ could be omitted since it is implied by the constraints that represent the start and end time of the lecture, but it is also fine to include it.

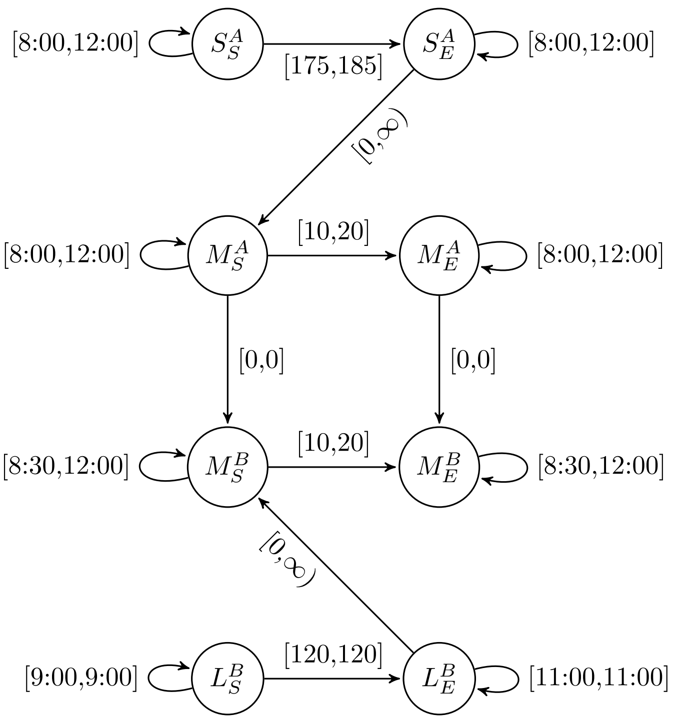


Figure 2.3: MaSTP for two agents (A and B) with two tasks each, from which the meeting times (start times M_S^A and M_S^B and end times M_E^A and M_E^B) need to be synchronized.

Formally, we need to find an extension of appropriate constraints C_Δ^a to C_L^a for each agent, such that the resulting local subproblems $\mathcal{S}_{L+\Delta}^a = \langle V_L^a, C_L^a \cup C_\Delta^a \rangle$ form a decoupling of the original MaSTP. A *minimal decoupling* is one in which no constraint can be relaxed or removed without making it no longer a decoupling.

The advantage of a temporal decoupling is that the resulting different subproblems can be solved independently and in parallel by the different agents: unless constraints are changing during run time, there is no longer a need to take into account the constraints that involve temporal variables of the other agents. Both from a computational complexity and from a privacy perspective, this can be very advantageous. Methods for finding a temporal decoupling are described below. (Note that the centralized FPC and PPC methods described before (Section 2.1.1) also solve the MaSTP, but cancel out both the computational and the privacy advantages.)

Example 2.7. *We consider again the MaSTP of Example 2.6 (Figure 2.3). A temporal decoupling can be obtained by restricting the bounds on the shared variables. If we set for example* $C_\Delta^A = \{210 \leq M_S^A - z \leq 210,\ 225 \leq M_E^A - z \leq 225\}$ *and* $C_\Delta^B = \{210 \leq M_S^B - z \leq 210,\ 225 \leq M_E^B - z \leq 225\}$*, then any solution to* $\mathcal{S}_{L+\Delta}^A$ *can be combined with any solution to* $\mathcal{S}_{L+\Delta}^B$*, resulting in a consistent schedule for the original MaSTP. Hence, agent A can at this point schedule* S_S^A *and* S_E^A *independently of agent B, when taking into account* C_Δ^A*. Hence, agent A has to schedule* S_E^A *before 11:30 in this case.*

Multi-agent Disjunctive Temporal Problems

The MaDTP resembles the MaSTP, but consists of local DTP subproblems instead of local STP subproblems. Again, we have a set $G = \{A, B, C, \cdots\}$ of m agents, a set V of temporal variables such that the partition $\{V_a\}_{a=1}^m$ of $V \setminus \{z\}$ describes for which temporal variables each agent is responsible, and a set C of constraints, this time consisting of disjunctive constraints. Local and external variables and constraints are defined as follows, where we represent a disjunctive constraint c as the set of its disjuncts $\{d_{ij} : -b_{ji} \leq v_j - v_i \leq b_{ij}\}$:

- $V_L^a = V_a \cup \{z\}$, as in the MaSTP.
- $C_L^a = \{c \in C \mid \forall d_{ij} \in c \ (v_i, v_j \in V_L^a)\}$, the set of constraints of which all disjuncts are defined over local variables only.
- $C_X^a = \{c \in C \mid \exists b \neq a \in G \ \exists d_{ij}, d_{\hat{\imath}\hat{\jmath}} \in c \ ((v_i \in V_L^a \setminus \{z\} \vee v_j \in V_L^a \setminus \{z\}) \wedge (v_{\hat{\imath}} \in V_L^b \setminus \{z\} \vee v_{\hat{\jmath}} \in V_L^b \setminus \{z\}))\}$, the set of constraints of which at least one disjunct is defined over a variable local to a, and at least one disjunct is defined over a variable local to another agent b.
- $V_X^a = \{v_i \in V \setminus V_L^a \mid \exists c \in C_X^a \ \exists v_j \in V \ (d_{ij} \in c)\}$, the set of variables that are local to some other agent, but involved in an external constraint of agent a.

The *local DTP subproblem* of agent a is, analogous to the local STP subproblem, defined to be $\mathcal{D}_L^a = \langle V_L^a, C_L^a \rangle$. Furthermore, shared variables, interface variables and private variables are defined as in the MaSTP. The definition of a temporal decoupling is also equivalent to that of an MaSTP:

Definition 2.2. A temporal decoupling of an MaDTP (Boerkoel & Durfee, 2013a, p. 125) is defined to be a set of local DTP subproblems $\{\mathcal{D}_{L+\Delta}^1, \cdots, \mathcal{D}_{L+\Delta}^m\}$ such that

- $\mathcal{D}_{L+\Delta}^1, \cdots, \mathcal{D}_{L+\Delta}^m$ are all consistent; and
- aggregating any combination of locally consistent solutions to $\mathcal{D}_{L+\Delta}^1, \cdots, \mathcal{D}_{L+\Delta}^m$ gives a solution to the original MaDTP.

Hence, for decoupling an MaDTP, we need to find an extension of appropriate constraints C_Δ^a to C_L^a for each agent, such that the resulting local subproblems $\mathcal{D}_{L+\Delta}^a = \langle V_L^a, C_L^a \cup C_\Delta^a \rangle$ form a decoupling of the original MaDTP.

Example 2.8. *When we extend the situation of Example 2.6 by allowing different orders for the tasks, we get the MaDTP shown in Figure 2.4. Instead of the constraints* $0 \leq M_S^A - S_E^A \leq \infty$ *and* $0 \leq M_S^B - L_E^B \leq \infty$*, we now have the constraints* $0 \leq M_S^A - S_E^A \leq \infty \vee 0 \leq S_S^A - M_E^A \leq \infty$ *and* $0 \leq M_S^B - L_E^B \leq \infty \vee 0 \leq L_S^B - M_E^B \leq \infty$*. The decoupling constraints* C_Δ^A *and* C_Δ^B *from Example 2.7 also establish a temporal decoupling for the MaDTP. Another decoupling for this MaDTP can be obtained by setting* $C_\Delta^A = \{35 \leq M_S^A - z \leq 35,\ 55 \leq M_E^A - z \leq 55\}$ *and* $C_\Delta^B = \{35 \leq M_S^B - z \leq 35,\ 55 \leq M_E^B - z \leq 55\}$*.*

Solutions for Multi-agent Temporal Problems

Different approaches to decouple the MaSTP have been developed. The first algorithms for solving them were given by Hunsberger (2002a,b). Later on, Planken, De Weerdt & Witteveen (2010) studied the temporal decoupling problem, and found that a decoupling for a certain constraint system can be found in polynomial time if and only if a solution for the problem can be found in polynomial time. This implies that a temporal decoupling for the MaSTP can be found quite efficiently (since the STP can be solved in polynomial time, see Section 2.1.1), in constrast to a decoupling for the MaDTP (since the MaDTP cannot be solved in polynomial time, see Section 2.1.2). Furthermore, Planken et al. (2010) investigated whether an optimal decoupling could be found when some quality metric, comparable to the preferences in the DTPP, was applied to an STP. They show that the general optimal decoupling problem for (Ma)STPs is NP-hard; however, it can be solved in polynomial time if the objective function is linear.

In most recent literature, there is a search for decouplings of MaSTPs that take into account the flexibility of the different agents (e.g. Wilson, 2016). When disturbances occur during execution of the schedule, it is desirable that this does not influence constraints on other time points. Intuitively, when variables can be chosen freely in a corresponding interval, there will be some flexibility in the decoupling. On the other hand, a decoupling in which each time point is fixed has no flexibility at all. Wilson et al. (2014) define a flexibility metric for STPs that is more accurate than previous flexibility metrics (Hunsberger, 2002a) since it does not overestimate the flexibility, and show that this metric leads to an optimal decoupling for MaSTPs, without loss of flexibility

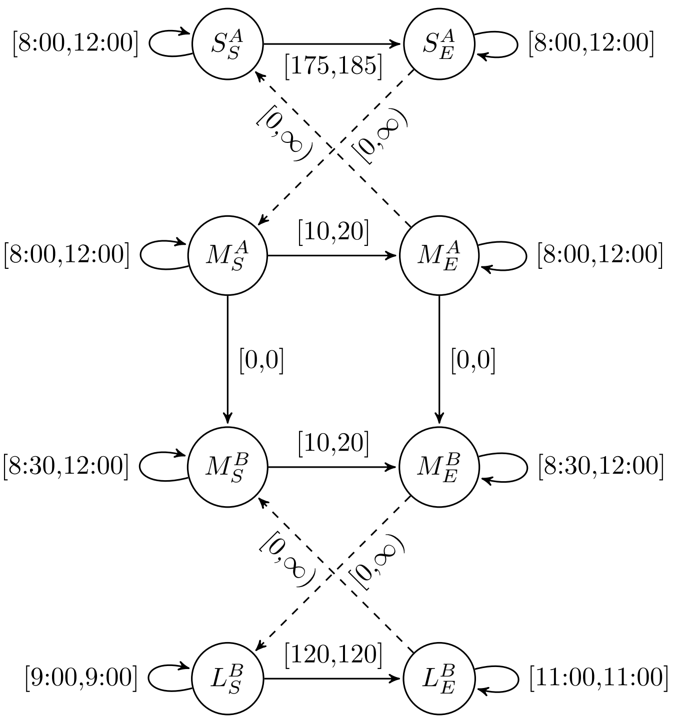


Figure 2.4: MaDTP for two agents (A and B) with two tasks each, from which the meeting times (start times M_S^A and M_S^B and end times M_E^A and M_E^B) need to be synchronized. The individual order of tasks is not prescribed.

with respect to the original problem.

A distributed algorithm for MaSTPs that respects the privacy of the agents but does not take optimality into account has been developed by Boerkoel & Durfee (2013b). However, an elementary heuristic for increasing the flexibility has been implemented. For MaDTPs, the distributed decoupling approach of Boerkoel & Durfee (2013a) is, for all we know, the only one available in the literature. Since we will build on the algorithms of Boerkoel & Durfee later on, we describe them in some more detail. See for an exact description the original papers.

For decoupling the MaSTP and MaDTP, Boerkoel & Durfee (2013b) use distributed forms of Directional Path Consistency and Partial Path Consistency (see Section 2.1.1). First, we describe their algorithm for obtaining Distributed Partial Path Consistency. Subsequently, we describe their adaptations to obtain the temporal decouplings. Note that we use the graph representation of the problems here.

Globally, the algorithm for Distributed Triangulating Partial Path Consistency (D△PPC) is for each agent as follows:

- **Eliminate private time points** First, an agent applies the Triangulating Directional Path Consistency (△DPC) algorithm on its local STP subproblem. This algorithm generates a local elimination order on the private nodes of the agent using a minimum fill heuristic, i.e. each time selecting a private node that adds fewest fill edges. Then, for all pairs of non-eliminated neighbors v_i and v_j of the eliminated node v_k, a fill edge is added if no such edge exits, and weights are updated by $w_{ij} \leftarrow \min(w_{ij}, w_{ik} + w_{kj})$ and $w_{ji} \leftarrow \min(w_{ji}, w_{jk} + w_{ki})$. This establishes DPC on a triangulated version of the local STP subproblem, which is necessary for the △PPC algorithm of the fourth step.

- **Eliminate shared time points** Next, the Distributed Triangulating Directional Path Consistency (D△DPC) algorithm is performed: after doing △DPC, a similar process is done for all shared variables. A shared elimination order is constructed on a first-come first-served basis, and for each node (after checking for update messages from other agents) the same updates as before are done for all not-eliminated neighbors in the set of shared nodes. These updates are also sent to the agents whose nodes are involved in them. By this, DPC is established on the global multi-agent network.

- **Reinstate shared time points** At this point, the shared part of the PPC algorithm can start. In reverse shared elimination order, shared time points v_k are selected, and neighboring edges to nodes later in the shared elimination order (v_i and v_j) are updated using the edge between v_i and v_j. This is done by the rules $w_{ik} \leftarrow \min(w_{ik}, w_{ij}+w_{jk})$, $w_{ki} \leftarrow \min(w_{ki}, w_{kj}+w_{ji})$, $w_{jk} \leftarrow \min(w_{jk}, w_{ji}+w_{ik})$, and $w_{kj} \leftarrow \min(w_{kj}, w_{ki}+w_{ij})$. Also here, update messages are sent to the appropriate agents, and agents will wait for the messages they need before updating.

- **Reinstate private time points** The last part for establishing PPC is to apply the △PPC algorithm on the local network, based on the local elimination order constructed during △DPC. This part is similar to that of the previous step, but without the messages. By now, PPC is established on a triangulated version of the original global MaSTP.

For an MaSTP decoupling, obtaining PPC is not enough. Strict separations or synchronizations need to be made between time points of different agents in order to give them the freedom to construct their local schedules individually, without influencing the schedules of other agents. Therefore, the reinstatement of shared time points (the third step of D△PPC) is replaced by a step in which shared time points are reinstated and assigned, and an optional step in which the domains are relaxed to obtain a minimal decoupling. These replacements form the Multi-agent Temporal Decoupling (MaTD) algorithm with optional relaxation (MaTDR). The two steps that replace the third step of D△PPC are as follows:

- **Reinstate and assign shared time points** Shared time points v_k are selected in reversed shared elimination order. For each neighbor v_j later in the elimination order, after checking for update messages, the domain of v_k (i.e. the constraints between v_k and the reference time point) is updated by the rules $w_{zk} \leftarrow \min(w_{zk}, w_{zj} + w_{jk})$ and $w_{kz} \leftarrow \min(w_{kz}, w_{kj} + w_{jz})$. Then, a value is assigned to node v_k using some heuristic, e.g. the midpoint of the current domain $[-w_{kz}, w_{zk}]$, messages are sent to the appropriate agents, and the domain for v_k is added to the set of decoupling constraints.

- **(Optional) Relax shared time point domains** Relaxation of assignments to shared time points is done in original elimination order. Based on the original MaSTP constraints, the bounds before reinstating shared time points, and the decoupling constraints from the previous step, it is checked whether a relaxation (consistent with all neighbor nodes) is possible.

Boerkoel & Durfee (2013b) claim that their MaTD algorithm is sound and complete for decoupling an MaSTP (i.e. the decoupling constraints given by the MaTD algorithm yield indeed a temporal decoupling of the original MaSTP, and given a consistent MaSTP, the algorithm is guaranteed to find a decoupling), and that the MaTDR algorithm yields a minimal decoupling for such a problem. Furthermore, the (local) time complexity is $\mathcal{O}((n_P + n_S) \cdot \omega_o^{*2})$, where n_P is the maximum number of private variables any agent has, n_S is the number of shared variables in the MaSTP, and ω_o^* is the induced width relative to the elimination order o (see Section 2.1.1).

For decoupling the MaDTP, Boerkoel & Durfee (2013a) make use of the MaTD algorithm described before. In general, agents individually compute the influence spaces of their component STPs and send these to a central coordinator. The coordinator tries to combine the influence spaces of the different agents to find a solution to the shared part of the MaDTP. This will be sent to the agents, and they can base their own local schedules on it.

In some more detail, the MaDTP Temporal Decoupling (MaDTP-TD) algorithm is as follows. For a complete description, we refer to the original paper.

- **Preprocessing** First, an agent preprocesses its local DTP. All constraints that consist of only one disjunct form together an STP that needs to be satisfied and is part of the larger DTP. Hence, bound tightening on these constraints can be done using the D$\triangle$PPC algorithm. Furthermore, any disjunct that is inconsistent with respect to these tightened constraints can be removed, and any subsumed constraint (i.e. a constraint that has an inherently satisfied disjunct) can also be safely removed.

- **Enumerating influence spaces of component STPs** Next, each agent uses an existing DTP solving approach to compute component STPs of its local DTP (e.g. Stergiou & Koubarakis, 2000; Tsamardinos & Pollack, 2003). From these, only the ones that result in different influence spaces (i.e., have different constraints on their interface variables) are sent to the coordinator, since only different influence spaces can give other possibilities for the shared DTP. When the coordinator finds a solution, the agent stops sending influence spaces to the coordinator.

- **Combining solutions** When an agent receives the interface variables part of the component STP that yields a shared solution, it will apply the MaTDR algorithm to it to coordinate the exact bounds with other agents, and subsequently, combine the result with its own local DTP solutions to enumerate its local solution space.

Boerkoel & Durfee (2013a) claim that the MaDTP-TD algorithm is sound and complete for solving the MaDTP decoupling problem.[6] Due to the combinatorics with respect to disjunct selection, the time complexity of the algorithm can be $\mathcal{O}(k^{|C|})$ in the worst case.

2.2 Automated negotiation

In this section, we give a brief overview of the field of automated negotiation, to be able to prudently decide which negotiation approaches can be applied to our domain of temporal problems. After a general overview, we focus on some basic strategies with respect to bidding, accepting and modeling the opponents, and on different more elaborate approaches that have been developed.

2.2.1 General overview

Jennings et al. (2001, p. 202) describe automated negotiation as "a distributed search through a space of potential agreements". In general, automated negotiation tactics are used when different agents need to assign values to a certain set of variables (e.g. the prices for a collection of goods), or need to divide a number of resources between them (where the agents have preferences for the different options), when exact optimal solutions cannot be found due to search space size or privacy requirements.

The variables in the search space that will be negotiated about are usually called *issues*. They have a domain of possible values that is commonly known to all agents. Each assignment of values from the domains to the issues can be seen as a potential agreement. Preferences for individual agents can be represented either as orderings of the potential agreements, or as utility values (e.g. in [0,1]) assigned to each potential agreement.

A search for an agreement consists of different proposals from the different agents, and reactions to them in the form of either an acceptation or a counterproposal. The negotiation ends when the agents agree on a certain proposal, or when a negotiation deadline is reached without an agreement.

Negotiation protocols can differ, dependent on (among others) the following properties:

- **Number of agents** A widely used protocol when the negotiation takes place between two agents is the alternating offers protocol (Wooldridge, 2009; Baarslag, 2016): one of the agents makes a proposal, the other either accepts this or makes a counterproposal, in which case the first agent has the choice between accepting or making a counteroffer, et cetera. On the other hand, when more than two agents are involved, other approaches can be used, e.g.

[6] Optimality with respect to some performance metric is however not guaranteed.

auction-based approaches (Fatima et al., 2015, Chapter 8) or approaches with a mediator (see below).

- **Number of issues** Negotiations on multiple issues can be much more complex than negotiations on single issues. In the case of a multiple-issue negotiation, the issues can be independent or interdependent. Multiple independent issues can be seen as separate single issues and treated with a sequential or simultaneous procedure, but also a package deal procedure that groups the issues and treats them as an entity is possible (Fatima, Wooldridge & Jennings, 2006, 2009; Kattan, Ong & Galván-López, 2013). Such a procedure is also appropriate for interdependent issues. An advantage of multi-issue negotiation is that it will possibly lead to a mutual gain due to trade-offs: an agent can be willing to make a concession on one of the issues if it can make a greater profit on another issue.

- **Nature of the utility functions** A distinction can be made between linear and non-linear utility functions (Ito, Hattori & Klein, 2007). For linear utility functions (e.g. for the price of a product for a buyer and a seller), optimal strategies are known (e.g. Fatima et al., 2006), but many real-world situations deal with non-linear utility functions. For example, when issues are dependent, their utility functions can not simply be summed up, but a more refined function is needed.

2.2.2 Strategy components in automated negotiation

Baarslag (2016) distinguishes three main parts in the study of automated negotiation strategies, which can be considered separately: proposing offers (bidding), deciding whether to accept offers, and constructing an opponent model.

- **Bidding** Most bidding strategies use an expected utility function over time that takes the maximal utility possible for the agent at the start of negotiation and decreases in time to expected utility 0 at the negotiation deadline. When the agent is expected to select a proposal at time t, it takes one with the expected utility corresponding to t. The idea is that if all agents concede during the negotiation, a consensus can be reached. Note that different expected utility functions are possible, e.g. a boulware function[7] where an agent only concedes at the end of the negotiation, or a conceder function, where the expected utility decreases early in the process (Wooldridge, 2009; Zhang, Song, Chen & Hong, 2011; Ren & Zhang, 2014; Baarslag, 2016, p. 27).

- **Accepting** A criterion used often for accepting a proposal is whether its utility is higher than the utility of the next own offer to propose. However, Baarslag (2016, Chapter 4, 5) shows that time and previous proposals can also be important in determining whether to accept.

- **Opponent model** Based on the sequence of bids of an agent, it is possible for other agents to make assumptions about its preferences and incorporate these in their bidding strategies. Baarslag (2016) describes approaches for bilateral negotiations, but e.g. Zheng, Chakraborty, Dai & Sycara (2016) use an approach in which an agent takes the average of the previous proposals of all other agents, and selects a new proposal that is nearest to it, respecting its expected utility function. De Weerd, Verbrugge & Verheij (2016) describe a negotiation approach for games with a fixed set of possible goals for each agent. Agents do not know the goals of their opponent agents, but sometimes an agent can infer from the proposals of an opponent agent that some goal in the set of possible goals cannot be the actual goal of the opponent agent. Based on the remaining possible goals, the agent then can try to make its new proposals more attractive. De Weerd et al. (2016) show that in particular using higher-order theory of mind results in better negotiation outcomes.

[7] The term *Boulwarism*, named after the negotiation stategies of L. R. Boulware, is commonly associated with a 'take-it-or-leave-it' approach, in which only one ultimate offer is proposed. However, the actual notions of Boulwarism are more complicated (Schmidt, 1970). In the negotiation literature, the term *boulware* is commonly used to describe a strategy in which conceding takes place only at the end of a negotiation.

In general, the proposals of the agents and their acceptance and rejection of the proposals of other agents are the only information they communicate to each other. Hence, the privacy of the agents is partially respected. Although the other agents can infer some information of the actions of an agent, it is not the case that all its preferences need to become public information.

2.2.3 Sophisticated negotiation approaches

Many variations on the basic approaches described above have been developed, from which we mention a few. Ito et al. (2007) explain that many approaches are based on multiple independent issues with linear utility functions, but that there is a need for approaches that handle interdependent issues and non-linear utility functions. They propose a protocol in which each agent locally samples its utility space and uses a simulated annealing approach to find the local optima and the surrounding regions. Subsequently, a bidding system is used to combine the local solutions and find a global one in their intersection.

Another approach is the use of genetic algorithms to explore the space of possible offers in situations with limited information (Lau, 2005; Kattan et al., 2013): a fitness function is defined based on the agent's own utilities and on the opponent's previous offers, and a population of feasible offers is created which is adapted in the subsequent negotiation rounds by mutations, crossovers and cloning.

Approaches with a coordinator are often used, both for bilateral negotiations and for negotiations with more than two agents. An example of the first is the Single Negotiation Text procedure (Ehtamo & Hämäläinen, 2001; Lin & Chou, 2004), in which a mediator proposes a solution, both agents can criticize that, and the mediator bases his new possible solution on the comments, until a mutual agreement is reached. Fujita, Ito & Klein (2014) use an approach with a coordinator where multiple agents can rate a certain proposal from the coordinator. The coordinator uses a simulated annealing approach to change the latest accepted proposals to overcome local optima. Moreover, they propose to group interdependent issues into independent groups to reduce the search complexity. Zhang & Liu (2016) use a mediator-based approach for privacy reasons. Sending bids to the mediator only and not directly to the opponent avoids agents to generate misrepresentations based on their opponent's bids.

Beside the heuristic approaches described in this section, Jennings et al. (2001) mention two other automated negotiation approaches. One is the argumentation-based approach, in which agents can e.g. explain explicitly their preferences, or try to change the opinions of other agents by giving arguments (Dimopoulos & Moraitis, 2014). We will not use this approach since its framework is much more complex than the framework for heuristic approaches, and less easily applied to our temporal problem research. The other approach is based on game theory, but for this approach, it is assumed that the influence of moves and the preferences of the opponents are known, which is not the case in our temporal problems context.

2.3 Formalized research goals

Now that we have seen the current frameworks and approaches both in temporal problem solving and in automated negotiation, we are able to formalize our research questions.

First, we observe that there exist frameworks both for temporal problems with preferences and for temporal problems with multiple agents involved in them, but that a combination of these two frameworks does not exist: To the best of our knowledge, neither for MaSTPs, nor for MaDTPs, an approach with preferences exists. In Chapter 1, however, we argued that such problems are highly relevant. Moreover, in Sections 3.1 and 4.1, we will give more detailed motivating examples that show the need for multi-agent temporal problems with preferences. Our first two research questions therefore are:

RQ1 **[MaSTPP definition]** How can a framework for the Multi-agent Simple Temporal Problem with Preferences be defined?

RQ2 [**MaDTPP definition**] How can a framework for the Multi-agent Disjunctive Temporal Problem with Preferences be defined?

When we will have developed the frameworks, we focus on solving them. As explained in Section 1.1 and elaborated in Section 2.2, we focus on applying automated negotiation approaches to improve current solutions for three reasons: In automated negotiation, cooperation and competition are combined, automated negotiation respects the privacy of the agents in general, and automated negotiation can be relatively computationally efficient.

For MaSTPPs, our goal is to change existing decouplings by automated negotiation in such a way that the new decoupling will be better for some agents and not worse for any agent. Also, we focus on the distributed approach of Boerkoel & Durfee (2013b), and observe that their MaTD approach (see Section 2.1.4) does not necessary result in optimal solutions. We wonder whether we can improve the algorithm itself by incorporating automated negotiation techniques into it. Hence, we formulate the following research questions:

RQ3 [**MaSTPP post-decoupling negotiation approach**] Given a decoupling solution to a Multi-agent Simple Temporal Problem with Preferences (i.e. a set of local STPP subproblems for the different agents that are all consistent and for which any combination of solutions yields a global solution to the MaSTPP, see Section 2.1.4), what automated negotiation protocol can be used as a post-processing step to make a Pareto improvement on the decoupling with respect to the expected preferences of the agents?[8]

RQ4 [**MaSTPP pre-decoupling negotiation approach**] Given the Multi-agent Temporal Decoupling algorithm with Relaxation of Boerkoel & Durfee (2013b) (see Section 2.1.4)[9] to solve a Multi-agent Simple Temporal Problem with Preferences, what automated negotiation approach can be incorporated into the algorithm to improve the resulting decoupling with respect to the expected preferences of the agents?

For the MaDTPP, we have comparable research goals. We want to improve current decouplings by automated negotiation, and we want to adjust the algorithm of Boerkoel & Durfee (2013a), that generates random solutions for the MaDTPP. Hence, we formalize our research questions as follows:

RQ5 [**MaDTPP post-decoupling negotiation approach**] Given a decoupling solution to a Multi-agent Disjunctive Temporal Problem with Preferences (in the form of a set of local STPP subproblems[10] for the different agents that are all consistent and for which any combination of solutions yields a global solution to the MaDTPP, see Section 2.1.4), what automated negotiation protocol can be used as a post-processing step to make a Pareto improvement on the decoupling with respect to the preferences of the agents?

RQ6 [**MaDTPP pre-decoupling negotiation approach**] Given the MaDTP Temporal Decoupling algorithm of Boerkoel & Durfee (2013a) (see Section 2.1.4) to solve a Multi-agent Disjunctive Temporal Problem, how can automated negotiation be incorporated into the algorithm to improve the resulting solutions with respect to the preferences of the agents?

In the above research questions, we focus on Pareto improvements, such that only changes will be made that are better for some agents and not worse for any agent. However, it is interesting to compare this approach with an approach in which also non-Pareto improvements are allowed: If agents allow another agent to have a great advantage at the cost of accepting only a limited disadvantage for themselves, social welfare can possibly increase further. Hence, we continue our research questions with the following one:

[8]In Chapter 3, it will become clear that for the MaSTPP, actual preferences can only be determined for a full solution and not for a decoupling without values assigned to all variables. Hence, we focus on improving the expected preferences instead of the actual preferences.

[9]We use the decoupling algorithm of Boerkoel & Durfee (2013b) since we need to build on this for solving the MaDTPP (see Research Question *RQ6* and Chapter 6).

[10]We will use STPP subproblems here instead of DTPP subproblems; this will be motivated in Section 6.1.

RQ7 [**Non-Pareto improvements**] What are the advantages for social welfare when non-Pareto improvements are allowed in the negotiation approaches, compared with negotiation approaches that induce only Pareto improvements?

We propose to focus on solving scheduling problems with arbitrary preference functions for the agents, in constrast to earlier approaches that focus on arbitrary solutions or on the flexibility of solutions. Now, an interesting point is whether our proposed preference methods can be compared with the approaches that focus on flexibility. Hence, we conclude our research questions with the following one:

RQ8 [**Flexibility**] In what way can preferences in scheduling problems be used to represent flexibility, and how do preference approaches relate to flexibility approaches?

Chapter 3

The Multi-Agent Simple Temporal Problem with Preferences

In the previous chapter (Section 2.1), we encountered a set of single-agent problems, consisting of the Simple Temporal Problem (STP), the Disjunctive Temporal Problem (DTP), the Simple Temporal Problem with Preferences (STPP), and the Disjunctive Temporal Problem with Preferences (DTPP), and some of their multi-agent counterparts: the Multi-agent Simple Temporal Problem (MaSTP) and the Multi-agent Disjunctive Temporal Problem (MaDTP). Natural successors in this sequence would be the MaSTP with Preferences and the MaDTP with Preferences, but strikingly, these are, to our knowledge, not defined in the literature. However, such problems, in which multiple agents with different preferences are involved in mutual (disjunctive) temporal constraints, are common in many fields.

We therefore introduce new frameworks for these problems. In this chapter, we give an example that illustrates the need for a Multi-agent Simple Temporal Problem with Preferences (MaSTPP), and subsequently, we give a formal definition of the MaSTPP. Then, we relate the framework to our research goals. The Multi-agent Disjunctive Temporal Problem with Preferences (MaDTPP) will be discussed in Chapter 4.

3.1 Example

Consider again the situation of Example 2.6 in which on a morning a student and a professor have to schedule a meeting of about 15 minutes. Before the meeting, the student has to study the topic himself for roughly 3 hours, and the professor has to give a lecture ending at 11:00. In Example 2.6 we did not append preferences to the problem, but it is natural to assume that both the student and the professor can have preferences on the start and end time of the meeting. Assume e.g. that the professor prefers to have the meeting either right after the lecture or at the end of the morning, and that the student prefers a longer meeting (20 minutes) to a shorter one (10 minutes) and prefers not to start before 9:00.

3.2 Framework definition

In earlier approaches (e.g. Peintner & Pollack, 2004; Moffitt, 2011), preferences were only defined for a single agent in problems where the agent manages all temporal variables itself. Now, we expand this in such a way that each agent can have its own preferences on the constraints it is aware of, i.e. agent a can have preferences on $C_L^a \cup C_X^a$. Recall that Peintner & Pollack (2004) defined preferences as follows: for a constraint $-b_{ji} \leq v_j - v_i \leq b_{ij}$ a preference function $f_{ij} : [-b_{ji}, b_{ij}] \to [0, \infty)$ maps each value for the temporal difference to a preference value. We can simply transfer this approach to the multi-agent case by assigning each agent a its own preference functions, defined only for the constraints $C_L^a \cup C_X^a$ that are known by the agent. Hence, we define the MaSTPP as follows:

Definition 3.1. A Multi-agent Simple Temporal Problem with Preferences $\mathcal{S}$ is a tuple $\langle V, C, G, g, F\rangle$ where

- $V = \{v_0, v_1, \cdots, v_{n-1}\}$ is a set of n temporal variables (with $z = v_0$ the reference time point);
- $C = \{c_1, c_2 \cdots, c_k\}$ is a set of temporal constraints, where each c_x is of the form $-b_{j_x i_x} \leq v_{j_x} - v_{i_x} \leq b_{i_x j_x}$ ($b_{j_x i_x}, b_{i_x j_x} \in \mathbb{R} \cup \{\infty\}$);
- $G = \{A, B, C, \cdots\}$ is a set of m different agents;
- $g : V \setminus \{z\} \to G$ is a function that maps each temporal variable except the reference time point to an agent, that owns the variable;
- F is the smallest set of functions such that $\forall c_x \in C\ \forall a \in G\ ((g(v_{j_x}) = a \vee g(v_{i_x}) = a) \to f_x^a \in F)$ and each $f_x^a : [-b_{j_x i_x}, b_{i_x j_x}] \to [0, \infty)$ is a function that maps the difference of two temporal variables in a temporal constraint to a preference value for agent a.

A *solution* σ to an MaSTPP $\mathcal{S}$ is an assignment of real values to the temporal variables V such that all constraints are satisfied, i.e. $\sigma : V \to \mathbb{R}$ is such that $\forall c_x \in C\ (-b_{j_x i_x} \leq \sigma(v_{j_x}) - \sigma(v_{i_x}) \leq b_{i_x j_x})$. Generally, $\sigma(z) = 0$. The set of all solutions to $\mathcal{S}$ is denoted by $\Sigma_{\mathcal{S}}$.

Definition 3.2. Given an MaSTPP $\mathcal{S}$ and a solution $\sigma \in \Sigma_{\mathcal{S}}$, the *preference value* for an agent a, p^a, is defined by the sum of all preference values for agent a that are established by σ, i.e., the function $p^a : \Sigma_{\mathcal{S}} \to [0, \infty)$ is given by

$$p^a(\sigma) = \sum_{f_x^a \in F} f_x^a(\sigma(v_{j_x}) - \sigma(v_{i_x})).$$

A solution σ to an MaSTPP $\mathcal{S}$ is *preferred* to a solution $\hat{\sigma}$ by agent a if $p^a(\sigma) > p^a(\hat{\sigma})$.

Definition 3.3. Given an MaSTPP $\mathcal{S}$ and a solution $\sigma \in \Sigma_{\mathcal{S}}$, the *social welfare* of the solution, $s(\sigma)$, is defined by the sum of the preference values of all agents. Hence, $s : \Sigma_{\mathcal{S}} \to [0, \infty)$ is given by

$$s(\sigma) = \sum_{a \in G} p^a(\sigma).$$

Example 3.1. *We represent our example from Section 3.1 in this framework. It mainly resembles the definition given in Example 2.6, but preferences are added. For the student, we model that the preference for the meeting is proportional to its duration by assigning a preference value of 0 to a duration of 10 minutes and, with a linear increase, a preference value of 10 to a duration of 20 minutes. Also, we add an increasing preference value for the first time point for the student (S_S^A) to be between 8:00 and 9:00, and a constant high preference value for S_S^A to be at 9:00 or later. For the professor, we base a preference value on the temporal difference between the end time of the lecture and the start time of the meeting. Both for a short distance (between 0 and 5 minutes) and for a long distance (more than 30 minutes) a high preference value is chosen. For the other cases, a preference value of 0 is assigned. Note that for clarity reasons, the functions that map all values to 0 are omitted in the description of F.*

- $V = \{z, S_S^A, S_E^A, M_S^A, M_E^A, M_S^B, M_E^B, L_S^B, L_E^B\}$
- $C =$

$$\begin{aligned}
\{\ c_1 :&\quad 175 \leq S_E^A - S_S^A \leq 185,\\
c_2 :&\quad 10 \leq M_E^A - M_S^A \leq 20,\\
c_3 :&\quad 0 \leq M_S^A - S_E^A \leq \infty,\\
c_4 :&\quad 120 \leq L_E^B - L_S^B \leq 120,\\
c_5 :&\quad 10 \leq M_E^B - M_S^B \leq 20,
\end{aligned}$$

$$
\begin{aligned}
c_6: &\quad 0 \leq M_S^B - L_E^B \leq \infty, \\
c_7: &\quad 0 \leq M_S^B - M_S^A \leq 0, \\
c_8: &\quad 0 \leq M_E^B - M_E^A \leq 0, \\
c_9: &\quad 0 \leq S_S^A - z \leq 240, \\
c_{10}: &\quad 0 \leq S_E^A - z \leq 240, \\
c_{11}: &\quad 0 \leq M_S^A - z \leq 240, \\
c_{12}: &\quad 0 \leq M_E^A - z \leq 240, \\
c_{13}: &\quad 30 \leq M_S^B - z \leq 240, \\
c_{14}: &\quad 30 \leq M_E^B - z \leq 240, \\
c_{15}: &\quad 60 \leq L_S^B - z \leq 60, \\
c_{16}: &\quad 180 \leq L_E^B - z \leq 180 \; \}
\end{aligned}
$$

- $G = \{A, B\}$
- $g^{-1}(A) = \{S_S^A, S_E^A, M_S^A, M_E^A\}$ *and* $g^{-1}(B) = \{M_S^B, M_E^B, L_S^B, L_E^B\}$
- $F = \{f_2^A, f_6^B, f_9^A, \cdots\}$, *where*
  - $f_2^A(x) = x - 10$
  - $f_6^B(x) = \begin{cases} 30 & \text{if } 0 \leq x \leq 5 \text{ or } 30 \leq x \\ 0 & \text{otherwise} \end{cases}$
  - $f_9^A(x) = \begin{cases} x & \text{if } 0 \leq x < 60 \\ 60 & \text{otherwise} \end{cases}$

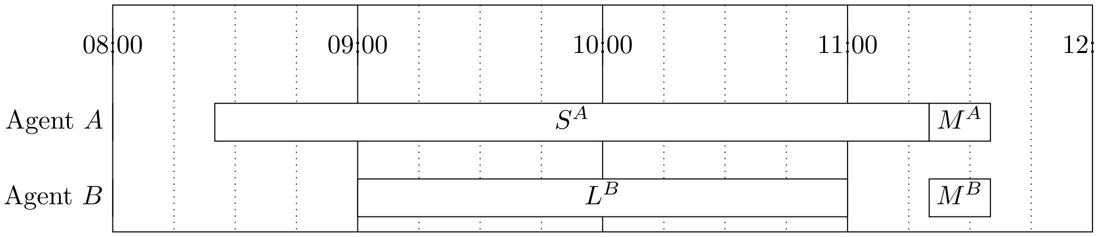
In Figure 3.1, different solutions to this problem are represented. The solution σ_1 *of Figure 3.1a is given by*

$$
\begin{aligned}
\sigma_1(S_S^A) &= 25 \\
\sigma_1(S_E^A) &= 200 \\
\sigma_1(M_S^A) &= 200 \\
\sigma_1(M_E^A) &= 215 \\
\sigma_1(M_S^B) &= 200 \\
\sigma_1(M_E^B) &= 215 \\
\sigma_1(L_S^B) &= 60 \\
\sigma_1(L_E^B) &= 180
\end{aligned}
$$

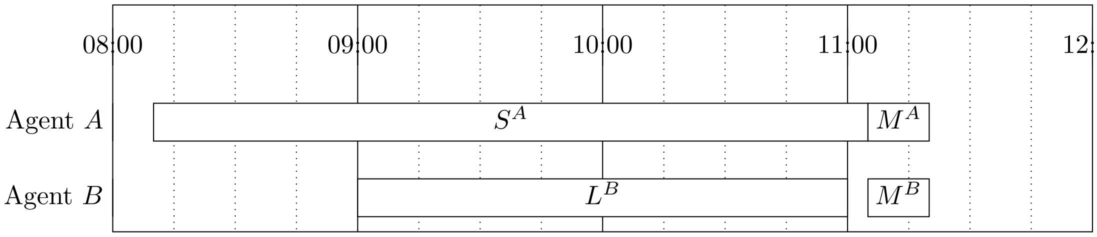
Preference values for the agents for this solution will be computed as follows:

$$
\begin{aligned}
p^A(\sigma_1) &= \sum_{f_x^A \in F} f_x^A(\sigma_1(v_{j_x}) - \sigma_1(v_{i_x})) \\
&= f_2^A(\sigma_1(M_E^A) - \sigma_1(M_S^A)) + f_9^A(\sigma_1(S_S^A) - \sigma_1(z)) \\
&= f_2^A(215 - 200) + f_9^A(25 - 0) \\
&= 5 + 25 \\
&= 30
\end{aligned}
$$

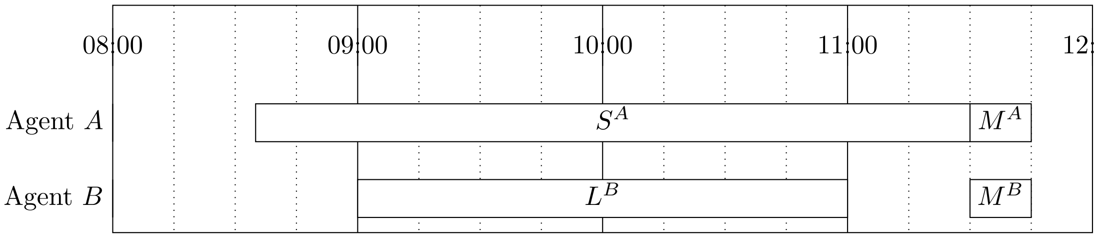
$$
p^B(\sigma_1) = \sum_{f_x^B \in F} f_x^B(\sigma_1(v_{j_x}) - \sigma_1(v_{i_x}))
$$



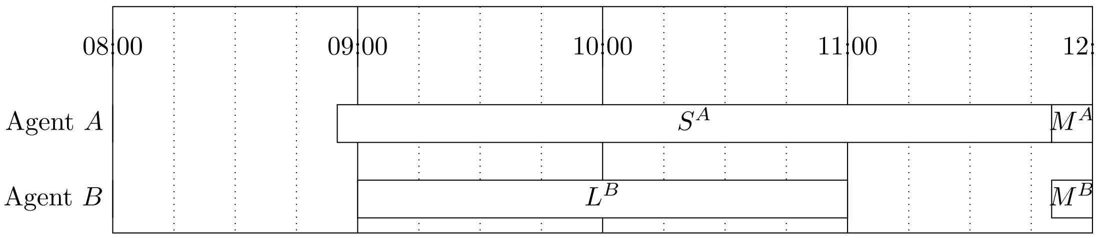
(a) Schedule σ_1: Initial MaSTPP schedule with meeting starting at 11:20. Preference values are $p^A(\sigma_1) = 30$ and $p^B(\sigma_1) = 0$.



(b) Schedule σ_2: Non-Pareto alteration (gain of 30 for agent B, loss of 15 for agent A) on initial MaSTPP schedule ($p^A(\sigma_2) = 15$ and $p^B(\sigma_2) = 30$).



(c) Schedule σ_3: Pareto improvement (gain of 10 for agent A, gain of 30 for agent B) on initial MaSTPP schedule ($p^A(\sigma_3) = 40$ and $p^B(\sigma_3) = 30$).



(d) Schedule σ_4: Pareto improvement (gain of 15 for agent A, no gain or loss for agent B) on MaSTPP schedule of Figure 3.1c ($p^A(\sigma_4) = 55$ and $p^B(\sigma_4) = 30$).

Figure 3.1: MaSTPP schedules with different preference values.

$$= f_6^B(\sigma_1(M_S^B) - \sigma_1(L_E^B))$$
$$= f_6^B(200 - 180)$$
$$= 0$$

The other solutions given in Figure 3.1 are captured by $\sigma_2(S_S^A) = 10$, $\sigma_2(M_S^A) = 185$, and $\sigma_2(M_E^A) = 200$, with $p^A(\sigma_2) = 5 + 10 = 15$ and $p^B(\sigma_2) = 30$ (Figure 3.1b); $\sigma_3(S_S^A) = 35$, $\sigma_3(M_S^A) = 210$, and $\sigma_3(M_E^A) = 225$, with $p^A(\sigma_3) = 5 + 35 = 40$ and $p^B(\sigma_3) = 30$ (Figure 3.1c); and $\sigma_4(S_S^A) = 55$, $\sigma_4(M_S^A) = 230$, and $\sigma_4(M_E^A) = 240$, with $p^A(\sigma_4) = 0 + 55 = 55$ and $p^B(\sigma_4) = 30$ (Figure 3.1d). Note that in the last case, the value of f_2^A has changed due to the shorter meeting time, and that in all schedules σ_2, σ_3, and σ_4, the value of f_6^B is 30 due to the appropriate temporal difference between the end of the lecture and the start of the meeting.

3.3 Analysis

We analyze the different solutions of Example 3.1 as depicted in Figure 3.1. Schedule σ_1 (Figure 3.1a) is an extension of a decoupling solution found by the MaTDR algorithm of Boerkoel & Durfee (2013b) (see Section 2.1.4). In this case, variable M_E^B was the last one in the shared elimination order, and hence, the first one to which a value was assigned. Since the possible domain for agent B for this variable was $[190, 240]$ (after the lecture ending at 11:00, at least 10 minutes were needed for the meeting), the midpoint heuristic assigned the value 215 to M_E^B. The next last element in the shared elimination order was M_S^B. Due to the update on M_E^B, the new domain of M_S^B was $[195, 205]$ (since the meeting needed to last 10 to 20 minutes), and hence the midpoint 200 was selected as value for M_S^B. M_S^A and M_E^A were also fixed by these assignments, such that the decoupling of the MaSTPP is given by adding constraints that fix the meeting time points to the local problems of the agents. Schedule σ_1 is the optimal schedule for the agents that respects the decoupling constraints: Agent A scheduled the study task immediately before the meeting and chose the shortest possible study time s.t. S_S^A took place as late as possible.

This initial solution has a preference value of 30 for agent A and a preference value of 0 for agent B. Hence, agent B will prefer σ_2 to σ_1, since $p^B(\sigma_2) = 30$. However, agent A will not accept such a change in the schedule since $p^A(\sigma_2) < p^A(\sigma_1)$. But schedule σ_3 is also preferred to schedule σ_1 by agent B, and agent A has also a higher preference value for this schedule in comparison with schedule σ_1. Hence, σ_3 is a Pareto improvement of σ_1 and will be accepted by all agents. Furthermore, σ_4 is even better for agent A, and also a Pareto improvement since it not worse than σ_3 for agent B. Both agents will therefore agree to schedule σ_4.

We have defined a framework in which preferences for the different agents can be added to an MaSTP, such that each agent is able to valuate the different solutions to an MaSTPP. We showed that based on an initial solution, Pareto improvements with respect to the preferences of the agents can be made. Furthermore, the above analysis and comparison of the different schedules fits naturally into a negotiation context. Our goal, described by research questions *RQ3* and *RQ4* in Section 2.3, is now to find automated negotiation methods for obtaining such solutions that are preferred by the agents, instead of random solutions or solutions that only focus on flexibility (Boerkoel, 2012; Boerkoel & Durfee, 2013b; Wilson et al., 2014; Wilson, 2016). We will focus on solving the MaSTPP in Chapter 5.

Chapter 4

The Multi-Agent Disjunctive Temporal Problem with Preferences

In Section 2.1, we encountered a set of single-agent problems consisting of the Simple Temporal Problem (STP), the Disjunctive Temporal Problem (DTP), the Simple Temporal Problem with Preferences (STPP), and the Disjunctive Temporal Problem with Preferences (DTPP). In the previous chapter, we extended the set of existing multi-agent counterparts (consisting of the Multi-agent Simple Temporal Problem (MaSTP) and the Multi-agent Disjunctive Temporal Problem (MaDTP)) with the Multi-agent Simple Temporal Problem with Preferences (MaSTPP). A natural successor in this sequence would be the Multi-agent Disjunctive Temporal Problem with Preferences (MaDTPP), which we introduce in this chapter.

First, we give a motivating example, subsequently, we discuss some advantages and disadvantages of possible approaches for adding preferences, and we give a formal definition of the MaDTPP. We conclude with an analysis of different solutions for our example and relate the goals for solving the problem to them.

4.1 Example

Consider our running example from Chapter 2 (Example 2.8): A student and a professor have to make an appointment of approximately 15 minutes for discussing a topic. Furthermore, the student has to study the topic himself for approximately 3 hours, and the professor has to give a lecture. However, we can imagine that in a real-world situation, this example is rather simplistic. For example, it is possible that the student can study much more efficiently when the discussion takes place before the study than when it is done afterwards. Also, one can imagine that both the student and the professor have preferences on the start and end times of their activities. We reformulate our example problem as follows:

A student (A) is studying a certain topic, and needs a short meeting with his professor (B) to ask some questions. If he first studies the material himself, the meeting can be very effective and they will be done in 15 minutes. However, the studying will require much time (approximately 3 hours). On the other hand, if they have a meeting first, the professor can give some more preliminary clarifications, and the study can be done much more efficiently (approximately 2 hours). However, in this case, the meeting will require about half an hour. Note that the professor has to give a lecture from 9:00 to 11:00 and cannot be present before 8:30. The student can be present from 8:00, but likes to start not before 9:00. Furthermore, both parties prefer to use as little total time as possible for all their tasks, and the professor prefers to have either no idle time or a large amount of idle time (more than half an hour) between the activities if the meeting is scheduled after the lecture, and strongly prefers a short meeting time and any idle time between meeting and lecture

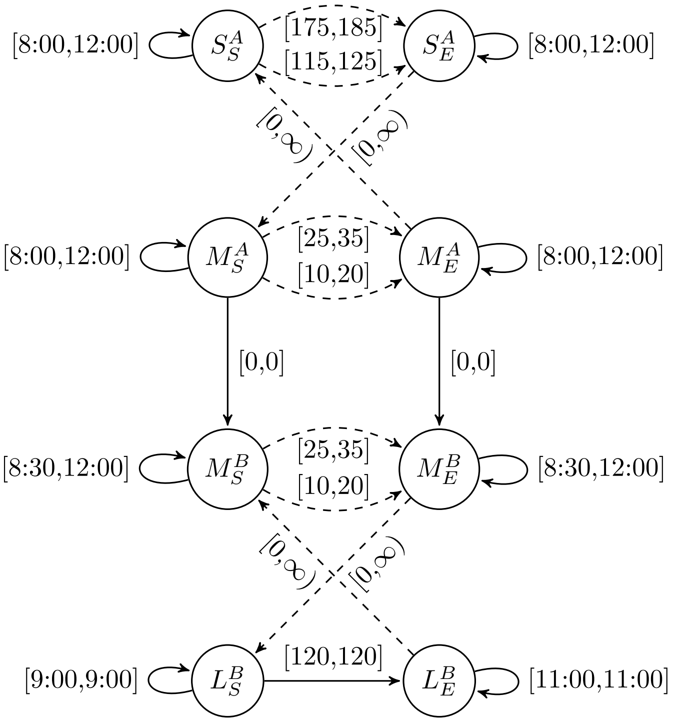


Figure 4.1: MaDTP for two agents (A and B) with two tasks each. Length of the tasks need to be modeled dependent on the order of the tasks.

$$
\begin{aligned}
c_1: &\quad (d_1^1)\ 175 \leq S_E^A - S_S^A \leq 185 \quad \vee \quad (d_1^2)\ 115 \leq S_E^A - S_S^A \leq 125 \\
c_2: &\quad (d_2^1)\ 25 \leq M_E^A - M_S^A \leq 35 \quad \vee \quad (d_2^2)\ 10 \leq M_E^A - M_S^A \leq 20 \\
c_3: &\quad (d_3^1)\ 25 \leq M_E^B - M_S^B \leq 35 \quad \vee \quad (d_3^2)\ 10 \leq M_E^B - M_S^B \leq 20 \\
c_4: &\quad (d_4^1)\ 120 \leq L_E^B - L_S^B \leq 120 \\
c_5: &\quad (d_5^1)\ 0 \leq S_S^A - M_E^A \leq \infty \quad \vee \quad (d_5^2)\ 0 \leq M_S^A - S_E^A \leq \infty \\
c_6: &\quad (d_6^1)\ 0 \leq M_S^B - M_S^A \leq 0 \\
c_7: &\quad (d_7^1)\ 0 \leq M_E^B - M_E^A \leq 0 \\
c_8: &\quad (d_8^1)\ 0 \leq M_S^B - L_E^B \leq \infty \quad \vee \quad (d_8^2)\ 0 \leq L_S^B - M_E^B \leq \infty \\
c_9: &\quad (d_9^1)\ 0 \leq S_S^A - z \leq 240 \\
c_{10}: &\quad (d_{10}^1)\ 0 \leq S_E^A - z \leq 240 \\
c_{11}: &\quad (d_{11}^1)\ 0 \leq M_S^A - z \leq 240 \\
c_{12}: &\quad (d_{12}^1)\ 0 \leq M_E^A - z \leq 240 \\
c_{14}: &\quad (d_{13}^1)\ 30 \leq M_S^B - z \leq 240 \\
c_{13}: &\quad (d_{14}^1)\ 30 \leq M_E^B - z \leq 240 \\
c_{15}: &\quad (d_{15}^1)\ 60 \leq L_S^B - z \leq 60 \\
c_{16}: &\quad (d_{16}^1)\ 180 \leq L_E^B - z \leq 180
\end{aligned}
$$

Figure 4.2: Constraints for the MaDTP of Figure 4.1.

if the meeting is scheduled before the lecture, such that she does not have to hurry for letting start the lecture on time. All activities must be finished before 12:00.

The graph representation of the example is given by the MaDTP in Figure 4.1. The constraints of the example without preferences are given in Figure 4.2.

4.2 Adding preferences

We want to extend the usual MaDTP framework by adding preferences for the different agents. Recall from Section 2.1.3 that different approaches have been used to represent preferences in single-agent problems, and consider Section 3.2 for our definition of MaSTPs with Preferences. We need to make a deliberate decision which aspects of the different approaches we can use in our multi-agent disjunctive case. Both the questions where exactly the preferences will be put (a preference value for each temporal difference between two variables in a constraint, or a preference value for a constraint as a whole) and how to handle dependencies between different constraints will be considered.

4.2.1 Preferences at time point level and at disjunct level

Recall from Section 2.1.3 the DTPP approach of Peintner & Pollack (2004), in which a preference function $f_{ij} : [-b_{ji}, b_{ij}] \to [0, \infty)$ can be added to each disjunct $-b_{ji} \leq v_j - v_i \leq b_{ij}$, such that the difference between two values for the variables in a disjunct will be mapped to a preference value. Given a solution σ, the preference value of a disjunctive constraint c is defined to be the maximum of the preference values assigned to its disjuncts: $\text{val}(c, \sigma) = \max_{d_{ij} \in c} f_{ij}(\sigma(v_j) - \sigma(v_i))$. The preference value of the solution can be defined as either the minimum of the preference values of the constraints, or the sum of the preference values of the constraints: $\text{val}(\sigma) = \min_{c \in C} \text{val}(c, \sigma)$ or $\text{val}(\sigma) = \sum_{c \in C} \text{val}(c, \sigma)$. In most cases, the sum of the preference values of the constraints is an appropriate measure (Moffitt, 2011). This approach was also used in our MaSTPP definition (Section 3.2).

Definition 4.1. We refer to preferences that are assigned to the difference between two values for the temporal variables in a disjunct, as described above, with the expression *preferences at time point level.*

Beside the DTPP approach, we have considered the VDTP approach of Moffitt (2011), in which hard and soft constraints are distinguished, and cost values are added to each constraint as a whole. A function $f : C \to [0, \infty)$ can be used that maps each constraint to a cost value, where cost values of ∞ denote hard constraints. The cost value of a solution σ is defined to be the sum of the costs of all the constraints that are violated by the solution: $\text{cost}(\sigma) = \sum_{c \in C | \text{viol}(\sigma, c)} f(c)$.

Definition 4.2. We refer to preferences that are assigned to a (disjunctive) constraint as a whole, as described above, with the expression *preferences at constraint level.*

Both approaches have advantages and disadvantages. The DTPP approach, with preferences at time point level, has the advantage that functions can be used to elegantly summarize multiple preference values, and that preferences can be represented very precisely on each disjunctive constraint. The disadvantage is, however, that only for complete solutions the preference value can be defined. No preference indication can be given for component STPs. This might be a real problem since all (Ma)DTP solving approaches (and also our MaDTPP approach) start with extracting component STPs; it would be preferable to be able to have a preference indication for these component STPs already, before exploring them in depth. The VDTP approach, with preferences at constraint level, does not have this problem: In this approach, the costs can already be determined for component STPs. The problem with this approach, however, is that preferences can be modeled less precisely, or at the cost of many extra disjuncts. For example, when we want to model that the student prefers to start not before 9:00, it is possible to define soft constraints $c_1, c_2, \cdots, c_7$, where c_1 encodes that the start time of the first activity of the student is between 8:00 and 8:10, where c_2 denotes that it is between 8:10 en and 8:20, and so on, and where c_7 denotes that it is after 9:00. We can give c_1 a minimal cost value and c_7 a maximal one, such that violating c_7 is much more expensive than violating c_1. The preferences are representable in this manner, but at the cost of seven extra disjuncts. Furthermore, the more precise we want to make it, the more disjuncts we need. On the other hand, when we use preferences at time point level, we can model the preferred student start time by a constraint with only one disjunct that encodes that it should be between 8:00 and 12:00, added with a preference function that is linear for values from 8:00 to 9:00 starting with minimum preference value and ending with maximum preference value, and having constant maximum preference value for values from 9:00 to 12:00. In this representation, the DTPP is less complex.

In our MaDTPP problem representation, we propose to combine both preferences at constraint level and preferences at time point level, to utilize the advantages of both approaches. However, we use a slight adaptation of the preferences at constraint level: instead of a cost value for a disjunctive constraint c as a whole, we use preference values for all disjuncts $d \in c$. Hence, for each disjunctive constraint c, their exists a function $f_c : c \to [0, \infty)$ that maps each disjunct in c to a preference value. Given a solution σ, the preference value for a disjunctive constraint is then defined to be the maximum of the preferences of all the disjuncts that are satisfied by the solution: $\text{val}(c, \sigma) = \max_{d \in c | \text{sat}(\sigma, d)} f_c(d)$. The preference value of the solution will then be defined by $\text{val}(\sigma) = \sum_{c \in C} \text{val}(c, \sigma)$. Note that with this approach, preference values still can be given for consistent component STPs.

Definition 4.3. We refer to preferences that are assigned to a disjunct as a whole, as described above, with the expression *preferences at disjunct level.*

We will propose a framework with preferences at disjunct level, such that preferences can be determined for consistent component STPs. Furthermore, to specify less important preferences, the disjuncts also contain preferences at time point level. Note however that these are subordinate to the preferences at disjunct level, i.e. a solution with a higher-valued disjunct is always preferred to a solution with a lower-valued disjunct, whatever the preference functions at time point level are. This for example enables us to specify a constraint with two disjuncts, one higher-valued in which the student starts after 9:00 without any preference function at time point level, and one lower-valued in which the student starts before 9:00, with a linear preference function that represents a low utility for starting at 8:00 and a higher utility when the starting time approaches 9:00.

4.2.2 Using Disjunctive Normal Form to handle dependencies between constraints

Micalizio & Torta (2015) describe an approach in which preferences on constraints can be modeled to be dependent on the values of other constraints. In our motivating example, we also have such dependencies. Consider the preferences of the professor: Dependent on the order of the lecture and the meeting, some preferences with respect to temporal distance between the activities are determined. Also for the student, there are dependencies: when he has the meeting before the study time, he prefers the longer meeting time and the shorter study time, and otherwise he prefers the shorter meeting time and the longer study time. Note, however, that these preferences are actually hard constraints in our example. Given the current frameworks (the VDTP approach of Moffitt (2011) and the dependency approach of Micalizio & Torta (2015)), we can model these constraints as dependent costs as in Figure 4.3. Violating constraint c_1^1 is no problem for agent A (cost 0) if c_5^1 is true (i.e. not having the long study time is acceptable if the study takes place after the meeting), but is a hard problem (cost ∞) if c_5^2 is instead true (i.e. the study takes place before the meeting). On the other hand, violating constraint c_1^2 (not having the short study time) is unacceptable (cost ∞) if c_5^1 is true, but is acceptable at a cost of 60 if c_5^2 is true. (The cost of 60 is proportional to the extra study time of c_1^1 and is chosen here to represent the preference of using as less total time as possible.)

Modeling these constraints as normal constraints, not as hard preferences, is impossible in this framework. However, the representation of constraints as hard preferences is somewhat counter-

$$
\begin{array}{lll}
c_1: & 175 \le S_E^A - S_S^A \le 185 \quad \vee \quad 115 \le S_E^A - S_S^A \le 125 \\
c_1^1: & 175 \le S_E^A - S_S^A \le 185 \\
c_1^2: & 115 \le S_E^A - S_S^A \le 125 \\
c_2: & 25 \le M_E^A - M_S^A \le 35 \quad \vee \quad 10 \le M_E^A - M_S^A \le 20 \\
c_2^1: & 25 \le M_E^A - M_S^A \le 35 \\
c_2^2: & 10 \le M_E^A - M_S^A \le 20 \\
\vdots & \\
c_5: & 0 \le S_S^A - M_E^A \le \infty \quad \vee \quad 0 \le M_S^A - S_E^A \le \infty \\
c_5^1: & 0 \le S_S^A - M_E^A \le \infty \\
c_5^2: & 0 \le M_S^A - S_E^A \le \infty \\
\vdots &
\end{array}
$$

(a)

$$
\begin{aligned}
f^A(c_1) &= \infty \\
f^A(c_1^1) &= \begin{cases} 0 & \text{if } c_5^1 \\ \infty & \text{if } c_5^2 \end{cases} \\
f^A(c_1^2) &= \begin{cases} \infty & \text{if } c_5^1 \\ 60 & \text{if } c_5^2 \end{cases} \\
f^A(c_2) &= \infty \\
f^A(c_2^1) &= \begin{cases} \infty & \text{if } c_5^1 \\ 0 & \text{if } c_5^2 \end{cases} \\
f^A(c_2^2) &= \begin{cases} 15 & \text{if } c_5^1 \\ \infty & \text{if } c_5^2 \end{cases} \\
&\vdots \\
f^A(c_5) &= \infty \\
f^A(c_5^1) &= 0 \\
f^A(c_5^2) &= 0 \\
&\vdots
\end{aligned}
$$

(b)

Figure 4.3: Part of the VDTP expansion of the constraints in Figure 4.2 (a), and the corresponding dependent cost function for agent A (b).

intuitive. Furthermore, the approach of Micalizio & Torta (2015) for solving DTPs with dependent preferences does not fit well into our approach for solving MaDTPPs (see Chapter 6). Therefore, we propose a quite simple extension of the MaDTP framework to deal with these problems: We will allow disjuncts in the disjunctive constraints not only to be simple constraints, but also to be conjunctions of simple constraints. Hence, a disjunctive constraint in our new framework is an expression in Disjunctive Normal Form. This allows us to group c_1^1, c_2^2, and c_5^2 into one disjunct, to group c_1^2, c_2^1, and c_5^1 into another disjunct, and to combine both disjuncts to form together a disjunctive constraint. This disjunctive constraint represents that one of the two orders with corresponding activity durations must be chosen, but does not require the many constraints of Figure 4.3a and the dependent preferences of Figure 4.3b.

4.3 Framework definition

Taking into account the considerations in the previous section, we define the Multi-agent Disjunctive Temporal Problem with Preferences as follows:

An MaDTPP consists of a set V of temporal variables and a set G of agents as defined for the MaSTP, MaDTP, and MaSTPP, such that each temporal variable (except the reference time point) is owned by exactly one agent. The set of constraints C, however, is defined slightly different. Each constraint $c \in C$ is a set of disjuncts, where each disjunct d is a set of conjuncts, where each conjunct e represents the bounds on the temporal difference of two variables (i.e. a conjunct equals an STP constraint $-b_{ji} \le v_j - v_i \le b_{ij}$). Furthermore, the MaDTPP has preferences both at disjunct level and at time point level: Each agent a has a preference value for each disjunct that occurs in a constraint for which 1) the number of disjuncts is greater than 1, and 2) at least one of the disjuncts is defined over a variable of agent a, i.e. the disjunct contains at least one conjunct for which at least one of the two involved temporal variables is owned by agent a. Furthermore, for each conjunct in the problem, the (one or two) agents that own the two involved temporal variables have a preference value for each possible temporal difference between the bounds $-b_{ji}$ and b_{ij}.

Hence, an MaDTPP extends an MaDTP with the possibility of multiple constraints per disjunct, and with preference functions added both at disjunct level and at time point level: Each agent has a preference value for each disjunct of a constraint that is defined over at least one of its variables, and each agent has a preference value for each actual difference in a temporal constraint that is defined over at least one of its variables.

The formal definition of the MaDTPP is given below.

Definition 4.4. A Multi-agent Disjunctive Temporal Problem with Preferences $\mathcal{D}$ is a tuple $\langle V, C, G, g, D, T \rangle$ where

- $V = \{v_0, v_1, \cdots, v_{n-1}\}$ is a set of n temporal variables (with $z = v_0$ the reference time point);
- $C = \{c_1, c_2 \cdots, c_h\}$ is a set of DNF constraints, where
  - each c_w is a set $\{d_{w1}, d_{w2}, \cdots, d_{wk}\}$ $(k \geq 1)$ of disjunctive constraints (c_w can be read as $d_{w1} \vee d_{w2} \vee \cdots \vee d_{wk}$), where
  - each d_{wx} is a set $\{e_{wx1}, e_{wx2}, \cdots, e_{wxl}\}$ $(l \geq 1)$ of conjunctive constraints (d_{wx} can be read as $e_{wx1} \wedge e_{wx2} \wedge \cdots \wedge e_{wxl}$), where
  - each e_{wxy} is a temporal constraint of the form $-b_{j_{wxy}i_{wxy}} \leq v_{j_{wxy}} - v_{i_{wxy}} \leq b_{i_{wxy}j_{wxy}}$ $(b_{j_{wxy}i_{wxy}}, b_{i_{wxy}j_{wxy}} \in \mathbb{R} \cup \{\infty\})$;
- $G = \{A, B, C, \cdots\}$ is a set of m different agents;
- $g : V \setminus \{z\} \to G$ is a function that maps each temporal variable except the reference time point to an agent, that owns the variable;
- D is the smallest set of functions such that $\forall c_w \in C\ \forall a \in G\ ((|c_w| > 1 \wedge \exists d_{wx} \in c_w\ \exists e_{wxy} \in d_{wx}\ (g(v_{j_{wxy}}) = a \vee g(v_{i_{wxy}}) = a)) \to f_w^a \in D)$ and each $f_w^a : c_w \to [0, \infty)$ is a function that maps each disjunct $d_{wx} \in c_w$ to a preference value for agent a;
- T is the smallest set of functions such that $\forall c_w \in C\ \forall d_{wx} \in c_w\ \forall e_{wxy} \in d_{wx}\ \forall a \in G\ ((g(v_{j_{wxy}}) = a \vee g(v_{i_{wxy}}) = a) \to f_{wxy}^a \in T)$ and each $f_{wxy}^a : [-b_{j_{wxy}i_{wxy}}, b_{i_{wxy}j_{wxy}}] \to [0, \infty)$ is a function that maps the difference of two temporal variables in a temporal constraint to a preference value for agent a.

A *solution* σ to $\mathcal{D}$ is an assignment of real values to the temporal variables V such that all constraints are satisfied, i.e. $\sigma : V \to \mathbb{R}$ is such that $\forall c_w \in C\ \exists d_{wx} \in c_x\ \forall e_{wxy} \in d_{wx}\ (-b_{j_{wxy}i_{wxy}} \leq \sigma(v_{j_{wxy}}) - \sigma(v_{i_{wxy}}) \leq b_{i_{wxy}j_{wxy}})$. Generally, $\sigma(z) = 0$. The set of all solutions to $\mathcal{D}$ is denoted by $\Sigma_{\mathcal{D}}$.

Definition 4.5. Given an MaDTPP $\mathcal{D}$ and a solution $\sigma \in \Sigma_{\mathcal{D}}$, the *preference value at disjunct level* for an agent a, p_d^a, is obtained by taking for each disjunctive constraint that the agent is aware of the maximum preference value of all satisfied disjuncts, and summing up these values. I.e., the function $p_d^a : \Sigma_{\mathcal{D}} \to [0, \infty)$ is defined by

$$p_d^a(\sigma) = \sum_{f_w^a \in D} \max_{d_{wx} \in c_w | \mathrm{sat}_w(\sigma, d_{wx})} f_w^a(d_{wx})$$

where $\mathrm{sat}_w : \Sigma_{\mathcal{D}} \times c_w \to \{\mathrm{true}, \mathrm{false}\}$ is a function such that $\mathrm{sat}_w(\sigma, d_{wx}) = \mathrm{true}$ iff $\forall e_{wxy} \in d_{wx}\ (-b_{j_{wxy}i_{wxy}} \leq \sigma(v_{j_{wxy}}) - \sigma(v_{i_{wxy}}) \leq b_{i_{wxy}j_{wxy}})$.

Definition 4.6. Given an MaDTPP $\mathcal{D}$ and a solution $\sigma \in \Sigma_{\mathcal{D}}$, the *preference value at time point level* for an agent a, p_t^a, is defined by the sum of all preference values assigned to time point differences for agent a corresponding to the constraints in the disjuncts that are made true by σ. I.e., the function $p_t^a : \Sigma_{\mathcal{D}} \to [0, \infty)$ is given by

$$p_t^a(\sigma) = \sum_{f_{wxy}^a \in T | \mathrm{sat}_w(\sigma, d_{wx})} f_{wxy}^a(\sigma(v_{j_{wxy}}) - \sigma(v_{i_{wxy}})).$$

A solution σ to an MaDTPP $\mathcal{D}$ is *preferred* to a solution $\hat{\sigma}$ by agent a if

- $p_d^a(\sigma) > p_d^a(\hat{\sigma})$, or
- $\forall c_w \in C\ \forall d_{wx} \in c_w\ (\text{sat}_w(\sigma, d_{wx}) = \text{sat}_w(\hat{\sigma}, d_{wx}))$ and $p_t^a(\sigma) > p_t^a(\hat{\sigma})$.

Note that preferences at time point level are subordinate to preferences at disjunct level: a solution with a higher preference value at disjunct level is always preferred to a solution with a lower preference value at disjunct level, regardless of the preference values at time point level. It would be more appropriate to define σ being preferred to $\hat{\sigma}$ if

- $p_d^a(\sigma) > p_d^a(\hat{\sigma})$, or
- $p_d^a(\sigma) = p_d^a(\hat{\sigma})$ and $p_t^a(\sigma) > p_t^a(\hat{\sigma})$,

but this will be impractical due to the solving approach: For solving a DTP, an appropriate component STP will be selected first; it will be expensive to check for equally-valued component STPs whether in one of them a higher preference value at time point level can be obtained. Therefore, we consider different equally-valued component STPs to be incomparable with respect to preferences at time point level.

Definition 4.7. Given an MaDTPP $\mathcal{D}$ and a solution $\sigma \in \Sigma_{\mathcal{D}}$, the *social welfare at disjunct level* and the *social welfare at time point level* of the solution, $s_d(\sigma)$ and $s_t(\sigma)$, are defined by the sum of the corresponding preference values of all agents. Hence, $s_d : \Sigma_{\mathcal{D}} \to [0, \infty)$ and $s_t : \Sigma_{\mathcal{D}} \to [0, \infty)$ are given by

$$s_d(\sigma) = \sum_{a \in G} p_d^a(\sigma) \text{ and } s_t(\sigma) = \sum_{a \in G} p_t^a(\sigma).$$

Note that we have the same problems as with the VDTP case (see Section 4.2.1) if a preference function with many different values for a temporal difference constraint is not subordinate to some interfering disjuncts. In that case, we still have to split the constraint into different disjuncts with each a preference value for an appropriate problem representation. However, it is in general possible to do this with some broad intervals to limit the number of disjuncts, and specify a preference function at time point level for each disjunct for fine-tuning the preferences.

Example 4.1. *Our example from Section 4.1 can be modeled as follows in our new framework. We introduce a node I^A that represents the start of the first activity of the student. Preferences at disjunct level are added to represent that the student likes to start not before 9:00, and preferences at time point level are added for the disjunct representing that I^A takes place before 9:00 (see the proposal at the end of Section 4.2.1). Furthermore, since both parties prefer to use as little total time as possible for all their tasks, we add preferences at disjunct level proportional to the estimated lengths of the tasks of the agents (see the disjunct preferences for constraints c_1 and c_2) and preferences at time point level (again proportional to task lengths) for another fine tuning when disjuncts are established (see the time point preferences for constraints c_1 and c_2). Since the professor prefers to have either no idle time or more than half an hour of idle time between the lecture and the meeting if the meeting is scheduled after the lecture, and strongly prefers as much idle time as possible between the meeting and the lecture if the meeting is scheduled before the lecture, we add no preferences at disjunct level for constraint c_6, but we add preferences at time point level for both disjuncts of this constraint.*

Note that for clarity reasons, there are commas inserted between the indices of the disjuncts in the description of C. Also, names of the conjuncts are not displayed, but these can easily be derived, e.g. $e_{1,1,2}$ will refer to $0 \leq M_S^A - S_E^A \leq \infty$. Furthermore, note that the zero values for f_6^B represent that agent B is indifferent about the disjuncts, but that for T, the functions that map all values to 0 are omitted for clarity reasons.

- $V = \{v_0, S_S^A, S_E^A, M_S^A, M_E^A, M_S^B, M_E^B, L_S^B, L_E^B, I^A\}$

- $C =$

$$
\begin{aligned}
\{\ c_1 :\quad & d_{1,1} :\ (175 \leq S_E^A - S_S^A \leq 185\ \wedge\ 0 \leq M_S^A - S_E^A \leq \infty\ \wedge\ 10 \leq M_E^A - M_S^A \leq 20)\\
& \vee\, d_{1,2} :\ (115 \leq S_E^A - S_S^A \leq 125\ \wedge\ 0 \leq S_S^A - M_E^A \leq \infty\ \wedge\ 25 \leq M_E^A - M_S^A \leq 35),\\
c_2 :\quad & d_{2,1} :\ 25 \leq M_E^B - M_S^B \leq 35 \quad \vee \quad d_{2,2} :\ 10 \leq M_E^B - M_S^B \leq 20,\\
c_3 :\quad & d_{3,1} :\ 120 \leq L_E^B - L_S^B \leq 120,\\
c_4 :\quad & d_{4,1} :\ 0 \leq M_S^B - M_S^A \leq 0,\\
c_5 :\quad & d_{5,1} :\ 0 \leq M_E^B - M_E^A \leq 0,\\
c_6 :\quad & d_{6,1} :\ 0 \leq M_S^B - L_E^B \leq \infty \quad \vee \quad d_{6,2} :\ 0 \leq L_S^B - M_E^B \leq \infty,\\
c_7 :\quad & d_{7,1} :\ 0 \leq S_S^A - z \leq 240,\\
c_8 :\quad & d_{8,1} :\ 0 \leq S_E^A - z \leq 240,\\
c_9 :\quad & d_{9,1} :\ 0 \leq M_S^A - z \leq 240,\\
c_{10} :\quad & d_{10,1} :\ 0 \leq M_E^A - z \leq 240,\\
c_{11} :\quad & d_{11,1} :\ 30 \leq M_S^B - z \leq 240,\\
c_{12} :\quad & d_{12,1} :\ 30 \leq M_E^B - z \leq 240,\\
c_{13} :\quad & d_{13,1} :\ 60 \leq L_S^B - z \leq 60,\\
c_{14} :\quad & d_{14,1} :\ 180 \leq L_E^B - z \leq 180,\\
c_{15} :\quad & d_{15,1} :\ 0 \leq S_S^A - I^A \leq \infty,\\
c_{16} :\quad & d_{16,1} :\ 0 \leq M_S^A - I^A \leq \infty,\\
c_{17} :\quad & d_{17,1} :\ 0 \leq I^A - z < 60 \quad \vee \quad d_{17,2} :\ 60 \leq I^A - z \leq 240\ \}
\end{aligned}
$$

- $G = \{A, B\}$
- $g^{-1}(A) = \{S_S^A, S_E^A, M_S^A, M_E^A, I^A\}$ *and* $g^{-1}(B) = \{M_S^B, M_E^B, L_S^B, L_E^B\}$
- $D = \{f_1^A, f_2^B, f_6^B, f_{17}^A\}$, *where*
  - $f_1^A(d_{1,1}) = 0$
  - $f_1^A(d_{1,2}) = 45$
  - $f_2^B(d_{2,1}) = 0$
  - $f_2^B(d_{2,2}) = 15$
  - $f_6^B(d_{6,1}) = 0$
  - $f_6^B(d_{6,2}) = 0$
  - $f_{17}^A(d_{17,1}) = 0$
  - $f_{17}^A(d_{17,2}) = 60$
- $T = \{f_{1,1,1}^A, \cdots, f_{1,1,3}^A, f_{1,2,1}^A, \cdots, f_{1,2,3}^A, f_{2,1,1}^B, f_{2,2,1}^B, \cdots, f_{6,1,1}^B, f_{6,2,1}^B, \cdots, f_{17,1,1}^A, \cdots\}$, *where*
  - $f_{1,1,1}^A(x) = 185 - x$
  - $f_{1,1,3}^A(x) = 20 - x$
  - $f_{1,2,1}^A(x) = 125 - x$
  - $f_{1,2,3}^A(x) = 35 - x$
  - $f_{2,1,1}^B(x) = 35 - x$
  - $f_{2,2,1}^B(x) = 20 - x$
  - $f_{6,1,1}^B(x) = \begin{cases} 30 & \textit{if } 0 \leq x \leq 5 \textit{ or } 30 \leq x \\ 0 & \textit{otherwise} \end{cases}$

- □ $f^B_{6,2,1}(x) = 3x$
- □ $f^A_{17,1,1}(x) = x$

Different solutions for this problem are given in Figure 4.4. Preference values for the agents for solution σ_1 (Figure 4.4a) will be computed as follows:

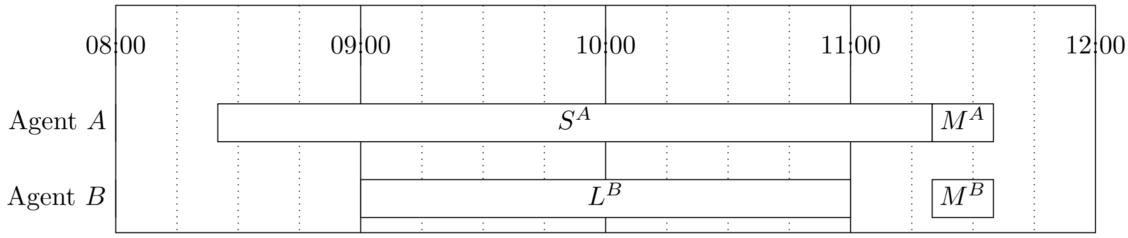
$$\begin{aligned}
p^A_d(\sigma_1) &= \sum_{f^A_w \in D} \max_{d_{wx} \in c_w | \text{sat}_w(\sigma_1, d_{wx})} f^A_w(d_{wx}) \\
&= \max_{d_{1,x} \in c_1 | \text{sat}_1(\sigma_1, d_{1,x})} f^A_1(d_{1,x}) + \max_{d_{17,x} \in c_{17} | \text{sat}_{17}(\sigma_1, d_{17,x})} f^A_{17}(d_{17,x}) \\
&= f^A_1(d_{1,1}) + f^A_{17}(d_{17,1}) \\
&= 0 + 0 \\
&= 0
\end{aligned}$$

$$\begin{aligned}
p^B_d(\sigma_1) &= \sum_{f^B_w \in D} \max_{d_{wx} \in c_w | \text{sat}_w(\sigma_1, d_{wx})} f^B_w(d_{wx}) \\
&= \max_{d_{2,x} \in c_2 | \text{sat}_2(\sigma_1, d_{2,x})} f^B_2(d_{2,x}) + \max_{d_{6,x} \in c_6 | \text{sat}_6(\sigma_1, d_{6,x})} f^B_6(d_{6,x}) \\
&= f^B_2(d_{2,2}) + f^B_6(d_{6,1}) \\
&= 15 + 0 \\
&= 15
\end{aligned}$$

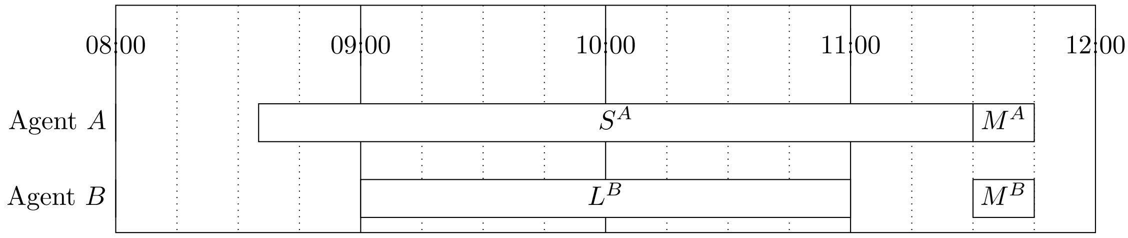
$$\begin{aligned}
p^A_t(\sigma_1) &= \sum_{f^A_{wxy} \in T | \text{sat}_w(\sigma_1, d_{wx})} f^A_{wxy}(\sigma_1(v_{j_{wxy}}) - \sigma_1(v_{i_{wxy}})) \\
&= f^A_{1,1,1}(\sigma_1(v_{j_{1,1,1}}) - \sigma_1(v_{i_{1,1,1}})) + f^A_{1,1,3}(\sigma_1(v_{j_{1,1,3}}) - \sigma_1(v_{i_{1,1,3}})) \\
&\qquad + f^A_{17,1,1}(\sigma_1(v_{j_{17,1,1}}) - \sigma_1(v_{i_{17,1,1}})) \\
&= f^A_{1,1,1}(\sigma_1(S^A_E) - \sigma_1(S^A_S)) + f^A_{1,1,3}(\sigma_1(M^A_E) - \sigma_1(M^A_S)) + f^A_{17,1,1}(\sigma_1(I^A) - \sigma_1(z)) \\
&= f^A_{1,1,1}(200 - 25) + f^A_{1,1,3}(215 - 200) + f^A_{17,1,1}(25 - 0) \\
&= f^A_{1,1,1}(175) + f^A_{1,1,3}(15) + f^A_{17,1,1}(25) \\
&= 10 + 5 + 25 \\
&= 40
\end{aligned}$$

$$\begin{aligned}
p^B_t(\sigma_1) &= \sum_{f^B_{wxy} \in T | \text{sat}_w(\sigma_1, d_{wx})} f^B_{wxy}(\sigma_1(v_{j_{wxy}}) - \sigma_1(v_{i_{wxy}})) \\
&= f^B_{2,2,1}(\sigma_1(v_{j_{2,2,1}}) - \sigma_1(v_{i_{2,2,1}})) + f^B_{6,1,1}(\sigma_1(v_{j_{6,1,1}}) - \sigma_1(v_{i_{6,1,1}})) \\
&= f^B_{2,2,1}(\sigma_1(M^B_E) - \sigma_1(M^B_S)) + f^B_{6,1,1}(\sigma_1(M^B_S) - \sigma_1(L^B_E)) \\
&= f^B_{2,2,1}(215 - 200) + f^B_{6,1,1}(200 - 180) \\
&= f^B_{2,2,1}(15) + f^B_{6,1,1}(20) \\
&= 5 + 0 \\
&= 5
\end{aligned}$$

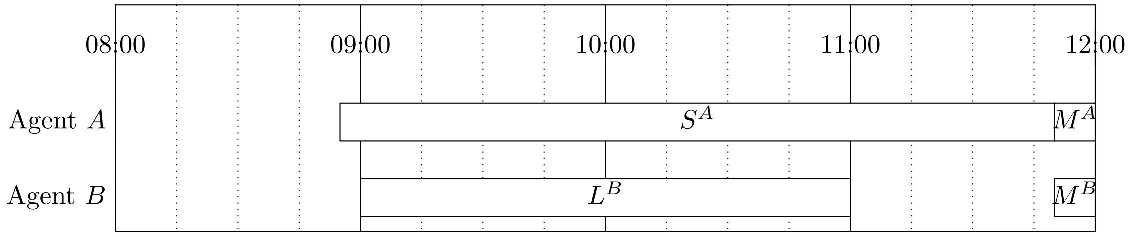
Computations for the other solutions are similar; all preference values are given in Figure 4.4.



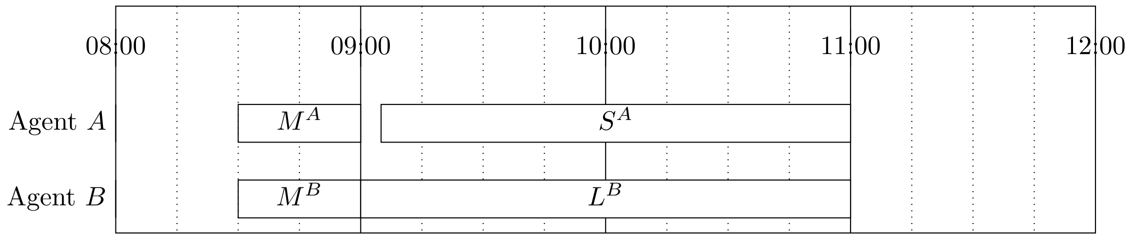
(a) Schedule σ_1: Initial MaDTPP schedule with meeting starting at 11:20. Preference values are $p_d^A(\sigma_1) = 0$, $p_d^B(\sigma_1) = 15$, $p_t^A(\sigma_1) = 40$, and $p_t^B(\sigma_1) = 5$.



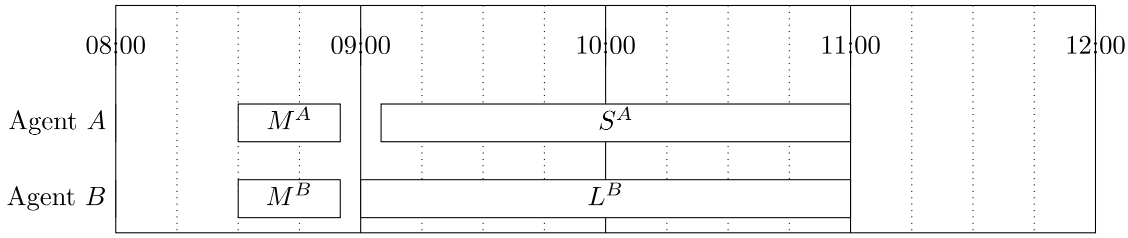
(b) Schedule σ_2: Pareto improvement (gain of 10 at time point level for agent A, gain of 30 at time point level for agent B) on initial MaDTPP schedule ($p_d^A(\sigma_2) = 0$, $p_d^B(\sigma_2) = 15$, $p_t^A(\sigma_2) = 50$, and $p_t^B(\sigma_2) = 35$).



(c) Schedule σ_3: Pareto improvement (gain of 25 at time point level for agent A, gain of 5 at time point level for agent B) on MaDTPP schedule of Figure 4.4b ($p_d^A(\sigma_3) = 0$, $p_d^B(\sigma_3) = 15$, $p_t^A(\sigma_3) = 75$, and $p_t^B(\sigma_3) = 40$).



(d) Schedule σ_4: Non-Pareto alteration (gain of 45 at disjunct level for agent A, loss of 15 at disjunct level for agent B) on initial MaDTPP schedule ($p_d^A(\sigma_4) = 45$, $p_d^B(\sigma_4) = 0$, $p_t^A(\sigma_4) = 45$, and $p_t^B(\sigma_4) = 5$).



(e) Schedule σ_5: Pareto improvement (gain of 5 at time point level for agent A, gain of 20 at time point level for agent B) on MaDTPP schedule of Figure 4.4d ($p_d^A(\sigma_5) = 45$, $p_d^B(\sigma_5) = 0$, $p_t^A(\sigma_5) = 50$, and $p_t^B(\sigma_5) = 25$).

Figure 4.4: MaDTPP schedules with different preference values.

4.4 Analysis

Note that schedule σ_1 (Figure 4.4a) is an extension of a decoupling found by the MaDTP-TD algorithm of Boerkoel & Durfee (2013a) (see Section 2.1.4). Once agent A had submitted its influence space in which $d_{1,1}$ was selected and agent B had submitted its influence space in which $d_{6,1}$ was selected, the coordinator reported a solution for these influence spaces. Next, the agents applied the MaTDR algorithm to the corresponding component STPPs. This process is the same as the process described in Section 3.3. Again, given the decoupling, we chose σ_1 such that it is an optimal solution with S^A as short as possible.

However, in this case, the decoupling could have been better with respect to preference values at time point level. Due to $f^B_{6,1,1}$, agent B prefers to start the meeting not before 11:30. For agent A, this is also desirable, since I^A can be later in that case, giving a higher preference due to $f^A_{17,1,1}$. Hence, both agents will agree to schedule σ_2 (Figure 4.4b). Furthermore, postponing the meeting further will be acceptable for both agents (no gain for agent B, but still for agent A due to $f^A_{17,1,1}$), and also a shorter meeting will be acceptable for the agents due to $f^A_{1,1,3}$ and $f^B_{2,2,1}$. Hence, they will agree on σ_3 (Figure 4.4c). Note that the preference values at disjunct level are equal for schedules σ_1, σ_2, and σ_3.

However, contrary to the MaSTPP, the MaDTPP also allows other orders of the tasks, by which the preferences at disjunct level can change. Agent A would prefer schedule σ_4 (Figure 4.4d) to the initial schedule σ_1, since its preference value at disjunct level is 45 instead of 0. Agent B will not accept this change since it is no Pareto improvement (the preference value at disjunct level for agent B decreases with 15). However, in a situation in which the agents do not only accept Pareto improvements, but are aimed at maximizing social welfare, selecting the decoupling of σ_4 instead of the decoupling of σ_1 could be acceptable for agent B. Then, given the decoupling of σ_4, the agents could establish a Pareto improvement for their preferences at time point level by shortening the meeting time (see σ_5, Figure 4.4e). The short break between the two activities will be highly beneficial for agent B, due to $f^B_{6,2,1}$.

We have defined the MaDTPP framework and have seen that within this framework, agents are able to prefer certain solutions to other ones. We analyzed different improvements on solutions, and showed that comparing solutions fits well within a negotiation approach. As described by research questions *RQ5* and *RQ6* in Section 2.3, we want to find automated negotiation methods for obtaining solutions that are preferred by the agents, instead of random solutions for the disjunctive problem (Boerkoel & Durfee, 2013a). Our approach for solving MaDTPPs will be presented in Chapter 6.

Chapter 5

Solving the Multi-Agent Simple Temporal Problem with Preferences

When applying automated negotiation to temporal problems, it seems most natural to consider the temporal variables in a temporal problem as the issues to which values will be assigned by negotiation. The main difference, however, between existing automated negotiation applications and an application of automated negotiation in temporal problems is that in the latter case not all assignments result in feasible schedules, while in the former case all choices of values in the domain are allowed. Furthermore, due to the different constraints, it is possible that different agents have different domains for the same temporal variables. A brute-force approach is possible in which existing negotiation algorithms are used as if each assignment is a possibility, and then afterwards it is calculated whether a proposal actually results in a feasible schedule. This, however, results in a great computational complexity, and biases the negotiation tactics. We therefore propose an approach in which the agents make only proposals that correspond with consistent temporal problems for themselves. This requires some preliminary calculations for the agents, before being able to make a proposal. For computational reasons, we try to keep these calculations as local as possible.

Another problem we encounter when applying negotiation approaches to temporal problems is that it is difficult to estimate the preferences of the opponents (see Section 2.2.2). Preference functions in temporal problems are not required to be linear or monotonic, but can be of any form. Furthermore, there is no small set of goals for the agents. Hence, an approach like the approach of De Weerd et al. (2016), in which the goal of an opponent is estimated based on its proposals, is not possible.

We therefore propose rather elementary automated negotiation algoritms for Multi-agent Simple Temporal Problems with Preferences, both for the post-decoupling and for the pre-decoupling negotiation approach. In the first approach, we assume that a temporal decoupling for the problem is given, and try to locally improve this decoupling. In the second approach, on the other hand, we try to improve the algorithm that established the decoupling.

5.1 Post-decoupling negotiation approach

Given a decoupling for an MaSTPP, our goal is to negotiate about the assigned times of the already fixed time points to find a Pareto improvement on the decoupling with respect to the expected preferences of the agents. We focus on changing the assigned values of temporal variables that have been fixed by the decoupling into values that have higher preference values associated with them, since only for these variables a preference value is known in the decoupling. (Recall from Section 2.1.4 that a decoupling of an MaSTP(P) is a set of STP(P)s, and from Section 3.2 that preference values for (Ma)STPPs are dependent on the assigned values to the variables.) Although the decoupling influences the possible values for other temporal variables of the agents, preference values for them can be given only when an exact solution is known.

We propose an alternating offers protocol for two or more agents in which the agents can make proposals in turns. In contrast to most alternating offers protocols, we allow further negotiation when a certain proposal is accepted, since we focus on local improvements. There can be multiple local improvements, possibly of different agents, that will be accepted by the other agents.

5.1.1 Protocol

In general, our algorithm is as follows: If it is an agent's turn to make a proposal, the agent selects a time point v_k that is fixed by the decoupling (i.e. for the updated bounds w_{kz} and w_{zk} it holds that $-w_{kz} = w_{zk}$) and that has another possible value in its original domain $[-b_{kz}, b_{zk}]$ for which the preference value is higher than the preference value for w_{zk}. The agent selects a not earlier proposed value x in $[-b_{kz}, b_{zk}]$ that has maximal preference value below a certain expected utility value (see Section 5.1.2). Subsequently, the agent changes w_{kz} and w_{zk} by assigning $-x$ and x to them, respectively, and tries to make its local STP consistent by local constraint propagation as described below (see Section 5.1.3 and Algorithm 1). For this, also bounds on interface variables are allowed to be changed: That is where the change will influence other agents. If indeed a consistent local STP is found with the new bounds on v_k, for which also the total preference gain is positive (i.e., the sum of the preference values for all fixed time points in the new STP is more than the sum of the preference values for all fixed time points in the old STP), the new STP will be proposed to other agents by communicating the changes on all the interface variables to the agents that know them.

The other agents then check whether it is possible for them to make their own local problems consistent with the proposed changes, also using Algorithm 1. However, to put a bound on the computational depth, it is not allowed for them to propagate changes in the constraints again to other agents, i.e. bounds on their interface variables not directly related to the interface variables of the proposing agent may not be changed. If all agents can make their local STPs consistent and their preference gain is above their expected utility value for the current nodes, then the proposal will be accepted and all changes will be established.[1] After an agent made a proposal, the next agent can make a proposal. If an agent doesn't have a new proposal since there are no consistent STPs with more preferred values for the fixed time points, the agent simply skips its turn and communicates to the next agent that it can make a proposal.

5.1.2 Expected utility

For the expected utility of an agent, we use a function frequently used in the automated negotiation literature (e.g. Wooldridge, 2009; Ren & Zhang, 2014; Baarslag, 2016). The expected utility of an agent is based on the current negotiation round number (r), the maximum allowed number of negotiation rounds (R), the minimum (L) and maximum (U) preference values that are obtainable for the selected node, and a concession parameter (ψ) as follows:

$$E(r, R, L, U, \psi) = L + (U - L) \cdot \left(1 - \frac{r}{R}^{\frac{1}{\psi}}\right) \tag{5.1}$$

such that for $\psi = 1$ the agent concedes linearly, for $\psi < 1$ the expected utility decreases mainly at the end of the negotiation process (a boulware function), and for $\psi > 1$, the expected utility decreases mainly at the start of the negotiation (a conceder function).

[1]Another option is to let the agents accept if their preference gain is not negative, but this can result in situations in which the agents are too cooperative: Assume for example that agent A proposes a change that results in a gain of 10 for itself, but induces no gain for agent B. If agent B accepts this proposal, a proposal with gain 8 for both agents relative to the initial situation will not be accepted by agent A in the next round, although it is considered to result in a global better solution.

5.1.3 Obtaining local consistency

The algorithm for making a local STP consistent when some bounds have been changed, is given in Algorithm 1. The main idea is that changes on a node in a PPC graph can be propagated to the neighbors until they fit well within the constraints of the original PPC graph, or until they reach the interface nodes, for which other bounds can be proposed to other agents. If this is not possible, the proposed change will be considered impossible.

If the agent executing the algorithm is the owner of the node that is changed, it will check whether the change is consistent with respect to the original bounds of the problem, and compute the gain in preference values (lines 3–10). For not-owning agents, this is not possible; for them, the difference in preference value is only dependent on their own nodes that need to be changed. Then, it will be recorded that the current node has been considered (line 11) and the change will be applied (line 12).

The propagation from the changes on a node v_k then will be done as follows: For all neighboring nodes v_i that are owned by the agent itself or have been considered already (this is possible when changes are conducted on the node of another agent) (line 15), the bounds with respect to v_k and z will be updated. If the new bound on the edge from z to v_k is higher than the old one, this doesn't influence any of the bounds on node v_i, and no updates are needed. Else (line 16), if the difference between the old and the new bound on e_{zk} can be fully counterbalanced by broadening the bound on e_{ki} (line 17), this will be done (line 18) and no update on w_{zi} is needed. Hence, w_{ki} is changed and the temporal distance from z to v_i remains consistent. If, however, fully broadening w_{ki} is not possible (due to the original bound b_{ki}), this bound will be broadened to the extent possible (line 20) and the other part of the change will be propagated to w_{zi} (line 21): The difference of w_{zk}^{old} and w_{zk}^{new} (w_{zi} is tightened by this amount) is subtracted from the orignal value of w_{zi}, and the difference of the new and old bound on e_{ki} (w_{zi} can in turn be broadened by this amount) is added. (Note that in lines 18, 20 and 21, there are minimum and maximum functions defined, these are useful in the case that some bounds were already broader than necessary.) In lines 24–31, the same computations are done for w_{ik} and w_{kz}.

Next, in lines 32–34, the gain in preference values will be computed for bounds between v_i and v_k (note that for the bounds between v_i and z, this will be done in lines 7–9 in the next call of the algorithm), and these bounds will be updated (line 35).

Finally, the new bounds between v_i and z are considered. If these are inconsistent with earlier changes in this process on v_i (earlier changes are made when $v_i \in N$), then the propagation is inconsistent; else, the new bounds are applied (lines 36–42). If $v_i \notin N$, then the consistency need to be checked for the new values of w_{zi} and w_{iz} (line 50). However, before this will be done, the interval for v_i is updated if only one of the bounds was changed due to which the other became inconsistent (lines 43–48).

Algorithm 1 Obtaining local consistency after changes

V and E are the vertices and edges of the network, P contains the proposed nodes v_k with new bounds w_{zk}^{new} and w_{kz}^{new}, N contains the already encountered nodes, and g the gain in preference values. The algorithm returns whether consistency can be obtained, and optionally g and N. The variables w^{old} and $w^{original}$ refer to the bounds just before the change of a value, and to the original bounds when entering the negotiation process, respectively; b refers to the bounds of the original MaSTPP.

1: **procedure** MAKECONSISTENT(V, E, P, N, g)
2:     **for all** $v_k \in P$ **do**
3:         **if** owns(v_k) **then**
4:             **if** $w_{zk}^{\text{new}} > b_{zk} \vee w_{kz}^{\text{new}} > b_{kz}$ **then**
5:                 **return** inconsistent
6:             **end if**
7:             **if** $w_{zk}^{\text{new}} = -w_{kz}^{\text{new}} \wedge w_{zk} = -w_{kz}$ **then** ▷ Continuation on next page

---

Algorithm 1 (Continued) Obtaining local consistency after changes

$g \leftarrow g + \text{pref}(w_{zk}^{\text{new}}) - \text{pref}(w_{zk})$
end if
end if
$N \leftarrow N \cup k$
$w_{zk} \leftarrow w_{zk}^{\text{new}}, w_{kz} \leftarrow w_{kz}^{\text{new}}$
end for
for all $v_k \in P$ **do**
for all $v_i \in V$ s.t. $e_{ik} \in E \wedge (\text{owns}(v_i) \vee v_i \in N)$ **do**
if $w_{zk}^{\text{new}} < w_{zk}^{\text{old}}$ **then**
if $w_{zk}^{\text{old}} - w_{zk}^{\text{new}} + w_{ki} \leq b_{ki}$ **then**
$w_{ki}^{\text{new}} \leftarrow \min(w_{ki} + w_{zk}^{\text{old}} - w_{zk}^{\text{new}}, w_{zi} - w_{zk}^{\text{new}})$
else
$w_{ki}^{\text{new}} \leftarrow \min(b_{ki}, w_{zi} - w_{zk}^{\text{new}})$
$w_{zi}^{\text{new}} \leftarrow \max(w_{zi}^{\text{original}} - (w_{zk}^{\text{old}} - w_{zk}^{\text{new}}) + (w_{ki}^{\text{new}} - w_{ki}), w_{ki}^{\text{new}} + w_{zk}^{\text{new}})$
end if
end if
if $w_{kz}^{\text{new}} < w_{kz}^{\text{old}}$ **then**
if $w_{kz}^{\text{old}} - w_{kz}^{\text{new}} + w_{ik} \leq b_{ik}$ **then**
$w_{ik}^{\text{new}} \leftarrow \min(w_{ik} + w_{kz}^{\text{old}} - w_{kz}^{\text{new}}, w_{iz} - w_{kz}^{\text{new}})$
else
$w_{ik}^{\text{new}} \leftarrow \min(b_{ik}, w_{iz} - w_{kz}^{\text{new}})$
$w_{iz}^{\text{new}} \leftarrow \max(w_{iz}^{\text{original}} - (w_{kz}^{\text{old}} - w_{kz}^{\text{new}}) + (w_{ik}^{\text{new}} - w_{ik}), w_{ik}^{\text{new}} + w_{kz}^{\text{new}})$
end if
end if
if $w_{ik} = -w_{ki} \wedge w_{ik}^{\text{original}} = -w_{ki}^{\text{original}}$ **then**
$g \leftarrow g + \text{pref}(w_{ki}) - \text{pref}(w_{ki}^{\text{original}})$
end if
$w_{ik} \leftarrow w_{ik}^{\text{new}}$, $w_{ki} \leftarrow w_{ki}^{\text{new}}$
if $v_i \in N$ **then**
if $w_{zi}^{\text{new}} < -w_{iz} \vee -w_{iz}^{\text{new}} > w_{zi}$ **then**
return inconsistent
else
$w_{iz} \leftarrow \min(w_{iz}, w_{iz}^{\text{new}}), w_{zi} \leftarrow \min(w_{zi}, w_{zi}^{\text{new}})$
end if
else
if $w_{zi}^{\text{new}} \neq w_{zi} \wedge w_{zi}^{\text{new}} < -w_{iz}$ **then**
$w_{iz}^{\text{new}} \leftarrow -w_{zi}^{\text{new}}$
end if
if $w_{iz}^{\text{new}} \neq w_{iz} \wedge w_{iz}^{\text{new}} < -w_{zi}$ **then**
$w_{zi}^{\text{new}} \leftarrow -w_{iz}^{\text{new}}$
end if
if $w_{iz}^{\text{new}} \neq w_{iz} \vee w_{zi}^{\text{new}} \neq w_{zi}$ **then**
if $\neg(g, N \leftarrow \text{MAKECONSISTENT}(V, E, \{i, w_{iz}^{\text{new}}, w_{zi}^{\text{new}}\}, N, g))$ **then**
return inconsistent
end if
end if
end if
end for
end for
return consistent, g, N
end procedure

5.1.4 Example

We illustrate the negotiation protocol with an adaptation of Example 3.1. We assume the following MaSTPP, in which preferences are defined on constraints on variables that will be fixed by a decoupling, relative to the reference time point, since our protocol works on such variables. For the student, the start time of the meeting is preferred to be as late as possible, since a later start time in the morning is possible then. Furthermore, we assume here that the professor prefers to have the meeting right after the lecture (within 5 minutes) or that there is a pause of at least half an hour between the lecture end time and the meeting start time. However, the professor does not want to end up too late, resulting in a linear decrease in preference values if the meeting starts later than 11:30.[2]

- $V = \{z, S_S^A, S_E^A, M_S^A, M_E^A, M_S^B, M_E^B, L_S^B, L_E^B\}$
- $C =$

$$
\begin{aligned}
\{\ c_1 :&\quad 175 \leq S_E^A - S_S^A \leq 185,\\
c_2 :&\quad 10 \leq M_E^A - M_S^A \leq 20,\\
c_3 :&\quad 0 \leq M_S^A - S_E^A \leq \infty,\\
c_4 :&\quad 120 \leq L_E^B - L_S^B \leq 120,\\
c_5 :&\quad 10 \leq M_E^B - M_S^B \leq 20,\\
c_6 :&\quad 0 \leq M_S^B - L_E^B \leq \infty,\\
c_7 :&\quad 0 \leq M_S^B - M_S^A \leq 0,\\
c_8 :&\quad 0 \leq M_E^B - M_E^A \leq 0,\\
c_9 :&\quad 0 \leq S_S^A - z \leq 240,\\
c_{10} :&\quad 0 \leq S_E^A - z \leq 240,\\
c_{11} :&\quad 0 \leq M_S^A - z \leq 240,\\
c_{12} :&\quad 0 \leq M_E^A - z \leq 240,\\
c_{13} :&\quad 30 \leq M_S^B - z \leq 240,\\
c_{14} :&\quad 30 \leq M_E^B - z \leq 240,\\
c_{15} :&\quad 60 \leq L_S^B - z \leq 60,\\
c_{16} :&\quad 180 \leq L_E^B - z \leq 180\ \}
\end{aligned}
$$

- $G = \{A, B\}$
- $g^{-1}(A) = \{S_S^A, S_E^A, M_S^A, M_E^A\}$ and $g^{-1}(B) = \{M_S^B, M_E^B, L_S^B, L_E^B\}$
- $F = \{f_{11}^A, f_{13}^B, \cdots\}$, where
  - $f_{11}^A(x) = \begin{cases} x - 175 & \text{if } 175 \leq x \leq 240 \\ 0 & \text{otherwise} \end{cases}$
  - $f_{13}^B(x) = \begin{cases} 30 & \text{if } 180 \leq x \leq 185 \\ 240 - x & \text{if } 210 \leq x \leq 240 \\ 0 & \text{otherwise} \end{cases}$

Application of the post-decoupling negotiation protocol with a maximum of 20 negotiation rounds on a standard decoupling in which $M_S^A = M_S^B = 200$ and $M_E^A = M_E^B = 215$ (the meeting starts at 11:20 and ends at 11:35) with preference value 25 for agent A and preference value 0 for agent

[2] This choice makes our example more interesting, since the parties have conflicting interests now.

	Proposing agent			Opponent		
Round nr.		Expected gain	Proposal $\langle M_S, M_E \rangle$	Expected gain	Gain	Action
1	A	40	$\langle 230, 240 \rangle$	30	10	Reject
2	B	27	$\langle 213, 223 \rangle$	36	13	Reject
3	A	33	$\langle 229, 239 \rangle$	24	11	Reject
4	B	23	$\langle 217, 227 \rangle$	30	17	Reject
5	A	28	$\langle 228, 238 \rangle$	21	12	Reject
6	B	19	$\langle 221, 231 \rangle$	26	21	Reject
7	A	24	$\langle 224, 234 \rangle$	18	16	Reject
8	B	16	$\langle 224, 234 \rangle$	22	24	Accept
9	A	8	$\langle 227, 237 \rangle$	7	-3	Reject
10	B	6	$\langle 218, - \rangle$	7	-6	Reject
11	A	6	$\langle 226, 236 \rangle$	5	-2	Reject
12	B	5	$\langle 219, - \rangle$	5	-5	Reject
13	A	5	$\langle 225, 235 \rangle$	4	-1	Reject
14	B	3	$\langle 222, - \rangle$	4	-2	Reject
15	B	2	$\langle 223, - \rangle$	3	-1	Reject

Table 5.1: Negotiation progress of the example for a maximum of 20 negotiation rounds and initial decoupling values $\langle M_S^A = M_S^B = 200, M_E^A = M_E^B = 215 \rangle$. The notation $-$ in a proposal denotes that no new value is proposed for the corresponding variable.

B, results in the following negotiation: First, agent A expects a gain of 40 on M_S^A since the value 240 results in the maximal preference value of 65. However, local consistency cannot be obtained for this value. The value 230 is the one with maximal preference value that results in a local consistent solution for agent A. Hence, agent A makes the proposal $\langle M_S^A = 230, M_E^A = 240 \rangle$. Agent B, however, expects a gain of 30 (which could be obtained for assigning the value 210 or a value in $[180, 185]$ to M_S^B), and hence rejects the proposal. In the next round, agent B can make a proposal, but due to time progress, its expected utility has decreased to 27. Hence, agent B proposes $\langle M_S^B = 213, M_E^B = 223 \rangle$, which has preference value 27 for agent B. Agent A expects in this round a gain of 36, and does not accept this offer with gain 13. The negotiation goes on as shown in Table 5.1. Both agents reduce their expected gain and make corresponding proposals. In round 7, agent B rejects the proposal $\langle M_S^A = 224, M_E^A = 234 \rangle$, but in round 8, its expected gain has decreased and now agent B makes the same proposal. This proposal is accepted by agent A, and an interim new decoupling is found. After this change, no other improvement can be made, since an increase in gain for one of the agents corresponds to a decrease for the other. Note that after round 14, agent A can make no new proposal and hence, agent B is also the proposer in round 15. After round 15, also agent B has no new proposals, and hence, the negotiation ends. The preference values for the new decoupling $\langle M_S^A = M_S^B = 224, M_E^A = M_E^B = 234 \rangle$, established in round 8, are 49 and 16 for agents A and B, respectively. Hence, a Pareto improvement has been found with gain 24 and 16 for agents A and B, respectively.

5.2 Pre-decoupling negotiation approach

Given the MaTDR decoupling algorithm of Boerkoel & Durfee (2013b), we propose to incorporate automated negotiation into the algorithm to improve the resulting decoupling with respect to the preferences of the agents. As with the post-decoupling negotiation case, we only focus on the variables that will be assigned fixed values by the decoupling, since preferences are not known for other variables.

Recall from Section 2.1.4 that the assignment of values to shared variables in the decoupling algorithm of Boerkoel & Durfee (2013b) is done in reversed shared elimination order. The agent that owns the temporal variable assigns itself a value to it after updates for variables later in the elimination order have been received. To increase the flexibility of the decoupling solution, Boerkoel & Durfee (2013b) use a midpoint heuristic, that assigns the midpoint of the current updated domain $[-w_{kz}, w_{zk}]$ to variable v_k. When working with MaSTPPs, in contrast, it is naturally to select not the midpoint of the interval, but the value that results in the maximal preference value.

However, this possibly results in suboptimal schedules since the assigned values also influence the subproblems, and hence the preference values, of other agents. We therefore propose an approach in which the agents jointly negotiate about the values to be assigned. A multi-issue negotiation, in which all shared variables are negotiated together, will be preferred since a mutual gain will possibly be obtained due to trade-offs. The structure of the decoupling algorithm, however, impedes this, since domains are updated based on earlier assignments. Therefore, we propose a negotiation approach comparable to that in the post-decoupling case.

5.2.1 Protocol

The initiation of a negotiation procedure takes places in line 12 of the MaTDR algorithm (Boerkoel & Durfee, 2013b, p. 134), and replaces the current heuristic assignment function. The other invited agents enter the negotiation algorithm just before line 9, when they are blocking for updates on the variable.

The negotiation algorithm is as follows: For the current variable v_k, the agent that can make a proposal (initially the agent that owns v_k) selects the value that maximizes, between the lower bound of the current preference value (initially 0) and the upper bound of the expected utility (given by Formula 5.1), the preference value for the agent. If the agent is the owner of v_k, the preference value is simply given by the preference value corresponding to the value that will be assigned to v_k. Otherwise, the agent does not have a preference value corresponding to v_k, but the preference value will be based on all the nodes of the agent that are directly related to, and fixed by v_k. For all such variables v_i s.t. $w_{ik} = -w_{ki}$, the preference values corresponding to the value $w_{zk} - w_{ik}$ (where w_{zk} denotes the value that will be assigned to v_k) will be summed up.

Subsequently, other agents check whether the proposed value fits in their domain for node v_k, and whether the value results for them in a preference value that is above their own current preference value and above their expected utility. If this is the case for all involved agents, they accept. Otherwise, the next agent can make a proposal on node v_k. This procedure stops when no agent can find a proposal that increases its preference value, which will be automatically the case when the maximum number of rounds has been reached.

5.2.2 Example

Applying the pre-decoupling negotiation protocol to the MaSTPP of Section 5.1.4 proceeds as follows: The shared elimination order in our case is $\langle M_S^B, M_E^B, M_S^A, M_E^A \rangle$. Hence, first a negotiation about M_E^A takes place. Since no agent has preferences for that variable, the midpoint 215 of the range [190,240] is selected. Next, the value of M_S^A is negotiated. Agent A proposes from the updated range [195,205] its most preferred value, namely 205. Agent B does not expect any gain in the domain [195,205] since all preference values for agent B are equal to 0 in this domain. Hence, agent B accepts. Agent B has no new proposals for this variable, and neither has agent A. Hence, the value 205 is selected. Next, M_E^B and M_S^B are treated, but their domains were tightened to [215,215] and [205,205], respectively, due to the earlier assignments. Hence, no changes are possible for these variables. After the decoupling, the preference value for agent A is 30 and the preference value for agent B is 0. We analyse that these values are better than the values for the standard decoupling $\langle M_S^A = M_S^B = 200, M_E^A = M_E^B = 215 \rangle$, but worse than the values obtained in the post-decoupling negotiation approach from Section 5.1.4. This is due to the non-existence of preference values on M_E^A and M_E^B, and the lack of considering propagations of choices in the negotiation algortihm (see also Chapter 8).

We adapt our example to illustrate the potential of the negotiation process. To the MaSTPP of Section 5.1.4, we add the following preference functions on M_E^A and M_E^B, that imitate the preference functions on M_S^A and M_S^B under assumption of a duration of 10 minutes for the meeting:

- $f^A_{12}(x) = \begin{cases} x - 185 & \text{if } 185 \leq x \leq 240 \\ 0 & \text{otherwise} \end{cases}$

- $f^B_{14}(x) = \begin{cases} 30 & \text{if } 190 \leq x \leq 195 \\ 250 - x & \text{if } 220 \leq x \leq 240 \\ 0 & \text{otherwise} \end{cases}$

The pre-decoupling negotiation protocol applied to this updated MaSTPP gives better results:

First, agent A proposes the value 240 (preference value 55) for M^A_E. Agent B rejects this proposal, since the preference value for agent B is too low. Then, agent B proposes in the second round the value 221 for M^A_E, since M^A_E is directly related to M^B_E, and hence, agent B would obtain the preference value 29 corresponding to $M^B_E = 221$. However, the preference value for this proposal is too low for agent A. In 15 rounds, the agents finally agree on the value 227 for M^A_E. Next, the domain for M^A_S has become [207,217] and agent A proposes the value 217 for it, resulting in a preference value of 42. Agent B, considering the implications on M^B_S, rejects this offer, and proposes a value of 211. After the proposals of 214 and 213, the value 213 is accepted, and hence assigned to M^A_S. Again, M^B_E and M^B_S are fixed by the choices for M^A_E and M^A_S, and hence, these variables cannot be negotiated anymore. The resulting decoupling $\langle M^A_S = M^B_S = 213, M^A_E = M^B_E = 227\rangle$ has preference values 80 and 50 for agents A and B, respectively. The preference values for this decoupling according to the original framework of Section 5.1.4 are 38 and 27 for agents A and B, respectively. Hence, this solution is less preferable to agent A than the post-decoupling negotiation solution, but more preferable to agent B. The social welfare is $38 + 27 = 65$, and equals that of the post-decoupling negotiation solution.

Chapter 6

Solving the Multi-Agent Disjunctive Temporal Problem with Preferences

When applying automated negotiation for improving solutions to Multi-agent Disjunctive Temporal Problems with Preferences, we encounter the same problems as with Multi-agent Simple Temporal Problems with Preferences, as described in Chapter 5: In contrast to standard negotiation approaches, assigning values to the temporal variables is highly restricted, and due to the unpredictable nature of preference functions of opponents and their wide range of possible goals, estimation of proposals that will be advantageous for them is highly complex. In this chapter, we therefore develop some rather elementary negotiation approaches. In Chapter 8, we suggest some ideas for more refined protocols.

The MaDTPP was defined with preference values both at disjunct level and at time point level (see Section 4.3). Preference values at disjunct level are dominant: A solution with a higher preference value at disjunct level is always preferred to a solution with a lower preference value at disjunct level, regardless of the preference values at time point level. In this chapter, we therefore focus on negotiation about the satisfaction of the disjunctive constraints of an MaDTPP. Note that when the disjunctive constraints have been selected, a solution refinement can be made based on the preference values at time point level. In fact, an MaDTPP for which the disjuncts have been selected reduces to an MaSTPP, and hence can be further solved by the approaches described in Chapter 5.

We describe approaches to negotiate about the disjuncts that will be selected, both for making changes in an already existing decoupling (Section 6.1) and for choosing appropriate disjuncts during the decoupling algorithm (Section 6.2).

6.1 Post-decoupling negotiation approach

Given a decoupling to an MaDTPP, we want to improve this solution with respect to the preference value at disjunct level for the different agents by letting them make local changes in the disjuncts that have been selected for being satisfied. Instead of a decoupling consisting of local DTPPs (see Definition 2.2), we work with local STPPs, as in the case of an MaSTPP decoupling (see Definition 2.1), since this is more practical in our negotiation approach (e.g., calculating the consistency of a simple problem after disjunct changes is less complex than calculating the consistency of a disjunctive problem). Getting an STPP from a local DTPP can be done by the agents individually, this does not affect the generality of the problem.

In general, agents need to check for themselves what the influences of satisfying another disjunct are. In general, this will give other bounds on interface time points, and hence influences the decoupling. The other agents that are affected by the proposed changes then need to check whether their local problems can be made consistent regarding the changes. We propose a protocol comparable to the post-decoupling negotiation protocol for MaSTPPs, in which agents alternate in doing offers. The main difference is that an algorithm for obtaining local consistency after disjunct changes will be much more complex than the algorithm of Section 5.1.3. This is caused by the fact that changing a disjunct has a high influence on the local graph: Plainly removing the edges that are described by the temporal constraints of the removed disjunct is not possible, since these edges can have been influenced by other constraints due to the graph processing methods of the decoupling algorithm. On the other hand, after adding the edges corresponding to the new disjunct, the graph needs to be made consistent if this is not automatically the case. In contrast to the consistency algorithm of Section 5.1.3, the new edge can be between two nodes that were not connected before. This can result in another structure of the graph, e.g. non-bisected cycles with a length of more than 3 can arise, making the graph no longer a triangulated one, and hence, disturbing the PPC properties. Obtaining consistency again needs a more complex method than the rather elementary constraint propagation method used for the MaSTPP case. We skip developing such a local mehod, since we focus on the negotiation approach. Instead, we use an existing approach for computing a consistent solution for the whole changed subproblem of the agent again.

6.1.1 Protocol

The negotiation algorithm is broadly as follows: The agent that is expected to make a proposal loops through its set of DNF constraints, until it encounters a constraint for which a non-optimal disjunct has been selected. It selects a better disjunct, removes the old disjunct, and applies a general △PPC algorithm to check for consistency. If this does not result in a consistent solution, the next possible disjunct will be selected. Else, the agent checks for which interface nodes of its graph the bounds with respect to the reference time point have been changed. These changes then are proposed only to the agents that are aware of these nodes due to their external constraints.

For these agents, a local check for modification is hard, due to the thorough influences of selecting other disjuncts. Also, a global check with the △PPC algorithm is not possible, since it is not known beforehand which disjuncts for these agents need to be changed. Hence, the agents construct the DTPP in which the proposed changes have been applied and in which the bounds on their own other shared nodes are equal to the original decoupling bounds. Then, the MaDTP-TD algorithm is applied by the participating agents (i.e., the set of agents that were affected by a part of the proposed change) to find a decoupling respecting these constraints. If such a decoupling is not found, the proposal will be rejected. Otherwise, the proposal will only be accepted if the preference gain at disjunct level for all participating agents is not negative. In either case, the next agent will be allowed to make a proposal, until no agent can make another proposal. Note that we do not work with expected preferences since it is not clear for all agents beforehand which of their disjuncts will be affected due to the proposed change. Hence, all Pareto improvements will be accepted directly.

6.1.2 Example

We illustrate the protocol with a simplified version of the MaDTPP of Example 4.1. We need only the preferences at disjunct level, and assume that only agent A has such preferences, which are related to the total length of the tasks. Hence, we show the performance of the algorithm on the following MaDTPP, where all preference functions other than f_1^A map all values to 0:

- $V = \{v_0, S_S^A, S_E^A, M_S^A, M_E^A, M_S^B, M_E^B, L_S^B, L_E^B\}$
- $C =$

$$\begin{aligned}
\{\ c_1 :\quad & d_{1,1} :\ (175 \leq S_E^A - S_S^A \leq 185\ \wedge\ 0 \leq M_S^A - S_E^A \leq \infty\ \wedge\ 10 \leq M_E^A - M_S^A \leq 20) \\
& \vee\, d_{1,2} :\ (115 \leq S_E^A - S_S^A \leq 125\ \wedge\ 0 \leq S_S^A - M_E^A \leq \infty\ \wedge\ 25 \leq M_E^A - M_S^A \leq 35), \\
c_2 :\quad & d_{2,1} :\ 25 \leq M_E^B - M_S^B \leq 35 \quad \vee \quad d_{2,2} :\ 10 \leq M_E^B - M_S^B \leq 20, \\
c_3 :\quad & d_{3,1} :\ 120 \leq L_E^B - L_S^B \leq 120,
\end{aligned}$$

$$
\begin{aligned}
c_4: &\quad d_{4,1}: \ 0 \leq M_S^B - M_S^A \leq 0,\\
c_5: &\quad d_{5,1}: \ 0 \leq M_E^B - M_E^A \leq 0,\\
c_6: &\quad d_{6,1}: \ 0 \leq M_S^B - L_E^B \leq \infty \quad \vee \quad d_{6,2}: \ 0 \leq L_S^B - M_E^B \leq \infty,\\
c_7: &\quad d_{7,1}: \ 0 \leq S_S^A - z \leq 240,\\
c_8: &\quad d_{8,1}: \ 0 \leq S_E^A - z \leq 240,\\
c_9: &\quad d_{9,1}: \ 0 \leq M_S^A - z \leq 240,\\
c_{10}: &\quad d_{10,1}: \ 0 \leq M_E^A - z \leq 240,\\
c_{11}: &\quad d_{11,1}: \ 30 \leq M_S^B - z \leq 240,\\
c_{12}: &\quad d_{12,1}: \ 30 \leq M_E^B - z \leq 240,\\
c_{13}: &\quad d_{13,1}: \ 60 \leq L_S^B - z \leq 60,\\
c_{14}: &\quad d_{14,1}: \ 180 \leq L_E^B - z \leq 180 \ \}
\end{aligned}
$$

- $G = \{A, B\}$
- $g^{-1}(A) = \{S_S^A, S_E^A, M_S^A, M_E^A\}$ and $g^{-1}(B) = \{M_S^B, M_E^B, L_S^B, L_E^B\}$
- $D = \{f_1^A, \cdots\}$, where
  - $f_1^A(d_{1,1}) = 0$
  - $f_1^A(d_{1,2}) = 45$
- $T = \{\cdots\}$

Given the standard initial decoupling with $M_S^A = M_S^B = 200$ and $M_E^A = M_E^B = 215$, with preference value 0 at disjunct level for agent A, the negotiation proceeds as follows: Agent A, allowed to make a proposal, prefers to have disjunct $d_{1,2}$ instead of $d_{1,1}$. Hence, agent A removes the constraints of $d_{1,1}$ and adds the ones of $d_{1,2}$ to its current selection of constraints. The check for consistency has a positive result, and hence, agent A selects its interface variables for which the bounds with respect to the reference time point have been changed. That are M_S^A with new bounds 0 and 100 instead of 200 and 200 (i.e. $0 \leq M_S^A - z \leq 100$ needs to hold instead of $200 \leq M_S^A - z \leq 200$) and M_E^A with new bounds 25 and 125 instead of 215 and 215 ($25 \leq M_E^A - z \leq 125$ instead of $215 \leq M_E^A - z \leq 215$).[1] Since agent B is aware of M_S^A and M_E^A due to c_4 and c_5, agent A proposes the new bounds to agent B. Hence, agent B constructs his local DTPP for the changes, and the MaDTP-TD algorithm finds a solution.[2] Hence, the proposed change is consistent for all agents. Since agent B has no preferences at all, agent B accepts the change in disjuncts of agent A. Since no preferred disjuncts can be selected by any agent, the negotiation ends, and agent A has obtained a preference gain of 45.

6.2 Pre-decoupling negotiation approach

We propose to incorporate automated negotiation in the MaDTP Temporal Decoupling (MaDTP-TD) algorithm of Boerkoel & Durfee (2013a), to improve solutions with respect to preferences of the agents. Recall from section 2.1.4 that the MaDTP-TD algorithm consists of the following steps:

- **Preprocessing** First, an agent preprocesses its local DTP. All constraints that consist of only one disjunct form together an STP that needs to be satisfied and is part of the larger

[1]These values are explainable: The meeting lasts at least 25 minutes in this case, so M_E^A cannot take place before 8:25, and a study time of at least 115 minutes needs to be planned after M_E^A but before 12:00, so M_E^A cannot take place after 10:05.

[2]In particular, the one in which $M_S^A = M_S^B = 32$ and $M_E^A = M_E^B = 58$; however, the exact values are not important at this point and could be negotiated later on with the algorithms for MaSTPPs.

DTP. Hence, bound tightening on these constraints can be done using the D△PPC algorithm. Furthermore, any disjunct that is inconsistent with respect to these tightened constraints can be removed, and any subsumed constraint (i.e. a constraint that has an inherently satisfied disjunct) can also be safely removed.

- **Enumerating influence spaces of component STPs** Next, each agent uses an existing DTP solving approach to compute component STPs of its local DTP (e.g. Stergiou & Koubarakis, 2000; Tsamardinos & Pollack, 2003). From these, only the ones that result in different influence spaces (i.e., have different constraints on their interface variables) are sent to the coordinator, since only different influence spaces can give other possibilities for the shared DTP. When the coordinator finds a solution, the agent stops sending influence spaces to the coordinator.
- **Combining solutions** When an agent receives the interface variables part of the component STP that yields a shared solution, it will apply the MaTDR algorithm to it to coordinate the exact bounds with other agents, and subsequently, combine the result with its own local DTP solutions to enumerate its local solution space.

Our improvement will take place in the second step.[3] The main idea is that, instead of submitting influence spaces to the coordinator in a random order, the agents consider their influence spaces as separately valued bids and submit them in a deliberate order at deliberate times, based on a negotiation strategy. The exact protocol will be given in Section 6.2.2, but first we focus on SMT solvers and their extensions that enable optimization.

6.2.1 Optimization modulo theories

For enumerating influence spaces of component STPs, Boerkoel & Durfee (2013a) notice that any existing approach that computes component STPs of DTPs can be used. They themselves use the satisfiability modulo theories (SMT) solver Yices (Dutertre, 2014).[4] Recall that an SMT problem is a formula in first-order logic that needs to be satisfied, for which the predicates have additional interpretations in some theory. For DTP(P)s, the predicates are linear inequalities of the form $-b_{ji} \leq v_j - v_i \leq b_{ij}$, and need to be evaluated in the theory of linear real arithmetic. An SMT solution consists of an exact assignment of values to the temporal variables in V, but this solution can be relaxed by selecting the corresponding DTP disjuncts to obtain the component STP.

For our improvement on the MaDTP-TD algorithm, it is important for the agents not only to find possible component STPs, but also to find the ones that are preferred by them. Hence, agents want to find the component STPs with highest preference value at disjunct level possible. To obtain this, we decided to use an *optimization modulo theories* (OMT) solver (Sebastiani & Trentin, 2015) instead of a general SMT solver. OMT problems can be seen as extensions of SMT problems with some objective functions: An SMT solution that is optimal with respect to the objective functions needs to be found. Recently, some extensions for SMT solvers that facilitate queries for optimal SMT solutions have been developed: For the SMT solver Z3 (De Moura & Bjørner, 2008), Bjørner et al. (2015) developed νZ and as an extension to MathSAT5 (Cimatti et al., 2013), the OMT solver OptiMathSAT was developed (Sebastiani & Trentin, 2015). In our approach, we will use OptiMathSAT, and add maximization of the preference value at disjunct level, p_d^a, as defined in Section 4.3, as objective function.

[3] Note that another general, not particularly to preferences related improvement to the MaDTP-TD algorithm can be a repetition of the preprocessing step: Due to the removal of inconsistent disjuncts, it is possible that constraints initially consisting of more than one disjunct will have left only one disjunct after the preprocessing step. In a next round of preprocessing, these constraints will also be included in the STP that can be processed with the D△PPC algorithm, possibly resulting in new pruning opportunities. The process can be advantageously repeated until the STP of singleton constraints does not change anymore. The large reduction in solving time reported by Boerkoel & Durfee (2013a) for one execution of the preprocessing step suggests that a repetition of this step, reducing the disjunctive problem even more, might be highly beneficial for solving MaDTP(P)s, but this needs to be examined.

[4] We do not go into the details of solving SMT problems, since we consider SMT solvers as a given tool that can be used in our approach, but refer to Barrett et al. (2009) and Barrett & Tinelli (2017) for more details. Our choice for using SMT solvers was supported by the experience of J. C. Boerkoel that using an SMT solver ended up being much more efficient than writing an own solver (personal communication).

6.2.2 Protocol

The negotiation protocol is completely incorporated into the second step of the MaDTP-TD algorithm, and consists of submitting the different influence spaces at deliberate times. All the agents work in parallel using a mediator-based negotiation approach (see Section 2.2.3). In contrast to the original MaDTP-TD approach, activities of the agents are synchronized using bidding rounds. In a round, each agent is able to compute a new component STP for its local DTP. By applying the OMT solver to obtain this, a local solution that maximizes the preference value at disjunct level for the agent is found. This solution will be relaxed to a component STP by selecting the appropriate disjuncts. Next, the component STP will be made minimal and decomposable by applying an FPC algorithm (see Section 2.1.1) to it, such that the bounds on the interface variables will be accurate, and then the influence space will be obtained by selecting only the interface variables of the local problem and the constraints associated with them. Note that the preference value at disjunct level remains the same for the component STP and the influence space. A negation of the conjunction of all constraints of the influence space will be added to the OMT problem in the next round, such that only solutions with another influence space will be generated. In contrast to the original MaDTP-TD algorithm, influence spaces will not be directly sent to the coordinator, but are stored by the agent.

The moment of submitting the influence spaces is based on the expected utility of the agents. The maximal utility possible is equal to the preference at disjunct level of the solution that is found in the first round. An expected utility function as defined before (see Formula 5.1) is used to compute the expected utility in each round. After computing a new influence space and storing this, each agent submits in each round its stored non-submitted influence spaces that have a preference value not below the current expected utility value.[5]

As soon as the coordinator finds a global solution, the agents stop computing and submitting new influence spaces and accept the solution of the coordinator. As in the original MaDTP-TD algorithm, the MaTDR algorithm is then applied to coordinate the exact decoupling bounds with other agents.

6.2.3 Example

The following illustrates the pre-decoupling negotiation protocol for MaDTPPs in a total of 50 rounds. We take the MaDTPP from Section 6.1.2, but add again f_2^B to represent that both agents prefer to use as less total time as possible:

- $f_2^B(d_{2,1}) = 0$
- $f_2^B(d_{2,2}) = 15$

In the first negotiation round, both agents select their component STP that maximizes their preference value at disjunct level. For agent A, this is the STP of Figure 6.1b, with a preference value of 45. For agent B, both the STPs of Figures 6.2a and 6.2b are preferred (preference value 15), but randomly, the STP of Figure 6.2a is selected. In this round, the expected utility of agent A is set equal to 45, and for agent B, the expected utility is defined to be 15. Both agents submit the influence space of their selected component STPs immediately, but the coordinator does not find a shared solution, since the meeting cannot be synchronized. In the next round, agent B selects the component STP of Figure 6.2b, again with preference value 15, and submits the influence space corresponding to this STP, since the preference value for this STP is not lower than its expected utility (in fact, the expected utility for agent B in this round is 14). Agent A selects in this round the component STP of Figure 6.1a, but does not submit this one since the preference value (0) is below agent A's expected preference value. The coordinator does again not find a solution: Although the domains of M_S^A and M_S^B and also the domains of M_E^A and M_E^B overlap

[5] For large problems, it is possible that not all influence spaces of an agent have been found when the maximum number of negotiation rounds has been reached. This can result in not finding any solution. To overcome this problem, the maximum number of negotiation rounds is added up with another 50 rounds if one of the agents is still finding new influence spaces in the 25th round before the last negotiation round.

(for the submitted influence spaces of Figures 6.1b and 6.2b), the lengths of the meetings in these component solutions are not compatible.

In the third and fourth round, agent B selects the influence spaces of Figures 6.2c and 6.2d, but does not submit them since the preference value for these component STPs is 0. Agent A does not find new influence spaces, but also refuses to submit its influence space from Figure 6.1a.

The next action takes place in round 46: At that point, agent B submits both its remaining influence spaces, since its expected utility has decreased to 0. For agent A, its expected utility is just above 0, and hence, agent A does not submit its last influence space. Now, the coordinator finds a solution in combining the influence spaces of Figures 6.1b and 6.2d. The coordinator immediately reports this result, and the agents accept the solution.[6] Hence, agent A has obtained a preference value of 45, while the preference value for agent B is equal to 0.

[6] Note that the MaTDR algorithm can be applied to find the exact decoupling bounds; for this example, however, knowing which disjuncts were selected is enough, since we focus on the preference value at disjunct level.

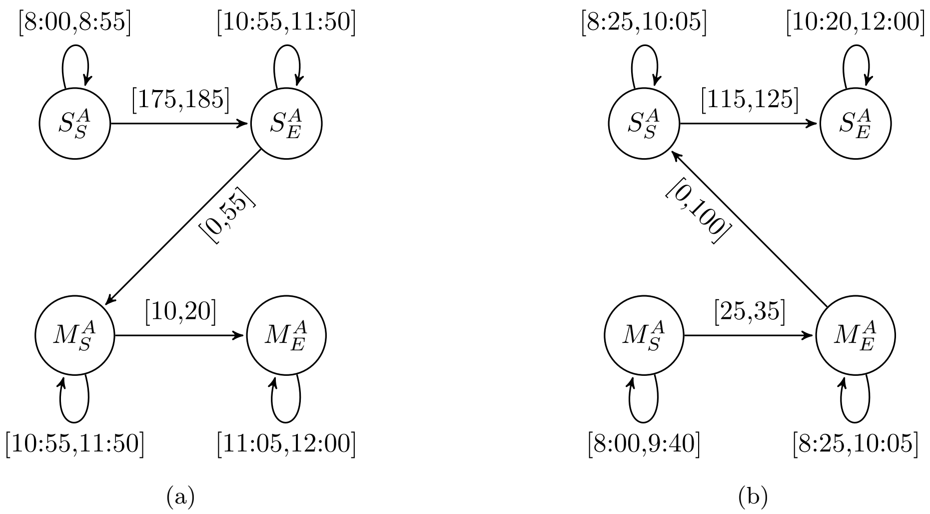


Figure 6.1: The two possible component STPs for agent A for the MaDTPP example.

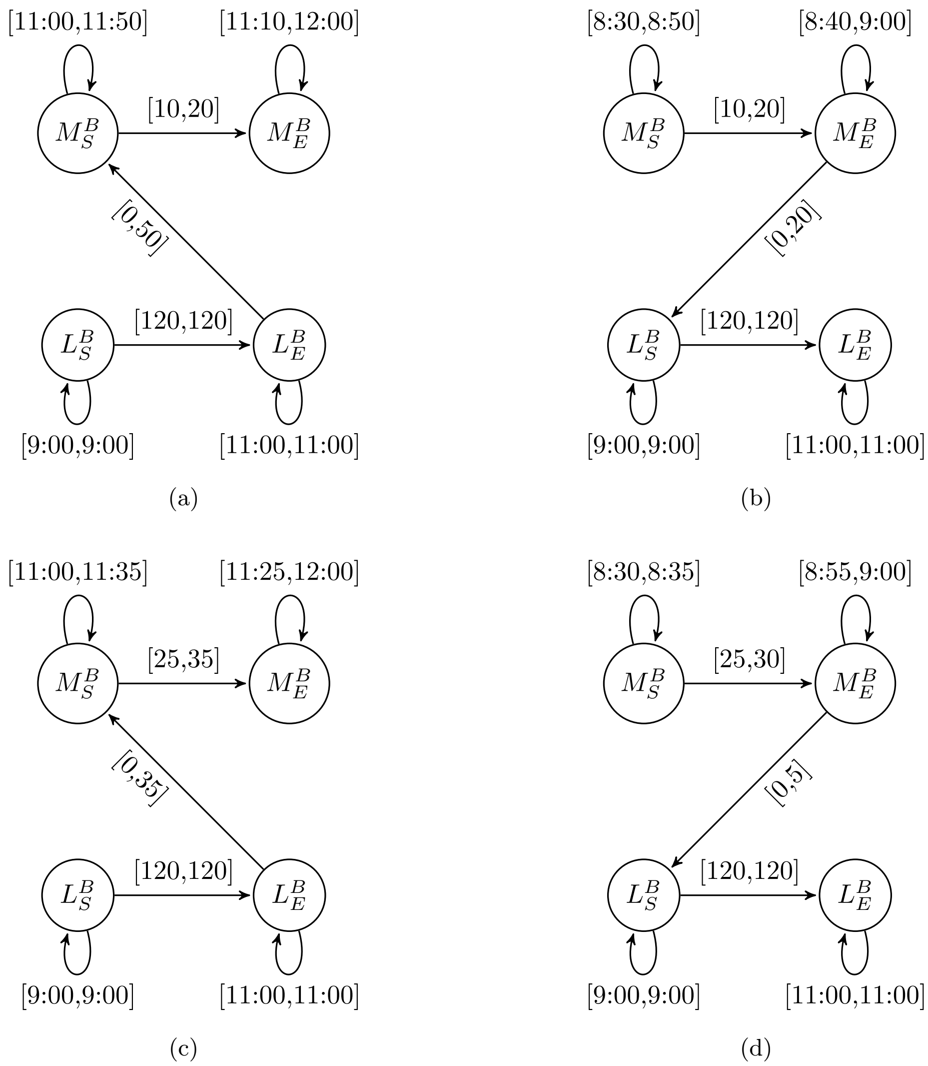


Figure 6.2: The four possible component STPs for agent B for the MaDTPP example.

Chapter 7

Experiments and Results

In this chapter, we examine the performance of the algorithms described in Chapters 5 and 6. Since there does not exist a benchmark set of temporal problems with preferences, we describe the construction of a problem set, based on the construction of other test sets of temporal problems. Next, we give some details of our implementation, we describe our experiments, and we conclude with the results.

7.1 Problem set

Since there is no benchmark set of temporal problems with preferences, we need to generate a set of random instances to test on. Comparable sets have been developed, but not for Multi-agent Temporal Problems with Preferences. We base the structure of our problem set on the construction of other random benchmarks (Boerkoel, Planken, Wilcox & Shah, 2013; Moffitt, 2011; Maratea & Pulina, 2014; Micalizio & Torta, 2015; Khatib et al., 2007). Our test set consists of 50 MaSTPP and 50 MaDTPP instances each for different numbers of agents ($m \in \{2, 3, \cdots, 10\}$). For each instance, first, as was done by Khatib et al. (2007), a random solution is generated, to make sure that the problem is consistent. Furthermore, we construct our problems in such a way that the initial solution is also an optimal one, to be able to compare our results to an optimal solution. Based on the initial solution, constraints and preferences are added in such a way that the initial solution remains optimal. The construction for each MaSTPP or MaDTPP instance is as follows:

- For each agent, 20 random time points are selected in $[0, 600]$, representing the start and end times of 10 subsequent tasks belonging to that agent. This forms the initial solution of the problem.

- For each time point variable v_i, a constraint with respect to the reference time point is constructed. With a chance of 0.7, this is given by $0 \leq v_i - z \leq 600$. With a chance of 0.3, the lower and upper bounds on $v_i - z$ are given by the initial solution, subtracted and added with random integers in $[0, 200]$, but set to 0 or 600 if the bounds exceed these values.

- For each start time point v_i and its corresponding end time point v_j, a constraint $l \leq v_j - v_i \leq u$ is given, where l and u equal the initial solution value, subtracted or added with random integers in $[0, 100]$. Again, values below 0 or above 600 are set to 0 and 600, respectively.

  For MaDTPPs, with chance 0.6, an additional disjunct $\hat{l} \leq v_j - v_i \leq \hat{u}$ is added, where with chance 0.5 (or when l is less than 5), $\hat{l}$ is a random value in $[1, 20]$ added to u, and $\hat{u}$ is a random value in [1,20] added to $\hat{l}$, and otherwise, $\hat{u}$ is a random value in $[1, l)$ and $\hat{l}$ is a random value in $[0, \hat{u})$.

- For each pair of succeeding tasks (based on the initial solution), a constraint is added for the MaSTPP that imposes a strict order of the tasks, i.e. $0 \leq v_j - v_i \leq \infty$ is added where v_i is the end time variable of one task and v_j is the start time variable of the next task. For

MaDTPPs, each task will be also restricted to take place after the previous group of tasks with chance 0.7, but with chance 0.3, the the task will be added to the previous group of tasks that can be done in any order. For example, if the task with start time v_S^1 and end time v_E^1 and the task with start time v_S^2 and end time v_E^2 are already separated by a constraint $0 \leq v_S^2 - v_E^1 \leq \infty$, then with chance 0.3, the third task with start time v_S^3 and end time v_E^3 will be allowed to take place either before or after the second task, resulting in constraints $0 \leq v_S^3 - v_E^1 \leq \infty$ and $0 \leq v_S^2 - v_E^3 \leq \infty \vee 0 \leq v_S^3 - v_E^2 \leq \infty$.

- We make sure that some tasks of the agents are synchronized. This is important for getting already fixed times in a decoupling (on which principle some of our negotiation approaches are based), but it is also a realistic assumption in real-world activities such as meetings. In our simulation, we require that the difference between the start times and the difference between the end times of two tasks are equal to the differences of the start and end times of the initial solution for these tasks. Such constraints are however not structurally different from constraints in which the start times and end times have a distance 0 to each other.

  For each pair of agents, 0, 1 or 2 times a random task is selected for each of the two agents. Start times v_S^i and v_S^j, and end times v_E^i and v_E^j of these tasks are constrained by $\sigma(v_S^j) - \sigma(v_S^i) \leq v_S^j - v_S^i \leq \sigma(v_S^j) - \sigma(v_S^i)$ and $\sigma(v_E^j) - \sigma(v_E^i) \leq v_E^j - v_E^i \leq \sigma(v_E^j) - \sigma(v_E^i)$, i.e. by $\sigma(v_S^j) - \sigma(v_S^i) = v_S^j - v_S^i$ and $\sigma(v_E^j) - \sigma(v_E^i) = v_E^j - v_E^i$, where σ represents the initial solution.

- Some random inter-agent constraints are added. For a total of $0.2 \cdot m^2$ times, two random time point variables v_i and v_j of two different random agents are selected, and a constraint $l \leq v_j - v_i \leq u$ is added, where l is with chance 0.3 set to the difference between v_j and v_i in the initial solution, and with chance 0.7 to this value minus a random integer in $[0, 100]$. A dual construction is applied for u. Values below 0 or above 600 are set to 0 and 600, respectively.

- Some random constraints on two temporal variables for the same agent are added. For a total of $10 \cdot m$ times, two variables from a random agent are selected, and a constraint is constructed in the same manner as for the random inter-agent constraints.

Preferences are added to an instance as follows:

- Preferences at time point level are added both for MaSTPPs and MaDTPPs. With chance 0.1, a preference function is added to each of the constraints described above (except for the constraints that leave only one value possible for a difference of two temporal variables).[1] The possible range of values $[l, u]$ for a constraint $l \leq v_j - v_i \leq u$ is divided into intervals with random length (maximal 20), such that the value $\sigma(v_j) - \sigma(v_i)$ is at an endpoint of one of the intervals. With chance 0.5, each value in an interval is assigned the same random preference value from $[0, 20]$. Otherwise, a random linear preference function with coefficient either -1 or 1 is applied to the interval, with the restriction that no value in the interval is assigned a preference value outside the range $[0, 20]$. For the interval containing the value $\sigma(v_j) - \sigma(v_i)$, the same procedure is applied, but with the restriction that the value $\sigma(v_j) - \sigma(v_i)$ gets assigned to it the maximal preference value of 20.

- Preferences at disjunct level are only added to MaDTPPs. For each disjunctive constraint (only the constraints describing the length of a task, and the constraints describing the order of some tasks can be disjunctive constraints), with chance 0.9 a preference function is added for the corresponding agent. Such a preference function assigns a random value $p \in [0, 20]$ to the disjunct that corresponds to the initial solution, and a random value $q \in [0, p]$ to the other disjunct, such that the initial solution will always be the optimal solution for each agent.

[1] If a constraint is defined over variables of multiple agents, for each of them with chance 0.1 a preference function is added.

7.2 Experiments

Our algorithms were implemented in the Go programming language.[2] This language was chosen for its ability to simulate parallel processes in a natural way. For obtaining a decoupling, the agents execute the same algorithm in parallel, but information between the different parallel processes can simply be sent by the message feature of Go. The OMT solver used in our implementation is OptiMathSAT (Sebastiani & Trentin, 2015).

The parameter for ψ in the computation of the expected utility (see Equation 5.1) was set to 1.3 for all agents for all experiments, such that the behaviour of the agents is rather conceding. The maximum number of negotiation rounds was set to 100 for all experiments, except for the pre-decoupling negotiation approach for MaDTPPs. For these experiments, the number of negotiation rounds is 50, since more rounds are in general not necessary. However, if more negotiation rounds are required, they will be automatically used (see Section 6.2.2).

7.2.1 General performance

For the MaSTPP, we compare the performance of the post-decoupling algorithm (Section 5.1) with the performance of the pre-decoupling algorithm (Section 5.2). Furthermore, we compare them to the original MaTDR algorithm (Boerkoel & Durfee, 2013b) (which was not developed to handle preferences), to be able to check for improvements with respect to current approaches.

For each $m \in \{2, 3, \cdots, 10\}$ (recall that m denotes the number of agents in the MaSTPP), we ran 50 experiments. An experiment consists of decoupling the MaSTPP with the original MaTDR algorithm, applying the post-decoupling negotiation on the result of the MaTDR algorithm, and independently applying the decoupling algorithm with pre-decoupling negotiation. For each agent a, this results in three forecast preference values: one for the original algorithm (p^a_{orig}), one for the post-decoupling negotiation result (p^a_{post}) and one for the pre-decoupling negotiation algorithm (p^a_{pre}). Note that these forecast preference values do not correspond to the exact preference values $p^a(\sigma)$ of Definition 3.2: The latter is only defined for complete schedules, while the former only uses the preference values that are available due to the fixed time points in the decoupling.

Since our benchmark problem set was constructed in such a way that there exists for each problem a solution that is optimal for all agents (with preference value p^a_{opt} for agent a), we can relate the forecast preference values to this global optimum. Hence, we define the social welfare utilities of the three different approaches for one experiment as follows:

$$U_{\text{orig}} = \frac{\sum_{a \in A} p^a_{\text{orig}}}{\sum_{a \in A} p^a_{\text{opt}}}$$
$$U_{\text{post}} = \frac{\sum_{a \in A} p^a_{\text{post}}}{\sum_{a \in A} p^a_{\text{opt}}}$$
$$U_{\text{pre}} = \frac{\sum_{a \in A} p^a_{\text{pre}}}{\sum_{a \in A} p^a_{\text{opt}}}$$

These utilities are, for each number of agents, averaged over the 50 experiments, to obtain an average utility for each number of agents for each method.

For the MaDTPP, the experiments were comparable: The post-decoupling algorithm (Section 6.1) was compared with the pre-decoupling algorithm (Section 6.2), and both were compared to the original MaDTP-TD algorithm (Boerkoel & Durfee, 2013a). Parameters of the experiments and the definition of the average utilities were the same as in the MaSTPP case, except that the preference values for the MaDTPPs are not forecast values, but are equal to the exact preference values at disjunct level $p^a_d(\sigma)$ as defined in Definition 4.5.

7.2.2 Allowing non-Pareto improvements

To be able to examine the influence of also allowing non-Pareto improvements in the negotiation, we adapt the algorithms of Chapters 5 and 6 by changing the acceptance conditions of the agents.

[2]https://golang.org/

For the MaSTPP, this change is possible in both the post-decoupling and the pre-decoupling negotiation algorithms, but for the MaDTPP, only the post-decoupling approach can be used, since the pre-decoupling negotiation does not deal with acceptance criteria (see Section 6.2.2). For the three algorithms, the acceptance criteria are determined as follows:

In the original approaches, an agent accepts a proposal (provided that it results in a consistent schedule) if the preference value for the agent for that proposal is larger than or equal to a certain threshold value, which is in general based on the expected utility of the agent (see Sections 5.1.1, 5.2.1, and 6.2.2). For non-Pareto improvements, this threshold value is decreased with $g/(m-1)$, where g is the gain of the proposal for the proposing agent, and m denotes the number of agents in the temporal problem. By this, the gain for the proposing agent does not exceed the maximal loss for the other agents, and the maximal individual losses for the accepting agents are equal.

Note however that we cede some privacy with this approach since proposing agents are required to communicate their gain for a proposal. Nevertheless, we chose to use this value in our models since otherwise, a criterion is hard to define. Of course, other criteria can be developed in real-world situations.

We repeated our experiments for the Pareto algorithms as described before for the non-Pareto algoritms, and will compare the results.

7.2.3 Comparing flexibility

With the described experiments, we compare our post- and pre-decoupling negotiation approaches with each other and with the optimal solution. We also relate them to solutions of the original algorithms of Boerkoel & Durfee (2013b,a), but a proper comparison is not possible, since we focus on another performance measure. However, since Boerkoel & Durfee (2013b) implement the midpoint heuristic for assigning shared time points (see Section 2.1.4) to improve the flexibility of the solutions, we can define preferences in such a way that they represent a flexibility metric, and then compare our solutions to the solutions of Boerkoel & Durfee (2013b).

For different possible definitions of the flexibility of an (Ma)STP, we stick to the notation of Wilson et al. (2014). A naive choice for the flexibility measure is to sum up the length of the intervals from which the temporal variables can be selected, based on an FPC graph to which a shortest path algorthm has been applied (see Section 2.1.1), i.e.:

$$\mathit{flex}_N(\mathcal{S}) = \sum_{v_i \in V} w_{zi} + w_{iz},$$

where w_{zi} and $-w_{iz}$ are the upper and lower tightest bounds on $v_i - z$. Generally, this metric overestimates the flexibility of an MaSTP, since it does not take into account dependencies between variables.

An improvement of the flexibility metric, based on Hunsberger (2002a), is one in which these dependencies are considered:

$$\mathit{flex}_H(\mathcal{S}) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} w_{ij} + w_{ji}.$$

Again, w_{ij} and $-w_{ji}$ are the upper and lower tightest bounds on the difference $v_j - v_i$, based on a shortest path algorithm on the FPC network. However, Wilson et al. (2014) show also this metric to be not accurate enough, and propose a metric $\mathit{flex}(\mathcal{S})$ that is defined by a linear program and does not overestimate the actual flexibility since it is based on an interval schedule with a maximalized sum of interval lengths.

Now, we try to represent the flexibility of a problem by assigning preference values to it. However, it is a problem that flexibility, as described above, is defined at the level of constraints, while our MaSTPP preferences are defined at the level of a solution: Only for an assignment of values to the temporal variables, a preference value can be computed. It is possible to represent the flexibility by using an MaDTPP and preference functions at disjunct level (i.e. functions in the set D of Definition 4.4), but only at the cost of many disjuncts: In that case, one disjunct for each possible combination of bounds on a temporal difference is needed. This will make the problem representation too complex.

We therefore only consider the running example from Boerkoel & Durfee (2013b), also examined by Wilson et al. (2014), and define preferences for it in a rational way, such that they approximate the naive flexibility metric. Then, we compare the solutions of our algorithm on this problem with the solutions of Boerkoel & Durfee (2013b).

The MaSTP example of Boerkoel & Durfee (2013b) is given in Figure 7.1. Maximizing the naive flexibility is equivalent to maximizing the sum of the intervals of the decoupling. For agent B, the task R^B will be fixed by a decoupling, since this task needs to be synchronized with agent A and lasts exactly 60 minutes. Hence, the flexibility for B will only be determined by W_S^B and W_E^B. The earlier R^B takes place, the larger the flexibility for W^B will be. We therefore define the following preference function for agent B:

$$f_1^B(x) = \begin{cases} 120 - x & \text{if } 0 \le x \le 120 \\ 0 & \text{otherwise} \end{cases}$$

for $c_1 : 0 \le R_S^B - z \le 240$. Here, the value 120 is chosen since this is the latest possible time for R_S^B to obtain a consistent schedule, hence letting no flexibilty at all for W^B.

For agent A, a similar construction holds: the earlier R^A, the more flexibility there is for T^A. Since T^A lasts at least 90 minutes, R_S^A can take place not later than at 9:30, and hence we assign to constraint c_2 defined by $0 \le R_S^A - z \le 240$ the preference function:

$$f_2^A(x) = \begin{cases} 90 - x & \text{if } 0 \le x \le 90 \\ 0 & \text{otherwise} \end{cases}$$

For agent C, the task L^C is fixed, so flexibility can be obtained by maximizing the intervals for T_S^C and T_E^C. This can be obtained by allowing T_E^C to be as late as possible, resulting in the following preference function:

$$f_3^C(x) = \begin{cases} x - 90 & \text{if } 90 \le x \le 120 \\ 0 & \text{otherwise} \end{cases}$$

for c_3 representing the constraint $0 \le T_E^C - z \le 240$.

Our solution for the MaSTPP with these preferences, approximating naive flexibility, will be compared with the MaSTP solution that Boerkoel & Durfee (2013b) report.

7.3 Results

The experiments were conducted as described in the previous section. The results for the MaSTPP experiments are given in Figure 7.2. For the MaDTPP experiments, the results are given in Figure 7.3.[3]

Note that for the MaSTPP, the pre-decoupling negotiation approach outperforms the original approach by approximately 2 percentage point for nearly all the numbers of agents. For the post-decoupling negotiation approach, there is also a gain in utility with respect to the original approach, but this gain decreases when the number of agents in the MaSTPP grows.

For the MaDTPP, we have comparable results: The post-decoupling negotiation approach obtains a higher utility than the original approach for a low number of agents ($m < 5$). For more agents, the improvement decreases. The pre-decoupling approach, however, is rather unstable: In some cases, the improvement with respect to the original algorithm is small, but for other numbers of agents, this approach clearly outperforms the original approach.

[3]Note that the absolute difference in utility between the MaSTPP and the MaDTPP (around 0.2 for MaSTPPs vs. around 0.9 for MaDTPPs) is due to the fact that forecast preference values are used for the MaSTPP and actual preference values for the MaDTPP.

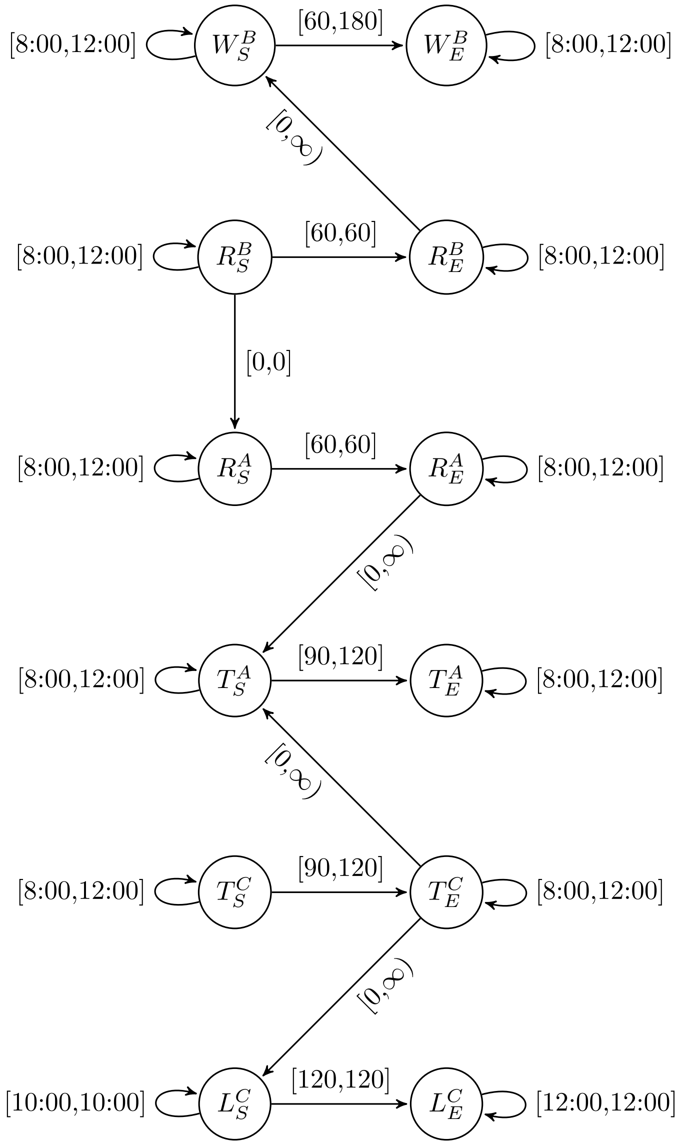


Figure 7.1: MaSTP for three agents (A, B, and C) with two tasks each, from which the start times of tasks R^A and R^B need to be synchronized.

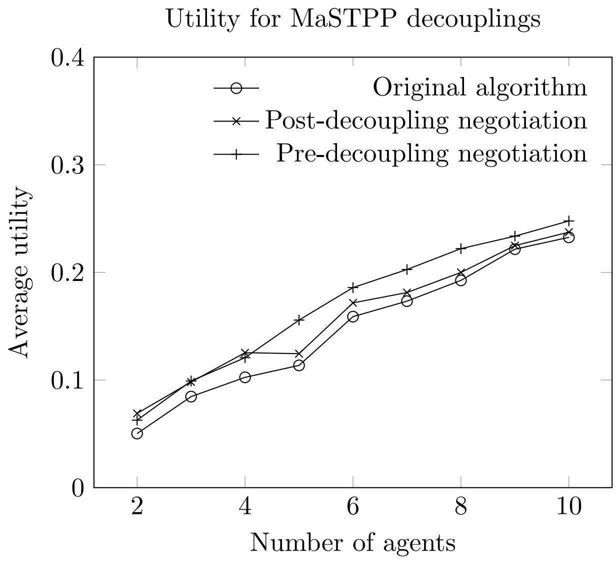


Figure 7.2: Average utilities obtained for the original algorithm, the post-decoupling negotiation and the pre-decoupling negotiation algorithm applied on the MaSTPP test set.

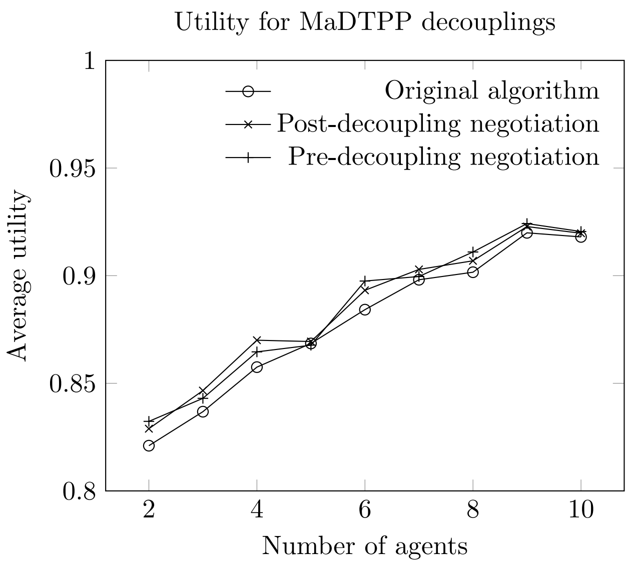


Figure 7.3: Average utilities obtained for the original algorithm, the post-decoupling negotiation and the pre-decoupling negotiation algorithm applied on the MaDTPP test set.

7.3.1 No profit from non-Pareto negotiations

The results of the experiments in which non-Pareto improvements were allowed, are shown in Figures 7.4 and 7.5, for the MaSTPP and the MaDTPP, respectively. Note that the results highly resemble the results of the Pareto algorithms: For the MaSTPP, both the post-decoupling and the pre-decoupling negotiation graphs show the same behaviour (cf. Figures 7.2 and 7.4); for the MaDTPP, the utilties of the post-decoupling algorithm decrease in both cases when the number of agents increases (cf. Figures 7.3 and 7.5).

Although the results of the non-Pareto algorithms resemble the results of the Pareto algorithms, we compare the two cases more carefully. For all results presented before, the utility of the original approach was subtracted from the utilities of the post- and pre-decoupling approaches to obtain the improvements of the negotiation approaches. Subsequently, these improvements for the Pareto algorithms were subtracted from the corresponding improvements for the non-Pareto algorithms. In Figures 7.6 and 7.7, the resulting difference in improvement between non-Pareto and Pareto negotiations is shown. Clearly, the difference between non-Pareto and Pareto algorithms is negligible. We conclude that for the present algorithms and problem set, allowing non-Pareto improvements is not advantageous compared to the original Pareto improvements.

7.3.2 Larger flexibility

The post-decoupling negotiation approach for the MaSTPP (see Section 5.1) applied to the modified example of Boerkoel & Durfee (2013b) as presented in Section 7.2.3 operates as follows: First, an initial decoupling is established in which the decoupling constraints for agent A are $30 \leq R_S^A - z \leq 30$ and $120 \leq T_S^A - z$, the decoupling constraints for agent B are $30 \leq R_S^B - z \leq 30$ and no decoupling constraints for agent C are established. In the first negotiation round, agent B, expecting a gain of 30, proposes to change his bounds $30 \leq R_S^B - z \leq 30$ into $0 \leq R_S^B - z \leq 0$. Agent A is involved in this change due to the constraint between R_S^A and R_S^B and accepts the change since it also gets a gain of 30. No other proposals are made by the agents, since their preference values cannot be further increased. The bounds for the nodes with respect to the reference time point for the final FPC graph are given in Table 7.1, and the naive flexibility is equal to 300.

For the pre-decoupling negotiation approach (see Section 5.2), we have shared elimination order $\langle T_E^C, R_S^B, R_S^A, T_S^A \rangle$. First the T_S^A variable is discussed by agents A and C, resulting in an assignment of value 120. Next, agent A selects R_S^A to negotiate on, and proposes to agent B a value of 0. For agent B, this value results in maximal gain, and hence agent B accepts the proposal. By this, the value for R_S^B is also fixed. Finally, agent C has to select a value for T_E^C, and since 120 is maximal and still possible, this value will be selected. Hence, the decoupling constraints for agent A are $R_S^A - z \leq 0$ and $120 \leq T_S^A - z$, the decoupling constraint for agent B is $R_S^B - z \leq 0$, and there are no extra decoupling constraints for agent C. The upper and lower bounds on the temporal variables of the corresponding FPC solution are given in Table 7.1, and the naive flexibility of this solution is equal to 360.

When comparing these solutions with the decoupling solution of Boerkoel & Durfee (2013b, p. 132) (also given in Table 7.1), we conclude that our negotiation approaches result in a higher flexibility (300 and 360 vs. 270).

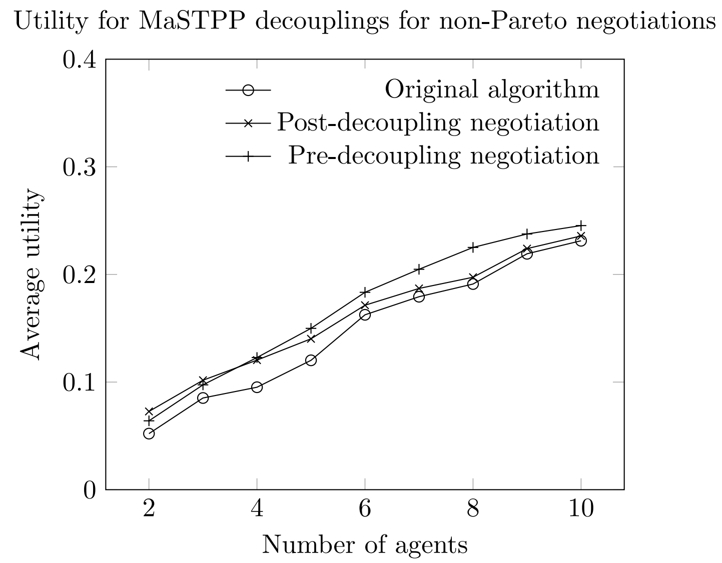


Figure 7.4: Average utilities obtained for the original algorithm, the post-decoupling non-Pareto negotiation algorithm and the pre-decoupling non-Pareto negotiation algorithm applied on the MaSTPP test set.

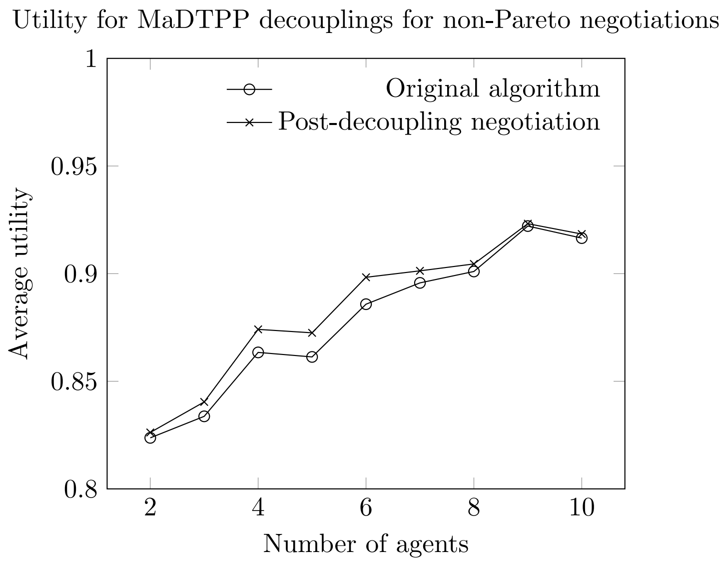


Figure 7.5: Average utilities obtained for the original algorithm and the post-decoupling non-Pareto negotiation algorithmn applied on the MaDTPP test set.

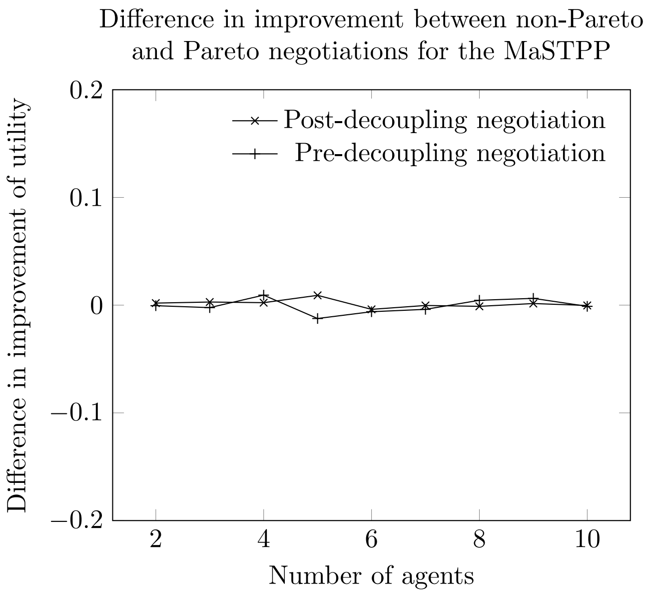


Figure 7.6: Difference in improvement of utility between non-Pareto and Pareto negotiations (both for the post-decoupling negotiation and the pre-decoupling negotiation algorithms) applied on the MaSTPP test set.

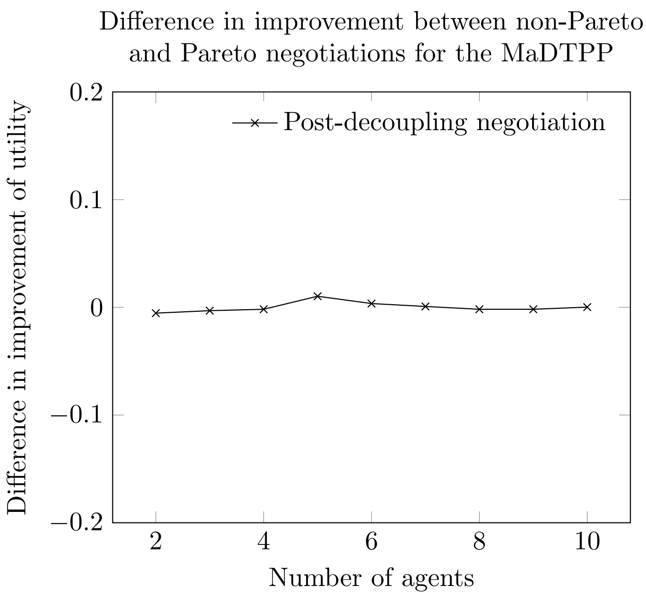


Figure 7.7: Difference in improvement of utility between non-Pareto and Pareto negotiations (for the post-decoupling negotiation algorithms) applied on the MaDTPP test set.

	Original decoupling			Post-decoupling negotiation			Pre-decoupling negotiation		
	lower	upper	interval	lower	upper	interval	lower	upper	interval
W_S^B	105	180	75	90	180	90	60	180	120
W_E^B	165	240	75	150	240	90	120	240	120
R_S^B	45	45	0	0	0	0	0	0	0
R_E^B	105	105	0	60	60	0	60	60	0
R_S^A	45	45	0	0	0	0	0	0	0
R_E^A	105	105	0	60	60	0	60	60	0
T_S^A	120	150	30	120	150	30	120	150	30
T_E^A	210	240	30	210	240	30	210	240	30
T_S^C	0	30	30	0	30	30	0	30	30
T_E^C	90	120	30	90	120	30	90	120	30
L_S^C	120	120	0	120	120	0	120	120	0
L_E^C	240	240	0	240	240	0	240	240	0
$flex_N$			270			300			360

Table 7.1: Lower and upper bounds for the temporal variables of the MaSTP example of Boerkoel & Durfee (2013b) (Figure 7.1) with respect to the reference time point, after decoupling and applying the FPC algorithm, for the original approach of Boerkoel & Durfee (2013b) and for our two negotiation approaches. Lengths of the intervals are given in the interval columns and $flex_N$ equals the sum of these interval lengts.

Chapter 8

Discussion

The goal of the research described in this thesis was to be able to deal with preferences in multi-agent scheduling problems, both from a theoretical perspective (the definition of an appropriate framework) and from a practical perspective (the development of algorithms for solving these problems). We consider again the research questions that we formulated in Section 2.3, and examine the answers that we developed in this thesis.

RQ1 [**MaSTPP definition**] How can a framework for the Multi-agent Simple Temporal Problem with Preferences be defined?

In Chapter 3, we developed a framework for dealing with preferences defined on the difference of two temporal variables. The approach was based on existing preference approaches for temporal problems, but was extended to the multi-agent case.

RQ2 [**MaDTPP definition**] How can a framework for the Multi-agent Disjunctive Temporal Problem with Preferences be defined?

For the disjunctive case, we developed a framework in which preferences at two different levels can be combined. In Chapter 4, we described how we integrated two different approaches, both with their advantages, into one framework. In our framework, agents are able to define their general preferences at disjunct level, and make some refinements by defining preferences at time point level. Furthermore, an important aspect is the possibility of constraints in disjunctive normal form, to handle dependencies between constraints.

RQ3 [**MaSTPP post-decoupling negotiation approach**] Given a decoupling solution to a Multi-agent Simple Temporal Problem with Preferences, what automated negotiation protocol can be used as a post-processing step to make a Pareto improvement on the decoupling with respect to the expected preferences of the agents?

In Section 5.1, we developed an alternating offers protocol for making Pareto improvements on a decoupling. In contrast to general protocols, our approach allows further negotiation when a proposal was accepted, since several local improvements were allowed. From Section 7.3 (Figure 7.2), we conclude that the approach indeed makes improvements on a given decoupling, especially for a low number of agents.

RQ4 [**MaSTPP pre-decoupling negotiation approach**] Given the Multi-agent Temporal Decoupling algorithm with Relaxation of Boerkoel & Durfee (2013b) to solve a Multi-agent Simple Temporal Problem with Preferences, what automated negotiation approach can be incorporated into the algorithm to improve the resulting decoupling with respect to the expected preferences of the agents?

Our main insight with respect to the decoupling algorithm of Boerkoel & Durfee (2013b) is that an improvement can be obtained at the assignment of shared variables. Instead of using a midpoint heuristic, we proposed an alternating offers protocol by which the agents together can establish a value for each shared variable. In Section 7.3 (Figure 7.2), we have shown that this approach improves the original approach for all numbers of agents.

RQ5 [**MaDTPP post-decoupling negotiation approach**] Given a decoupling solution to a Multi-agent Disjunctive Temporal Problem with Preferences, what automated negotiation protocol can be used as a post-processing step to make a Pareto improvement on the decoupling with respect to the preferences of the agents?

For the Multi-agent Disjunctive Temporal Problem with Preferences, we developed a negotiation protocol comparable to that of the Multi-agent Simple Temporal Problem with Preferences. It turned out, however, that checking for local consistency for a proposal was much harder for the MaDTPP. Hence, an existing global approach was used instead of a local approach. In Section 7.3 (Figure 7.3), we have shown that our method improves on average the decouplings of an original algorithm.

RQ6 [**MaDTPP pre-decoupling negotiation approach**] Given the MaDTP Temporal Decoupling algorithm of Boerkoel & Durfee (2013a) to solve a Multi-agent Disjunctive Temporal Problem, how can automated negotiation be incorporated into the algorithm to improve the resulting solutions with respect to the preferences of the agents?

The decoupling algorithm of Boerkoel & Durfee (2013a) makes use of a coordinator-based approach, in which the different agents send the influence spaces of their local solutions to a central coordinator. We combined this approach with a coordinator-based negotiation approach. Our main improvement was to let the submissions of the agents be based on preference values and on time: The most preferred solutions are submitted first. From Section 7.3 (Figure 7.3), it is clear that our method improves the original algorithm.

RQ7 [**Non-Pareto improvements**] What are the advantages for social welfare when non-Pareto improvements are allowed in the negotiation approaches, compared with negotiation approaches that induce only Pareto improvements?

The comparison of the negotiation approaches in which non-Pareto improvements were allowed with the original negotiation approaches (Section 7.3.1) shows, in constrast to our hypothesis, that there is no improvement in social welfare when non-Pareto improvements are allowed. An explanation for this lack of improvement might be that the problem test set is overconstrained, such that non-Pareto improvements are actually not possible in this test set.

In general, this hypothesis of a too specific, probably overconstrained test set is endorsed by the fact that the average utility for MaDTPP solutions of the original algorithm is around 87 per cent (see Figure 7.3), instead of the expected 50 per cent. Moreover, an explanation can be that for our test set, always a solution exists that is optimal for all agents. Hence, non-Pareto improvements are not required to obtain this specific solution, since the same result can be obtained with Pareto improvements.

RQ8 [**Flexibility**] In what way can preferences in scheduling problems be used to represent flexibility, and how do preference approaches relate to flexibility approaches?

In Section 7.2.3, we argued that flexibility cannot be represented directly in our preference frameworks. However, we used some knowledge of a specific problem to approximate the flexibility by preference values. From the analysis in Section 7.3.2, we can conclude that a larger flexibility was obtained by our negotiation approaches than by the original decoupling approach. However, it is important to keep in mind that the solution that Boerkoel & Durfee (2013b) report is one of several possibilities: Their algorithm is not deterministic. Although our method is also not deterministic, we have improved the given decoupling. Furthermore, our pre-decoupling negotiation solution turns out to be optimal with respect to the flexibility approach of Wilson et al. (2014).

8.1 Algorithms

Our algorithms have different computational complexity. The post-decoupling negotiation approaches are rather complex: For the MaSTPP, the algorithm for obtaining local consistency (Algorithm 1) has two nested loops (starting in lines 14 and 15) that are both bounded by the number of nodes in the problem, but there is a recursive call of the algorithm in the inner loop (line 50). The post-decoupling negotiation algorithm for the MaDTPP makes use of the computationally complex MaDTP-TD algorithm in each round.

For the pre-decoupling negotiation methods, we are able to relate our approaches to the algorithms of Boerkoel & Durfee (2013b,a). For the pre-decoupling negotiation for MaSTPPs, the algorithm of Boerkoel & Durfee (2013b) is added with a fixed maximum number of negotiation rounds for each assignment to a shared variable. The computation of maximal preference values in the negotiation process depends on the preference representation, but is bounded by the product of the number of nodes connected to the node that is discussed, and the number of different assignable values, which can be limited when only integer values are allowed.

The pre-decoupling negotiation for MaDTPPs makes use of the OMT solver instead of a general SMT solver or another solution method for DTPs, as was done by Boerkoel & Durfee (2013a). Hence, the increase in complexity depends on the complexity of the OMT solver used, relative to the complexity of the original DTP solution method. Furthermore, in our negotiation approach, we work with rounds, hence restricting the advantages of speed and individuality of the original MaDTP-TD algorithm. However, in general, our pre-decoupling algorithms are more or less comparable to the original algorithms of Boerkoel & Durfee (2013b,a).

Different improvements on our algorithms are possible. In the post-decoupling negotiation algorithm for MaSTPPs, proposals are based only on preference values on variables related to the reference time point. It would be interesting to take also the preferences on distances of two random variables into account. Furthermore, both in the post- and pre-decoupling negotiation algorithms for MaSTPPs, we only consider preferences on time points that are fixed. However, in general, only interface time points will be fixed for a decoupling, although preferences can also be defined on private time points. It would be an interesting possibility to compute minimum and maximum preference values based on the intervals of possible values for all variables, and let these expected values have a weight in the negotiation. Also, a multi-issue negotiation is possibly an interesting option: In the current algorithm, we negotiate each time on one variable separately, but due to the depenencies between variables, a multi-issue negotiation would possibly give better results.

Another interesting point with respect to the post-decoupling negotiation approach for the MaSTPP is broadening intervals. With our current consistency checking algorithm (Algorithm 1), only changes that are necessary will be made. However, it can be advantageous to make also other changes. Consider the application of our post-decoupling negotiation approach to the example of Boerkoel & Durfee (2013b) as described in Section 7.3.2. When the value of R_S^B changes from 30 into 0, R_E^B will be fixed at 60 and hence, the lower bounds on W_S^B and W_E^B can be changed into 60 and 120, respectively. With these changes, a larger flexibility (360, as in the pre-decoupling negotiation case) would have been obtained. However, since these changes were not necessary, they have not been performed by the algorithm.

In the post-decoupling negotiation for MaDTPPs, agents that receive a proposal from another agent are not allowed to change other disjuncts to make the proposal possible: The bounds on their variables that are shared with an agent other than the proposing agent remain equal to the original decoupling bounds. However, it is imaginable that by also changing some of these bounds, the proposal becomes feasible for the agent. Hence, considering a propogation of changes to other agents will be interesting.

8.2 Implications

One of the main contributions of this research is that we added to the literature a new framework to be able to deal with preferences in multi-agent temporal problems. To our knowledge, such a framework did not exist before. However, since the kind of problems we are able to model in our framework is quite natural in real world situations, our contribution can be an important one.

Our second main contribution is that we have shown that the application of automated negotiation approaches for solving the kind of problems we defined is beneficial. For the algorithms that we have developed, we have shown an increase in performance on our problem test set. Although we did not investigate the performance of our algorithms on real world data, the results are promising and we expect that our methods could be applied to e.g. gate assignment at airports, where different airways have competing preferences. For applications to resource-constrained problems, either our methods could be extended, or the problems could be transformed into temporal constraint problems by precedence constraint posting (Wilson, 2016). Furthermore, as a practical contribution, we have shown that our methods can be used to obtain improvements in the flexility of decoupling solutions to temporal problems.

8.3 Future work

Different new research directions are possible based on our work. First, it is important to note that we defined a new problem and proposed rather specific methods to solve this problem: We motivated why we used automated negotiation, and we based our approaches on the notion of a decoupling since in this way, they could be related to and compared with earlier approaches. However, it will be interesting to think about other possible solving methods. For example, a negotiation could also take place without the intermediate result of a decoupling.

Second, there are different improvements possible on our approach. The algorithms could be adjusted as described in Section 8.1. Furthermore, it is interesting to try to apply theory of mind in subtle ways. Probably, in some cases, agents are willing to share (part of) their preferences with some other agents, to obtain a higher utility at the cost of some privacy. Another option is to study the behaviour of agents with different power in the negotiation, e.g. by allowing different values of ψ in the expected utility functions (5.1).

A third aspect is the choice of preference values. We defined a rather powerful framework for describing preferences, but, as Zhang et al. (2011, p. 95) remark, it is not trivial for agents to define their preference functions. We encountered this in defining preferences representing flexibility (see Section 7.2.3): Agents needed some insight in the consequences of constraint propagation in the STPs to be able to choose their preference functions. It would be interesting to develop methods for more intuitive or automatical preference assignment.

Further, it would be interesting to make our frameworks and methods applicable for harder scheduling problems, for example for resource-constrained problems. In general, it would be important to refine the methods based on specific real world problems. Different approaches (fully distributed approaches versus approaches with a coordinator, and different negotiation protocols) are possible. Dependent on the application, an appropriate approach can be developed.

Bibliography

Adhau, S., Mittal, M. L., & Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, *25*, 1738–1751.

Agnetis, A., Billaut, J.-C., Gawiejnowicz, S., Pacciarelli, D., & Soukhal, A. (2014). *Multiagent Scheduling: Models and Algorithms*. Berlin, Heidelberg: Springer.

Baarslag, T. (2016). *Exploring the Strategy Space of Negotiating Agents: A Framework for Bidding, Learning and Accepting in Automated Negotiation*. Springer Theses. Cham: Springer.

Barrett, C., Sebastiani, R., Seshia, S., & Tinelli, C. (2009). Satisfiability modulo theories. In A. Biere, M. J. H. Heule, H. van Maaren, & T. Walsh (Eds.), *Handbook of Satisfiability* chapter 26, (pp. 825–885). Amsterdam: IOS Press.

Barrett, C. & Tinelli, C. (2017). Satisfiability modulo theories. In E. M. Clarke, T. A. Henzinger, & H. Veith (Eds.), *Handbook of Model Checking*. Cham: Springer. In press.

Bjørner, N., Phan, A.-D., & Fleckenstein, L. (2015). νZ - An optimizing SMT solver. In C. Baier & C. Tinelli (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems*, volume 9035 of *LNCS* (pp. 194–199). Berlin, Heidelberg: Springer.

Boerkoel, J. C. (2012). *Distributed Approaches for Solving Constraint-based Multiagent Scheduling Problems*. PhD thesis, University of Michigan.

Boerkoel, J. C. & Durfee, E. H. (2010). A comparison of algorithms for solving the multiagent simple temporal problem. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS 2010)*, (pp. 26–33).

Boerkoel, J. C. & Durfee, E. H. (2012). A distributed approach to summarizing spaces of multiagent schedules. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, (pp. 1742–1748).

Boerkoel, J. C. & Durfee, E. H. (2013a). Decoupling the multiagent disjunctive temporal problem. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13)*, (pp. 123–129).

Boerkoel, J. C. & Durfee, E. H. (2013b). Distributed reasoning for multiagent simple temporal problems. *Journal of Artificial Intelligence Research*, *47*, 95–156.

Boerkoel, J. C., Planken, L. R., Wilcox, R. J., & Shah, J. A. (2013). Distributed algorithms for incrementally maintaining multiagent simple temporal networks. In *Proceedings of the Twenty-Third International Conference on Automated Planning and Scheduling (ICAPS 2013)*, (pp. 11–19).

Cimatti, A., Griggio, A., Schaafsma, B., & Sebastiani, R. (2013). The MathSAT5 SMT Solver. In N. Piterman & S. Smolka (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *LNCS* (pp. 93–107). Berlin, Heidelberg: Springer.

Cohen, J., Cordeiro, D., & Trystram, D. (2015). Coordination mechanisms for decentralized parallel systems. *Concurrency and Computation: Practice and Experience*, *27*, 1255–1272.

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence*, *49*, 61–95.

Dimopoulos, Y. & Moraitis, P. (2014). Advances in argumentation-based negotiation. In F. Lopes & H. Coelho (Eds.), *Negotiation and Argumentation in Multi-agent Systems: Fundamentals, Theories, Systems and Applications* (pp. 82–125). Sharjah: Bentham.

Duan, L., Doğru, M. K., Özen, U., & Beck, J. C. (2012). A negotiation framework for linked combinatorial optimization problems. *Autonomous Agents and Multi-Agent Systems*, *25*, 158–182.

Dutertre, B. (2014). Yices 2.2. In A. Biere & R. Bloem (Eds.), *Computer-Aided Verification*, volume 8559 of *LNCS* (pp. 737–744). Cham: Springer.

Ehtamo, H. & Hämäläinen, R. P. (2001). Interactive multiple-criteria methods for reaching Pareto optimal agreements in negotiations. *Group Decision and Negotiation*, *10*, 475–491.

Fatima, S., Kraus, S., & Wooldridge, M. (2015). *Principles of Automated Negotiation*. Cambridge: Cambridge University Press.

Fatima, S., Wooldridge, M., & Jennings, N. R. (2009). An analysis of feasible solutions for multi-issue negotiation involving nonlinear utility functions. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, (pp. 1041–1048).

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2006). Multi-issue negotiation with deadlines. *Journal of Artificial Intelligence Research*, *27*, 381–417.

Floyd, R. W. (1962). Algorithm 97: Shortest path. *Communications of the ACM*, *5*(6), 345.

Fujita, K., Ito, T., & Klein, M. (2014). Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents. *Decision Support Systems*, *60*, 10–17.

Hunsberger, L. (2002a). Algorithms for a temporal decoupling problem in multi-agent planning. In *Proceedings of The Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, (pp. 468–475).

Hunsberger, L. (2002b). *Group Decision Making and Temporal Reasoning*. PhD thesis, Harvard University.

Ito, T., Hattori, H., & Klein, M. (2007). Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, (pp. 1347–1352).

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. J., & Sierra, C. (2001). Automated negotiation: Prospects, methods and challenges. *Group Decision and Negotiation*, *10*, 199–215.

Kattan, A., Ong, Y.-S., & Galván-López, E. (2013). Multi-agent multi-issue negotiations with incomplete information: A genetic algorithm based on discrete surrogate approach. In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2556–2563).

Khatib, L., Morris, P., Morris, R., Rossi, F., Sperduti, A., & Venable, K. B. (2007). Solving and learning a tractable class of soft temporal constraints: Theoretical and experimental results. *AI Communications*, *20*, 181–209.

Lang, F., Fink, A., & Brandt, T. (2016). Design of automated negotiation mechanisms for decentralized heterogeneous machine scheduling. *European Journal of Operational Research*, *248*, 192–203.

Lau, R. Y. K. (2005). Towards genetically optimised multi-agent multi-issue negotiations. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, (pp. 1–10).

Lin, R. J. & Chou, S. T. (2004). Mediating a bilateral multi-issue negotiation. *Electronic Commerce Research and Applications*, *3*, 126–138.

Maratea, M. & Pulina, L. (2014). Efficient DTPP solving with a reduction-based approach. *Intelligenza Artificiale*, *8*, 101–111.

Micalizio, R. & Torta, G. (2015). Scheduling with structured preferences. In M. Bramer & M. Petridis (Eds.), *Research and Development in Intelligent Systems XXXII* (pp. 155–168). Cham: Springer.

Moffitt, M. D. (2011). On the modelling and optimization of preferences in constraint-based temporal reasoning. *Artificial Intelligence*, *175*, 1390–1409.

De Moura, L. & Bjørner, N. (2008). Z3: An efficient SMT solver. In C. R. Ramakrishnan & J. Rehof (Eds.), *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *LNCS* (pp. 337–340). Berlin, Heidelberg: Springer.

Oddi, A. (2014). Constraint-based strategies for the disjunctive temporal problem: Some new results. In *Proceedings of the Sixth European Conference on Planning*, (pp. 159–165).

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

Peintner, B. M. (2005). *Algorithms for Constraint-based Temporal Reasoning with Preferences*. PhD thesis, University of Michigan.

Planken, L., De Weerdt, M., & Van der Krogt, R. (2008). P^3C: A new algorithm for the simple temporal problem. In *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008)*, (pp. 256–263).

Planken, L. R., De Weerdt, M. M., & Witteveen, C. (2010). Optimal temporal decoupling in multiagent systems. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, (pp. 789–796).

Ramacher, R. & Mönch, L. (2016). An automated negotiation approach to solve single machine scheduling problems with interfering job sets. *Computers & Industrial Engineering*, *99*, 318–329.

Ren, F. & Zhang, M. (2014). Bilateral single-issue negotiation model considering nonlinear utility and time constraint. *Decision Support Systems*, *60*, 29–38.

Schmidt, E. P. (1970). The truth about Boulwarism. *The Intercollegiate Review*, *7*, 63–65.

Sebastiani, R. & Trentin, P. (2015). OptiMathSAT: A tool for optimization modulo theories. In D. Kroening & S. C. Păsăreanu (Eds.), *Computer-Aided Verification*, volume 9206 of *LNCS* (pp. 447–454). Cham: Springer.

Stergiou, K. & Koubarakis, M. (2000). Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, *120*, 81–117.

Tsamardinos, I. & Pollack, M. E. (2003). Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence*, *151*, 43–89.

De Weerd, H., Verbrugge, R., & Verheij, B. (2016). Negotiating with other minds: The role of recursive theory of mind in negotiation with incomplete information. *Autonomous Agents and Multi-Agent Systems*. In press.

Wilson, M. (2016). *Robust Scheduling in an Uncertain Environment*. PhD thesis, Delft University of Technology.

Wilson, M., Klos, T., Witteveen, C., & Huisman, B. (2014). Flexibility and decoupling in simple temporal networks. *Artificial Intelligence*, *214*, 26–44.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. Chichester: John Wiley & Sons.

Zhang, L. & Liu, Q. (2016). An automated multi-issue negotiation mechanism based on intelligent agents in e-commerce. *Journal of Advanced Management Science*, *4*, 172–175.

Zhang, L., Song, H., Chen, X., & Hong, L. (2011). A simultaneous multi-issue negotiation through autonomous agents. *European Journal of Operational Research*, *210*, 95–105.

Zheng, R., Chakraborty, N., Dai, T., & Sycara, K. (2016). Automated multilateral negotiation on multiple issues with private information. *INFORMS Journal on Computing*, *28*, 612–628.