2009

Spatio-Temporal Context in Agent-Based Meeting Scheduling

Hijaz Al-Ani
University of Windsor

SPATIO-TEMPORAL CONTEXT IN AGENT-BASED MEETING SCHEDULING

By

Hijaz Al-Ani

A Thesis
Submitted to the Faculty of Graduate Studies
Through the School of Computer Science
In Partial Fulfillment of the Requirements for the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada
2009

Spatio-Temporal Context in Agent-Based Meeting Scheduling

By

Hijaz Al-Ani

APPROVED BY

Dr. A. Asfour
Department of Civil and Environmental Engineering

Dr. B. Boufama
School of Computer Science

Dr. A. Tawfik, Advisor
School of Computer Science

Dr. I. Ahmad, Chair of Defense
School of Computer Science

May 19, 2009

DECLERATION OF PREVIOUS PUBLICATION

This thesis includes one original paper that has been previously submitted for publication in peer reviewed conference proceedings as follows:

Thesis Chapter	Publication title/full citation	Publication status
Chapter 3 and 4	*Meeting Scheduling Assembles Children in the Rectangular Forest*	*submitted*

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Meeting scheduling is a common task for organizations of all sizes. It involves searching for a time and place when and where all the participants can meet. However, scheduling a meeting is generally difficult in that it attempts to satisfy the preferences of all participants. Meeting scheduling negotiation tends to be an iterative and time consuming task. Proxy agents can handle the negotiation on behalf of the individuals without sacrificing their privacy or overlooking their preferences.

This thesis examines the implications of formalizing meeting scheduling as a spatiotemporal negotiation problem. In particular, the "Children in the Rectangular Forest" (CRF) canonical model is applied to meeting scheduling. By formalizing meeting scheduling within the CRF model, a generalized problem emerges that establishes a clear relationship with other spatiotemporal distributed scheduling problems. The thesis also examines the implications of the proposed formalization to meeting scheduling negotiations. A protocol for meeting location selection is presented and evaluated using simulations.

DEDICATION

To my beloved parents,

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my family and loved ones for their support and assistance. My parents deserve special mention for their inseparable support and prayers. I am indebted to my father, Dr. Faiez Al-Ani for his continuous encouragement and guidance. He nurtured my gift for deeper thinking and encouraged me to aspire to greater things, showing me the joy of intellectual pursuit ever since I was a child.

Next, I would like to thank my advisor, Dr. Ahmed Tawfik, for his unending encouragement and enthusiasm. His constructive criticism along the way helped shape my thesis into its current form. He was always able to find a positive way to look at things or find another aspect to investigate. I greatly appreciate the many hours of discussions we have had over the past two years.

I am also grateful to my readers, Dr. Asfour and Dr. Boufama, for taking the time and effort to read and comment on this thesis. Their feedback was very useful in improving the quality of my thesis.

Zina Ibrahim is a Ph.D. student working under same supervision. She has been a great friend and colleague since my first days at University of Windsor. Her company at the lab has always been pleasant and encouraging. I greatly appreciate her valuable advices and long discussions in formulating things mathematically. I am proud to record that I had several opportunities to work with an exceptionally talented scientist like her.

Table of Content

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In almost all organizations, scheduling meetings is an important yet iterative and time-consuming task. As a result, a lot of research has been dedicated to finding efficient yet functionally competent ways to organize the task and deal with its iterative nature and reach good solutions for the constraints to be enforced on the meetings. Hence, the problem has been named the *Meeting Schedulin*g (MS) problem and involves searching for a time and place for which all of the meeting participants are free and available while keeping in mind global organization and local individual constraints and preferences with respect to the meeting time and location.

Meeting Scheduling is a naturally distributed task (Ephrati *et al.,* 1994; Garrido L and Sycara K, 1996; BenHassine *et al.,* 2004) that requires the availability of two or more persons. Meetings may be scheduled individually or within a series or group of meetings. Each potential attendee needs to take into account his/her own meeting preferences and calendar availability. Most of the time, each attendee has some uncertain and incomplete knowledge about the preferences and calendars of the other attendees. In fact, people usually try to keep their calendar and preference information private. During the meeting scheduling process, all attendees should consider the main group goal but they also take into account individual goals (i.e. to satisfy their individual preferences). Solving the MS problem involves finding a compromise between all attendees requirements, usually conflicting, for meeting (i.e. date, time and duration). Thus, this problem is subject to several constraints, essentially related to availability, timetabling and preferences of each user in terms of location or other preferences.

Hence, automating the MS problem is a challenging task as many constraints are to be considered. Essentially, two important types of constraints include the location of the meeting and its time. With respect to location, most prior research assumes that meetings are held in stationary locations which ignore the possibility of having a mobile meeting that may better suit participants' preferences. Moreover, the temporal aspects of the meeting scheduling problems enforce additional constraints to be considered along with the location constraints.

Current approaches to tackle the problem consider one constraint at a time. For instance, (Chithambaram, Miller, 2005; Santos and Vaughn, 2007) present approaches to search for an appropriate location for a meeting only considering spatial constraints while (Modi *et al.,* 2004; Crawford and Veloso, 2004) focus on searching for the most convenient time. Moreover, participants in the meeting may have to disclose information that they may consider private for the purpose of finding an appropriate situation or making the meeting scheduling problem more efficient.

In order to tackle the issues presented above, this thesis presents a framework for scheduling meetings that combines spatial and temporal constraints and coordinate meetings accordingly while keeping in mind issues that maybe of importance to the participants such as privacy, individual schedules, and personal preferences.

1.2 PROBLEM STATEMENT

Enable agents who are acting on behalf of meeting participants to negotiate an appropriate time and place for a stationary or a mobile meeting.

1.3 OBJECTIVES

1. Establish a formal model to represent the meeting scheduling problem.
2. Map the problem to a canonical agent-based model for spatio-temporal collaboration.

3. Develop an autonomous negotiation protocol based on the model developed for selecting a meeting place, keeping in mind temporal constraints.
4. Implement various scenarios for verifying and validating the developed negotiation protocol.

1.4 CONTRIBUTIONS

The model and negotiation protocols developed in this thesis supplement the literature in the following ways:

1. A new model that solves the meeting scheduling problem via autonomous agent-based negotiation while considering spatial and temporal meeting constraints.
2. Privacy-efficiency tradeoff: The negotiation protocol does not require agents to share their personal calendars with the other agents participating in the meeting therefore preserving their privacy. Moreover, if in certain scenarios the agents agree to share their calendars, then the meeting initiator can schedule the meeting without negotiation.
3. The spatio-temporal nature of the model proposed makes it easily extensible to scheduling mobile meetings, which is an extension that has not been approached by those who are interested in the meeting scheduling problem.
4. If a meeting is allowed to end in a different location, then a new class of meetings emerges, the mobile meeting. The meeting scheduling problem becomes a generalization problem that captures useful aspects of some of other problems like the car pooling(Burmeister and Haddadi, 1997) and flight crew scheduling(Castro and Oliveira, 2005).

1.5 THESIS STRUCTURE

The thesis is structured as follows. In chapter 2, we review previous research on meeting-scheduling problem and multi-agent negotiation model comparing different approaches. In chapter 3, we introduce a mapping of MS problem to an existing canonical model, the *Children in the Rectangular Forrest* (Luo and Boloni, 2007) (CRF), formulate distributed negotiation algorithms for the MS problem, and discuss the properties of the proposed protocol. In chapter 4, we present a simulation environment that we have developed and used to test and validate the negotiation protocol introduced in chapter 5. We conclude the thesis in chapter five by tying some loose ends and outlining future work.

CHAPTER 2

MEETING SCHEDULING: A LITERATURE REVIEW

This chapter reviews the main contributions pertaining to meeting scheduling and compares different approaches and methods focusing mainly on viewing meeting scheduling as a distributed task for multi-agent negotiation. The chapter introduces the main issues relating to MS, along with the most recent approaches to incorporating spatial constraints as well as temporal constraints.

The chapter is organized as follows. After an introduction to Agent and Multi-Agent systems in Section 2.1, we define the MS problem and its various facets in Section 2.2 as described in the literature. Section 2.3 presents past efforts for solving MS problem using mutli-agent systems by summarizing different approaches. Section 2.4 introduces canonical negotiation models for collaboration in time and space. Finally, the conclusion is presented in Section 2.5.

2.1 MULTI-AGENT SYSTEMS

Multi-Agent systems (MAS) are systems composed of multiple interacting elements, known as agents. (Jennings, 2000; Wooldridge, 2002) define multi-agent systems as the system that contains a number of agents, which interact with one another through communication. The agents are able to act in an environment; different agents have different 'spheres of influence', in the sense that they will have control over, or at least be able to influence, different parts of the environment.

Multi-agent systems are a relatively new sub field of Computer Science, they have only been studied since about 1980, and the field has only gained widespread recognition since about the mid 1990s. However, since then, international interest in the field has grown enormously. This rapid growth has been spurred at least in part by the belief that agents are an appropriate software paradigm through which to exploit the possibilities presented by massive open distributed systems.

Agent

An obvious way to start this survey would be by introducing a definition of the term agent. After all, this survey is based on multi-agent systems. Wooldridge mentioned in his book (Wooldridge, 2002) that there is no universally accepted definition of the term agent, but some sort of definition is important. An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. The definition presented here is adapted from (Wooldridge and Jennings, 1995)

2.2 MEETING-SCHEDULING PROBLEM

Meeting Scheduling is the process of determining a starting time and an ending time of an event in which several individuals will participate. Various requirements, constraints, of these participants must be taken into account in scheduling meetings (Tsuruta and Shintani, 2000).

The meeting-scheduling problem is a type of negotiation problem. A negotiation problem is associated with a set of fixed and variable attributes. The initiator of the meeting determines which attributes are fixed and which are variable. Those that are variable will be negotiated. For example, a person calling a meeting might tell his assistant: ‘‘I would like to hold a project meeting sometime next week, preferably next Wednesday afternoon, with Tom’’ In this example, the type of meeting, desired time period when the meeting is to be held, and the attendees are fixed attributes, while the day and time are variable attributes. The following are examples of meeting attributes taken from the literature (Chun *et al.,* 2003):

- **Initiator:** The host or initiator of the meeting. A person might consider a meeting called by his/her immediate supervisor to be more important than the others, for example.
- **Rank:** The rank or position of the person calling the meeting. Values can be any rank or position within the organization, such as: CEO, CFO, CTO, VP R&D, VP Sales, etc.
- **Attendees:** The participants or invitees, a list of individuals that may or need to attend the meeting. This list may be further classified according to priorities of the attendees, such as those that ‘‘must’’, ‘‘should’’ or ‘‘can’’ attend the meeting. All those that ‘‘must attend,’’ must be all available before the meeting can be confirmed, otherwise it will be cancelled. ‘‘Should attend’’ are the normal participants of the meeting. ‘‘Can attend’’ are casual observers; their availability will not affect the meeting schedule.
- **Type:** The type of meeting. For example, values might include: general, departmental, group, strategic, inter-departmental, technical, marketing, sales, project, interview, etc. This value can be used to determine the priority of the meeting during scheduling. Higher-priority meetings might be scheduled earlier or might even take over timeslots from previously scheduled lower priority meetings, i.e., unschedule a meeting, which might get automatically rescheduled by the Meeting Agent that is looking after that meeting. Unscheduling is performed using conflict resolution.
- **Period:** The time period that the meeting should be held, such as ‘‘within the coming 2 weeks’’, ‘‘within this week’’, ‘‘on Friday’’, etc. The exact date and time is represented by other attributes.
- **Duration:** The length of the meeting. Values may be a number of hours or minutes.
- **Part-of-day**: The part of the day that the meeting will be held. For example, values may be from: breakfast, morning, lunch, afternoon, dinner or evening. This is a coarser grain classification than hours and seems to be more natural in defining time preferences.

2.3 APPROACHES TO SOLVING MEETING-SCHDULING

There exist many approaches to deal with MS in the literature (BenHassine *et al.,* 2004; Franzin *et al.,* 2004; Modi and Veloso, 2004; Freuder *et al.,* 2001; Tsuruta and Shintani, 2000; Luo *et al.,* 2000; Sen *et al.,* 1997; and Ephrati *et al.,* 1994). Recently, the focus on solving MS using multi-agent approaches has increased due to the obvious commonalities between the two as agents can accomplish their tasks through cooperation while allowing the user to keep their privacies (BenHassine, 2005).

In this section, we discuss the main contributions to solving MS based on multi-agent paradigms found in the literature.

2.3.1 MS as a distributed constraints satisfaction problem

Distributed Constraint Satisfaction Problems (DisCSP) have long been considered an important area of research for multi-agent systems (Mailler and Lesser, 2003). This is partly due to the fact that many real-world problems can be represented as a constraint satisfaction problem (Russell and Norvig, 2003). However, there are not many published efforts on DisCSP to solve the meeting-scheduling problem. Only a few works related to DisCSP in a multi-agent system can be found in the literature (Luo *et al.,* 1992; Yokoo and Hirayama, 1996, 1998; Mailler and Lesser, 2003).

In 1992, (Luo *et al.,* 1992) presented a major decomposition technique based on breaking apart the search space by assigning particular domain elements from one or more of the variables to individual agents. One major drawback of this technique is that each of the agents has to know the variables, domains, and constraints for the entire problem. The other DisCSPs work (Yokoo and Hirayama, 1996) which presents a variable decomposition technique that involves assigning each agent one or more variables to manage giving each knowledge of the constraints on their variables.

In 1998, Yokoo and Hirayama stated that previous algorithms to solve distributed constraint satisfaction problems (DisCSPs) are neither efficient nor scalable to larger problems, since they assume each agent has only one local variable. In this work (Yokoo and Hirayama, 1998) the authors' intent is to develop a better algorithm that can handle multiple local variables efficiently based on Asynchronous Weak-Commitment search algorithm. According to (Yokoo, 1995) the priority order of agents can be changed dynamically. One limitation of this work as mentioned by the author is that the algorithm assumes each agent has only one local variable, and this assumption cannot be satisfied when the local problem of each agent becomes large and complex. They propose an algorithm based on Asynchronous Weak-Commitment search algorithm in which each agent sequentially performs the computation for each variable, and communicates with other agents only when it can find a local solution that satisfies all local constraints. By using this algorithm bad local solutions can be modified without forcing other agents to exhaustively search local problems, which leads to decreasing the number of interactions among agents. From their experimental evaluations the authors claim that the proposed algorithm is far more efficient than their previous algorithm (Yokoo, 1995) that uses the prioritization among agents. Another attempt to overcome the previous limitations of (Yokoo, 1995) and solving DisCSPs was by (Mailler and Lesser, 2003) which presents a better method of cooperative mediation by allowing agents to extend and overlap the context they use for making their local decisions. The proposed method is based on a negotiation protocol called Asynchronous Partial Overlay (APO) algorithm. The idea of this algorithm is that agents mediate over conflicts, the context they use to make local decisions overlaps with that of the other agents, the agents gain more context information along with critical paths of a constraints graph to improve their decision. To evaluate the proposed algorithm the authors implement the APO algorithm and compare the results with AWC algorithm (Yokoo and Hirayama, 2000). From the conducted experiments the authors claim that APO algorithm is both "sound and complete", and performs better than AWC algorithm for both sparse and critical graph coloring problems.

In 2005, Ferreira and Bazzan state that a previous approach (Mailler and Lesser, 2003) yields good results in simple scenarios, but there is a lack of analysis in complex real-world ones such as, distributed meeting scheduling problem. In their work (Ferreira and Bazzan, 2005) discuss the difficulties of applying the cooperative mediation OptAPO algorithm to real-world problems. The authors use Distributed meeting scheduling problem mapped as Distributed Constraint Optimization Problem DCOP using DiMES (Maheswaran *et al.,* 2004) to solve MS problem and compare the performance of the OptAPO with Adopt algorithms, then they proposed the use of heuristic search mechanisms to replace branch-and-bound search (B&B) in the cooperative mediation. They claim that the obtained results are “very promising” the heuristic version of OptAPO achieves the best solution with significant better performance, outperforming Adopt even with speedup heuristics.

The Partial-Constraints-Satisfaction Problem (PCSP) approach

The first multi-agent approach to MS problems using partial CSP was introduced by (Freuder and Wallace, 1992). The second work (Lemaitre and Verfaillie, 1997) used Distributed-Valued Constraint-Satisfaction Problem DVCSP to formalize MS problems. This work proposes a formalization of DVCSP and a greedy distributed repair algorithm for solving DVCSP. In this algorithm during an agent turn, other agents must not change their local assignment. The third work (Tsuruta and Shintani, 2000) applies a distributed synchronous algorithm to MS problems formalized as DVCSP and some agents can change their local assignments simultaneously. The problem addressed by the authors is that sometimes meeting scheduling is over-constrained and no solution exists that can satisfy all constraints. They try to develop new method for scheduling meetings that satisfies as many of the important constraints as possible by formalizing MS problem as Distributes Valued Constraints Satisfaction. The idea of this algorithm as presented is that an agent corresponds to each group member, this agent maintains its user calendar and preferences for meetings and acts on behalf of its user in meeting scheduling; users are able to keep information regarding their calendars and preferences private.

From their experiments, the authors claim that the proposed algorithm is cost- effective in comparison to the DOC method (Bakker *et al.,* 1993), it can discover a semi-optimal solution to over constrained MS problem in practical time, and determining an optimal solution for MS problem is very expensive. In 2000, (Luo *et al.,* 2000) offered a new approach for MS problems using Fuzzy Constraints Satisfaction Problem (FCSP), the authors mentioned that most existing work in MS problems ignores the issue of fusing agents’ individual evaluations for a feasible time slot. They address this issue and suggest an axiomatic framework for the fusion operation. This work proposes a kind of selfish protocol for organizing negotiation among agents. The basic idea of the protocol is in some round of negotiation the coordinator agent makes a proposal, and other agents check if the proposal can be accepted. The procedure continues until a proposal is accepted by all agents or the coordinator cannot propose any more proposals. The authors claim that the proposed approach is "novel" compared to the previous work, since MS problem is modeled by FCSP in multi-agent environment, a kind of selfish protocol is presented and an axiomatic framework is identified for fusing agents' preferences.

Constraint Logic Programming

Constraint Logic Programming has been a promising approach for solving scheduling problems (Fruhwirth and Abdennadher, 1997; Marriot and Stuckey, 1998; and Abdennadher and Schlenker, 1999). CLP combines the advantages of two declarative paradigms: logic programming and constraint solving. In logic programming, problems are stated in a declarative way using rules to define relations (predicates). Problems are solved using chronological backtrack search to explore choices. In constraint solving, efficient special-purpose algorithms are employed to solve sub-problems involving distinguished relations referred to as constraints, which can be considered as pieces of partial information.

Abdennadher and Schlenker, (1999) state that no general method exists for solving efficiently many real life problems that lead to combinatorial search, such as automatic generation of duty roster for hospital wards, the authors attempt to solve this problem using Constraints Logic Programming (CLP) framework, to model nurse scheduling problem as partial constraint satisfaction problem in CLP framework, the authors referred to (Freuder and Wallace, 1992), which deals with soft constraints by proposing a Hierarchical Constraint Logic Program (HCLP) approach to support a hierarchical organization of constraints. While another work by (Meyer, 1997) avoids the inter-hierarchy comparison in HCLP; the soft constraints are encoded in Hierarchical Constraints Satisfaction Problem (HCSP). Nurse-scheduling problem can be modeled as a partial constraint satisfaction problem that requires processing of hard and soft constraints. Hard constraints are conditions that must be satisfied, soft constraints maybe violated, but should be satisfied as far as possible. To evaluate the CLP approach, the authors have developed the INTERDIP system implemented with Siemense-Nixdorf-Informationssysteme using IF/prolog that includes a constraints package. INTERDIP has been successfully tested on a real ward of “Klinikum Innenstadt” hospital in Munich, Germany. The authors claim that the generated schedules using INTERDIP are “better” compared to those manually generated by a well-experienced head nurse.

The DRAC model

BenHassine, (2004) argued that most prior approaches to solve MS problems are centralized CSP such as (Abdelnnadher and Schlenker, 1999; Bakker *et al.,* 1993), and claims that MS problem is naturally distributed and it cannot be solved by a centralized approach. Other researchers (Garrido and Sycara, 1996) focused on using distributed autonomous and independent agents to solve MS problem where each agent knows its user’s preferences and calendar availability. However, BenHassine *et al.* (2004) mention that the majority of prior works on MS tackle it as static problem, and allow for relaxation of any constraints and do not deal with achieving any level of consistency. In an attempt to overcome these limitations the authors present a new distributed approach MSRAC based on DRAC model (Distributed Reinforcement of arc Consistency). The basic idea of this approach consists of two steps. The first reduces the initial problem by reinforcing some level of local consistency. The second step solves the resulting MS problem while maintaining arc-consistency. The authors have developed the multi-agent dynamic with Acttalk, using the Smalltalk-80 environment and generating random meeting problems, then they compared their approach with other approaches including the Asynchronous Backtracking approach ABT (Yokoo and Hirayama, 2000) and Tsuruta’a approach (Tsuruta and Shintani, 2000). They claim that the obtained results show the MSRAC approach requires in the majority of cases less CPU time than other approaches. As for the number of scheduled meetings, ABT and MSRAC schedule almost the same number of meetings.

Table 2.1 below provides a summary of the approaches the consider MS as a Constraint Satisfaction problem.

Table 2.1: Summary of DCSP approaches

Work	Approach	Main Contribution
Luo *et al.*,1992	DisCSP	Assigning particular domain elements from one or more of the variables to individual agents
Yokoo and Hirayama, 1996	Modeling and communication of constraints and preferences	Assigning each agent one or more variables to manage giving each knowledge of the constraints on their variables
Yokoo and Hirayama, 1998	Asynchronous Weak-Commitment search	Bad local solution can be modified without forcing other agents to exhaustively search local problems, decrease the number of interactions among agents.
Mailler and Lesser, 2003	Cooperative mediation	Solving DisCSPs by developing cooperative mediation protocol based on negotiation protocol APO
Ferreira and Bazzan, 2005	DCOP based on DiMES	The heuristic version of OptAPO achieves the best solution with significant better performance, outperforming Adopt even with speedup heuristics.

Table 2.2 provides a summary of the main approaches in PCSP

Table 2.2: Summary of PCSP approaches

Work	Approach	Main Contribution
Freuder and Wallace, 1992	Standard constraint satisfaction problem	Using (PCSPs), cope with CSP and take advantage of the differences between CSP and PCSP
Lemaitre and Verfaillie, 1997	Based on DVCSP to formalize MS problems	Incomplete method to solve DVCSP based on greedy repair centralized algorithm.
Tsuruta and Shintani, 2000	Distributed synchronous algorithm based on DVCSP	Discover a semi-optimal solution to over constrained MS problem,
Luo *et al*., 2000	Fuzzy Constraints	Selfish protocol is presented and axiomatic framework is identified for fusing agents'

2.3.2 Approaches based on user preferences

Multi-agent meeting scheduling involves issues like privacy, privacy loss, efficiency, and solution quality in multi-agent systems with preferences. Most studies of the meeting scheduling problem have included preference representations in their analysis and their systems (Sen and Durfe, 1995; Garrido and Sycara, 1996; Sen *et al.,* 1997; Luo *et al.,* 2000; Crawford and Veloso 2004; Franzin *et al.,* 2004). In this section we review and discuss research efforts for solving MS problems based on user preferences and discuss privacy loss. The most interesting comparisons to be made regarding preferences pertain to the way they are combined to produce global evaluations. In most existing work on meeting scheduling, it is assumed that preference values for different agents can be combined directly (Ephrati *et al.,* 1994; Garrido and Sycara, 1996; Luo *et al.,* 2000).

Ephrati *et al.* (1994) focus on two basic research problems in meeting scheduling. First, is the problem of timing - when to set a meeting. Second, how to choose the most appropriate time for a particular meeting. The authors attempt to solve these problems by introducing three scheduling mechanisms for setting up meetings in a closed system. Scheduling mechanisms are: Calendar Oriented Scheduling, Meeting Oriented approach and Schedule-Oriented Scheduling. All three mechanisms make use of primitive economic markets, where users assign “Convenience Points” to indicate their preferences over alternatives. Then each alternative is examined to establish the group decision that maximizes utility. The authors claim that the more complex the mechanism is, the better it maintains the privacy of the users.

Similarly to (Garrido and Sycara, 1996), Ephrati *et al.* (1994) state that none of the prior work took a truly autonomous agent view by considering meeting scheduling as distributed task where each agent knows its user preferences and calendar availability in order to act on behalf of its user. The idea is to build a distributed system based on the proposed system by Sycara and Liu (1994) in which agents can exchange their meeting preferences and calendar information according to some privacy policy, each agent is able to relax three constraints: date, start-time and duration. In addition, each agent has weights that indicate how to relax each time constraint.

Crawford and Veloso (2004) argue that Ephrati *et al.* (1994) three mechanisms approach are not considered as Incentive Compatible (IC), due to the fact that their proof did not account for the repeated application of the Clarke tax mechanism, rather it looked at single steps. As an alternative, Crawford and Veloso propose a mechanism in which each agent specifies own preferences for schedules and the utility of a schedule is reduced by the absence of every combination of the other participants from each meeting in the schedule. A schedule may be picked where some agents are not available for all meetings. In future work, the authors would like to further explore the problem of IC in multi-agent meeting scheduling and exploring software agents that can learn participants' scheduling preferences.

Privacy Loss Issues in User Preferences Approaches

In recent years the issue of privacy has been considered in the field of distributed constraint satisfaction problem (Franzin *et al.,* 2004). This topic has been discussed within the larger field of distributed artificial intelligence and in literature on meeting scheduling. Because privacy loss has not been a major concern, many systems use distributed protocols based on a single coordinator agent that collects all the useful information from other distributed agents (Scott *et al.,* 1998).

Starting with (Freuder *et al.,* 2001), the authors focus on an important issue that arises in cooperative communication involving independent agents, which is privacy; there are cases where some individuals are interested in restricting the information communicated to other individuals. To measure the efficiency of problem solving and privacy, the authors have implemented a multi-agent meeting scheduling system in which each agent has its own calendar, which consists of appointments in various cities on different days. The authors claim that when privacy concerns are overriding, no explicit information should be exchanged, but if efficiency is the more important concern, the best method is to combine a minimum of explicit information exchange with constraint-based inferences.

Franzin *et al.* (2004) built a meeting scheduling system based on an earlier (Freuder *et al.,* 2001) system. The authors add preferences to the new system and report the behavior of the new system under several conditions. Assuming that all agents try to maximize their preference subject to some global optimization criterion considering two basic global criteria: fuzzy optimality, having preference between 0 and 1, and maximizing the minimum preference and Pareto optimality, where a solution is optimal if there is no way to improve the preference of any agent without decreasing the preference of some other agent. From the observed result, the authors claim that fuzzy criterion can be used to lessen privacy loss with regard to preferences, and the Pareto procedure can be used to minimize information exchange and privacy loss. This work extends (Garrido and Sycara, 1996), by developing a method whereby agents can find a common meeting based on a joint function of individual preferences without actually revealing them either before or after an agreement has been reached.

Maheswaran *et al.* (2006) note that a general quantitative framework to compare existing metrics for privacy loss, and to identify dimensions along which to construct/classify new metrics, is currently lacking. The authors in this work develop a method to address these shortcomings. In particular; this paper provides additional experiments and analysis, detailed and formal descriptions of inference rules when detecting privacy loss. They refer to their previous work (Maheswaran, 2005) based on Valuation of Possible States (VPS) framework, which is designed as a quantitative model for comparing privacy loss metrics and developing new metrics. The authors present a VPS (Valuations of Possible States) framework, a general quantitative model from which one can analyze and generate metrics of privacy loss. VPS is shown to capture various existing measures of privacy created for specific domains of DisCSPs. The utility of VPS is further illustrated via analysis of privacy loss in DCOP algorithms, when such algorithms are used by personal assistant agents to schedule meetings among users. In addition, the authors develop techniques to analyze and compare privacy loss in DCOP algorithms; in particular, when using approaches ranging from decentralization (SynchBB (Hirayama and Yokoo, 1997), partial centralization (OptAPO (Mailler and Lesser, 2004)), as well as centralization. This involves constructing principled sets of inference procedures under various assumptions of knowledge by the agents. From their experimental evaluations the authors claim that decentralization by itself does not provide superior protection of privacy in DisCSP/DCOP algorithms, when compared with centralization. Instead, privacy protection requires the additional presence of uncertainty in agents' knowledge of the constraint graph. As future work, the authors state that they intent to investigate algorithms or preprocessing strategies that improve DCOP solution efficiency if privacy is a major motivation for DCOP.

Voting Schemes in User Preference Approaches

Many research efforts have addressed user preference mechanism based on voting scheme and preference estimation (Sen and Durfee, 1991; Sen and Durfee, 1994; Sen and Durfee, 1996; Sen *et al.,* 1997 and Chun *et al.,* 2003; Shakshuki *et al.,* 2007).

Early works (Sen and Durfee, 1991; Sen and Durfee, 1994) focus on the problem of how an application domain for intelligent surrogate agents can be analyzed. One drawback of this work is that it does not address many implementation issues like communication medium, user interaction, and use of preferences. Sen *et al.* (1997) stress the importance of user preference and they model preferences as elections between different alternative proposals. To avoid conflicts among user preferences they use a technique from voting theory which allows agents to arrive at a consensus choice for meeting times while balancing different user preferences. They have implemented distributed meeting scheduling system in a work-station-based computing environment, in this system the user interacts with the meeting scheduling system through the user interface. The authors claim that their autonomous scheduling can approximate the privacy and security concerns of users, and allows for better throughput and better fault tolerance.

Chun *et al.* (2003) state that traditional optimal algorithms do not work without complete information about individual preferences, their work presents a new technique called "preference estimation" using "preference rules" that allow to find optimal solution to negotiations problems without needing to know the exact preference models of all the meeting participants beforehand, the authors describe and use two algorithms based on the negotiation protocol. Algorithm 1 (NWOPI) and Algorithm 2 (NWPI) implemented on an environment called Mobile Agent for Office Automation MAFOA (Wong *et al.,* 2000). Simulations performed to compare these algorithms support the algorithm 2, using preference estimation, finds the optimal solution at only a slightly higher cost than algorithm1, which relies on relaxation.

Shakshuki *et al.* (2007) mention that current approaches in scheduling meetings do not act on behalf of users to manage and negotiate meetings automatically. The authors intend to equip these agents with negotiation strategies to help users automatically book meetings with minimum conflicts. They refer to an automated agent-based meeting scheduler that has been proposed by Berres and Oliveira, (2005) which proposes two negotiation strategies to find free time slots for booking meetings. They also refer to Modi and Veloso, (2006) that proposes three useful negotiation strategies including greedy bumping and NCost. It is assumed in these strategies that the initiators always propose a single time slot and an agent either accepts or rejects time slots. To overcome previous limitations they propose an approach in which individual agents are able to vote on meeting times. They develop a client-server architecture that consists of a Meeting Scheduling Server Agent (MSSA) and Meeting Scheduling Client Agent (MSCA). The client communicates with the server through TCP/IP sockets. The proposed system uses two negotiation strategies one strategy equipped with MSSA the other strategy on MSCA both strategies are able to vote on a meeting time. No experiments have been provided in their work to evaluate performance to their proposed strategies. However, they implement the system as a middleware that connects to all MSCAs, they claim that proposed voting approach reduces wait times and allow the initiation agents to maintain control of the negotiation process.

Table 2.3 given below provides a comprehensive summary of user-based approaches to meeting scheduling.

Table 2.3: Summary of user preference methods

Work	Approach	Main Contribution
Ephrati *et al*., 1994	Primitive economics markets	Additional user privacy could be maintained at the cost of decreased stability of the system
Garrido and Sycara, 1996	Modeling and communication of constraints and preferences	MS performance is more stable and constant when agents try to keep their calendar and preference information private.
Sen *et al*., 1997	User preference based on Voting scheme	The proposed system can approximate the privacy and security concerns of users, and allows for better throughput and better fault tolerance
Freuder *et al*., 2001	Constraint-based inferences	When private information exchanged efficiency not improved unless sing o inference
Chun *et al*., 2003	Preference estimation based on the preference rules	Find optimal solution to negotiations problems without needing to know the exact preference models of all the meeting participants beforehand
Crawford and Veloso, 2004	Maximizing agents' utilities	Show how IR problem can be reduced, making mechanism design work in real-world is a theoretically challenging problem
Franzin *et al*., 2004	Fuzzy optimality, Pareto optimality	Minimize privacy loss, maximize solution quality, to be fast
Maheswaran *et al*., 2006	Valuation of Possible States (VPS) framework	Privacy protection requires the presence of uncertainty about agents' knowledge of the constraint graph
Shakshuki *et al*., 2007	Negotiation based on voting	Present an approach in which individual agents are able to vote on meeting times

2.3.3 Approaches that consider spatial constraints for meetings

Recently, the problem of choosing a meeting location has become a research focus, Chithambaram and Miller (2005) introduce a system to find meeting location that is closest to the geographic center of meeting participants. Their method averages the latitude and longitude of each participant then proposes the best meeting place by selecting the nearest location to the center from a list of points of interest. Kaufman and Ruvolo (2006) introduce a method to optimize location selection considering the current locations of participants that is obtained from GPS coordinates or the location of other events in the participants' calendars. Their method calculates the proposed location based on proximity to the participants and availability of the resources needed at the location.

Santos and Vaughn (2007) mention that many people have tried to address the problem with networked calendars and web-based conference room schedulers to select adequate times. However, there are not yet many solutions that handles geographically dispersed participants or venues. To better find a location to meet the authors propose a method for finding location-based meeting venue. They refer to the earlier works of (Chithambaram and Miller, 2005; Kaufman and Ruvolo, 2006). The authors argue that the latter serially applies filter such as airfares but does not solve in aggregate such potentially conflicting multi-critical costs as money, time or social constraints. To overcome their deficiencies in selecting central locations to several participants, they propose a method that combines selection of optimal meeting locations from a list of candidates points of interest. By minimizing the travel cost for each participant, the cost function changes based on the meeting scenario. For example, for the scenario of friends meeting, the cost is the time to get to the restaurant, if the cost is high the system gives a set of candidate locations to the organizer. No algorithm has been presented. However, the authors show an example of optimization results from their system. In their future research the authors intend to address more issues that can arise given times and sites. In addition, they intend to consider ways to deal with unforeseen factors outside system controls.

More recently, Berger *et al.* (2008) state that most of current meeting scheduling systems take into consideration only time and not location geometry to schedule a meeting, they derive an efficient algorithms for solving meeting scheduling problems, and integrate the solutions into an application that allows users who are connected over a network to schedule a meeting. They use linear programming concepts to provide a solution to meeting scheduling. The authors formalize meeting scheduling problem geometrically, the choice of location relies on the geometry of the problem (participant locations with respect to meeting places). The objective of the linear programming optimization is to schedule the meeting of the longest possible duration for participants.

None of the mentioned methods have formalized selecting meeting place to be solved using multi-agent negotiation. More specifically, the approaches presented here suffer from the following:

- Chithambaram and Miller (2005) rely on geographical proximity to decide a meeting place for all participants. Although this is generally a good approach, it does not take into account several factors which may affect the complexity of the task, such as the cost of traveling for the participants and the local geography which may make traveling difficult despite the short distance.
- Santos and Vaughn, (2007) use a centralized approach that relies on the meeting organizer to decide whether or not to select a point for meeting. The only form of participation the agents have in the selection process is through voting on the point selected by the organizer. Therefore, the negotiation takes place between every participant and the organizer. No negotiation takes place among the participants.
- In Santos and Vaughn (2007), the organizer provides a list of possible meeting places and it is up to the participants to select an ultimate place. Although this may seem advantage at first sight, it often happens that some agents do not vote on the eventual meeting place and end up meeting in a place that is not suitable for them.
- Santos and Vaughn (2007) do not consider the type of meeting or whether or not a subset of the participants is obliged to attend. This eventually affects the complexity of the selection method.

Table 2.4 given below provides a summary of the approaches provided in this section.

Table 2.4: Summary of spatial approaches

Work	Approach	Main Contribution
Chithambaram and Miller, 2005	Spatio-semantic modeling	Find meeting location hat is closest to the geographic center of several participants
Kaufman and Ruvolo, 2006	Optimize location selection based on GPS coordinates	Find optimized location by calculates the proposed location based on proximity to the participants and availability of the resources needed at the location.
Santos and Vaughn, 2007	Multi-criterial calculations	Propose a method that combines selection of optimal meeting locations from list of candidates' points of interest.
Berger *et al.,* 2008	Linear programming	Formalizes meeting scheduling problem geometrically, the choice of location relies on the geometry of the problem

2.4 MEETING SCHEDULING NEGOTIATION MODELS

Negotiation about collaboration in space and time is becoming important in a large class of real world problems. In meeting scheduling, most research efforts targeted time only while some paid attention to both time and space. In this section, we identify recent work in two categories: research efforts in presenting canonical negotiation models and proposed negotiation strategies for meeting scheduling.

2.4.1 Canonical negotiation models

Canonical problems are simplified representations of a class of real world problems. A canonical problem captures the most important challenges of a real world problem to allow researchers to develop algorithms for a class of real world problems by associating them with canonical models (Luo and Boloni, 2007). Examples of canonical problems pertaining to collaboration are:

- Splitting the Pie Games

- Children in the Rectangular Forest(CRF)

Splitting the Pie Games

The pie-splitting game is a well studied classic problem in game theory and is also known as the fair division problem, cake-cutting problem, or split-the-pie game.

In 1994 [Osborne and Rubinstein, 1994] studied the pie-splitting game by considering it as a model for bargaining among several participants over splitting the pie. According to this model, participants negotiate and bargain to maximize their individual piece from the pie.

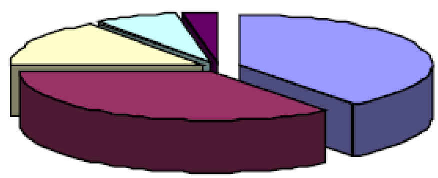


Figure 2.1: Participants bargaining to maximize their individual piece

(Luo and Boloni, 2007) mentioned splitting multiple pies negotiation where participants negotiate over multi-issue that can be handled by having to split multiple pies. The agents' total utility is represented by a function dependant on pie shares. For reasons related to computation complexity, they represent the utility function by a weighted sum over the pie shares received by each agent. The agent might or might not know the utility function of their negotiation partners.

An example of splitting multiple pies (taken from Luo and Boloni, 2007) is given below. Assume a manufacturer's suggested retail price P_{MSRP} for a car, and the dealer's invoice price $P_{invoice}$. The "pie" will be represented by the amount of P_{MSRP} - $P_{invoice}$ which represents the net amount of profit split by the dealer and buyer when negotiating a deal between them. Extended negotiations would reduce this profit through inflation or through the cost of storage to the dealer, or cost of renting a car for the buyer and so on. Therefore, it is a shrinking pie.

Children in the Rectangular Forest

Luo and Boloni, (2007) state that there are cases when splitting multiple pies can not capture the essential challenges of real world problems. In other words, it can not be considered as a canonical problem. To represent real world problem, the authors propose an alternative canonical problem to splitting multiple pies model, called the Children in the Rectangular Forest, they also present a negotiation model that is used to analyze the components of the proposed negotiation approach including negotiation procedures, negotiation protocol, strategies and utility function.

The authors argue that spatio-temporal negotiation is technically a multi-issue negotiation and splitting multiple pies does not capture the essence of these problems. Therefore, they propose Children in the Rectangular Forest as an alternative to splitting multiple pies where cooperating agents are represented by children whose shortest paths to their respective destinations crosses a rectangular forest. However, one agent cannot cross the forest alone. Therefore, the agents negotiate a common path to cross the forest that will save them from having to go around the forest independently. Figure 2.2 illustrates how it can be beneficial to the agents to join each other in traversing the forest.

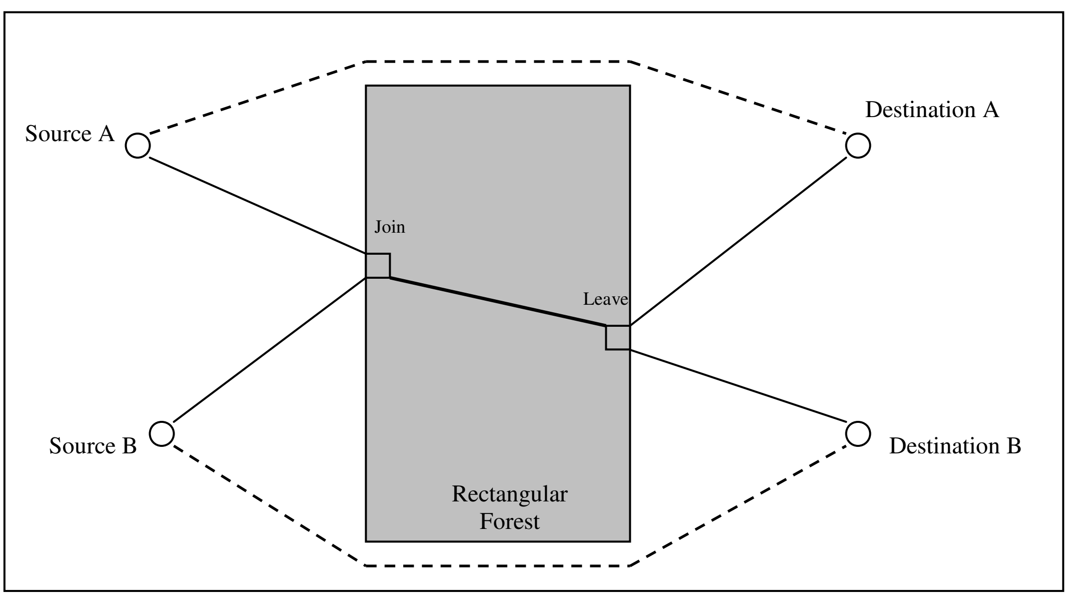


Figure 2.2: Children A and B cross the forest together (solid lines) or go around the forest separately (dashed lines)

In the CRF model, the object of the negotiation is to agree on a join time and join location as well as a leave location and a speed for traversing the forest. Therefore it is a four-issue negotiation model with points along the join edge of the forest reached by both agents at a certain time forming a Pareto optimal front. An agent would prefer a collaborative deal to cross the forest with other agent if it saves time and/or travel distance compared to the conflict deal going alone around the forest. The model generalizes to any number of children and an alliance emerges between any two who agree to cross the forest together.

Negotiation strategies and protocols for the proposed CRF model have also been introduced; the first strategy is supervised negotiation that uses an external mediator to select the offer on which agreement will be based, while the two agents judging offers based on their utility evaluations, then a mediator selects an offer based on its view as supervisor. The second protocol relies on internal urgency criteria of the agent in which every agent separately decides on the maximum number of negotiation rounds in a away that other agents (opponent) does not know, then agents start evaluating each other offers based on pre determined OPT value. The authors did not mention any experimental results nor implementation for the proposed negotiation protocols. However, they state that the first strategy is time consuming since it amounts to an exhaustive exploration of the solution space.

2.4.2 Negotiation strategies in MS

While automated negotiation generated a lot of interest in recent years negotiation about spatio-temporal issues in embodied agents has received relatively little attention (Luo and Boloni, 2008).

Starting with (Sandholm and Vulkan, 1999) that analyze the problem of negotiating with internal deadlines where the deadlines are private information of the agents. The negotiation problem is a split a single pie zero sum negotiation, they find that for rational agents, the sequential equilibrium is a strategy which requires agents to wait until their deadline, and at the moment the agent with earliest deadline concedes the whole cake. Fatima *et al.* (2001) present a single-issue model for negotiation between two agents under time constraints and incomplete information setting determined optimal strategies for agents but did not address the issue of the existence of equilibrium.

Fatima *et al.* (2002) adopt their earlier framework to examine the strategic behavior of agents that result in equilibrium. They extend their framework to multi-issue negotiation between a buyer and seller for the price of more than one good or service based on specifying a deadline for each agent before which agreement must be reached on all the issues. A model for multi-issue bargaining has been developed where the issues are independent of each other. The authors show two possible implementation schemas. The first one is a sequential implementation in which agreement on an issue implemented as soon as it is settled, while the other is a simultaneous implementation in which agreement is implemented only after all the issues are settled. The sequential implementation of equilibrium agreement results in an outcome that is no worse than the outcome for simultaneous implementation when agents have similar as well as, conflicting time preferences. The authors claim that negotiation issues in the proposed model is considered to be independent of each other. In addition to the important property of the model is existence of a unique equilibrium which resulted in agreement at the earlier deadline.

Fatima *et al.* (2006) study the problem of multi-issue negotiation under deadline. They compare three negotiation procedures: the package deal procedure, where all the issues are negotiated together. Simultaneous procedure where issues are discussed independently but simultaneously, and sequential procedure where issues are discussed one after another. Crawford and Veloso (2006) state that previous work on negotiation for multi-agent meeting scheduling has not looked at how agents can negotiate strategically. In order to better satisfy their user preferences, the authors propose an approach for agents to negotiate where the agents learn which strategies to use in different situations. They present a framework that shows how the problem of learning to negotiate strategically with other agents can be framed as an experts problem by adapting two previously proposed algorithms: the Exploration-Exploitation Experts (EEE) Algorithm (de Farias, D., Megiddo, 2005) and the playbook approach (Bowling, 2004). The experts algorithm is then used to instruct the learning agent on which strategy to use each time it negotiates for a meeting. From performing experiments the authors claim that agents can learn to select good strategies for different situations

Zaki and Pierre (2007) point out that millions of users will have an access in future to portable devices such as next generation of phones. To take advantage of software agent-based solutions the authors propose solutions based on mobile agents for distributed meeting scheduling and evaluate the performance of these solutions in terms of required scheduling time. They referred to other contributions related to the use of mobile agents in meeting scheduling. For example Sanchez and Alonso (2003) develop a multi-agent system that implements distributed meeting scheduler and personal agenda meeting. Sciaffino and Amandi (2002) present an architecture that enables interface agent developers to build software secretaries. The authors propose an architecture that is based on blackboard system. The key component in this architecture is the client agent (CA) and scheduling agent (SA) in addition to other modules and components. The authors did not mention a negotiation algorithm among these agents. However, they propose a coordination algorithm that uses ontology elements to accomplish the scheduling application confirming the capacity of MAS to achieve this distributed task, by combining the results obtained from different system agents. The algorithm runs until a scheduling succeeds or fails. To evaluate the performance of the proposed algorithm, two metrics are used: network load, which measures total load generated on a network, and scheduling time, which measures the time required to schedule a meeting from launch of the application until reaching a successful or a failed state. The experiments using the proposed MAS prototype show that robust meeting scheduling performance can be achieved at a determined computational and communication cost. Luo and Boloni (2008) state that equilibrium negotiation strategies are not practically possible. They are interested in strategies with bounded rationality to achieve good performance in wide range of negotiation scenarios. They introduce three negotiation strategies for the CRF problem, with "no initial" information, Monotonic Concession in Space (MCS), Internal Negotiation Deadline (IND) and Estimate of the Opponents Parameters approach (EOP). MCS strategy is where the monotonic concession in space agent is parameterized by pain representing the concession rate in the meeting point and splitting point respectively. IND strategy where the agent sets to itself a deadline, if the deadline expired without an agreement being reached the agent breaks the negotiation deadline, so it is parameterized by deadline. The EOP strategy tries to improve its offer formation by estimating opponents speed and current location based on offers and evaluations made by opponents. The authors mention that none of the strategies are representing particular offer concession. The authors perform a set of experiments that measure the utility achieved by each specific strategy against specific opponents under scenarios with various levels of collaborativeness. They run the experiments using rejection sampling borrowed from MonteCarlo simulation methods. The results show that relative utility increases with collaborativness. The agents using EOP strategy are able to consistently achieve higher utility values than the IND agents which shows the superiority of EST strategy. For low collaborativeness levels, the IND strategy performs worse against the EOP strategy than against EOP strategy. The authors claim that negotiation against a sophisticated strategy leads to lower relative utility for scenarios with low collaborativeness levels, but it becomes an advantage when negotiating in scenarios with high collaborativeness levels.

Table 2.5 given below provides a summary of the main approaches provided in this section.

Table 2.5: Summary of negotiation strategies approaches

Work	Approach	Main Contribution
Sandholm and Vulkan, 1999	Internal deadline negotiation	Problem of negotiating with internal deadlines where the deadlines are private information of the agents
Fatima *et al*., 2001	Single offer, combined offer negotiation based	Developed a model for bargaining
Fatima *et al*., 2002	Multi-issue negotiation	Prove strategic behavior of agent that result in equilibrium
Crawford and Veloso, 2006	Experts algorithm	Present a framework to show how agents learn to negotiate strategically
Zaki and Pierre, 2007	Mono-agent and multi-agent strategies	Propose a coordination algorithm that uses ontology elements to accomplish the scheduling application
Luo and Boloni, 2008	Multi-issue negotiation	Propose three negotiation strategies for CRF problem, with “no initial” information, MCS, IND and EOP

2.5 CONCLUDING COMMENTS

In this chapter, we have shown that the meeting scheduling problem cannot be solved by a centralized approach due to its dynamic features. It requires the cooperation of different distributed agents to reach an agreement of coordination in order to achieve scheduling task with minimal cost. Current approaches suffer from the disadvantages of not considering combined spatio-temporal constraints that are required in order to render the meeting scheduling solution a real-world reliable one capable of dealing with the heterogeneous preferences of the individual participants while keeping in mind privacy issues.

The CRF Canonical problem deals with collaborating agents aiming to reach an agreement on time and space. Our aim is to model the MS as a canonical problem and use the useful properties of the CRF to embed spatio-temporal constraints in MS.

In the following chapter, we present a model that formalizes the meeting scheduling problem as a CRF canonical problem. The end result is a generalized model that can deal with different constraints to be enforced on the MS problem such as combine spatio-temporal constraints.

CHAPTER 3

ASSEMBLING CHILDREN IN THE RECTANGULAR FOREST

In this chapter we present a formal model to represent meeting scheduling constraints and examine existing canonical models by mapping meeting scheduling problem to CRF problem. We also formulate a negotiation algorithm that captures spatio-temporal constraints resulting from this framework.

This chapter is organized as follows. After presenting our motivation to include spatio-temporal aspects in meeting scheduling in Section 3.1, we introduce mapping of meeting scheduling to CRF model in Section 3.2 that includes time and space along with new parameters and metrics. Section 3.3 presents the proposed initiator-agent negotiation strategy followed by a proposed set of algorithms that formalize the negotiation protocol followed by an explanation for each algorithm along with an example to illustrate the idea. Section 3.4 presents the algorithm properties and its use and Section 3.5 concludes the chapter.

3.1 MOTIVATIONS

3.1.1 Limitations of the current approaches

Meeting scheduling is a common task for organizations of any size. Most of the research efforts consider finding a suitable time for a meeting and ignore choosing a location or simply assume a central location. Central locations are not the best for all people.

The "Children in the Rectangular Forest" model (Luo and Boloni, 2007) claims to be a simplified representation of class for real world problem. However, the CRF model has not been considered for scheduling events that can involve constraints in time and space.

Therefore, in order to make CRF model represent more complex collaboration patterns, it must accommodate the possibility of having a meeting in different locations in space.

3.1.2 Potential benefits

Integrating spatial and temporal constraints is relevant to many applications including meeting scheduling. We would like agents to organize and schedule our events autonomously by scheduling meetings for us. Another potential feature resulting from jointly considering time and space constraints is mobility. A mobile meeting can take place between individuals traveling towards some destination which can save time and effort for participants in the meeting. Moreover, a mobile meetings are useful for problems like airplane crew scheduling and carpool shuttles scheduling systems.

3.1.3 Aim

In this chapter we incorporate the temporal and spatial aspects of meeting scheduling into the CRF model and present a negotiation protocol for meeting selection.

3.2 MAPPING MEETING SCHEDULING TO CRF

Previous approaches show that negotiation about collaborative actions in space and time is a large class of problems with important practical applications. In many cases, negotiating a meeting schedule involves reaching an agreement on the meeting location meeting start time, and meeting duration. The CRF model is a rich canonical problem for spatio-temporal negotiation that can be mapped to formalize meeting scheduling.

The following considerations are important in the proposed approach for meeting scheduling:

- The characteristics of meeting scheduling such as, start time, end time and number of attendees can also mapped to multi-issue negotiations.
- Split the pie game cannot capture the characteristics of negotiation in time and space as previously argued by (Luo and Boloni, 2007;2008).
- Spatio-temporal collaboration in meeting scheduling can be translated to a negotiation protocol that can be used in embodied agents.

Location can be mapped to CRF join point, start time is mapped to CRF join time, and the meeting duration represents the time to cross the forest. In addition, the conditions for crossing the forest must be modified to ensure that the meeting constraints are met. For example, instead of allowing any two agents to cross the forest, we require that a quorum including all essential participants is present.

The mapping of the meeting scheduling problem to the CRF model is not yet complete. Two aspects remain outstanding: the first is the conflict deal, and the second is the leave location. The conflict deal in meeting scheduling represents the penalty associated with failure to participate in a meeting. Such penalty is context-dependent and can be specified by the user as in some previous work on meeting scheduling (Garrido and Sycara, 1995; BenHassine *et al.,* 2004; Crawford and Veloso, 2004)

3.2.1 Mobile meeting

The last outstanding element in mapping meeting scheduling to the CRF model is leave location. In most cases, meetings are held in stationary locations. However, if meetings are allowed to end in a different location, then a new class of meetings emerges: mobile meetings.

The meeting scheduling problem then becomes a generalization that captures useful aspects of some other problems like the car pooling problem (Burmeister *et al.,* 1997), and the flight crew scheduling problem (Castro and Oliveira, 2005). Moreover, integrating mobile meetings in a meeting scheduling system allows users to become more efficient by holding meetings on their way to other destinations as appropriate.

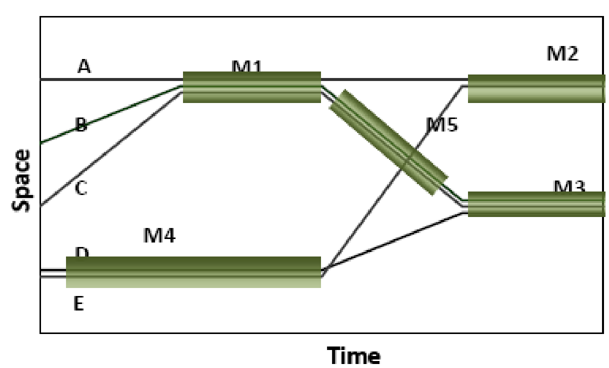


Figure 3.1: Five meetings involving 5 individuals (A, B, C, D, and E). Meetings M1 to M4 are stationary and M5 is a mobile meeting.

Normally, an individual participates in many meetings on a given day with some possible commutes between meeting locations. Figure 3.1 represents an illustrative example. In the figure each continuous line represents an individual and a sloping line represents mobility (spatial change over time).

3.2.2 Negotiation in meeting scheduling

In this section we introduce a proposed negotiation framework that has resulted from mapping meeting scheduling to the CRF model.

There are essentially four elements that govern negotiations in meeting scheduling:

1. type of the meeting
2. utility functions
3. user preferences
4. negotiation strategy

Type of the meeting: To establish negotiations for meeting scheduling attendance of participants in the meeting needs to be defined. It is necessary for agents to know whether attending the meeting is necessary or optional since attending the meeting has a direct effect on the cost and utility for every agent. In this negotiation model the initiator indicate' the type of meeting and whether all participants must attend, a quorum must be present, or whether the attendance of certain individuals is optional.

Utility function: Utility is a numeric value that represents how desirable a state is [Wooldridge, 2002]. Utility functions could be any real valued function that depends on environment state.

Agents negotiate deals that maximize their utility and minimize their cost. For example, in negotiating a meeting location the cost function for meeting scheduling can be the distances between agents locations and meeting location

$$\text{C:}\ \vartheta(Dis(Start\ , p_{join}, p_{leave})_{agent(i)}) \leftarrow Cost$$

The cost function must satisfy two constraints. First, it must be monotonic which means that adding more offers should not decrease the cost. In meeting scheduling the initiator proposes one initial offer. The second constraint is that the cost of doing nothing is zero $C(\phi)=0$. Then, every agent calculates what every offer will cost when it evaluates the offer. Evaluating offers is calculating the difference between the value assigned to the offer, and the cost associated with it.

$$Utility(Offer) = value(meeting_{agent(i)}) - Cost(offer_{agent(i)})$$

The utility of an offer represents how much the agent has to gain from the offer, if the utility is negative then the agent is worse off if it accepts the offer. Every agent generates its utility based on the $value(meeting)$. In the above formula $value(meeting)$ is the value gained from attending the meeting, $Cost(offer)$ is value of the cost to attend the meeting, $Utility(Offer)$ is the utility value for the agent from accepting an offer.

User preferences Meeting scheduling has been the means of studying relations among privacy, privacy loss and solution quality in multi-agent systems with preferences. Most studies in meeting scheduling have included preferences representation in their analysis. User preferences can change the agreement between the agents during negotiations. For instance assuming $Utility(Offer)_{Agent(A)} > Utility(Offer)_{Agent(B)}$, the agent would choose A's offer even if it had chosen B's offer earlier.

In meeting scheduling agents assign weights for each meeting. The weights represent their private preferences. For example, some participants will not prefer to have a meeting in early morning, and would prefer having it in afternoon. Therefore they assign more weight for "afternoon" than the "morning" meeting. User preferences should be private for all attendees.

Negotiation strategies: Selecting a negotiation strategy is essential in meeting scheduling. A negotiation strategy is a collection of rules and procedures for agents to decide when to propose a specific offer and when to insist or propose something new. After all, it is a negotiation problem that requires cooperation from all agents to find a solution or a deal. Automating negotiation can be very useful as it provides a distributed method of aggregating distributed knowledge among agents.

Finding a good strategy in meeting scheduling negotiation is vital. There are different ways to select a negotiation strategy. Market models that have been studied in the literature include auction-based approaches or seller-buyer models. Other ways to build a good strategy is to construct a model-based approach for agents and based on this model select the strategy that can maximize the utility and best satisfy preferences. We followed this approach in this work since new parameters have been introduced in CRF negotiation. The next sections explain the proposed negotiation strategy in more detail.

3.2.3 Notations and definitions

Definition 1 Agent-in-Meeting is a predicate with 7 attributes

Agentinthemeeting(*Start*, *Curr*, *FinalDis, role, status, velocity,duration*)

These attributes specify the location: *Start* is the original location of agent i in (x,y), *Curr* is the current location in case agent travel from his original location, *FinalDis* to give destination location.

Definition 2 A proposed offer for agent i, includes a value for a negotiated attribute

$$(O^{pro})_i \xleftarrow{\quad} offer \ \ (p_{join});(p_{leave});(t_{total});(velocity)_i$$

In sequential negotiation, all offers deal with one issue until an agreement is reached. For examples, agents may negotiate a point to join P_{join}, then propose place to leave P_{leave}, t_{total} total time required to travel from P_{join} to P_{leave} based on agents' $velocity$.

Definition 3 The agent's resistance level controls whether to propose offer for a meeting mj:

$$\lambda_{R+1} = \lambda_R - \Delta_R$$

λ_R signify agent own resistance to accept offer in negotiation round R, λ_0 is typically high. When λ is high the agent insists on its offer (O^{pro}) and is not willing to accept others' proposals, after each round of negotiation λ decreases making the agent more likely to accept others proposals based on a changing Δ value. Δ represents the concession rate per round which typically decreases with time. See 3.2.4 for more detail about λ and Δ.

Definition 4 A negotiation round R is the time period taken by agents to make one proposal. Each agent repeats the negotiation protocol until they reach an agreement or a maximum number of negotiation rounds R_{Max}

Definition 5 The best evaluation for a proposed offer ($E_{\max}^{pro}$) for a specific agent is given by:

$$E_{\max}^{pro} \longleftarrow Max(utility\,(O^{pro}))$$

Identifying the best proposed offer among all offers is based on the utility of potential partner offers

Definition 6 The agent whose offer is the closest to the agents' is considered the nearest agent to agent i (NA_i). We use a distance to compare two offers.

Definition 7 $(O^{new})_i$ is a new offer that agent i proposes after refusing other proposed offers, Thus an agent generates alternative offers to what it has proposed previously $(O^{pro})_i$

Definition 8 The essential participants in a meeting mj where meeting mj is a quorum-based meeting and meeting mj can not be held without their attendance is called EP

$$EP \longleftarrow EssentialParticipant_{mj}$$

The set of EP is populated based on the *status* parameter in meeting constraints which specifies whether a participant is required or optional to attend

Definition 9 A group of agents whose proposed offers are identical form group G_i

$$G_i \longleftarrow \{Agent_1,....Agent_n \mid \forall agent_i, agent_j \in G_i, (O^{pro})_{agenti} = (O^{pro})_{agentj}\}$$

Where the set $\{Agent_1,....Agent_n\}$ include only agents who agreed on the same offer

Definition 10 we call the list of all incoming offers within a negotiation round $Olist$

$$Olist \longleftarrow OfferList\,\{O^{pro}{}_n, O^{pro}{}_{n+1}\}_{Mj}$$

Where $Olist = \{\phi\}$ at beginning of negotiation

Definition 11 we call the list of known locations by $Agent_i$ for meeting mj

$$KL_{mj} \longleftarrow KnownLocation(x_i,.....x_n)$$

Where x_i first location in the list up to x_n, location expressed by a place and its coordinates.

3.2.4 Resistance representation

As mentioned earlier, the negotiation problem is one where multiple agents try to reach an agreement or deal α, that maximizes their own utility $utility(O^{pro})_i$. We say that every agent has a $utility_i$ that is defined from $\alpha \in \{O^{pro}\}_i$

The *Network Exchange Theory* (NET) (Willer, 1999) represents negotiation and finds the deal based on a resistance equation. In this model agents are represented by nodes in a graph, and annotated edges between nodes represent the possibility of negotiation between two agents.

Agents in the CRF model are negotiating a set of proposed offers over spatio-temporal parameters according to their utility functions. Every agent in the CRF creates a directional edge to the other agents around the forest to express their chance to negotiate. Assume three agents negotiate to cross the forest in the CRF model as follows:

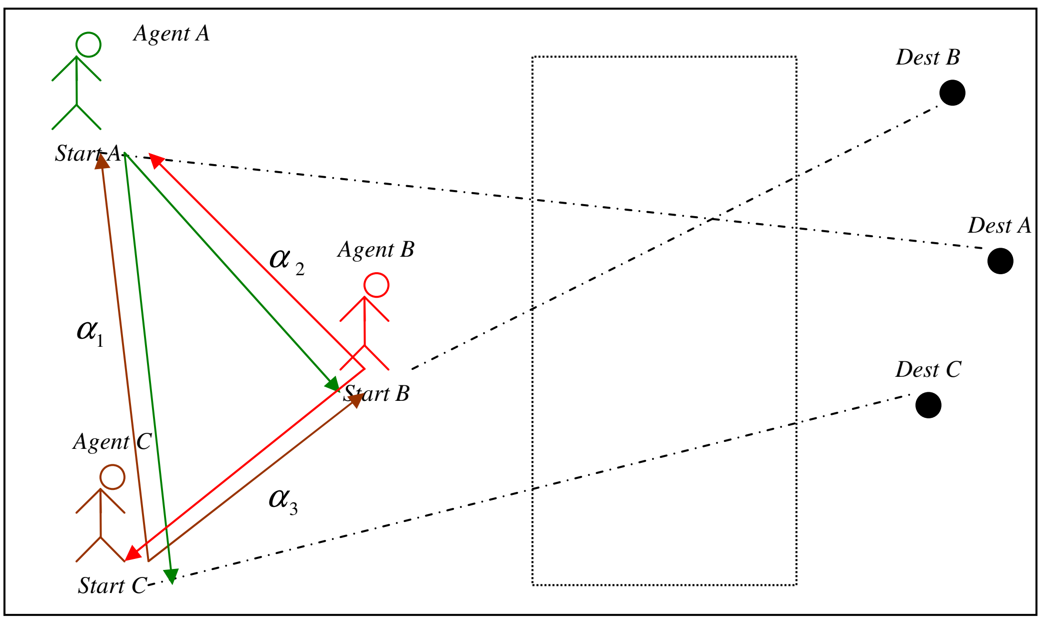


Figure 3.2: Represent the possible interaction between agent *A,B,C*. The edges between three agents represent possible negotiation where α_1 is a set of proposed offers that agent *A, B* can agree on

Based on the *Network Exchange Theory,* each agent has a resistance to each particular offer given by resistance equation. Moreover, it should be a point (offer) between *A, C* in Figure 3.2 where both agents can agree on deal $\subseteq \alpha_1$

$$Resistance = \frac{utility(O^{best})_A - Utility(O^{pro})_A}{Utility(O^{pro})_A - \min(utility(O^{pro})_A)}$$

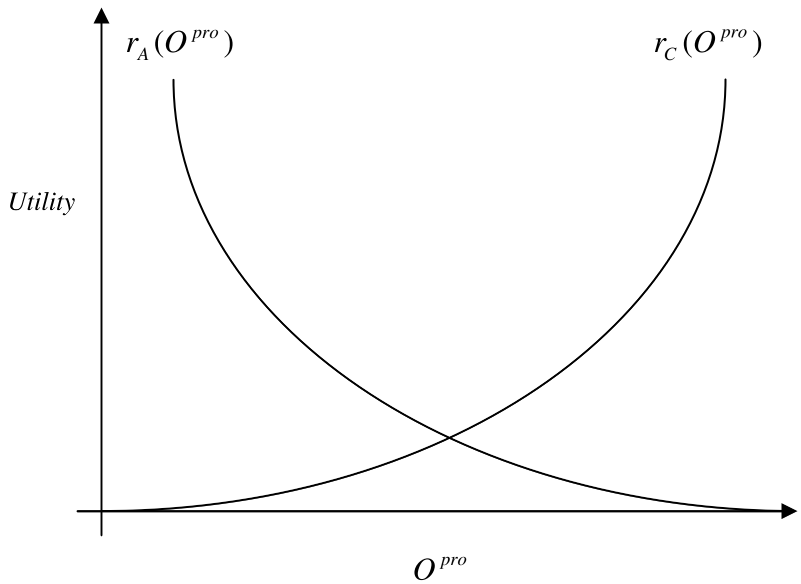
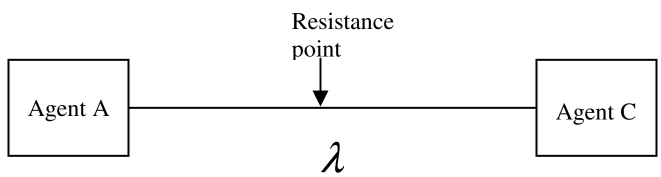


Figure 3. 3: Resistance point of two agents *A,C* for two offers. *Taken from (Vidal, 2004)*

A resistance point in this scenario captures the agents willingness to whether agree on the proposed offer or propose another offer that can maximize the utility. Agents negotiate to maximize their utility until they reach the no deal which is a point that every agent can not agree on anymore, at this point agent would have a minimum utility to agree on that offer.



As defined earlier, λ expresses the agents' resistance point or willingness to accept offers. λ_0 starts high when agents first proposes their offers and try to insist on their initial offers O^{pro} but as negotiation goes agent start expressing different resistance towards proposed offers where Δ controls the change of resistance in every round, at the beginning $\Delta_0 = 0$ this value gradually reduces λ according to $\lambda_{R+1} = \lambda_R - \Delta_R$

The more negotiation rounds agent goes, the less resistance and agent *i* becomes more likely to accept other proposals.

How to obtain Δ for every round

The concession rate Δ is a value that changes every round Δ_R, it has been used to determine resistance level where $\lambda_{R+1} = \lambda_R - \Delta_R$. In our experiment and implementation we use

$$\Delta_R = \frac{k}{R^2}$$

Where k is a constant and R is round of negotiation, based on Riemam Zeta function:

$$\sum_{n=1}^{\infty} \frac{1}{n^x} ,$$

Two considerations influence the value of the constant k: the initial value for λ, and the maximum number of rounds R_{max}. The summation of the chosen function converges at infinity to a value close to $k\ \pi^2/6$. Therefore, if we want λ at a sufficiently large R_{max} to approach 0, then we should set $k = 6\ \lambda_0 / \pi^2$.

3.3 INITIATOR BASED NEGOTIATION STRATEGY

In many negotiations agents must reach an agreement on matters of common interest with other agents. In meeting scheduling, negotiation needs some sort of control to help make decisions whether it is possible to schedule a meeting or cancel the meeting. The initiator organizes a meeting among other agents and balances as well as controls offers in the negotiation process.

The main advantage of using an initiator in meeting scheduling is to protect the negotiations from continuing forever without result. By specifying a maximum number of rounds for every meeting, and monitoring the progress of negotiations, the initiator takes proper actions.

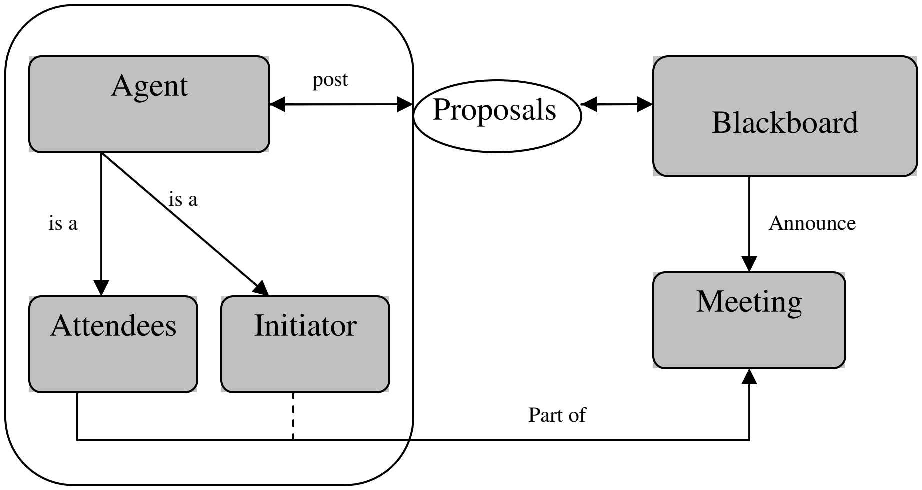


Figure 3.4: Initiator strategy interactions

To establish any negotiation these entities/objects must be involved in the negotiation process: agents as attendees, initiator, meeting and a set of proposals that is generated by agents. As Figure 3.4 shows, there are two entities derived from agents, attendees as normal agent and an initiator agent, they both interact through a blackboard. The initiator announces the meeting details and attendees on the blackboard where its visible for all agents to read.

3.3.1 Initiator agent

It is the agent who requests and organizes the meeting to be held with information about attendees and monitors whether meeting can be held or not. The role of the initiator is to:

- Determine whether the meeting will be held or dropped by checking for a quorum.
- Monitor the progress of the negotiation and identify if the negotiations have been stalled.

There are two types of agents that are involved in the negotiation based on the proposed negotiation strategy: initiator agent and normal agent. As mentioned above, the initiator has a control on proceeding or terminate any negotiation at any time. The design of the negotiation protocol is divided mainly into two parts: initiator-agent side and normal agent side.

Initiator-agent side

- set meeting properties by identifying participants, set minimum number of required attendees for every meeting, set agenda, and specify status of participants
- initiate a meeting by generating an initial proposal
- send request for meeting to all agents post it on the blackboard
- wait x-seconds
- initiator agent checks whether there is a counter offer/agreement
- end/ proceed with negotiation, based on checking criteria
- monitor the progress of negotiations

Normal-agent side

- receive proposals from initiator on blackboard
- evaluate proposals based on utility: Respond whether to accept or propose new offer
- generate new offers if no agreement is reached
- continue to evaluate offers until instructed to stop by the initiator

3.3.2 Proposed negotiation protocol

Each meeting has an initiator agent (N) and set of potential participants $(Pset)$. A participant can be an essential participant (EP), a quorum participant (QP), and observer (OP), and so on. The $Pstatus$ vector specifies the type of participation of each participant. The initiator starts the negotiations by posting to a shared blackboard a meeting notification including an agenda and an initial proposal $(O^{pro})_N$ for meeting time, and location. The initiator then waits for responses from other agents, and then checks if a consensus has already been formed by calling *InitiatorCheck*. The pseudocode for meeting initiation is shown below.

On InitiatorAgent side

```
InitMeeting
Purpose: Agent N initiate the meeting, make an initial proposal
Input:   (O^pro, Δ)_i
Output:  Offer proposed by initiator
KL_mj <—— KnownLocation(x_i, x_{i+1}, x_{i+2}, ...x_{i+n})   //set known location for every agent
M_j <—— Meeting(Pset, Pstatus, Quorum, Agenda)   //set meeting properties
R_Max <—— MaxNegotiationRounds(Mj)   //set number of negotiation rounds
Olist <—— OfferList {O^pro_n, O^pro_{n+1}}_Mj   //list of incoming offers from agents (for next rounds use)
(R = 0)   //to indicate initial round started
(O^pro)_N =AgentGenerate( )   //Generate proposals
BlackBoard.post(O^pro)_N   //post proposal from initiator's to BB
Wait-for-respones   //wait till other agents respond
If Olist != {ϕ}   //to check if there are offers in Olist
    InitiatorCheck({Olist})   //let's check if we are done
else
   R = R + 1   //go for another round
```

Once potential participants are notified about the meeting, they respond either accepting the proposed offer by sending an offer identical to the proposed offer or generating a counter offer. The decision whether to accept or reject an offer is based on the utility assessment function. In rounds following the first round, each agent reads offers from all other agents. If the offer of highest utility to the agent exceeds the current resistance level λ, the offer is accepted otherwise the agent generates a counter offer. The value of the resistance level decreases in each negotiation round by a concession rate Δ, which starts large and gradually becomes smaller. Each agent repeats the following routine in each round of negotiation.

AgentNegotiate
Purpose: Agent to read and evaluates offers in round R
Input: $\forall i, get(O^{pro})_i$ from the BlackBoard
Output: Accept or generate new offer

$E^{pro}{}_{\max} \longleftarrow Max_i[Utility\,(O^{pro})_i]$ // offer for agent $_i$ that maximize agents utility

$NA \longleftarrow \arg Max_i[Utility\,(O^{pro})_i]$ // NA: Agent that made the best offer

If $(E^{pro}{}_{\max}) > \lambda_R)$ // overcomes resistance

$(O^{New})_i = (O^{pro})_{NA}$ // Accept NA's offer

Else

$(O^{New})_i =$ **AgentGenerate** $(O^{pro})_{NA}$ //Regenerate new offer that can maximize utility

BlackBoard.post $(O^{New})_i$ // post new offer

$\lambda_{R+1} = \lambda_R - \Delta_R$ //Decrease resistance level Δ_R for next round

Wait until other agents have posted their new offers

Offer generation is a crucial piece of the negotiation process. Initially, agents make offers that are most suitable to them. In each round, the offer generated by an agent is the agent's response to the best offer received in the previous round $(O^{pro})_{NA}$ proposed by agent NA. As negotiation progresses, an agent may insist on its previous offers, try to compromise, or simply concede to another agent.

An agent would insist on its previous offer if it cannot generate an offer that is both acceptable to itself and closer to its most recent offer. Acceptability to self is determined by the agent's current resistance level λ_R. The agent's ability to generate a compromise offer may also be restricted by problem constraints. For example, if both $NA's$ offer and the agent's own offer agree on all the details except the location and there isn't a compromise location that can be used for the meeting then the agent would insist on its offer. In subsequent round a more attractive offer may be generated by another agent or the resistance level λ would have gone down to a level that makes other offers acceptable. An agent would make a compromise offer with respect to $(O^{pro})_{NA}$ if it perceives that NA already considers the agent's previous offer as the best offer it received but could not accept it. The agent would then consider the differences between its previous offer and $(O^{pro})_{NA}$ to generate a compromise offer. For example, the compromise offer could be simply obtained by trying to meet NA halfway.

An agent concedes if it perceives that NA is forming an agreement with another agent. At this point, the agent tries to lure NA by making an offer as close as possible to NA's first offer (the first offer by an agent is considered its most desirable). Such an offer should still be acceptable to the agent and more attractive to the agent than the current second best offer. It should also be more attractive to NA than the agreement it was entering into. The following routine outlines the agent offer generate process.

AgentGenerate

Purpose: Agent $_i$ generates offer in round R

Input: $(O^{pro})_{NA}$

Output: new offer $(O^{New})_i$

If ($R = 0$) // first round
$(O^{new})_i = (O^{best})_i$
else
Find the agent NA_j whose offer is closest to $(O^{pro})_i$
If (NA_j = i) // $Agent_i(me)$ made $NA's$ best offer
then
$(O^{new})_i$ = Compromise($(O^{pro})_i, (O^{pro})_{NA}$) // let's meet halfway- can fail
else
$(O^{new})_i$ = Concede($(O^{best})_{NA}$ // Make best possible offer to NA- can fail
If $(O^{new})_i$ = fail // Either compromise or concede failed
then
$(O^{new})_i = (O^{pro})_i$ // insist by proposing previous offer again
$(O^{pro})_i = (O^{new})_i$

return $(O^{new})_i$

The initiator monitors the progress of the negotiations and decides after each round whether negotiations should continue. If the minimum requirements to hold the meeting have been met (e.g. quorum and all essential participants agreed on a meeting), the initiator stops the negotiation and announces that the meeting has been scheduled. The initiator would cancel the meeting if the maximum number of negotiation rounds has been reached without reaching an agreement that will allow the meeting to take place.

The initiator also checks if the negotiation got stalled. The negotiations get stalled if disconnected clusters are formed such that each agent finds its NA within the same cluster. In such cases, each cluster converges on a meeting scheduling choice different from the other clusters. If this happens, the initiator starts new level of negotiations that includes one representative agent from each cluster.

InitiatorCheck
Purpose: Monitor the progress of negotiations
Input: List of offers at the end of round R
Output: Decision: continue, cancel, reset, or done

If $(R \leq R_{Max})$ // rounds < maximum rounds specified by initiator
then
Group identical offers together forming G_1 to G_m
For every G_i in $\{G_1, \ldots\ldots., G_m\}$ //to scan all groups in G_i
If meeting requirements are met for G_i //check if essential participants EP and quorum are in G_i
BlackBoard.post $(Meeting\,(m_j), Sched\,(O^{pro})_{Gi})$ // announce meeting scheduled on BB
$Exit$ //Stop negotiation, no further negotiation for Mj
else
For each group in G_i
$NA(G_i) = \{x \mid x = NA_y, y \in G_i\}$ //form the set of best offers for each G_i
If (forAll G_i $\{G_i \cup NA(G_i)\} = G_i$) // to check if G_i located as $NA(G_i)$
then //Negotiation stalled
Randomly select from each G_i Agent A_i
InitMeeting for A_i's Essential participants // let initiator to setup EP
else
$R = R + 1$ //negotiate for one more round
else
Cancel meeting Mj //cancel meeting since negotiation rounds < max and EP can not make it

The last condition that needs to be checked for is oscillation which occurs when agents A and B try to compromise with each other but cannot find an appropriate compromise solution. A generates an offer as close as possible to B's first offer and B does the same. In the following round each agent accepts the offers made in the previous round but no agreement is reached. To remedy this problem, an agent who wants to concede must first flip a coin and thus concede with a 50% probability.

3.3.3 Example

Assume the initiator agent would like to organize/negotiate a meeting place between 4-agents A_1, A_2, A_3 and A_4 whose locations are shown in Table 3.1

Table 3. 1: Agent's original locations

Agent	Original location before start negotiation
A_1	$(O^{pro})_{A1} \longleftarrow Detroit$
A_2	$(O^{pro})_{A2} \longleftarrow Hamilton$
A_3	$(O^{pro})_{A3} \longleftarrow Toronto$
A_4	$(O^{pro})_{A4} \longleftarrow Barrie$

Their aim is to find a location to meet. Applying the proposed negotiation protocol discussed earlier would result in following as negotiation rounds:

Round = 1, $\lambda_1 = \lambda_0 - \Delta_1$ **,** at this round the blackboard looks as follows

At first round agents try to propose compromise offer for their NA

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A2}\} : London$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : Oakville$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A2}\} : Oakville$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A3}\} : Aurora$

Round = 2, $\lambda_2 = \lambda_1 - \Delta_2$, evaluate previous offers, then propose offers

A1, A4 concede at this round to lure A2, A3 respectively

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A2}\} : Hamilton$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : Oakville$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A2}\} : Oakville \quad A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A3}\} : Toronto$

Round = 3, $\lambda_3 = \lambda_2 - \Delta_3$

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A2}\} : Hamilton \quad A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : Hamilton$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A2}\} : Toronto \quad A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A3}\} : Toronto$

Round = 4, $\lambda_4 = \lambda_3 - \Delta_4$

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A2}\} : Hamilton \quad A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : Hamilton$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A2}\} : Toronto \quad A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A3}\} : Toronto$

Group identical offers together forming G_1 to G_m

Round = 5, $\lambda_5 = \lambda_4 - \Delta_5$

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A2}\} : Oakville \quad A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : Oakville$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A2}\} : Oakville \quad A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A3}\} : Oakville$

At end of this round all agents agreed on same location

At end of this round, performing *InitCheck* would terminate the negotiations to announce reaching an agreement for a meeting location in Oakville. Final travel distances for the agents would be:

$A_1 \longleftarrow Travel\_Dist\{Detroit, Oakville\}$: 333 km

$A_2 \longleftarrow Travel\_Dist\{Hamilton, Oakville\}$: 36 km

$A_3 \longleftarrow Travel\_Dist\{Toronto, Oakville\}$: 40 km

$A_4 \longleftarrow Travel\_Dist\{Barrie, Oakville\}$: 114 km

Once the agents agreed on a meeting location they negotiate a time to meet. Assuming that the specific time is considered for all agents to negotiate but the day of the meeting is set by the initiator, for the same agents $A_1 - A_4$ Table 3.2 gives the original proposals

Table 3.2: Agent's original proposals for time

Agent	Agent's time proposals
A_1	$A_2 \longleftarrow 17:30$
A_2	$A_2 \longleftarrow 23:00$
A_3	$A_3 \longleftarrow 18:20$
A_4	$A_4 \longleftarrow 6:00$

$(O^{pro})_{NA}$ for agent i would be the closest time for agent $i's$ proposal.

Round = 1, $\lambda_1 = \lambda_0 - \Delta_1$, agents determine their best $(O^{pro})_{NA}$

At first round agents try to propose a compromise offer for their $NA's$

$A_1 \longleftarrow \{(O^{pro})_{A1},(O^{pro})_{A3}\}:18:00$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2},(O^{pro})_{A3}\}:21:20$

$A_3 \longleftarrow \{(O^{pro})_{A3},(O^{pro})_{A1}\}:18:00$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4},(O^{pro})_{A1}\}:11:30$

Round = 2, $\lambda_2 = \lambda_1 - \Delta_2$,

$A_1 \longleftarrow \{(O^{pro})_{A1},(O^{pro})_{A3}\}:18:00$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2},(O^{pro})_{A3}\}:18:20$

$A_3 \longleftarrow \{(O^{pro})_{A3},(O^{pro})_{A1}\}:18:00$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4},(O^{pro})_{A1}\}:17:30$

Round = 3, $\lambda_3 = \lambda_2 - \Delta_3$,

$A_1 \longleftarrow \{(O^{pro})_{A1},(O^{pro})_{A3}\}:17:30$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2},(O^{pro})_{A3}\}:18:20$

$A_3 \longleftarrow \{(O^{pro})_{A3},(O^{pro})_{A1}\}:18:20$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4},(O^{pro})_{A1}\}:17:30$

Round = 4, $\lambda_4 = \lambda_3 - \Delta_4$,

$A_1 \longleftarrow \{(O^{pro})_{A1},(O^{pro})_{A3}\}:17:30$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2},(O^{pro})_{A3}\}:18:20$

$A_3 \longleftarrow \{(O^{pro})_{A3},(O^{pro})_{A1}\}:18:20$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4},(O^{pro})_{A1}\}:17:30$

At End of this round, group identical offers together to form G_1 to G_m

(A_1, A_4) and (A_3, A_2), set their original (O^{pro}) to the current time, and start negotiation

Round = 5, $\lambda_5 = \lambda_4 - \Delta_5$,

$A_1 \longleftarrow \{(O^{pro})_{A1}, (O^{pro})_{A3}\} : 18:00$ $\quad$ $A_2 \longleftarrow \{(O^{pro})_{A2}, (O^{pro})_{A3}\} : 18:00$

$A_3 \longleftarrow \{(O^{pro})_{A3}, (O^{pro})_{A1}\} : 18:00$ $\quad$ $A_4 \longleftarrow \{(O^{pro})_{A4}, (O^{pro})_{A1}\} : 18:00$

At end of this round all agents agreed on same time

3.4 PROPERTIES OF THE PROPOSED NEGOTIATION PROTOCOL

Several properties result from the design of the negotiation protocol affecting the outcome. This section analyzes various aspects of the proposed negotiation protocol

3.4.1 Effect of the resistance and concession parameters

The negotiation protocol proposed uses a market model to assess the utility of an offer. Progress in negotiation is controlled by the resistance level λ which starts high and decreases with negotiation rounds. When λ is at its highest level, agents can only generate or accept offers that are locally optimal. To ensure that progress will be made from round to round the concession rate Δ must be applied. Starting with a large Δ and reducing its value seems to work well in allowing negotiation to progress without ending up accepting poor solutions. As mentioned earlier in our experiments we use a Riemann zeta function in the form:

$$\Delta_R = \frac{k}{R^2}$$

The performance of the algorithm depends heavily on the proper setting of Δ, if it is set too low, progress towards the solution is too slow and agents may not be able to change their offers for many rounds. However, setting Δ too high allows agents to accept bad solution after a small number of negotiation rounds. Ideally, the choice of Δ should allow agents to generate at least one offer to the desired effect (conceding or compromising offer) each round.

3.4.2 Privacy-efficiency tradeoff

The negotiation protocol does not require agents to share their personal calendars, their individual utility functions reflecting individual preferences, nor their resistance level and concession rate. If in a certain application, agents agreed to share such information, then the initiator would be able to figure out the outcome of the negotiation and meetings can be scheduled without any negotiations.

3.4.3 Negotiation a mobile meeting

Agents trying to schedule a mobile meeting will follow the same protocol as for stationary meeting. However, to schedule a mobile meeting, the agents will have to negotiate an end location (leave point) as well. For the leave point to be different from the join point, at least some of the agent must have a destination distinct from their original locations.

3.4.4 Negotiation will always give a solution of either

- there is a meeting with known meeting information in case of agreement
- no meeting, meeting is being canceled

Assuming the initiator specifies maximum number of negotiation rounds R_{Max} and agent i negotiates in a current negotiation round (R) and did not reach maximum $R \geq R_{Max}$ then meeting mj is scheduled if

$$(O^{pro})_i = (O^{pro})_j \qquad \forall_{i,j} \text{ in a quorum that includes } EP \text{ in meeting } mj$$

Satisfying this condition means that an agreement has been reached which resulted in schedule a meeting.

Assuming $R = R_{Max}$ and the above condition has not been met then meeting mj is canceled.

3.5 CONCLUDING COMMENTS

This chapter, presents a model that formalizes the meeting scheduling problem using the CRF canonical model, from this formalization we identify the following:

- There are many potential benefits from mapping MS problem to CRF model
- The resulting negotiation protocol is suitable for selecting time and place for meeting
- The proposed protocol has some desirable properties. It converges if an agreement can be reached within the prescribed number of rounds. It also allows agents to tradeoff privacy and efficiency. It can be used to schedule a mobile meeting as well as stationary ones.

CHAPTER 4

IMPLEMENTATION AND RESULTS

The previous chapter proposed negotiation strategies along with a protocol for multi-agent negotiation. Here, we show the benefits of adopting our approach through a sample of negotiation scenarios among agents.

More specifically, this chapter presents two experimental studies to evaluate the proposed negotiation protocol along with the results. The results describe agents' proposals, offers and agreements reached among agents that are negotiating on spatio-temporal objects. We have developed a small environment capable of simulating agents negotiating to find a meeting place and time that best fits everybody.

The chapter starts by introducing the environment that we use, followed by implementation details of the proposed protocol. The last part of the chapter details a negotiation sample followed by a series of experiments along with their results.

4.1 THE ENVIRONMENT

Several agent-based simulation environments for distributed agents exist. Given below is a list of the three multi-agent environments investigated for this study, along with the reasons that deem them unsuitable for the purpose.

- NetLogo[1] is a cross-platform programmable multi-agent modeling environment and is a dialect of the Logo language. NetLogo is a great environment for simulating agent behavior in a small world. However, due to limitations in the visualization capabilities, NetLogo is not suitable for our experiments.
- StarLogo[2] is a programmable modeling environment for exploring the behaviors of decentralized systems such as bird flocks, traffic jams, and ant colonies. It is

especially designed to be used by students. StarLogo is a specialized version of the Logo programming language that uses turtles for graphical objects and patches for creating the world. However both turtles and patches are command-line constructs that must be programmed. Moreover, to implement our protocol, a simplified map is required to create the agents' world and represent their location visually. In starLogo we found that we need to create thousands of patched to represent the simplest map.

As the investigated environments were not suitable for our purpose, we decided to create our own simulation environment that fits the negotiation scenario discussed in Chapter 3. Hence, we developed a small simulation environment capable of running agents. Since we are interested in scheduling a meeting spatially, a simple map of southwestern Ontario was sufficient to test our approach. The map shows major cities and towns in southwestern Ontario and was taken from Google maps[3].

On this map agents are located randomly in main cities, defined as known locations in Chapter 3, for each agent. We calculate distances between agents' locations on the map using pixels distance representation. For instance, if we want to find any distance between two agents and their location we use Euclidean distance formula to find distance between two points. Location coordinates for known cities are provided in a pre-configuration file. Also, the number of agents is also configurable along with all mid-point coordinates (see Figure 4.1).

This is a simplified model/framework that may be combined to cover bigger map locations and actual distance from google maps.

[1,2] Both NetLogo and StarLogo are used for social and science simulations.

[3] Google Maps, www.maps.google.com

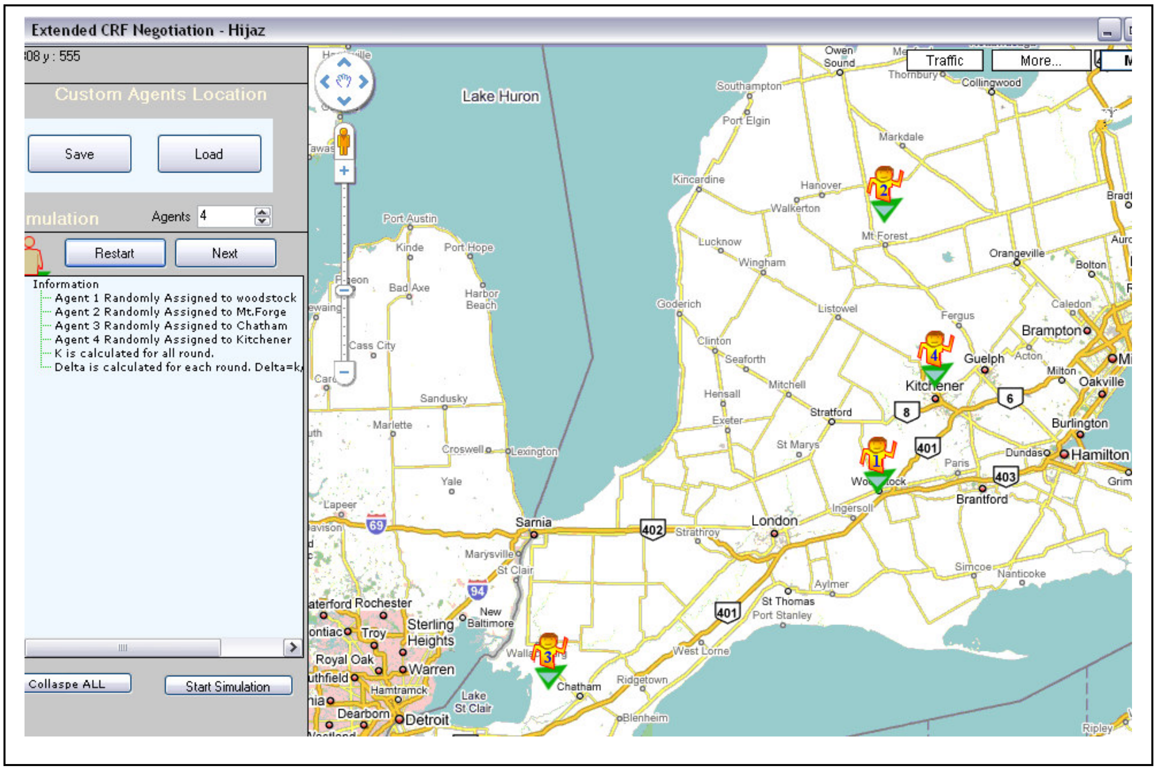


Figure 4.1: Simplified agent environment

4.2 IMPLEMENTATION LAYOUT

We implemented the proposed negotiation protocol using VB.Net language provided by Microsoft Visual Studio 2008 using .Net framework 3.5 with a machine equipped with a Pentium Celeron D processor, having 2.80 GHz with 704 MB of RAM running Windows XP SP2 as operating system. In this section we describe the main classes that we use in implementation.

We have used 4 classes to build the environment and implement the negotiation protocol.

Negotiation

The main class implements the main algorithm and methods used to support the implementation and negotiation. This class inherits Agent since it applies to every agent's negotiation. The class starts by simulating the number of agents then run negotiation protocol for all agents.

Agent is a class that represents agent's properties. In the implementation, agents have location, name, i.d. and proposal point. Every agent generates proposals taken from proposal class. Also, Agent uses city and proposal classes to obtain agents' current proposal along with their locations

Proposal is a class that manages the agents' current and previous proposals and includes agents' name, location and id.

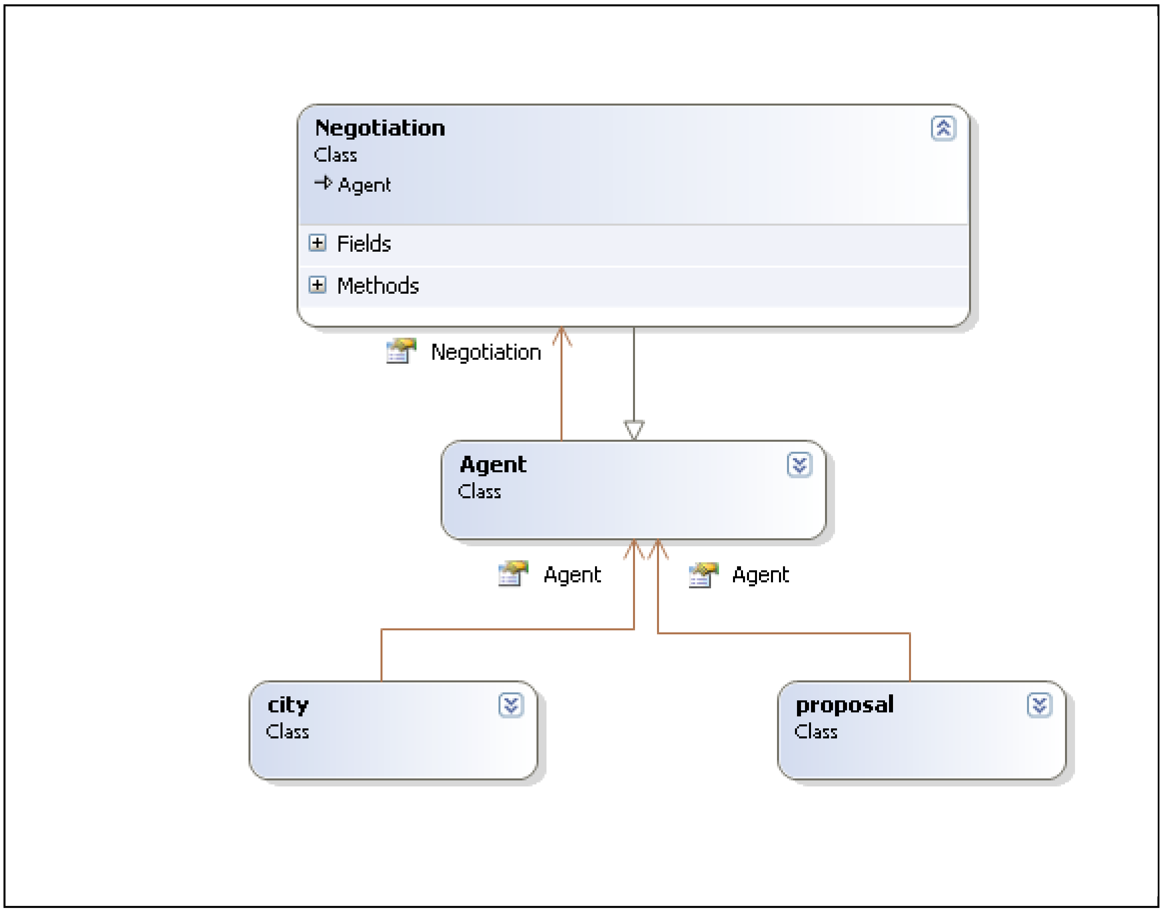


Figure 4.2: Class diagram layout

City is a class used by agents to obtain and locate cities in the simplified map picture based on pixels coordinates.

4.3 EXPERIMENTS

To evaluate our spatial negotiation protocol, a series of experiments have been conducted. For each experiment we describe the parameters, experiment performed and an analysis of the results. Two sets of experimental studies have been carried out; a study of the behavior of the negotiation protocol, and a study of optimality of solution.

4.3.1 Study of convergence behavior of negotiations

Purpose

The purpose of this experiment is to study how the number of negotiation rounds could be affected by the number of agents and the effect of changing the set of known location cities.

Parameters

Parameters that affect the negotiation among agents are the number of agents on one hand, and the number of known locations (cities) by every agent on the other hand. To evaluate negotiation behavior in both parameters we have performed two sets of experiments.

Experiment 1: negotiation behavior in same known locations

In this experiment, agents are distributed randomly in assigned cities on the map. The number of cities is predefined to be 12 cities. The number of agents changes in every run, and we collect agent's logs in every round which includes their previous proposal and current proposal. We determine the end of negotiation, when all agents agree on a meeting place.

In the first set of experiments, we did not change the number of known locations (cities) for all agents. We assumed, the number of known locations is 12 cities in southern Ontario. Agents are randomly distributed among 12 cities, and in every run we change the number of agents. We started from 2 agents up to 10 agents, negotiating on a place to meet, and take average negotiation rounds of 10 runs. Figure 4.3 shows how the number of rounds changes by changing the number of agents.

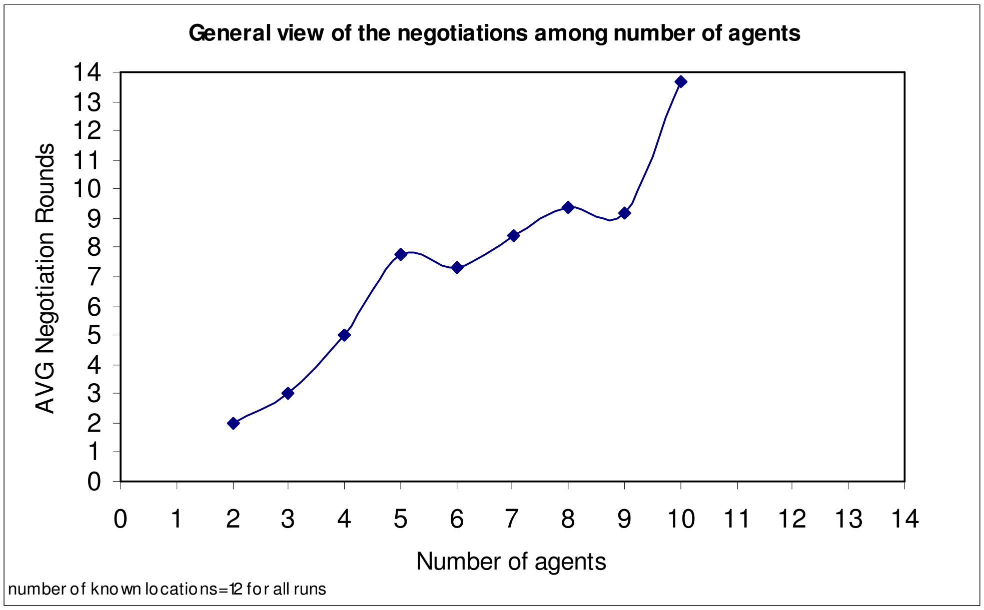


Figure 4.3: Effect of number of agents on negotiation rounds number of cities are the same

As the results in Figure 4.3 show, the number of negotiation rounds between agents increases as we increase the number of agents. This is expected, since they are negotiating on the same number of meeting locations in every run. This makes it harder on 10 agents to negotiate over their original 12 location than 5 agents negotiating on 12 locations. This brings us to conduct a study on changing the number of location which is going to be our second set of experiments.

Experiment 2: negotiation behavior when known locations change

In the second set of experiments, we change the number of known location (cities) from n known location to *n+4,* in every run. Also, we change number of agents from 2-10. We started with two agents on two locations. The experiments were conducted by taking average of negotiation rounds for 10 runs for each set of agents, Figure 4.4 shows how the number of rounds changes with the number of agents and the number of known locations.

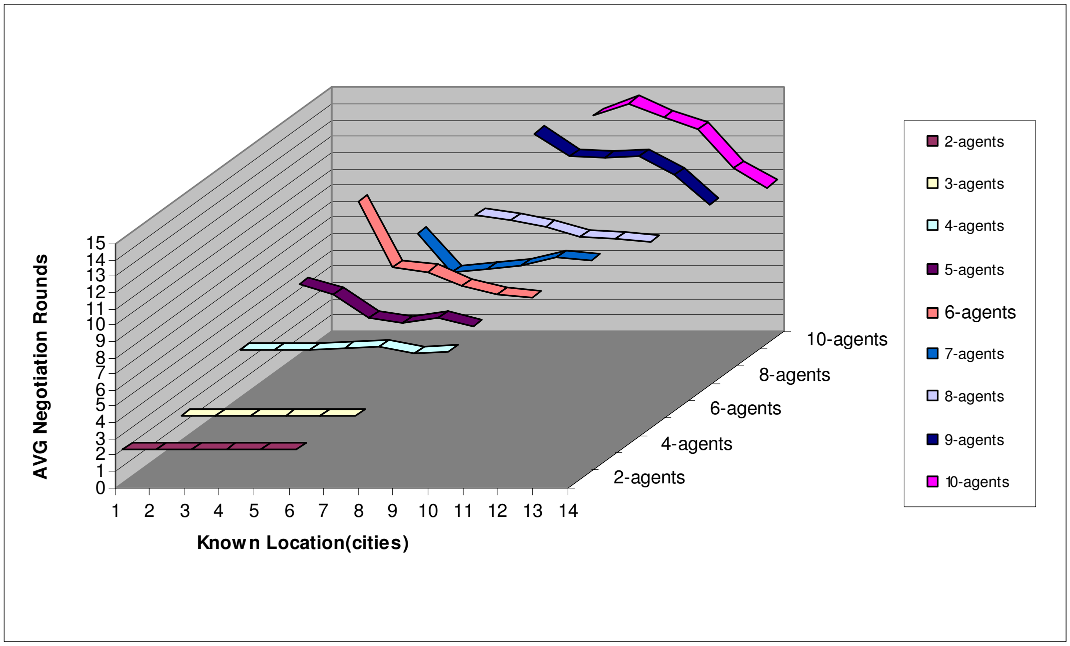


Figure 4.4: Effect number of agents and cities on negotiation rounds

As the results in Figure 4.4 show, we did not notice any change in the number of negotiation rounds in 2-agents and 3-agents cases. As 2-agents negotiate on 2 locations, it requires 2 rounds only; one to propose an offer as mid-point, and the second round is evaluating proposal and reachinging an agreement, which is the same in 3-agents.

Also, Figure 4.4 shows the number of negotiation rounds starts increasing from 2 rounds in 2-agents to 15 rounds in 10-agents. In addition, the notice monotonic decrease then stay constant in negotiation rounds as we increase the number of known locations. This is expected as it gives agents more options for selecting locations. However, in some cases cities were located near the edges of the map, and therefore of no use in the negotiation.

4.3.2 Study of optimality for the negotiation protocol

Purpose

The purpose of this study is to assess the quality of the results obtained for meeting location.

Parameters

The quality of selected meeting places can be measured by taking the travel distance for every agent from its original location to the final meeting location, which resulted from running the negotiation protocol. In addition to the number of agents.

Experiment: quality of the solution when known locations are same

The negotiation protocol was used to find meeting location, by summing up travel distances from agent's location to each possible meeting location and selecting the location with minimum total travel. This gives us the optimal solution for every agent, and then compare results for same scenario with results from our implementation.

The experiments were conducted by changing the number of agents starting with 4-agents, 6-agents and ending up with 10-agents. For each set of agents we run 10 times, and then we manually find the minimum distances and record actual and optimal solution for every run. Figures 4.5, 4.6 and 4.7 show negotiated meeting city for 10 runs

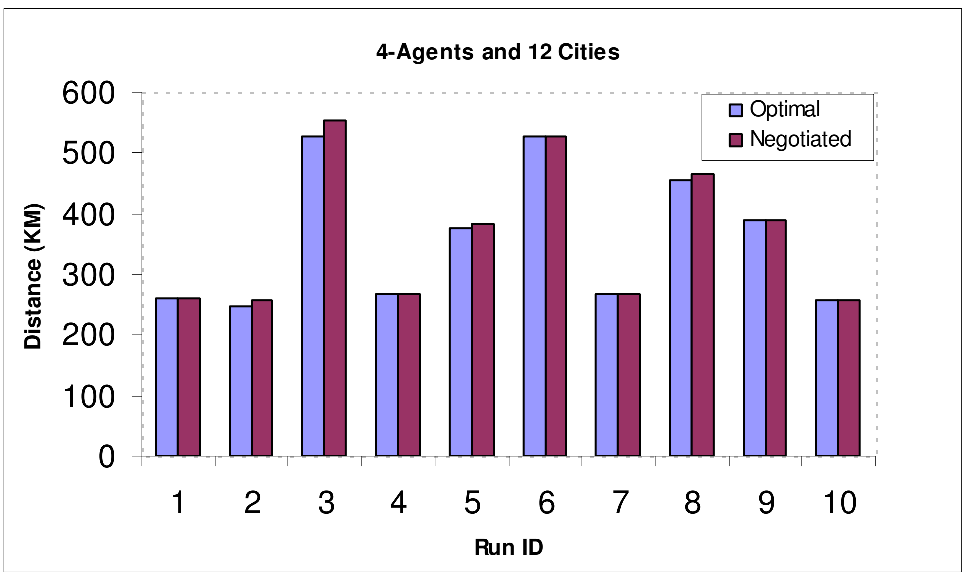


Figure 4.5: Total travel distance to meeting by 4 agents

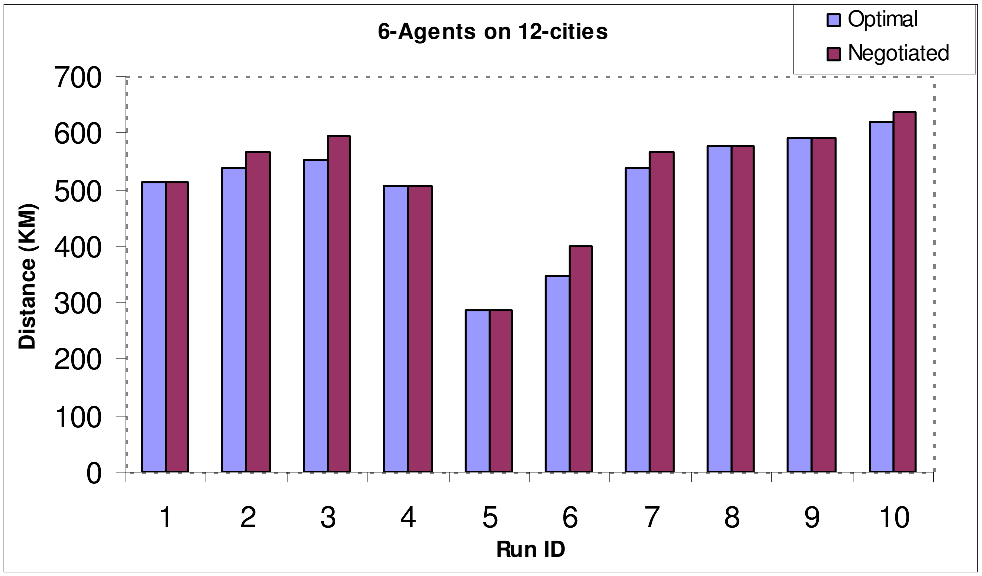


Figure 4.6: Total travel distance to meeting by 6 agents

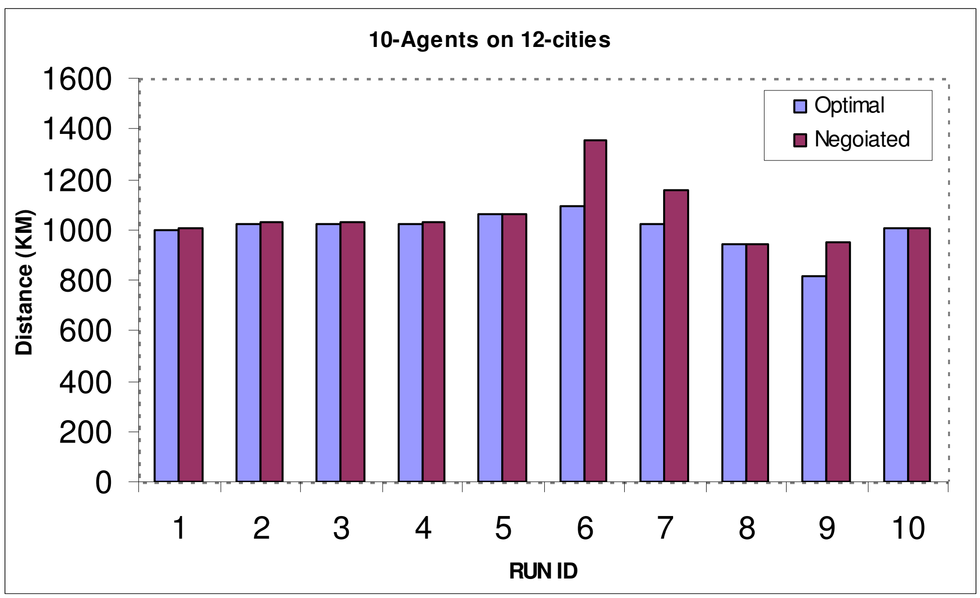


Figure 4.7: Total travel distance to meeting by 10 agents

Figures 4.5, 4.6 and 4.7 show that the negotiated meeting city represents the optimal choice in many runs. The travel distances have changed significantly from one run to the other as a result of changes in the random starting location of agents.

These results were consistent as we changed the number of agents from 4, to 6, and then to 10 as shown in Figures 4.5, 4.6 and 4.7 respectively.

4.4 SUMMARY OF RESULTS

The results show a benefit from formalizing meeting scheduling within CRF model to produce spatiotemporal negotiation among agents.

A negotiation protocol for location selection that deals with spatial issues has been implemented along with its experiments results.

The protocol has been shown to produce near optimal results and converge after a number of rounds that grows linearly with the number of negotiating agents.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

This thesis presents an agent-negotiation model for incorporating spatio-temporal constraints in the Meeting Scheduling problem. The work done includes the following:

- The formalization of the Meeting Scheduling problem as a canonical "Children in the Rectangular Forrest" model. The mapping between the two models formalizes and define spatio-temporal constraints in the Meeting Scheduling model.
- The design of a set of algorithms that define a negotiation protocol suitable for selecting locations and time for proposed meetings based on the formal model defined. The set of algorithms have been shown to always converge. Convergence occur after a number of negotiation rounds that grows linearly with the number of negotiating agents. The algorithms also possess near-optimal performance.
- The identification of a set of properties for the proposed protocol that render it easily extensible to model and implement notions such as mobile meetings and also make it possible to preserve the privacy of the negotiators while retaining an efficient platform for negotiation.
- The design and implementation of a simulation environment on which the negotiation protocol has been implemented and tested for verification and validation on a simplified map. The experiments were designed to test the convergence and optimality of the algorithms for choosing a meeting location for the negotiating agents.

5.2 LIMITATIONS AND LOOSE ENDS

The mapping of the meeting scheduling problem to the CRF model is not yet complete. Three aspects remain outstanding. They are:

- **Managing Conflicts:** The *conflict deal* in meeting scheduling represents the penalty associated with a failure to participate in a meeting. Such penalty is context-dependent and can only be specified by the user. As in some previous work on meeting scheduling (Garrido and Sycara, 1995; BenHassine et. al, 2004; Crawford and Veloso, 2004), the user specifies a utility for a meeting (or meeting type) otherwise the system may be able to learn this utility from history (Zunino and Campo, 2009). The conflict deal is then the loss of the utility associated with the meeting.
- **Oscillation:** In some cases, given two agents A and B attempting to compromise with each other but cannot find an appropriate compromise solution, a situation occurs where A generates an offer as close as possible to B's and B does the same. In the following round, each agent accepts the offers made in the previous round but no agreement is reached. We currently resolve this issue by having the agent wanting to concede flip a coin and therefore concede with a 50% probability, but are looking to find better ways to deal with the situation.
- **Temporal constraints** some aspect of temporal constraints have not been fully integrated in the negotiation protocol. For example, the negotiation protocol does not deal with proposals that include intervals or that tie time and space together.

5.3 RECOMMENDATIONS FOR FUTURE RESEARCH

- **Mobile Meetings:** The model we have presented enables the conception of a new class of meetings. These are meetings that can end in different locations, or in other words, *mobile meetings.* In this case, the meeting scheduling problem becomes a generalization that captures useful aspects of some other problems like the car pooling problem (Burmeister et. al, 1997) or the flight crew scheduling (Castro and Oliveira, 2005). As future endeavor, it would be useful to apply the proposed negotiation protocol to these problems.

Moreover, integrating mobile meetings in a meeting scheduling system may allow users to become more efficient by holding meetings on their way to the other destinations as appropriate. Due to its flexibility, the model we have presented is easily extensible to define such concept. This is done by adding a *leave location* to the already existing set of meeting attributes. This feature remains to be implemented and further investigated.

- **Dealing with Multiple Issues:** For treating multiple issues (e.g. two or more types of constraints), the current models assumes a sequential approach, which is known to be suboptimal. Generating Pareto optimal solutions to the problem requires further investigation.

BIBLIOGRAPHY

[Abdennadher and Schlenker, 1999] Abdennadher S, Schlenker H (1999) *Nurse scheduling using constraint logic programming*. In: Proceedings of (IAAI), pp 838–843.

[Bakker *et al.,* 1993] Bakker R, Dikker F, Tempelman F, Wongum P (1993) *Diagnosing and solving over-determined constraint satisfaction problems*, In: Proceedings of 13th International Joint Conference on Artificial Intelligence (IJCAI), pp. 276-281.

[BenHassine *et al.,* 2006] BenHassine A, Ho T, Ito T (2006) *Meeting scheduling solver enhancement with local consistency reinforcement*, In the International Journal of Applied Intelligence, 24(2) Springer, pp. 143-154.

[BenHassine *et al.,* 2004] BenHassine A, Ho T, Ito T (2004) *Agent-Based Approach to Dynamic Meeting Scheduling Problems*. In: Proceedings of 3rd International ACM Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS), pp. 1132-1139.

[BenHassine, 2005] BenHassine A (2005) Distributed Reinforcement of local Consistency for General Constraints Network An Investigation of Meeting Scheduling Problems, Ph.D thesis, Japan Advanced Institute of Science and Technology.

[Berres and Oliveira, 2005] Berres R and Oliveira E (2000) *Automated Distributed Meeting Scheduler*, Faculade de Engenharia, Universidade do Oorto, NIAD &R-LIACC, Rua dos Bragas, 4099 Porto Codex, Portugal.

[Berger *et al.,* 2005] Berger F, Klein R and Nussbaum D (2008) *A meeting scheduling problem respecting time and space*. In Algorithmic Aspects in Information and Management (AAIM), 5034, pp. 50-59.

[Bowling *et al.,* 2004] Bowling, M, Browning, B, Veloso, M (2004) *Plays as effective multiagent plans enabling opponent-adaptive play selection*. In Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04).

[Burmeister *et al.,* 1997] Burmeister B, Haddadi, A and Matylis, G (1997) *Application of multi-agent systems in traffic and transportation*. In IEE proceedings on Software Engineering, 144,pp. 51-60.

[Castro and Oliveira, 2005] Castro, A and Oliveira, E (2005) *A multi-agent system for intelligent monitoring of airline operations*. In proceedings of the 3rd European Workshop on Multi-agent Systems, pp. 91-102.

[Chithambaram and Miller, 2005] Chithambaram, N and Miller, C.A (2005) *Meeting Location Determination using Spatio-Semantic Modeling*. United States Patent 6,865,538

[Chun *et al.,* 2003] Chun A, Wai H, Wong R (2003) *Optimizing agent-based meeting scheduling through preference estimation,* Engineering Applications of Artificial Intelligence, 16(7) pp. 727-743.

[Crawford and Veloso, 2006] Crawford E and Veloso M (2006) *An experts approach to strategy selection in multiagent meeting scheduling*. In AAMAS, 15(1), pp.5-28.

[Crawford and Veloso, 2004] Crawford E, Veloso M (2004) *Mechanism design for multi-agent meeting scheduling, including time preferences, availability, and value of presence*. In: Proceedings of International Conference on Intelligent Agent Technology, pp 253 – 259.

[Davis and Patterson, 1975] Davis E, Patterson J (1975) *A comparison of heuristic and optimum solution in resource-constrained project scheduling*, Management Science 21: pp.994-955.

[de Farias and Megiddo, 2005] de Farias, D., Megiddo, N (2005) *Exploration-exploitation tradeoffs for experts algorithms in reactive environments*. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17, pp. 409–416.

[Dumond and Mabert, 1988] Dumond J, Mabert V (1988) *Evaluating projects scheduling and due date assignment procedures: An experimental analysis*, Management Science 34: pp101-118.

[Eaton *et al.,* 1998] Eaton P, Freuder E, Wallace R (1998) *Constraints and Agents Confronting ignorance*. AI magazine, 19(2), pp 51-65.

[Ephrati *et al.,* 1994] Ephrati E, Zlotkin G, Rosenschein J (1994) *A Non-manipulable Meeting Scheduling System,* In 13th International Distributed Artificial Intelligence Workshop, pp.105-125.

[Fatima *et al.,* 2006] Fatima S, Wooldridge M and Jennings N (2006)*Multi-issue negotiation with deadlines*. In journal of Artificial Intelligence (JAIR) 26,pp 381-417.

[Fatima *et al.,* 2002] Fatima S, Wooldridge M and Jennings N (2002) *Multi-issue negotiation under time constrain***.** In Proceedings of the 1st international joint conference on Autonomous agents and multiagent systems, part 1, pp 143 – 150.

[Fatima *et al.,* 2001] Fatima S, Wooldridge M and Jennings N (2001) *Optimal negotiation strategies for agents with incomplete information*. In the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL), pp. 53-68.

[Ferreira and Bazzan, 2005] Ferreira J, Bazzan A (2005) *Distributed meeting schedule through cooperative mediation*. In: Proceedings of 6th International Workshop on Distributed Constraints Reasoning.

[Franzin *et al.,* 2004] Franzin M, Rossi F, Freuder E , Wallace R (2004) *Multi-Agent Constraint Systems with Preferences: Efficiency, Solution Quality, and Privacy Loss*, Computational Intelligence 20 (2), pp. 264-286.

[Freuder *et al.,* 2001] Freuder E, Minca M, Wallace R (2001) *Privacy/efficiency tradeoffs in distributed meeting scheduling by constraint-based agents.* In: Proceedings of (IJCAI) Workshop on Distributed Constraint Reasoning, pp.63-71.

[Freuder and Wallace, 1992] Freuder E, Wallace R (1992) *Partial constraint satisfaction*, In: Proceedings of the 11th International Joint Conference on Artificial Intelligence. 58 (1-3). pp.21-70.

[Fruhwirth and Abdennadher,1997] Fruhwirth T and Abdennadher S. (1997) Constraint Programmierung: Grundlagen and Anwendungen. Springer-Verlag.

[Hirayama and Yokoo, 1997] Hirayama K and Yokoo M (1997) *Distributed partial constraint satisfaction problem*. In: G. Smolka (ed.): Principles and Practice of Constraint Programming. pp. 222–236.

[Garrido and Sycara, 1996] Garrido L, Sycara K (1996) *Multi-agent meeting scheduling: Preliminary experimental results*. In: Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS), pp. 95-102

[Garrido *et al.,* 1996] Garrido L, Brena R, Sycara K(1996) *Cognitive Modeling and Group Adaptation in Intelligent Multi-Agent Meeting Scheduling.* In: 1st Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems, pp. 55 – 72.

[Jennings, 2000] Jennings N (2000) *On agent-based software engineering*. Artificial Intelligence, 117(2), pp.277-296.

[Kaufman and Ruvolo, 2006] Kaufman, J.H and Ruvolo, J (2006) Dynamic Resource Scheduling to Optimize Location of Meeting Participants. United States Patent 7,027,995

[Kelly and Chapanis, 1987] Kelly J, Chapanis A (1987) *How professional persons keep their calendars: Implication for computerization*. In Journal of Occupational Psychology, 55, pp.141-156.

[Lemaitre and Verfaillie, 1997] Lemaitre M, Verfaillie G (1997) *An incomplete method for solving distributed valued constraint satisfaction problems*, In: Proceedings of AAAI workshop on constraints and agents, pp. 14-20

[Luo and Boloni, 2008] Luo Y and Boloni L (2008) *Collaborative and competitive scenarios in spatio-temporal negotiation with agents of bounded rationality*. In Proceedings of the 1st International Workshop on Agent-based Complex Automated Negotiations, in conjunction with (AAMAS 08), pp. 40–47

[Luo and Boloni, 2007] Luo, Y and Boloni, L (2007) *Children in the forest: towards a canonical problem of spatio-temporal collaboration*. In Proceedings of the 6th international joint conference on Autonomous Agents and Multiagent Systems, pp. 987 – 993.

[Luo *et al.,* 2000] Luo X, Leung H, Lee J (2000) *A multi-agent framework for meeting scheduling using fuzzy constraints*. In: Proceedings of the 4th International Conference on MultiAgent Systems, pp. 409–410.

[Luo *et al.,* 1992] Luo Q, Hendry P, Buchanan J (1992) *A hybrid algorithm for distributed constrains satisfaction problems.* In: Proceedings of the 9th European workshop on Parallel Computing (EWPC), pp. 375-379

[Maheswaran *et al.,* 2006] Maheswaran R, Pearce J, Bowring E, Varakantham P, Tambe M (2006) *Privacy loss in distributed constraint reasoning: A quantitative framework for* *analysis and its applications*. In: Autonomous Agents and Multi-Agent Systems (JAAMAS), pp 27 – 60.

[Maheswaran *et al.,* 2005] Maheswaran R, Pearce J, Varakantham P, Bowring E, Tambe M (2005) *Valuations of Possible States (VPS): A unifying quantitative framework for analysis of privacy loss in collaboration*. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS05), pp. 1030–1037.

[Maheswaran *et al.,* 2004] Maheswaran R, Tambe M, Bowring E, Pearce J, Varakantham P (2004) *Taking DCOP to the real world: Efficient complete solution for distributed multi-event scheduling*. In: Proceedings of 3rd international Joint Conference on Autonomous Agents and Multiagent Systems, Volume 1, pp. 310-317.

[Mailler and Lesser, 2004] Mailler R and Lesser V (2004) *Solving distributed constraint optimization problems using cooperative mediation*. In: Proceedings of Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 438–445.

[Mailler and Lesser, 2003] Mailler R, Lesser V (2003) *A mediation based protocol for distributed constraint satisfaction*. In: Proceedings of the 4th International Workshop on Distributed Constraint Reasoning, pp. 49-58.

[Matousek, 1994] Matousek J (1994) *on geometric optimization with few violated constraints*. In symposium on Computational Geometry, pp. 312-321

[Marriot and Stuckey, 1998] Marriot K and Stuckey P (1998) *Programming with Constraints: An Introduction*. The MIT Press.

[Meyer, 1997] Meyer H (1997) *ConPlan/SIED Aplan: Personnnel assignment as a problem of hierarchical constraint satisfaction*. In: Proceedings of the 3rd International Conference on the Practical Application of Constraint Technology, pp. 63-92

[Modi *et al.,* 2004] Modi P, Veloso M., Smith S, Cmradar J, Oh J (2004) *A personal assistant agent for calendar management*. In 6th International Workshop on Agent-Oriented Information Systems (AOIS), pp.134-148.

[Modi and Veloso, 2004] Modi P, Veloso M (2004) *Multiagent Meeting Scheduling with Rescheduling*, In: Proceedings of 5th Workshop on Distributed Constraint Reasoning (DCR), pp. 76-92.

[Noronha and Sarma, 1991] Noronha S, Sarma V (1991) *Knowledge-based approaches for scheduling problems: A survey*. In IEEE Transactions on Knowledge and Data Engineering, 3(2): pp. 160-171.

[Russell and Norvig, 2003] Russell S, Norvig P.(2003) *Artificial Intelligence A Modern Approach*. Second edition, chapter 2, Prentice Hall.

[Sanchez and Alonso, 2003] Sanchez M.A. and Alonso A.R. (2003) *Development of personal agenda and distributed meeting scheduler based on jade agents*, Telcom Italia Lab (TILAB) in journal of "EXP in search of innovation"

[Sandholm and Vulkan, 1999] Sandholm T and Vulkan N (1999) *Bargining wih deadlines*. In AAAI/IAAI, pp. 44-51.

[Santos and Vaughn, 2007] Santos, P and Vaughn, H (2007) Where *shall we meet? Proposing optimal locations for meetings*. In INTERACT 2007 Workshop on Map Based Interaction in Social Networks.

[Sciaffino and Amandi 2002] Sciaffino S and Amandi A (2002) *On the design of software secretary*, In proceeding of Argentine Symposium on Artificial Intelligence, pp. 218-230.

[Shakshuki *et al.,* 2007] Shakshuki E, Koo H and Benoit D (2007) *Negotiation Strategies for Agent-based Meeting Scheduling System*. In proceedings of the International Conference on Information Technology, pp. 637-642.

[Sharir and Welzl, 1992] Sharir M and Welzl E (1992) *A Combinational bound for linear programming and related problems*. In STACS, LNCS, vol.57, pp. 569-579.

[Scott *et al.,* 1998]Scott A, Jenkin J, Senjen R (1998) *Design of An-Agent based, Multi-user Scheduling Implementation*. In Multi-agent system-theories Languages and Applications, Lecture Notes in Artificial Intelligence, 1544, pp.152-165.

[Sen *et al.,* 1997] Sen S, Haynes T, Arora N (1997) *Satisfying User Preferences While Negotiating Meetings*. In International Journal of Human-Computer Studies 47: pp. 407-415.

[Sen and Durfee, 1996] Sen S, Durfee E (1996) *A Contracting Model for Flexible Distributed Scheduling*. In Annals of Operations Research, 65: pp.195–222.

[Sen and Durfee, 1995] Sen S, Durfee E (1995) *Unsupervised surrogate agents and search bias change in flexible distributed scheduling*. In proceedings of the first international conference on Multi-Agents System, ICMAS'95, pp 336-343.

[Sen and Durfee, 1994] Sen S, Durfee E (1994) *On the design of an adaptive meeting scheduler*. In: Proceedings of 10th IEEE Conference on Artificial Intelligence Application, pp.40-46.

[Sen and Durfee, 1991] Sen S, Durfee E (1991*) A formal study of distributed meeting scheduling :Preliminary Results,* in Proceeding of the ACM conference on Organization Computing System, pp. 58-68.

[Sycara and Liu, 1994] Sycara K, Liu J (1994) *Distributed meeting scheduling*. In 16th Annual Conference of the Cognitive Society, pp. 538-588.

[Tsuruta and Shintani, 2000] Tsuruta T, Shintani T (2000) *Scheduling meetings using distributed valued constraint satisfaction algorithm*. In Proceedings of the 14th European Conference on Artificial Intelligence-(ECAI 2000), pp.383-387.

[Vidal and 2009] Vidal J, (2009) *Fundamentals of Multiagent Systems*. Chapter6, pp. 98-101.

[Wong *et al.,* 2000] Wong Y, Ho T, Fung K, Chun H (2000) *A model for resources negotiation using mobile agents*. In: proceedings of 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000).

[Wooldridge, 2002] Wooldridge M. (2002) *An Introduction to MultiAgent Systems*. John Wiley.

[Wooldridge and Jennings, 1995] Wooldridge M, Jennings N (1995) *intelligent agent: theory and practice*. The knowledge engineering review, 10(2), pp.115-152.

[Yokoo and Hirayama, 2000] Yokoo M, Hirayama K (2000) *Algorithms for distributed Constraints Satisfaction : A Review.* In: Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS) 3(2), pp.198-212.

[Yokoo and Hirayama, 1998] Yokoo M, Hirayama K (1998) *Distributed Constraint Satisfaction Algorithm for Complex Local Problems.* In: Proceedings of 3rd International Conference on Multiagent Systems (ICMAS-98), pp.372–379

[Yokoo and Hirayama, 1996] Yokoo M, Hirayama K (1996) *Distributed breakout algorithm for solving distributed constraint satisfaction problems*. In: Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS).

[Yokoo, 1995] Yokoo M. (1995) *Asynchronous weak-commitment search for solving distributed constraint satisfaction problems*. In Proceedings of the first International Conference on Principles and Practice of Constraint Programming (CP-95), pp.88–102. Springer.

[Zaki and Pierre 2007] Zaki Y and Pierre S (2007) *Mobile agent in distributed meeting scheduling:A case study for distributed application*, In Intelligent Decision Technologies, vol 1, pp. 71-82.

[Zunino and Campo, 2009] Zunino, A and Campo, M (2009) Chronos: A multi-agent system for distributed automatic meeting scheduling. In Expert Systems with Applications 36, pp. 7011-7018.

[Radar Project, 2004] The Radar Project. www.radar.cs.cmu.edu, 2004.

VITA AUCTORIS

Hijaz Al-Ani was born in 1983 in England, UK. He earned B.Sc. in Computer Science form Al al-Bayt University, Jordan in 2005. He joined the University of Windsor in January 2006 since then he has been working on many aspects of Multi-agent systems. He is currently working in R&D Infrastructure Engineering at Research In Motion in Waterloo, Ontario.