2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops

Meeting Scheduling Assembles Children in the Rectangular Forest

Ahmed Y. Tawfik and Hijaz Al-Ani
School of Computer Science
University of Windsor
Windsor Ontario, Canada
<atawfik|alani>@uwindsor.ca

***Abstract*—This paper 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 paper also examines the implications of the proposed formalization to meeting scheduling negotiations. A protocol for meeting location selection is presented and evaluated using simulations.**

Keywords-meeting scheduling; multiagent negotiation; canonical models.

I. INTRODUCTION

In almost all organizations, scheduling meetings is an important yet iterative and time consuming task. Delegating it to agents enhances efficiency and reduces friction. Multi-agent solutions use negotiation and distributed constraint satisfaction to schedule meetings (e.g., [1, 2, 3, and 4]). Issues of user privacy [5] and learning user scheduling preferences [6] have also attracted attention. However, these studies considered the problem of scheduling a time for a meeting and largely ignored the meeting location selection. The travel time to meeting location has, in some cases, been introduced as a constraint [5].

A simple way to choose a meeting location would be to select a central location or the closest location to the centroid of participants [7]. A linear programming approach for scheduling a meeting time and location frames the problem as an optimization problem [8]. 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. Approaches to select a location that minimizes participants' travel distance or cost have relied on analyzing maps incorporating GPS locations, user activity preferences, and available modes of transportation [9, 10, and 11].

However, these approaches cannot cope with the heterogeneous nature of individual preferences and privacy requirements. Negotiating meeting parameters including its time and location is a more suitable option because each agent acting on behalf of an individual can verify how well a proposal fits the person's own preferences without compromising too much on privacy as calendars are not shared between agents.

Usually, meeting scheduling negotiations are multi-party and multi-issue negotiations. Like any multi-issue negotiation, the parties can use a package deal strategy, a simultaneous negotiation strategy, or a sequential strategy. Package deal negotiators require that each proposal addresses all issues jointly while simultaneous (or parallel) negotiation addresses each issue separately. In sequential negotiation, issues are addressed according to an agreed upon sequence. Package deal negotiations are Pareto optimal but suffer from high complexity as the number of parties and issues increases [12]. Simultaneous negotiation is not suitable for meeting scheduling because of the interaction between various issues. Therefore, we assume here that negotiation will proceed sequentially and that location will be determined before negotiating starting time, and that the agenda will be negotiated before the end time.

This paper formalizes meeting scheduling as a special case of a recently proposed canonical negotiation model for spatiotemporal negotiations known as "Children in the Rectangular Forest" (CRF) [13]. Section 2 introduces the CRF canonical model along with its implications to meeting scheduling. Section 3 presents a negotiation protocol for meeting scheduling negotiation. Section 4 examines the properties of the proposed protocol. Section 5 presents some experimental results obtained from simulating the proposed protocol in a variety of situations. Section 6 concludes the work with summary and future extensions.

II. CHILDREN IN THE RECTANGULAR FOREST

The CRF model has been proposed as an alternative to splitting multiple pies in spatiotemporal negotiations [13]. Cooperating agents are represented by children whose shortest paths to their respective destinations cross a rectangular forest. However, one agent cannot cross the forest alone. Therefore, the agents negotiate a common path across the forest and avoid going around the forest independently. Figure 1 illustrates a simplified example.

In the CRF model, the object of the negotiation is to agree on a join time and a 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

978-0-7695-3801-3/09 $26.00 © 2009 IEEE
DOI 10.1109/WI-IAT.2009.193

edge of the forest reachable by both agents at a certain time forming a Pareto optimal front. An agent would prefer a collaborative deal (to cross the forest with the 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.

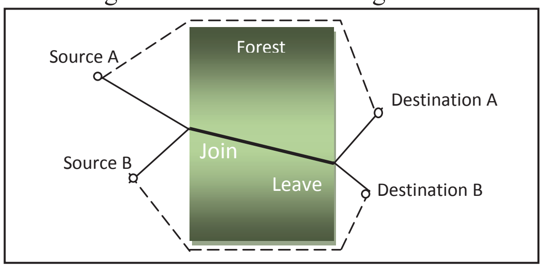

Figure 1. Children A and B cross the forest together (solid lines) or go around it separately (dashed lines).

A. *The CRF model and meeting scheduling*

In many cases, negotiating a meeting schedule involves reaching an agreement on the meeting location, meeting start time, and meeting duration. 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 and all essential participants be 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 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 [1, 14, 15], the user specifies a utility for a meeting (or meeting type) otherwise the system may be able to learn this utility from history [6]. The conflict deal is then the loss of the utility associated with the meeting as perceived by a particular agent.

B. *Mobile meetings*

The last outstanding element in mapping meeting scheduling to the CRF model is the leave location. In most cases, meetings are stationary. However, if meetings are allowed to end in a different location, then a new class of meetings emerges: the mobile meeting. The meeting scheduling problem can then become a generalization that captures useful aspects of some other problems like the car pooling problem [16] and flight crew scheduling [17]. Moreover, integrating mobile meetings in a meeting scheduling system may allow users to become more efficient by holding meetings on their way to other destinations as appropriate.

Normally, an individual participates in many meetings on a given day with some possible commutes between meeting locations. Figure 2 represents an illustrative example. In the figure each continuous line represents one individual, and space has been reduced to a single dimensional space such that a horizontal line represents a stationary individual and a sloping line represents mobility (spatial change over time).

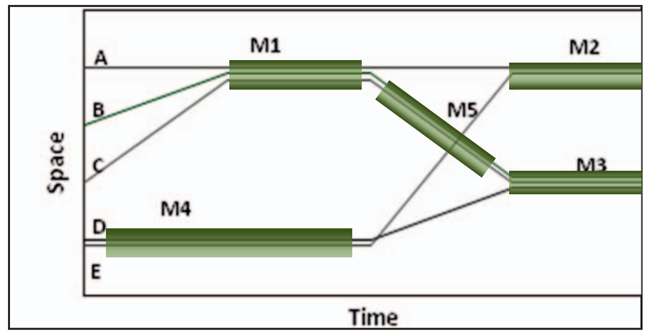

Figure 2. Five meetings involving 5 individuals (A, B, C, D, and E). Meetings M1 to M4 are stationary and M5 is a mobile meeting.

III. THE NEGOTIATION PROTOCOL

Each meeting has an initiator agent '*N'* and a set of potential participants '*Pset*'. A participant can be an essential participant '*EP*', a quorum participant '*QP*', an observer '*OP*', and so on. The *Pstatus* vector specifies the type 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 waits for responses from other agents, and then checks if a consensus has been reached by calling *InitiatorCheck*. The pseudocode for meeting initiation is shown in the box *InitMeeting*.

InitMeeting
Purpose: Agent N initiates the meeting and makes an initial proposal.
$M_j \leftarrow Meeting(Pset, Pstatus, Quorum, Agenda)$
//set meeting properties
$R_{Max} \leftarrow MaxNegotiationRounds(Mj)$
//set number of negotiation rounds
$Olist \leftarrow \{\phi\}$ // list of incoming offers from agents
$R \leftarrow 0$ //initial round
$(O^{pro})_N$ =**AgentGenerate( )** // Generate proposal
BlackBoard.post $(O^{pro})_N$ //post initiator's proposal
Wait-for-responses //wait till other agents respond
If $Olist \neq \{\phi\}$ //If there are offers in *Olist*
InitiatorCheck $(\{Olist\})$
//let's check if we are done
else
$R = R + 1$ // 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 a 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 *AgentNegotiate* procedure in each round of negotiation.

AgentNegotiate
Purpose: Agent evaluates offers in round R
Input: $\forall i, get(O^{pro})_i$ from the Black Board
Output: Accept or generate new offer
$E^{pro}{}_{\max} \leftarrow \underset{i}{Max}[Utility\,(O^{pro})_i]$
// Offer that maximizes the agent's utility
$NA \leftarrow \arg\max_i [Utility\,(O^{pro})_i]$
// *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 by Δ_R
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 *NA*'s 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 a subsequent round a more attractive offer may be generated by another agent or the resistance level λ would have decreased making the 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 $(O^{pro})_{NA}$ and generate a compromise offer. For example, the compromise offer could simply be 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.

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 $(O^{pro})_{NA}$
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 // $NA_j \neq i$ (A better offer was made to NA)
$(O^{new})_i = Concede(O^{best})_{NA}$
// 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 negotiations 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 a new level of negotiations that includes one representative agent from each cluster.

The last condition that needs to be checked for is oscillation. Oscillation occurs when agents A and B try to concede to each other. In such cases, 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 concedes with a 50% probability.

```
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 ≤ R_max  //rounds < maximum rounds specified by initiator
 then
    Group identical offers together forming G_1 to G_m
   for each group G_i in {G_1 , …, G_m}
     if meeting requirements are met for G_i
     //e.g. check for quorum and essential participants
     then
           BlackBoard.post(Meeting(M_j),Sched(O^pro)_Gi)
             //Announce meeting scheduled
          Stop  negotiation // no further negotiations for M_j
          exit
     else
          // check if negotiations stalled
          for each group G_i
             NA( G_i ) = {x | x = NA_y and y ∈ G_i}
            // form the set of best offerers for each G_i
          If  forall G_i {G_i U NA( G_i )} = G_i )
              then //Negotiations stalled
              Randomly select from each G_i , agent A_i
               InitMeeting for A_i's essential participants
               else
                R= R+1      //Negotiate for one more round
else  Cancel meeting M_j
      //cancel meeting since rounds limit has been exceeded.
```

IV. PROPERTIES OF THE NEGOTIATION PROTOCOL

A. Effect of the resistance and concession parameters

The negotiation protocol proposed in Section III 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. In our experiments we use a Riemann zeta function in the form: $\Delta_R = a_0 / R^2$.

Two considerations influence the value of the constant a0: the initial value for λ, and the maximum number of rounds R_{max}. The summation of the Riemann zeta function converges at infinity to a value close to $a_0 \pi^2/6$. Therefore, if we want λ at a sufficiently large R_{max} to approach 0, then we should set $a_0 = 6 \lambda_0/\pi^2$.

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 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.

B. 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 could be scheduled without any negotiations.

C. Negotiating a mobile meeting

Agents trying to schedule a mobile meeting will follow the same protocol for stationary meetings. 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 agents must have a destination distinct from their original location.

V. EXPERIMENTAL RESULTS

Two sets of simulations have been carried out to study the convergence behavior of the negotiation protocol and assess the optimality of the results. The simulated scenarios focused on a group of physically dispersed agents trying to negotiate a location for a meeting where they are all essential participants. The reason for choosing this scenario is to illustrate that proposed technique caters to meeting location selection, unlike much of the work in meeting scheduling that focuses more on choosing a time for the meeting. Moreover, it is easier to verify the correctness of the solution in spatial negotiation and visualizing the progress of the negotiations on a map.

A. Convergence behavior of negotiations

The purpose of these experiments is to study how the number of negotiation rounds is affected by the number of agents participating in the negotiations and the number of possible meeting places. To study the effect of the number of agents on the negotiations, we have randomly located agents in some of a set of Southern Ontario cities such that each city has one agent. The experiments were conducted by changing the number of agents from 2 to 10, and the average number of negotiation rounds is reported for 10 runs for each set of agents. We have also changed the number of known cities from *n* cities to *n+4* where *n* is the number of agents. Figure 4 shows how the number of rounds changes with the number of agents and the number of known locations. The number of rounds increased from 2 rounds when two agents are negotiating to 15 rounds when 10 agents are negotiating. Adding more cities generally helped in reducing the number of negotiation rounds as it gave agents more options. However, in some cases, the additional cities were located near the edges of the map and therefore of no use in the negotiations.

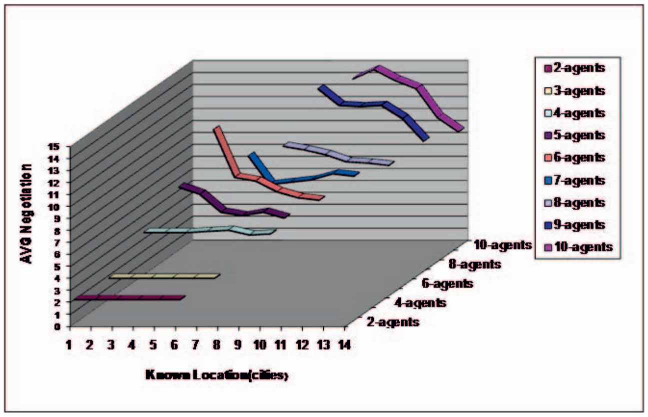

Figure 4. Effect of number of agents and solution options (cities) on negotiation rounds.

B. *Optimality of results*

To assess the quality of the results obtained for meeting location, the utility function of all agents was set as to minimize the distance travelled by the agent to the meeting location. The negotiation protocol was used to find a meeting location and we computed the optimal meeting location by summing up travel distances from agents' locations to each possible meeting location and selecting the location with minimum total travel.

Table 1. Optimality Results

Number of agents	Total distance in kms to optimal meeting location (10 runs)	Total distance in kms to negotiated meeting location (10 runs)	Relative difference
4	9997	10562	5.65%
6	5065	5233	3.32%
10	3570	3626	1.57%

In most runs, the negotiated meeting location was the same as the optimal location. Table 1 shows that the average difference between the negotiated and optimal locations in all runs was small. These results were consistent as we changed the number of agents from 4, to 6, and then 10 while the number of known locations remained constant at 12 possible locations.

VI. Conclusions and Future Research

A negotiation protocol for meeting scheduling that takes into account spatial issues has been introduced. 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.

The formalization of meeting scheduling within the framework of the CRF canonical model led to a generalized model that can deal with different problems like mobile meetings and others.

The treatment has assumed a sequential approach to the multi-issue negotiation problem. This is known to be suboptimal and techniques to generate Pareto optimal solutions to the problem require further investigation.

References

[1] Garrido, L. and Sycara, K A. Multi-agent meeting scheduling: Preliminary experimental results, Proceedings of the First International Conf. on Multi-Agent Systems (ICMAS'95), 1995.

[2] Sen, S. Developing an Automated Distributed Meeting Scheduler. IEEE Expert 12(4): 41-45, 1997.

[3] Modi, P.J. and Veloso, M. Multiagent meeting scheduling with rescheduling, Distributed Constraint Reasoning Workshop, 2004.

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

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

[6] Oh, J. and Smith, S.F. Learning user preferences in distributed calendar scheduling. LNCS. Vol. 3616, pp. 3-16, 2005.

[7] Chithambaram, N. and, Miller, C.A. Meeting Location Determination using Spatio-Semantic Modeling. US Patent 6,865,538, 2005.

[8] Santos, P. and Vaughn, H. Where shall we meet? Proposing optimal locations for meetings. INTERACT 2007 Workshop on Map Based Interaction in Social Networks (MapISNet'07), 2007.

[9] Bingham, G., Martin, C., Meeting Site Selection Based on All-Inclusive Meeting Cost, United States Patent 6,324,517, 2001.

[10] Ambite, J.L, Barish, G., Knoblock, C.A, Muslea, M., Oh, J., Minton, S. Getting from Here to There: Interactive Planning and Agent Execution for Optimizing Travel. Proceedings of of the Fourteenth Conference on Innovative Applications of Artificial Intelligence, AAAI Press and MIT Press, pp. 862–869, 2002.

[11] Kaufman, J.H., and Ruvolo, J. Dynamic Resource Scheduling to Optimize Location of Meeting Participants. US Patent 7,027,995, 2006.

[12] Fatima, S., Wooldridge, M., and Jennings, N. Multi-issue negotiation under time constrain. In Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 143 – 150, 2002.

[13] Luo, Y. and Bölöni, L. 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. AAMAS '07, 2007.

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

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

[16] Burmeister, B. and Haddadi, A. and Matylis, G. Application of multi-agent systems in traffic and transportation, IEE Proceedings on Software Engineering, 144(1):51—60, 1997.

[17] Castro, A., and Oliveira, E. A multi-agent system for intelligent monitoring of airline operations, Proceedings of the Third European Workshop on Multi-Agent Systems, pp. 91--102, 2005.