holder in the system) and negotiates on behalf of this user. The preference information provided by the user is kept private by the agent. The system decomposition knowledge is public to all agents and to the mediator. The negotiation process is conducted in a top-to-bottom order (i.e. starting with the highest level). At each level, there is a meta-negotiation session to determine the negotiation agenda followed by a regular negotiation session to select the preferred *common choices* (CC) based on agents' bids.

Meta-negotiation: determining negotiation agenda

A *decision group* contains a number of (highly) interdependent issues, and a negotiation agenda is a partial ordering of a set of decision groups $\{DG_i\}$. An agenda is represented as a directed acyclic graph. A link from DG_i to DG_j specifies that the negotiation of DG_i should be performed before the negotiation of DG_j. Having such an agenda allows negotiation to be conducted for each DG separately, and reduces the search space for bidding generation and evaluation, since each DG only encompasses a subset of the issues.

The agenda selection process works as follows. First, each agent submits meta-level information about the dependency relationships among all the attributes $\{X_i\}$ at the current level l. The agent infers the dependency relationships from the constraints provided by the user. Attributes involved in the same constraint are considered dependent. The more constraints these attributes share, and the more important these constraints are, the stronger the dependency is. The dependency relationship is computed based on the rank/weight information of the constraints, and is classified as *strong*, *weak* or *none*. More specifically, each agent A_m submits a n by n matrix D_m, where n is the number of attributes at the current level, $D_m[i, j]$ represents the dependency relationship between attributes X_i and X_j for agent A_m. The mediator then creates a global dependency matrix GD based on the matrix D_m submitted by all agents:

$$GD[i, j] = max_{1 \leq m \leq agentNum} D_m[i, j]$$

According to GD, the mediator then clusters all attributes into three types of groups: strongly dependent groups (all attributes inside are strongly dependent); weakly dependent groups (all attributes inside are weakly dependent) and independent attributes. The clustering mechanism is a graph traversal process using depth-first search with GD as the graph adjacent matrix. A *preferred group size limit* parameter can be used to influence group formation.

The mediator then sends the group divisions to all agents, and each agent submits its preferences about the ordering of these decision groups in negotiation (*agenda*). Since the latter issues will be negotiated in the context of the previous negotiation results, the agenda actually affects the negotiation process and thus potentially impacts the negotiation outcome. The decision groups with the biggest potential impact on the outcome utility should be negotiate earlier.

An agent evaluates the impact of group DG_i as:

$$utilityImpact(DG_i) = \sum_{\gamma \in \Delta_i} weight(\gamma),$$

where Δ_i is the set all constraints over DG_i. All decision groups are sorted based on their utility impact in decreasing order (i.e. highest impact group first). A numeric *precedence value* is assigned to each decision group corresponding to its utility impact with normalization. This ordered decision group list is returned to the mediator. Each agent may (very likely) have different views of the importance of those DGs. Though agents could lie about the dependency relationships among attributes and/or their true preferences, it may not be computationally feasible to lie efficiently (Bartholdi, Tovey, and Trick 1989). It is our future work to study how agents can lie to guarantee benefit in such a multi-level clustered negotiation setting, and how to discourage such lying.

Finally, the mediator computes an impact value for each decision group by summing the agent precedence values for each group. A global directed acyclic graph is generated based on both the dependent relationships and the impact values of all decision groups. This is the negotiation agenda for the current level.

Negotiation as a tree search process

With the negotiation agenda available at each level, the mediator conducts a tree search process in the structured agenda space. As illustrated in Figure 1, each node represents a partially specified *state* (partial contract). To expand a node n at depth l, the mediator executes the negotiation agenda at the next level $l + 1$, by requesting bids for each DG according to the specified order in the agenda. The request is accompanied by the bidding context information Γ, as recorded in node n, which describes the attributes in previous decision groups (on the path from the root to node n) with assigned value ranges (r). This represents the restrictions that previous higher-level agreements impose in the current-level negotiation. The mediator also informs the agent the limit of number of bids to submit, BL.

Upon receiving a bidding request for the decision group DG_i and the given context Γ, each agent submits its most preferred BL bids (choices) for all attributes in DG_i. Each bid has an associated preference value, which is the sum of the weights of all constraints satisfied by this bid. Agents use the following *iterative search* procedure to generate bids:

$\Delta \leftarrow$ all constraints over $DG_i \cup \Gamma$;
bid set $\Theta \leftarrow \emptyset$;
$BL \leftarrow$ bids limit provided by the mediator;
while $\Delta \neq \emptyset \wedge |\Theta| < BL$ **do**
  newBids = findBids(Δ);
  for all $B \in$ newBids **do**
    $preference(B) = \sum_{\gamma \in \Delta \wedge Sat.(B,\gamma)} weight(\gamma)$;
  end for
  $\Theta \leftarrow$ newBids $\cup \Theta$;
  $\gamma_{min} \leftarrow arg\ min_{\gamma \in \Delta} weight(\gamma)$
  remove γ_{min} from Δ;
end while

The *findBids(Δ)* currently adopts the max-product algorithm (Marsa-Maestre et al. 2009), using message-passing to solve the constraint satisfaction problem formulated as a maximum weight independent set (MWIS) problem.

The mediator selects the τ top preferred *common choices* (CC) by finding the intersections among the bids submitted by all agents. A valid common choice of bid $B_m = \{X_i = r_{m_i} : 1 \leq i \leq k\}$ and $B_n = \{X_i = r_{n_i} : 1 \leq i \leq k\}$ is

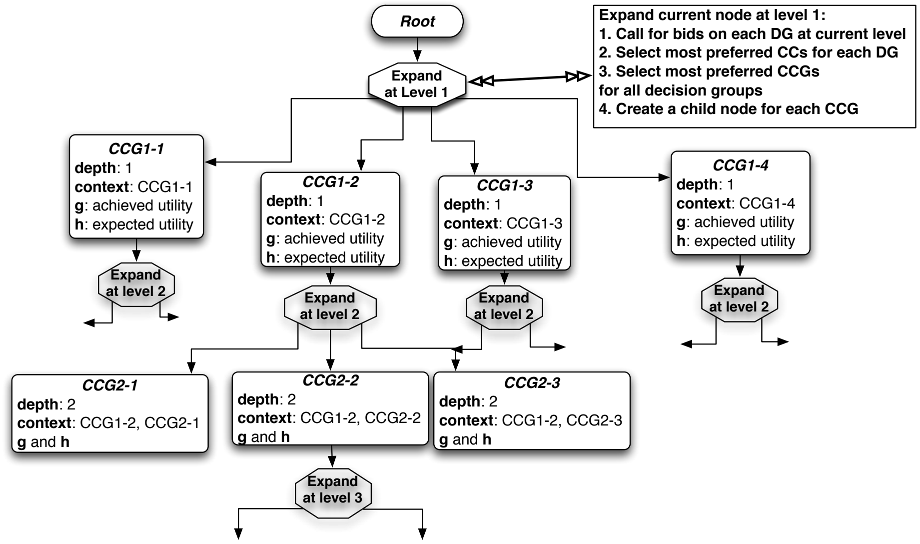

Figure 1: Negotiation Search Tree

the non-empty intersection of these bids, and its preference value is the sum of the preference values of different agents: $CC_{mn} = B_m \cap B_n = \{X_i = r_{m_i} \cap r_{n_i} \neq \emptyset : 1 \leq i \leq k\}$, and its preference value is the sum of the preference values of different agents. A complete search of all possible combinations takes $O(BL^n)$ time for n agents, which is not computationally feasible for large BL and n. To increase the possibility of finding valid combined bids with limited computational cost, the value for the bid limit BL is chosen based on $|DG|$. In the current implementation, L is set as $2^{|DG|}$, bounded by two constant parameters $[minBidsNum, maxBidsNum]$. When no valid bid combination is found, the mediator will double the value of BL and repeat the bid requesting process until BL reaches $maxBidsLimit$ (a constant parameter, value 20 is used currently for hierarchical negotiation).

After the negotiation for all decision groups has been finished, the mediator has a set of ordered preferred common choices $\{CC_{j_1}, CC_{j_2}, ..., CC_{j_\tau}\}$ for each decision group DG_i. A *common choice group* CCG_l is a group of common choices $\{CC_{1_l}, CC_{2_l}, ..., CC_{g_l}\}$, each for one decision group at the current level l. The preference of a CCG is the sum of preference values for all common choices included in this group: $preference(CCG_l) = \sum_{CC_{i_l} \in CCG_l} preference(CC_{i_l})$. The mediator then selects the most preferred χ common choice groups and create a new node for each CCG, and this CCG is the state of the corresponding node and then becomes part of the context of any further search continuing from this node. These new nodes are inserted into the open list.

As the next step, the best node is selected from the open list and examined; if it is not a complete solution (i.e. a solution found at the bottom level where all attributed have been negotiated and decided upon) then the node is expanded. This process is repeated until a complete solution is found. Evaluation function $f(n) = w * g(n) + (1 - w) * h(n)$ is used to evaluate each node n, where w is a parameter with a value between 0 and 1 and $g(n)$ represents the achieved utility of node n. In our negotiation search, it is the preference value of the current context of node n, which measures the sum of weights of all constraints all already considered and satisfied by the context. $h(n)$ is the heuristic function to estimate the expect utility that may be achieved with the current context. It is a challenge to develop a good h function. We have considered the following two approaches:

- $h_a(n)$ is the sum of weights of all un-evaluated constraints that potentially can be satisfied when extends the context CCG in node n. It requires additional computational cost to evaluate all constraints below current level.
- $h_b(n)$ is the sum of weights of all un-evaluated constraints. It's easy to obtain but not very informative.

Using the aforementioned function, we implement an anytime search procedure for the mediator, as follows:

1. Using $w = 1$, meaning $f(n) = w * g(n)$, the mediator conducts a greedy search until a first solution S_1 is found.
2. Update w, $w = 0.5$.
3. Using the solution quality of S_1, $g(S_1)$, to prune the open list, removing all nodes with $f^*(n) = 0.5 * g(n) + 0.5 * h(n)$ less than $g(S_1)$.
4. Continue search until the termination condition is met. Adjust w depending on the size of open list and the search progress to direct the search in either deepening direction (finding a solution quicker) or broadening direction (find a better solution).

This anytime approach allows finding a solution quickly and then continue to find better solutions within available computational resources, which is very helpful for dealing with large-size negotiation problems.

Problem Structure

The way that this hierarchical negotiation approach works is dependent on the problem structure. To better understand the influence of the input problem structure on the performance of this protocol, we defined five parameters to capture the topological and the interdependent characteristic of

Table 1: Scenario Characteristics

name	min	max
# levels	2	8
# attributes	50	100
# constraints per agent	250	500

the problem structure, which is modeled as a tree, where the parent-children relationship represents the decomposition of a component as sub-components:

1. *NumIssues*: number of issues (attributes) in the negotiation. Since all issues have the same domain (integers from 0 to 9), this parameter directly determines problem size. We generated scenarios with 50 and 100 issues.
2. *ShapeBias*: controls the shape of the tree. A bigger value of shape bias produces wider and shallower trees, a small value results in narrower and deeper trees. We generated two different types of trees according to *ShapeBias*, *narrow* (0) and *wide* (10).
3. *WeightBias*: controls how quickly constraint weights decrease with depth. It is assumed that the constraints at each level have different importance, the higher the *WeightBias* is, the more important the higher level constraints are. Two values 0.3 and 0.7 are used.
4. *ScopeProbs*: describes the issue dependency structure. It is expressed as relative frequency of having constraints with different scopes (e.g. involving issues at different levels). Two different settings were used in the current experiments:
   - tight = ((Component 80) (Sibling 10) (Child 10)). For all involved attributes of any constraint created, there is 80% probability that these attributes belong to the same component, 10% probability that they belong to sibling components and 10% probability that they belong to one component and its sub-components.
   - loose = ((Component 50) (Sibling 25) (Child 25)), where there is a higher chance to find constraints which involve attributes belonging to different components and at different levels.
5. *DimProbs*: describes the order dependency, expressed as the relative frequency of constraints with different number dimensions. Two settings were used in the current experiments:
   - low = ((1 50) (2 50)). Half of the constraints involve 1 attribute and the other half involve 1 attributes.
   - high = ((1 20) (2 20) (3 20) (4 20) (5 20)). There is equal chance (20% of probability) for a constraint to have 1, 2, 3, 4, or 5 attributes.

The above parameters describe the topological structure of the system tree, the interdependency between attributes, the complexity of the constraints and the relative importance of constraints at different levels. In the next Section, we study the influence of these parameters on the performance of the hierarchical negotiation mechanisms.

Experimental Studies

Using the five parameters described in previous section with two different categories per parameter, 10 scenarios were generated for each setting, for a total of 320 different testing scenarios, with characteristics measured as in Table 1. For each negotiation scenario, we ran negotiations comparing three different approaches:

1. *Hierarchical Negotiation - Anytime (HNA)*. Negotiation is conducted using our hierarchical structured anytime search with agenda management approach. Multiple solutions may be found within the given time limit and the best one is reported.
2. *Hierarchical Negotiation - Greedy (HNG)*. As above, negotiation is conducted taking advantage of the hierarchical structure, but the mediator conducts a greedy search, which terminates after one solution is found or a given search limit is reached.
3. *Flat Negotiation (Flat)*. All attributes and constraints are put into a single decision group for negotiation. The same heuristic bid generation mechanism based on MWIS with a very large bid limit value is used here, hence the Flat approach cannot guarantee finding the optimal solution. The usage of the same basic mechanism allows the focus on studying the impact of the structured negotiation agenda by comparing the hierarchical approach and traditional one-level approach. We have performed other experiments to evaluate the solution optimality.

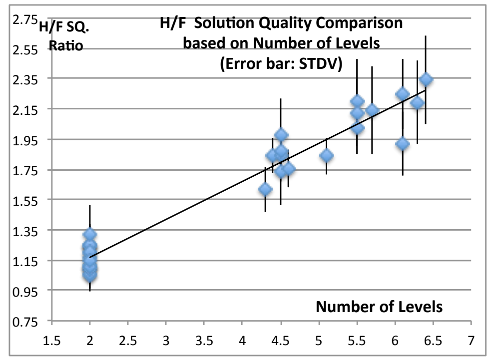

Figure 2: Solution Quality (SQ) Comparison of HNG and Flat based on System Levels; *H/F SQ Ratio = HNG-SQ / Flat-SQ; T-Test(HNG-SQ, Flat-SQ) = 0.00*

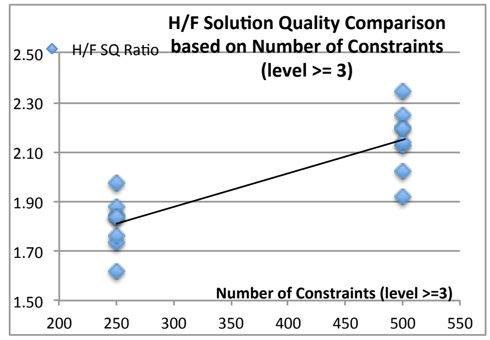

Figure 3: Solution Quality Comparison of HNG and Flat based on #Constraints

The data below shows the results from running each of the three algorithms once on each of 320 scenarios. For each scenario, we record the solution found by each approach and the computational time spent on finding that solution. Each data point shows the average value of the results form 10 different scenarios generated with the same five parameter values. We did not perform large number of repetitions over the same scenario because our approach is primarily deterministic, the only randomness happens when choosing from bids or solutions with the exactly same preference values. We did two repetitions over each scenario and the result are very similar, so only one of them is presented here.

Lesson 1 Learned: *HNG finds better solutions compared to the Flat approach, and its advantage becomes more significant as #levels increases*, supported by Figure 2. This is

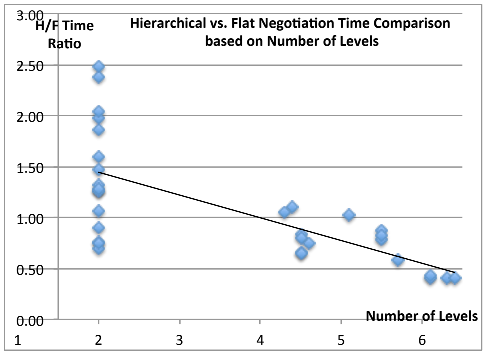

Figure 4: Computation Time Comparison of HNG and Flat

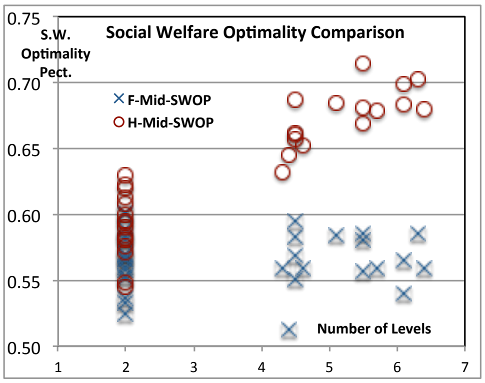

Figure 5: Social Welfare Optimality Percentage Comparison of HNG and Flat (Mid-Point Solution). *T-Test(F-Mid-SWOP, H-Mid-SWOP) = 0.00*

coherent with our expectations, because the hierarchal approach exploits the system hierarchy structure while the flat approach does not.

Lesson 2 Learned: *HNG's advantage over Flat becomes considerably noteworthy as #constraints increases, considering scenarios with more than two levels*, supported by Figure 3. *WeightBias* has much less impact over the algorithms' relative performances than #levels or #constraints. Data is not shown here due to space limitation.

Lesson 3 Learned: *HNG is more efficient, it spends much less time than Flat negotiation as # levels increases*, supported by Figure 4. These results show that our hierarchical negotiation approach greatly improves scalability with #levels and #constraints.

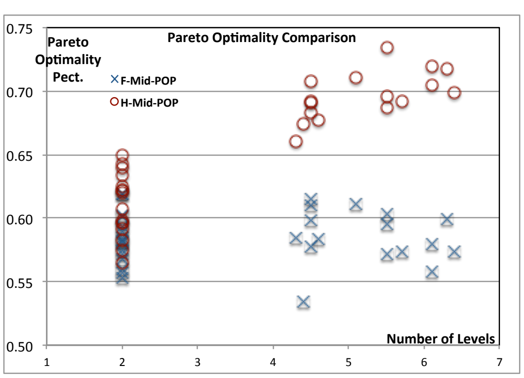

Figure 6: Pareto Optimality Percentage Comparison of HNG and Flat (Mid-Point Solution) *T-Test(F-Mid-POP, H-Mid-POP) = 0.00*

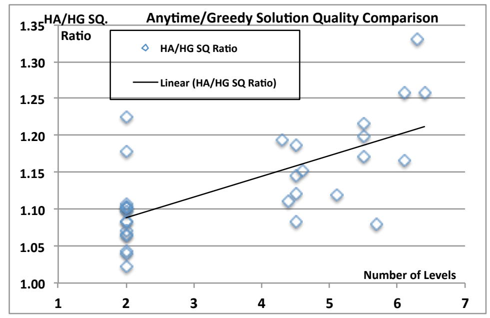

Figure 7: Solution Quality Comparison of HNA and HNG

To investigate the social welfare and Pareto optimality of this hierarchical approach, we used the benchmarking tools provided at Negowiki (Marsa-Maestre et al. 2011) to find the social welfare maximum and the Pareto front for each scenario we tested. We consider these findings are quite reliable because they have been cross verified by different non-linear negotiation mechanisms, and there is no computational limitation applied. Using these findings as benchmark, for each solution found by HNG or Flat approach, we compute its social welfare optimality percentage (SWOP) and Pareto optimality percentage (POP). In order to compute the SWOP and POP, the original range solutions found by HNG and Flat is converted to point solutions by selecting a middle value from each range (referred as mid-point solution).

Lesson 4 Learned: *The solutions found by HNG are superior than Flat both in social welfare optimality and Pareto optimality, and the advantage of HNG becomes more evident as #levels increases*, supported by Figures 5 and 6.

Finally, we compare the HNA approach and the HNG approach. HNA spends much more time (10 times on average) than HNG. Figure 7 shows that HNA (with the quicker heuristic function $h_b(n)$) does find better solutions (by design), and the margin of gain seems also to increase as the number of levels increase. However, this margin of gain is not very significant, thus hardly justifying the remarkable extra time spend by HNA. We tested both $h_a(n)$ and $h_b(n)$ described in Section , neither one seems to be very effective.

Lesson 5 Learned: *In order to improve the performance of HNA, we need to develop more informative heuristics functions that can better predict the expected final quality of a partial solution.* This is a very interesting direction for future research.

Conclusion and Future Work

In this paper we present a hierarchically structured negotiation search process with agenda management based on meta-negotiation. Different subsets of issues are negotiated at each level and the agreements made on the higher levels prunes the search space that has to be considered at lower levels. This approach shows promising results, significantly reducing the computational effort and the potential of finding better negotiation outcomes for complex problems with large number of interdependent attributes. We formally defined a set of parameters to capture the topological and the interdependent characteristic of the problem structure. We have conducted extensive experimental work to study the impact of different scenario parameters on the performance of various negotiation algorithms, and investigated the Pareto efficiency and social welfare optimality using benchmark functions from Negowiki. There are a lot of important and interesting issues that need to be studied in the future, including: investigating the impact of negotiation agenda on negotiation performance, intelligently navigating in the hierarchical search space with better-informed and more efficient heuristic functions, improving the Pareto efficiency, fairness and incentive compatibility of this negotiation protocol, and exploring the possibilities for automatic system decomposition into a hierarchical structure.

References

Abedin, F.; Chao, K.-M.; Godwin, N.; and Arochena, H. 2009. Preference ordering in agenda based multi-issue negotiation for service level agreement. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications Workshops*, WAINA '09, 19–24. Washington, DC, USA: IEEE Computer Society.

Aknine, S. 2011. A multi-agent model for overlapping negotiations. *Group Decision and Negotiation* 1–44.

An, B.; Lesser, V. R.; and Sim, K. M. 2011. Strategic agents for multi-resource negotiation. *Autonomous Agents and Multi-agent Systems* 23:114–153.

Bar-Yam, Y. 1997. *Dynamics of Complex Systems*. Addison-Wesley stydies in nonlinearity. Westview Press.

Bartholdi, J.; Tovey, C.; and Trick, M. 1989. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6(3):227–241.

Bruns, G., and Cortes, M. 2011. A hierarchical approach to service negotiation. In *Web Services (ICWS), 2011 IEEE International Conference on*, 460–467. IEEE.

Dang, J., and Huhns, M. N. 2006. Concurrent multiple-issue negotiation for internet-based services. *IEEE Internet Computing* 10:42–49.

Fatima, S., and Wooldridge, M. 2013. Optimal agendas for sequential negotiations. In *W21-The 15th International Workshop on Agent-Mediated Electronic Commerce (AMEC 2013)*, 59.

Fatima, S.; Wooldridge, M.; and Jennings, N. R. 2004. An agenda based framework for multi-issues negotiation. *Artificial Intelligence Journal* 152(1):1–45.

Hindriks, K.; Jonker, C.; and Visser, W. 2009. Reasoning about multi-attribute preferences. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 1147–1148. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Hujala, T., and Kurttila, M. 2010. Facilitated group decision making in hierarchical contexts. In *Handbook of Group Decision and Negotiation*. Springer. 325–337.

Inderst, R. 2000. Multi-issue bargaining with endogenous agenda. *Games and Economic Behavior* 30(1):64 – 82.

Ito, T.; Hattori, H.; and Klein, M. 2007. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *IJCAI*, volume 7, 1347–1352.

Karaenke, P., and Kirn, S. 2010. Towards model checking & simulation of a multi-tier negotiation protocol for service chains. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, 1559–1560. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Klein, M.; Faratin, P.; Sayama, H.; and Bar-Yam, Y. 2003. Negotiating complex contracts. *Group Decision and Negotiation* 12(2):111–125.

Luncean, L. 2011. An overview on interrelated negotiations. In *System Theory, Control, and Computing (ICSTCC), 2011 15th International Conference on*, 1–6.

Mansour, K.; Kowalczyk, R.; and Wosko, M. 2012. Concurrent negotiation over quality of service. In *Proceedings of the 2012 IEEE Ninth International Conference on Services Computing*, SCC '12, 446–453. Washington, DC, USA: IEEE Computer Society.

Marsa-Maestre, I.; Lopez-Carmona, M. A.; Velasco, J. R.; and de la Hoz, E. 2009. Effective bidding and deal identification for negotiations in highly nonlinear scenarios. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 1057–1064. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Marsa-Maestre, I.; Klein, M.; Hoz, E.; and Lopez-Carmona, M. 2011. Negowiki: A set of community tools for the consistent comparison of negotiation approaches. In Kinny, D.; Hsu, J.-j.; Governatori, G.; and Ghose, A., eds., *Agents in Principle, Agents in Practice*, volume 7047 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 424–435.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V., eds. 2007. *Algorithmic Game Theory*. Cambridge Univ. Press.

Pendergast, W. R. 1990. Managing the negotiation agenda. *Negotiation Journal* 6(2):135–145.

Sim, K. M., and Shi, B. 2010. Concurrent negotiation and coordination for grid resource coallocation. *Trans. Sys. Man Cyber. Part B* 40(3):753–766.

Zhang, X., and Klein, M. 2012. Hierarchical negotiation model for complex problems with large-number of interdependent issues. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 2, 126–133.

Zhang, X., and Lesser, V. 2007. Meta-level coordination for solving negotiation chains in semi-cooperative multi-agent systems. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 50–57. AAMAS-07.

Zheng, R.; Chakraborty, N.; Dai, T.; and Sycara, K. 2013. Multiagent negotiation on multiple issues with incomplete information. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 1279–1280. International Foundation for Autonomous Agents and Multiagent Systems.