Negotiation Protocols for Distributed Nurse Rostering

Ruben Lagatie Stefaan Haspeslagh Patrick De Causmaecker

K.U.Leuven, Department of Computer Science

Abstract

One intricate computational problem still researched today is the Nurse Rostering Problem. Most literature presents different solutions to the local problem, but in recent years the concept of problem decomposition was identified as one way to improve performance. The downside of this is that the quality of separate rosters is much lower than that of a central solution, where personnel can be scheduled in other wards. Some recent papers introduce negotiation as a way to handle crucial problems such as personnel shortages.

This contribution presents a method to use negotiation in solving these problems, but at the same time improve the overall quality. A framework for such a system is presented and five different negotiation protocols are compared.

1 Introduction

One intricate computational problem still researched today is the Nurse Rostering Problem (NRP). The problem can be defined as finding an (near) optimal allocation of nurses to shifts in a hospital. Quality is measured by the number (and severity) of preference and coverage violations, represented by penalties.

Studied for over more than 40 years, the NRP has an extensive literature. Due to its highly constrained nature and large search space, solving a NRP is far from easy. For this reason most attention has been paid to the local NRP. An overview is given in [5].

Recently, some research has been done concerning the decomposition and distribution of the NRP. In contrast to earlier methods, a roster covering the entire hospital is decomposed in smaller rosters limited to a single ward. Each part can then be solved separately and preferably simultaneous.

Problem decomposition, often used in search algorithms, has a number of benefits. If implemented on a distributed system, we increase among others the scalability, flexibility and autonomy [11]. Specifically for the NRP, we gain a higher performance, mainly because smaller rosters are faster to solve.

Even if not distributed, a significant performance gain is achieved on multi-core processors. According to Amdahl's law, parallelization directly influences the performance gain for applications running on these kind of processors. A decomposed NRP is easier to be parallelized and is therefore faster on multi-core computers.

Although much faster, the quality of a decomposed NRP problem is often much lower. Within a central approach, all personnel information is available. If necessary, nurses can be temporarily scheduled for shifts in another ward. The decomposed NRP on the other hand, can have many planners, each responsible for its own ward. Because information is limited to the local problem, personnel shortages or problematic shifts, e.g. shifts requiring a very specific skill, cannot be resolved and introduce high penalties.

We solve this problem by using negotiation. Five different negotiation protocols are compared. In section 2 we briefly review related literature. Section 3 summarizes the model used. In section 4 we show results of the experiments. Section 5 concludes.

2 Related work

As already mentioned, the literature concerning the local NRP is extensive. An overview of the different problem solving methods is given by Burke et al. [5]. Petrovic and Vanden Berghe [15] introduced a number of criteria with which these techniques can be compared. Our focus is mainly on the distributed NRP and more specifically on how negotiation is used in solving personnel shortages.

Kaplansky and Meisels [12] were the first to use negotiation for a distributed NRP. In their method, a Multi-Agent System is set up with two agents. For each ward, a *Scheduling Agent* (SA) is introduced, in charge of its local roster. It is the task of the *Central Resource Agent* (CRA) to endorse global constraints by sending requests to the SAs. Negotiation is used between the SAs before contacting the CRA. In doing so, scalability is improved as a lot of conflicts can be solved without the help of the CRA. This method was initially used to minimize transportation costs, but can easily be adapted to minimize undercoverage.

Another way of coping with personnel shortages is to introduce a pool of float nurses, as described by Bard and Purnomo [3]. Negotiation is used here to determine the most suitable float nurse for shifts that were unable to be assigned.

Di Gaspero et al. [10] use negotiation to exchange roomslots in a Distributed Course Timetabling Problem, which appears to be similar to the Distributed NRP. Each department of the university solves its local Course Timetabling Problem. A regular Vickrey auction is then used to sell so called roomslots, in order to maximize sharing of common resources. They make use of a market based strategy and introduce an artificial currency that is used to buy and sell the resources.

Reeves et al. [16] use bidding strategies in combination with market based rostering algorithms. One of the described strategies is the *Simultaneous Ascending Auction*. They state that combinatorial auctions are often too complex to be used for resource allocation. Reeves et al. also remark that real-world markets typically operate separately and concurrently.

De Causmaecker et al. [8] describe a coordination model for distributed personnel scheduling. They make use of the Contract Net Protocol. We use the same basic model in our system and is further described in section 3.1.

3 Model

3.1 Framework

The framework, also used in [8], is designed as a Multi-Agent System with three types of agents: a single OmbudsAgent, DepartmentAgents and EmployeeAgents. Each DepartmentAgent solves its own local roster. If no local improvements can be found, the agent identifies problematic shifts and a message is sent to the OA. A negotiation protocol is then used to determine the best suited EA for each of these shift.

3.2 Variable Neighbourhood Search (VNS)

We implemented a VNS algorithm as described in [7]. The algorithm consists of two phases. In the first phase a feasible roster is constructed. We start by ordering the shifts based on a number of heuristic rules. For each task, the nurse who's penalty improves most or worsens least is then identified. The heuristic ordering ensures that problematic shifts are assigned first, reducing the chance that the best candidate has already been assigned an overlapping shift. We will use this ordering later on in our negotiation, it appears to be beneficial to single sequential auctions.

In the second phase, this initial roster is iteratively improved by using a quickest decent VNS algorithm with two neighbourhoods. The first neighbourhood consists of all rosters found by changing nurses of single shifts. If there is a roster in this environment with lower penalties, a new neighbourhood is constructed and the algorithm repeats. If no better roster is found, the algorithm constructs a larger environment with rosters constructed by switching the nurses of each two shifts. If a better roster is found, the algorithm starts over by testing the first neighbourhood from this new roster. The algorithm stops when there is no improvement found in either neighbourhoods.

3.3 Objects

For some shifts it might be impossible to find a local nurse, resulting in an infeasible roster. This is either because the required skill is absent or because there is a shortage of personnel. These shifts are the subject of the negotiation.

Another problem is that the quality of a decomposed roster is often much worse than that of a central roster. In a central system the pool of available nurses comprises the entire hospital and is therefore much larger. As a result, shifts that are hard to schedule in the ward can be assigned to an external nurse. To produce a decomposed roster approximating the same quality, highly penalized shifts are also to be negotiated about.

Ideally, we would like to negotiate about the shifts that introduce the highest penalties to the roster. Often however, penalties are raised because of a particular combination of shifts has been assigned. This relation between shifts is still hard to detect and will be addressed in future work.

For now, high penalty shifts are identified by searching the shift that when unassigned, introduces the best overall improvement. When found, it is unassigned and the process repeats. The number of unassigned shifts is limited by a variable percentage of the total number of shifts, 10% resolved most problems in our test cases. If there is still a high penalty, this value can be increased.

Our method does not put a restriction on the nurses able to bid. Nonetheless, our approach can be easily extended to allow for a pool of float nurses. This can be modelled as a virtual ward without coverage constraints. This department has float nurses assigned with specific skills and constraints. If a regular nurse is a better candidate than a float nurse, the shift can be assigned to the regular nurse. In most cases however the float nurses will be able to bid higher and on more shifts, as they initially have no shifts assigned.

By using this method, we do more than trying to make the rosters feasible. We improve overall quality by assigning problematic shifts to external nurses.

3.4 Constraints

The quality of a roster is inversely proportional to the sum of all penalties. Each penalty indicates the violation of a constraint. Our model uses the constraints described in ANROM [4, 6]. The fact that a nurse should not work more than a given number of consecutive days or prefers not to work at the same time as another nurse are just two examples of constraints. ANROM defines 26 constraints.

In some cases, introducing an extra shift can actually lower the penalty, this is called positive synergy and influences the bidding strategy of nurses. Sometimes a low bid is made for a shift, but in the knowledge of also obtaining another shift the bid can be much higher. This correlation is used in some protocols described in section 3.6.

3.5 Bidding

In each of the negotiation protocols described below, bidding is used to determine the best candidate for a problematic shift. Two approaches were investigated. The first was a monetary system, not unlike the one used in [10]. Another approach is to simply bid the improvement of the penalty. Some negotiation protocols, such as the *Simultaneous Ascending Auction*, require the ability to raise a bid. We can use the provisional winners to alter bids for the next rounds.

We assume the second method yields better results as it takes more complex correlations into consideration. Future work should test this assumption.

3.6 Negotiation protocols

Five negotiation protocols have been implemented and tested. The first three are single auctions, the last two are examples of combinatorial auctions. Due to space limitations, we briefly summarize each protocol and note some alterations. For more details, we refer to the MSc thesis of the author.

3.6.1 Contract Net Protocol [17]

The Contract Net Protocol (CNP) starts by sending a *call for bid* message to each nurse for an unassigned task. Each nurse can then reply with a bid. After all bids for have been received, the highest bidder is assigned the task. This is repeated for each task, until all are assigned.

Applying this to the NRP is straightforward. To improve the quality, the tasks are first sorted using the heuristic rules of the VNS algorithm, described in section 3.2.

3.6.2 Extended Contract Net Protocol [1]

The *Extended Contract Net Protocol* (ECNP) allows for changing bids by introducing temporary bidding and temporary accepting. Using this protocol, it is possible to take the provisional winners into consideration and alter previously made bids. Our protocol has been adapted to allow lowering bids and still guarantee finiteness. In contrast to the CNP, the *Extended Contract Net Protocol* considers positive synergy.

3.6.3 Simultaneous Ascending Auction [14]

The *Simultaneous Ascending Auction* (SAA) is another multi-round auction protocol. All bidders send bids for all tasks at the same time. After each round, the bidders receive the current highest bids per task and bidding restarts after setting the new highest bid as the minimum bid.

Again this can be used in combination with the NRP. Bids can change after assigning the tasks to the currently highest bidders. Because lowering a bid should be possible, some alterations were made to avoid infinite loops.

3.6.4 Limited Vickrey [18, 9]

The generalized Vickrey auction is straightforward. It sends bids for all possible combinations of tasks. The server then chooses the best combination of packages. Although this protocol is optimal, it is very inefficient.

Instead of bidding for the combinations of all tasks, we adapted the protocol to bid for the ten task with the highest single bid. It is no longer optimal, but if the computation time is less important than the quality, the number of tasks to consider can be increased.

3.6.5 Ascending Proxy Auction [2]

The *Ascending Proxy Auction* (APA) is similar to the *Simultaneous Ascending Auction*. Bids are sent to the initiator and after each round, the winners are disclosed. This protocol is however a type of combinatorial auction, bids comprise a package and a proposed payment.

Packages are incrementally constructed during the protocol. A participant starts bidding for a package consisting of a single task. If this package is accepted, the bidder adds another task to the package, we call this expanding.

Allowing bidders to send lower bids for expanded packages, again raises some complex problems. Adaptations were made to solve these problems, without loosing the concept of the original protocol.

4 Results

In this section we review some results of the performed tests. First, we describe the used test case. After that a comparison is made between the different negotiation protocols. Then we compare the centralized and the distributed approaches.

4.1 Used problem

Due to space limitations, all the results presented in this section cover a single problem. This NRP is a combination of three separate rosters consisting of 12, 8 and 4 nurses. The used rosters are slight variations of rosters BCV-A.12.1, BCV-1.8.1 and BCV-5.4.1, available in the Roster Booster application of Tim Curtois [1]. Each roster covers a period of one month. Five types of shifts are used and eleven contracts are defined.

Because the *Variable Neighbourhood Search* is indeterministic, we use fifteen solutions for this problem, i.e. fifteen different rosters to negotiate about. We have achieved similar results with other problems, but this should be further tested in future work.

4.2 Negotiation protocols

The criteria we use to compare these different protocols are: quality, computation time, communication time and success rate. We discuss these performance measures in the following subsections.

[1] `http://www.cs.nott.ac.uk/~tec/NRP/`

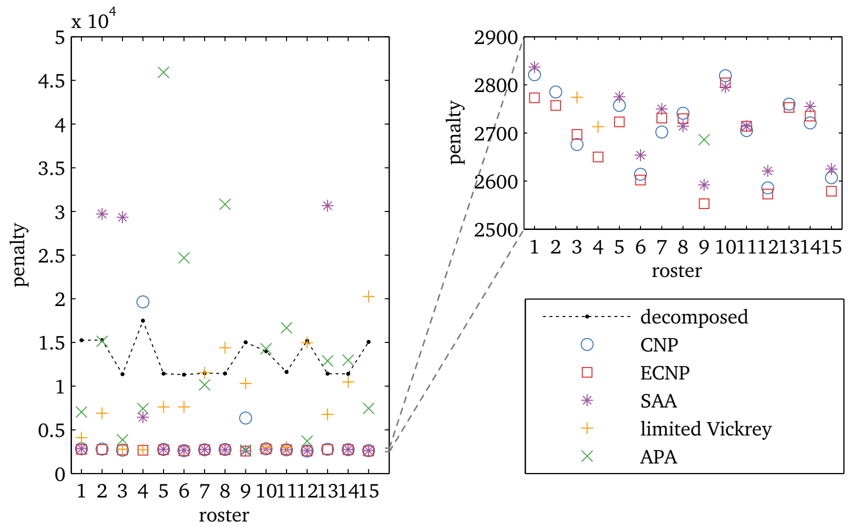

Figure 1: Comparison of protocols on penalty

4.2.1 Quality

Our aim is to use negotiation to improve the quality of the individual rosters after solving each entirely decomposed. In other words, we would like to lower the total sum of all penalties. For each of the 15 solutions to the problem, figure 1 shows the total penalty before (dotted line) and after negotiation.

It is clear that most protocols sometimes return a solution that is worse than the result found by the local VNS algorithms. Unassigned shifts from a nurse are sometimes very specific and no better nurse can be found. If an overlapping shift has been assigned to the scheduled nurse, the negotiation process has to assign the shift to another nurse, possibly resulting in a higher penalty. This can be avoided by reassigning the original nurse. This will be implemented in future work.

Secondly, for this problem, the lowest penalty is about 2550. If we zoom in on this part of the graph, we can see that in most cases three protocols show similar results: the CNP, the ECNP and the SAA. In all cases the ECNP either finds a best solution or one that is close to the best.

It is remarkable that the CNP does not perform much worse than the more complex methods. As expected though, the ECNP improves on this in most cases and where the CNP fails miserably (i.e. in roster 4), the ECNP still manages to find an excellent solution.

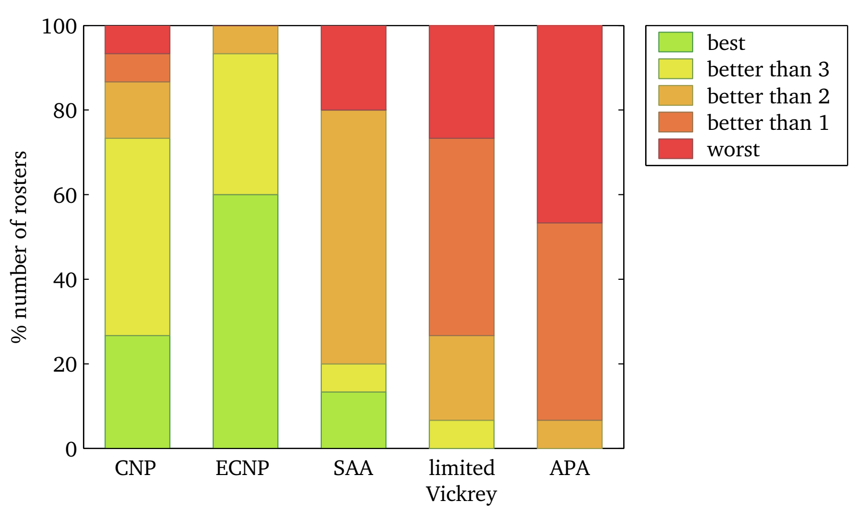

Figure 2: Ranking of protocols

Looking at the combinatorial auctions, we can see the limited Vickrey auction often improves the local rosters, yet it is still not as good as the single sequential auctions. Of course as already mentioned, the protocol took the ten best single bids to send combinations for. If this value is increased, better solutions are found. This would however affect the total time spent, which is, as we will see below, already quite high.

In half of the rosters, the *Ascending Proxy Auction* returns a (slightly) worse roster. This is mainly because of the way packages are constructed. For instance, it is possible that a package is accepted, but cannot be favourably expanded. As a result, other packages without the task are possibly never tried. The second best package could have been expanded and because of positive synergy, its bid could have been higher than for the first package.

Figure 2 shows the ranking of each protocol. We see that the ECNP offers in 60% of the cases the best solution and in more than 90% either the best or second best solution.

4.2.2 Computation time

Figure 3 (a) shows the *CPU time* of each negotiation. Notice the logarithmic scale.

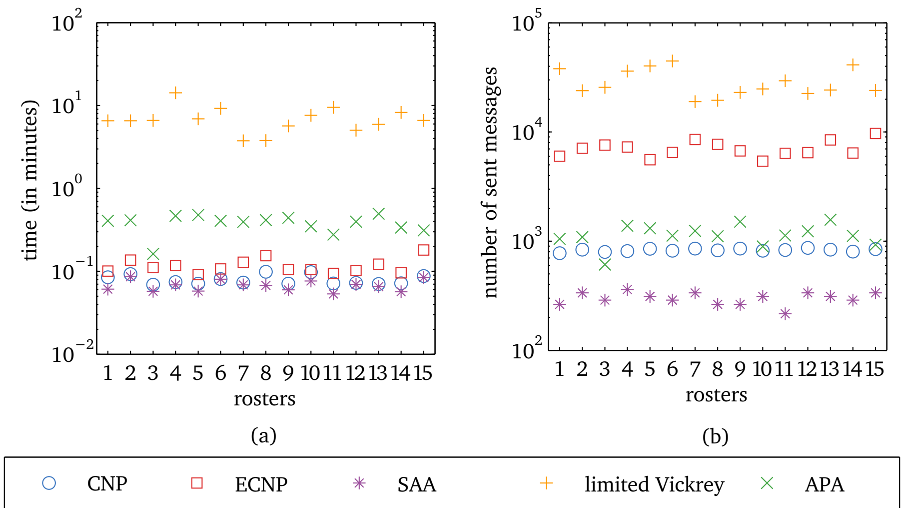

Figure 3: Comparison of protocols on (a) the total computation time and (b) the number of sent messages

The limited Vickrey, which takes about 10 minutes, is much slower than the others. This is within expectation, the server has to find optimal combinations for a vast amount of packages.

Obviously the CNP is reasonably fast. There is only one round in which all tasks have to be assigned. The ECNP on the other hand consists of multiple rounds in which the best bidders often change after considering provisional winners. All bids have to be confirmed, resulting in extra rounds and computation time.

For large problem, the SAA appears to be the fastest approach. The bidders bid on all items simultaneously, so fewer messages have to be sent and less rounds are needed to find an acceptable solution.

4.2.3 Communication time

A decomposed NRP is often implemented on a distributed system. It is therefore important to consider the communication time of the different protocols.

Figure 3 (b) shows the total number of messages sent between all participants and the server.

The most noticable difference to the CPU time is that of the ECNP. Although the total time the CPU is used is low, a great deal of messages are sent. So, seemingly faster than the APA, the wall clock time of the ECNP is presumably longer.

Again from this we can see that the the SAA not only needs least computation time, the number of messages sent is also minimal. Of course, the size of the messages is larger, a complete list is sent assigning a bid to each task.

4.3 Centralized vs distributed

For this comparison, two criteria are used, speed and quality. Both are plotted in figure 4.

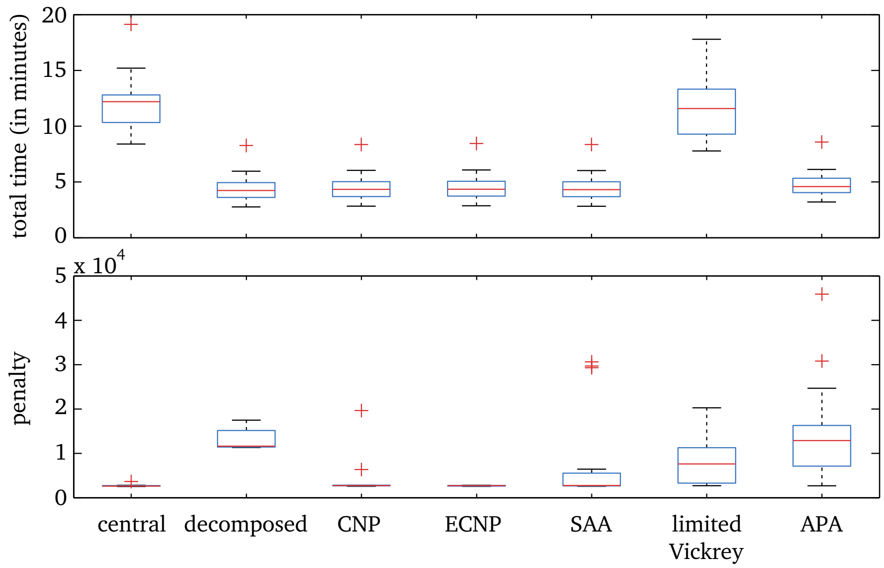

Figure 4: Comparison centralized vs distributed

The upper part shows the total CPU time. As expected, solving the smaller rosters in parallel results in a performance boost. We can also see that all but the limited Vickrey protocol add little to this time.

The quality of the results is depicted in the bottom part of the figure. After performing the negotiations the penalty is lowered significantly and approximates that of the central solution.

Surprisingly, on average, the use of decomposition and the ECNP results in a better roster than the central system (2691.6 to 2710.33). Also the standard deviation is much lower (79.9 to 269.9).

It is clear that solely based on quality, the Extended Contract Net Protocol is superior to the other presented methods.

5 Conclusion

These preliminary results show that it is possible to successfully tackle the *Nurse Rostering Problem* using problem decomposition with negotiation. In future work testing with more and larger problems should support this theory. Some aspects should also be further tested, such as the actual network performance, scalability and usability.

Previous methods focussed mainly on solving personnel shortages or maximizing a global utility. Our method can easily be combined with these. It was shown that infeasible rosters immediately offer shifts to negotiate about. Other problematic shifts are easy to be identified and future research can further improve on this.

One important conclusion is that most of the presented negotiation protocols add very little computation time. In comparison to the central method, the quality is comparable, but the performance is greatly improved.

Of the presented protocols, contracting offers the best results. If speed is important, the *Contract Net Protocol* is often sufficient. If a better quality is required, the *Extended Contract Net Protocol* is a better candidate.

In implementing the *Ascending Proxy Auction*, a lot of problems concerning finiteness and performance became apparent and solving these was not always straightforward. So from our experience, combining combinatorial auctions with the NRP is either very inefficient, as in the *limited Vickrey auction*, or very complex, as in the *Ascending Proxy Auction*. Of course we cannot generalize this to all combinatorial auctions, but it seems some researchers agree to the fact that sequential single-item auctions are much easier and in many cases faster and better than combinatorial auctions [13, 19, 16].

References

[1] S. Aknine, S. Pinson, and M. F. Shakun. An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems*, 8:5–45, 2004.

[2] L. M. Ausubel and P. R. Milgrom. Package bidding: Vickrey vs ascending auctions. *Revue Economique*, 53(3):391–402, 2002.

[3] J.F. Bard and H.W. Purnomo. Real-time scheduling for nurses in response to demand fluctuations and personnel shortages. *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, pages 67–87, 2004.

[4] G. Vanden Berghe. *An Advanced Model and Novel Meta-Heuristic Solution Methods to Personnel Scheduling in Healthcare*. Phd dissertation, University of Gent, Belgium, 2002.

[5] E. K. Burke, P. De Causmaecker, G. Vanden Berghe, and H. Van Landeghem. The state of the art of nurse rostering. *Journal of Scheduling*, 7:441–499, 2004.

[6] E. K. Burke, T. Curtois, R. Qu, and G. Vanden Berghe. Problem model for nurse rostering benchmark instances. `http://www.cs.nott.ac.uk/~tec/NRP/papers/ANROM.pdf`, 2008.

[7] E.K. Burke, T. Curtois, G. Post, R. Qu, and B. Veltman. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research*, 188:330–341, 2008.

[8] P. De Causmaecker, P. Demeester, G. Vanden Berghe, and B. Verbeke. A coordination model for distributed personnel scheduling. *Proceedings of the 19th Orbel Conference*, 2005.

[9] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:19–33, 1971.

[10] L. Di Gaspero, S. Mizzaro, and A. Schaerf. A multiagent architecture for distributed course timetabling. In *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2004)*, pages 471–474, 2004.

[11] S. Haspeslagh, P. De Causmaecker, and G. Vanden Berghe. Framework for negotiation in distributed nurse rostering problems. *Practice and Theory of Automated Timetabling, Sixth International Conference*, pages 426–431, 2006.

[12] E. Kaplansky and A. Meisels. Distributed personnel scheduling negotiation among scheduling agents. *Annals of Operations Research*, 155(1):227–255, 2007.

[13] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain. The power of sequential single-item auctions for agent coordination. In *Proceedings of the National Conference on Artificial Intelligence*, page 16251629, 2006.

[14] P. R. Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108(2):245–272, 2000.

[15] S. Petrovic and G. Vanden Berghe. Comparison of algorithms for nurse rostering problems. *Procedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*, pages 1–18, August 2008.

[16] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85, 2005.

[17] R. G. Smith and R. Davis. Framework for co-operation in distributed problem solving. *IEEE Transaction on System, Man and Cybernetics*, 11(1), 1981.

[18] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[19] M. P. Wellman. Auction protocols for decentralized scheduling. *Games and economic behavior*, 35(1-2):271, 2001.