Commentary on
"Can Multiple Learning Agents Implement Responsiveness, Conflict Resolution, and Collaboration Between Autonomous Scheduling Systems and Human Operators"

Reid Simmons
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
reids@cs.cmu.edu

The paper by Takadama, Nakasuka and Shimohara presents an architecture where multiple agents collaborate in solving scheduling problems. The agents learn scheduling rules and loosely collaborate by exchanging local information and (at times) learned rules. Experiments in the paper demonstrate that the learned rule sets help in solving similar problems, especially when changes occur, such as reordering of tasks or anomalies in the environment.

Scheduling is a difficult problem. Human schedulers need all the help they can get from automated systems. While early research focused on knowledge-intensive approaches [2, 4], researchers have obtained surprisingly good results with simpler, more local methods. In particular, the approach of starting with an infeasible (often random) plan and iteratively making changes in it using simple decisions has proven to be quite efficient and produces remarkably good results [5, 6, 8, 10]. Work in scheduling has a long history of applying other stochastic search methods to the scheduling problem, such as Genetic Algorithms (GA) [3, 9], simulated annealing [10] and randomization and restarting [1, 7], all with good success.

It is within this context that I view the current work. The process described in the paper of randomly generating rules and exchanging rules between agents strikes me as similar to a GA-type approach. The distinctions, though, are several. Foremost, in most GA-type approaches to scheduling, the individuals in a population are complete *schedules*. In this architecture, each agent (individual) is a single *job*. Thus, the agents must exchange sufficient information to enable them to avoid inconsistent schedules. Apparently, the authors limit this information to 1) the latest time of all jobs (which provides an upper bound on the total length of the schedule) and 2) overlaps with other jobs, in terms of requests for power, communication links, machines and crew.

Another distinction is how rules are created. As in GA, at set points, agents exchange their low-scoring rules with other agents and/or randomly generate new rules. However, in the architecture described in the Takadama paper, agents use reinforcement learning to learn rules that help decide how to schedule the jobs, given the available information from other agents and overall task constraints (such as ordering constraints).

It is surprising to me that this architecture performs as well as it does (although it would have been nice to see comparisons with other iterative-repair type scheduling algorithms). In particular, since agents have no way of knowing what the other agents may be doing, it would seem that this architecture could lead to thrashing. For instance, if two jobs over-subscribe for a given resource, it seems that there is nothing to prevent the two agents from changing their allocations in a way that happens to still be inconsistent. With more global information (or more synchronization between agents) this would not happen.

Similarly, it is not clear how the communication requirements between agents affect the efficiency of the scheduling task. In approaches where each agent is a complete schedule, the agents can solve their problems independently; Here, they must communicate. If there is a lot of potential for resource contention among tasks, there will be a lot of communication amongst agents. It would seem that at some point this communication would outweigh the multi-agent advantages of the approach. In such cases, a more reasonable decomposition might be to have each agent be a separate *resource*, and have each resource agent responsible for managing its own schedule by trading tasks with other resource agents.

Finally, it is not clear how this approach could handle other scheduling metrics, such as minimizing overall resource utilization, since that would involve global calculations. It would seem that, in such situations, either all the agents would need to collect, and process, this information, or else one would need a separate agent that is responsible for maintaining the global state of the system.

The learning aspects of the paper seem strong. In particular, the combination of randomization (generation and exchange of rules) and task-directed learning of rules (using reinforcement learning) is intriguing. This seems to combine the best of genetic-mutation type approaches and more knowledge-based approaches. I think this is a generally useful technique, and could probably be fruitfully applied to other GA-type approaches. I would really like to see more rigorous experiments comparing this hybrid learning approach with the individual methods alone. My intuition says that the hybrid approach will show distinct advantages, but this really needs to be demonstrated.

In summary, I am intrigued by this approach. However, it must be viewed in the context of a very large body of work in scheduling that uses stochastic search and GA-type approaches. In particular, the current approach needs to be evaluated in realistic scenarios and must be compared to benchmarks. While I have doubts about the ability of the "one agent per job" approach to scale and perform well in more complex problems, I think that the hybrid learning approach could prove to be quite useful. I look forward to more detailed and rigorous studies by the authors.

References

[1] A. Cesta, A. Oddi and S.F. Smith. "An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows". In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm Sweden, August 1999.

[2] C. Cheng and S.F. Smith. "Applying Constraint Satisfaction Techniques to Job Shop Scheduling". *Annals of Operations Research, Special Volume on Scheduling: Theory and Practice*, 1997.

[3] H-L Fang, P.M.Ross and D.Corne, "A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling and Open-Shop Scheduling Problems". In *Proceedings of Fifth International Conference on Genetic Algorithms*, pp. 375-382, Morgan Kaufmann, 1993.

[4] M. Fox. "ISIS: A Retrospective". In *Intelligent Scheduling*, eds. Zweben and Fox, Morgan Kaufmann, 1994.

[5] S. Minton, M. Johnston, A. Philips, P. Laird. "Minimizing Conflicts: A Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems". *Artificial Intelligence* **58**:161-205, 1988.

[6] N. Muscettola, S.F. Smith, A. Cesta and D. D'Aloisi. "Coordinating Space Telescope Operations in an Integrated Planning and Scheduling Framework". *IEEE Control Systems*, **12(1)**, February 1992.

[7] A. Oddi and S.F. Smith. "Stochastic Procedures for Generating Feasible Schedules. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97)*, Providence RI, July 1997.

[8] G. Rabideau, S. Chien, J. Willis, T. Mann. "Interactive, Repair-Based Planning and Scheduling for Shuttle Payload Operations". In *Proceedings of the 1999 Conference on Innovative Applications of Artificial Intelligence (IAAI)*, Orlando FL, July 1999.

[9] C.R.Reeves. "A Genetic Algorithm Approach to Stochastic Flowshop Sequencing". In *Proceedings IEE Colloquium on Genetic Algorithms for Control and Systems Engineering*, Digest No.1992/106, 1992.

[10] M. Zweben, B. Baun, E. Davis and M. Deale. "Scheduling and Rescheduling with Iterative Repair". In *Intelligent Scheduling*, eds. Zweben and Fox, Morgan Kaufmann, 1994.