Stable Scheduling of Airport Ground Handling Services by Heterogeneous Agents[1]

Xiaoyu Mao [a] Nico Roos [b] Alfons Salden [a]

[a] *Almende B.V., Westerstraat 50, 3016 DJ, Rotterdam*
[b] *DKE, Universiteit Maastricht, P.O.Box 616, 6200 MD, Maastricht*

1 Introduction

Scheduling airport ground handling (AGH) services is an instance of a *multi-project scheduling problem* (MPSP) [1]. In such MPSP, each aircraft turnaround process is a separate project and the ground service providers that perform the activities of the aircraft turnarounds are the resource managers.

Aircraft and ground service providers at an airport are independent self-interested parties. We propose a *heterogeneous* multiagent system (MAS) in which aircraft and ground service providers are modelled as autonomous agents: *project agents* and *resource agents*, respectively. The projects agents make their scheduling decisions independently of each other, they negotiate with resource agents over the time slots and associated prices for performing the neccessary activities.

In order to handle the project release uncertainty, we present an *online scheduling* scheme that reduces the risk of rescheduling by starting the scheduling process of a project when it is released. Online scheduling may lead to sub-optimal schedules compared to an offline approach. To compensate for the inefficiencies, we investigate a *cooperative* online scheduling scheme.

In a dynamic environment such as AGH, the execution of the project schedules may be invalidated by various disruptions that extend the durations of activities. As a result, project agents may incur high costs if they have to reschedule some of their activities. In this context, the insertion of *slack time* between activities is a well known solution. The delay cost incurred by the insertion of slack time should balance the expected rescheduling costs. Since in a dynamic MAS it is hard to analytically calculate optimal slack time between activities, we propose that agents acquire these slack time using a *co-evolutionary learning* method.

2 Multiagent scheduling

We employ a MAS scheduling framework with a market-based mechanism in order to enable the autonomous parties to consider their individual interests while making scheduling decisions. In this market-based mechanism resource agents sell their time slots to project agents. Each time slot reservation corresponds to the schedule of an activity. To negotiation about the schedule, the resource agent first provides a list of slot offers to the project agent. The price of each slot is calculated based on the value system of the resource agent. This list of slot offers is then re-evaluated by the project agent which will select the best slot based on its own preference.

2.1 Cooperative scheduling

We denote the above mechanism a *non-cooperative* scheduling scheme in which project agents communicate only the information necessary to make a slot reservation, no additional information is revealed. In

[1]The full version of this paper appeared in: Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10-15, 2009, Budapest, Hungary, pp. 537-544.

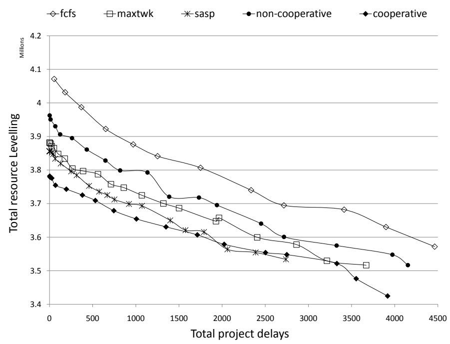

Figure 1: MAS Scheduling vs. Heuristics

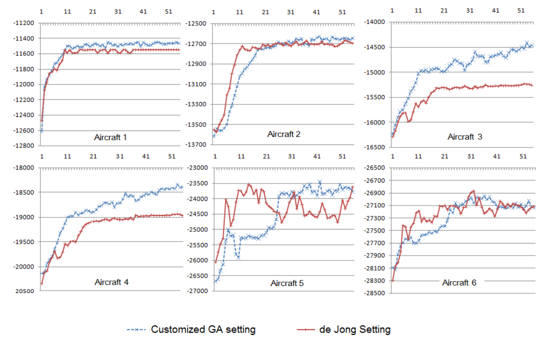

Figure 2: Aircraft Learning Curves

a *cooperative* scheduling scheme, agents are willing to share part of their personal information, under the condition that they don't suffer any loss from information sharing.

To illustrate the cooperative agent scheduling scheme, we introduce the concept of a *secure time window* (STW) for each scheduled activity. An STW starts from the latest scheduled finish time of this activity's predecessors, and ends by the earliest scheduled starting time of its successors. Any slot shifting within this STW will not cause further delay of the project that the activity belongs to. By knowing the STWs of activities, resource agents are given more flexibilities to shift the slots for a lower cost. Subsequently, cheaper slot options are provided to other project agents.

2.2 Learning slack time to handle incidents

Determining optimal slack time is a strategic game with no prior information about the project agents' payoff matrix. Analytically determining the optimal slack time is an impossible task in such a complex setting. Instead, agents can gain some experience about their own rewards by playing the game. We have chosen to use genetic algorithms (GAs) and employ a GA learner within each project agent. Therefore, project agents are co-evolving their individual strategies. An individual I_i of project agent i is encoded to represent an unique slack time configuration. To determine the fitness value $f(I_i)$ of each individual I_i, we need to run simulations with incidents disrupting the execution of certain activities. While making scheduling decisions, each project agent will take the slack time configuration into account. The fitness of an individual is then obtained when the project completes.

3 Experimental results

The performance of our MAS scheduling approaches are evaluated in both deterministic and dynamic environment. In deterministic environment, we compare the generated MAS schedules (both non-cooperative and cooperative) with those of some well known priority-based centralized heuristic approaches (i.e., FCFS, MAXTWK, SASP, see [2]). For project agents, we consider the objective of minimizing the project delays; and resource agents try to level the resource utilization. From the results shown in Figure 1, non-cooperative approach provides a schedule that is of comparable quality as the centralized heuristic methods. Moreover, cooperative approach improves the overall schedule and outperforms all other heuristics.

Next we simulate a dynamic airport situation where 6 aircraft schedules their AGH services with minor incidents. Figure 2 shows the learning curves of all aircraft with two sets of GA parameters. These learning curves show that such online co-evolutionary learning methods is capable of readily absorbing the uncertainties in the execution of the project activities and converge rapidly to a stable situation.

References

[1] U. Dorndorf. *Project Scheduling with Time Windows: From Theory to Applications*. Physica-Verlag, Heidelberg-New York, 2002.

[2] A. Lova and P. Tormos. Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling. *Annals of Operations Research*, 102(1-4):263–286, Feb 2001.