Using Neuro-Evolution in Aircraft Deicing Scheduling *

Xiaoyu Mao[1,2], Adriaan ter Mors[1,3], Nico Roos[2], and Cees Witteveen[3]

[1] Almende B.V, 3016DJ Rotterdam, The Netherlands
[2] MICC/IKAT, Universiteit Maastricht
[3] EWI, Technische Universiteit Delft
xiaoyu,adriaan@almende.com

Abstract. Resource scheduling problems with incomplete information and environment changes have been studied for decades. The vast majority of the research efforts in the scheduling problems under uncertainty assume a central authority with a global objective of maximizing the resource usage. In real-life scheduling problems, in addition to the dynamic environment, sometimes conflicting interests of different parties renders a centralized approach undesirable. Applying multi-agent approaches in resource scheduling overcomes the restrictions associated with traditional static centralized scheduling. This paper studies a multi-agent scheduling system in the context of an interesting airport planning problem: the planning and scheduling of deicing and anti-icing activities. In this application domain, self-interested aircraft agents have an incentive to reserve a deicing resource as early as possible, leading to sub-optimal schedules. To counter this effect, we propose the use of *decommitment penalties*, forcing agents to reserve the deicing resources at a later time point, which results in a better overall schedule. This paper investigates the effects of agents learning an 'optimal' strategy in this context. To learn an 'optimal' strategy, we apply genetic algorithms to train a neural network, the agent use to decide when to reserve the deicing resource. Experiments show that the neural-evolution algorithm outperforms the derived strategy based on simplified cost estimation in the decommitment penalties mechanism, which in turn significantly improve the efficiency and fairness compared with naive First Come, First Served approach.

1 Introduction

Aircraft deicing refers to the process of removing frost, snow or ice from aircraft surfaces to ensure safe take-off. The process of deicing is not part of the original flight plan at the airports in temperate climate zones, so existing departure schedules must be revised online. Moreover, the wintry conditions necessitating aircraft deicing often result in many more unexpected incidents at the airport,

* This research is supported by Dutch technology exchange program Casimir, funded by NWO (Netherlands Organization for Scientific Research). Project No.CSI4006: Hybrid Human-Agent Networks

for example due to delays in ground handling (refuelling, luggage loading and unloading, etc.).

Scheduling aircraft deicing involves several self-interested agents competing for the scarce slots at deicing stations. Each aircraft (agent) is concerned with its own punctual departure, and it does not care about the schedules of aircraft belonging to different airlines. Thus, the planning and scheduling of aircraft deicing has to be done in a highly dynamic environment involving several autonomous and self-interested parties.

Different multi-agent systems (MAS) have been designed to tackle specific real-world scheduling problems, from patient scheduling in the hospital (cf. [6] [7]) to more general job shop scheduling problems (cf. [1] [4]). In these works, different coordination mechanisms have been proposed to coordinate agents' plans, from more cooperative agents (coalition formation) to more competitive agents (market-based mechanisms). Liu and Sycara [4] developed a MAS for job shop scheduling problems in which standard operating procedures are combined with a look-ahead coordination mechanism that should prevent 'decision myopia' on part of the agents. Using their approach, system performance is said to improve in tightly-coupled, real-time job-shop scheduling environments. However, their coordination mechanism is not appropriate for competitive, self-interested agents, which makes it an undesirable choice for coordination in a deicing setting. The slot swapping mechanism in the work of Vermeulen et al. [7] in patient scheduling may be a valuable optimization tool in a dynamic schedule repair context. However, there is still a need for a coordination mechanism that finds a satisfying initial schedule.

In our previous work [5], an agent-based scheduling mechanism that makes use of decommitment penalties was proposed. Given the autonomy in the deicing slot reservation, self interested aircraft agents have an incentive to reserve a deicing slot as early as possible, leading to sub-optimal schedules. The use of decommitment penalties forces agents to reason about the uncertainty, and to reserve the deicing resource at a later time when they have a certain degree of confidence they can honour their agreement.

In [5], we derived an aircraft agent deicing slot reservation strategy, where agents were assumed to have complete knowledge about the probability function governing the occurrence of incidents that might cause them to miss their slot. Despite this simplifying assumption, determining the best strategy for an agent proved a daunting task, as it has to know at least the following: (*i*) the scheduling strategies of the other agents; (*ii*) how much additional delay will be incurred if the agent has to decommitment from a slot; (*iii*) when free slots will become available at the deicing station. Hence, the agent has neither the information nor the computational resources available to find the optimal strategy, so a better way is to let the agent *learn* how to make its decisions.

Our choice of learning mechanism for the agents was induced by the complicated dynamics underlying the deicing scenario:

– deicing is required in a highly dynamic environment,
– state values of the environment are, in principle, continuous,

- the decision to reserve a slot are based on the current information of the state of the environment,
- the reward of a decision is often delayed, and depends also on the decisions of other agents.

Genetic algorithms are suited for learning a strategy under these conditions. Therefore, we have chosen to use a genetic algorithm to train a feed-forward neural network for selecting an action.

The use of Neuro-evolution in resource distribution and machine scheduling problems with incomplete information and environment changes have been studied for years. Most approaches, however, assume a centralized scheduler entity [2, 3], neglecting the separate and often conflicting interests of the individual agents. A difference with these *neuro-evolution* approaches is that we do not train a single neural network for the whole problem, but each agent has its own private neural network. In this paper, we will investigate the quality of a strategy learned using Genetic Algorithm by comparing the derived strategy based on cost estimation (cf. [5]) and the baseline First Come, First Served scheduling approach.

The remainder of this paper is organized as follows. In Section 2 we describe the deicing scenario, and the decision problem faced by each of the agents. Next, we will outline our neuro-evolution approach in Section 3, followed by a brief discussion of the experimental results, in Section 4. In Section 5 we will conclude this work with look to the future.

2 Problem Description and Agent-based Model

Given a set of aircraft agents and a single deicing station[4], each aircraft needs exactly one slot at the deicing station to receive the deicing and anti-icing service. An aircraft can receive deicing from its *actual off-block time* (its release time from a gate) onwards, and the time between its initial *target off-block time* and the start of its deicing slot is counted as delay. The cost associated with a minute of delay may differ between aircraft, this difference reflects the fact that different aircraft agent may have different value systems. Finally, each agent knows the time the first available free slot at the deicing station by communicating with the deicing agent.

When an aircraft agent reserves a particular time slot at a deicing station, it will commit to turn up at that deicing station at the specified time. An aircraft fails to show up at the deicing station, with some probability unknown to the agents themselves because of an incident. If an aircraft has already reserved a deicing slot, it will have to decommit from that slot, and pay a decommitment penalty, which we assume to be an airport-wide constant value.

[4] Having multiple deicing stations makes the problem more interesting from a combinatorial optimization point of view, but it is not especially relevant to our investigation into the relative merits of auctioning and decommitment.

Hence, with the introduction of decommitment penalties, an agent has an incentive to reserve a slot as late as possible, because then there is only a short span of time in which an incident can disrupt its schedule. On the other hand, if an agent waits too long to reserve the next available free slot, another aircraft might reserve it. Hence, the agent will also have an incentive to reserve a slot as early as possible.

Each aircraft agent therefore has the following decision problem to solve:

> *Do I reserve the currently available first slot, or do I reserve a slot at a later time?*

To answer this question, the aircraft agent has to be able to evaluate these two different options. And to judge whether the decision to reserve now has any merit, In the previous research [5], we assumed that agents have the complete information therefore are able to make an estimation of the incidents, and make a decision by comparing of the cost of those two options.

Trying to incorporate complete environment factors and total airport information into a realistic model is a formidable task, especially as the slot-reserving behaviour of agents may be subject to their perception (and prediction) of other agents' behaviour. Therefore, we investigate in this work agents that learn a strategy for tackling the decision making problem.

3 Solution Method

To learn a good decision-making strategy, we associate a neural network with every aircraft agent. In this section we will describe (1) the structure of this neural network, and the choice of input/output representation layers; (2) the evaluation function that measures the performance of each neural network; (3) the genetic algorithm used to train the neural networks.

Neural network representation We use a multi-layer feed-forward neural network with three input nodes and one bias node, one hidden layer consisting of three nodes, and one output node; this implies a total of fifteen connections between the nodes, each with its own weight factor (see Figure 1(a)). The three input nodes are the partial observations of the aircraft agent and are specified as follows:

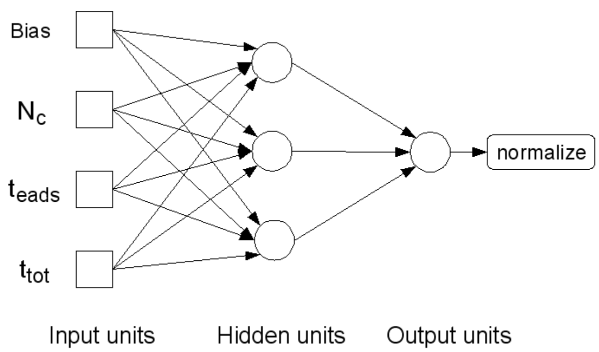
1. the time until the earliest available deicing slot t_{eads} [5];
2. the time until the target off-block time t_{tot} [6];

[5] The earliest available deicing slot are assumed to be observable for aircraft agent by sending request to deicing resource.

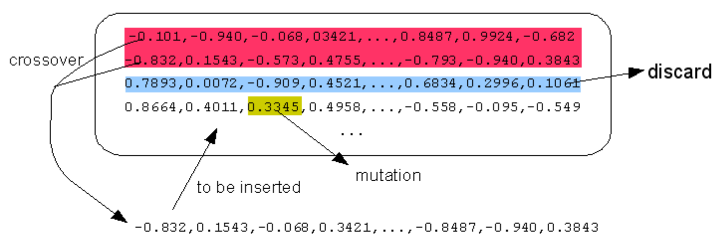
[6] The target off-block time is the estimated time point when all ground handling for an aircraft are assumed to be finished and aircraft are allowed to off-block from the gate.

3. the number of potential competitors N_c for the next deicing slot [7].

The **tanh** function $\frac{e^x - e^{-x}}{e^x + e^{-x}}$ is used for the network activation function. The output of the network should provide an answer to the decision question in the end of Section 2, network output is eventually normalized to a boolean value for the decision making.



(a) Neural network representation



(b) An iteration of the Genetic Algorithm

Fig. 1. Our neuro-evolution setup

Evaluation function After an aircraft has received deicing, it knows its delay relative to its original flight schedule. As each agent has a private function to associate a cost value with a delay, it can evaluate how well it has done. This delay-to-cost function is also the evaluation function of the neural network, and

[7] We assume that the aircraft target off-block times $t_{tot}s$ are known to all agents since they are public information as in original flight schedules. Agent may derive N_c as the number of aircraft that is ready to be deiced but has not been assigned to any deicing slots yet.

it is most of the time a delayed 'reward' since the decision made by a neural network does not provide a direct feedback until the actual deicing is performed.

A genetic algorithm for neural network training As we discussed in the previous section, in this dynamic environment with a delayed reward, genetic algorithms are suited for learning a strategy. We train the neural network by evolving its weights using a genetic algorithm. One individual in the population is a vector of the fifteen weights, encoded as a list of real numbers. All members of the population are initialized to a list of random numbers, uniformly distributed between -1.0 and 1.0. To generate new members of the population, we use both crossover and mutation, and to select which individuals will be inserted into or removed from the population, we use tournament selection. See Figure 1(b) for an illustration of these concepts.

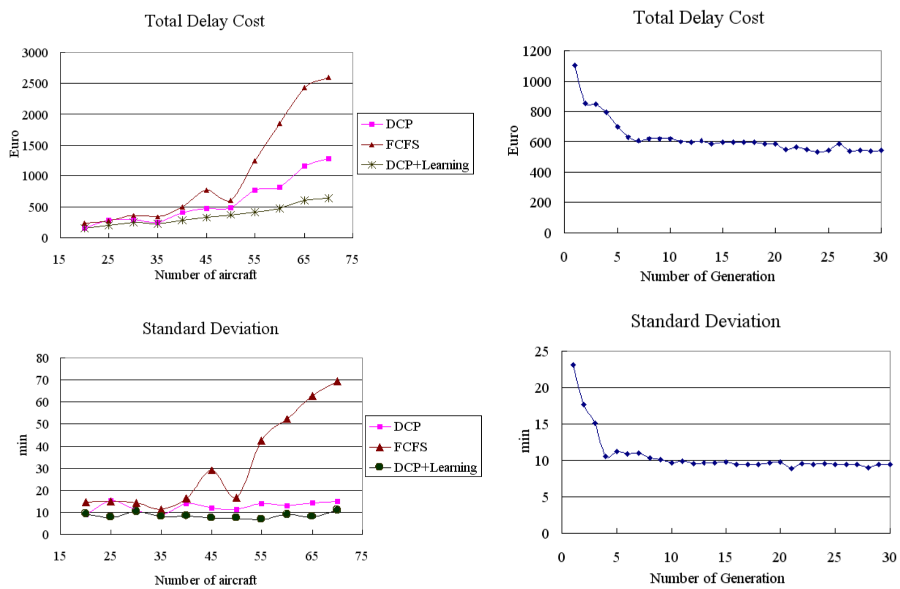
4 Experimental Results

We conducted these experiments using a fix-length deicing time of 5 minutes. Aircraft target off-block times (t_{tot}) are randomly distributed over five simulation hours. Deicing slots may still be allocated after the initial five hours in case of a high congestion; in fact, the simulation continues until all aircraft have received a deicing slot. For these parameters, the number of aircraft n that can maximally be serviced without any delay equals $n = \frac{5\times 60}{5} = 60$, assuming a maximally convenient distribution of target off-block times. This means that with a random distribution of t_{tot}, we can expect some delays regardless of the scheduling strategy in case we have more than 60 aircraft. Some further parameter values include: the fixed airport-wide decommitment penalty $\delta = 10$; the time in between two rounds of requiring the earliest available slot from deicing station is set to 5 minutes. The number of aircraft in the experiment ranges from 10 to 70, and the population size for genetic algorithm is set to 50. An important note is that the population here is not the number of aircraft but the number of weight vectors for each aircraft agent.

We compared our learning algorithm with the simple estimation strategy based on incidents probability analysis (first presented in [5]), and also with the naive, baseline scheduling strategy called First Come, First Served, where agent reserve their deicing when they land at the airport.

We judge the algorithms on two criteria: the first one is the measurement of total delay cost of all aircraft. Recall that the delay cost of one agent is a function mapping the delay in minute to a cost value based on individual agent's value system. This criterion measures the efficiency of the coordination mechanism. As a second criteria, we also record the standard deviation of delay in minutes, summed over all agents. The standard deviation can be interpreted as a measure of fairness: if it is low, then all agents suffer a comparable amount of delay.

From the result shown in Fig.2(a), the neuro-evolution algorithm outperforms the simple estimation strategy based on incidents probability analysis, which in turn outperforms the FCFS strategy in both criteria.



(a) Relative performance of FCFS, DCP, DCP+learning

(b) Convergence rate of learning

Fig. 2. Experimental results

Fig.2(b) shows the convergence rate of the genetic training, we can see from this figure that with a population of 50 individual weight spaces, we achieve the near-optimal after about 10 generations.

5 Conclusion

In this paper we have discussed a multi-agent system for the scheduling of airport deicing services, in a dynamic and competitive resource scheduling environment. By introducing the neuro-evolution algorithm for agents learning a decision-making strategy, we improved the overall scheduling efficiency and fairness compared with a simple estimation strategy based on incidents probability analysis, which in turn significantly improved the schedule made by a naive random-order First Come, First Served queue.

Number of options for future work are considered. First, we would like to investigate other scheduling strategies in conjunction with decommitment penalties. Second, other multi-agent learning approaches will be investigated in this dynamic resource scheduling setting and will be compared with the neuro-evolution learning.

Another extension is to look at the relation with other airport planning and scheduling problems. What makes this extension interesting is the interaction with other planning and scheduling problems, possibly involving other planning agents. The challenge for airport deicers lies in inserting the deicing activities into the existing plans from landing to take-off for aircraft at an airport.

6 Acknowledgments

The authors would like to thank Peet van Tooren, Alfons Salden, Jeroen Valk, Tamás Máhr, Gerald de Jong, and Jan Peter Larsen (Almende B.V.) as well as Steven de Jong, Jaap van den Herik (Maastricht University) for their help in this research.

References

1. Cardon Alain, Galinho Thierry, and Vacher Jean-Philippe. An agent based architecture for job-shop scheduling problem using the spirit of genetic algorithms. In *Proceedings of Evolutionary Algorithms in Engineering and Computer Science , EUROGEN99, p. 20-27, Finlande, 30 May - 3 Juin 1999*.
2. Tatiane R. Bonfim and Akebo Yamakami. Neural network applied to the coevolution of the memetic algorithm for solving the makespan minimization problem in parallel machine scheduling. In Teresa Bernarda Ludermir and Marcílio Carlos Pereira de Souto, editors, *SBRN*, pages 197–199. IEEE Computer Society, 2002.
3. Faustino J. Gomez, Doug Burger, and Risto Miikkulainen. A neuroevolution method for dynamic resource allocation on a chip multiprocessor, June 10 2001.
4. Jyi-Shane Liu and Katia P. Sycara. Multiagent coordination in tightly coupled task scheduling. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pages 164-171, May 23 1996.
5. Xiaoyu Mao, Adriaan ter Mors, Nico Roos, and Cees Witteveen. Agent-based scheduling for aircraft deicing. In Pierre-Yves Schobbens, Wim Vanhoof, and Gabriel Schwanen, editors, *Proceedings of 18th Belgium - Netherlands Conference on Artificial Intelligence, BNAIC06*, Namur, Belgium, 2006.
6. Torsten O. Paulussen, Nicholas R. Jennings, Keith S. Decker, and Armin Heinzl. Distributed patient scheduling in hospitals. In Georg Gottlob and Toby Walsh, editors, *IJCIA*, pages 1224–1232. Morgan Kaufmann, 2003.
7. I.B. Vermeulen, S.M. Bohte, D.J.A. Somefun, and J.A. La Poutré. Improving patient schedules by multi-agent pareto appointment exchanging. In *In Proceedings of 2006 IEEE International Conference on E-Commerce Technology (CEC/EEE 2006), San Francisco, California, June 26-29, 2006*.