Coordinating Competitive Agents in Dynamic Airport Resource Scheduling

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

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

Abstract. In real-life multi-agent planning problems, long-term plans will often be invalidated by changes in the environment during or after the planning process. When this happens, short-term operational planning and scheduling methods have to be applied in order to deal with these changed situations. In addition to the dynamic environment, in such planning systems we also have to be aware of sometimes conflicting interests of different parties, which render a centralized approach undesirable. In this paper we investigate two agent-based scheduling architectures where stakeholders are modelled as autonomous agents. We discuss this approach in the context of an interesting airport planning problem: the planning and scheduling of deicing and anti-icing activities. To coordinate the competition between agents over scarce resources, we have developed two mechanisms: one mechanism based on *decommitment penalties*, and one based on a more traditional (Vickrey) auction. Experiments show that the auction-based mechanism best respects the preferences of the individual agents, whereas the decommitment mechanism ensures a fairer distribution of delay over the agents.

1 Introduction

Aircraft *deicing* and *anti-icing* is required in winter time when frost, snow, and ice form on the wings and fuselage of an aircraft. Such a layer of frost or ice on aircraft surfaces influences the aircraft's aerodynamic properties which may cause a loss of lift that could result in a crash. Deicing refers to the removal of frost, snow, or ice from aircraft surfaces, while anti-icing is the application of a layer of viscous fluid onto aircraft surfaces that should prevent snow or ice from accumulating. Since the deicing and anti-icing operations are always performed together, in the remainder of this paper we will not distinguish them and will use the term *deicing* to refer to both deicing and anti-icing.

The planning and scheduling of deicing activities at airports is an important and challenging part of airport departure planning. Like other real-life planning problems, long term planning can be invalidated by the dynamic changes in the environment during or after the planning process. In these cases, short-term operational planning and scheduling methods have to be applied. In addition to

P. Petta et al. (Eds.): MATES 2007, LNAI 4687, pp. 133–144, 2007.

the dynamic environment, in such planning systems we also have multiple self-interested parties that often have conflicting interests, which makes a centralized approach less appropriate.

The dynamic nature of the aircraft deicing problem stems from the fact that in many temperate climate zones as found in Western Europe, the process of deicing is not part of the original flight plan, and thus it has to be scheduled as part of *operational* (i.e., short-term) planning. Moreover, during wintry conditions involving snow and ice, airport capacities will be greatly reduced — again, in temperate climate zones, this is not taken into account in the flight schedules — putting a great strain on the re-planning capabilities of all parties involved. These parties are self-interested and often have conflicting interests. For instance, airlines and pilots will be concerned with the effects of deicing on their flight schedules, air traffic control will be responsible for safe flight movements, the airport itself will strive for a maximum utilization of its facilities (runways, gates, etc.), and the ground servicing companies performing the deicing will want to operate as efficiently as possible. To resolve the dependencies between self-interested parties, we need some form of coordination.

In this paper, we investigate two coordination mechanisms: *i*) coordination based on decommitment penalties and *ii*) a Vickrey auction mechanism. The decommitment-penalties mechanism aims at minimizing the total delay on the airport, and distributing this delay evenly over the agents in the system. The auction mechanism aims to find an allocation of slots that matches the preferences or priorities of the agents (for instance, a fully-loaded Airbus 380 aircraft with many passengers on board may value its punctual departure higher than a half-empty Fokker 50).

This paper is organized as follows. In Sect. 2, we will describe the background of the airport deicing scheduling problem, and we link it to the problem of multi-agent scheduling. In Sect. 3 we will give a formal model of the deicing scheduling problem and we will introduce a simple solution scheme. The agent coordination mechanisms will be discussed in Sect. 4; in Sect. 5 we will show the experimental comparison of the auction and decommitment coordination mechanisms. Section 6 concludes with a look to the future.

2 Background and Related Work

Like many real world problems, the problem of managing deicing resources exhibits characteristics of both planning and scheduling. It is a scheduling problem in the sense that aircraft tasks have to be allocated to resources over time, and it is a planning problem in the sense that an aircraft has a number of choices with regard to which deicing resource to make use of — and this choice of deicing resource has implications for other airport planning problems like arrival planning, departure planning, and taxiway planning. Nevertheless, the management of deicing resources can best be characterized as a scheduling problem as it involves only a small, fixed number of choices, and because the focus is more on time and resource constraints, rather than on ordering of actions (cf. [1]).

If agents were to schedule completely independently of each other, the union of their plans would show many conflicts. In the airport deicing domain, these conflicts will concern the simultaneous use of scarce resources. We therefore define the problem of multi-agent scheduling as follows:

Definition 1 (Multi-Agent Scheduling). *Given a set of agents each with a set of tasks to schedule, and a set of resources to schedule them on, each agent should find an individual schedule for its tasks in such a way that none of the resource capacity constraints are violated.*

Obviously, satisfying all resource constraints will not happen by magic; the agents will need some coordination mechanism that will safeguard these constraints. Therefore, we can summarize the multi-agent scheduling problem as follows:

Multi-Agent Scheduling = Distributed Scheduling + Coordination

Within multi-agent scheduling research, two main tracks can be identified: cooperative agent scheduling and competitive agent scheduling (or selfish scheduling). The scheduling of deicing resources has characteristics of both cooperative and competitive scheduling, as the aircraft/airline agents are competing for access to scarce resources, whereas deicing-resource agents are collaborating in order to maximize resource utilization. In this paper, we will focus our attention on mechanism design for selfish agents.

Since the work of Nisan and Ronen [2] on mechanism design, in 1999, selfish scheduling has recently been studied by many researchers. Some researchers consider the machines to be the selfish agents machines [3,4,5], while others associate an agent with a single task or job [6,7]. However, all these works differ from our paper since they dealt with scheduling problems in a static environment.

Related work on dynamic selfish scheduling is by Vermeulen et al. [8], who developed a Pareto-optimal appointment exchanging algorithm in a patient-scheduling problem. The objective is to improve upon the initial schedule, constructed using first come, first served, by letting patient-agents exchange their slots. It is quite similar to the work of Paulussen et al. [9] where the agent coordination mechanism is a dynamic schedule-repair affair that can be classified as an after-scheduling coordination mechanism. Although Vermeulen's slot swapping mechanism may be a valuable optimization tool in a dynamic schedule repair context, there is still a need for a coordination mechanism that finds a satisfying initial schedule.

In this paper we present and compare two coordination mechanisms for obtaining an initial schedule: the first is based on an auction for selling deicing slots, the second is based on decommitment penalties. In previous research, auction-based scheduling methods have been well studied since they respect the natural autonomy and private information in decentralized systems [10,11,12]. In contrast to these previous approaches, we investigate the auction-based scheduling scheme in a dynamic scheduling environment. Decommitment research has been primarily used to enable agents to explore new opportunities from the domain or from other agents [13,14]; an example is a package-delivery agent that decommits the contract for one package so that it is able to accept a more profitable package to deliver [13]. Another use of decommitment penalties is to allow agents to speculate on future events [15]. We propose that the concept of decommitment penalties can also be used to coordinate agents, by associating a penalty with the occurrence of an agent decommitting from a slot because it could not make the agreed time. In this sense, the decommitment mechanism curbs the greedy tendency of agents to grab the deicing station resource as early as possible, before other agents have a chance to take it. Now, every agent gets that chance, but it has to suffer the consequences if it miscalculated its ability to make its slot.

3 Modelling the Aircraft Deicing Scheduling Problem

In this section we will present a formal model of the aircraft deicing scheduling problem and discuss how uncertainty in the environment influences the scheduling process.

Definition 2 (Aircraft Deicing Scheduling Problem). *The aircraft deicing scheduling problem is a tuple* $\langle A, D, c, \tau, p, P, l\rangle$ *where*

- A *is a set of* n *aircraft agents,*
- D *is a set of* m *deicing station resources,*
- $c : D \to \mathbb{N}$ *is a capacity function specifying the number of aircraft that can simultaneously be serviced at the deicing station (i.e., the number of* bays*),*
- $\tau : A \to \mathbb{R}$ *is a function associating a Target Off-block Time with every agent, which is in fact the time aircraft is able to leave the gate for deicing,*
- $p : A \to \mathbb{R}$ *is function that specifies the deicing process duration for a certain aircraft,*
- $P : \mathbb{R} \times A \to \mathbb{R}$ *is a function that gives the the probability that an incident will happen to a certain agent,*
- $l : \mathbb{R} \times A \to \mathbb{R}$ *is a function that assigns a cost to the delay of an aircraft.*

The incident probability $P(t, a_i)$ indicates the probability that an incident will occur in the interval $[t, \tau(a_i)]$, i.e., the time during which the aircraft agent will receive ground services at the gate. The occurrence of such an incident may delay the Target Off-block Time, and rescheduling will therefore be needed for an aircraft having a deicing slot right after τ. The aircraft delay cost function $l : \mathbb{R} \times A \to \mathbb{R}$ maps delay in minutes to cost, reflecting the fact that different agents may have different value systems.

A solution to an instance $\langle A, D, c, \tau, p, P, l\rangle$ is a multi-agent schedule given by the vector $S = \langle(d_1, I_1), \ldots, (d_n, I_n)\rangle$ where (d_i, I_i) is a tuple in which $d_i \in D$ is the deicing station assigned to agent a_i during interval I_i such that

$$I_i = [s_i,\ s_i + p(a_i)] \wedge s_i \geq \tau(a_i) \tag{1}$$

where s_i is the deicing start time of a_i. A feasible schedule satisfies the following resource constraints: at every point in time t, the deicing resource utilization for every resource does not exceed its capacity:

$$\forall t \forall d \in D |\{a_j \in A \mid (d, I_j) \in S \wedge t \in I_j\}| \leq c(d) \tag{2}$$

Given a Target Off-block Time for each aircraft agent a_i, an individual agent tries to minimize its delay $dl_i = s_i + p(a_i) - \tau(a_i)$. For the set of all agent schedules, we can define two optimization criteria: the first is to minimize the total delay cost of all aircraft: $\min \sum_{a_i \in A} l(dl_i,\ a_i)$ as a measure of social welfare; another criterion is to minimize the sum of standard deviations in individual aircraft delay, which reflects the fairness of resource allocation at the airport.

Although the list of things that can go wrong in airport deicing operations is too extensive to fit into an elegant model of agent reasoning with uncertainty, observations from real and simulated deicing operations lead us to conclude that many incidents are concentrated in the ground servicing of the aircraft. For example, if the apron in front of an aircraft accumulates too much snow, it becomes difficult for ground servicing vehicles like baggage carts to reach the aircraft, and push-back vehicles cannot find the grip required to tow an aircraft away from the gate. Hence, there is a great deal of uncertainty surrounding the Target Off-block Times of aircraft.

If an aircraft agent is considering at time t whether to reserve (or bid for) the deicing slot starting at time t_s ($t_s \geq t$), then two factors are relevant:

1. $\delta_1 = \tau(a) - t$: If δ_1 is large, then there are many ground servicing tasks that still need to be performed, in which case the probability that something will cause a delay is considerable.
2. $\delta_2 = t_s - \tau(a)$: If the reserved slot is very far away from the Target Off-block Time, then a small delay during ground handling will not necessarily mean that the deicing slot will be missed.

In this paper, we assume that the probability-of-decommitment only depends on δ_1 and no incidents will occur after ground services are finished. Hence, we assume that the probability-of-decommitment function has the following form:

$$P(t, a) = \begin{cases} 0 & \tau(a) < t \\ \min(c, \alpha \cdot (\tau(a) - t)) & \text{otherwise} \end{cases} \tag{3}$$

where c and α are constant values between 0 and 1. The constant c provides an upper bound on the probability of having an incident, even if t is an arbitrarily early time of requesting the de-icing slot. The constant α regulates the rate of incident-occurence. If α is very large, then even when a de-icing slot is requested close to the off-block time, there is a high probability of suffering an incident.

4 Coordination Mechanisms

In this section we will describe two coordination mechanisms: coordination using a Vickrey auction to sell deicing slots to the highest bidder and coordination through decommitment penalties.

A simplifying assumption we will make for both coordination settings is that there is only a single deicing station having a single deicing bay. 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.

4.1 Vickrey Auction Mechanism

Bidding for a (deicing) slot is a straightforward way of distributing the scarce deicing slots over the self-interested aircraft agents[1]. Our idea is that the aircraft agents with the highest need get the best slots. In the airport scheduling case, the different preferences of the aircraft agents can be the result of, for example, the number of passengers aboard an aircraft, or the level-of-service that an airline wishes to maintain. If we assume that an agent may not sell a slot to another agent in case it has to decommit, then the value of the slot is a *private value*. In private value auctions all auction types give the same result according to the revenue equivalence theorem. Therefore, we choose the Vickrey auction (a closed-bid, second price auction), because of its property that (rational) agents are encouraged to bid their true value. Hence, deicing of aircraft should occur in the order of agents who are willing to pay the most. We will now describe how we set up the auction.

The deicing station will initiate a new auction when the start of next free deicing slot (starting at t_{nextslot}) is approaching, e.g. half an hour before t_{nextslot}. In each auction, the deicing station auctions off the next available deicing slot (alternative auction schemes like accepting bids for multiple deicing slots are less appropriate given the dynamic nature of the setting). To determine its value for a certain slot, an aircraft agent a should first check whether the start time of this slot t_{nextslot} is greater than its Target Off-block Time $\tau(a)$; if it is not, then the agent can't make use of this slot. In case $\tau(a) < t_{\text{nextslot}}$, an agent needs to estimate the delay it will incur by not obtaining the current slot. If there are m other aircraft in the system that also need deicing, then the value of the $(m+1)$-th slot is 0, because all competing agents can be served before this time. Then, the private value for agent a of the slot starting at t_{nextslot} is:

$$pv(a) = l(t_{m+1} - t_{\text{nextslot}}, a) \tag{4}$$

However, not all aircraft agents in the system will be able to compete for the next slot, in case their Target Off-block Times are greater than t_{nextslot}. Therefore, the number m may be smaller than the total number of agents (left) A^* in the system. At the same time, we cannot simply equate m to the number k of *direct competitors* —agents having a $\tau < t_{\text{nextslot}}$— because after the first

[1] Note that in the General Motors paint station problem [11], the roles of the agents are reversed: there, the resource agents have needs due to e.g. switching costs from one colour of paint to another. In our deicing problem, the jobs (aircraft agents) have needs, based on their flight schedules and other considerations such as service levels that must be maintained towards their customers.

k aircraft have been serviced, more agents will be ready for deicing. Finding the set A_c of competing agents can be done simply by extending, agent by agent, the set of direct competitors for a slot (see Algorithm 1). Note that in case of insufficient deicing capacity, the set A_c will quickly equal the set of all agents that have not yet received deicing.

Algorithm 1. Calculate the set A_c of agents competing for slot $t_{nextslot}$

```
t_next := t_nextslot; A_c := φ
boolean isDone := false
while !isDone do
  A' := {a ∈ (A*)|τ(a) ≤ t_next}
  A* := A* \ A'
  A_c := A_c ∪ A'
  t_next := t_next + |A'| · p(n)
  if A' = φ then
    isDone := true
  end if
end while
return A_c
```

Having described how to determine the number of competitors for a slot, we now return to the definition of an agent's private value for a slot.

Formula 4 ignores the possibility that incidents can occur during other ground services that will cause an agent to miss its reserved slot. Taking into account the incident probability $P(t, a)$, we get the following private value:

$$pv(a) = l(t_{m+1} - t_{\text{nextslot}}, a) \cdot (1 - P(t, a)) \tag{5}$$

Equation 5 thus expresses that an agent's private value of a slot decreases as the probability increases that it will not make that slot. In the next subsection, we will introduce an alternative coordination mechanism that focuses not so much on agent preferences, but more on the effects of decommitting on the schedule of an agent.

4.2 Decommitment Penalty Mechanism

When an aircraft agent reserves a particular time slot at a resource such as a deicing station, it will commit to turn up at that deicing station at the specified time. If the aircraft fails to show up, it has to pay a decommitment penalty to the deicing station. Hence, with the introduction of decommitment penalties, agents have an incentive to reserve as late as possible; after all, if it reserves a slot five minutes from now, it will be fairly certain it can make this slot. On the other hand, if an agent waits too long to reserve the next available free slot, another aircraft might reserve it. Therefore, the agent will also have an incentive to reserve a slot as early as possible.

Our approach to coordination using decommitment penalties can be described as follows. An agent can reserve any free slot at a deicing station, as the deicing station will accept all requests. However, with a certain probability incidents occur that make it impossible for the aircraft to be present at the deicing station at the agreed time. When such an event occurs, it must decommit and pay a decommitment penalty, which we assume to be an airport-wide constant δ. We assume that the availability of the deicing resource is known to all aircraft agents. Therefore, an aircraft agent a can see when the first available slot starts, and it has to solve the following decision problem:

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

To judge whether the decision to reserve now has any merit, the agent needs to estimate the probability it will have to decommit from the slot. For this, we can make use of Equation 3. Judging the option of reserving a slot at a later time is more difficult, as it needs to predict the availability of deicing slots in the future. This availability depends on at least the following factors:

1. the passage of time; if a slot is available 10 minutes from now, then, if no-one else takes it, there will be a slot 5 minutes in the future 5 minute from now,
2. other agents can reserve slots while an agent is waiting to decide.

Trying to incorporate all these factors 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 will make the following simplifying assumptions to make the task of foretelling the future a more tractable one:

- If an agent has to decommit from a slot, then it will have to find a new slot. Apart from the time lost in decommitment, we assume that the number of aircraft needing deicing per hour stays constant throughout the day. Hence, an agent will not suddenly find itself in a departure peak, after having to decommit.
- When an aircraft opts to postpone its decision to reserve a slot until the next round, and it turns out that another agent has reserved the previously earliest slot, then the start time of the new earliest slot is simply the start time of the old slot plus the deicing time, which we assumed to be equal for all aircraft.

Armed with these simplifications, we can develop a strategy for an aircraft agent.

Strategy (Deicing Slot Reserving Strategy). *Reserve the earliest available slot if the expected cost of reserving this slot is less than the expected cost of reserving a slot the next round*[2]*; otherwise, postpone the reservation decision until the next round.*

[2] We assume a short and constant period of time in between two rounds of the agent's decision process.

We will now introduce a number of functions to be able to define the expected cost of reserving the earliest available slot, which takes into account the results of having to decommit. First of all, an agent has to pay the decommitment penalty δ; second, if t_d stands for the time decommitment occurs ($t_d < \tau$), then the aircraft has wasted $(t_d - t)$ minutes (where t is the time at which the slot was reserved). We assume that this quantity $(t_d - t)$ will in fact delay deicing by $(t_d - t)$ minutes. As the delay cost $l(dl, a)$ defined in Def. 2 is a linear function, we can calculate the expected cost of decommitment for agent a as:

$$E_{dcp}(t, a) = \delta + \frac{l(\tau(a) - t, a)}{2} \tag{6}$$

Using the above definitions, an aircraft agent a can calculate the expected cost of reserving a slot at time t with earliest available slot time t_s:

$$E_{\text{res}}(t, t_s, a) = P(t, a) \cdot E_{dcp}(t, a) + (1 - P(t, a)) \cdot l(t_s + p(a) - \tau(a), a) \tag{7}$$

Note that a more realistic model for the cost of reserving a slot would be forward recursive: in case an aircraft has to decommit, it will have to try to get a slot again in subsequent rounds, again with the possibility of having to decommit, adding to its cost. Equation 7 effectively cuts off this forward recursion after one step, by taking into account only the immediate cost for decommitment.

To determine the expected cost $E_{\text{wait}}(t, t_s, a)$ of reserving a slot in the next round, we need the current time t, the time of the next reservation decision t^+, the start time of the first available deicing slot t_s, and the start time of the second available slot (in our case t_s plus the standard deicing time), then the expected cost of waiting until the next round is given by the following function:

$$E_{\text{wait}}(t, t_s, a) = P_T(t) \cdot E_{\text{res}}(t^+, t_s + p(a), a) + (1 - P_T(t)) \cdot E_{\text{res}}(t^+, t_s, a) \tag{8}$$

in which $P_T(t)$ stands for the probability of another agent having reserved the next available slot between time t and t^+. This probability function is based on the number of aircraft in the system, and the scarcity of the deicing resources. We assume aircraft take-off times are independent of each other and are uniformly distributed over time, and so we model the probability $P_T(t)$ with a Poisson distribution $f(k; \lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ where:

$$P_T(t) = 1 - f(0, \frac{t^+ - t}{|D| \cdot T}) = 1 - e^{-\frac{|A| \cdot (t^+ - t)}{|D| \cdot T}} \tag{9}$$

and T is the time in minutes over which these aircraft are distributed (e.g., we could have a simulation run of $T = 300$ minutes in which $|A| = 100$ aircraft have to be deiced using $|D| = 4$ deicing stations).

Equation 8 basically expresses that by not reserving a slot this round, there is a chance that another agent reserves the previously earliest available slot, and you consequently have to schedule a later slot, which will result in more delay; on the other hand, if no agent has reserved the slot starting from t_s, then this possibility is still open to you at time t^+. By this time, the probability of decommitment will have lowered (i.e., $P(t^+, a) < P(t, a)$), and thus reserving this slot at time t^+ will have a lower expected cost.

The agent strategy we propose in this section is simple: in case $E_{\text{res}} < E_{\text{wait}}$, the agent will reserve at time t the slot starting at t_s, otherwise it will wait until the next round. In the next section, we will investigate whether reasoning about decommitment in this way results in improved performance.

5 Experimental Results

In this section, we will compare the two coordination mechanisms of Sect. 4 with each other, and also with a naive, baseline scheduling strategy, which we have termed the Naive Scheduling Strategy (NSS). This strategy schedules deicing slots on a first come, first served basis. When an aircraft arrives at the airport, NSS assigns to this aircraft the first available slot after its target off-block time.

We judge the algorithms on two criteria: the first one is the total delay cost of all aircraft, given by the sum of the delay costs of all agents. Recall that the delay cost of one agent a is given by $l(dl, a)$, where dl is the agent's delay in minutes — this means that we do not take auction fees and decommitment penalties into account when calculating the global cost. Hence, this criterion measures the efficiency of the coordination mechanisms. As a second criterion, 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.

We conducted these experiments using only a single deicing station with a single deicing bay, and a deicing time of 5 minutes. Target Off-block Times (τ) are randomly distributed over 5 simulation hours. Deicing slots may be allocated after the initial five hours; 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 τ. This means that with a random distribution of τ, we can expect some delays regardless of the scheduling strategy in case we have more than 60 aircraft. Some further parameter values include: the delay cost per time unit in the function l for agent a is randomly distributed over $[0.5, 1.0]$; a fixed value for the decommitment penalty $\delta = 50$; and the maximum decommitment probability $c = 1.0$; In the auction setting, slots are auctioned half an hour in advance; and the time in between two rounds in decommitment penalties is set to 5 minutes. The number of aircraft in the experiment ranges from 10 to 90. The results of the experiments are displayed in Figure 1.

The first thing that catches the eye in Figure 1 is that the NSS strategy is outperformed by the two other mechanisms on all counts, except for runs having a very small number of aircraft, in which the auction setting does not perform very well. The reason for this is that in the auction setting we sell slots starting from specific times, such as 10:00, 10:05, etc. In case there is a mismatch with aircraft Target Off-block Times, for example if $\tau(a_i) = $ 10:03 for some aircraft a_i, then small delays will be incurred by the aircraft. As competition for the deicing resources increases, these small delays become less significant.

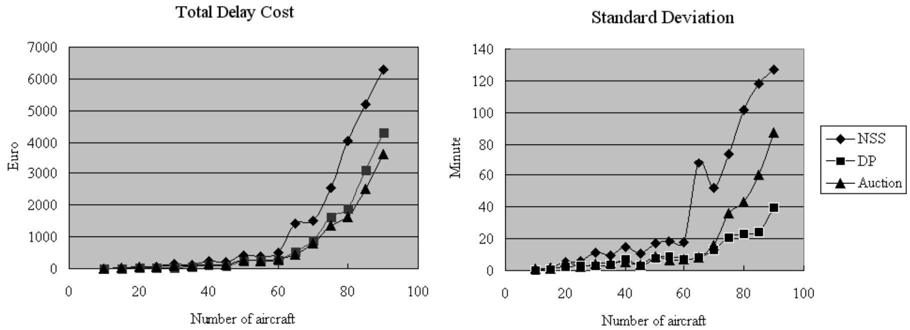

Fig. 1. Total delay cost and standard deviation in NSS, Decommitment Penalty (DP) and Auction

Another observation is that as soon as the airport starts getting congested —from around 70 aircraft— the standard deviation for the auction mechanism shoots up, leaving the decommitment mechanism 'behind'. Note that Figure 1 shows that for the less efficient NSS strategy, airport congestion starts from around 60 aircraft.

As a final remark, we can conclude that the auction mechanism is the most efficient choice for congested airports in terms of total delay cost. However, when there are relatively few aircraft that need to be deiced, the auction mechanism (at least in its current implementation) is not as efficient. The increased efficiency of the auction mechanism does come with a price, however, namely that delay is distributed more unevenly over the aircraft.

6 Conclusion and Future Work

In this paper we have proposed an agent-based model for the scheduling of aircraft deicing services. We introduced two agent coordination mechanisms — a Vickrey auction and a mechanism based on decommitment penalties. The former best caters to the preferences and relative priorities of the agents, the latter one ensures the fairest distribution of delay over the agents. Both mechanisms outperform a naive coordination mechanism based on first come, first served.

Options for future work are too numerous to list exhaustively. We would like to investigate other scheduling strategies in conjunction with the mechanisms presented in this paper. Also, our results currently rely on some simplifying assumptions, and it would be interesting to see whether the conclusions of this paper hold up if we relax some of these assumptions. Another extension is to look at the relation with other airport planning and scheduling problems. In itself, the deicing problem as formulated in the formal model of Sect. 3 is not that exceptional. What makes the problem interesting to look into is its relation to other planning problems, possibly involving other planning agents.

References

1. Smith, D.E., Frank, J., Jónsson, A.K.: Bridging the gap between planning and scheduling. Knowl. Eng. Rev. 15(1), 47–83 (2000)
2. Nisan, N., Ronen, A.: Algorithmic mechanism design. In: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC'99), pp. 129–140. ACM Press, New York (1999)
3. Andelman, N., Azar, Y., Sorani, M.: Truthful approximation mechanisms for scheduling selfish related machines. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, pp. 69–82. Springer, Heidelberg (2005)
4. Auletta, V., Prisco, R.D., Penna, P., Persiano, G.: Deterministic truthful approximation mechanisms for scheduling related machines. In: Diekert, V., Habib, M. (eds.) STACS 2004. LNCS, vol. 2996, pp. 608–619. Springer, Heidelberg (2004)
5. Kovács, A.: Fast monotone 3-approximation algorithm for scheduling related machines. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 616–627. Springer, Heidelberg (2005)
6. Angel, E., Bampis, E., Pascual, F.: Truthful algorithms for scheduling selfish tasks on parallel machines. Theor. Comput. Sci 369(1-3), 157–168 (2006)
7. Immorlica, N., Li, L., Mirrokni, V.S., Schulz, A.: Coordination mechanisms for selfish scheduling. In: Deng, X., Ye, Y. (eds.) WINE 2005. LNCS, vol. 3828, pp. 55–69. Springer, Heidelberg (2005)
8. Vermeulen, I., Bohte, S., Somefun, D., Poutré, J.L.: Improving patient schedules by multi-agent pareto appointment exchanging. In: Proceedings of 2006 IEEE International Conference on E-Commerce Technology (CEC/EEE 2006), San Francisco, California, June 26-29, p. 9 (2006)
9. Paulussen, T.O., Jennings, N.R., Decker, K.S., Heinzl, A.: Distributed patient scheduling in hospitals. In: IJCIA-03, pp. 1224–1232. Morgan Kaufmann, San Francisco (2003)
10. Attanasio, A., Ghiani, G., Grandinetti, L., Guerriero, F.: Auction algorithms for decentralized parallel machine scheduling. Parallel Comput. 32(9), 701–709 (2006)
11. Lewin, R.: Embracing Complexity: Exploring the Application of Complex Adaptive Systems to Business. Ernst & Young (1996)
12. Parkes, D.C., Ungar, L.H.: An auction-based method for decentralized train scheduling. In: Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada, pp. 43–50. ACM Press, New York (2001)
13. 't Hoen, P.J., Poutre, J.A.L.: A decommitment strategy in a competitive multi-agent transportation setting. In: AAMAS '03, pp. 1010–1011. ACM Press, New York (2003)
14. Sandholm, T., Lesser, V.: Leveled commitment contracts and strategic breach. Games and Economic Behaviour 25, 212–270 (2001)
15. Collins, J., Tsvetovas, M., Sundareswara, R., van Tonder, J., Gini, M., Mobasher, B.: Evaluating risk: flexibility and feasibility in multi-agent contracting. In: Proceedings of the Third International Conference on Autonomous Agents (Agents'99), Seattle, WA, USA, pp. 350–351. ACM Press, Seattle, WA (1999)