EFFICIENCY AND FAIRNESS IN AIR TRAFFIC CONTROL[1]

Geert Jonker John-Jules Meyer Frank Dignum

Utrecht University

Abstract

Air Traffic Control Planning is a complex area of research in which there is a great need for new and efficient coordination techniques. The tight connection between several parties with different interests makes it a typical and challenging area for the application of multiagent techniques. We study the relation between two important underlying principles in airport traffic planning, namely utility and fairness. We model the problem as a multiagent resource allocation problem and show how one can improve on global utility and fairness if planning history is involved. We introduce three techniques using history and evaluate their performance by experiments.

1 Introduction

Airports nowadays are more and more faced with air traffic congestions as a result of increased capacity demands. Much effort is being put into the development of software tools to assist the air traffic controllers in their decision-making process. There is a growing awareness nowadays that to achieve more efficiency, distributed and autonomous techniques are needed [6]. In this context several initiatives are coming forward, such as the Free Flight concept [8], in which aircraft coordinate their path of flight between themselves, and the Collaborative Decision Making concept [3] which is aimed at involving the information and interests of many parties in the decision-making process. In general we can say that there is a need for *multiagent techniques* in air traffic control (ATC), where autonomy and initiative is shifted from the air traffic controllers to the executing parties, based on individual preferences of the agents involved.

ATC planning is historically concerned with *safety* and *punctuality*. However, we will argue that *fairness* is a very determining and thus important factor as well. When designing multiagent techniques for ATC, one should capitalize on this insight by handling fairness explicitly. We will show that this leads to an increase of utility.

Within the multiagent community, *multiagent resource allocation* is a term used to denote allocation problems that occur in a multiagent context. Our problem shares a lot of features with this field of research. We will therefore model it as a multiagent resource allocation problem.

In the remainder of this article, we will describe the problem of *ATC tactical planning*, our point of focus. We will approach it as a multiagent resource allocation problem, after which we will introduce three decision methods and measure their performance by experiment.

2 Air Traffic Control Tactical Planning

In Air Traffic Control, tactical planning is the phase between *strategic planning* and execution. In strategic planning every flight is assigned departure and arrival slots, flight routes and gates. After the strategic planning has been finalized, disruptions such as delays and technical problems can invalidate the planning. The task of tactical planning is to repair the planning as efficient as possible, i.e. with as little delays and schedule changes possible. These repairs can be done to up to three minutes before execution.

[1]This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs. Project DIT5780: Distributed Model Based Diagnosis and Repair.

Tactical ATC planning is mainly a manual process nowadays. Last-minute plan changes are made using pen and paper, looking out of the tower window, making telephone calls, etc. To be able to deal with disruptions quickly, air traffic controllers use strict rules of thumb. For example, *first come first serve* is a rule often used for aircraft that want to depart or have missed their arrival or departure slot; they are directly put at the end of the queue of aircraft waiting to be scheduled. Another example are the *ground delay programs*: if an airport is at its maximum capacity, all incoming aircraft are delayed an equal amount of time [1].

These rules are very important in current ATC practise. In fact, air traffic controllers can hardly be persuaded to deviate from them, although it is sometimes more efficient. These rules show an important underlying principle in ATC planning: *fairness*. The first-come-first-serve rule and ground delay programs provide some assurance that different airlines will be treated equally. If air traffic controllers wouldn't abide to this, they risk protests of airline companies who feel that they are treated unfairly.

An important implication is that air traffic controllers would be willing to deviate from their rules of thumb if fairness could be guaranteed in another way. In this context multiagent technology could provide a solution. If tactical planning was to be done by agents, they would negotiate among each other about plan changes. When doing this, they could explicitly compare each others utilities and adapt their behaviour accordingly to achieve fairness. In this way they could deviate from the rules of thumb, allowing them to achieve higher efficiency. The key here is that agents have enough computational capacity to do the administration needed for fairness while a single human air traffic controller has not.

3 Multiagent Resource Allocation

Multiagent resource allocation can be described as the overlap between two fields of research: that of classical resource allocation as researched in Economics and Social Sciences, and that of multiagent systems. In classical resource allocation the central question is "*what* is the best allocation of a number of items over a number of parties?" In multiagent systems, because of the autonomous and pro-active behaviour of the parties involved, the question becomes "*how* can we find the best allocation of a number of items of a number of agents?" A good overview of the field of multiagent resource allocation can be found in [4].

There are many criteria by which to judge the quality of an allocation. *Utilitarian welfare* or *efficiency* is a term used to denote the sum of the utilities of the agents[2]. *Egalitarian welfare* denotes the utility of the agent that is worst off. Thus, maximizing egalitarian welfare is primarily aimed at achieving fairness.

We have seen that efficiency and fairness are important properties in the ATC case. Although ATC tactical planning is not a typical allocation problem, it can be viewed as one in the following way. In case of a disruption, an online planner generates a set of possible repairs. These are the resources, out of which one resource has to be chosen and enforced, which has consequences for all agents involved. Note that this resembles resource allocation with *externalities*, where the allocation of a resource to an agent has consequences for other agents as well.

Furthermore, ATC tactical planning poses an interesting addition to the field of multiagent resource allocation problems: that of *repeated resource allocation*. Tactical ATC planning is a continuous process, where the aim is not to achieve efficiency and fairness over a single allocation, but over a series of subsequent allocations. This has a great impact on the decision function to be used. To the best of our knowledge, there hasn't been a lot of research done on this subject.

To see which notion of fairness is best suited for the ATC domain, note that maximizing the quantitative notion of egalitarian social welfare isn't exactly right for this case; sometimes it is justifiable to delay an aircraft so that many others can be in time. Therefore we introduce a new quantitative notion of fairness called *σ-unfairness*, which is defined as the standard deviation in utilities of the agents. We use this criterion because minimizing σ-fairness achieves fairness, but allows an agent to receive a low utility if this is beneficial for many others.

It is well known that perfect efficiency and fairness are incompatible in many domains [2, 7]. Often a compromise has to be made between the two. In fact this is what air traffic controllers

[2] In the remainder of this article we will adhere to this definition of efficiency.

now do, using their common sense. An automated decision-making system should capture the same kind of behaviour; achieve a reasonable amount of efficiency while the planning remains reasonably fair. The fact that we consider sequences of allocations has its consequences for the measurement of efficiency and fairness. We will see this in the next section where we model the problem formally.

4 Efficiency and Fairness

The aim of *repeated resource allocation* is to perform a finite but arbitrarily large number of resource allocations subsequently, where the performance criteria are measured over all allocations. In our case every allocation problem consists of a set of candidate allocations out of which one has to be chosen. The agents have preferences over all candidate allocations, and the criteria that have to be met are efficiency and fairness over all allocations.

Efficiency is defined for a single allocation a or a history of allocations $H = \{a_1, a_2, \ldots, a_n\}$ by

$$\text{eff}(a) = \sum_{i=1}^{k} u_i(a) \qquad \text{and} \qquad \text{eff}(H) = \sum_{i=1}^{k} u_i(H)$$

respectively, where $u_i(a)$ is the utility that agent i ascribes to a and $u_i(H)$ is the utility of agent i for history H, which is the sum of its utilities over the allocations in H. We say that from a set of candidate allocations $C = \langle c_1, c_2, \ldots, c_n \rangle$ a candidate allocation is *optimal* if it is the most efficient allocation. It is easy to see that a solution $H = \langle a_1, a_2, \ldots, a_n \rangle$ to a sequence of allocation problems $S = \langle C_1, C_2, \ldots, C_n \rangle$ is optimal if every allocation a_x is optimal for C_x. Thus, we can achieve maximal efficiency by choosing the most efficient allocation in every round.

The same doesn't hold for fairness. If we don't allow side-payments, i.e., transfer of utility between agents, and assume future candidate sets to be unknown, there is no mechanism that is guaranteed to find the maximally fair allocation. To see this, consider the following example. In a two agent scenario, we have a two sequence allocation problem $\langle \langle (2,3), (5,4) \rangle, \langle (6,7) \rangle \rangle$ and $\langle \langle (2,3), (5,4) \rangle, \langle (7,6) \rangle \rangle$, where we have identified allocations with their utilities to the respective agents. Note that in both cases, in round two there is only one candidate allocation. Because the options in round two are not known in round one, a decision mechanism will make the same choice in round one of both sequences. To achieve egalitarian or σ-fairness however, the mechanism should choose $(5,4)$ in the first sequence but $(2,3)$ in the second. This is contradictory, therefore such a mechanism cannot exist.

With perfect fairness being impossible, the question becomes how to maximize fairness. The most straight forward method is to choose in every round the allocation that maximizes fairness over the sequence of allocations up until then. Although this method doesn't guarantee maximal overall fairness, it is the best you can do if you don't know anything about future candidate allocations. We will therefore use it as a benchmark in our evaluation.

The central question we want to address in this paper is how to achieve a good balance between efficiency and fairness in the repeated resource allocation. We introduce three methods that hold the middle between an efficiency maximizer and a fairness maximizer. To achieve fairness, one can say that agent with low utility should be *compensated* somewhere in the future. It is to be expected that some candidate sets provide a better opportunity for compensation than others. The idea is to determine every round whether the candidate set provides a good opportunity for compensation or for efficiency. We do this by scoring the candidates according to these two criteria and choosing the winner by a weighted sum of these scores. The weights can be adjusted as to set the desired ratio between efficiency and fairness.

5 Formal description

In this section we formally describe a number of decision methods which we will compare by experiment. First we describe three methods that we will use as benchmark functions. The *efficiency* *maximizer* chooses in each round the candidate with the highest utility:

$$winner_{EFF} = \max_a \{\text{eff}(a)\}$$

As we mentioned before, choosing the most efficient candidate in each round achieves the optimal allocation sequence over all decision problems. The *fairness maximizer* chooses in each round the most fair solution:

$$winner_{FAIR} = \min_a \{\sigma\text{-unf}(A \bullet \{a\})\}$$

where $A \bullet \{a\}$ is the concatenation of history A and $\{a\}$, and σ-unf(H) is a function yielding the standard variation over the utilities the agents have after the allocations in H.

As we mentioned before, this doesn't necessarily yield the most fair allocation over a sequence of decision problems, but we assume that one can't do much better than this when nothing is known about future candidate sets. The third benchmark we will use is what we will call the *standard solution*. This method corresponds roughly to what an air traffic controller does now: choosing a reasonably efficient and fair solution without considering past events. To determine the balance between fairness and efficiency we use a slider value s, which should correspond with how much the air traffic controller values efficiency compared to fairness:

$$winner_{STD} = \max_a \{s * \text{eff}(a) + (1 - s) * -1 * \sigma\text{-unf}(\{a\})\}$$

We will test three methods of involving history in the decision making function. As we explained before, we use the *compensation potential* of a candidate in the decision. Given a candidate allocation a and a history A, it is defined as follows:

$$\text{cp}(a, A) = \sigma\text{-unf}(A) - \sigma\text{-unf}(A \bullet \{a\})$$

The first decision function involving history resembles the standard function, but uses the history of allocations in its calculation of the compensation potential:

$$winner_{HIST} = \max_a \{s * \text{eff}(a) + (1 - s) * \text{cp}(a, A)\}$$

We are also interested to know whether it is possible to put a maximum on the unfairness without too much loss in efficiency. It would be useful to have a mechanism which is able to guarantee agents that the situation will not become arbitrarily unfair. We propose two methods that use a *threshold value* below which the unfairness should be kept. The first one simply selects from all the candidates those candidates that will not let the unfairness go above the threshold value, and selects from those the most efficient one:

$$winner_{THRES1} = \max_a \{\text{eff}(a) \mid \sigma\text{-unf}(A \bullet \{a\}) < threshold\}$$

The second one is an adaptation of the HIST-function. It uses a *dynamic slider*. As soon as the unfairness of the allocations is above the threshold value, the value of the slider is lowered in order to select fairer solutions. When the unfairness is below the threshold value, the slider can be increased again to select more efficient solutions.

$$winner_{THRES2} = \max_a \{s * \text{eff}(a) + (1 - s) * \text{cp}(a, A)\}$$

with the rules

```
if unfairness > threshold then decrease s
if unfairness < threshold then increase s
```

6 Experimental results

In this section we will describe and give the results of the implementation of the decision methods described in previous sections. An important part of the experiment is also to measure the effect of the actual distribution of the input values on the performance of the decision functions, as we

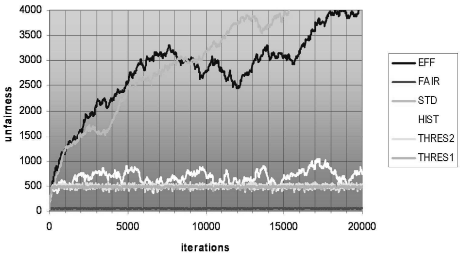

Figure 1: Fairness in the scenario with normal distributions.

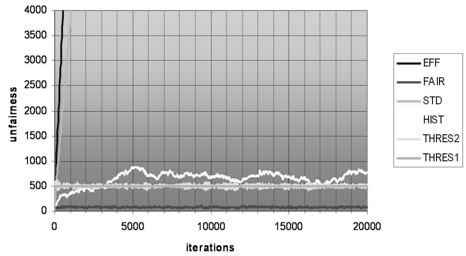

Figure 2: Fairness in the scenario with biased normal distributions.

expect it to have quite some impact. We have chosen to test the functions against increasingly realistic distributions. First we have taken a scenario where the utilities of the agents are normally distributed, i.e., the utility that an agent ascribes to a candidate is usually close to a certain value. Our second scenario is chosen such as to reflect a property existing in many real life domains, that of *unequal possibilities* (cf. [5]). In the ATP domain for instance, different airlines will have different preferred gates assigned to them. This makes some airlines subject to plan changes more often than others. We model this property by biasing the utility functions of agents; although the utilities are still normally distributed, different agents will have different average utilities over the allocation candidates.

In the third scenario we have chosen to reflect another realistic property. Often in ATC planning, a problem can be solved either by making one agent very unhappy, or making many agents a bit unhappy. For instance, an incoming flight that is delayed can either be inserted in the arriving queue directly by delaying all subsequent flights a bit, or it can be inserted in the first occurring gap in the queue which can still be far away. We have simulated this by choosing the candidate sets in such a way that there are both candidate allocations where many utilities are zero and few are very low, and candidate allocations where all utilities are a bit low. We will call this *clustered utilities*.

We have tested the six described methods in all three domains. As evaluation criteria we take the average utility per agent and σ-fairness. In our test bed we have chosen to use values that are realistic in the ATC scenario. We assume that on a medium airport there are 30 airlines involved in the schedule. In every plan change situation, the planner generates 25 possible solutions. From these one is chosen and enforced in every round. As we have performed 20000 iterations in every run, if the mechanism is used 200 times every day at an airport our experiments show the performance over several months.

In the first scenario we have chosen to let the input values be normally distributed with a mean of -30 and a standard deviation of 40. This reflects the fact that plan changes are usually inconvenient to airlines, but might sometimes be beneficial. All decision methods caused the average efficiency per agent to become stable after many iterations. The results are shown in table 1, along with the results for the other scenario's. Figure 1 shows the unfairness of the allocations for the scenario with normal distributions. It can be seen that EFF achieves an efficiency of about -15, but lets the unfairness grow arbitrarily large. FAIR achieves a fairly constant maximal unfairness, but does nothing for efficiency, since -30 is the average of the input values. Remember that EFF and FAIR show the maximum possible efficiency and fairness respectively. STD shows an efficiency level in between those of EFF and FAIR. Its unfairness however grows arbitrarily large, worse even than that of EFF. HIST, THRES1 and THRES2 all perform quite well, achieving about the same efficiency as EFF with an unfairness between 0 and 1000.

input characteristics	EFF	FAIR	STD	HIST	THRES1	THRES2
normally distributed	-15.5	-29.7	-23.6	-15.5	-16.0	-15.7
biassed	-19.4	-29.1	-25.6	-22.3	-24.6	-23.2
clustered	-0.6	-2.0	-1.2	-0.7	-1.4	-0.9

Table 1: Efficiency of the different decision methods in different scenario's.

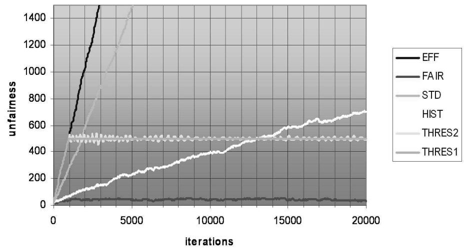

Figure 3: Fairness in the scenario with clustered utilities.

In the third scenario we have chosen to bias agents' utilities by varying the means and deviations per agent. Table 1 and figure 2 show that this poses a more difficult challenge to the decision functions. While EFF and STD quickly let the allocations become very unfair, THRES1 and THRES2 have to lower their efficiency considerably to stay below the threshold. HIST manages to achieve a reasonable efficiency and fairness.

In the fourth scenario, candidate allocation sets include both allocations with few involved agents and with many. The results can be seen in table 1 and 3. The most important observations here are the fact that HIST shows linear behaviour in fairness and that THRES1 performs relatively bad on efficiency.

7 Evaluation and conclusion

The experiments show a number of things. First, involving history in the decision function greatly improves the efficiency and fairness of the mechanism; HIST, THRES1 and THRES2 all perform better than STD. Also, the actual distribution of the input values has a great impact on the performances of the system. In the first domain the decision functions EFF, THRES1 and THRES2 all performed near maximum efficiency, with reasonable unfairness. In the second and third domain however, where the biases of the agents differed, the methods showed some differences. Here HIST outperformed THRES1 and THRES2 on efficiency, but showed greater unfairness. In the scenario with clustered utilities the experiment showed that when a threshold on the unfairness is desired, THRES2 is a better choice than THRES1.

Some remarks that are not evident from the graphs should be made. First, the amount of biassing greatly influences fairness. If the biassing is done more extreme, it can be the case that fairness can no longer be maintained. In that case, even FAIR will let the unfairness grow arbitrarily large. Secondly, the choice of the slider value for HIST is very important. It can be chosen such that the function behaves more like EFF or more like FAIR. In fact, in our experience it can be chosen such that the algorithm performs better than THRES1 and THRES2, i.e., has better fairness with equal efficiency. The only problem here is to correctly estimate this value, which is not easy to do beforehand.

It should also be noted that there are probably better ways of controlling the dynamic slider value than the one we used in THRES2. Further refinement could thus improve the performance of THRES2.

Summarizing, we can say that repeated resource allocation is a challenging problem which occurs in air traffic control. We have argued that efficiency and fairness are important criteria in this domain. We have proposed a number of decision mechanisms involving history and shown that they perform better compared to what is being used now. It is even possible to put a limit on the unfairness, while efficiency is kept reasonably high.

The experiments showed that the actual distribution of the utility of the agents can have an impact on the performances of the methods. When biasses are being used, it is much more difficult to guarantee fairness. Overall, HIST and THRES2 proved to be the best decision functions.

References

[1] M.O. Ball, R. Hoffman, D. Knorr, J. Wetherly, and M. Wambsganss. Assessing the benefits of collaborative decision making in air traffic management. In *Proceedings of 3rd USA/Europe air Traffic Management R&D Seminar*, Napoli, Italy, 2000.

[2] Steven J. Brams, Paul H. Edelman, and Peter C. Fishburn. Fair division of indivisible goods. *Theory and Decision*, 55(2):147–180, 2003.

[3] Chien-yu Chen, Michael O. Ball, Robert Hoffman, and Thomas Vossen. Collaborative decision making in air traffic management: Current and future research directions. Technical report, October 04 2000.

[4] Y. Chevaleyre, P. E. Dunne, U. Endriss, J. Lang, M. Lematre, N. Maudet, J. Padget, S. Phelps, J. A. Rodrguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. www.doc.ic.ac.uk/˜ue/MARA/mara-may-2005.pdf.

[5] Marc Fleurbaey and Francois Maniquet. Fair allocation with unequal production skills: The no envy approach to compensation. *Mathematical Social Sciences*, 32:71–93, 1996.

[6] IATA. IATA position on the global Air Traffic Management operational concept and the need for an ATM global implementation roadmap. Eleventh Air Navigation Conference, Montreal, 2003.

[7] M. Lematre, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver. Equitable allocation of earth observing satellites resources. In *Proc of 5th ONERA-DLR Aerospace Symposium (ODAS'03)*, Toulouse, France, June 2003.

[8] R.L. Schultz, D.A. Daner, and Y. Zhao. Free-flight concept. Technical Report AIAA-97-3677, AIAA, 2000.