# Air-Crew Scheduling through Abduction

A.C. Kakas and A. Michael

Department of Computer Science, University of Cyprus,
P.O.Box 537, CY-1678 Nicosia, CYPRUS
Email: {antonis, antonia}@cs.ucy.ac.cy

**Abstract**. This paper presents the design and implementation of an air-crew assignment system based on the Artificial Intelligence principles and techniques of abductive reasoning as captured by the framework of Abductive Logic Programming (ALP). The aim of this work was to produce a system for Cyprus Airways that can be used to provide a solution to the airline's crew scheduling problem whose quality was comparable with the manual solutions generated by human experts on this particular problem. In addition to this the system should also constitute a tool with which its operators could continually customize the solutions to new needs and preferences of the company and the crew. The abductive approach (using ALP) adopted in our work offers a flexible modeling environment in which both the problem and its constraints can be easily represented directly from their high-level natural specification. This high-level representation offers two main advantages in the development of an application: (i) *modularity* with a clear separation of the two main issues of validity and quality (optimality) of the solution and (ii) *flexibility* under changes of the requirements of the problem.

## 1. Introduction

Abduction is the process of reasoning to explanations for a given observation according to a general theory that describes the domain of observation. In Artificial Intelligence abduction manifests itself as an important form of inference for addressing a variety of problems (eg. [14, 16, 10, 5, 2]). These problems include, amongst others, reasoning with incomplete information, updating a database or belief revision and formalizing and addressing application problems like that of planning, diagnosis, natural language understanding and user modeling. Several applications of abduction exist (eg. [10, 4, 3, 13, 17]) and we are now beginning to see the emergence of the first "engineering principles" for the development of abductive applications.

In this paper we use the framework of Abductive Logic Programming (ALP) [10, 5] to develop an abductive-based system for the application of air-crew scheduling. The problem of air-crew scheduling (e.g. [1, 9]) is concerned with the assignment of air-crews to each of the flights that an airline company has to cover over some specific time period. This allocation of crew to flights has to respect all the necessary constraints (validity) and also minimize the crew operating cost (quality or optimality). The validity of a solution is defined by a large number of complex constraints, which express governmental and international regulations, union rules, company restrictions etc. The quality of the schedule is specified by its cost, but also by the needs and preferences of the particular company or crew.

We present an air-crew scheduling system based on abduction, developed for the application of Cyprus Airways Airlines. The aim of this work was to produce a system that can be used to provide a solution to the airline's crew scheduling problem whose quality was comparable with the manual solutions generated by human experts. In addition to this the system should also constitute a tool with which it will be possible for its operators to continually customize the solutions to new unexpected requirements while the solution (crew-roster) is in operation or additional foreseen needs and preferences of the company and the crew. The system therefore can be used, not only for the production of a valid and good quality initial schedule, but also as a rescheduling tool that can help human operators to adjust an existing solution to the necessary changes.

Various approaches, within OR and AI (e.g. [8, 15, 1, 9]), have been adopted for solving the crew scheduling problem. Although these methods have succeeded in solving the problem efficiently, for specific problem cases, they lack a general and simple way of specifying the problem its constraints. Moreover, these methods lack flexibility and their adaptation to changes in airline company rules is not very practical. Recent Constraint Logic Programming based approaches (e.g. [8, 15, 7]) have succeeded in providing a more declarative way of describing scheduling problems, but these still have to be represented directly in terms of the special domain constraints which can be handled by the constraint solver. Thus again the representation of the problem looses some of its flexibility.

The rest of the paper is organized as follows. Section 2 introduces the problem of air-crew scheduling and rescheduling. In section 3, we give a brief overview of the ALP framework and how this can be used in applications. Section 4 presents the architecture of the air-crew scheduling system. In section 5, we present some of the experiments carried out to evaluate this system. The paper concludes, in section 6. An extended version of this paper [12] can be obtained from the authors.

## 2. Air-Crew Scheduling

Air-Crew Scheduling is the problem of assigning cockpit and cabin crews to the flights that a company has to carry out over some predefined period of time. The flight schedule of a company consists of flight legs, which are non-stop flights between two airports, for a predefined period (usually a month). Given the flight schedule, the crew scheduling task is concerned with the allocation of crew personnel to each flight leg, in a way that minimizes the crew operating cost.

The crew scheduling problem is extremely difficult and combinatorial in nature due to the large number and the complexity of the constraints involved, e.g. governmental and international regulations, union rules, company restrictions etc.
An abductive approach (e.g. using Abductive Logic Programming - ALP) offers a flexible modeling environment in which both the basic problem and its constraints can be easily represented directly from their high-level natural specification. It then provides an effective automatic translation (reduction) of this high-level representation to the lower-level computational goals (constraints) which need to be solved. In this way ALP offers the possibility to combine the advantages of modularity and flexibility with efficient constraint solving.

## 2.1. The Problem Domain

Let us first introduce some of the basic terms used in air-crew scheduling.

*flight leg :* The smallest planning unit and it's a non-stop flight (link) between two airports. *pairing or rotation :* A sequence of flight legs which could be taken by a single crew member.
*flight duty time (FDT) :* The sum of the flight time of a rotation, the time spent on the ground in between two succeeding legs and the time required for briefing and debriefing of the crew.
*deadhead (DH) :* The positioning (by another plane) of crew, as a passenger, to another airport in order to take a flight assignment.
*transfer :* The transferring of crew via road (by bus or car) from one airport to another in order to take an assignment.
*day-off :* A full non-working day, with some additional requirements on the time which the duty of the previous day ends or on the beginning of the next day.
*rest period :* A continuous period of time between two duties when the crew has no assignments.
*stay-over (S/O):* A rest period that a crew spends away from base. Stay-Overs are necessary because of the upper bounds which exist on the duty period of a crew.
*stand-by :* A period of time when the crew has to be near the base and ready to take an assignment, if necessary, within one hour's notice.
*available (AV) :* A crew member is said to be available on a day that is not off and on which the crew has no duty.

The input to the crew scheduling problem is the flight schedule of an airline company for a specific period (usually a month). The flight schedule consists of all the flight legs that the company has to cover in that period. Given the flight schedule, together with the necessary information about each available crew member (e.g. their name and rank), the required output is a set of valid and good quality, if possible optimal, (monthly) rosters for each member.

The **quality** (or optimality) of the solution depends on three major issues: cost, fairness and preferences. A good solution should minimize the crew operating cost. Among the main factors which can result in high-cost schedules are: the number of DHs, transfers, S/Os and the total number of duty hours, over some specific period, of a crew (overtime rate is paid if this number exceeds a certain limit). A second factor for quality is the fairness of the solution on the distribution of the flying hours, type of allocated flights, DHs, S/Os, stand-bys, availability, day-offs etc. Another important issue for the quality of a solution is the extent in which the (crew or company) preferences are satisfied. This issue can sometimes also determine the validity of a solution.

The **validity** of the rosters is defined through a number of complex rules and regulations which can be grouped in classes according to the type of information that they constrain. Some of the most common groups of constraints in air-crew scheduling are the *temporal* and *location* constraints, the *FDT* constraints and the constraints on the *rest* and *working times* of the crew. A number of other constraints which express specific requirements, or preferences, of the particular application, or rules of the specific airline company or union may also be necessary for valid or better quality solutions.

### 2.2. Rescheduling - Changes

In the domain of air-crew scheduling changes occur very frequently either due to flight delays or cancellations, new flight additions, or because of crew unavailability. It is common that such changes happen on the day of operation, in which case they have to be dealt with immediately. Also apart from these changes which happen unwillingly, a number of other changes may often be voluntarily made by the crew administration on the solution found initially, in order to reflect special preferences and thus make the final solution more desirable. For these reasons the automation of the rescheduling process is crucial in the crew scheduling problem.

In general, the changes which are required fall in one of the following categories.

- Change the crew of a flight.
- Reschedule a flight, when the times of a flight are changed (e.g. delayed).
- Cancel a flight.
- Add a new flight.

These changes may violate the validity of the old solution or make the old schedule of poorer quality. We want to reestablish validity or quality without having to discard the old solution and recompute from the beginning, but rather by keeping the unaffected part unchanged and by rescheduling only those assignments which may be affected. In air-crew scheduling it is also necessary that any change can be accommodated by changing the old schedule within 48 hours with the fewest possible changes on the old assignments of crew members.

## 3. The ALP Approach to Applications

Abduction is the process of reasoning to explanations for a given goal (or observation) according to a general theory that describes the problem domain of the application. The problem is represented by an abductive theory. In Abductive Logic Programming (ALP) an abductive theory is defined as follows.

**Definition 3.1 (Abductive Theory)**
An abductive theory in ALP is a triple <P,A,IC> where P is a logic (or constraint logic) program, A is a set of predicate symbols, called abducibles, which are not defined (or are partially defined) in P, and IC is a set of first order closed formulae, called integrity constraints.

In an abductive theory <P,A,IC>, the program P models the basic structure of the problem, the abducibles play the role of the answer-holders, for the solutions to particular tasks (goals) in the problem, and the integrity constraints IC represent the validity requirements that any solution must respect. A goal G is a logic (or constraint logic) programming goal. A solution to a goal G is an abductive explanation of G defined as follows.

**Definition 3.2 (Abductive Solution)**
An abductive explanation or solution for a goal G is a set $\Delta$ of ground abducible formulae which when added to the program P imply the goal G and satisfy the integrity constraints in IC, i.e.

$P \cup \Delta \models_{lp} G$ and $P \cup \Delta \models_{lp} IC$

where $\models_{lp}$ is the underlying semantics of Logic Programming or Constraint Logic Programming. More generally, the solution set $\Delta$ may also contain existentially quantified abducibles together with some constraints on these variables (see [11]).

The computational process for deriving the abductive solution (explanation) consists of two interleaving phases, called the abductive and the consistency phases. In an *abductive phase*, hypotheses on the abducible predicates are generated, by reducing the goals through the model of the problem in P, thus forming a possible solution set. A *consistency phase* checks whether these hypotheses are consistent with respect to the integrity constraints. During a consistency phase it is possible for new goals to be generated, if these are needed in order to ensure that the hypotheses so far can satisfy the integrity constraints. In turn these new goals can generate further abducible assumptions, to be added to the solution set. It is also possible that the consistency phase refines the solution set of assumptions generated originally - by setting constraints on the existential variables involved in the abducible assumptions - when this restriction can help ensure the satisfaction of (some of) the integrity constraints.

# 4. The Air-Crew Scheduling System

### 4.1. The System Architecture

A crew scheduling system has been developed using the ALP approach and applied to the real-life problem of the airline of Cyprus Airways. The general structure of the system is shown in Figure 4.1.

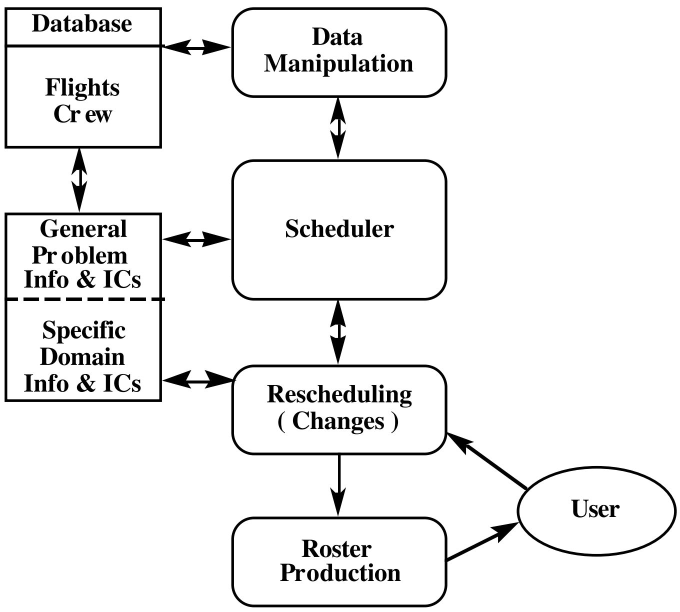


Figure 4.1 : Air-Crew Scheduling System Architecture

The Database holds information about all available crew members and the flight schedule of the company, which consists of all the flight legs that have to be covered for a predefined period (usually a month). Given the flight schedule of an airline, a preprocessing phase ("Data Manipulation") constructs possible pairings of flight legs which can be assigned to a single crew without exceeding the given maximum flight duty time (FDT). In FDT we include 1hr for briefing and 1/2hr for debriefing of crew after each leg, plus the time spent on the ground between the succeeding legs. One such pairing, in the Cyprus Airways problem, might for example be the "Larnaca→Athens→Larnaca" rotation. Other flight legs which cannot be paired in one duty remain as "single" flights.

In the main "computational" part of the system, the "Scheduler", flights (pairings or single) are assigned, via the process of abductive reasoning, to the selected crew members. The output of this phase is a consistent set of assignments of flights, deadhead flights, transfers, stay-overs and day-offs to crew members. The required solution should cover all flight legs exactly once and it should be of good quality (e.g. with low cost, balanced etc.). This set of assignments will go through the "Roster Production" phase which will then extract all the information needed, like for example the (monthly) roster for each crew member, the total flying and duty hours of each crew for that period, the crew available on some specific day etc.

### 4.2. The Scheduler

Given the set of flights (pairings or single), generated in the preprocessing ("Data Manipulation") phase, the scheduler is concerned with the assignment of crew members to each of these flights and to other related duties. Figure 4.2 shows a graphical representation of the model of this problem, captured by the program P of the overall ALP theory <P, A, IC> which represents the whole problem. With this modeling of the problem there is a need for only one abducible in A, which *assigns* crew members to different types of duty tasks (e.g. flights, stand-bys, day-offs etc.). Therefore the solution (schedule) will consist of a set of assignments of crew to tasks, in a way that satisfies the integrity constraints IC on the problem.

One natural feature, which is represented in the basic model of this problem in the program P, is that the flights on consecutive days are more related among them than with flights on other days, in the sense that many constraints involve flights which are temporally near to each other. Furthermore, an even stronger dependency exists between flights which are on the same day. This observation suggests that a better (computationally) modeling of the problem would be one where the flights are scheduled in increasing date order and, within the same day, in increasing order of departure time.

Scheduling the flights per day and in chronological order can also facilitate the derivation of better quality schedules, in the sense that certain operations ("Day Operations"), which improve the solution, can be performed at the end of each day. For example, consider the case of deadhead crew which has to return to base after some number of stay-overs (depending on his/her location). This crew has to be assigned as a passenger on a return (deadhead) flight on some specific day. Other additional assignments which are required, on a daily basis, are assignments of crew to transfers, stand-bys and day-offs.

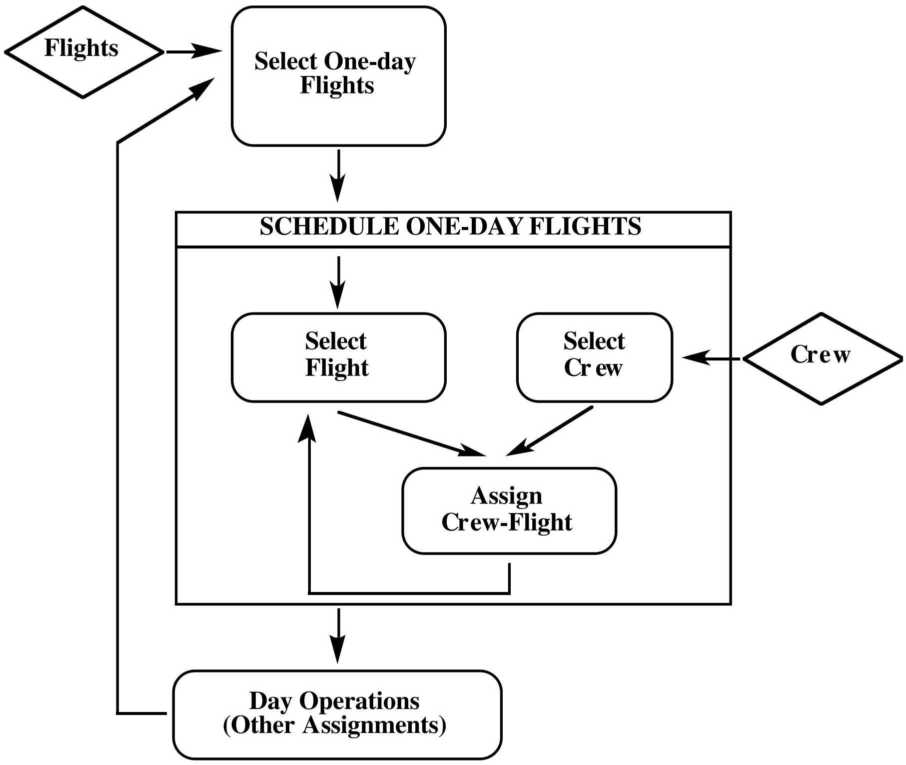


Figure 4.2 : The Scheduler Module

Another important quality of solution issue is that of balancing. A good schedule should be fair, or well balanced, at several levels. One such level is the balancing of the flying and duty hours for each crew member, another would be the balancing on the stand-bys, balancing on the stay-overs, or even fairness on the flight destinations of each crew. These issues can be treated incrementally in the model by adopting different strategies (e.g. load balancing algorithms) in the selection of the crew members.

**The Constraints**

One of the characteristics of the crew scheduling problem is the large number and the variety of the constraints to be considered. We present here a small representative sample of these, as they are included in the integrity constraints IC of the ALP formulation of the problem, that demonstrate the concepts proposed by this approach.

Temporal constraints

One hard temporal constraint necessary for valid solutions is that no crew member can be assigned two flights at the same time. This is represented as an integrity constraint, by the logical formula in IC,

$\neg$ ( assign(Crew,FlightA), assign(Crew,FlightB),
FlightA $\neq$ FlightB, overlap(FlightA,FlightB) ).

where "overlap" is defined in the program P by

overlap(FlightA,FlightB) $\leftarrow$
flight_details(FlightA,DeptDateA,DeptTimeA,ArrDateA,ArrTimeA),
departure(FlightB,DDB,DTB),
between(DeptDateA,DeptTimeA,DDB,DTB,ArrDateA,ArrTimeA).

The predicate "between" holds when the departure date and time of one flight is in the flight period of the other flight. This will generate, when this integrity constraint is reduced in the consistency phase of the computation, equality and inequality constraints on the values of the two variables "FlightA" and "FlightB".

Duty Period
The duty period is a continuous block of time when a crew member is on duty. The maximum duty period of a crew cannot exceed a certain amount (MaxDuty), which is (usually) less than or equal to 12 hours and it is calculated according to this crew's previous assignments. This is represented by the following integrity constraint in IC.

$\neg$ ( assign(Crew,Flight),
flight_period(Flight,Period), departure(Flight,DeptDate,DeptTime),
MaxDuty(Crew,MD), DutyPeriod is MD-Period,
subtract(DeptDate,DeptTime,DutyPeriod,NewDate,NewTime),
on_duty(Crew,NewDate,NewTime) ).

The predicate "on_duty" is defined in the model of the problem in the program P and decides if a crew is on duty on a particular date and time. Note that again the reduction of the predicates "on_duty", "between" and "between_flights" will generate a number of arithmetic constraints on the date and time of the flight that can be assigned to the selected crew.

Rest Period
There must be at least MinRest hours rest period between any two consecutive duties. MinRest is greater than or equal to 12 and it is calculated according to the previous assignments of the crew. This integrity constraint is given by the following logical formula in IC :

$\neg$ ( assign(Crew,Flight),
departure(Flight,DeptDate,DeptTime),
on_new_duty(Flight,DeptDate,DeptTime),
end_prev_duty(Crew,DeptDate,DeptTime,EndOfDuty),
time_difference(EndOfDuty,DeptDate,DeptTime,RestPeriod),
MinRest(Crew,MR), RestPeriod < MR ).

where "on_new_duty" defines whether a flight is the beginning of a new duty period and "end_prev_duty" specifies the end of the duty which is immediately before some given date and time.

In addition, to general constraints as above there are often other requirements on the problem stemming from particular policies of the specific company and crew preferences. The abductive formulation with its modular representation of the problem, facilitates a direct representation of these either with additional integrity constraints in IC or through the further refinement of the model in P.

### 4.3. The Rescheduling Module

The problem of rescheduling is an important task in air-crew scheduling due to the frequent changes that occur in this environment, which often make the existing schedule inconsistent or of poorer quality. In the ALP framework we can develop a rescheduling system from the initial representation in <P,A,IC>. The sets of abducibles A and integrity constraints IC remain the same with the possibility of new constraints to be added, whereas P is extended to model the rescheduling process. The process of rescheduling, in ALP, is facilitated by the (natural) ability of abduction to reason with an existing set of hypotheses, in this case the old solution. Thus abduction can help to restore the consistency of the old solution with minimal changes on the affected assumptions and others depending on them.

The air-crew scheduling system presented above has been extended to model rescheduling. The rescheduling module was implemented to handle four general types of changes, flight cancellation, new flight addition, the rescheduling (e.g. change on the date or time) of an existing flight, or the selection of an alternative crew for a given flight. This module is interactive in the sense that the user can either suggest a crew for a particular flight, in which case the system will check if this choice is valid and to what extend it will affect the rest of the old schedule, or decide whether to accept a system proposed selection of crew. This rescheduling facility allows the system to be used (by an expert) both as a tool for improving the quality of an existing solution as well as a rescheduling tool that will adapt, on the day of operation, an old schedule to the changes that may occur. More details can be found in the extended version of this paper [12].

## 5. Evaluation of the System

An abduction-based system for air-crew scheduling has been implemented for the problem of Cyprus Airways Airlines in ALP on top of ECLiPSe [6]. Several experiments have been carried out in order to evaluate the system along the following three dimensions :

- The quality of the solution, compared to the solution of an expert.
- The flexibility of the system, and its ability to be specialized to additional or changing needs and preferences of the problem.
- The ability of the system to help the process of rescheduling.

The Cyprus Airways airline company, although small in size, contains the full complexity of the air-crew scheduling problem. The problem can be divided into a number of independent phases each of which is concerned with the scheduling of one class of crew members. For the purposes of these experiments we focus only on one type of crew, namely the captains. Cyprus Airways has 28 captains of several ranks (e.g. captains, training captains, technical pilots and flight managers). The system has been tested for busy, with a heavy flight schedule, and for less busy months of the year. The generated schedules were of good quality on the cost, as compared with an expert's solution (e.g. comparable number of deadheads, stay-overs etc.). Furthermore, our solutions were better in the balancing of the duty and flying hours assigned to each crew member over the month. This also affects (favourably) the total cost of the solution.

Other experiments were carried out with this system to show that equally good solutions can be found with less number of pilots. Below is an indicative table on some of these experiments (on a SUN Ultra 64Mb RAM), carried out for busy and less busy (ordinary) months of the year.

| | **Busy** | | **Ordinary** | | |
|---|---|---|---|---|---|
| | **28 plts** | **26 plts** | **28 plts** | **24 plts** | **21 plts** |
| **Time** | 560 sec | 950 sec | 145 sec | 150 sec | 152 sec |
| **Deadheads** | 23 | 25 | 10 | 10 | 10 |
| **Stay-Overs** | 9 | 8 | 16 | 16 | 16 |
| **Transfers** | 83 | 85 | 39 | 39 | 39 |

The flexibility of the system is demonstrated through the addition of new integrity constraints which express requirements or preferences of the company or crew. These constraints are divided in three categories, namely the **local**, **group** or **global**, according to the number of resources (pilots) they involve.

**Local** integrity constraints refer to individual crew members and have no significant effect on the computational time. **Group** constraints involve a number of pilots which have something in common (e.g. of the same rank) and have a minor effect on the efficiency of the system. The third type of constraints, the **global** constraints, affect all crew members and can affect the computation by increasing the time from 10% up to 86%. In some cases, when these constraints make the problem very tight, the system is unable (within a reasonable time) to find a solution. Experiments were again carried out for busy and ordinary months of the year. The results are displayed in the table below.

| | **Local ICs** | **Group ICs** | **Global ICs** |
|---|---|---|---|
| **Ordinary** | 146 sec | 160 sec | 170 sec |
| **Busy** | 560 sec | 768 sec | 1047 sec |

It is important to note that, due to the modularity of the abductive system (as discussed in section 3.2 previously), such additional requirements are accommodated into the system with no extra (programming) effort. We simply express these in logical form as integrity constraints and add them to the system.

**Rescheduling**

Using the solutions generated for a busy and an ordinary month, a number of rescheduling experiments were carried out for each class of changes, as presented in section 2.2. The rescheduling module of the system is interactive and can be used as an advisory to the operator, offering incrementally choices that would correct the old schedule. In each case, the operator is informed of other changes that would be needed if s/he accepts this choice.

On average for changes of the schedule (apart from cancelations which as expected take very little time) on an ordinary month the system suggests new assignments within 2-3 secs. For busy months the systems needs 3-6 secs on average. Again more details can be found in [12].

## 6. Conclusions and Future Work

In this paper we have presented an air-crew scheduling system based on Abduction. The system was developed and tested for a medium-sized application, that of Cyprus Airways Airlines, which, although average in size, it still contains the full complexity of the crew scheduling problem. Several experiments were carried out, using this system, with promising results.

The paper has explored the advantages of using an abductive approach for solving scheduling problems. It has demonstrated how the high-level modeling environment, offered by abduction, helped to develop a *modular and flexible* scheduling system: *modularity* with a clear separation of the two main issues of validity and quality (optimality) of the solution and *flexibility* under changes of the requirements of the problem. The modularity of the system allows the adoption of different strategies or heuristics which can lead to better quality of the solutions. The flexibility of the approach can facilitate the adaptation (or specialization) of the system to different airline companies.

We have also studied how abduction can help us to tackle the problem of adjusting or correcting existing solutions in order to accommodate changes in the application environment, namely the problem of rescheduling. An interactive rescheduling module has been integrated in the developed system that allows the operator to adjust or perfect a schedule to new (unexpected) requirements or additional specific needs of the company or crew. Thus the utility of the system can be summarized as follows:

- Generate a good first solution to the problem.
- Adjust a generated solution and improve its quality.
- Adapt an existing solution to unexpected changes occurring before or on the day of operation.

In the future we want to investigate how the AI techniques, that we have adopted in this work, and the system developed based on these can be applied for the solution of other air-crew scheduling problems in different airline companies. Future work can also study improvements on the computational efficiency of the system. One such interesting direction would be to integrate in the abductive process a specialized constraint solver, for the effective handling of the lower-level goals which are generated through the abductive reductions.

## Acknowledgments

We would like to thank Cyprus Airways Airlines for their collaboration in the development of this application. Part of this work has been carried out with financial help from the EU, KIT program.

## References

1. L.Bianco, M Bielli, A.Mingozzi, S.Ricciardelli, M.Spadoni. A heuristic procedure for the crew rostering problem. European Journal of Operational Research, 58:272-283, 1992.
2. G.Brewka. Principles of Knowledge Representation. CSLI Publications, 1996.
3. L. Console, L. Portinale and D. Dupré. Using Compiled knowledge to guide and focus abductive diagnosis. Journal of IEEE Transactions on Knowledge and Data Engineering, 8(5):690-706, 1996.
4. L.Console, M.L.Sapino, D.Theseider Dupre. The role of abduction in database view updating. Journal of Intelligent Systems, 1994.
5. M.Denecker and D.De Schreye. SLDNFA: an abductive procedure for abductive logic programs. Journal of Logic Programming, 1997.
6. ECLiPSe User Manual. ECRC Munich, Germany, 1993.
7. F.Focacci, E.Lamma, P.Mello and M.Milano. Constraint Logic Programming for the Crew Rostering Problem. In Proceedings of the Third International Conference on the Practical Applications of Constraint Technology, PACT'97, pp. 151-164, 1997.
8. N.Guerinik, M.Van Caneghem. Solving Crew Scheduling Problems by Constraint Programming. Proc. 1st International Conference on Principles and Practice of Constraint Programming, 1995.
9. M.Kress, B.Golany. Optimizing the assignment of aircrews to aircraft in an airlift operation. European Journal of Operational Research, 77:475-485, 1994.
10. A.C.Kakas, R.A.Kowalski, F.Toni. Abductive Logic Programming. Journal of Logic and Computation 2(6), 1993.
11. A.C.Kakas, A.Michael. Integrating Abductive and Constraint Logic Programming. Proc. 12th International Conference on Logic Programming, MIT Press, pgs. 399-415, 1995.
12. A.C.Kakas, A.Michael. An abductive-Based Scheduler for Air-Crew Assignment. Department of Computer Science, University of Cyprus, Technical Report TR-98-17, 1998.
13. T.Menzies. Applications of Abduction: Knowledge-Level Modeling. International Journal of Human Computer Studies, August, 1996.
14. D.Poole, R.G.Goebel, Aleliunas, Theorist: a logical reasoning system for default and diagnosis. The Knowledge Fronteer: Essays in the Representation of Knowledge, (Cercone and McCalla eds), Springer Verlag Lecture Notes in Computer Science, 1987.
15. C.Pavlopoulou, A.P.Gionis, P.Stamatopoulos, C.Halatsis. Crew Pairing Optimization Based on CLP. Proc. 2nd International Conference on the Practical Application of Constraint Technology, London, UK, 1996.
16. D.Poole. A logical framework for default reasoning. Artificial Intelligence vol.36, 1988.
17. M.P.Shanahan. Event Calculus Planning Revisited. Proceedings of the 1997 European Conference on Planning (ECP 97), 1997.