Crew Scheduling Based on Constraint Programming: The PARACHUTE Experience

Constantin Halatsis Panagiotis Stamatopoulos Isambo Karali
Tilemahos Bitsikas Georgios Fessakis Athanassios Schizas
Stelianos Sfakianakis Constantin Fouskakis
Univeristy of Athens
Themistoklis Koukoumpetsos Dimosthenis Papageorgiou
Olympic Airways

Abstract

Scheduling flying crews of airline companies is a hard combinatorial problem, taken the complexity of constraints that have to be satisfied. Constraint-based programming techniques is a new method, apart to the traditional Operations Research-based techniques used so far for the solution of the problem. The presentation will discuss the findings and results of applying constraint-based programming techniques and associated tools for the solution of the crew scheduling problem in the context of the ESPRIT HPCN project PARACHUTE, a project that combines constraint programming with parallel computing.

1 Introduction

The major difficulty of the combinatorial optimization problems is the size of the problem search space. Finding the optimal solution is a very demanding task which implies the testing of every possible solution as a candidate for being an optimal solution. A simple method to achieve this, is the "generate and test" method. This method invokes the generation of all possible solutions and then tests every solution one-by-one in order to find the least costly one. The "generate and test" method is easy to be implemented and appears to be successful for problems with a small search solution space, but it is inefficient for large scale problems.

Constraint programming is an attempt to overcome the difficulties of traditional programming by enhancing a programming language with constraint solving mechanisms. When constraints are introduced using a constraint programming scheme, a solving mechanism is applied to variables involved in the constraints, in order to reduce the problem search space. Constraint solving has been used in many different application areas such as scheduling, planning and resource allocation.

Software systems are inherently complex when they model real-world domains and their complexity often exceeds the human intellectual capacity. An elegant solution to this problem is to adopt an object-oriented programming approach. By applying object-oriented design, the constructed software is resilient to change and written with economy of expression. In addition, a greater level of confidence is achieved in the correctness of the software. Ultimately, the risks of developing complex software systems are reduced.

Parallelism is a straight way to achieve efficient results. This is done by dividing the computational complexity of a task among several machines or several processors of the same machine which have to solve less resource consuming problems. The need for parallelism becomes even more evident when the encountered problems are combinatorial ones. There are various issues that have to be tackled in order to reach an efficient parallel execution of a program. The way the problem at hand is decomposed into subproblems and the communication means among these subproblems are quite crucial and have to be studied thoroughly. In addition, the modeling of the problem using a specific parallel programming language and the way this language maps onto candidate hardware parallel environments are things that have to be taken into account.

In the context of the ESPRIT III Project PARACHUTE (PARAllel Constraint Handling for User TEchnologies) (EP 9134), the University of Athens (UoA) is developing an application, named CREM. CREM stands for CREw Management and addresses the crew management problem of airline companies. Olympic Airways (OA) is the end–user of the application. This application has been selected as one that is of strategic relevance to the company's operation. More precisely, it refers to the scheduling of the flight personnel of the company for all the various aircraft flights both national and international as well as the assignment of other company tasks, e.g. stand by duties. It involves a combinatorial search optimization problem. This type of application is highly compute intensive and is proven to require high performance systems and advanced software platforms. At the same time this application is of highly complexity given the fact that both cockpit and cabin crews of various speciality and rank have to be scheduled. There are numerous types of constraints that have to taken into account such as temporal, local, working time and other contractual or governmental ones.

A prototype of the system has been implemented and now the full implementation is being built. The software platform that the CREM application is built on is the object-oriented C++ language enhanced with the constraint programming oriented ILOG Solver library. This library supports the efficient solution of combinatorial problems through their formulation as constraint satisfaction problems. Parallel execution is carried out using the parallel version of the ILOG Solver. The system is accessible via a graphical user interface that is being built using the ILOG Views library.

In this paper, we present the approach that we have followed to solve the cockpit crew scheduling problem. We also discuss the exploitation of constraint technologies in the approach. The whole approach has been facilitated by the formulation of the various application constraints in a formal mathematical way.

The paper is organized as follows. Firstly, some terminology is introduced. The general characteristics of the various modules of the CREM system follow discussing also the methodologies used. Finally, some conclusions are included.

2 Terminology

In this section, we introduce some terminology that is used by the application.

Home base City of residence of crews. It can be any one of those designated by the company.

Sector A flight between two airports (the smallest flight unit).

Duty A sequence of crew tasks in which a crew member can operate without rest.

Pairing A sequence of duties, separated by the required rest time, starting and ending at the same home base which may not contain any day off.

Line of Work A sequence of pairings, separated by the required days off, assigned to a specific crew member for the rostering time.

Anonymous Crew Schedule A set of pairings such that all company flights are included.

Assigned Crew Schedule The set of Lines of Work covering all crew members and all crew tasks for the rostering period.

Rostering Time A time period for which an Assigned Crew Schedule is produced.

3 Crew Scheduling by the CREM System

The main function of the CREM system is the crew scheduling. This is carried out by the Crew Scheduling subsystem. More precisely, this subsystem performs the necessary scheduling of the different types of crews in order to ensure that all the flights are carried out in such a way that the airline company gains the maximum profit without violating the working conditions of the crew members. The crew members according to their specialization are divided into two main groups: cockpit crew and cabin crew. The scheduling is performed for each type of crew separately. The Crew Scheduling subsystem is further divided into the Anonymous Crew Scheduling and the Crew Assignment parts. Furthermore, the Anonymous Crew Scheduling is divided into the Duty Construction, the Pairing Construction and the Pairing Selection modules.

3.1 The Duty Construction Module

The duty construction module is responsible for constructing duties. This module operates in the following way:

1. All company sectors for a single aircraft type are grouped together.

2. We construct all possible duties that can be constructed containing i sectors, where i is an integer between 1 and a predefined maximum number of sectors $Smax$.

3. Duties of i sectors are constructed as follows:

   (a) A duty with i sectors is created. The sectors of a duty are domain variables. Each sector's domain contains all the sectors. At this stage, the sectors are not instantiated in a specific day.

   (b) The duty constraints are set on the uninstantiated duty.

They are two types of constraints that are set on the generated duties. These are the regulations which are set by the airline company and the heuristics which are constraints that the crew schedule experts set on the duties in order to avoid unnecessary duties. Some examples of the regulations are the following:

- The number of sectors in a duty are limited.

- The flight time has a maximum value and depends on the aircraft type, on the geographic type of the duty, on the number of sectors and on whether it is accomplished during day or night.

- The duty time has a maximum value and it depends on the aircraft type, on the geographic type of the duty, on the number of sectors and on whether it is accomplished during day or night.

All these regulations have been given a mathematical formulation. For example: "The arrival date and time of a sector must precede the departure date and time of its successor" has been formulated as: $d_{i+1} \geq d_i$ or, if $d_{i+1} = d_i$ then $a_{i+1} \geq b_i$)

Constraint programming is exploited in the construction of duties. Regulations and heuristic constraints are implemented as constraints and are treated by the constraint solver. In the generation of duties there are few choices and few constraints. These constraints, though, limit the duty sequences generation.

3.2 The Pairing Construction Module

The main function of this subsystem is to combine duties into pairings. This combination has to satisfy all work rules and regulations imposed on crew pairings.

The algorithm that is followed in this module is the following: We construct N_i pairings containing i duties in a stochastic way. Each pairing is constructed as follows: Let $Dmax$ be the maximum number of duties in a pairing. For each i where $0 < i \leq Dmax$,

1. A pairing with i duties is created. Each duty's domain contains all the duties. At this stage the duties are not instantiated in a specific day.

2. The pairing constraints are set on the uninstantiated pairing.

3. The requested number of pairings with i duties are generated.

Some regulations that apply to the pairing construction module are the following:

- A pairing must start and end at the home base.

- The number of duties in a pairing is limited.

- The flight time in a pairing has a maximum value.

- The duty time in a pairing has a maximum value.

- The flight time and duty time in 24 consecutive hours is limited.

Some heuristics also apply to the pairing construction module such as "the number of same duties in a pairing has a maximum value". Every pairing is assigned a cost value as the problem is faced as an optimization one. The cost is a measure of how good utilization of the crew a pairing does.

Constraint programming is significantly exploited in the pairing construction module. Here, the choices are more than the ones in the duty construction module and the number of constraints is larger also. A lot of choices are eliminated a priori or fail as soon as possible. Constraints implement both regulations and heuristics.

3.3 The Pairing Selection Module

In this module, the selection of the appropriate pairings to cover the flight sectors with minimal cost is carried out. A pairing is broken into on-duty and off-duty periods, and the major cost considerations involve flight time and total duration of the pairing.

The pairing selection problem has been highly studied in the literature and usually has been treated either as a set partitioning or set covering problem. It has been a favorite subject in the Operations Research community. The pairing selection problem is formulated as a set partitioning problem in the following way: Suppose there are m pairings and n sectors. If

- A is a $n \times m$ matrix whose columns correspond to the pairings, that is

$$a_{ij} = \begin{cases} 1 & \text{if } sector_i \in pairing_j \\ 0 & \text{otherwise} \end{cases}$$

- V is a $m \times 1$ matrix of domain variables, where

$$v_i = \begin{cases} 1 & \text{if } pairing_i \in solution \\ 0 & \text{otherwise} \end{cases}$$

- E is the $n \times 1$ unary column vector

then $A \times V = E$ and $C \times V$ is minimum, where C is a cost matrix assigned to the pairings.

The algorithm that is followed in the pairing selection module is:

1. Create matrix A from the pairings.
2. Create matrix V as an array of domain variables.
3. For each i set $A_i \times V = 1$.
4. Generate solution starting from the variable with the smallest cost per sector and giving to it the value 1.

The equations are handled by the constraint solver of the platform. In addition parallelism is exploited in the production of the best solution.

3.4 The Crew Assignment Module

The crew assignment problem involves the allocation of the planned pairings from the previous phase to individual crew members to form a line of work over the rostering period, usually one month. In the construction of lines of work, days off are assigned to crew members and standby crew is also scheduled.

The full crew assignment process is usually performed separately for the cabin and cockpit crew, since their duty is ruled by different constraints and regulations. Cockpit crew assignment can usually be broken into smaller independent subproblems corresponding to groups of crew members of the same rank, that is for captains, first officers and flight engineers.

The approach followed for the solution of the problem is the fair assignment method. According to this method, the assignment of pairings is approximately the same for all crew members regardless their flight experience.

The construction of lines of work is governed by a set of constraints that each line of work must satisfy. Some of them are:

- A crew member cannot carry out more than 2 duties containing 5 sectors within 7 consecutive days.
- A crew member must not work more than 40 hours within 7 consecutive days.
- A crew member must not fly for more than 28 hours within 7 consecutive days. There are also limits for cumulative flight time for the past 3, 9 and 12 months according to the aircraft type.

An example of the mathematical formulation of a constraint that is applied in the crew assignment module follows.

Verbal description: "In 7 consecutive days, 2 days off must be included". This verbal constraint is denoted as: Let U be the number of calendar days in the rostering period and let DOF_d be equal to 1 if day d is a day off or 0 otherwise. Then,

$$\sum_{d'=d}^{d+6} DOF_{d'} \geq 2, \quad \text{for } d = 1, 2, \ldots, U - 7$$

Except for the well defined set of constraints that each line of work must satisfy and derive from the various contracts and regulations, there are also rules reflecting the crew member's or the company's preferences. Such rules that can be taken into account are:

- Personal preferences about specific flights he/she want to perform, or personal limitations about flights he/she does not want to perform.
- Personal preferences about specific dates he/she wants to have his/her days off.

Some rules derive from the requirement of OA for fair assignment. Fairness is required on various criteria, some more important than the others. Like the phase of the Anonymous Crew Scheduling, the crew assignment phase deals with an optimization problem. Therefore, the existence of a cost function evaluating the cost of each solution is necessary in order to choose the optimal one. The cost function refers to the various fairness criteria taking into account their degree of importance.

In the implementation of the crew assignment, crew members are modeled as resources and pairings, stand by duties and days off are activities that have to be allocated to resources. The other tasks that are preassigned to a crew member, such as training etc., are activities that are considered to have been already assigned. Consequently, the algorithm that is used in this module is:

1. Perform pairing assignment.
   (a) Represent crew members as resources and pairings as activities that require crew members.
   (b) Set constraints on pairings and crew members.
   (c) Generate assignments of pairings and crew members covering all pairings and all crew members.
2. Perform standby pairing assignment.

Constraint programming is heavily exploited in the crew assignment module. Actually, the whole pairing allocation problem is handled by the constraints' engine. The OA constraints are implemented using the facility of implementing user defined constraints that ILOG Solver supports. At each pairing allocation, all other pairings that violate the constraints are pruned and cannot be considered as candidate to be allocated to the crew member where the pairing has been assigned. For finding the optimum solution, parallelism is exploited to speed up the execution.

4 Conclusions

In this paper we presented the approach that we follow to solve the cockpit crew scheduling problem of airline companies. The effort is made in the context of the ESPRIT III Project PARACHUTE. The problem has been divided into four subproblems, namely duty construction, pairing construction, pairing selection and crew assignment, in order to face the complexity that it presents. Constraint programming facilitates the modeling and the formulation of the problem and accelerates the execution as a priori pruning of the search space is achieved. Parallelism is exploited in the optimization requirement of the solution. The system has been implemented using the ILOG Solver platform. The results have been assessed by OA, which is the end–user of the application, and have been found to be quite satisfactory. What is more is that the system carries out automatically functions that so far are made by humans. A final remark is that the application inspired new generic constraints which are examined by ILOG with the continuous feedback of UoA.

References

[AFST69] J. P. Arabeyre, J. Fearnley, F. C. Steiger, and W. Teather. The airline crew scheduling problem: A survey. *Transportation Science*, 3:140–168, 1969.

[AGPT91] R. Anbil, E. Gelman, B. Patty, and R. Tanga. Recent advances in crew-pairing optimization at American Airlines. *Interfaces*, 21(1):62–74, 1991.

[Bal80] E. Balas. Cutting planes from conditional bounds: A new approach to set covering. *Mathematical Programming*, 12:19–36, 1980.

[BBM+92] L. Bianco, M. Bielli, A. Mingozzi, S. Ricciardelli, and M. Spadoni. A heuristic procedure for the crew rostering problem. *European Journal of Operational Research*, 58:272–283, 1992.

[BH80] E. Balas and A. Ho. Set covering algorithms using cutting planes, heuristics, and subgradient optimization: A computational study. *Mathematical Programming*, 12:37–60, 1980.

[BJ92] J. E. Beasley and K. Jørnsten. Enhancing an algorithm for set covering problems. *European Journal of Operational Research*, 58:293–300, 1992.

[BP76] E. Balas and M. W. Padberg. Set partitioning: A survey. *SIAM Review*, 18:710–760, 1976.

[CK75] N. Christofides and S. Korman. A computational survey of methods for the set covering problem. *Management Science*, 21(5):591–599, 1975.

[CM92] I. F. Croall and J. P. Mason, editors. *Industrial Applications of Neural Networks — Project ANNIE Handbook*. Springer-Verlag, 1992.

[Etc77] J. Etcheberry. The set-covering problem: A new implicit enumeration algorithm. *Operations Research*, 25(5):760–772, 1977.

[FK90] M. L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36(6):674–688, 1990.

[Ger89] I. Gershkoff. Optimizing flight crew schedules. *Interfaces*, 19(4):29–43, 1989.

[GN69] R. S. Garfinkel and G. L. Nemhauser. The set-partitioning problem: Set covering with equality constraints. *Operations Research*, 17:848–856, 1969.

[HP93] K. L. Hoffman and M. Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6):657–682, 1993.

[ILO95a] *ILOG Solver: Reference Manual — Version 3.0*, 1995.

[ILO95b] *ILOG Solver: User Manual — Version 3.0*, 1995.

[KG94] M. Kress and B. Golany. Optimizing the assignment of aircrews to aircraft in an airlift operation. *European Journal of Operational Research*, 77:475–485, 1994.

[LMO88] S. Lavoie, M. Minoux, and E. Odier. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35:45–58, 1988.

[LSS71] C. E. Lemke, H. M. Salkin, and K. Spielberg. Set covering by single-branch enumeration with linear-programming subproblems. *Operations Research*, 19:998–1022, 1971.

[Mar74] R. E. Marsten. An algorithm for large set partitioning problems. *Management Science*, 20(5):774–787, 1974.

[MMK79] R. E. Marsten, M. R. Muller, and C. L. Killion. Crew planning at Flying Tiger: A successful application of integer programming. *Management Science*, 25(12):1175–1183, 1979.

[MS81] R. E. Marsten and F. Shepardson. Exact solution of crew scheduling problems using the set partitioning model: Recent successful applications. *Networks*, 11:165–177, 1981.

[Pie68] J. F. Pierce. Application of combinatorial programming to a class of all-zero-one integer programming problems. *Management Science*, 15(3):191–209, 1968.

[PL73] J. F. Pierce and J. S. Lasky. Improved combinatorial programming algorithms for a class of all-zero-one integer programming problems. *Management Science*, 19:528–543, 1973.

[Ric89] H. Richter. Thirty years of airline operations research. *Interfaces*, 19(4):3–9, 1989.

[Rub73] J. Rubin. A technique for the solution of massive set covering problems, with application to airline crew scheduling. *Transportation Science*, 7:34–48, 1973.

[Rya92] D. M. Ryan. The solution of massive generalized set partitioning problems in aircrew rostering. *Journal of the Operational Research Society*, 43(5):459–467, 1992.

[Spi61] M. Spitzer. Solution to the crew scheduling problem. In *Proceedings of the First AGIFORS Symposium*, 1961.