Solving Larger Crew Pairing Problems

Curt Hjorring

Carmen Systems AB
Odinsgatan 9
Gothenburg, Sweden
curt.hjorring@carmensystems.com

Abstract

Column generation has been applied successfully to solve airline crew pairing problems. However, there is a desire to solve larger problems in order to give further improvements in the airline planning process, and to solve problems originating from the railway market. We present algorithmic techniques that are useful to solve such problems. For instance, Lagrangian relaxation, along with a subgradient solver, gives much better dual feedback than an LP based approach, and has much lower computational overhead. With these techniques we are able to find high quality solutions to daily railway pairing problems with over 7,000 legs.

1 Introduction

The airline crew pairing problem has been studied intensively for many years by OR practitioners. This is because crew costs are very large (annual figures range from hundreds to thousands of millions of dollars), and because it is a complex task to find high quality solutions. Delayed column generation has shown to be a key approach to find these solutions within a reasonable amount of time, and is now used by most major airlines.

The crew pairing problem is part of the airline planning process, which in practice decomposes an extremely complex problem into a number of interrelated problems. The problems are solved sequentially, in order to reduce computational complexity. Firstly fleet types are assigned, which then induces a number of separate pairing problems, one for each crew compatible aircraft family. Unfortunately this decomposition introduces extra costs into the final solution since crew costs are only approximated in the fleet assignment problem. In order to build a more globally accurate model, researchers are investigating an integration of the fleet assignment and crew pairing problems. One such model would resemble a very large crew pairing model, along with some additional constraints that model various fleet aspects.

The railway market also requires very large pairing problems to be solved. Frequently the problems are larger than those of major airlines for two main reasons: firstly the railway pairing problems do not naturally decompose by train type, and secondly each train stops at many stations during one day, with most of these stops allowing a crew change to occur.

In this paper we address some algorithmic techniques that are useful in the solution of these very large pairing problems. Earlier work is reviewed in Section 1.1, and pertinent differences between railway and airline problems are discussed in Section 2. We then group the techniques into the three main components of a column generation scheme, the pricing subproblem (Section 3), the master optimizer (Section 4), and the integrality strategy (Section 5). Finally future directions are outlined in Section 6.

1.1 Earlier approaches

The crew pairing problem can be formulated as an integer problem, in which columns correspond to pairings, and rows to flights. Unfortunately the number of columns is immense. Snowdon et al. (2000) report that a daily problem for a large American carrier has roughly 10^{14} columns. Therefore delayed column generation is frequently used. An early application of this is due to Minoux (1984), in which it is shown that columns can be priced by solving a shortest path problem on a network with arcs representing flights and overnight connections. In the airline business the sequence of flights flown in a typical working day is known as a *duty period*, and for many rules the legality of a single duty period is independent of preceding or succeeding duties. Lavoie et al. (1988) take advantage of this structure and form a duty period network where nodes are duty periods with state information, and arcs represent legal overnights.

Vance et al. (1997) present results for flight and duty based implementations. In both versions, resources are added to each node to track legality conditions, and a *resource constrained shortest path problem* is solved. Comparisons between the two versions are difficult to make since they implement different variants of the rules, but in general the flight based version spends a larger portion of time in the pricing routine, than does the duty based version. However, the duty version cannot solve as large problems as the flight version due to memory limitations. Storage of the duty periods and the duty period connections is prohibitive for large problems (there is one arc for each legal duty period connection).

The excessive memory usage of the duty network is addressed in Hjorring and Hansen (1999) by creating an initial relaxed network where duty-duty connections are replaced by flight-flight connections. Portions of the network are dynamically refined when required, in order to capture additional cost and legality information. The method also uses a k-shortest path approach, instead of a resource constrained shortest path. This allows rules and costs to be implemented in a separate module that presents a black box interface to the pricing subproblem. The rules module can then be implemented using a modeling language that allows the airline to easily write their own rules and cost function.

2 Differences between airline and railway problems

As mentioned above the number of flights, or more generically *legs*, in a railway pairing problem is typically much larger than for an airline problem, due to the absence of fleet decomposition.

Another significant difference is the average duration of the legs. For Deutsche Bahn's (German Railways) long distance network, the average duration is around 30 minutes, while in a typical shorthaul airline network the average duration would be between 120 and 150 minutes. Thus a train crew can operate many more legs per day than a pilot, and enumeration of all duties becomes too time consuming. The railway pricing subproblem must therefore cope with all the meal and work time regulations that apply to individual duties, as well as for the whole pairing, making the pricing problem much more complicated. Fortunately a railway pairing is at most two days long, whereas a shorthaul airline pairing will have an upper limit of four or five days.

For typical airline problems the average number of nonzeros per variable is between nine and ten, while for a railway the average is between 13 and 15. This indicates that the master optimization problem for railways will be computationally tougher than for airlines.

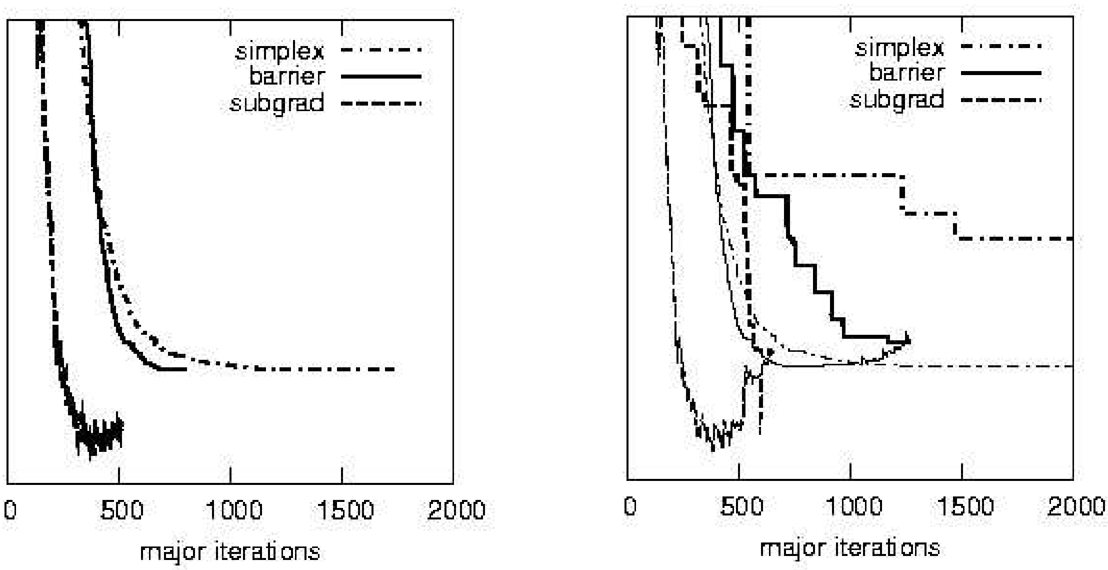
The next three sections describe techniques to solve very large crew pairing problems, and usually apply to both airline and railway problems.

3 The pricing subproblem

The results in this paper are based on an extended version of the pricing subproblem that is described in Hjorring and Hansen (1999) and in Galia and Hjorring (2003). In order to model the credit cost function common to North American airline pairing problems, the multiobjective shortest path algorithm of Azevedo and Martins (1991) was incorporated into the pricing subproblem. Since we are not interested in complete paths in which any of the objective components has positive reduced cost, we introduced a *bounding check* when finding nondominated alternatives. The check tests if a path exists from the current node to the sink, such that path's cost for each component is negative. In order to make this test fast, each component is considered independently. Then the test can be implemented by performing a preprocessing stage that calculates a "backward shortest path tree" for each cost component from the sink to all nodes. The check at each node simply adds the costs of the current node, to the costs from the backward shortest path tree. If the sum is positive, then no useful complete path exists for the current node, and it can be discarded. A similar technique is used in Dumitrescu and Boland (2003).

Since we cannot enumerate all duty periods for railway problems in reasonable time, we implemented basic work and duty time rules using resource dependent cost elements, as described in Galia and Hjorring (2003). For Deutsche Bahn this requires three resources for each duty, plus one for each pairing. Other rules and penalties are handled by the k-shortest path code.

Railway problems have many crew bases, 20 to 30 being common. This stresses the pricing subproblem. In order to improve performance we take advantage of additional geographical structure. The railway timetable is built with a number of train lines, and the bases are distributed along these lines. Each base connects with only a subset of the lines, and it is generally preferred to operate legs with a crew base that directly connects to the leg's line. Thus we can perform heuristic pricing steps with a reduced network. If the heuristic fails to find a negative reduced cost pairing, we perform an exact pricing step with the complete network.



(a) lower bound till convergence

(b) LB and UB with connection fixing

Figure 1: The effect of different master optimizers

4 The master optimizer

The task of the master optimizer is to determine dual variable values based on the columns that have been generated so far. These values are then used as input to the pricing subproblem, which generates improving columns. One approach to implementing the master optimizer is to relax the integrality conditions from the integer program, and solve the resulting LP, either using a simplex solver, or by using an interior point solver without crossover. A simplex solver returns values corresponding to extreme points of the optimal face, whereas interior point solutions are in the relative interior of the optimal face, which is usually a better representative of the possible dual solutions (Lübbecke and Desrosiers, 2002).

Another approach is to Lagrangian relax all the constraints and solve the relaxation using a subgradient approach. The details of this, along with a dual ascent approach to find integer solutions, are presented in Gustafsson (1999).

Figure 1 shows a comparison of these three approaches on a 530 leg, North American flight crew problem. Figure 1a tracks the progress of the lower bound with all integrality restrictions removed. As expected, the LP solvers converge to the same lower bound, while the subgradient solver provides a significantly weaker lower bound (in this case around 0.5% lower). However, the simplex approach has a very long tail, requiring 1,800 iterations to reach convergence, whilst the interior point approach requires 800 iterations, and the subgradient only 500.

In Figure 1b the process is continued by adding a simple integrality strategy. When no more negative reduced cost columns exist, a number of leg-leg connections are fixed to one, and the column generation process continues. The fix-and-regenerate steps continue until the lower

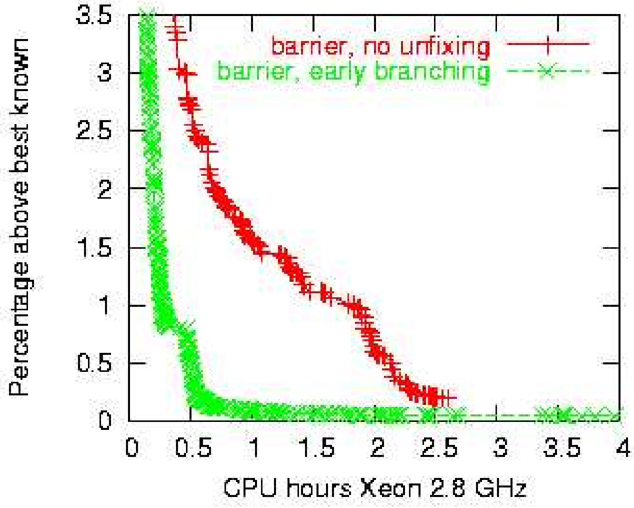


Figure 2: Different integrality strategies

bound to the integer restricted problem is fathomed by the best known solution. The upper bound converges fastest for the subgradient approach, and slowest for the simplex case. In fact the graph truncates the simplex run, convergence is reached after 5,000 major iterations, and then to a poorer quality solution.

Whilst Figure 1 applies to one particular problem, the results are similar for other problems. As the size of the pairing problem grows, the difference between the approaches becomes even more significant. A further advantage of the subgradient approach is that the computations are much quicker. For the 530 leg problem, a subgradient major iteration is on average twice as fast as an interior point iteration, and four times faster than a simplex major iteration.

5 Integrality strategy

The previous section presented a simple integrality strategy. We have improved upon that by implementing *early branching* and by unlocking some or all connections when the lower and upper bounds meet, and then continuing. In early branching we no longer wait for convergence to the LP, but instead fix connections after a set number of major iterations have passed. This has the effect of returning high quality integer solutions early in the search history, and can even reduce overall run time. If desired, a valid lower bound can be determined after the early branching by unlocking all connections, and generating columns until convergence is reached. Figure 2 shows the effectiveness of this approach. In both cases an interior point solver was used as the master optimizer. Each line shows the primal objective averaged over eight runs, each with a different starting solution produced by a very simple construction heuristic.

The connection fixing strategies we have tried take a fractional solution to an integrality relaxed problem, determine fractional leg-leg connections, and fix the connections that are closest to one. When solving very large pairing problems ($> 2,000$ constraints), the time taken to calculate the fractional solution can be very large if standard LP solvers are used. Instead we have implemented an approximate solver that is similar to the volume algorithm of Snowdon et al. (2000).

6 Applications and future work

The above techniques have been applied to Deutsche Bahn's long distance and regional networks, for both train drivers and conductors. The results are promising and are in the process of being taken into production. However solution times to high quality solutions for large dailies ($> 7,000$ legs) can be long, over 20 hours on a 2.8 GHz Xeon. This is especially true when there are many complicating global constraints. We are investigating ways to better tune the subgradient solver to handle such cases.

For the integrated airline cases we have made some initial runs where we have merged the large fleets of a U.S. major carrier. These require more run time than the sum of the decomposed problems, but less than twice that sum. We hope to be able to report soon on runs where fleet and crew aspects are considered simultaneously and accurately.

References

J. Azevedo and E. Martins, "An algorithm for the multiobjective shortest path problem on acyclic networks", *Investigacao Operacional*, 11(1):52–69, 1991.

I. Dumitrescu and N. Boland, "Improved preprocessing, labeling and scaling algorithms for the weight constrained shortest path problem", *Networks*, 42(3):135–153, Oct. 2003.

R. Galia and C. Hjorring, "Modelling of complex costs and rules in a crew pairing column generator", presented at Operations Research 2003, Heidelberg, Germany, 2003.

T. Gustafsson, A Heuristic Approach to Column Generation for Airline Crew Scheduling, Ph.D. thesis, Chalmers University of Technology, Gothenburg, Sweden, 1999.

C. Hjorring and J. Hansen, "Column generation with a rule modelling language for airline crew pairing", presented at 34th Annual Conference of the Operational Research Society of New Zealand, pages 133–142, 1999.

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 Operations Research*, 35:45–58, 1988.

M. E. Lübbecke and J. Desrosiers, "Selected topics in column generation," Technical report, GERAD, Dec. 2002.

M. Minoux, "Column generation techniques in combinatorial optimization: A new application to crew pairing problems", presented at XXIVth AGIFORS Symposium, 1984.

J. Snowdon, R. Anbil, and G. Pangborn, "The airline crew scheduling problem: Dual simplex, volume, and volume/sprint solutions", Technical report, IBM, T. J. Watson Research Center, Jan. 2000.

P. H. Vance, C. Barnhart, E. L. Johnson, and G. L. Nemhauser, "Airline crew scheduling: A new formulation and decomposition algorithm", *Operations Research*, 45(2):188–200, 1997.