Column Generation and the Airline Crew Pairing Problem

Ranga Anbil, John J. Forrest and William R. Pulleyblank

Abstract. The cost of flight crews is the second largest operating cost of an airline. Minimizing it is a fundamental problem in airline planning and operations, and one which has leant itself to mathematical optimization. We discuss several recent advances in the methods used to solve these problems. After describing the general approach taken, we discuss a new method which can be used to obtain approximate solutions to linear programs, dramatically improving the solution time of these problems. This is the so-called volume algorithm. We also describe several other ideas used to make it routinely possible to get very good solutions to these large mixed integer programs.

1991 Mathematics Subject Classification: 90B35, 90C09, 90C10
Keywords and Phrases: linear programming, integer programming, volume algorithm, crew pairing, column generation

1 Airline Operations and the Crew Pairing Problem

The Airline Crew Pairing problem has become well known as a prototypical applied problem which lends itself to a column generation approach. A *pairing*, or *tour of duty*, is a sequence of flights to be flown by a crew, starting and ending at the crew member's home base. Typically the length of a pairing will range from a single day to four or five days. The pairing problem is to compute a set of pairings covering all the scheduled flights of an airline, which has minimum, or near minimum, cost. This problem normally forms the third major stage in a four part planning process:

1. Schedule Creation. Flights are planned, based on market demands and competitive analysis. For example, we may schedule a flight from New York's La Guardia Airport to Chicago's O'Hare Airport every non-holiday weekday departing at 6:15 P.M.
2. Fleet Scheduling. Airlines normally have several different types (fleets) of aircraft. A fleet is chosen for each flight scheduled in the first stage, subject to the constraints that the type of aircraft chosen for each flight be of suitable capacity and flying characteristics and subject to "Kirchoff's law"

for aircraft – each aircraft that departs from an airport must have landed there first. Subject to these constraints, the operating cost of the aircraft should be minimized.

3. CREW PAIRING. A set of pairings is created which assigns crews to aircraft flights. This will normally be solved separately for each fleet, as typically a crew member only has current authorization for one type of fleet.

4. ROSTERING. Individual crew members are assigned to each pairing, often based on seniority, sometimes by an auction process.

The difficulty of the crew pairing problem comes from several factors:

1. The rules which govern the feasibility of a pairing are very complex, often combining FAA regulations with sets of rules coming from union negotiations.

   For example, the rules governing a feasible assignment of duties of a major US carrier, include the following rules: the maximum trip length is 4 duty days; the maximum duty flying time is 8 hours; the minimum time allowed between connecting flights is 40 minutes; at most one aircraft changes is permitted in a duty.

   There are also more complicated rules. One example requires that in any 24 hour window of a pairing, a crew member flying up to 8 hours is entitled to an intervening rest period of 540 minutes before the next duty. However, if necessary, this can be shortened to 510 minutes, but if this is done, then the following intervening rest must be at least 630 minutes.

2. The calculation of the cost of a pairing is complex, and usually incorporates several nonlinear components. Most of the cost is for the crew pay. This is based on the amount of actual flight time, but also has minima that are applied based on the time of duty each day and the number of days in the pairing. There are also costs associated with hotel costs and per diem expenses.

3. The number of feasible pairings is very large, so it is not practical to generate the complete set of feasible pairings for a problem.

   The table below illustrates the number of feasible pairings, with maximum length three or four days, for several fleets.

Fleet	Max Days	#Flights	#Bases	#Duties	#Pairings (millions)
AAS80	3	1152	12	690,000	48,400
AA757	3	251	15	7,000	1
AA727	3	375	11	31,000	36
AAF10	4	307	3	55,000	63,200
UA737	4	773	7	568,000	100,000,000
USDC9	4	478	4	562,000	105,000,000

Note that although the number of feasible pairings is very large, the number of feasible one day duties is of a much more manageable size.

2 Overall Approach and Formulation

2.1 Basic formulation

The crew pairing optimization problem is normally solved in several stages. First, a daily problem is solved, which handles all the flights which are scheduled every day. Then a weekly problem is solved which reuses as much of the daily solution as possible, and also handles weekends. Then a monthly or dated problem is solved which keeps as much as possible of the weekly solutions and creates a complete monthly schedule, dealing with holidays and week to week transitions. Each stage of this problem can be formulated as an integer program.

Let P be the set of all feasible pairings. For each $j \in P$, we define a $0,1$ variable x_j with the interpretation $x_j = 1$ if the pairing j is used and $x_j = 0$ if it is not used. Let c_j denote the cost of pairing j, which can be computed based on the cost structure as described above.

Let F be the set of all flights that must be covered in the period of time under consideration. For each $i \in F$, let P^i denote the set of all pairings which cover the flight i. Then the problem becomes:

$$
\begin{array}{llr}
\text{Minimize} \quad & \sum_{j \in P} c_j x_j & (1)\\
\text{subject to} \quad & x_j \in \{0,1\} \qquad \text{for all } j \in P, & (2)\\
& \sum_{j \in P^i} x_j = 1 \qquad \text{for all } i \in F. & (3)
\end{array}
$$

Let $A = (a_{ij} : i \in F, j \in P)$ denote the $0,1$ flight-pairing incidence matrix, where $a_{ij} = 1$ if pairing j includes flight i and 0 otherwise. Then this can be written as the set partitioning problem:

$$\text{Minimize } cx \text{ subject to } 0 \leq x \leq 1, \text{ integer, and } Ax = 1. \tag{4}$$

A is a matrix which typically has several hundred rows (one for each flight to be covered) and many millions of columns, corresponding to the set of feasible pairings. The size of A makes exhaustive column enumeration impractical.

Early works, discussed in Arabeyre et al.[3], Bornemann[11], and Marsten and Shepardson[23], restrict column generation *a priori* to a small, manageable subset of preferred pairings and solve the resulting set partitioning problem by implicit enumeration or branch-and-bound via linear programming or Lagrangian relaxation. Other works, reported in Rubin[26], Baker et al.[4], Ball and Roberts[5], Etschmaier and Mathaisel[15], Gershkoff[18], Anbil et al.[1] and Graves et al.[19] employ iterative column generation schemes centered around local improvement procedures but do not guarantee global convergence.

Anbil, Johnson and Tanga[2], present a method to handle more than 5 million columns and report excellent cost savings over a local improvement scheme, but the search is nonetheless limited to this subset. Recently, Housos and Elmroth

[21] solve the problem a day at a time over a week horizon and report excellent results, however, it is not clear if their iterative scheme has global convergence.

Minoux[24] is perhaps the first to present a globally converging column generation method for the linear relaxation of the crew pairing problem. This is described in the next solution. Crainic and Rousseau[13] also report a similar but less formal procedure.

Desaulniers et al.[14] and Barnhart et al.[9] discuss convergence rate issues and column generation integrated into branch-and-bound for solving the overall mixed integer program. Barnhart, Hatay and Johnson[8] discuss a scheme for interactively expanding the duty network to include deadhead flights as they are needed. Ryan and Falkner[28], Hoffman and Padberg[20], Bixby et al.[10], Chu et al.[12] and Wedelin[30] discuss specialized optimization algorithms for the underlying mixed integer programs. Vance, Barnhart, Johnson and Nemhauser[29] present a different formulation for the crew pairing problem but with similar algorithmic issues.

Some additional constraints may also be added to this formulation. For example, *Crew base constraints* associate each pairing with a crew base from which it can be flown. Then the number of pairings associated with each crew base is given both upper and lower bounds. *Quality constraints* may apply to an entire problem, and restrict the number of three and four day pairings which can be used in an optimal solution. Often these are soft constraints, in that they can be violated, but an artificial penalty cost is added to the solution if this happens.

2.2 Solution Approach

Our solution approach combines rapid solutions of linear programming relaxations with column generation and specialized branching:

1. Generate an Initial Solution. If a warm start solution is available, use it. (A warm start solution is a solution obtained from an earlier, similar run.) If not, use heuristics to create a feasible solution. Add additional columns to the problem, as described in the following section.

2. Solve the linear program using the Sprint method, described below.

3. Generate additional columns to improve the quality of the linear program solution. We focus on flights which are being covered by expensive pairings, but also randomize. Also, we use the column generation method described in the next section.

4. Repeat Steps 2 and 3 until the linear program solution only improves negligibly.

5. Collapse flights based on best follow-on using the linear program solution. This is described in Section 4.

6. Reduce the matrix, generate additional columns to improve the linear programming solution using the reduced matrix, and return to Step 5.

7. When the number of flights in Step 5 is less than 100, solve for the best integer programming solution.

The selection of flights to be used for deadheading creates an additional problem. Airlines generally prefer to use their own flights, but will permit the use of competitor flights, if it substantially improves the objective function. This then has the potential to substantially increase the problem size, even to an extent that it cannot be solved. For that reason, careful selection of a suitable set of deadheads is crucial.

2.3 The Sprint method

The linear programs solved in the above procedure have a small number of rows but a very large number of columns. As the problems get large, the solution times using standard linear programming methods grow too rapidly to be practical. John Forrest[16] introduced the so-called *Sprint* method. The example below, based on a problem from American Airlines, had 837 flights and generated a total of 5.5 million feasible pairings. The problem was solved with IBM's OSL Primal, Dual and Sprint, for successively larger subsets of the 5.5 million columns. In each case, the pairings represented by the first 110 columns gave a good feasible solution. All times are minutes of solution time on a S/390 (in 1989).

Number of Pairings	Primal Iterations	Time	Dual Iterations	Time	Sprint Iterations	Time
5,000	479	.05	203	.05	933	.10
10,000	1416	.35	295	.10	1072	.14
50,000	7239	11.1	1861	3.99	2870	.86
100,000	17273	57.7	4804	21.5	6981	2.68
200,000	33950	226.8	8008	74.3	36356	12.7
250,000	44215	397.8	8727	100.7	57181	21.5
500,000			11105	258.6	106002	54.5

The Sprint method proceeds as follows: It begins by solving the linear program obtained by keeping a small subset of the columns and then uses the optimal dual variables to price out all columns of the original problem. If the solution is not optimal to the original problem, then there are columns having negative reduced cost. A new problem is formed by keeping the columns in the optimal basis and then adding a subset of the best unused columns, based on the current reduced cost. This is done by first bucketing the columns based on the reduced cost, adding the columns in the best buckets, plus a random selection of columns from other buckets. Note that even though the number of Sprint iterations grows rapidly, they are performed very quickly, enabling our low solution times.

3 Optimal Column Generation

In the solution approach described above, we go through a number of waves of column generation. Minoux[24] introduced an approach, based on the optimal dual variables to the linear program described above, which would generate new columns that could reduce the cost of the current solution, if any existed.

The linear programming relaxation of the integer program above is

$$\text{Minimize } cx \text{ subject to } 0 \leq x, \text{ and } Ax = 1. \tag{5}$$

We have dropped the upper bounds on the variables, as they are implied by the non negativity constraints, combined with the equations $Ax = 1$. The dual problem will have a variable y_i corresponding to each flight $i \in F$.

Minoux observed that the total number of feasible one-day duties is manageable. (See the table above.) He suggested building a tree of duties to represent pairings as paths in this tree. A path includes both duty arcs and ground arcs that link consecutive duties, and the cost of a path or pairing is viewed as a linear sum of duty and ground arc costs. Initially, some paths are selected and a linear programming solver gives the dual variables corresponding to each flight. The duty arc costs are then reduced by their flight duals and a shortest path algorithm applied to the duty tree gives the most negative reduced cost pairing to enter the problem. It is then resolved and the updated duals drive the next shortest path generation. This process will eventually reach a point when the shortest path yields a non-negative reduced cost. The columns present in the current linear program contain the optimum solution.

Current column generation methods reported in Barnhart et al.[7] Desaulniers et al.[14] Lavoie et al.[22], Wise[31] and Barnhart et al.[9] are mostly further extensions to Minoux's seminal work that address one or more of three main limitations: the path cost is not necessarily linear in the arc costs; all paths may not be legal; there are extra constraints such as crew base constraints that may limit the number of pairings that may be flown from a specific base.

The most common approach is dynamic programming on a shortest-path formulation with side constraints to ensure accurate costing, legal paths and the proper accounting of the extra constraints.

We use a variant of the shortest path column generation scheme. Like Minoux[24], we use a duty tree with duty arcs and ground arcs, but carry a vector of arc costs to enable the different ways of computing the cost. Using the set of dual variables from a previously solved linear program, we reduce the arc costs component-wise. We then perform a depth first traversal of the duty tree, ensuring we are feasible as we proceed, and keeping a component-wise tally of the reduced arc costs. We now add this tally to the cost of the shortest path from our location to the end of the tree, again component-wise, and use the maximum of the component sums as a lower bound on the true shortest path cost. If this lower bound exceeds a threshold, then we backtrack. We ensure that every pairing we generate will be feasible as we proceed, by ensuring that all constraints are satisfied as we go along. This is in contrast with some earlier methods which generated a superset of the feasible columns and then were filtered afterwards. Our method also avoids the excessive memory requirements inherent in dynamic programming.

Each time we generate columns, we set a threshold and generate all columns whose current costs will be less than this value. This value is changed during the running of the method.

4 Branching Methods

Normally, a branch and bound approach is used to repeatedly add constraints to the linear programming relaxation until an integral optimum is found. A key to the success of this approach is the method used to generate these constraints. We use a method called *branch on follow-ons*, introduced by Ryan and Falkner [27]. A *follow-on* is the second of a pair of consecutive flights flown in a pairing. We examine the solution obtained to a linear relaxation, and look for follow-ons which occur in a set of pairings j for which $\sum x_j$ is large, that is, close to 1. (Note that the values x_j will, in general, be fractional.) The sum is over all pairings j in the optimum linear programming solution for which x_j is positive. When we find such a follow on, we fix it, that is, we force it to be part of the solution by creating an artificial flight which combines the two consecutive flights, and reducing the problem.

Note that the column generation phase will still consider the original set of flights, so it is possible to recover from a bad choice.

5 The Volume Algorithm

Barahona and Anbil[6] developed an extension to the subgradient algorithm which will produce approximate optimal feasible primal and dual solutions to a linear program, much more quickly than solving it exactly. These methods yield significant improvements to the running time of our crew pairing approach.

Since the early 1970's, subgradient algorithms have been used to rapidly produce good lower bounds for linear programs, see for example Nemhauser and Wolsey[25]. The method produces a sequence of feasible solutions to the dual problem, which will converge to the optimum. Barahona and Anbil[6] present a new method which also produces a feasible solution to the primal problem. It is based on the following result:

Theorem 1 *Consider the linear program Maximize z, subject to $z + a_i\pi \leq b_i$, for $i = 1, ..., m$, where π is a vector with $n-1$ components. Let $(\hat{z}, \hat{\pi})$ be an optimal solution and suppose that the constraints $1, ..., m', m' \leq m$ are active. Let $\bar{z} < \hat{z}$ and assume that $z + a_i\pi \leq b_i$, for $i = 1, ..., m'; z \geq \bar{z}$ defines a bounded polyhedron.*

For $1 \leq i \leq m'$, let γ_i be the volume between the face defined by $z + a_i\pi \leq b_i$ and the hyperplane defined by $z = \bar{z}$. Then an optimal dual solution is given by

$$\lambda_i = \frac{\gamma_i}{\sum_{j=1}^{m'} \gamma_j}.$$

They also show how this theorem can be used to produce a feasible dual solution which approximates the optimum.

We apply this method to the linear programs we encounter when solving the crew pairing problem. In addition to the performance improvements, the dual solutions produced tend to be very good for the column generation phase. This is because they are highly nonbasic, that is, a large number of variables are nonzero. Dual solutions obtained using the barrier or interior method for linear programming share this property. However, those used by the simplex algorithm tend to be very sparse, and do not work as well.

We also use the volume algorithm as a crash procedure for the simplex method when the Sprint method performs poorly. These situations arise after we generate columns for a period of time using only duals from the Volume algorithm. The table below summarizes runs on several problems obtained from US Airways and Southwest Airlines. All times are CPU seconds on an RS6000/590.

Using the Volume algorithm dual solution to crash the dual simplex is, on average, nine times faster than the dual simplex algorithm and two times faster than the interior method. We did not consider the primal simplex algorithm, since it performs very poorly for these problems.

No. of Rows	No. of Columns	No. of Elements	Primal-Dual Interior	Dual Simplex	Volume plus Dual Simplex
2504	53226	553148	1341	3747	320
2991	46450	502338	1984	6928	923
4810	95933	1009283	4165	25917	2576

6 CONCLUSIONS

The methods we have described here are used successfully by several airlines to solve their monthly crew planning problems. In fact, it is part of a larger system which combines management of a database of previously created good solutions, used for warm starts, with interactive tools permitting the scheduler to manually change the solutions produced.

Presently attention is switching to the *schedule repair problem*. This is the operational problem which occurs when a schedule is disrupted, for example, due to weather or mechanical problems, and it is desired to get back on schedule as efficiently as possible. This problem is complicated by the fact that we must produce feasible solutions in minutes, rather than hours, and by the problems of aircraft availability as well as the possibility of canceling flights.

REFERENCES

[1] R. Anbil, E. Gelman, B. Patty and R. Tanga, Recent Advances in Crew Pairing Optimization at American Airlines, *Interfaces* 21 (1991), 62-74.

[2] R. Anbil, E.L. Johnson and R. Tanga, A Global Approach to Crew Pairing Optimization. *IBM Systems Journal* 31 (1991), 71-78.

[3] J.P. Arabeyre, J. Fearnley and W. Teather, The Airline Crew Scheduling Problem: A Survey, *Transportation Sci.* 3 (1969), 140-163.

[4] E.K. Baker, L.D. Bodin and M. Fisher, The Development of a Heuristic Set Covering Based System for Aircrew Scheduling, *Transportation Policy Decision Making* 3 (1985), 95-110.

[5] M. Ball and A. Roberts, A Graph Partitioning Approach to Airline Crew Scheduling. *Transportation Sci.* 19 (1985), 95-110.

[6] F. Barahona and R. Anbil, The Volume Algorithm: producing primal solutions with a subgradient method, Research Report RC 21103 (94395), IBM T.J. Watson Research Center, Yorktown Heights, NY, October 1997.

[7] C. Barnhart, E.L. Johnson, R. Anbil and L. Hatay, A Column Generation Technique for the Long-Haul Crew Assignment Problem, Computational Optimization Center Working Paper COC-91-01, Georgia Tech. (1991).

[8] C. Barnhart, L.Hatay and E.L.Johnson, Deadhead Selection for the Long-Haul Crew Pairing Problem, *Operations Research* 43 (1995), 491-499.

[9] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P.Savelsbergh and P.H.Vance, Branch-and-Price: Column Generation for Solving Huge Integer Programs, *Operations Research* 46, No.3 (1998), to appear.

[10] R. E. Bixby, J.W. Gregory, I.J. Lustig, R.E. Marsten and D.F. Shanno, Very Large-Scale Linear Programming: A Case Study in Combining Interior Point and Simplex Methods, *Operations Research* 40 (1992), 885-897.

[11] D.R. Bornemann, The Evolution of Airline Crew Pairing Optimization, *22nd AGIFORS Symposium Proceedings*, (1982).

[12] H.D. Chu, E. Gelman and E.L. Johnson. Solving Large Scale Crew Scheduling Problems, *European Journal of Operations Research* 97 (1997) 260-268.

[13] T. Crainic and J. Rousseau, The Column Generation Principle and the Airline Crew Scheduling Problem, *INFOR* 25 (1987) 136-151.

[14] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M.M. Solomon and F. Soumis, Crew Pairing at Air France. *European Journal of Operations Research* 97 (1997), 245-259.

[15] M.M. Etschmaier and D.F.X. Mathaisel, Airline Scheduling: An Overview, *Transportation Sci.* 19 (1985), 127-138.

[16] J.J. Forrest, Mathematical programming with a library of optimization subroutines, presented at the ORSA/TIMS Joint National Meeting, New York, October 1989.

[17] R. Gerbracht, A New Algorithm for Very Large Crew Pairing Problems. *18th AGIFORS Symposium Proceedings* (1978).

[18] I. Gerschkoff, Optimizing Flight Crew Schedules. *Interfaces* 19 (1989), 29-43.

[19] G.W. Graves, R.D. McBride, I. Gershkoff, D.A. Anderson and D.Mahidhara, Flight Crew Scheduling, *Mgt. Sci.* 39 (1993), 736-745.

[20] L.H. Hoffman and M. Padberg, Solving Airline Crew Scheduling Problems by Banch-and-Cut, *Management Science* 39 (1993), 657-682.

[21] E. Housos and T. Elmroth, Automatic Optimization of Subproblems in Scheduling Airline Crews, *Interfaces* 27 (1997), 68-77.

[22] S. Lavoie, M. Minoux and E.Odier, A New Approach for Crew Pairing Problems by Column Generation with Application to Air Transportation. *European Journal of Operations Research* 35 (1988), 45-58.

[23] R.E. Marsten and F. Shepardson, Exact Solution of Crew Problems using the Set Partitioning Mode: Recent Successful Applications, *Networks* 11 (1981), 65-177.

[24] M. Minoux, Column Generation Techniques in Combinatorial Optimization: A New Application to Crew Pairing Problems, *24th AGIFORS Symposium* (1984), 15-29.

[25] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, New York, (1988).

[26] J. Rubin, A Technique for the Solution of Massive Set Covering Problems, with Applications to Airline Crew Scheduling, *Transportation Sci.* 7 (1973), 34-48.

[27] D.M. Ryan and J.C. Falkner, A bus crew scheduling system using a set partitioning mode, *Annals of Operations Research* 4 (1987), 39-56.

[28] D.M. Ryan and J.C. Falkner, On the Integer Properties of Scheduling Set Partitioning Models. *European Journal of Operations Research* 35 (1988), 442-456.

[29] P.H. Vance, C. Barnhart, E.L. Johnson and G.L. Nemhauser, Airline Crew Scheduling: A New Formulation and Decomposition Algorithm. *Operations Research*, Vol. 45. No. 2 (1997), 188-200.

[30] D. Wedelin, An Algorithm for Large Scale 0-1 Integer Programming with Application to Airline Crew Scheduling *Annals of Operations Research*, Vol. 57 (1995), 283-301.

[31] T.H. Wise, *Column Generation and Polyhedral Combinatorics for Airline Crew Scheduling*, Ph.D. Dissertation, Cornell U., Ithaca, N.Y. (1995).

Ranga Anbil
Mathematical Sciences
IBM Research
P.O Box 218
Yorktown Heights, NY 10598
anbil@watson.ibm.com

John J. Forrest
Mathematical Sciences
IBM Research
P.O Box 218
Yorktown Heights, NY 10598
forrest@watson.ibm.com

William R. Pulleyblank
Mathematical Sciences
IBM Research
P.O Box 218
Yorktown Heights, NY 10598
wrp@watson.ibm.com