Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

Ralf Borndörfer[1] Uwe Schelten[2]
Thomas Schlechte[1] Steffen Weider[1]

# A Column Generation Approach to Airline Crew Scheduling

[1]Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustr. 7, 14195 Berlin, Germany; Email:{borndoerfer, schlechte, weider}@zib.de
[2]Lufthansa Systems Berlin, Fritschestraße. 27-28, 10585 Berlin, Germany; Email: uwe.schelten@lhsystems.com

Supported by Lufthansa Systems Berlin

ZIB-Report 05-37 (August 2005)

# A Column Generation Approach to Airline Crew Scheduling [‡]

Ralf Borndörfer  Uwe Schelten  Thomas Schlechte
Steffen Weider

**Abstract**

The airline crew scheduling problem deals with the construction of crew rotations in order to cover the flights of a given schedule at minimum cost. The problem involves complex rules for the legality and costs of individual pairings and base constraints for the availability of crews at home bases. A typical instance considers a planning horizon of one month and several thousand flights. We propose a column generation approach for solving airline crew scheduling problems that is based on a set partitioning model. We discuss algorithmic aspects such as the use of bundle techniques for the fast, approximate solution of linear programs, a pairing generator that combines Lagrangean shortest path and callback techniques, and a novel "rapid branching" IP heuristic. Computational results for a number of industrial instances are reported. Our approach has been implemented within the commercial crew scheduling system NetLine/Crew of Lufthansa Systems Berlin GmbH.

## 1 The Airline Crew Scheduling Problem

The Airline Crew Scheduling Problem (CSP) plays a prominent role in the operations research literature not only because of its economic significance, but also because of its influence on the development of important mathematical techniques, among them branch-and-cut [7], branch-and-price [1], shortest path algorithms [4], stabilization [5], aggregation [10], and heuristics [11], see also the book of Yu [12] for a general overview.

The CSP can be described in terms of a *pairing digraph* $N = (V, A)$. Its nodes $V$ are called *tasks*. They can be subdivided into *legs* $L \subseteq V$ that model flights and have to be assigned to crews, *supplementary tasks* that model crew activities such as check-in and check-out, deadheading (flying as a passenger), and ground transports, and *artificial tasks*, among them two tasks $s$ and $t$ that model the beginning and the end of a pairing. The arcs

[‡]Supported by Lufthansa Systems Berlin.

$A$ are called *links*. They connect tasks that can be performed consecutively by a single crew. A digraph as just described is known as a leg-on-node network; we assume that it is acyclic.

Associated with $N$ is a set $R$ of *pairing resources* (flight duty time, landings, etc.), a set $K$ of *pairing types*, and a set $B$ of *base resources* (no of crews, production days, etc.). The links are labeled with costs $c \in \mathbb{Q}^A$, and pairing and base resource consumptions $w \in \mathbb{Q}^{A\times R}$ and $d \in \mathbb{Q}^{A\times B}$, respectively (node costs and resource consumptions can be adding to adjacent arcs). There are pairing resource limits $u_k \in \mathbb{Q}^R$ for each pairing type $k \in K$, and base resource limits $\ell \in \mathbb{Q}^B$.

A path $p$ in $N$ has cost $c_p := \sum_{a\in p} c_a$ and consumes pairing and base resources $w_p := \sum_{a\in p} w_a$ and $d_p := \sum_{a\in p} d_a$, respectively. A path $p$ is a *pairing* of pairing type $k$ if $w_p \leq u_k$. We assume w.l.o.g. (by introducing pairing type resources) that each pairing is of exactly one type. A *cover* is a set of pairings that contains each leg exactly once; a cover $C$ is a *schedule* if $d(C) := \sum_{p\in C} d_p \leq \ell$, its cost is $c(C) := \sum_{p\in C} c_p$. The CSP is to find a schedule of minimum cost.

Denoting by $\mathcal{P}$ the set of all pairings and introducing decision variables $x_p$ for each pairing, and slack variables $s_b$ and costs $c_b$ for each base resource, the CSP can be stated as

$$\text{(CSP)} \qquad \min \sum_{p\in\mathcal{P}} c_p x_p + \sum_{b\in B} c_b s_b$$

$$\sum_{p\ni v} x_p = 1 \qquad \forall\, v \in L \tag{1a}$$

$$\sum_{p\in\mathcal{P}} d_{bp} x_p - s_b \leq \ell_b \qquad \forall\, b \in B \tag{1b}$$

$$0 \leq x_p \leq 1 \qquad \forall\, p \in \mathcal{P} \tag{1c}$$

$$x_p \in \{0,1\} \qquad \forall\, p \in \mathcal{P}. \tag{1d}$$

Here, the *partitioning constraints* (1a) guarantee that every leg is covered exactly once; to ensure feasibility, we assume that there is a "slack" pairing type with single-leg pairings of high cost. Let $\mathcal{S} \subseteq \mathcal{P}$ be the set of these pairings, one for each leg. Similarly, the slack variables ensure feasibility of the *base constraints* (1b), which control base resource consumption; strict compliance can be forced by choosing sufficiently high costs $c_b$. Using the leg-pairing incidence matrix $A := (a_{vp})$, i.e., $a_{vp} = 1$ if $v \in p$ and 0 otherwise, and collecting costs and base resource consumptions in vectors $c_{\mathcal{P}} := (c_p)$, $c_B := (c_b)$, $c := (c_{\mathcal{P}}, c_B)$, and a matrix $D = (d_{bp})$, (CSP) reads

$$\text{(CSP)} \qquad \min\ c^{\mathsf{T}}(x,s) \quad Ax = \mathbb{1},\ Dx \leq \ell,\ 0 \leq x \leq \mathbb{1},\ x \in \{0,1\}^{\mathcal{P}}.$$

## 2 A Column Generation Algorithm

We use a column generation algorithm to solve (CSP). Denote by $\mathcal{S} \subseteq \mathcal{P}' \subseteq \mathcal{P}$ some subset of pairings, by $A' := A_{\cdot\mathcal{P}'}$ the submatrix of $A$ restricted to the pairings in $\mathcal{P}'$, and similarly $c'$, $x'$, and $B'$, by $A_p := A_{\cdot\{p\}}$ and $D_p := D_{\cdot\{p\}}$, and by

$$\begin{array}{lllll} \text{(MLP)} & \min\ c^{\mathsf{T}}(x,s), & Ax = \mathbb{1}, & Dx \leq \ell, & 0 \leq x \leq \mathbb{1} \\ \text{(RMLP)} & \min\ c'^{\mathsf{T}}(x',s), & A'x' = \mathbb{1}, & D'x' \leq \ell, & 0 \leq x' \leq \mathbb{1} \end{array}$$

the LP-relaxation associated with (CSP), the *master LP*, and the *restricted master LP*, respectively. Denoting for a given dual solution $(\pi, \mu)$ to (RMLP) (where $\pi$ is associated with the partitioning and $\mu$ with the base constraints) by $\overline{c}_p := c_p - \pi^{\mathsf{T}} A_p + \mu^{\mathsf{T}} D_p$ the *reduced cost* of pairing $p$ and by

$$\text{(PRICE)} \qquad \min_{p\in\mathcal{P}}\ \overline{c}_p$$

the *pricing problem* associated with (RMLP), our method can be outlined as follows. It tries to solve the master LP in a first phase. In a second phase, a plunging heuristic is started that fixes and generates pairings to (hopefully) produce a feasible solution. Such a method is known as a branch-and-generate algorithm [9]. We will sketch in this section three important components.

### 2.1 LP Solution

The proximal bundle method [8, 6] is a fast subgradient-type method for convex programming that can be used to solve Lagrangean relaxations of linear programs. It computes Lagrangean multipliers of the relaxed constraints, an approximate primal solution, and a bound on the optimum objective value of the original LP. We consider the Lagrange function arising from (CSP) by relaxing the partitioning and base constraints

$$L(\pi, \mu) := \pi^{\mathsf{T}} 1 - \mu^{\mathsf{T}} \ell + \min_{0 \leq x \leq 1} (c_{\mathcal{P}}{}^{\mathsf{T}} - \pi^{\mathsf{T}} A + \mu^{\mathsf{T}} D)x + \min_{s \geq 0} (c_B - \mu)^{\mathsf{T}} s$$

Applying the bundle method to compute $\max_{\pi \text{ free}, \mu \geq 0} L(\pi, \mu)$, the main work turns out to be the computation of the expressions $\lambda^{\mathsf{T}} A$ and $\mu^{\mathsf{T}} B$ in the function $L(\pi, \mu)$. We use an active set method to speed up this step.

It restricts the evaluation of $L$ to a subset $I \subset \mathcal{P}$ of pairings. This set $I$, the *active set*, gives rise to a function $L_I$ by replacing $A$, $D$, and $c_{\mathcal{P}}$ by submatrices $A_{\cdot I}$, $D_{\cdot I}$, and $c_I$. We have $L_I(\pi, \mu) \geq L(\pi, \mu)$ for all $\pi$ and for all $\mu \geq 0$, and it is easy to see that $L_I(\pi, \mu) = L(\pi, \mu)$ holds if $\overline{c}_p = c_p - \pi^{\mathsf{T}} A_p + \mu^{\mathsf{T}} D_p \geq 0$ for all pairings $p$. We use this observation to restrict $I$ to pairings $p$ with reduced cost $\overline{c}_p \leq \epsilon$ for some threshold $\epsilon > 0$. We update the active set $I$ only if the so-called *stability center* of the bundle method changes. This can lead to situations where the active set does not contain all columns with $\overline{c}_p < 0$, which can result in a model of $L_I$ that overestimates the real value of $L$ at some points. If we notice that, we repair the model of $L_I$.

## 2.2 Column Generation

As all pairings end in the non-leg task $t$, we can define the *reduced cost of an arc* $ij \in A$ as $\overline{c}_{ij} := c_{ij} - \sum_{v=i} \pi_v + \sum_{b \in B} \mu_b d_{ij,b}$ and the pricing problem to construct a pairing of type $k$ of negative reduced cost becomes a constrained shortest path problem in the acyclic digraph $N$:

$$\text{(PRICE)} \qquad \min \sum_{a \in A} \overline{c}_a x_a$$

$$\sum_{a \in \delta^{\text{out}}(v)} x_a - \sum_{a \in \delta^{\text{in}}(v)} x_a = \delta_{st}(v) \qquad \forall\, v \in V \tag{2a}$$

$$\sum_{a \in A} w_{ar} x_a \le u_{kr} \qquad \forall\, r \in R \tag{2b}$$

$$0 \le x_a \le 1 \qquad \forall\, a \in A \tag{2c}$$

$$x_a \in \{0,1\} \qquad \forall\, a \in A. \tag{2d}$$

Here, $\delta_{st}(v) = 1$ if $v = s$, $\delta_{st}(v) = -1$ if $v = t$ and $\delta_{st}(v) = 0$ else. We solve this problem using a branch-and-bound algorithm similar to [2], using lower bounds derived from a Lagrangean relaxation of the resource constraints (2b), see [3] for more details. Using configurable classes of linear resource constraints and multilabel methods, we can handle most pairing construction rules directly. Some rules, however, are so complex, that these techniques would become unwieldy or require too much customization. For such cases, we use a callback mechanism, that is, we ignore the rule in our pricing model, construct a pairing, and send it to a general *rule verification oracle* that either accepts or rejects the pairing.

## 2.3 IP Heuristic

The idea of our "rapid branching" heuristic is to produce a solution quickly by iteratively solving the (RMLP) using the bundle method and fixing large numbers of pairing variables to one. The fixed variables are selected according to their $x$-value, i.e., the closer $x_p$ is to 1, the higher the probability that $x_p$ is fixed to 1. In order to have a large number of variables with values close to 1 available, the heuristic perturbs in each iteration the objective function according to the formula $c_p := c_p(1 - \alpha x_p^2)$ ($\alpha$ is a control parameter), which favors such pairings; we call this method "perturbation branching", see also [11] for a similar idea. Between fixes, pairings are generated to complement the fixes; backtracks, i.e., unfixings of pairings, are also performed sometimes. The frequency and intensity of column generation and backtracks is

Table 1: Test Scenarios.

| Name | Scenario 1 | Scenario 2 |
|---|---|---|
| #Days | 14 | 31 |
| #Home Bases | 3 | 2 |
| #Pairing Types | 4 | 2 |
| #Legs | 4104 | 2154 |
| #Tasks | 32832 | 14373 |
| #Links | 1438659 | 168352 |

controlled by *targets*, i.e., estimates on the increase of the objective function under fixings, see also [9]. If the objective develops as expected, no pairings are generated; if it increases more than expected, we try to correct the problem by generating new pairings, if this does not work, we backtrack, and if we are out of time, we output the best solution found.

# 3 Computational Results

We now present computational results on industrial data provided by Lufthansa Systems Berlin, see Table 1. Scenario 1 is a 14-days problem with linear pairing rules; the main objective was to minimize the number of production days, secondary objectives were to minimize flight transports and rest periods. We consider two variants, an unconstrained scenario and one with 56 base constraints on the distribution of crews at home bases and of pairing types for each day. Scenario 2 is a 31-day instance with more complex rules, some of which had to be handled with callbacks. All computations were made single threaded on a Dell Precision 650 PC with 2GB of main memory and a dual Intel Xeon 3.2 GHz CPU running SUSE Linux 9.3.

Comparing the bundle method to an exact LP solver within a column generation algorithm is not straightforward. Bundle solves an individual LP clearly faster, but it produces approximate solutions, which may lead to

Table 2: Solving the Reduced Master LP.

| Scenario 1 | Unconstrained with Bundle | Unconstrained with CPLEX | Base Constraints with Bundle | Base Constraints with CPLEX |
|---|---|---|---|---|
| LP-Value | 833 | 836 | 859 | 860 |
| IP-Value | 835 | 836 | 862 | 864 |
| #Production days | 1055 | 1057 | 1089 | 1091 |

**Table 3:** Constructing Pairings with Lagrangean Pricing and Callbacks.

| Scenario 2 | with Deadheads | without Deadheads |
|---|---|---|
| LP-Value | 2977 | 3757 |
| IP-Value | 2981 | 3762 |
| #Production days | 1668 | 1660 |
| Time | 8h | 38 min. |
| Callbacks failed/overall | 174130/270378 | 0/58648 |
| in % | 64.40 | 0.00 |

more pairing generation iterations. Table 2 reports the results of running our optimizer on scenario 1, solving RMLPs with our bundle code and with the barrier implementation of CPLEX 9.0; the overall time limit was 2 days. It can be seen that there is no loss of solution quality using the bundle method, and that the rapid branching heuristic constructs a solution with an objective value that almost equals that of the master LP.

Table 3 gives some details on pairing construction for scenario 2. This scenario involves a non-linear rule on the reduction of resources in the presence of deadheads, which was modeled using callbacks. It can be seen that the percentage of rejected pairings is relatively high, because the pairing generator tends to produce infeasible pairings repeatedly. The scenario can, however, be handled successfully.

It is not only possible to model this particular rule in a linear way. We are currently working on more general classes of linear and multilabel rules in order to cover all important rules directly in the pairing generator. We are confident that we can improve the performance of our optimizer significantly in this way in the future.

## References


[1] 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 Res., 46 (1998), pp. 316–329.

[2] J. E. Beasley and N. Christofides, *An algorithm for the resource constrained shortest path problem*, Networks, 19 (1989), pp. 379–394.

[3] R. Borndörfer, M. Grötschel, and A. Löbel, *Duty scheduling in public transit*, in Mathematics – Key Technology for the Future, W. Jäger and H.-J. Krebs, eds., Springer, 2003, pp. 653–674.

[4] M. Desrochers, *A new algorithm for the shortest path problem with resource constraints*, Tech. Rep. 421A, Centre de Recherche sur les Transports, Univ. Montréal, 1986.

[5] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen, *Stabilized column generation*, Discrete Math., 194 (1999), pp. 229–237.

[6] C. Helmberg and K. C. Kiwiel, *A spectral bundle method with bounds*, Math. Prog., 2 (2002), pp. 173–194.

[7] K. Hoffman and M. W. Padberg, *Solving airline crew scheduling problems by branch-and-cut*, Management Sci., 39 (1993), pp. 657–682.

[8] K. C. Kiwiel, *Proximal bundle methods*, Math. Prog., 46 (1990), pp. 105–122.

[9] R. Marsten, *Crew planning at delta airlines.* Talk at the 15$^{\text{th}}$ Int. Symp. Math. Prog., 1994.

[10] D. Villeneuve, J. Desrosiers, M. E. Lübbecke, and F. Soumis, *On compact formulations for integer programs solved by column generation*, Tech. Rep. No. 2003/25, TU Berlin, 2003.

[11] D. Wedelin, *An algorithm for a large scale 0-1 integer programming with application to airline crew scheduling*, Ann. Oper. Res., 57 (1995), pp. 283–301.

[12] G. Yu, *Operations Research in the Airline Industry*, Kluwer, 1997.