

Tabu Search-Based Algorithm for Large Scale Crew Scheduling Problems¹

Marco Caserta*

Abstract

In this paper, the problem of finding a work schedule for airline crew members in a given time horizon is tackled. This problem is known in the literature as Airline Crew Scheduling. The objective is to define the minimum cost schedules where each crew, associated to a combination of commercial flights or “legs” called “pairing”, is assigned to one or more flights ensuring that the whole set of flights is covered by crew members. The Crew Scheduling Problem can be modeled by using the Set Covering formulation. This paper presents a new algorithm whose centerpiece is a primal-to-dual scheme aimed at linking any primal solution to the dual feasible vector that best reflects the quality of the primal solution. This new mechanism is used to intertwine a tabu search based, primal intensive, scheme with a lagrangian based, dual intensive, scheme to design a primal-dual algorithm that progressively reduces the gap between upper and lower bound.

The algorithm has been tested on benchmark problems from the literature. In this paper, results on real-world airline instances are presented: out of six well-known problems, the algorithm is able to match the optimal solution for four of them while for the last two, whose optimal solution is not known, a new best known solution is found.

Key words: set covering, tabu search, metaheuristic, lagrangian optimization, primal-to-dual.

¹ The author is grateful to the anonymous referees for their useful comments, and to Alberto Caprara for providing the instances presented in Table 1.

* Profesor Investigador del Instituto Tecnológico y de Estudios Superiores de Monterrey, *campus* Ciudad de México. Correo electrónico: marco.caserta@itesm.mx

1. Introduction

The Airline Crew Scheduling problem is a well know problem where one desires to compute the set of minimum pairings covering the whole set of scheduled flights of an airline. A pairing is a sequence of flights covered by a crew, where the starting point and the ending point are the crew member's home base. The typical length of a pairing, or leg, is comprised between one day and five or six days. A leg is an individual flight segment. Consequently, a pairing is made up by a set of legs, where the first leg departs from the same city where the last leg ends. Each pairing corresponds to a possible flight schedule for a single flight crew. Finally, a crew schedule is a set of pairings that covers all the legs at exactly once (or at least once if deadheading, which means flight without working, is allowed.)

As explained by Anbil *et al.* (1998), the crew scheduling problem is the third stage of a four-stage process. The first stage of the airline planning problem is the “Schedule Creation”, which consists of the definition of a plan of flights, based on marked demand. For example, it is possible to schedule a flight from Mexico City to Monterrey every weekday at 9:00A.M. The flight from Mexico City to Monterrey and return is called “leg”.

The second step of the planning process is made up by the “Fleet Schedule”. The objective of this problem is to choose a fleet for each flight defined in the first step. The assignment of the fleet to the flight needs to take into account capacity constraints and flying characteristics. The objective at this stage is to minimize the operating costs of the aircraft.

The third stage is the “Crew Scheduling”, where, as mentioned, each leg, or individual flight segment, is assigned to a pairing, or crew. The problem is modeled using the Set Covering Problem, where one aims at finding the minimum set of columns (pairings) that can cover the whole set of rows (commercial flights, or legs) of the problems.

Finally, the forth and last stage of the planning problem is the “Crew Rostering”, where each individual crew member is assigned to each pairing according to different mechanisms, such as seniority, auction, etc.

Several factors contribute to enhance the difficulty associated to the crew scheduling problem, such as, for example:

- ♦ The feasibility of pairing is governed by a complex set of rules, both of governmental nature (FAA) as well as union-related. For example, typical restrictions are those imposed on the maximum duration of a pairing length in terms of days, the maximum duty flight time, the minimum time allowed between connecting flight and the maximum number of aircraft changes permitted.
- ♦ The calculation of the pairing cost is complex, usually taking into account several non linear components. These include crew pay as well as hotel costs, etc.
- ♦ The total number of feasible pairs is very large, hence increasing the size of the corresponding crew scheduling problem. For this reason, it is not feasible to generate all the possible pairings.

Let N be the set of all feasible pairings. For each pair “ j ”, we define a binary variable x_j whose value is 1 if the pairing x_j is in the minimum set and 0 otherwise. Let c_j be the cost of pairing j , which is given for this problem. In addition, let us indicate with M the whole set of commercial flights, or legs, to be covered.

The Crew Scheduling Problem is modeled using the Set Covering formulation. The Set Covering Problem (SC) is a 0-1 integer problem, defined as

$$(SC) \quad \left| \begin{array}{ll} \min & z = \sum_{j \in N} c_j x_j \\ & s.t. \quad \sum_{j \in N} a_{ij} x_j \geq 1, \quad \forall i \in M \\ & \quad \quad x_j \in \{0,1\}, \quad \forall j \in N \end{array} \right.$$

where $a_{ij} = 1$ if pairing j serves leg i and 0 vice versa. In the following, we call a *cover* a vector $\mathbf{x} \in \{0,1\}^n$ that is a feasible solution of SC, and a *prime cover* is a cover with no redundant columns. Also, let $J_i = \{j \in N: a_{ij}=1\}$ be the index set of columns covering row i , and $I_j = \{i \in M: a_{ij}=1\}$ the index set of rows covered by column j .

Typically, Crew Scheduling Problems are characterized by the existence of millions of columns (pairings) and hundreds, or thousands in the case of the biggest carriers, of rows (commercial flights, or legs.)

SC is *NP*-complete (*e.g.*, Garey and Johnson 1979), hence exact solution procedures are doomed to fail in solving practical SC problems. Supported by its applicability, a great deal of effort has been directed toward the development of approximate algorithms for SC. As a result, some algorithms are capable of solving SCs with thousands of rows and millions of columns (*e.g.*, Ceria *et al.* 1998, Caprara *et al.* 1999).

To summarize, most approximate solution procedures for SC are dual heuristic procedures based upon the lagrangian relaxation of SC and subgradient optimization (*e.g.*, Balas and Ho 1980, Vaasko and Wilson 1984, Fisher and Kedia 1990, Balas and Carrera 1996, Ceria *et al.* 1998, Caprara *et al.* 1999). As the dual procedures require greedy-type primal heuristics in order to build a primal cover based upon the dual solution obtained via subgradient optimization, they can also be viewed as primal-and-dual algorithms with “dual-to-primal” mechanisms. In addition, more “advanced” dual procedures for SC in the literature typically feature some form(s) of probing and variable fixing schemes that dynamically update primal and dual information of SC and aid in more effective solution of SC (*e.g.*, Beasley 1990, Balas and Carrera 1996, Ceria *et al.* 1998, Caprara *et al.* 1999).

A major contribution of the paper is the development of a “primal-to-dual” (p2d) mechanism that, for any given primal solution, constructs a feasible dual vector that minimizes the gap between the upper bound of SC given by *the* cover and the lower bound by a feasible dual solution with respect to the sufficient optimality conditions that we provide in Theorem 1. The benefit of the primal-to-dual mechanism is two-fold: (i) if the current cover is optimal to SC, it verifies the optimality and the search process can be terminated; (ii) otherwise, it constructs a dual vector \mathbf{u} that serves as a new starting vector for subgradient optimization. If different prime covers are provided, the primal-to-dual scheme constructs different \mathbf{u} 's, allowing subgradient optimization to explore different regions of the dual solution space and converge to respective local solutions therein. This, in turn, allows greedy-type dual-to-primal heuristics to construct different prime covers for SC.

Integrating effective dual-to-primal mechanisms from the literature and the novel primal-to-dual mechanism with a specialization of the tabu search metaheuristic of Caserta and Ryoo (2001) for solving SC, in this paper we develop a primal-intensive, “dynamic” primal-and-dual metaheuristic for large-scale SC with $m \ll n$. Computational experiments with the proposed metaheuristic on 6 well known instances taken from the airline industry indicate that the proposed algorithm

advances the state-of-art in SC quite substantially. In order to prove the robustness of the algorithm, we tested it on the Beasley's OR Library. Out of 94 benchmark problems, 21 of them have not been solved to optimality. For 16 of the 21 problems, our algorithm reduces the gap between and best lower and upper bounds: the algorithm finds new best solutions for 3 problems and improves the lower bounds for 15 problems. For the 73 benchmarks solved to optimality, the proposed algorithm finds the optimal solutions. In this paper, we limit the presentation of the results to those problems taken from the airline industry, provided by Wedelin (1995).

The proposed algorithm is made up of dynamically interacting and intertwined (meta)-heuristic components that synergistically contribute to the efficiency and efficacy of the proposed algorithm. We present these (meta)-heuristic components of the proposed algorithm in Sections 2 to 7, both from the primal perspective and from the dual perspective. Later, a complete presentation of the overall algorithm is offered in Section 8. Computational experiments with 6 benchmark problems are summarized in Section 9 and concluding remarks are provided in Section 10.

2. Tabu Search Metaheuristic

The tabu search metaheuristic of the proposed algorithm is based upon the work of Glover (See Glover 1989, Glover 1990, and Glover and Laguna 1997) and uses the result of a specialization of the metastrategy provided in Caserta and Ryoo (2003) for SC. For reasons of space, we provide details for those components that are problem-specific in nature for SC.

Let \mathbf{x}^k denote the current prime cover. In the k^{th} iteration of tabu search, we move from \mathbf{x}^k to a different prime cover \mathbf{x}^{k+1} through a sequence of moves to \mathbf{x}^{k_t} , $t=1, \dots, T_k$, some of which are covers and the others are not. Let $\mathbf{x}^0 = \mathbf{x}^k$ and $\mathbf{x}^{k_{T_k}} = \mathbf{x}^{k+1}$. For simplicity, \mathbf{x}^{k_t} will be denoted as \mathbf{x}^t in the remainder of this section. For reasons of space, the procedural steps will be presented in a pseudo-language form with comments provided in brackets.

2.1 Allowed Infeasible Region and Attractive Feasible Space

One of the basic ideas of the proposed algorithm is related to the definition of sub-areas of the feasible and infeasible spaces. We differentiate between the "Allowed Infeasible Region" and the "Attractive Feasible Space". The former enhances the search space by including a portion of the infeasible space thought to be "attractive".

On the other hand, the latter reduces the search space by “cutting out” those areas of the feasible spaces thought to be “non-attractive”, hence limiting the search to a portion of the whole feasible space.

For $\mathbf{x}^t \in \{0,1\}^n$ ($t=0, \dots, T_k$), denote by B the index set of columns that take value 1 in \mathbf{x}^t . Let $M^0 = \{i \in M : J_i \cap B = \emptyset\}$, $M^1 = \{i \in M : |J_i \cap B| = 1\}$, and $M^2 = \{i \in M : |J_i \cap B| = 2\}$ denote the set of rows that are uncovered, the set of rows that are uniquely covered, and the set of rows that are covered twice by \mathbf{x}^t , respectively. Furthermore, let $I_j^+ = I_j \cap M^0$ for $j \in N \setminus B$ and $I_j^- = I_j \cap M^1$ for $j \in B$. Let TL denote the list of current tabu moves.

With the notation above, the feasible space of SC can be defined as $\mathbf{X} := \{\mathbf{x} \in \{0,1\}^n : |M^0| = 0\}$.

In contrast to \mathbf{X} , let us define the “allowed infeasible space” of SC as

$$\bar{\mathbf{X}} := \{\mathbf{x} \in \{0,1\}^n : |M^0| \leq \alpha m\},$$

where α is a predetermined parameter chosen in $[0, 1)$. A key feature of the proposed tabu search metaheuristic is its ability to escape from a locally optimal solution via an excursion into an allowed infeasible space. Owing to the monotone decreasing property of the objective function in \mathbf{x} , solutions in $\bar{\mathbf{X}}$ are, usually, more attractive than the feasible solutions. Hence, even if \mathbf{x}^k is a locally dominant prime cover, the search path will be able to escape from it to a remote, different prime cover \mathbf{x}^{k+1} through a sequence of $\mathbf{x}^t \in \mathbf{X} \cap \bar{\mathbf{X}}$.

In addition, let us define the set of “attractive” covers of SC as

$$\mathbf{X}^* := \{\mathbf{x} \in \mathbf{X} : |M \setminus M^1| \leq \beta m\},$$

where $\beta \in [0, 1)$ is a predetermined parameter. Denote the set of prime covers by \mathbf{X}^p . Let $j \bullet$ indicate a 1-neighborhood move that sets the j^{th} component of \mathbf{x}^t to 1 if $\bullet = +$ (a set covering move) and to 0 if $\bullet = -$ (a set releasing move) and let $\bar{\bullet}$ denote the move in the opposite direction of \bullet . Then, each composite move from \mathbf{x}^t to \mathbf{x}^{t+1} is comprised of a sequence of a finite number of 1-neighborhood moves, and the choice of the first move plays a critical role in the proposed metastrategy.

2.2 Selection of First Move and Three Types Moves

As mentioned in the previous section, each iteration of the algorithm is characterized by a composite move, made up by a first move, chosen according to information

related to search status, memory and learning mechanisms, and a second move, which is directly related to the choice of the first move.

In light with this observation, it is vital to devise “intelligent” mechanisms aimed at choosing the first move in the current neighborhood. For the selection of j_1 , the first move, we design four 1-neighborhood move selection criteria. Let us first identify four sets with variables, rearranged in the order of four scores that are employed in the tabu search mechanism:

$$\Gamma_1^- = \{j_l \in B, I_{j_{l1}}^- \neq 0, j_l^- \notin TL : c_{j_l} | I_{j_l}^- | \geq c_{j_{l+1}} | I_{j_{l+1}}^- | \}$$

$$\Gamma_2^- = \left\{ j_l \in B, I_{j_{l1}}^- \neq 0, j_l^- \notin TL : \frac{c_{j_l}}{I_{j_l}^- \cap M^2} \geq \frac{c_{j_{l+1}}}{I_{j_{l+1}}^- \cap M^2} \right\}$$

$$\Gamma_1^+ = \{j_l \in N - B, I_{j_{l1}}^+ \neq 0, j_l^+ \notin TL : \bar{r}_{j_l} \leq \bar{r}_{j_{l+1}} \}$$

$$\text{where } \bar{r}_{j_l} = c_j - \sum_{i \in I_j^+} u_i ,$$

$$\Gamma_2^+ = \left\{ j_l \in N - B, I_{j_{l1}} \cap M^1 \neq 0, j_l^+ \notin TL : \frac{\bar{r}_{j_l}}{I_{j_l} \cap M^1} \leq \frac{\bar{r}_{j_{l+1}}}{I_{j_{l+1}} \cap M^1} \right\}$$

To allow the search path to deviate from following a pre-determined trajectory given by the use of the greedy merit functions, we select j_1^t probabilistically. Each move is, in turn, classified as either a regular, diversified or intensified move depending upon the way j_1^t is selected.

Remark. In order to allow for a more rigorous search of the solution space, we recur to three different strategies that define three search phases of the algorithm, namely the regular, diversification, and intensification phases by adapting the probabilistic scheme.

2.3. Move Assignment Scheme

The next step, after choosing the first move in the current neighborhood, is to define the composite move, which allows reaching a new point in the feasible

space. In this section, we present the procedure used to define the composite move. The presentation is offered in the form of “meta-language” code.

Denote by $\bullet \mathbf{e}_1$ a unit vector whose 1th component is -1 if $\bullet = -$ and +1 otherwise.

procedure *Composite Move Assignment*;

input: \bullet, \mathbf{x}^t , TL, (core) problem instance

output: $\bullet, \mathbf{x}^{t+1}$, TL

begin

10 call *Select First Move*

if $\mathbf{x}^t + \bullet \mathbf{e}_{j_1} \notin \mathbf{X}^* \cup \bar{\mathbf{X}}$ **then** $\bullet \leftarrow$, go to line 10

$j_1^t \leftarrow j_1$

TL \leftarrow TL \cup

if $\mathbf{x}^t \in \mathbf{XP} \cup \bar{\mathbf{X}}$ **then**

if $|I_{j_1^t}^*| \geq \sum_{j \in \Gamma_1^+} |I_j^*|$ **then** $\varsigma^{\bar{\bullet}} := \Gamma_1^{\bar{\bullet}}$

else

if $\bullet = -$ **then** $\varsigma^+ := \{j_1, \dots, j_i \in \Gamma_1^+ : |I_{j_1^t}^-| \geq \sum_{j=j_1}^{j_i} |I_j^+|, |I_{j_1^t}^-| < \sum_{j=j_1}^{j_{i+1}} |I_j^+|\}$

else $\varsigma^- := \{j_1, \dots, j_i \in \Gamma_1^- : |I_{j_1^t}^+| \geq \sum_{j=j_1}^{j_i} |I_j^-|, |I_{j_1^t}^+| < \sum_{j=j_1}^{j_{i+1}} |I_j^-|\}$

endif

elseif $\mathbf{x}^t \hat{=} \mathbf{X}^* \setminus \mathbf{X}^P$ **then**

if $|I_{j_1^t}^*| \geq \sum_{j \in \Gamma_2^+} |I_j^*|$ **then** $\varsigma^{\bar{\bullet}} := \Gamma_2^{\bar{\bullet}}$

else

if $\bullet = -$ **then** $\varsigma^+ := \{j_1, \dots, j_i \in \Gamma_2^+ : |I_{j_1^t}^-| \geq \sum_{j=j_1}^{j_i} |I_j^+|, |I_{j_1^t}^-| < \sum_{j=j_1}^{j_{i+1}} |I_j^+|\}$

else $\varsigma^- := \{j_1, \dots, j_i \in \Gamma_2^- : |I_{j_1^t}^+| \geq \sum_{j=j_1}^{j_i} |I_j^-|, |I_{j_1^t}^+| < \sum_{j=j_1}^{j_{i+1}} |I_j^-|\}$

endif

endif


```

if  $\mathbf{cx}^{k+1} < \text{UB}$  then           { update primal information }
     $\mathbf{x}^* \leftarrow \mathbf{x}^{t+1}$ 
     $\text{UB} \leftarrow \mathbf{cx}^{t+1}$ 
endif
if  $\mathbf{x}^t \in$  or  $\bullet = +$  then           { end tabu iteration }
    solve (p2d)                       { see Section 5 }
    partial pricing                     { see Section 6 }
    call Lagrangian Optimization     { see Section 3 }
endif
endif
end while
end for
end

```

3. Lagrangian Relaxation and Greedy Heuristics

The best known primal heuristic is the greedy one, which uses the reduced cost information provided by the dual phase to construct a prime cover. Balas and Ho (1980) presented a list of scores based upon the column cost per row covered to create a prime cover. Vaasko and Wilson (1984) selected a column to be added to the partial cover according to the value of a score function, randomly chosen among a pool of functions based upon the column cost per row covered. At each iteration the primal heuristic is run 30 times with randomly chosen score functions. Beasley (1990a) proposed a Lagrangian based primal heuristic scheme that extended the partial cover of the Lagrangian problem to a prime cover. A score based upon the column cost per row covered is used to rank the columns. Fisher and Kedia (1990) proposed as score the reduced cost computed using only the multipliers of rows left uncovered, rather than the actual reduced cost. Bricker and Techapicjetvanich (1993) studied the effectiveness of five different primal heuristic scores, based upon the column cost per row covered and the reduced cost per row covered, both the real and the modified reduced cost. Balas and Carrera (1996) coupled the approach of Vaasko and Wilson (1984) with a primal scheme that creates a prime cover extending the partial cover of the Lagrangian phase by choosing columns based upon their reduced cost. The primal scheme, as a byproduct, produces an improved dual vector.

Let M^* denote the set of rows left uncovered by \mathbf{x} , and B denotes the set of columns fixed to 1 in the current (partial) cover \mathbf{x} . Let $k_1 = |I_j \cap M^*|$ be the number

of rows currently uncovered that would be covered by setting x_j to 1. During the dual lagrangian phase we use the score:

$$s = \bar{r}_j \sum_{i \in I_j \cap M^*} u_i$$

in a fashion similar to what presented in Caprara *et al.* (1999).

4. Subgradient Optimization

The lagrangian relaxation of SC is defined as

$$L(u) = \min_{\mathbf{x} \in \{0,1\}} \sum_{j=1}^n r_j x_j + \sum_{i=1}^m u_i, \quad (\text{LP})$$

where r_j (the reduced cost for $j, j=1, \dots, n$) is defined as $c_j - \sum_{i \in I_j} u_i$ and requires \mathbf{u} such that a vector \mathbf{x}^L minimizing the lagrangian function can be computed by a standard technique:

$$x_j^L = \begin{cases} 1, & \text{if } c_j - \sum_{i \in I_j} u_i < 0 \\ 0, & \text{otherwise} \end{cases}$$

Most successful approaches for SC in the literature solve a series of lagrangian relaxation of SC and use the subgradient optimization technique to generate the vector \mathbf{u} for the lagrangian optimization.

For subgradient optimization, we use the formula of Held and Karp (1971)

$$u_i^{k+1} = \max \left\{ u_i^k + \lambda \frac{\text{UB} - \text{LB}}{\|s(u^k)\|^2} s_i(u^k), 0 \right\}, i \in M$$

where UB and LB are the upper and lower bounds of the optimum of SC, λ is the step size parameter, and $s_i(\mathbf{x}^k) = 1 - \sum_{j \in I_i} x_j$ is the component i of the subgradient. As in Caprara *et al.* (1999), \mathbf{u}^0 is initialized as

$$u_i^0 = \min_{j \in J_i} \frac{c_j}{|I_j|}, \quad i \in M$$

and \tilde{e} is updated after every $p = 20$ iterations, utilizing the best and worst lower bounds information obtained during the last p iterations. In addition, if the lower bound improvement in the last $4p$ iterations is below the threshold limit of 1% we apply a “perturbation scheme” based upon the primal-to-dual scheme of 5 to enforce a drastic modification of the vector \mathbf{u} . Furthermore, whenever the “perturbation scheme” is applied for more than 5 times in the same lagrangian phase, we introduce an amplified modification of the vector \mathbf{u} .

5. Primal-to-Dual Scheme

Let \mathbf{x}^p denote a (prime) cover for SC. Similarly, let \mathbf{x}^L denote an optimal solution to the Lagrangian problem with respect to the current dual vector \mathbf{u} . Denote by $z(\bullet)$ and $L(\bullet, \mathbf{u})$ the objective value of SC and the value of the Lagrangian function evaluated at \bullet , respectively. Let $B^p = \{j \in N : x_j^p = 1\}$ and $B^L = \{j \in N : x_j^L = 1\}$.

Lemma 1. *Suppose that given $\mathbf{x}^p \in \{0,1\}^n$, $\mathbf{x}^L \in \{0,1\}^n$, and $\mathbf{u} \in \Re_+^m$ satisfy $u_i \times (\sum_{j \in J_i} x_j^p - 1) = 0$, for all $i \in M$. Then, $z(\mathbf{x}^p) - L(\mathbf{x}^L, \mathbf{u}) = \sum_{j \in B^p - B^L} r_j - \sum_{j \in B^L - B^p} r_j$.*

Proof. According to the definition of the Lagrangian function (see (LP) at Section 4), we have

$$L(\mathbf{x}^L, \mathbf{u}) = \sum_{i \in M} u_i + \sum_{j \in B^L} r_j$$

The Lagrangian function can be rewritten as

$$\begin{aligned}
 L(\mathbf{x}^L, \mathbf{u}) &= \sum_{i \in M} u_i + \sum_{j \in B^L} \left(c_j - \sum_{i \in I_j} u_i \right) \\
 &= \sum_{i \in M} u_i \sum_{j \in J_i} x_j^P + \sum_{j \in B^L} c_j - \sum_{j \in B^L} \sum_{i \in I_j} u_i \\
 &= \sum_{j \in B^L} c_j + \sum_{j \in B^P} \sum_{i \in I_j} u_i - \sum_{j \in B^L} \sum_{i \in I_j} u_i \\
 &= \sum_{j \in B^L} c_j + \sum_{j \in B^P - B^L} \sum_{i \in I_j} u_i - \sum_{j \in B^L - B^P} \sum_{i \in I_j} u_i
 \end{aligned}$$

where the second equality is obtained via $u_i \times (\sum_{j \in J_i} x_j^P - 1) = 0, \forall i \in M$. Now, we have

$$z(\mathbf{x}^P) = \sum_{j \in B^P} c_j.$$

Consequently, we can write

$$\begin{aligned}
 z(\mathbf{x}^P) - L(\mathbf{x}^L, \mathbf{u}) &= \sum_{j \in B^P} c_j - \sum_{j \in B^L} c_j - \sum_{j \in B^P - B^L} \sum_{i \in I_j} u_i + \sum_{j \in B^L - B^P} \sum_{i \in I_j} u_i \\
 &= \sum_{j \in B^P - B^L} c_j - \sum_{j \in B^L - B^P} c_j - \sum_{j \in B^P - B^L} \sum_{i \in I_j} u_i + \sum_{j \in B^L - B^P} \sum_{i \in I_j} u_i \\
 &= \sum_{j \in B^P - B^L} \left(c_j - \sum_{i \in I_j} u_i \right) - \sum_{j \in B^L - B^P} \left(c_j - \sum_{i \in I_j} u_i \right) \\
 &= \sum_{j \in B^P - B^L} r_j - \sum_{j \in B^L - B^P} r_j
 \end{aligned}$$

Theorem 1 (Sufficient Conditions). Suppose that given $\mathbf{x}^P \in \{0,1\}^n$, $\mathbf{x}^L \in \{0,1\}^n$, and $\mathbf{u} \in \mathfrak{R}_+^m$ satisfy:

- (i) $u_i \left(\sum_{j \in J_i} x_j^P - 1 \right) = 0, \quad \forall i \in M$
- (ii) $r_j = c_j - \sum_{i \in I_j} u_i, \quad \forall j \in B^P$
- (iii) $r_j = c_j - \sum_{i \in I_j} u_i \geq 0, \quad \forall j \in N - B^P$

Then, \mathbf{x}^P solves SC.

Proof. Immediate.

The sufficient optimality conditions of \mathbf{x}^* for SC in Theorem 1 can be exploited in the derivation of a mechanism that constructs a “feasible” dual solution \mathbf{u} that properly reflects the importance of each constraint of SC with respect to the characteristics of \mathbf{x}^P .

First, note that Conditions (ii) and (iii) of Theorem 1, along with the requirement $\mathbf{u} \in \mathfrak{R}_+^m$ give the feasibility of \mathbf{u} to the dual linear program of the linearized SC. Conditions (i) and (ii), along with $\mathbf{x}^* \in \{0,1\}^n$ ensure that the primal and dual solutions are optimal to their respective programs. Let $M^0 := \{i \in M : \sum_{j \in J_i} x_j^P > 1\}$, $\overline{M}^0 := M - M^0$ and $N_c \subseteq N$ with $|N_c| \ll |N|$. Consider the following linear program:

$$(p2d) \quad \left| \begin{array}{ll} \min & g = \sum_{j \in B^P - B^L} \left(c_j - \sum_{i \in I_j} u_i \right) \\ & c_j - \sum_{i \in I_j} u_i = 0, \quad j \in B^L \cap B^P \\ & c_j - \sum_{i \in I_j - M^0} u_i \geq 0, \quad j \in N_c - B^P \\ & u_i \geq 0, \quad i \in \overline{M}^0 \end{array} \right|$$

First, note that (p2d) is an LP with $|(N^c \setminus B^P) \cup (B^P \cap B^L)|$ rows and $|\overline{M}^0|$ columns. Also, note that the two non-trivial constraints of (p2d) set $u_i = 0$ for all $i \in M^0$, and, through the minimization process, (p2d) modifies the remaining components of the vector \mathbf{u} feasible to the dual of the linearized SC that satisfies the sufficiency conditions of Theorem 1 “as much as possible” to yield \mathbf{u} that reflects the characteristics of \mathbf{x}^P .

The following is an obvious consequence of (p2d) and Theorem 1:

Corollary 1. *If the optimum of (p2d) is equal to zero, then \mathbf{x}^P solves SC.*

The following also holds true:

Theorem 2. *Of all dual feasible $\mathbf{u} \in \mathcal{R}_+^m$, \mathbf{u}^* obtained from solving (p2d) minimizes the gap $z(\mathbf{x}^P) - L(\mathbf{x}^L, \mathbf{u}^*)$ with respect to \mathbf{x}^P and \mathbf{x}^L and Condition (i) of Theorem 1.*

Proof. The dual feasibility of \mathbf{u}^* is immediate. The formulation of (p2d) and Theorem 1 easily show that $z(\mathbf{x}^P) - L(\mathbf{x}^L, \mathbf{u}^*)$ is minimized by the \mathbf{x}^P and \mathbf{u}^* pair.

6. Variable Fixing with Probing Techniques

Ceria *et al.* (1995) fixed a variable to zero if its reduced cost is greater than the gap between upper and lower bound. Balas and Carrera (1996) computed a factor \ddot{A}_j for every column $j \in N \setminus B$ defined as the improvement in the value of the vector \mathbf{u} obtained by fixing x_j to one. Subsequently, one column j is fixed to zero if $\lceil LB + r_j + \ddot{A}_j \rceil \geq UB$. In this section we give the probing techniques and variables fixing schemes for SC.

When probing x_j at 1, not only the lagrangian multipliers of all rows $i \in I_j$ must be set to 0 but also all r_q , $q \in J_i$ for every $i \in I_j$ must be reduced to properly reflect the new importance of the columns with respect to setting $x_j = 1$. Let:

$$\delta_j^1 = \sum_{i \in I_j} u_i \times \left(\left| q \in J_i : r_q \leq 0 \right| \right)$$

The proposed score is embedded in the following recursive fixing-to-zero scheme as in Balas and Carrera (1996):

procedure *Fix-to-Zero*;

input: $N, N_C, LB, UB, \mathbf{u}$

output: N, N_C

begin

for $j \in N_C \setminus B^P$ **do**

if $\lceil LB + r_j + \delta_j^1 \rceil \geq LB$ **then**

$N_C \leftarrow N_C \setminus \{j\}$

```

        N ← N \ {j}
    endif
end for
end

```

To fix a column $j \in B^P$ permanently at 1 compute, for each $i \in I_j$, the amount of modification that needs to be made to the value u_i such that at least one other column $q \in J_i$, $q \neq j$ has a reduced cost that is non-positive and row i will be covered again. This amount of modification required for u_i is

$$\delta_j^0 = \sum_{i \in I_j} \min_{q \in J_i} \{r_q\}$$

and $x_j, j \in B^P$, can be permanently to 1 if $\lceil LB - r_j + \delta_j^0 \rceil \geq UB$.

procedure *Fix-to-One*;

input: $N, N_C, LB, UB, \mathbf{u}$

output: N, N_C

```

begin
    for  $j \in B^P$  do
        if  $\lceil LB - r_j + \delta_j^0 \rceil \geq UB$  then
             $x_j \leftarrow 1$ 
             $N_C \leftarrow N_C \setminus \{j\}$ 
             $N \leftarrow N \setminus \{j\}$ 
        endif
    end for
end

```

7. Pricing and Core Problem Generation

To define core problems, we employ the pricing scheme below that resembles the one presented in Caprara *et al.* (1999).

procedure *Define Core Problem*;

begin

$NC \leftarrow \emptyset$

for $j \in N$ **do**

if $r_j \leq 0.1$ **then** $N_c \leftarrow N_c \cup \{j\}$

end for

for $i \in M$ **do**

if $|J_i| \geq 5$ and $|J_i \cap N_c| < 5$ **then**

add $(5 - |J_i \cap N_c|)$ columns in $J_i \setminus N_c$ with smallest r_j values to N_c

endif

end for

end

As shown, we ensure that each row is covered by at least 5 columns, if possible, in a core problem.

8. Proposed Algorithm

The following summarizes the steps of the proposed metaheuristic algorithm, whose behavior is illustrated in Figure 1. This figure presents the algorithm's steps from the "feasibility" point of view as well as from the "optimality" point of view.

S1. Initialization: definition of a vector \mathbf{u} via (5), definition of a core problem as

presented in Section 7, and definition of the total number of cycles as $K = \left\lceil \frac{N}{N_c} \right\rceil$

S2. Call *Tabu Search Metaheuristic* as illustrated in Section 2. Primal intensive phase, with search both in the feasible and infeasible space. The overall time complexity for this phase is $O(r)$, where $r \ll m$ is the cardinality of the solution found.

S3. Solve (p2d), as presented in Section 5. Find a new dual vector \mathbf{u} , which best describe the current primal solution. This phase takes up $O(p)$ time, where p is the number of rows in (p2d) and is less than $|N_c|$.

S4. Define a new *Core Problem*, as presented in Section 7. The new core problem is defined using the dual vector \mathbf{u} provided as solution of (p2d). This phase is executed in $O(n)$ time.

S5. Lagrangian Optimization Phase, as described in Section 4. Dual intensive phase, with search of a new dual vector \mathbf{u} and a new lower, tighter, lower bound. Each iteration of this phase requires the computation of the Lagrangian vector \mathbf{u} , which is obtained in $O(q)$ time, where q is the number of non-zero entries in the matrix A (coefficient matrix), the computation of the subgradient, done in $O(n \log n + q)$, due to column sorting and other computation, and, finally, the update of the multiplier vector, in $O(m)$ time. Overall, this phase requires $O(m(n \log n + q))$ time.

S5. Probing Techniques Application. Fix variables to zero and to one, using updated dual information. This phase takes up $O(q)$ time.

S6. If the maximum number of iterations has been reached, STOP. Otherwise, go to **S2**.

9. Computational Results on SC Benchmarks

In this section we present the results obtained testing the algorithm on benchmark problems. The algorithm was implemented in GNU C++. The (p2d) problem is solved using Clp COIN-OR (Lougee-Heimer, 2003). The computing platform used is a Linux workstation with AMD 1.1GHz processor and 512 Mb of memory.

Table 1 reports the results on the instances appeared in Wedelin (1995). All the problems are real-world problems from the airline industry. A comparison with the performances of commercial software packages is infeasible, since, at the moment, no results have been provided with respect to these instances. However, the results obtained with the proposed algorithm are compared with the best results available, obtained through *ad-hoc* algorithms developed by researchers and practitioners of the area. As a general rule, “*ad-hoc*” algorithms, such as the one proposed in this paper, offer the advantage that very large instances of the Airline Crew Scheduling problem can be tackled in a short computational time.

The major characteristic of these instances is that they present real costs for each crew assignment, having costs in a range of 1,600 – 2,111,450. Out of 6 instances, for 4 of them the algorithm finds the optimal solution, on the fifth and sixth it finds a solution that is better than any other solution found so far.

Table 1: Results on instances from Wedelin (1995)
Time measured in CPU seconds

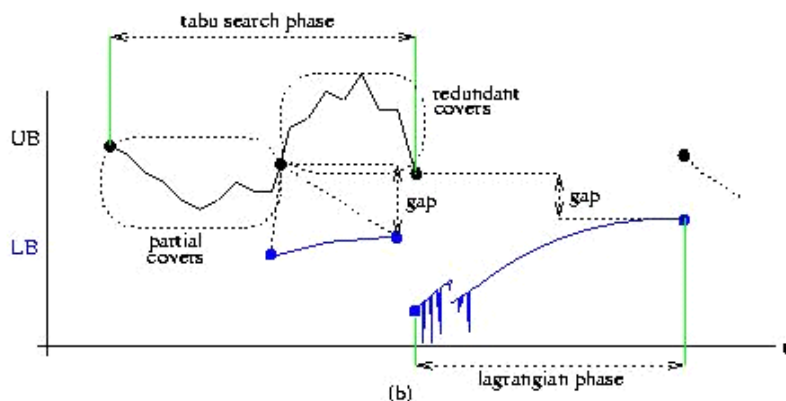
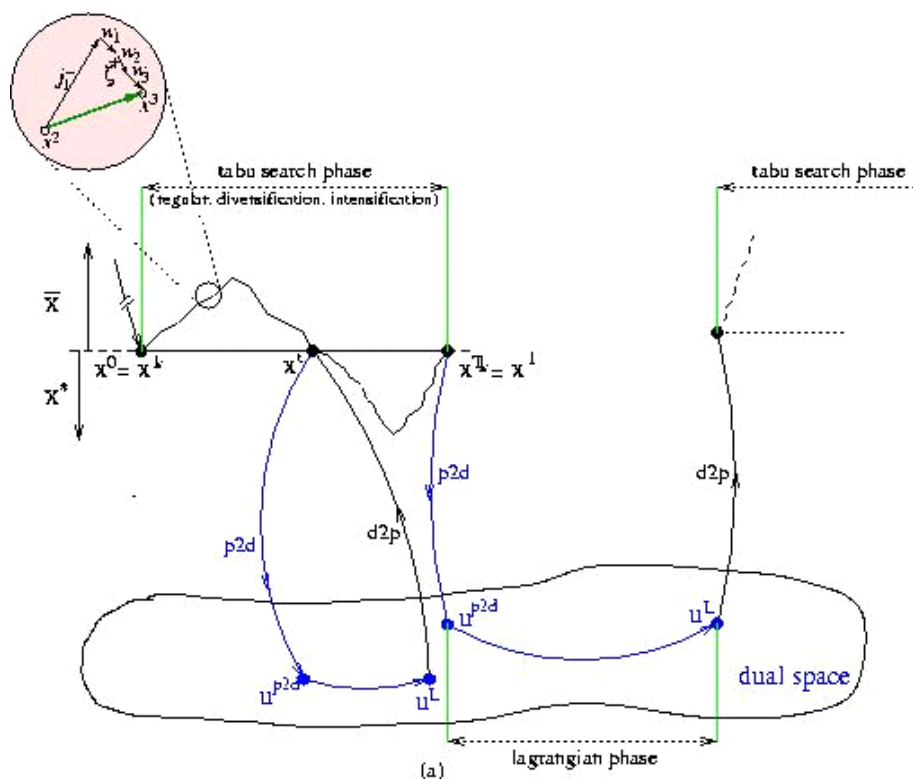
Name	Size	Density	Range	Best		Proposed Algorithm	
				UB	Time	UB	Time
b727scratch	29x157	8.20%	1,600-11,850	94400 ⁺	0.3	94,400	0.5
alitalia	118x1,165	3.10%	2,200-2,110,900	27258300 ⁺	6.2	27,258,300	8.5
a320	199x6,931	2.30%	1,600-2,111,450	12620100 ⁺	79.5	12,620,100	23.5
a320coc	235x18,753	1.90%	1,900-1,812,000	14495500 ⁺	1,023.70	14,495,500	134.6
sasjump	742x10,370	0.60%	4,720-55,849	7339537 ⁺	396.3	7,339,521	221.7
sasd9imp2	1,366x25,032	0.30%	3,860-35,200	5262190 ⁺	1,579.70	5,262,140	1,066.30

10. Conclusions

We have presented a new dynamic scheme for large scale set covering problems. The backbone of the algorithm is a new primal-to-dual mechanism that, given any prime cover, it constructs the dual feasible vector that better reflects the quality of the prime cover. Using this new mechanism, the algorithm constantly updates the status of the search in the dual space any time a new prime cover is found and vice versa, dynamically linking the primal intensive phase with the dual intensive phase.

Owing to the intensive use of primal-based schemes, the algorithm is especially suited for those instances of SC with a number of rows much larger than the number of columns. Some new applications of SC, such as probe selection problem for hybridization experiment as well as attributes identification and patterns selection in logical analysis of data, can be better tackled with a primal intensive approach rather than via the traditional Lagrangian based approach.

Figure 1: algorithm behavior (a) feasibility, and (b) optimality.



References

- Anbil, R., Forrest, J. and Pulleyblank, W. 1998. "Column Generation and the Airline Crew Pairing Problem." *Documenta Mathematica*, Extra Volume ICM, III, 677-786.
- Balas, E., M. C. Carrera. 1996. "A dynamic subgradient-based branch-and-bound procedure for set covering problem." *Operations Research*, 44, 875-890.
- Balas, E., A. Ho. 1980. "Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study." *Mathematical Programming Study*, 112, 37-60.
- Beasley, J. E. 1990a. "A lagrangian heuristic for set covering problem." *Naval Research Logistics*, 37, 151-164.
- Beasley, J. E. 1990b. "OR-Library: distributing test problems by electronic mail." *Journal of Operations Research Society*, 41, 1069-1072.
- Beasley, J. E., P. C. Chu. 1996. "A genetic algorithm for the set covering problem." *European Journal of Operational Research*, 94, 392-404.
- Bricker, D., K. Techapicjetvanich. 1993. "Investigation of lagrangian heuristics for set covering problems." *Technical Report*, Department of Industrial Engineering, The University of Iowa, Iowa City, IA.
- Caprara, A., M. Fischetti, P. Toth. 1999. "A heuristic method for set covering problem." *Operations Research*, 47, 730-743.
- Caserta, M., H. S. Ryoo. "Efficient Tabu Search-based Procedure for Optimal Redundancy Allocation in Complex System Reliability." *Proc. 5th Intl Conference on Optimization: Techniques and Applications*, vol. 2, 592-599, Dec. 01.
- Ceria, S., P. Nobili, A. Sassano. 1998. "A lagrangian based heuristic for large-scale set covering problems." *Mathematical Programming*, Series B 81, 215-228.

- Fisher, M. L., P. Kedia. 1990. "Optimal solution of set covering/partitioning problems using dual heuristics." *Management Science*, 36, 674-688.
- Garey, M. R., D. S. Johnson. 1979. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman.
- Glover, F. 1989. "Tabu Search - Part I." *ORSA Journal on Computing*, 1, 190-206.
- Glover, F. 1990. "Tabu Search - Part II." *ORSA Journal on Computing*, 2, 4-32.
- Glover, F., Laguna, M. 1997. *Tabu Search*. Kluwer Academic Publishers.
- Held, M., R. M. Karp. 1971. "The travelling salesman problem and minimum spanning tree: part ii." *Mathematical Programming*, 1, 6-25.
- Lougee-Heimer, R. 2003. "The Common Optimization INterface for Operations Research." *IBM Journal of Research and Development*, 47(1), 57-66.
- Vaasko, F. J., G. R. Wilson. 1984. "An efficient heuristic for large set covering problem." *Naval Research Logistics*, 31, 163-171.
- Wedelin, D. 1995. "An algorithm for Large Scale 0-1 Integer Programming with Applications to Airline Crew Scheduling." *Annals of Operational Research*, 57, 283-301.