Chapter 9

COMBINING COLUMN GENERATION AND LAGRANGIAN RELAXATION

Dennis Huisman
Raf Jans
Marc Peeters
Albert P.M. Wagelmans

Abstract Although the possibility to combine column generation and Lagrangian relaxation has been known for quite some time, it has only recently been exploited in algorithms. In this paper, we discuss ways of combining these techniques. We focus on solving the LP relaxation of the Dantzig-Wolfe master problem. In a first approach we apply Lagrangian relaxation directly to this extended formulation, i.e. no simplex method is used. In a second one, we use Lagrangian relaxation to generate new columns, that is Lagrangian relaxation is applied to the compact formulation. We will illustrate the ideas behind these algorithms with an application in lot-sizing. To show the wide applicability of these techniques, we also discuss applications in integrated vehicle and crew scheduling, plant location and cutting stock problems.

1. Introduction

In this chapter we consider (mixed) integer programming problems in minimization form. Obviously, lower bounds for such problems can be computed through a straightforward calculation of the LP relaxation. Dantzig-Wolfe decomposition and Lagrangian relaxation are alternative methods for obtaining tighter lower bounds. The key idea of Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) is to reformulate the problem by substituting the original variables with a convex combination of the extreme points and a linear combination of the extreme rays of the polyhedron corresponding to a substructure of the formulation. Throughout the paper, we will assume that this polyhedron is bounded. Therefore, only the extreme points are needed. This substitution results in the master or extended formulation, which contains the linking constraints from the original compact formulation and additional convexity constraints. When solving the LP relaxation of the master problem, column generation is used to deal with the large number of variables. Starting with a restricted master problem which contains only a small subset of all columns, we generate the other columns when they are needed. This is done by solving a so called pricing problem in which one or more variables with negative reduced costs are determined. After each execution of the pricing procedure, we calculate the optimal value of the LP relaxation of the restricted master problem, $\overline{v}_{RDW}$. This provides an upper bound on the optimal value of the Dantzig-Wolfe relaxation, $\overline{v}_{DW}$, which itself is a lower bound for the optimal IP value v_P. When a simplex algorithm is used to solve the restricted master problem, we obtain optimal values of the dual variables corresponding to the linking and convexity constraints. These values are used in the pricing problem to check if we can generate new columns with negative reduced cost. If we find such columns, we add them to the relaxed master problem and reoptimize, otherwise we have found the optimal value of the Dantzig-Wolfe relaxation $\overline{v}_{DW}$. This value will usually be tighter than $\overline{v}_P$, the value of the LP relaxation of the original compact formulation.

In Lagrangian relaxation, the complicating constraints are dualized into the objective function. Given a specific vector of positive multipliers l, the Lagrangian relaxation problem always gives a lower bound, $\overline{v}_{LR}(l)$, on the optimal IP value v_P. The Lagrangian dual problem consists of finding the maximum lower bound: $\overline{v}_{LD} = \max_{l \geq 0} \overline{v}_{LR}(l)$. Typically, the latter problem is solved using an iterative procedure, where in subsequent iterations, the Lagrangian multiplier vector l is updated and we solve a new Lagrangian problem with these updated multipliers. In this chapter we focus on the subgradient method (Fisher, 1985, e.g.) for approximating the optimal multipliers, although more advanced methods such as the bundle method (Lemaréchal, Nemirovskii and Nesterov, 1995, e.g.) or the volume algorithm (Barahona and Anbil, 2000) exist.

There exists a strong relationship between Dantzig-Wolfe decomposition and Lagrangian relaxation. It is well known that when the Lagrangian relaxation is obtained by dualizing exactly those constraints that are the linking constraints in the Dantzig-Wolfe reformulation, the optimal values of the Lagrangian dual, $\overline{v}_{LD}$, and the LP relaxation of the Dantzig-Wolfe reformulation, $\overline{v}_{DW}$, are the same. In fact, one formulation is the dual of the other (Geoffrion, 1974; Fisher, 1981). Furthermore, the optimal dual variables λ for the linking constraints in the master problem correspond to optimal multipliers l for the dualized constraints in the Lagrangian relaxation (Magnanti, Shapiro and Wagner, 1976). Moreover, the subproblem that we need to solve in the column generation procedure is the same as the one we have to solve for the Lagrangian relaxation except for a constant in the objective function. In the column generation procedure, the values for the dual variables are obtained by solving the LP relaxation of the restricted master problem, whereas in the Lagrangian relaxation, the Lagrangian multipliers are updated by subgradient optimization.

Both approaches have advantages and disadvantages. Lagrangian relaxation provides a lower bound on the optimal IP value v_P at each iteration of the subgradient algorithm, but no primal solution is available. In many applications, the dual information is used in a heuristic fashion to obtain a primal solution. On the other hand, column generation directly provides a primal solution at each iteration, which can be used to construct feasible solutions for the MIP in a rounding heuristic. Further, the Lagrangian lower bound can be computed without much difficulty at each step of the column generation process. There are also differences in the computational implementation and convergence behaviour. The subgradient algorithm is usually stopped after a fixed number of iterations, without the guarantee of having found the optimal value $\overline{v}_{LD}$ (Fisher, 1985). However, the subgradient optimization for updating the Lagrangian multipliers is computationally inexpensive and easy to implement. The simplex optimization of the master problem, on the other hand, is computationally expensive and a tailing-off effect, i.e. slow convergence towards the optimum in the final phase of the algorithm, is generally observed (Barnhart et al., 1998; Vanderbeck and Wolsey, 1996). The use of problem specific information can guide the choice of the Lagrangian multipliers and can lead to a faster convergence, whereas we do not have the same freedom in the column generation approach where the master problem provides the values of the dual variables.

In this chapter we will discuss how the relationship between Dantzig-Wolfe decomposition and Lagrangian relaxation can be exploited to develop improved algorithms combining the strengths of both methods. We discuss two ways in which the two techniques can be combined efficiently. To be more specific, Lagrangian relaxation can be applied to the master problem to approximate the optimal values of the dual variables or it can be used on the original compact formulation of the problem to generate good columns. However, notice that we will only discuss column generation within the framework of DW decomposition, but it can also be considered as a general LP pricing technique. For the combination of column generation and Lagrangian relaxation within this framework, we refer to Löbel (1998); Fischetti and Toth (1997). In order to explain the general principles within the framework of DW decomposition in Section 2, we use the example of capacitated lot-sizing. In Sections 3–5, other applications and their specific implementation issues are discussed.

2. Theoretical framework and basic approaches

2.1 Preliminaries

We will illustrate the basic approaches for combining column generation and Lagrangian relaxation using the Capacitated Lot-Sizing Problem (CLSP). In this problem we determine the timing and level of production for several items on a single machine with limited capacity over a discrete and finite horizon. For a more comprehensive description, we refer to Kleindorfer and Newson (1975) or Trigeiro, Thomas and McClain (1989). Let P be the set of products $\{1, \ldots, n\}$ with index i and T the set of time periods $\{1, \ldots, m\}$ with index t. We have the following parameters: d_{it} is the demand of product i in period t; sc_i, vc_i and hc_i are the set up cost, variable production cost and holding cost for product i, respectively; vt_i is the variable production time for product i and cap_t is the capacity in period t. There are three decision variables: x_{it} is the amount of production of product i in period t; s_{it} is the inventory level of product i at the end of period t; $y_{it} = 1$ if there is a set up for product i in period t, $y_{it} = 0$ otherwise. The mathematical formulation of the CLSP is then as follows:

$$\min \sum_{i \in P} \sum_{t \in T} (sc_i y_{it} + vc_i x_{it} + hc_i s_{it}) \tag{9.1}$$

$$\text{subject to } s_{i,t-1} + x_{it} = d_{it} + s_{it} \quad \forall i \in P, \forall t \in T, \tag{9.2}$$

$$x_{it} \leq M y_{it} \quad \forall i \in P, \ \forall t \in T, \tag{9.3}$$

$$\sum_{i \in P} vt_i x_{it} \leq cap_t \quad \forall t \in T, \tag{9.4}$$

$$y_{it} \in \{0, 1\}, \quad x_{it} \geq 0, \quad s_{it} \geq 0, \quad s_{i,0} = 0 \qquad \forall i \in P, \ \forall t \in T. \tag{9.5}$$

The objective function (9.1) minimizes the total costs, consisting of the set up cost, the variable production cost and the inventory holding cost. Constraints (9.2) are the inventory balancing constraints: Inventory left over from the previous period plus current production can be used to satisfy current demand or build up more inventory. Constraints (9.3) are the set up forcing constraints: If there is any positive production in period t, a set up is enforced. In order to make the formulation stronger, the 'big M' is usually set to the minimum of the sum of the remaining demand over the horizon and the total production which is possible with the available capacity. Next, there is a constraint on the available capacity in each period (9.4). Finally, there are the non-negativity and integrality constraints (9.5). We let v_{LS} and $\overline{v}_{LS}$ denote the optimal objective value for problem (9.1)–(9.5) and its LP relaxation, respectively.

Decomposition approaches for this problem hinge on the observation that when we disregard the capacity constraints (9.4), the problem decomposes into an uncapacitated lot-sizing problem for each item i. Let S^i be the set of feasible solution for subproblem i: $S^i = \{(x_{it}, y_{it}, s_{it}) \mid (9.2), (9.3), (9.5)\}$ and $S = \bigcup_{i \in P} S^i$. In the Dantzig-Wolfe decomposition, we keep the capacity constraints in the master problem and add a convexity constraint for each item (Manne, 1958; Dzielinski and Gomory, 1965)). The new columns represent a production plan for a specific item over the full time horizon. Let Q_i be the set of all extreme point production plans for item i; z_{ij} is the new variable representing production plan j for item i; c_{ij} is the total cost of set up, production and inventory for production plan j for item i and r_{ijt} is the capacity usage of the production in period t according to plan j for item i. The LP relaxation of a restricted master problem then looks as follows:

$$\overline{v}_{RDWLS} = \min \sum_{i \in P} \sum_{j \in \widetilde{Q}_i} c_{ij} z_{ij} \tag{9.6}$$

$$\text{subject to} \sum_{i \in P} \sum_{j \in \widetilde{Q}_i} r_{ijt} z_{ij} \le cap_t \quad \forall t \in T, \tag{9.7}$$

$$\sum_{j \in \widetilde{Q}_i} z_{ij} = 1 \quad \forall i \in P, \tag{9.8}$$

$$z_{ij} \ge 0 \quad \forall i \in P,\ \forall j \in \widetilde{Q}_i. \tag{9.9}$$

where $\widetilde{Q}_i$ is a subset of Q_i. Additional columns (variables) are generated when they are needed, using the information of the optimal dual variables λ_t (≤ 0) and π_i of the capacity and convexity constraints, respectively. In the pricing problem, we check for each item i if we can generate a new column by solving the following subproblem:

$$rc_i^*(\lambda, \pi) = \min_{(x,y,s) \in S^i} \sum_{t \in T} (sc_i y_{it} + vc_i x_{it} + hc_i s_{it}) - \sum_{t \in T} vt_i x_{it} \lambda_t - \pi_i. \tag{9.10}$$

If such a column with negative reduced cost is found, we add it to the restricted master problem, reoptimize this problem and perform another pricing iteration; otherwise we have found the optimal Dantzig-Wolfe bound, $\overline{v}_{DWLS}$.

In Lagrangian relaxation, the capacity constraints (9.4) are dualized in the objective function with non-positive multipliers $l = \{l_1, l_2, \ldots, l_m\}$:

$$\overline{v}_{LRLS}(l) = \min_{(x,y,s)\in S} \sum_{i\in P}\sum_{t\in T}(sc_i y_{it} + vc_i x_{it} + hc_i s_{it}) + \sum_{t\in T} l_t \left(cap_t - \sum_{i\in P} vt_i x_{it}\right). \quad (9.11)$$

Note that we use here non-positive Lagrangian multipliers in order to show the similarity with the non-positive dual variables λ. The Lagrangian problem also decomposes into single item uncapacitated lot-sizing problems. For each item i we have the following subproblem:

$$\overline{v}_{LRLS,i}(l) = \min_{(x,y,s)\in S^i} \sum_{t\in T}(sc_i y_{it} + vc_i x_{it} + hc_i s_{it}) - \sum_{t\in T} vt_i x_{it} l_t. \quad (9.12)$$

We see that the subproblem of calculating the minimum reduced cost (9.10) in the Dantzig-Wolfe decomposition and the subproblem in the Lagrangian relaxation (9.12) are identical, except for a constant in the objective function. The solution of the Lagrangian dual problem gives the maximum lower bound $\overline{v}_{LDLS} = \max_{l\le 0} \overline{v}_{LRLS}(l)$. In iterative steps, the multipliers are updated in order to attain this Lagrangian dual bound. Let $x^* = (x^*_{11}, x^*_{12}, \ldots, x^*_{1m}, \ldots, x^*_{n1}, x^*_{n2}, \ldots, x^*_{nm})$ be the optimal production quantities for the Lagrangian problem (9.11) with multipliers l^k at iteration k, then the following standard subgradient update formulas (Fisher, 1981) result in a new vector of multipliers l^{k+1}:

$$l_t^{k+1} = \min\left(0, l_t^k + \beta_k\left(cap_t - \sum_{i\in P} vt_i x^*_{it}\right)\right) \quad t = 1, \ldots, m, \quad (9.13)$$

$$\beta_k = \alpha \frac{\left(ub - \overline{v}_{LRLS}(l^k)\right)}{\sum_{t\in T}(cap_t - \sum_{i\in P} vt_i x^*_{it})^2}. \quad (9.14)$$

Equation (9.14) determines the step-size, where $0 < \alpha \le 2$ and the value ub is an upper bound on v_{LS}.

During column generation, the value of the restricted master problem $\overline{v}_{RDWLS}$ provides an upper bound on the optimal Dantzig-Wolfe relaxation value $\overline{v}_{DWLS}$. However, a lower bound can be easily calculated as well. Let $rc^*_i(\lambda, \pi)$ be the minimum reduced cost for subproblem i with the current optimal dual variables λ and π, then

$$\sum_{i\in P} rc^*_i(\lambda, \pi) + \overline{v}_{RDWLS} \le \overline{v}_{DWLS} \le \overline{v}_{RDWLS}. \quad (9.15)$$

This lower bound is actually equal to the Lagrangian lower bound using the current optimal dual variables λ as multipliers:

$$\begin{aligned}\overline{v}_{LRLS}(\lambda) &= \sum_{i\in P}\overline{v}_{LRLS,i}(\lambda) + \sum_{t\in T}\lambda_t cap_t \\ &= \sum_{i\in P}\overline{v}_{LRLS,i}(\lambda) - \sum_{i\in P}\pi_i + \sum_{i\in P}\pi_i + \sum_{t\in T}\lambda_t cap_t \\ &= \sum_{i\in P} rc_i^*(\lambda,\pi) + \overline{v}_{RDWLS},\end{aligned}$$

where in the final step, equivalence between $\sum_{i\in P}\pi_i + \sum_{t\in T}\lambda_t cap_t$ and $\overline{v}_{RDWLS}$ follows from LP duality. This lower bound was already proposed by Lasdon and Terjung (1971) who used column generation to solve a large production scheduling problem. It has also been discussed for other specific problems such as discrete lot-sizing and scheduling (Jans and Degraeve, 2004), machine scheduling (Van den Akker, Hurkens and Savelsbergh, 2000), vehicle routing (Sol, 1994), a multicommodity network-flow problem (Holmberg and Yuan, 2003) and the cutting stock problem (Vanderbeck, 1999). A general discussion can be found in Wolsey (1998); Martin (1999). Vanderbeck and Wolsey (1996) provide a slight strengthening of this bound. The bound can be used for early termination of the column generation procedure, reducing the tailing-off effect. For IP problems with an integer objective function value, we can also stop if the value of this lower bound rounded up is equal to the value of the restricted master problem rounded up.

2.2 Using Lagrangian relaxation on the extended formulation

Instead of using the simplex algorithm to obtain the optimal dual variables of the (restricted) master problem, one can also use Lagrangian relaxation to approximate these values. Cattrysse et al. (1993); Jans and Degraeve (2004) apply this technique for solving a variant of the capacitated lot-sizing problem. A similar integration of Dantzig-Wolfe decomposition and Lagrangian relaxation is also used for the generalized assignment problem (Cattrysse, Salomon and Van Wassenhove, 1994), and integrated vehicle and crew scheduling which is the topic of Section 3.

In order to approximately solve the LP relaxation of the restricted master problem (9.6)–(9.9), we dualize the capacity constraint (9.7) into the objective function (9.6) with non-positive multipliers l_t:

$$\overline{v}_{LR-RDW}(l) = \min \sum_{i \in P} \sum_{j \in \widetilde{Q}_i} c_{ij} z_{ij} + \sum_{t \in T} l_t \left(cap_t - \sum_{i \in P} \sum_{j \in \widetilde{Q}_i} r_{ijt} z_{ij} \right) \tag{9.16}$$

$$\tag{9.17}$$

$$\text{subject to} \sum_{j \in \widetilde{Q}_i} z_{ij} = 1 \quad \forall i \in P, \tag{9.18}$$

$$z_{ij} \geq 0 \quad \forall i \in P,\ \forall j \in \widetilde{Q}_i. \tag{9.19}$$

The problem decomposes into subproblems per item that are easy to solve, because taking the column with the lowest total cost for each item results in the optimal solution. The optimal Lagrangian multipliers are iteratively approximated via a standard subgradient optimization procedure. At the end of a subgradient phase, the Lagrangian multipliers l_t are an approximation of the optimal dual variables λ_t. Next, the optimal dual variable π_i of the convexity constraint for item i can be approximated by the value p_i as follows:

$$p_i = \min_{j \in \widetilde{Q}_i} \left(c_{ij} - \sum_{t \in T} l_t r_{ijt} \right). \tag{9.20}$$

The Lagrangian multipliers l_t and p_i can be used to generate new columns in the pricing subproblem (9.10). The new columns are added to the restricted master problem and in a subsequent step the optimal dual variables λ and π for the updated restricted master problem are again approximated by Lagrangian relaxation.

Given the Lagrangian multipliers l_t and p_i, we can still compute a lower bound:

$$\sum_{i \in P} rc_i^*(l, p) + \overline{v}_{LR-RDW}(l) \leq \overline{v}_{DWLS}. \tag{9.21}$$

This can again be proven by starting from the Lagrangian relaxation $\overline{v}_{LRLS}(l)$ (9.11), which gives a valid lower bound for any $l \leq 0$:

$$\begin{aligned}
\overline{v}_{LRLS}(l) &= \sum_{i \in P} \overline{v}_{LRLS,i}(l) - \sum_{i \in P} p_i + \sum_{i \in P} p_i + \sum_{t \in T} l_t cap_t \\
&= \sum_{i \in P} rc_i^*(l, p) + \sum_{i \in P} p_i + \sum_{t \in T} l_t cap_t
\end{aligned}$$

$$= \sum_{i \in P} rc_i^*(l,p) + \sum_{i \in P} \min_{j \in \widetilde{Q}_i} \left(c_{ij} - \sum_{t \in T} l_t r_{ijt} \right) + \sum_{t \in T} l_t cap_t$$
$$= \sum_{i \in P} rc_i^*(l,p) + \overline{v}_{LR-RDW}(l).$$

What are the advantages of approximating the optimal dual variables by Lagrangian relaxation instead of computing them exactly with a simplex algorithm? Bixby et al. (1992); Barnhart et al. (1998) note that in case of alternative dual solutions, column generation algorithms seem to work better with dual variables produced by interior point methods than with dual variables computed with simplex algorithms. The latter give a vertex of the face of solutions whereas interior point algorithms give a point in the center of the face, providing a better representation of it. From that perspective, Lagrangian multipliers may also provide a better representation and speed up convergence. Computational experiments from Jans and Degraeve (2004) indicate that using Lagrangian multipliers indeed speeds up convergence and decreases the problem of degeneracy. Lagrangian relaxation has the additional advantage that during the subgradient phase possibly feasible solutions are generated. The subgradient updating is also fast and easy to implement. Finally, this procedure eliminates the need for a commercial LP optimizer.

2.3 Using Lagrangian relaxation on the compact formulation

This approach is based on the observation that when the Lagrangian relaxation is obtained by dualizing exactly those constraints that are the linking constraints in the Dantzig-Wolfe reformulation, the same subproblem results. Consequently, the solutions generated by the Lagrangian subproblems can also be added as new columns to the master problem. This was first proposed by Barahona and Jensen (1998) for a plant location problem and by Degraeve and Peeters (2003) for the cutting stock problem. These applications are discussed in Sections 4 and 5, respectively. It has also been applied successfully to the capacitated lot-sizing problem (Degraeve and Jans, 2003), that is used again to illustrate the technique. The procedure essentially consists of a nested double loop. In the outer loop, optimal dual variables for the restricted master problem (9.6)-(9.9) are obtained by the simplex method. In the inner loop, the Lagrangian subproblem of the compact formulation (9.11) is solved during several iterations, each time with dual variables which are updated with a subgradient optimization procedure. A generic procedure is depicted in Figure 9.1.

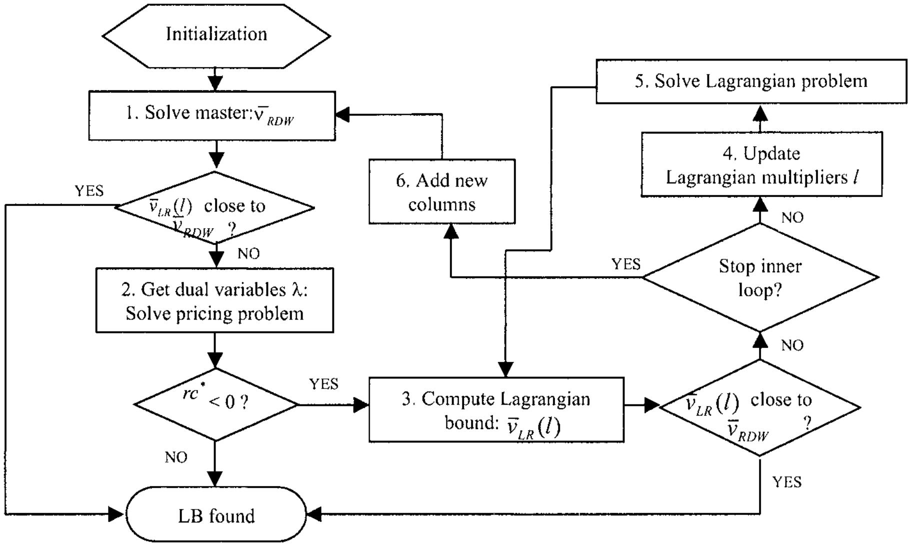

Figure 9.1. Outline of algorithm.

After initialization, the LP relaxation of the restricted master problem (9.6)–(9.9) is solved (Box 1). Next the optimal dual variables λ and π are passed to the pricing problem (9.10), which is then solved to find a new column (Box 2). If the reduced cost is non-negative for each subproblem, then the Dantzig-Wolfe bound $\overline{v}_{DW}$ is found. Otherwise, the inner loop starts (Box 3), where in the first iteration the Lagrangian bound $\overline{v}_{LR}(l)$ (9.11) is computed, using the optimal dual variables of the restricted master problem. This bound is then compared with the objective value of the restricted master problem $\overline{v}_{RDW}$. For a pure integer programming problem with integer coefficients in the objective function, the procedure terminates if both values rounded up are equal, and the Dantzig-Wolfe bound equals $\lceil \overline{v}_{RDW} \rceil = \lceil \overline{v}_{LR}(l) \rceil$. For a mixed integer programming problem, the algorithm may be terminated, if the difference between both values is smaller than a pre-specified percentage. Other stopping criteria could also be checked. For instance, Barahona and Jensen (1998) stop the inner loop after a fixed number of iterations. If no stopping criteria are satisfied, then the Lagrangian multipliers are updated using subgradient optimization (Box 4). The value ub in (9.14) is an upper bound on $\overline{v}_{LD}$, and therefore, ub can be set equal to the LP bound of the last solved restricted master problem $\overline{v}_{RDW}$, since $\overline{v}_{RDW} \geq \overline{v}_{DW} = \overline{v}_{LD}$. Next the algorithm proceeds with solving a new Lagrangian problem, with the updated multipliers (Box 5). The Lagrangian bound is computed again and the inner loop continues, until a stopping criterion is met. Next, we switch back to the outer loop. We add to the restricted master problem the columns with negative reduced costs (obtained in Box 2) and the ones generated in the inner loop if they are not yet present (Box 6).

The main advantage of this procedure is that the LP relaxation of the master problem does not need to be solved each time to get new dual variables necessary for pricing out a new column. Solving the LP relaxation to optimality is computationally much more expensive than performing an iteration of the subgradient optimization procedure. At each subgradient iteration, a new column is found and these columns are expected to be "good" because the Lagrangian multipliers prices converge towards the optimal dual variables of the LP relaxation of the restricted master problem. A second advantage is that we can stop the column generation short of proving LP optimality of the master problem, because the Lagrangian relaxation provides lower bounds on the optimal LP value. Barahona and Jensen (1998) mention this fact as the main motivation for performing a number of subgradient iterations between two consecutive outer loop iterations. This procedure tries to combine the speed of subgradient optimization with the exactness of the Dantzig-Wolfe algorithm. In addition, the procedure provides a primal solution on which branching decisions or rounding heuristics can be based, which is not the case if only subgradient optimization is used. Computational results from Degraeve and Jans (2003) indicate that this method speeds up the column generation procedure. With this hybrid method, it takes about half the time to find the lower bound compared to the traditional method.

3. Application 1: Integrated vehicle and crew scheduling

In this section we discuss the application of a combined column generation/Lagrangian relaxation algorithm to the integrated vehicle and crew scheduling problem. Vehicle and crew scheduling are two of the most important planning problems in a bus company. After a short problem description, we present a formulation for the integrated problem (in case of multiple-depots) to which we apply the approach outlined in Subsection 2.2. Some interesting, recent references on the integrated problem are Freling (1997); Haase, Desaulniers and Desrosiers (2001); Freling, Huisman and Wagelmans (2003) for the single-depot case, and Gaffi and Nonato (1999); Huisman, Freling and Wagelmans (2003) for the multiple-depot case.

3.1 Problem description

The *multiple-depot vehicle and crew scheduling problem* (MD–VCSP) can be defined as follows. Given a set of *trips* within a fixed planning horizon, it minimizes the total sum of vehicle and crew costs such that both the vehicle and the crew schedule are feasible and mutually compatible. Each trip has fixed starting and ending times, and can be assigned to a vehicle and a crew member from a certain set of depots. Furthermore, the travelling times between all pairs of locations are known. A vehicle schedule is feasible if (1) all trips are assigned to exactly one vehicle, and (2) each trip is assigned to a vehicle from a depot that is allowed to drive this trip. From a vehicle schedule it follows which trips have to be performed by the same vehicle and this defines so-called vehicle *blocks*. The blocks are subdivided at *relief points*, defined by location and time, where and when a change of driver may occur and drivers can enjoy their break. A *task* is defined by two consecutive relief points and represents the minimum portion of work that can be assigned to a crew. These tasks have to be assigned to crew members. The tasks that are assigned to the same crew member define a crew *duty*. Together the duties constitute a crew schedule. Such a schedule is feasible if (1) each task is assigned to one duty, and (2) each duty is a sequence of tasks that can be performed by a single crew, both from a physical and a legal point of view. In particular, each duty must satisfy several complicating constraints corresponding to work load regulations for crews. Typical examples of such constraints are maximum working time without a break, minimum break duration, maximum total working time, and maximum duration.

3.2 Mathematical formulation

Let $N = \{1, 2, \ldots, n\}$ be the set of trips, numbered according to increasing starting time. Define D as the set of depots and let s^d and t^d both represent depot d. Furthermore, for the crew we distinguish between two types of tasks, viz., *trip tasks* corresponding to trips, and *dh-tasks* corresponding to deadheading. A *deadhead* is defined as a period that a vehicle is moving in time or space without passengers. E^d is the set of deadheads between two trips i and j.

We define the vehicle scheduling network $G^d = (V^d, A^d)$, which is an acyclic directed network with nodes $V^d = N^d \cup \{s^d, t^d\}$, and arcs $A^d = E^d \cup (s^d \times N^d) \cup (N^d \times t^d)$. Note that N^d is the subset of N that can be serviced by depot d, since it is not necessary that all trips can be served from each depot. Let c_{ij}^d be the vehicle cost of arc $(i, j) \in A^d$.

Furthermore, let K^d denote the set of duties corresponding to depot d and f_k^d denote the crew cost of duty $k \in K^d$, respectively. Moreover, $K^d(i)$ denotes the set of duties covering the trip task corresponding to trip $i \in N^d$, which means that we assume that a trip corresponds to exactly one task. $K^d(i,j)$ denotes the set of duties covering the dh-tasks corresponding to deadhead $(i,j) \in A^d$. Decision variable y_{ij}^d indicates whether an arc (i,j) is used and assigned to depot d or not, while x_k^d indicates whether duty k corresponding to depot d is selected in the solution or not. The multiple-depot vehicle and crew scheduling problem (MD–VCSP) can be formulated as follows.

$$\min \sum_{d \in D} \sum_{(i,j) \in A^d} c_{ij}^d y_{ij}^d + \sum_{d \in D} \sum_{k \in K^d} f_k^d x_k^d \tag{9.22}$$

$$\text{subject to} \sum_{d \in D} \sum_{j:\ (i,j) \in A^d} y_{ij}^d = 1 \quad \forall i \in N, \tag{9.23}$$

$$\sum_{d \in D} \sum_{i:\ (i,j) \in A^d} y_{ij}^d = 1 \quad \forall j \in N, \tag{9.24}$$

$$\sum_{i:\ (i,j) \in A^d} y_{ij}^d - \sum_{i:\ (j,i) \in A^d} y_{ji}^d = 0 \quad \forall d \in D,\ \forall j \in N^d, \tag{9.25}$$

$$\sum_{k \in K^d(i)} x_k^d - \sum_{j:\ (i,j) \in A^d} y_{ij}^d = 0 \quad \forall d \in D,\ \forall i \in N^d, \tag{9.26}$$

$$\sum_{k \in K^d(i,j)} x_k^d - y_{ij}^d = 0 \quad \forall d \in D,\ \forall (i,j) \in A^d, \tag{9.27}$$

$$x_k^d \in \{0,1\} \quad \forall d \in D,\ \forall k \in K^d, \tag{9.28}$$

$$y_{ij}^d \in \{0,1\} \quad \forall d \in D,\ \forall (i,j) \in A^d. \tag{9.29}$$

The objective is to minimize the sum of vehicle and crew costs. The first three sets of constraints, (9.23)–(9.25), correspond to the formulation of the vehicle scheduling problem. Notice that in this formulation constraints (9.24) are redundant. However, it is useful to have these constraints when we relax constraints (9.25), as will be done in the algorithm. Constraints (9.26) assure that each trip task will be covered by a duty from a depot if and only if the corresponding trip is assigned to this depot. Furthermore, constraints (9.27) guarantee the link between vehicles and crews. That is, a vehicle performs deadhead (i,j) if and only if the corresponding dh-task is assigned to a driver from the same depot.

Notice that this formulation is already an extended one. We would obtain a similar formulation, if we would apply Dantzig-Wolfe decomposition on a compact formulation of this problem. Desrosiers et al. (1995) show how such a transformation can be applied on the multicommodity flow problem with resource constraints, which has as special case all kind of vehicle and crew scheduling problems.

3.3 Algorithm

Below we first give a schematic overview of a combined column generation/Lagrangian relaxation algorithm to solve the MD–VCSP. Afterwards, we discuss the steps related to Lagrangian relaxation (1, 2 and 4) in more detail. For details about the other steps, we refer to Huisman, Freling and Wagelmans (2003).

STEP 1 Find an initial feasible solution and take as initial set of columns the duties in that solution.

STEP 2 Solve a Lagrangian dual problem with the current set of columns approximately, i.e. perform some subgradient optimization steps to update the multipliers. This gives a lower bound for the current restricted master problem.

STEP 3 Modify multipliers to prevent that columns are generated twice.

STEP 4 Generate columns (duties) with negative reduced cost and update the set of columns.

STEP 5 Compute an estimate of a lower bound for the (full) master problem. If the gap between this estimate and the lower bound found in Step 2 is small enough (or another termination criterion is satisfied), go to Step 6; otherwise, return to Step 1.

STEP 6 Construct feasible solutions by applying a Lagrangian heuristic.

To approximate the optimal value of the restricted master problem in Step 1, we use the relaxation of model MD–VCSP, where the equality signs in the constraints (9.25)–(9.27) are first replaced by "greater-than-or-equal" signs. These constraints are subsequently relaxed in a Lagrangian way. That is, we associate non-negative Lagrangian multipliers κ_j^d, λ_i^d, μ_{ij}^d with constraints (9.25), (9.26), (9.27), respectively. Then the optimal solution of the remaining Lagrangian subproblem can be obtained by inspection for the x variables and by solving a large *single-depot vehicle scheduling problem* (SDVSP) for the y variables.

The values of the Lagrangian multipliers obtained after applying a subgradient algorithm can be used to generate new columns. However, to assure that all columns in the current restricted master problem have non-negative reduced costs such that the corresponding duties will not be generated again in the pricing problem, we use an additional procedure (Step 3) to update the Lagrangian multipliers after solving the Lagrangian relaxation. This can be done with a greedy heuristic, that modifies these multipliers in such a way that columns in the current restricted master problem $\widetilde{K}^d$ have non-negative reduced costs and the value of the Lagrangian function does not decrease. We denote $\bar{f}_k^d$ as the reduced cost of column $k \in K^d$, which is equal to

$$f_k^d - \sum_{i \in N(k,d)} \lambda_i^d - \sum_{(i,j) \in A(k,d)} \mu_{ij}^d, \tag{9.30}$$

where $N(k,d)$ and $A(k,d)$ are the set of trip tasks and dh-tasks in duty k from depot d, respectively. The heuristic is described below (see also Freling, 1997; Carraresi, Girardi and Nonato, 1995):

for each column $k \in \widetilde{K}^d$ with $\bar{f}_k^d < 0$;

$$\delta := \frac{\bar{f}_k^d}{|N(k,d)| + |A(k,d)|};$$

for each trip task $i \in N(k,d)$: $\lambda_i^d := \lambda_i^d + \delta$;

for each dh-task $(i,j) \in A(k,d)$: $\mu_{ij}^d := \mu_{ij}^d + \delta$;

update the reduced costs for all columns $l \in \widetilde{K}^d$ and $l > k$.

Finally, we will discuss Step 4, where we compute an estimate of a lower bound for the master problem given a lower bound for the current restricted master problem. The latter bound, denoted by $\Phi'(\kappa, \lambda, \mu)$, is obtained in Step 1. Then the expression:

$$\Phi'(\kappa, \lambda, \mu) + \sum_{d \in D} \sum_{k \in K^d \setminus \widetilde{K}^d} \min(\bar{f}_k^d, 0) \tag{9.31}$$

is a lower bound for the (full) master problem for each vector (κ, λ, μ). This can be proven in a similar way as in Subsection 2.2. Therefore, we will skip this proof here.

Notice, however, that we do not calculate this lower bound in each iteration, since for generating new columns it is not necessary to calculate the reduced costs for all of them. Therefore, we estimate this bound in each iteration by taking only into account the reduced costs of the columns that we actually add to the master problem. This estimate can be used to stop the column generation part of the algorithm earlier without exactly obtaining a lower bound.

Table 9.1. Computational results MD–VCSP.

# trips	*80*	*100*	*160*	*200*
# iter.	17.4	25.2	36.8	39.5
cpu m.	154.7	403.9	982.8	1641.5
cpu p.	148.7	510.7	3529.8	4769.5
cpu t.	317.5	942.3	4721.3	6675.0
# found	10	10	4	2
gap (%)	5.37	5.31	5.75	6.52

3.4 Some results

The algorithm presented in the previous subsection has been used to solve several problem instances arising from real-world applications as well as randomly generated instances. In Table 9.1 we summarize some of the results for randomly generated instances with two depots (see Huisman, Freling and Wagelmans, 2003). We report the average number of iterations of the column generation algorithm, and the average computation times for the master problem (cpu m.) and pricing problem (cpu p.), respectively. Furthermore, we give the total average computation time for computing the lower bound (cpu t.). These averages are computed over the instances for which a lower bound is found within 3 hours of cpu time on a Pentium III 450MHz personal computer (128MB RAM). Therefore, we also report the number of instances (out of 10) for which we actually found a lower bound. In the remainder of the table, we report the average gaps between the lower and upper bounds. Notice that all computation times are mentioned in seconds.

In Table 9.1, we only provide results for instances up to 200 trips, since for larger instances we were not able to compute a lower bound within 3 hours computation time. The average gaps between the feasible solutions and the lower bound are about 5% for those instances. However, for large instances we can still use the suggested algorithm to compute feasible solutions by terminating the lower bound phase after a maximum computation time and then continue with Step 5. In practice, this is already quite satisfactory. Therefore, these types of algorithms can be used to solve practical problem instances in an integrated way.

4. Application 2: Plant location

Barahona and Jensen (1998) apply the procedure described in Subsection 2.3 to a plant location problem with minimum inventory. Given a set N of customers, each requiring a set of parts $D_i \subset P, i \in N$, where P denotes the set of all parts, and a set of M possible locations, the objective is to minimize the total costs such that every customer is served, a bound on the total number of warehouses is not exceeded and a service criterion is met. The total costs consist of a fixed costs f_j, for $j \in M$, if a warehouse is opened at location j, a transportation cost c_{ij} if customer i is served from warehouse j, and an inventory cost h_{jk}, if part k is stored in warehouse j. A part must be stored in a warehouse if a customer, requiring that part, is assigned to the warehouse. The service criterion implies that a given percentage of the total demand must be delivered within a certain time limit. Let y_j be 1, if warehouse j is opened, and 0 otherwise, let x_{ij} be 1 if customer i is assigned to warehouse j, and 0 otherwise, and let z_{jk} be 1, if part k must be stored in warehouse j, and 0 otherwise. Then the model can be stated as follows.

$$\min \sum_{j \in M} f_j y_j + \sum_{i \in N} \sum_{j \in M} c_{ij} x_{ij} + \sum_{j \in M} \sum_{k \in P} h_{jk} z_{jk} \tag{9.32}$$

$$\text{subject to } \sum_{j \in M} x_{ij} = 1 \quad \forall i \in N, \tag{9.33}$$

$$\sum_{i \in N} \sum_{j \in M} d_{ij} x_{ij} \geq t, \tag{9.34}$$

$$\sum_{j \in M} y_j \leq L, \tag{9.35}$$

$$x_{ij} \leq y_j \quad \forall i \in N,\ \forall j \in M, \tag{9.36}$$

$$x_{ij} \leq z_{jk} \quad \forall i \in N,\ \forall j \in M,\ \forall k \in D_i, \tag{9.37}$$

$$x_{ij}, y_j, z_{jk} \in \{0,1\} \quad \forall i \in N,\ \forall j \in M,\ \forall k \in P. \tag{9.38}$$

The objective (9.32) is to minimize the total costs, i.e. the sum of fixed costs for opening warehouses, transportation and inventory costs. Constraints (9.33) impose that every customer must be assigned to one location. Constraint (9.34) is the service criterion, i.e. suppose that the company would like that 95% of the demand can be served within two hours, then t equals 95% of the total demand and d_{ij} is equal to the demand of customer i, if the travel time between i and j is less than two hours, and 0 otherwise. Constraint (9.35) implies that at most L locations can be opened. Constraints (9.36) and (9.37) define the relations between the variables, i.e. a customer can only be assigned to a warehouse, if the warehouse is open (9.36), and, if customer i is assigned to a warehouse, then all parts D_i of customer i must be present in the warehouse (9.37).

The Dantzig-Wolfe reformulation consists of implicitly considering every possible assignment of customers to locations. Hence, the objective function and constraints of (the LP relaxation of) the master problem correspond to (9.32)-(9.35) and the original variables are replaced by a convex combination of the extreme points of the polytope defined by (9.36)-(9.38). Barahona and Jensen (1998) show that the pricing problem is equivalent to a minimum cut problem. They observed that the convergence of the Dantzig-Wolfe algorithm is very slow for this problem and that the lower bound obtained by adding the reduced cost of the columns that price out to the value of the current restricted master problem, is very poor in the first iterations of the Dantzig-Wolfe algorithm and improves only slowly. After solving the LP relaxation of the current restricted master problem, they perform a fixed number of subgradient iterations on the original problem to improve the bound, using the master problem's optimal dual variables as starting values for the subgradient procedure. Next, all columns are added to the LP relaxation of the restricted master problem, which is then re-optimized. If the new optimal objective value and the Lagrangian lower bound are close to each other, then a heuristic is applied to obtain an integer solution. They are able to obtain good solutions for problems with about 200 locations, 200 parts and 200 customers within about one hour of computation time on a RS6000-410, using OSL (IBM Corp., 1995) to solve the LPs.

5. Application 3: Cutting stock

Degraeve and Peeters (2000) use a combination of the simplex method and subgradient optimization to speed up the convergence of the column generation algorithm of Gilmore and Gomory (1961) for the one-dimensional cutting stock problem (CSP). This procedure is used to compute the LP relaxation at every node of the branch-and-price tree of the algorithm described in Degraeve and Peeters (2003). The CSP can be defined as follows. Given an unlimited stock of a raw material type of length c and a set of n items with widths $w_1, \ldots, w_n$ and demands $d_1, \ldots, d_n$, cut as few raw material types as possible, such that the demand is satisfied and the total width of the items cut from a raw material type does not exceed its length c. Let P be the set of all feasible cutting patterns, or

$$P = \left\{ p \in \mathbb{Z}_+^n : \sum_{i=1}^{n} w_i p_i \leq c \right\}. \tag{9.39}$$

Let z_p be the number of times pattern p is selected in the solution, then the Gilmore and Gomory formulation can be stated as follows:

$$\min \sum_{p \in P} z_p \tag{9.40}$$

$$\text{subject to} \sum_{p \in P} p_i z_p \geq d_i \quad \forall i \in 1, \ldots, n, \tag{9.41}$$

$$z_p \in \{0, 1, 2, \ldots\} \quad \forall p \in P. \tag{9.42}$$

The objective function (9.40) minimizes the total number of cut raw material, whereas constraints (9.41) are the demand constraints and constraints (9.42) the integrality and non-negativity restrictions. The LP relaxation of (9.40)–(9.42) can be solved by column generation, where the pricing problem is a bounded knapsack problem, if one does not allow that the number of items present in a cutting pattern exceeds the demand, i.e. $p_i \leq d_i$.

Using the procedure described in Subsection 2.3, Degraeve and Peeters (2000) are able to achieve a substantial reduction in required CPU time to solve the LP relaxation of (9.40)–(9.42). Like Barahona and Jensen (1998), they use a limit on the number of subgradient iterations in the inner loop of Figure 9.1, but, in addition, the inner loop is interrupted, if a new column has non-negative reduced cost, or if the Lagrangian bound rounded up equals the master problem's objective value rounded up, as explained earlier in Figure 9.1. If this last condition holds, the Dantzig-Wolfe lower bound is found. Otherwise, all different columns generated in the inner loop are added to the restricted master problem. First it is checked if the value of the best Lagrangian lower bound rounded up is equal to the value of the new restricted master problem rounded up. Then, the algorithm can be terminated, otherwise the next iteration of the outer loop continues.

Table 9.2 presents the results of the computation times for cutting stock instances with 50, 75 and 100 items for 4 different width intervals given in the first row, in which the item widths are uniformly distributed. The demand is uniformly distributed with an average of 50 and the raw material length equals 10000. The experiments were run on a Dell Pentium Pro 200Mhz PC (Dell Dimension XPS Pro 200n) using the Windows95 operating system, the computation times are averages over 20 randomly drawn instances and given in seconds. The LPs are solved using the industrial LINDO optimization library version 5.3 (Schrage, 1995). The columns labelled "DW" present the traditional Dantzig-Wolfe algorithm and the columns labelled "CP" present the results of the combined procedure of Figure 9.1. We observe that the

Table 9.2. Computational results, Cutting Stock Problem.

int	[1,2500]		[1,5000]		[1,7500]		[1,10000]	
n	*DW*	*CP*	*DW*	*CP*	*DW*	*CP*	*DW*	*CP*
50	0.44	0.21	1.47	0.52	0.67	0.46	0.14	0.10
75	1.14	0.47	4.82	1.12	4.26	1.14	0.53	0.27
100	3.19	0.84	15.96	2.05	14.78	3.99	1.65	0.73

reduction in CPU time is higher, when the number of items is higher, and can be as high as a factor 8.

6. Conclusion

We discussed two ways to combine Lagrangian relaxation and column generation. Since this combination has not been used quite often, there are many interesting research questions open. For example, should we use another method to approximate the Lagrangian dual, e.g. a multiplier adjustment method? Furthermore, when implementing such algorithms one has to make decisions with respect to issues such as column management.

In the first method, we used Lagrangian relaxation to solve the extended formulation. Therefore, no simplex method was necessary anymore, which has several advantages. First of all, it decreases the problem of degeneracy and speeds up the convergence. Furthermore, master problems with a larger number of constraints are most often faster solved with Lagrangian relaxation than with a LP solver. We showed this by solving the multiple-depot vehicle and crew scheduling problem.

In the second method, Lagrangian relaxation was used to generate new columns. It is an effective method to speed up convergence of the Dantzig-Wolfe column generation algorithm. The method seems to be quite robust, since it gives good results on three totally different problems, and this without much fine-tuning of the parameters. Several issues can be further investigated. For example, how many subgradient iterations do we allow in the inner loop of Figure 9.1? This is also related to the number of columns that we want to add in an inner loop: All new columns, the ones with negative reduced cost or only the ones with the most negative reduced cost? Adding more columns leads possibly to a faster convergence, but larger restricted master problems are also more difficult to solve. Do we initialize the multipliers in the Lagrangian relaxation part with the best Lagrangian multipliers of the previous step, with the optimal dual variables provided by the simplex algorithm for the current restricted master problem, or some combination? Clearly, there are ample opportunities for research into the effective combination of column generation and Lagrangian relaxation.

References

Barahona, F. and Anbil, R. (2000). The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming Series A*, 87:385–399.

Barahona, F. and Jensen, D. (1998). Plant location with minimum inventory. *Mathematical Programming*, 83:101–112.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., and Vance, P.H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329.

Bixby, R.E., Gregory, J.W., Lustig, I.J., Marsten, R.E., and Shanno, D.F.(1992). Very large-scale linear programming: A case study in combining interior point and simplex methods. *Operations Research*, 40:885–897.

Carraresi, P., Girardi, L., and Nonato, M. (1995). Network models, Lagrangean relaxation and subgradients bundle approach in crew scheduling problems. In: *Computer-Aided Transit Scheduling, Proceedings of the Sixth International Workshop* (J.R. Daduna, I. Branco, and J.M. P. Paixão, eds.), pp. 188–212. Springer Verlag.

Cattrysse, D.G., Salomon, M., Kuik, R., and Van Wassenhove, L.N. (1993). A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. *Management Science*, 39:477–486.

Cattrysse, D.G., Salomon, M., and Van Wassenhove, L.N.(1994). A set partitioning heuristic for the generalized assignment problem. *European Journal of Operational Research*, 72:167–174.

Dantzig, G.B. and Wolfe, P. (1960). Decomposition principle for linear programming. *Operations Research*, 8:101–111.

Degraeve, Z. and Jans, R. (2003). A new Dantzig-Wolfe reformulation and Branch-and-Price algorithm for the capacitated lot sizing problem with set up times. *Technical Report* ERS-2003-010-LIS, ERIM, Erasmus University Rotterdam, the Netherlands.

Degraeve, Z. and Peeters, M. (2000). *Solving the linear programming relaxation of cutting and packing problems: A hybrid simplex method/subgradient optimization procedure.* Technical Report OR0017, Departement Toegepaste Economische Wetenschappen, K.U. Leuven, Belgium.

Degraeve, Z. and Peeters, M. (2003). Optimal integer solutions to industrial cutting-stock problems: Part 2. Benchmark results. *INFORMS Journal on Computing*, 15:58–81.

Desrosiers, J., Dumas, Y., Solomon, M.M., and Soumis, F. (1995). Time Constrained Routing and Scheduling. In: *Network Routing* (M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, eds.), Handbooks in Operations Research and Management Science, Volume 8, pp. 35–139. North-Holland.

Dzielinski, B.P. and Gomory, R.E. (1965). Optimal programming of lot sizes, inventory and labour allocations. *Management Science*, 11:874–890.

Fischetti, M. and Toth, P. (1997). A polyhedral approach to the asymmetric traveling salesman problem. *Management Science*, 43:1520–1536.

Fisher, M.L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science*, 27:1–18.

Fisher, M.L.(1985). An applications oriented guide to Lagrangian relaxation. *Interfaces*, 15:10–21.

Freling, R. (1997). Models and techniques for integrating vehicle and crew scheduling. *Ph.D. Thesis*, Tinbergen Institute, Erasmus University Rotterdam.

Freling, R., Huisman, D., and Wagelmans, A.P.M. (2003). Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6:63–85.

Gaffi, A. and Nonato, M. (1999). An integrated approach to extra-urban crew and vehicle scheduling. In: *Computer-Aided Transit Scheduling* (N.H.M. Wilson, ed.), pp. 103–128. Springer Verlag.

Geoffrion, A.M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114.

Gilmore, P.C. and Gomory, R.E. (1961). A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859.

Haase, K., Desaulniers, G., and Desrosiers, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science*, 35:286–303.

Holmberg, K. and Yuan, D. (2003). A multicommodity network flow problem with side constraints on paths solved by column generation. *INFORMS Journal on Computing*, 15:42–57.

Huisman, D., Freling, R., and Wagelmans, A.P.M. (2003). *Multiple-depot integrated vehicle and crew scheduling.* Technical Report EI2003-02, Econometric Institute, Erasmus University Rotterdam, the Netherlands. Forthcoming in: *Transportation Science.*

IBM Corp. Optimization subroutine library: Guide and references. (1995).

Jans, R. and Degraeve, Z. (2004). An industrial extension of the discrete lot sizing and scheduling problem. *IIE Transactions*, 36:47–58.

Kleindorfer, P.R. and Newson, E.F.P. (1975). A lower bounding structure for lot-size scheduling problems. *Operations Research*, 23:299–311.

Lasdon, L.S. and Terjung, R.C.(1971). An efficient algorithm for multi-item scheduling. *Operations Research*, 19:946–969.

Lemaréchal, C., Nemirovskii, A.S., and Nesterov, Y.E. (1995). New variants of bundle methods. *Mathematical Programming*, 69:111–148.

Löbel, A. (1998). Vehicle scheduling in public transit and Lagrangian pricing. *Management Science*, 44:1637–1649.

Magnanti, T.L., Shapiro, J.F., and Wagner, M.H. (1976). Generalized linear programming solves the dual. *Management Science*, 22:1195–1203.

Manne, A.S. (1958). Programming of economic lot sizes. *Management Science*, 4:115–135.

Martin, R.K. (1999). *Large Scale Linear and Integer Optimization: A Unified Approach.* Kluwer Academic Publishers.

Schrage, L. (1995). LINDO: Optimization Software for Linear Programming. Lindo Systems Inc., Chicago IL.

Sol, M. (1994). *Column generation techniques for pickup and delivery problems.* Ph.D Thesis, Eindhoven University of Technology.

Trigeiro, W., Thomas, L.J., and McClain, J.O. (1989). Capacitated lot sizing with set-up times. *Management Science*, 35:353–366.

Van den Akker, J.M., Hurkens, C.A.J., and Savelsbergh, M.W.P. (2000). Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing*, 12:111–124.

Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming Series A*, 86:565–594.

Vanderbeck, F. and Wolsey, L.A. (1996). An exact algorithm for IP column generation. *Operations Research Letters*, 19:151–159.

Wolsey, L.A. (1998). *Integer Programming*. Wiley.