14 ACCELERATING STRATEGIES IN COLUMN GENERATION METHODS FOR VEHICLE ROUTING AND CREW SCHEDULING PROBLEMS

Guy Desaulniers[1], Jacques Desrosiers[2], and Marius M. Solomon[3]

[1]École Polytechnique and GERAD
C.P. 6079, Succ. Centre-Ville
Montréal, Québec, Canada H3C 3A7
guyd@crt.umontreal.ca

[2]École des Hautes Études Commerciales and GERAD
3000, chemin de la Côte-Sainte-Catherine
Montréal, Québec, Canada H3T 2A7
Jacques.Desrosiers@hec.ca

[3]Northeastern University and GERAD
314 Hayden Hall Boston
MA 02115, USA
Solomon@cba.neu.edu

Abstract: This paper focuses on accelerating strategies used in conjunction with column generation to solve vehicle routing and crew scheduling problems. We describe techniques directed at speeding up each of the five phases of the solution process: pre-processor, subproblem, master problem, branch-and-bound, and post-optimizer. In practical applications, these methods often were key elements for the viability of this optimization approach. The research cited here shows their use led to computational gains, and notably to solutions that could not have been obtained otherwise due to practical problem complexity and size. In particular, we present recent methods directed at the integer programming aspect of the approach that were instrumental in substantially reducing the integrality gap found in certain applications, thereby helping to efficiently produce excellent quality solutions.

C. C. Ribeiro et al., *Essays and Surveys in Metaheuristics*

14.1 INTRODUCTION

Vehicle routing problems consist of finding least cost or maximum profit feasible *routes* for a set of *vehicles* such that all requirements imposed by a given set of tasks are met. The definition of crew scheduling problems is similar and is obtained by replacing *routes* and *vehicles* with *schedules* and *crews*, respectively. Vehicle routes are generally subject to simple feasibility rules such as vehicle capacity and task precedence, while crew schedules must often comply with a variety of complex government regulations and collective agreement rules that increase the difficulty of these latter problems. Given the excellent results achieved in separately solving Vehicle Routing and Crew Scheduling (VRCS) problems, combined VRCS problems have recently been addressed in the literature. These problems, which involve routing vehicles and scheduling crews simultaneously, introduce new challenges mostly due to the need for synchronizing vehicles and crews.

VRCS problems, either considered independently or simultaneously, are important but complex practical problems. Their importance results from the need to manage vehicle expenses and crew salaries which can reach hundreds of millions of dollars annually. The main difficulty of VRCS problems comes from their size as they give rise to very large-scale mixed integer programming problems. Additionally, the modeling aspects must deal with the complexity of various work rules. Over the past ten years, column generation embedded in a branch-and-bound search tree has been used with great success to find optimal or high quality approximate solutions for this type of problems.

VRCS problems can be formulated as mixed integer programs with decomposable structures. Column generation (or equivalently, Dantzig-Wolfe decomposition) applied to these problems is considered a primal method; Lagrangian relaxation (see Geoffrion [25]) used on the same decomposable structures is seen as an equivalent dual method. Column generation was first suggested by Ford and Fulkerson [22] for a maximal multicommodity network flow problem. It was then presented by Dantzig and Wolfe [7] for linear programs with decomposable structures and by Gilmore and Gomory [26] for the cutting-stock problem.

Column generation involves a master problem (or a coordinator) and a subproblem which comprises one or several column generators. It is a two-level method which allows for an intensive use of accelerating strategies at both the coordinator and the generator levels. Furthermore, schemes directed at boosting efficiency can be used as well during the branch-and-bound phase to get integer solutions, or in suitable pre-processing and post-optimizing phases. In fact, to obtain the most of such a combined branch-and-bound and column generation approach (also known as branch-and-price), one must exploit to a maximum all problem structures at hand through the use of such techniques.

This paper presents several accelerating strategies that have been used in conjunction with column generation to solve VRCS problems. Most of them are components of the GENCOL optimizer developed at GERAD, an Operations Research center in Montréal, which has been used successfully to solve problems in: vehicle routing with time windows (Desrosiers, Soumis, and Desrochers [18]; Desrochers, Desrosiers, and Solomon [15]; Kohl et al. [31]), pickup and delivery with time windows (Dumas et al. [20]), as well as dial-a-ride for physically disabled persons (Desrosiers, Dumas, and Soumis [17]), urban transit crew scheduling (Desrochers and Soumis [14]; Rousseau and Desrosiers [37]), multiple depot vehicle scheduling (Ribeiro and Soumis [36]; Desaulniers, Lavigne, and Soumis [11]), aircraft routing (Desaulniers et al. [8]), air crew pairing (Desaulniers et al. [10]), air crew rostering for pilots (Gamache and Soumis [23]) and flight attendants (Gamache et al. [24]), and locomotive and car assignment (Ziarati et al. [39, 40]; Cordeau et al. [5]; Lingaya et al. [34]). Recently, Barnhart et al. [3] offered their perspective on computational issues that need to be addressed when implementing column generation methods for huge integer programs.

Often, these speed up techniques were key elements for the viability of the column generation approach. Without them, it would have been almost impossible to obtain quality solutions to various applications, in a reasonable amount of computer time. By using efficient internal approximations and early termination, column generation approaches can also be turned into optimization based heuristics and used for comparison with other approximate methods for a given class of VRCS problems.

We will present accelerating strategies for each of the five phases of the solution process: pre-processor, subproblem, master problem, branch-and-bound, and post-optimizer, while noting that some involve interactions between several phases. To put things in context, the next section provides a unified solution framework for column generation approaches applied to VRCS problems. Then, the subsequent sections describe various methods that have been used in the literature to substantially accelerate each of the above stages of the solution process.

14.2 A SOLUTION FRAMEWORK FOR VRCS PROBLEMS

Essentially all VRCS problems can be formulated as mixed integer multicommodity network flow problems with additional linking constraints as well as resource variables (see Desaulniers et al. [9]). Without loss of generality, we assume throughout this paper that the objective function consists of minimizing a given cost function.

Commodities represent vehicle or crew types. Each commodity can be associated with a single vehicle or crew member. The flow structure of each commodity is defined as an elementary path on a time-space network, where arcs represent various activities while nodes are locations at specific times. Linking constraints are imposed on all tasks to perform: customers and dial-a-ride requests to satisfy; flight, train, and bus legs to cover; pairings to assign; and so on. They also limit, for example, the number of available vehicles and crews of each type at all depots and bases. Resource variables are used to model local restrictions on a single route or schedule. They are accumulated along the arcs of a path, and updated and verified at nodes. A resource represents a constraint-specific unit of measurement, such as the elapsed time since vehicle departure from the depot location, or the cumulative work time of a crew member since the last periodic break.

In an application such as the Vehicle Routing Problem with Time Windows (VRPTW), the nodes are the locations of the customers to service exactly once, while the arcs represent the paths between these customers. Resources are associated with the visiting times and the vehicle loads at the customer locations. For the air crew pairing problem, the nodes are the airports, and the arcs can represent single flights, connections and ground waitings, among other things. Arcs can also correspond to multiple activities such as duties defined as full workdays. Tasks are the flight legs to cover exactly once, while resources are used to model most of the collective agreement restrictions. In some environments such as the locomotive assignment problem and the air crew rostering problem for flight attendants, tasks (e.g., trains and pairings, respectively) must be covered more than once.

By using an extension of the Dantzig and Wolfe [7] decomposition approach, we define a master problem - which involves all the linking constraints of the mixed integer multicommodity network flow problem treated, and a subproblem - which keeps all the local restrictions on the feasibility of each path (see Desaulniers et al. [9]). The master problem becomes a generalized set partitioning problem where each variable or column is a feasible path on the corresponding commodity network. The subproblem is a collection of resource constrained shortest path problems, one for each commodity.

An integer solution to the multicommodity flow problem is obtained using branching and cutting decisions. At each node of the search tree, the classical column generation approach proceeds as follows to derive a lower bound:

- The simplex method is used to solve the linear relaxation of the generalized set partitioning problem for the current set of columns, in the process calculating dual variables to be used for pricing out new columns.

- The subproblem is used to check whether a new column with a negative marginal cost exists. If so, it is added to the current linear programming problem, which is then re-optimized; if not, the solution to the linear relaxation of the master problem is optimal and provides a lower bound for the current node.

Because VRCS problems are complex, a number of enhancements are necessary to help the standard column generation approach accommodate very large-scale problems. They comprise heuristic methods used to find approximate solutions at the various phases of the solution process. These may also be considered control strategies designed to reduce solution time by exploiting distinctive features of the application on hand. These techniques make decisions regarding the distribution of the effort between the algorithm's components.

14.3 PRE-PROCESSOR STRATEGIES

Heuristics designed for a pre-processor are utilized at the start of the solution process, and if necessary, at every node of the branch-and-bound search tree.

Resource window reduction. Methods that reduce the time windows or, more generally, the resource windows, speed up the constrained shortest path algorithms that are used each time the column generators are called. Indeed, they reduce the resource state space. This technique was originally introduced by Desrochers, Desrosiers, and Solomon [15] and widely applied thereafter.

Heuristic arc elimination. Eliminating undesirable arcs or those unlikely to be used reduces the network sizes. This is common practice for the huge networks arising in air transportation as described, for example, by Desaulniers et al. [10] for air crew pairing applications.

Permanent arc elimination. As shown by Mingozzi et al. [35], arcs can be permanently eliminated from the subproblem networks when a given criterion is met. To be used, this criterion requires a primal integer solution as well as a feasible dual solution to the linear relaxation of the problem.

Initial primal and dual solutions. An initial primal solution provides an upper bound on the optimal value that can be used to prune the branch-and-bound search tree. Moreover, such a solution is helpful to derive an initial dual solution which in turn is used in the pricing out phase. Better dual values can reduce the number of iterations between the coordinator and the column generators. Indeed, the stopping criterion of the column generation process is based on the column marginal costs, and therefore it is directly related to the values of the dual variables. This justifies the need to use efficient mechanisms in order to obtain good approximate dual variable values rapidly. This idea which only recently received attention can be exemplified as follows. For the VRPTW, the insertion heuristic of Solomon [38] could provide a fast initial primal solution, while the insertion costs generate a readily available dual solution.

Task aggregation. Another way to reduce the size of the networks, and subsequently, the row- and column-sizes of the master problem is by combining tasks. This has been used by Ioachim et al. [30] for the dial-a-ride problem, by means of mini-clusters, that is, groups of requests arising from persons willing to travel in the same direction at about the same time. Another example can be found in Hane et al. [28] where they addressed fleet assignment problems occurring in hub-and-spoke networks. In such a network structure, it is easy to identify pairs of flights incident to a spoke that have to be flown by the same aircraft.

Task sequencing. In the same dial-a-ride application for physically disabled persons, *a priori* sequencing of requests at the same location decreases the combinatorial aspect of the constrained shortest path algorithm.

Problem partitioning. Based on different criteria such as time or space, very large-scale problems can be divided into smaller ones, and if possible, recombined at the end of the solution process. Examples of such partitioning strategies can be found in Desrosiers, Dumas, and Soumis [17] for a dial-a-ride application and in Gamache et al. [24] for an airline crew rostering application.

Rolling horizon. This is a time oriented problem partitioning. Given a division of the problem into possibly overlapping time slices, the first slice is solved and the earlier half (or a portion) of its solution is frozen. The next slice is then solved using the results of the preceding frozen solution as initial conditions on vehicles and/or crews. This classical technique continues to be used frequently as exemplified by the work of Ziarati et al. [40] on rail transport.

14.4 SUBPROBLEM STRATEGIES

Dynamic programming procedures for solving resource constrained shortest path problems are pseudo-polynomial algorithms that allow for multiple visits at a node (see Desrosiers et al. [16]). These provide heuristic solutions to the elementary shortest path versions which are $\mathcal{NP}$−hard problems in the strong sense (Dror [19]). Such an algorithm associates with each partial path a label L, i.e., an $(r+1)$-vector composed of the r resources and the marginal cost component of the state space. Given two labels L_1 and L_2 at the same node, the first eliminates the second, if $L_1 \leq L_2$. This is the dominance process, that is, states are compared and only Pareto optimal ones are kept. Below are some of the procedures that can be used to accelerate these algorithms.

Restricted networks. The solution process can use restricted networks to obtain a good initial set of columns. One can *a priori* select just a subset of the best arcs or dynamically ignore some arcs and nodes according to the dual information. Similar to the scaling algorithms used in network theory (see Ahuja et al. [1]), the discarded arcs can be reintroduced subsequently. Examples of such strategies can be found in Barnhart et al. [2], Dumas et al. [20], and Ioachim et al. [30], to name a few. If some nodes and arcs cannot be part of any negative marginal cost path, these can be eliminated from the current computations. For the VRPTW for example, a threshold value on the dual variable at a customer node can be computed *a priori* as the minimum insertion cost of that customer within a route. Discarding a customer node also eliminates all its incident arcs.

State space reduction. In this special case of network restriction, time resources for example, can be temporarily computed to the nearest ten minutes instead of the nearest minute (see Gamache et al. [24]). The range of the new units is smaller compared to the initial ones which means that different levels of consumption are now represented by the same value. This increases the dominance between labels and reduces the solution time. Additionally, dominance can be increased by making use of some tolerance parameters.

Node treatment. Given that in several applications such as the classical VRPTW, the networks are time oriented but may contain cycles, it is useful to consider the nodes in increasing order of a time resource. (This heuristic treatment of the nodes has later conducted to the development of the more efficient concept of label order rather than node order; see Desrosiers et al. [16]).

Pulling versus reaching. While algorithmic complexity is the same for these two strategies, from an implementation point of view it is more efficient to pull to a node all the labels from predecessor nodes so that the dominance process is only applied once at this node while considering all its labels (see Desrochers and Soumis [13]).

Marginal cost interval reduction. As proposed in Crainic and Rousseau [6] for the airline crew scheduling problem, interval reduction can be applied to the marginal cost component since column generators are only looking for negative marginal cost paths.

Maximum size of the label list. In order to control the combinatorial aspect of the shortest path dynamic programming algorithm, a maximum number of labels can be retained at each node of the networks.

Partial pricing of the commodity networks. Since the simplex method does not require the selection of the most negative marginal cost variable, it is often beneficial to consider only some of the commodity networks each time the pricing out phase is called. The rationale for this is that all column generators use the same dual variables at a given iteration, hence the generators tend to produce columns covering the same tasks. Considering the results of the previous iterations, non promising commodity networks can temporarily be discarded. The reduction in the solution time per iteration may easily compensate for the increase in the total number of iterations required. This is particularly important when the number of commodities is rather large, as is the case for rostering problems where it equals the number of employees (typically, several hundreds). Gamache et al. [24] used this approach successfully to produce flight attendant rosters.

Multiple paths. A well known strategy to accelerate column generation is to return to the coordinator many negative marginal cost columns at each iteration. This is even more beneficial if these replicate the structure of an optimal integer solution. In many VRCS problems for example, solutions are made of task-disjoint columns. This can be achieved by using greedy strategies to select the columns. In the context of the analytic center cutting plane method, Goffin and Vial [27] have recently shown that the performance of this decomposition method is mathematically related to the variance-covariance matrix of the selected columns: the performance increases with the selection of non-correlated columns.

Subproblem aggregation. A very popular approach when the number of subproblems is large consists of aggregating identical subproblems (see Desaulniers, Lavigne, and Soumis [11]). This diminishes the chances of generating the same columns and reduces the number of subproblems to solve at each iteration.

Subtour elimination. As mentioned early in this section, the elementary path condition has been relaxed so that cycles can occur within a path. Small subtours of the form (i, j, i), called *2-cycles*, can however easily be eliminated. An early implementation of this method for the VRPTW can be found in Kolen, Rinnooy Kan, and Trienekens [33].

Path composition. In certain cases, it is known that optimal or near optimal solutions comprise columns with only a very small number of tasks, say 2 to 5, with a large proportion of columns involving just 2 or 3. In this case, it is preferable to sequentially generate columns with 2 or 3 tasks, and generate new ones involving up to 5 tasks only toward the end of the solution process. This can be generalized to other column structures. In the air crew pairing problem for example, it might be interesting to sequentially generate pairings with an increasing number of working days (see Crainic and Rousseau [6]).

14.5 MASTER PROBLEM STRATEGIES

As previously stated, the simplex method is used to solve the linear relaxation of the generalized set partitioning problem for the current columns. The primary role of the master problem is to provide the dual variables used for pricing out new columns. Its secondary role is to act as a lower bounding procedure for the branch-and-bound tree. Accelerating techniques for the master phase are as follows.

Column elimination. When the number of columns in the master problem becomes too large, a subset of the nonbasic columns can be removed according to their current marginal cost or other criteria (for instance, columns that have not entered the master problem's basis in the last n column generation iterations, where n is a predefined parameter). To ensure proper convergence of the column generation process, this procedure should not be performed too often and a certain number of nonbasic columns should remain in the master problem after column removal.

Dynamic constraints. When the master problem contains a large number of constraints, it may be possible to relax some of them and reintroduce them when needed (that is, when the solution of a linear relaxation violates them). This is widely used for non convex constraints such as the integrality requirements. Such a strategy is also advantageous when the relaxed constraints have a low probability of being violated. Furthermore, this approach can be used as a heuristic to handle certain nonlinear constraints. For instance, in a railcar assignment application (Lingaya et al. [34]), nonlinear constraints of the form $\prod_i F_i(X) = 0$, where X is the vector of decision variables and the functions F_i are linear, were relaxed and evaluated after the solution of each linear relaxation. When such a constraint was violated, a heuristic branching decision setting function $F_i(X) = 0$ for a selected i was imposed to ensure its satisfaction.

Covering versus partitioning. In many VRCS problems, allowing over-covering of tasks that need to be covered exactly, that is changing the task covering constraints sign from "equal to" to "greater than or equal to" (e.g., for the VRPTW, replacing $= 1$ by ≥ 1, for the number of times a customer is visited) has no impact on the optimal solution. This is true if each additional column covering a subset of the tasks already covered by another column is feasible and less costly than the original column (see Desrochers, Desrosiers, and Solomon [15]). Computationally, the linear relaxation of a set covering problem is easier to solve than that of a set partitioning problem. Hence, over-covering can be used as a heuristic method to rapidly compute good estimates of the dual variables.

Perturbation. Surplus variables can be used to replicate over-covering of the tasks to perform. Additionally, slack variables can be of interest for under-covering. Limiting these slack and surplus variables to small intervals gives a perturbation strategy of the right-hand side of the task-covering constraints. Moreover, the solution of the perturbed problem provides a valid lower bound for the branch-and-bound process. This and the prior technique were illustrated by Desaulniers et al. [12] for a crew pairing application.

Stabilization. This is the next step to the above perturbation strategy. Given *a priori* interval estimates of the dual values, this information can be easily incorporated into the master problem structure by assigning these lower and upper bounds as the cost coefficients of the slack and surplus variables (du Merle et al. [21]). Restricting the dual variables is equivalent to relaxing the primal problem so that, again, the stabilized problem also provides a valid lower bound for the branch-and-bound process. It can be observed here that the set covering problem, as compared to the set partitioning problem, halves the dual space and therefore produces a powerful restriction of it.

Multipliers adjustment. Any heuristic for the adjustment of the dual variables can be useful. One of these is Lagrangian relaxation applied to the master problem. For instance, in Kohl and Madsen [32], the optimal multipliers are found using a method exploiting the benefits of sugradient and bundle methods.

Competing algorithms. When the master problem is highly degenerated, it is often better to use the dual simplex algorithm rather than the primal simplex algorithm. Other algorithms, such as barrier and interior point methods, may also be available to solve the master problem. Therefore, a heuristic can be used at each iteration of the column generation process to select which algorithm should be applied to solve the current master problem. This heuristic can even trigger a change of algorithms if the algorithm first chosen does not perform as anticipated.

14.6 BRANCH-AND-BOUND STRATEGIES

This section examines methods developed for various components of a branch-and-bound approach: a bounding procedure to evaluate the potential of each node in the search tree, a way to explore the search tree, a set of rules to fathom unwanted branch-and-bound nodes, and a branching procedure to separate the feasible region of the unfathomed nodes. Each of these can benefit from the use of accelerating techniques such as the ones proposed below. Moreover, the next strategy proved fundamental in obtaining integer solutions.

Dynamic column generation. In current practice, columns are generated at the root node of the tree, and then used throughout. Yet, as illustrated by the results obtained by Hoffman and Padberg [29] on airline crew pairing problems, and Caprara et al. [4] on rail crew pairing problems, finding quality integer solutions from this static set of columns is often difficult. This is because this *a priori* set is not large enough to force the closure of the integrality gap below 1%. Dynamic column generation at each node of the search tree introduces new columns that are instrumental in completing a partial integer solution.

Depth first search. It is well known that depth first exploration, when combined to an appropriate branching strategy, usually provides a good first integer solution. The upper bound derived from this solution can then be used to fathom several of the unexplored nodes. When solution time is of prime importance, the depth first search is often performed without backtracking.

Search stopping rule. Once a good solution has been found, a maximum amount of additional computer time to use and/or a maximum number of additional branch-and-bound nodes to explore can be imposed to limit overall solution time.

Early branching. In order to avoid the well-known tailing off behavior of column generation, a heuristic rule can be devised to prematurely stop the linear relaxation solution process, for example, when the value of the objective function does not improve sufficiently in a given number of master problem iterations. In this case, the approximate linear relaxation solution does not necessarily provide a lower bound. However, since such a bound is not needed when using a depth first strategy without backtracking, branching decisions can be imposed to define a new restricted master problem to solve. In this case, the value of the objective function can still decrease at a son node (see Gamache et al. [24]).

Note that, using the appropriate Lagrangian function, a lower bound can still be computed when the column generation process is stopped prematurely (see Desaulniers et al. [9]). However, that bound may not be as good as the one provided by the master problem linear relaxation optimal value which could be obtained by completing the column generation process. This tradeoff between bound quality and computational time also occurs when applying Lagrangian relaxation since it is very difficult in practice to find a null subgradient.

Heuristic cutting plane separation algorithm. To obtain a tighter lower bound at each node of the search tree, it might be possible to add violated cutting planes to the corresponding linear relaxation. Since the identification of violated cuts can be quite expensive, it might be preferable to use a heuristic separation algorithm that will speed up the search process but will not guarantee the identification of such a cut if one exists. For instance, Kohl et al. [31] have used a greedy separation algorithm to identify the violated 2-path cuts for the VRPTW.

Fathoming. Fathoming branch-and-bound nodes can be done using tolerance parameters. For instance, a node can be fathomed if its lower bound is greater than the cost of the best integer solution found so far minus a given tolerance value.

Rounding to 1. Given a fractional solution to the linear relaxation of the generalized set partitioning problem, decisions are imposed by fixing at 1 the corresponding selected variables, and removing incompatible columns accordingly. This is generally done for variables close to 1. However, in an air rostering application (see Gamache and Soumis [23]), it was valuable to fix at 1 all variables with a value greater than or equal to 0.6. In this environment, the right-hand side of the task-covering constraints is usually greater than 1 which provides more flexibility and reduces the risk of making a poor decision. Note that, when time permits, it is also possible to fix at 1 selected arc flow variables or combinations of them. These less aggressive decisions can often be handled at the subproblem level.

Rounding to the next integer. A similar strategy can be used when aggregated subproblems are used and the right-hand sides of the task-covering constraints are greater than 1. In this case which may arise for instance in a locomotive assignment problem (see Ziarati et al. [40]), selected master problem variables can be rounded to the next integer.

Multiple decisions. As seen above, many decisions can be taken simultaneously to reduce the size of the search tree in a depth first approach (see Gamache et al. [24], and Ziarati et al. [40]). These decisions can be of various types and should be chosen to be complementary in order to avoid undesirable infeasibility.

Competing branching strategies. When multiple branching strategies are available to derive integer solutions, they can be ordered a priori and, at any branching node, the strategy selected is the first applicable at that node when the order is followed. Additional flexibility can be achieved by using a heuristic scoring system during the solution process that will order the different strategies at each node of the search tree. Applications with multiple branching strategies can be found in Ziarati et al. [40] and Cordeau et al. [5].

14.7 POST-OPTIMIZER STRATEGIES

In this last phase, given an integer solution to a problem, it is sometimes still possible to improve it in several ways. Here are three heuristics that can be used:

Local reoptimizations. Well known improvement procedures such as the 2- and 3-opt interchange heuristics as well as metaheuristics such as tabu search, variable neighborhood search and simulated annealing can be applied in an attempt to improve the current integer solution.

Partial reoptimization. As previously discussed, very large-scale problems are often solved using partitioning and rolling horizon strategies. By successively re-considering some new portions of a problem, one can improve the value of the objective function.

Local relaxations. This has been used specifically for the dial-a-ride application. Bus itineraries were obtained using hard time windows at the pickup and delivery locations. By using local reoptimizations with soft time windows on certain itineraries, i.e., relaxing the hard time window restrictions, it was possible to eliminate several route patterns disliked by the schedulers. It should be mentioned that in this kind of application, one of the main goals is to reduce the number of complaints from both the schedulers and the clients.

14.8 COMPUTATIONAL BENEFITS

The literature cited here highlights the importance of using acceleration methods. Their use has not only resulted in efficiency gains, but more importantly has produced solutions that would have otherwise been out of reach due to practical problem size and complexity. Moreover, generic techniques can be tailored to specific applications to get better results. Note that in many cases, the accelerated process remains optimal. Nevertheless, some procedures destroy the optimality of the branch-and-price approach.

While the overall results obtained by different researchers speak to the benefits of acceleration, specific comparisons illustrating actual savings are scant. The only attempt that we are aware of is that by Gamache et al. [24] for the air crew rostering problem. Embedding many heuristics described above in the proposed method, the authors have accelerated the standard column generation by three orders of magnitude and obtained excellent quality solutions. In test problems where comparison was possible, the optimality gap was no wider than 0.6%.

14.9 CONCLUSIONS

We have discussed accelerating strategies for column generation directed at the solution of vehicle routing and crew scheduling problems. We discussed techniques for each of the five phases of the solution process. In practice, these techniques have proved instrumental for the viability of this optimization approach. Since the advantages of individual procedures may vary with the application at hand, implementation should proceed in a step-by-step fashion. This way, the incremental benefit of each method may become more readily apparent.

Concerning the LP aspect of the method, accelerating strategies are now part of all commercial simplex software packages. Since column generation is mainly based on the same mathematical structure, it is not surprising that good implementations of column generation approaches make intensive use of similar heuristic strategies.

In addition to these more well known methods, substantial progress has recently been made for the IP aspect. A case in point is dynamic column generation at each branching node. Indeed, the results obtained by generating new columns at each node show the advantages of using such techniques for speed and solution quality. They further suggest that it may be more important to refresh the set of columns than to design a better branching strategy. We believe these specific results will be shown to be valid in most cases by future research.

References

[1] R.K. Ahuja, T. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, 1993.

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

[3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46:316–329, 1998.

[4] A. Caprara, M. Fischetti, and P. Toth. A Heuristic Algorithm for the Set Covering Problem. *Lecture Notes in Computer Science*, 1084:72–84, 1996.

[5] J.-F. Cordeau, G. Desaulniers, N. Lingaya, F. Soumis, and J. Desrosiers. Simultaneous Locomotive and Car Assignment at VIA Rail Canada. *Les Cahiers du GERAD*, G–98–61, Montréal, 1998.

[6] T.-G. Crainic and J.-M. Rousseau. The Column Generation Principle and the Airline Crew Scheduling Problem. *INFOR*, 25:136–151, 1987.

[7] G.B. Dantzig and P. Wolfe. Decomposition Principle for Linear Programs. *Operations Research*, 8:101–111, 1960.

[8] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M.M. Solomon, and F. Soumis. Crew Pairing at Air France. *European Journal of Operational Research*, 97:245–259, 1997.

[9] G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, and D. Villeneuve. *A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems.* In: *Fleet Management and Logistics*, T.G. Crainic and G. Laporte, editors, pages 57–93, Kluwer, 1998.

[10] G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. Daily Aircraft Routing and Scheduling. *Management Science*, 43:841–855, 1997.

[11] G. Desaulniers, J. Lavigne, and F. Soumis. Multi-Depot Vehicle Scheduling Problems with Time Windows and Waiting Costs. *European Journal of Operational Research*, 111:479–494, 1998.

[12] G. Desaulniers, J. Desrosiers, A. Lasry, and M. M. Solomon. Crew Pairing for a Regional Carrier. *Lecture Notes in Economics and Mathematical Systems*, 471:19–41, 1999.

[13] M. Desrochers and F. Soumis. A Reoptimization Algorithm for the Shortest Path Problem with Time Windows. *European Journal of Operational Research*, 35:242–254, 1988.

[14] M. Desrochers and F. Soumis. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, 23:1–13, 1989.

[15] M. Desrochers, J. Desrosiers, and M.M. Solomon. A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*, 40:342–354, 1992.

[16] J. Desrosiers, Y. Dumas, M.M. Solomon, and F. Soumis. Time Constrained Routing and Scheduling. In: *Network Routing*, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Handbooks in Operations Research and Management Science*, vol. 8, pages 35–139, Elsevier, 1995.

[17] J. Desrosiers, Y. Dumas, and F. Soumis. The Multiple Vehicle Dial-a-Ride Problem. *Lecture Notes in Economics and Mathematical Systems*, 308:15–27, 1988.

[18] J. Desrosiers, F. Soumis, and M. Desrochers. Routing with Time Windows by Column Generation. *Networks*, 14:545–565, 1984.

[19] M. Dror. Note on the Complexity of the Shortest Path Models for Column Generation in VRPTW. *Operations Research*, 42:977–978, 1994.

[20] Y. Dumas, J. Desrosiers, and F. Soumis. The Pickup and Delivery Problem with Time Windows. *European Journal of Operational Research*, 54:7–22, 1991.

[21] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized Column Generation. *Discrete Mathematics*, 194:229–237, 1999.

[22] L.R. Ford, and D.R. Fulkerson. A Suggested Computation for Maximal Multi-commodity Network Flows. *Management Science*, 5:97–101, 1958.

[23] M. Gamache and F. Soumis. A Method for Optimally Solving the Rostering Problem. In: *Operations Research in the Airline Industry*, G. Yu, editor, pages 124–157, Kluwer, 1998.

[24] M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A Column Generation Approach for Large Scale Aircrew Rostering Problems. *Operations Research*, 47:247–263, 1999.

[25] A.M. Geoffrion. Lagrangean Relaxation for Integer Programming. *Mathematical Programming Studies*, 2:82–114, 1974.

[26] P.C. Gilmore and R.E. Gomory. A Linear Programming Approach to the Cutting-Stock Problem. *Operations Research*, 9:849–859, 1961.

[27] J.-L. Goffin and J.-P. Vial. Multiple Cuts in the Analytic Center Cutting Plane Method. *SIAM Journal on Optimization*, 11:266–288, 2000.

[28] C. Hane, C. Barnhart, E.L. Johnson, R. Marsten, G.L. Nemhauser, and G. Sigismondi. The Fleet Assignment Problem: Solving a Large-Scale Integer Program. *Mathematical Programming*, 70:211–232, 1995.

[29] K.L. Hoffman and M. Padberg. Solving Airline Crew Scheduling Problems by Branch-and-Cut. *Management Science*, 39:657–682, 1993.

[30] I. Ioachim, J. Desrosiers, Y. Dumas, M.M. Solomon, and Daniel Villeneuve. A Request Clustering Algorithm for Door-to-Door Handicapped Transportation. *Transportation Science*, 29:63–79, 1995.

[31] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis. 2-Path Cuts for the Vehicle Routing Problem with Time Windows. *Transportation Science*, 33:101–116, 1999.

[32] N. Kohl and O.B.G. Madsen. An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation. *Operations Research*, 45:395–406, 1997.

[33] A.W.J. Kolen, A.H.G. Rinnooy Kan, and H.W.J.M. Trienekens. Vehicle Routing with Time Windows. *Operations Research*, 35:266–273, 1987.

[34] N. Lingaya, J.-F. Cordeau, G. Desaulniers, J. Desrosiers, and F. Soumis. Operational Car Assignment at VIA Rail Canada. *Les Cahiers du GERAD*, G-2000-55, Montréal, 2000.

[35] A. Mingozzi, M.A. Boschetti, S. Ricciardelli, and L. Bianco. A Set Partitioning Approach to the Crew Scheduling Problem. *Operations Research*, 47:873–888, 1999.

[36] C. Ribeiro and F. Soumis. A Column Generation Approach to the Multiple Depot Vehicle Scheduling Problem. *Operations Research*, 42:41–53, 1994.

[37] J.M. Rousseau and J. Desrosiers. Results Obtained with CREW-OPT, a Column Generation Method for Transit Crew Scheduling. *Lecture Notes in Economics and Mathematical Systems*, 430:349–358, 1995.

[38] M.M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. *Operations Research*, 35:254–265, 1987.

[39] K. Ziarati, F. Soumis, J. Desrosiers, S. Gélinas, and A. Saintonge. Locomotive Assignment with Heterogeneous Consists at CN North America. *European Journal of Operational Research*, 97:281–292, 1997.

[40] K. Ziarati, J. Desrosiers, F. Soumis, and M.M. Solomon. A Branch-First, Cut-Second Approach for Locomotive Assignment. *Management Science*, 45:1156–1168, 1999.