Approach based on Reduction the State Space to Solve the Vehicle Routing Problem

ABDELKADER LAMAMRI[1*], HACENE AIT HADDADENE[2]
[1]Universty of Djilali Bounaama, Khemis Miliana, Algeria
[2]University of Science and Technology Houari Boumediene - USTHB, Bab-Ezzouar, Algeria
*Email: plamamri@yahoo.fr

Abstract: - We are interested in problems from combinatorial optimization, more precisely, the vehicle routing problem with resource constraints. Given the large size of the problems encountered in practice, these models are solved by an approach based on column generation that can handle implicitly all feasible solutions and a master problem determining the best solution. We propose in this paper an approach to improve the acceleration of the method of column generation for solving the problem of construction vehicle routing, it is projected in each arc, the resources a vector of size smaller by using a Lagrangean relaxation algorithm to determine the coefficients of the projection arc combined with an algorithm for re-optimization, then generates a sub-set of complementary solutions to the master problem. The preliminary experiments of our technique gave good results on instances of random vehicle routing.

Key-Words: - Vehicle routing problem, path problems in graphs, decomposition methods, column generation

1 Introduction

The vehicle routing problem with time windows (VRPTW) is given by a set of customers N and a set of K vehicles available in a repository. This problem is to find a set of minimum cost tour, departing and returning to a single repository, where each customer is visited by one vehicle to satisfy some demand. Each customer must be served during a given time window. A vehicle arriving in advance at a customer waits until the start date of service without additional cost, time windows in this case are called 'hard'. Some models penalize the hold with an extra cost, these models are called 'soft ', but much research is devoted to the time windows 'hard'. A vehicle arriving late at a customer is not allowed to perform his service. A feasible tour is a series of visits (conducted by the same vehicle) respecting the time windows, which begins and ends at the same depot.

The VRPTW is defined on the networks $G^k = (X^k, A^k)$, X = $N \cup \{s^k, t^k\}$, where the depot is represented by the two nodes s and t, and $N = \{1, ..., n\}$ is the set of vertices representing the customer, and A the set of arcs that interconnect the customers and the depot. An arc (i, j) ∈ A means the possibility of linking the service customers i and j. To write the formulation of this problem, we introduce the following notations:

– c_{ij} • is the cost of the arc (i, j) ∈ A.

– t_{ij} • The arc duration (i, j) ∈ A.

– $[a_i, b_i]$ The time window during which the customer service $i \in N$ • must start.

– d_i • Customer demand i ∈ N.

– Q • capacity of each vehicle.

Assignment of customers to vehicles is called feasible if:

– The combined demand of customers visited by a vehicle does not exceed its capacity.
– Time constraints are met by each vehicle.
– Each customer is visited by one vehicle.
– Every vehicle that leaves the depot back to depot after completing his tour.

The problem is to find a feasible assignment of vehicles to tour the minimum cost.

2 Formulation

The VRPTW can then be formally described as the following multi-commodity network flow model with time window:

$$
\equiv \begin{cases}
\min \sum_{k=1}^{K} \sum_{(i,j)\in A} c_{ij}\, x_{ij}^{k} & (1) \\
\sum_{k=1}^{K} \sum_{j:(i,j)\in A} x_{ij}^{k} = 1 \ \text{ for } \ i \in N = \{1, \ldots\ldots, n\} & (2) \\
\sum_{(i,j)\in A} d_i x_{ij}^{k} \leq \mathrm{Q} \quad \text{ for } \ k \in \mathcal{K} & (2') \\
\sum_{i:(o^k,i)\in A} x_{o^k i}^{k} = 1 \ \text{ for } \ k \in \mathcal{K} & (3) \\
\sum_{i:(i,d^k)\in A} x_{id^k}^{k} = 1 \ \text{ for } \ k \in \mathcal{K} & (4) \\
\sum_{i:(i,j)\in A} x_{ij}^{k} = \sum_{l:(j,l)\in A} x_{jl}^{k} \ \text{ for } \ j \in N & (5) \\
x_{ij}^{k}\left(T_i^k + t_i^k - T_j^k\right) \leq 0 \ \text{ for } (i,j) \in A\,, k \in \mathcal{K} & (6) \\
a_i^k \leq T_i^k \leq b_i^k \ \text{ for } \ i \in N, k \in \mathcal{K} & (7) \\
x_{ij}^{k} \in \{0,1\}\,, T_i^k \geq 0 \ \text{ for } (i,j) \in A\,, k \in \mathcal{K} & (8)
\end{cases}
$$

Binary variables x_{ij}^{k} indicate if the tour takes the arc $(i,j) \in A$, while the variable T_i^k indicates the cumulative consumption of resource at each node i.

The objective function (1) minimizes the total travel cost. The constraints (2) ensure that each customer is visited exactly once, and $(2')$ state that a vehicle can only be loaded up to it's capacity. Next, equations $(3-5)$ indicate that each vehicle must leave the depot s ; after a vehicle arrives at a customer it has to leave for another destination; and finally, all vehicles must arrive at the depot t. The inequalities (6) establish the relationship between the vehicle departure time from a customer and its immediate successor. Finally constraints (7) affirm that the time windows are observed, and (8) are the integrality constraints. Note that an unused vehicle is modeled by driving the "empty" route (s,t), and the constraints (5) provides the cumulative consumption of resource at node j, since we have :

$$T_j^k = \max\left(a_i^k, T_i^k + t_{ij}^k\right) \qquad (9)$$

Note that the constraints $(3-7)$ are local constraints valid only for the network G. Only the partitioning constraints (2) are global constraints linking the K sub-networks. The relaxation of these binding constraints and the decomposition of the initial problem by sub-network will be an interesting option for a resolution. Finally, note that resource constraints $(6-7)$ make the problem (VRPTW) NP-hard. Even the problem of realizability is associated NP-complete [5].

3 Solving Approaches

3.1 Principles of Decomposition

There are two types of constraints in the system $(2)-(7)$:

(i) The partitioning constraints (2), said binder or global, binding all vehicles $k = 1, \ldots., K$,

(ii) Constraints $(3)-(7)$ of each vehicle $k \in \{1, \ldots., K\}$ and defining a legal road. The matrix associated with constraints $(3)-(7)$ is block diagonal, and the objective (1) is separable (for linear), solving the continuous relaxation of this model may be based on the Dantzig-Wolfe decomposition. In this type of decomposition, the constraints $(3)-(7)$ define K independent sub-problems and global constraints (2) are stored in the master problem. In a schema type column generation, it is alternatively solve the problem and the K master sub-problems. For a complete solution, this scheme can be applied at each node of the search tree. The major difficulty in solving sub-problems whose state spaces can grow exponentially with the number of resources, making essential use of heuristics. On the other hand, the convergence of the scheme of column generation is sensitive to the quality of solutions provided by the resolution of these sub-problems, the effective resolution of instances from real industry needs to find a good compromise between solution quality and time resolution of sub-problems. In what follows, we describe the general principle of column generation for the problem (VRPTW).

3.2 Column Generation, Master Problem and Sub-Problem

In this approach, the master problem is rewritten by a Set Partitioning (*MP*):

$$\min \textstyle\sum_{r\in\mathcal{R}} c_r x_r \qquad (10)$$

$$
\text{s.t} \equiv \begin{cases}
\sum_{r\in\mathcal{R}} a_{ir} x_r = 1 \ \text{for} \ i \in N = \{1, \ldots\ldots, n\} & (11) \\
x_r \in \{0,1\} \quad \text{for} \quad r \in \mathcal{R} & (12)
\end{cases}
$$

Or $\mathcal{R}$ means all feasible tours satisfying resource constraints and sequence between customers, c_r represents the cost of the tour $r \in \mathcal{R}$, $a_{ir} = 1$ if and only if customer i is visited by the tour r, and the binary variable x_r indicates the choice whether or not the tour r in the solution.

We note $(\overline{MP})$ the continuous relaxation of problem (MP) where the integrity constraints (12) are replaced by $x_r \geq 0$ for $r \in \mathcal{R}$. The total number of eligible tour $|\mathcal{R}|$ is generally an exponential function of the number $n = |N|$ of customers to cover the full enumeration of $\mathcal{R}$ is prohibited. However, it is possible to find in a reasonable time an optimal solution of $(\overline{MP})$ does generate a small subset of tours (ie, columns of the matrix of constraints).

In general, we solve at each iteration t the restricted master problem$(\overline{MP}^t)$:

$$\min \sum_{r\in\mathcal{R}^t} c_r x_r \qquad (13)$$

$$\text{s.t} \equiv \begin{cases} \sum_{r\in\mathcal{R}^t} a_{ir} x_r = 1 \text{ for } i \in N = \{1, \dots\dots, n\} & (14) \\ x_r \geq 0 \quad \text{for} \quad r \in \mathcal{R}^t & (15) \end{cases}$$

or, if δ^{t-1} denotes the vector of multipliers associated with no flights in the resolution $(\overline{MP}^{t-1})$, the tour r^{t-1} lower cost of reduced negative is defined by

$$r^{t-1} = arg \min_{r\in\mathcal{R}} \left(c_r - \sum_{i=1}^{n} \delta_i^{t-1} a_{ir} \right) \qquad (16)$$

The term column generation coms from the addition of column a_{r^t} in the constraint matrix of the master problem at each iteration t. This iterative process of solving the master problem $(13-15)$ and the sub-problem (16) is stopped when all tours are positive reduced cost in solving the problem by a sign that the continuous optimum is reached.

A variant of this method to accelerate the process in practice [3], is to add at each iteration a subset of complementary rounds of negative reduced cost instead of the single best round of the sub-problem (16). The desired maximum size of this subset of columns may be set to inbound to evolve during the algorithm. The overall complexity of the method is highly dependent on the complexity of the sub-problem that resource constraints make it NP-hard. However, it is often possible to solve in a reasonable time by an implicit enumeration of $\mathcal{R}$ by exploiting the graph structure of the sub-problem and using variants of shortest path algorithms.

3.3 Resolution of Sub-Problem for Column Generation

Noting that in the case of several sub-networks $= 1, \dots\dots, K$, under the resolution of problem (16) is decomposed by sub-networks, we omit the index k and the graph of the problem will be denoted as $G = (\{o\} \cup V \cup \{d\}, A)$.

Definition 1: A path from each origin o to node j, we associate a label $(\mathrm{C_j}, \mathrm{T_j})$ representing the state of its resources and cost.

Definition 2: Let $(\mathrm{C_j}, \mathrm{T_j})$ and $(\mathrm{C'_j}, \mathrm{T'_j})$ two labels associated with two feasible paths P and $P'$from o to j. We say that$(\mathrm{C_j}, \mathrm{T_j})$ dominates $(\mathrm{C'_j}, \mathrm{T'_j})$ (or alternatively that P dominates P' and there $(\mathrm{C_j}, \mathrm{T_j}) \leq (\mathrm{C'_j}, \mathrm{T'_j})$ (or as $\leq P'$) if and only if et

Definition 3: A label associated with a feasible path from o to d, is called effective if it is minimal in the sense of the order relation ≤. A path is said to be efficient if it is associated to a label effective

To solve this problem, Desrochers and Soumis [1] proposed a dynamic programming algorithm of pulling type. The nodes are sequentially treated in a topological order from source to destination. At each node the algorithm generates the labels by extending the paths corresponding to non dominated labels from its predecessor nodes. An extension is valid if the path is legal, otherwise it is suppressed. The dominance rule is then applied to eliminate all paths corresponding to dominated labels.

The algorithm proceeds in two basic steps. At each $j \in N$, we have the following operations:
1. Extension of roads (label generation),
2. Filtering (feasibility test),
3. Dominance (elimination of dominated labels).

For each given node j, the labels are created by extending those present at node i, such that $(i, j) \in A$. More precisely a new label $E(\mathrm{C_j}, \mathrm{T_j})$ given by

$$\mathrm{C}_j = \mathrm{C}_i + c_{ij}$$

$$T_j = \max\{a_j, T_i + t_{ij}\}$$

By assuming that all predecessors of node $j \in N$ have been considered, the dominance at node j can be interpreted as the determination of the Pareto optima for the multicriteria problem of 2 functions:

$$\begin{cases} \min_{i;(i,j)\in A} \left(C_i + c_{ij} \; ; \max\{a_j, (T_i + t_{ij})\} \right) \\ T_i + t_{ij} \leq b_j \end{cases}$$

Dominance relation ≤ is a partial order relation, the effective number of labels to be treated increases exponentially with the number of resources, which makes the procedure very difficult to extend.

In a recent work Nagih and Soumis [2] propose a method of aggregation of resources for PCC-CR by projection, in each node simultaneously using an algorithm of dynamic programming and Lagrangean relaxation.

3.4 Algorithms

And as the number of coefficients to adjust will be more important for the approach of projection arcs, finding the optimal multiplier u_{ij}^{*} require several iterations of DPA-L [2], this method can be expensive. To quickly obtain good heuristic solutions (feasible), our approach applied once DPA-L and then apply DPA-LND [2], using multipliers to find u_{ij} generate feasible columns and negative marginal cost. Specifically, we first choose a sequence of steps (p_k) as the standard $(\sum p_k)$ is divergent and $\lim_{k\to\infty} p_k = 0$, i.e. conditions ensuring convergence of the algorithm sub gradient. It applies primarily DP-L using multipliers u_{ij}^{k-1} of the previous iteration, we find the sub gradients Sg_{ij}^{k} corresponding arc (i, j) then we calculate the new Lagrange multipliers u_{ij}^{k} . This heuristic is certainly based on the fact that when k is large, the vector C_k reduced costs on the arcs of the network do not change much from one iteration to another of the algorithm for column generation. Thus, for k large, we can expect to see u_{ij}^{k} converge to an optimal value.

The main steps of our approach are summarized below:

Master problem
- Solve PMR

Subproblem

$-$ calculate the Lagrange multipliers u_{ij}^{k} (Projection arc).

$-$ calculate the solution $ZL(u_{ij}^{k})$, used in DPA $-$ L

$-$ calculate feasible solutions $ZLND(u_{ij}^{k})$, used in DPA $-$LND

Generated the complementary solutions negative reduced cost.

---

4 Numerical Results

This section presents the preliminary evaluation of our approach to the problem of construction of vehicle routing with a single resource. Solomons 100-customer Euclidean (VRPTW) instances are used to test our algorithm. In these instances, the travel time and the Euclidean distance between two customer locations are the same and this value is truncated to two decimal places. There are six deferent classes of instances depending on the geographic location of the customers (R: random; C: clustered; RC: mixed) and width of the scheduling horizon (1: short horizon; 2: long horizon). In this work, instances of type 1 are discarded due to the short horizon that does not allow a signicant number of routes to be sequenced to form a workday. Results are thus reported for R2, C2 and RC2. Due to the limitations of our exact approach, the computational study focuses on instances obtained by taking only the rst 25 customers from each original instance. Solomons (V RPTW) test instances are modied to t our problem. In particular, a value tmax to limit route duration is needed. This value was rst set to 100 in the case of R2 and RC2, and 200 in the case of C2. The value is larger for C2 because the service or dwell time at each customer is 90, as opposed to 10 for R2 and RC2. Finally, a gain of 1 is associated with each customer and weighted by an arbitrarily large constant to maximize rst the number of served customers, and then minimize the total distance. The results for the instances with reduced time windows are shown in Table1. In the table1, a particular instance is identied by its class and its index followed by a dot and the number of customers considered. For example RC202:25 is the second instance of class RC2, where only the rst 25 customers are considered. In these table, column Problem is the identier of the problem instance, ItrGC is the total number of iteration of (PM) solved by Simplex,Col is the total number of columns generated during the branch-and-price algorithm, T(ssp) is the computation time in seconds and Obj. is the total distance.

Table 1: Results comparison for Solomon instances

Problems	ItrGC	Col	T(ssp)	Obj
RC201.25	123	609	0.9	967.9
RC202.25	110	1132	221.0	961.6
RC203.25	713	2589	2566.2	751.3
RC205.25	218	944	5.4	974.9
RC206.25	444	1703	4.6	977.1
RC207.25	3119	13989	418.4	819.6
R201.25	218	577	1.0	772.8
R202.25	108	1030	127.0	694.0
R205.25	1326	4930	60.1	761.2
R210.25	71	918	121.4	704.6
R211.25	57	1150	42.9	623.7
C201.25	329	3448	5.1	679.5
C202.25	4023	13860	782.8	677.3

The comparison between the deferent methods and our approach has revealed that it has provided good results. These are best when certain conditions are met: The initialization of the algorithm and the choice of Lagrange multipliers and the displacement step.

5 Conclusion

In this paper, we proposed an algorithm for vehicle routing problem with resource constraints (VRPTW) which is an extension of (VRP) standard to take into account the more practical problem; we have mainly developed approaches to column generation and decomposition master problem and sub-problem. The difficulty of solving the sub-problem is directly related to the number of resources, we particularly studied the techniques of reduction of space resources, and this notion of reduction is a key element of the effectiveness of the overall resolution of problem.

References:

[1] M. Desrochers, F. Soumis, "A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows", INFOR 26, 191–212 (1988).

[2] A. Nagih, F. Soumis "Nodal aggregation of resource constraints in a shortest path problem", European Journal of Operational Research (2005).

[3] N. Touati, L. Létocart, and A. Nagih. "Solutions diversification in a column generation scheme". Discrete Optimization (2008).

[4] N. Touati, L. Létocart, and A. Nagih. "Reoptimization in a column generation scheme". Computers and Operations Research (2008).

[5] Vangelis Paschos, "Livre, optimisation combinatoire 3: applications", Hermès Science. ch 10 (2005).