On the Capacitated Vehicle Routing Problem

T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter, Jr.*

1 Introduction

We consider the *Vehicle Routing Problem* (VRP), introduced by Dantzig and Ramser [10], in which a central depot $\{0\}$ uses k independent vehicles of identical capacity C, to service integral demands d_i for a single commodity from customers $i \in N = \{1, \cdots, n\}$. Delivery is to be accomplished at minimum total cost, with $c_{ij} \geq 0$ denoting the i to j transit cost, $0 \leq i, j \leq n$. The cost structure is assumed *symmetric*, i.e., $c_{ij} = c_{ji}$ and $c_{ii} = 0$.

Combinatorially, a solution for this problem consists of a partition of N into k *routes* $\{R_1, \ldots, > R_k\}$, each satisfying $\sum_{j \in R_i} d_j \leq C$, and a corresponding permutation, or *tour*, σ_i of each route specifying the service ordering. This problem is naturally associated with the complete undirected graph consisting of nodes $N \cup \{0\}$, edges E, and edge-traversal costs $c_{ij}, \{i,j\} \in E$. In this graph, a solution is the union of k cycles whose only intersection is the depot node. Each cycle corresponds to the route serviced by one of the k vehicles. An integer programming formulation can be given as follows:

$$
\begin{aligned}
min \quad & \sum_{e \in E} c_e x_e & & & \\
& \sum_{e=\{0,j\} \in E} x_e & = & 2k & (1.1)\\
& \sum_{e=\{i,j\} \in E} x_e & = & 2 \quad \forall i \in N & (1.2)\\
& \sum_{\substack{e=\{i,j\} \in E \\ i \in S, j \notin S}} x_e & \geq & 2b(S) \quad \forall S \subset N, \ |S| > 1 & (1.3)\\
& 0 \leq x_e & \leq & 1 \quad \forall e = \{i,j\} \in E, \ i,j \neq 0 & (1.4)\\
& 0 \leq x_e & \leq & 2 \quad \forall e = \{0,j\} \in E & (1.5)\\
& x_e & & integral \quad \forall e \in E. & (1.6)
\end{aligned}
$$

For ease of computation, we define $b(S) = \lceil (\sum_{i \in S} d_i)/C \rceil$, an obvious lower bound on the number of trucks needed to service the customers in set S. A (possibly) stronger inequality may be obtained by computing the solution to a Bin Packing Problem (BPP) with the customer demands in set S being packed into bins of size C. Constraints (1.1) and (1.2) are the *degree constraints*. The *capacity constraints* (1.3) can be viewed as a generalization of

*Research partially supported by NSF Grant DMS-9527124 and Texas ATP Grant 97-3604-010.

the subtour elimination constraints from the Traveling Salesman Problem (TSP); they serve to enforce connectivity of the solution, and that the total demand of no route exceeds C.

It is clear that the VRP is closely related to two difficult combinatorial problems. When $C = \infty$, we have an instance of the Multiple Traveling Salesman Problem (MTSP), which can be transformed into an equivalent TSP instance by adjoining to the graph $k-1$ additional copies of node 0 and its incident edges (there are no edges among the k depot nodes). On the other hand, the question of whether there exists a feasible solution for a given instance of the VRP is an instance of the BPP. The decision version of this problem is conceptually equivalent to a VRP model in which $c_{ij} = 0\ \forall i, j$ (any feasible solution is optimal). A feasible solution to the full problem is a TSP tour (in the expanded graph) for which total demand along each of the k segments joining successive depot copies does not exceed C.

Because of the interplay between the two underlying models, instances of the Vehicle Routing Problem can be extremely difficult to solve in practice. This difficulty stems from the fact that the cost structure is dictated purely by routing considerations and does not account for the packing structure. Hence, the routing and packing requirements are sometimes in conflict. This observation suggests the exploration of decomposition-based optimization techniques involving relaxation of one or the other of the underlying structures. Here we investigate a novel way to isolate the TSP structure from that of the BPP, permitting exploitation of known techniques for optimizing over the TSP polytope; this leads to a separation routine for the capacity constraints and other classes of valid inequalities. This approach for separation can be applied more generally to combinatorial optimization problems that are the intersection of two underlying models.

Here we outline the details of this approach and present computational results obtained using the SYMPHONY framework for parallel branch, cut, and price (see the *SYMPHONY User's Guide* [19] and [13]).

2 Separation of the Capacity Constraints

Many classes of valid inequalities for the VRP polytope have been reported in the literature (see [14, 15, 1, 7, 9, 16, 5, 4, 6]), but the separation problem remains difficult to solve for most known classes. Classes that can be effectively separated tend to be ineffective in the context of branch and cut. We focus primarily on the problem of separating an arbitrary fractional point from the VRP polytope using the capacity constraints (1.3). The separation problem for these constraints was shown to be $\mathcal{NP}$-complete by Harche and Rinaldi [5], even for $b(S) = \lceil (\sum_{i \in S} d_i)/C \rceil$; i.e., the LP relaxation of our formulation is $\mathcal{NP}$-complete.

Because of the apparent intractability of solving the separation problem, a good deal of effort has been devoted in the literature to developing effective separation heuristics for these constraints. However, until the paper by Augerat, et al. [5], most known algorithms were not very effective at locating violated capacity constraints. In [4], it is shown that the fractional version of these constraints (i.e., with $b(S) = (\sum_{i \in S} d_i)/C$) is polynomially separable. This method can be used as a heuristic for finding violated constraints of the form (1.3).

Returning to the model discussed earlier in which the TSP is viewed as a relaxation of the VRP on a slightly expanded graph, a TSP tour provides a feasible VRP solution when it also yields a feasible solution to the BPP. We denote by $\mathcal{T}$ the TSP polytope, i.e., the

convex hull of incidence vectors of all TSP tours, and by $\mathcal{R}$ the polytope generated by the incidence vectors of tours corresponding to feasible VRP solutions. Thus, $\mathcal{R} \subseteq \mathcal{T}$ and the extremal elements of $\mathcal{R}$ are among those of $\mathcal{T}$.

Suppose that at some node in the branch and cut search tree, the LP solver has returned an optimal solution $\hat{x}$ to the current LP relaxation. We define the *support graph* or *fractional graph* corresponding to $\hat{x}$ as $\hat{G} = (N \cup \{0\}, \hat{E})$ where $\hat{E} = \{e : \hat{x}_e 0\}$. Suppose that $\hat{x}$ is integral. If $\hat{x} \notin \mathcal{T}$, then $\hat{G}$ must be disconnected since the degree constraints are included explicitly in each LP relaxation. In this case, the set of nodes in any connected component of $\hat{G}$ that does not include the depot induces a violated capacity constraint, so we *CUT*; that is, we add this inequality to the LP and re-optimize. On the other hand, when $\hat{x} \in \mathcal{T}$, we consider whether $\hat{x} \in \mathcal{R}$. If not, then $\hat{x}$ must again violate a capacity constraint induced by the nodes of some depot-to-depot segment of the tour corresponding to $\hat{x}$, so we CUT. Finally, when $\hat{x} \in \mathcal{R}$, then $\hat{x}$ provides a feasible solution to the VRP and investigation of the current search node terminates. Thus we assume henceforth that $\hat{x}$ is not integer-valued.

2.1 Heuristics

Because of the difficulty of this separation problem, we first apply several simple heuristics in an attempt to determine a capacity constraint violated by $\hat{x}$. These heuristics work within the support graph $\hat{G}$; we assume $\hat{G}$ is connected, and associate with it a vector $\omega \in \mathbb{R}^{\hat{E}}$ of edge weights whose components are defined by $\omega_e = \hat{x}_e$. We use the notation

$$\omega(F) = \sum_{e \in F} \omega_e, \; F \subseteq \hat{E}, \text{and} \tag{2.7}$$

$$\delta(S) = \{\{i,j\} \in \hat{E} : i \in S, j \notin S\}, \; S \subseteq N \cup \{0\}. \tag{2.8}$$

The *connected components heuristic* considers one-by-one the components of the support graph after removing the depot. Suppose S is the node set of such a component. If $\hat{x}$ violates the capacity restriction determined by S, we CUT. If not, we replace $S \leftarrow S \backslash \{v\}$, with $v \in S$ chosen so that violation becomes more likely after its removal (specifically, a node $v \in C_i$ such that $b(C_i \setminus \{v\}) = b(C_i)$ and $\omega(\delta(C_i \setminus \{v\})) - 2b(C_i \setminus \{v\}) < \omega(\delta(C_i)) - 2b(C_i)$). The procedure iterates in this fashion until no such node can be found. If no violated inequality is discovered, we move on to consider the next component of the support graph. A variant, the *2-connected components heuristic*, which begins with the 2-edge connected components (those for which the removal of at least two edges is required to disconnect the component) of the support graph after the depot's removal, was also implemented.

In the *shrinking heuristic*, if the two end nodes of any edge of the support graph not adjacent to the depot determine a capacity constraint violated by $\hat{x}$ (i.e., the set $S = \{i,j\}$ induces a violated capacity constraint), we CUT. If not, suppose edge $e = \{i,j\}$ not adjacent to the depot satisfies $\omega_e \geq 1$. In this case, we *shrink*, or *contract*, edge e, identifying its end nodes and summing components of ω for edges which are identified in consequence. In the resulting *contracted graph*, the two endpoints that are identified form a *supernode* associated with a subset of N. The demand associated with this new supernode is the sum of the demands of the nodes it contains. It is not difficult to see that this shrinking process does not interfere with violated capacity constraints – if S induces a violated capacity constraint,

then there must exist a set S', with either $i, j \in S'$ or $i, j \notin S'$, which also induces a violated capacity constraint. Thus the heuristic proceeds iteratively, alternately shrinking an edge e for which $\omega_e \geq 1$ and checking whether any pair of end nodes determines a violated capacity constraint. If a violated constraint is produced, we CUT; if not, the procedure iterates until every edge e is either adjacent to the depot or satisfies $\omega_e < 1$. Note that in a contracted graph, it is possible that $\omega_e > 1$ for some edge e.

The *extended shrinking heuristic* is based on the minimum cut algorithm of Nagamochi and Ibaraki [17] and is similar in spirit to the shrinking heuristic. In this extension, shrinking continues with edges of weight less than one in the order prescribed by the algorithm of [17]. As in the shrinking heuristic, each edge is checked before contraction to determine if it induces a violated capacity constraint. Because the sequence of edges is chosen in such a way that the weight of each cut examined is "small," this algorithm may lead to the discovery of violated capacity constraints not discovered by other heuristics.

If the above heuristics fail to locate a violated capacity constraint, we apply the *greedy shrinking heuristic* [5]. Here we begin with a small set of nodes S, selected either at random or by one of several heuristic rules. In each iteration, we try to grow the set S by adding a node $j \notin S$ such that $\sum_{e \in \{\{i,j\} \in E : i \in S\}} x_e$ is maximized (and hence $\sum_{e \in \delta(S)} x_e$ is minimized). Notice the contrast between this strategy and that taken by the *connected components heuristic*, where a large initial set is shrunk using a similar rule.

2.2 The Decomposition Algorithm

If the heuristics described above have failed to produce a violated capacity constraint, then we resort to a *decomposition algorithm*, which originally appeared in [18]. Given a fractional solution $\hat{x}$ to some LP relaxation, we attempt to determine whether $\hat{x}$ lies within $\mathcal{T}$ by expressing $\hat{x}$ as a convex combination of incidence vectors of tours. More precisely, where T denotes the matrix whose columns are the extreme points of $\mathcal{T}$, we are asking whether

$$\max\{\mathbf{0}^\top \lambda : T\lambda = \hat{x},\ \mathbf{1}^\top \lambda = 1,\ \lambda \geq 0\}, \tag{2.9}$$

has a finite optimum. If such a decomposition of $\hat{x}$ into a convex combination of tour vectors is possible, Carathéodory's Theorem assures only a modest number of columns of T will be required. Generating the matrix T is, needless to say, difficult at best, and the reader should for the moment put aside consideration of exactly *how* this is to be accomplished.

Note that when $\hat{x}$ violates a capacity constraint and $\hat{x}$ is a convex combination of tour vectors, then some member of the decomposition must also violate that same restriction. Thus, given such a convex representation for $\hat{x}$, we examine the depot-to-depot segments of its tours attempting to find a violated capacity restriction. If successful, we use the violated inequality to CUT. If not, i.e., when there is a convex decomposition of $\hat{x}$ (hence $\hat{x} \in \mathcal{T}$), yet no evident capacity violation, then separation fails and we must BRANCH.

When no convex decomposition of $\hat{x}$ exists, then $\hat{x} \notin \mathcal{T}$ and the Farkas Theorem provides a hyperplane separating $\hat{x}$ from $\mathcal{T}$. Specifically, there must exist a vector a and scalar α for which $at \geq \alpha$ for each column t of T, yet $a\hat{x} < \alpha$. This inequality corresponds to a row of the current basis inverse when solving the LP defined in (2.9) and can be readily obtained. The inequality $ax \geq \alpha$ is returned to CUT, and the process iterates.

We suggest two means for dealing with the intractability of matrix T. One approach is to restrict the column set of the matrix T. Note that any tour t present in a convex decomposition of $\hat{x}$ must *conform* to $\hat{x}$; i.e., t must satisfy $t_e = 1$ whenever $\hat{x}_e = 1$ and $t_e = 0$ when $\hat{x}_e = 0$. Thus we may require that these stipulations hold for all columns of T. When the *fractional support* of $\hat{x}$, the set of edges e for which $0 < \hat{x}_e < 1$, is of small cardinality, we can simply enumerate all such tours using depth-first search and thereby create T *explicitly*. The disadvantage of this is that any resulting Farkas inequalities must be "lifted" in order to be made valid, another computationally intensive task.

A second approach is to use column generation to handle T *implicitly*. Here T is initially comprised of only a partial collection of tours. The algorithm proceeds as before, asking whether $\hat{x}$ is a convex combination of the columns of T. When a convex representation is found among the columns of T, we again check whether any of its tours determines a violated capacity constraint. If so, we CUT. As before, when there is no convex decomposition of $\hat{x}$ using the known tours, the Farkas Theorem provides (a, α) for which $at \geq \alpha$ for each column t of T, but $a\hat{x} < \alpha$. Now we minimize over $\mathcal{T}$ with cost vector a using TSP optimization techniques. Suppose t^* is the incidence vector of the optimal tour which results. If $at^* \geq \alpha$, then the minimization guarantees that $ax \geq \alpha$ separates $\hat{x}$ from $\mathcal{T}$, so this inequality is passed to CUT. If $at^* < \alpha$, then t^* is not among the columns of T, so t^* is appended to T and we ask again whether a convex decomposition of $\hat{x}$ can be obtained from the columns of T. The process iterates until either a decomposition is found or it is proven that $\hat{x} \notin \mathcal{T}$.

2.3 Extensions

We indicate several means to improve the efficiency of the decomposition algorithm, as reported in [11]. Consider the effect of further restricting the columns of T to only extreme points of $\mathcal{P}'$. At first, this may seem to defeat our purpose since in this case, we cannot possibly succeed in finding a decomposition. However, the Farkas cut generated when we fail to find a decomposition separates $\hat{x}$ from $\mathcal{P}'$ and hence can still be used to CUT. This basic approach has also been used to generate valid inequalities for the TSP in [2].

Next, consider the polytope $\mathcal{P}''$ defined as the convex hull of incidence vectors of solutions whose cost is less than or equal to the current upper bound. It is immediate that $\mathcal{P}'' \subseteq \mathcal{P}'$ and furthermore, it is easily seen that $\min\{cx : x \in \mathcal{P}'\} = \min\{cx : x \in \mathcal{P}''\}$. We can therefore further limit enumeration to those columns whose cost is less than the current upper bound and still generate a valid Farkas inequality.

This observation suggests considering what happens when we limit the enumeration to the point where T becomes empty. In this case, the algorithm has proven that there is no feasible solution with cost below the current upper bound and which conforms to the current fractional solution. Hence, we can impose the following *no-columns cut*:

$$\sum_{e:\hat{x}_e=1} x_e - \sum_{e:\hat{x}_e=0} x_e \leq |E_1| - 1. \tag{2.10}$$

These cuts are a special case of *hypo-tours* introduced in [5]. If we limit column generation by both feasibility and cost, as suggested, we can *always* generate one of these inequalities. To see this, suppose T is composed only of extreme points of $\mathcal{P}''$. Then each of these columns generates a possible new upper bound and can hence be removed from T, leaving T empty.

3 Computational Results

The separation routines were embedded in the generic, parallel branch and cut framework SYMPHONY developed by Ralphs [18] and Ladányi [12]. The implementation of this shell is reviewed in [13]. Our test set consisted of medium-sized VRP instances taken from the TSPLIB [20] repository and from that maintained by Augerat [3]. The full test set and source code used in this paper are available at `http://www.branchandcut.org/VRP`.

Ten of the problems in this set are derived from that used by Christofides and Eilon in [8], which is included among the VRP instances at TSPLIB. Three of those (E-n76-k7, E-n76-k8, and E-n101-k8) were solved for the first time during this work using the parallel version of SYMPHONY on an IBM SP2 parallel computer with up to 80 processors. Recently, Blasum and Hochstättler solved E-n76-k7 and E-n76-k8 on a single processor using additional cutting planes [6]. We have also solved E-n76-k7 sequentially, but required a larger tree. It is worth noting that the smallest instance we are aware of that has not been solved to optimality is B-n50-k8. We have solved a version of this model with truck capacity increased to 150, but have been unable to solve the original instance.

References

[1] J.R. Araque, L. Hall, and T. Magnanti (1990): Capacitated Trees, Capacitated Routing and Associated Polyhedra. Discussion paper 9061, CORE, Louvain La Nueve

[2] D. Applegate, R. Bixby, V. Chvátal, and W. Cook (2001): TSP Cuts Which Do Not Conform to the Template Paradigm. *Computational Combinatorial Optimization*, D. Naddef and M. Jünger, eds., Springer, Berlin, 261–303

[3] P. Augerat (1995): VRP problem instances. Available at `http://www.branchandcut.org/VRP/data/`

[4] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef (1998): Separating Capacity Constraints in the CVRP Using Tabu Search. European Journal of Operations Research, **106**, 546–557

[5] P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef, G. Rinaldi (1995): Computational Results with a Branch and Cut Code for the Capacitated Vehicle Routing Problem. Research Report 949-M, Université Joseph Fourier, Grenoble, France

[6] U. Blasum and W. Hochstättler (2000): Application of the Branch and Cut Method to the Vehicle Routing Problem. Zentrum für Angewandte Informatik, Köln, Technical Report zpr2000-386

[7] V. Campos, A. Corberán, and E. Mota (1991): Polyhedral Results for a Vehicle Routing Problem. European Journal of Operations Research 52, 75–85

[8] N. Christofides and S. Eilon (1969): An Algorithm for the Vehicle Dispatching Problem. Operational Research Quarterly **20**, 309–318

[9] G. Cornuéjols and F. Harche (1993): Polyhedral Study of the Capacitated Vehicle Routing Problem. Mathematical Programming **60**, 21–52

[10] G.B. Dantzig and R.H. Ramser (1959): The Truck Dispatching Problem. Management Science **6**, 80–91

[11] L. Kopman (1999): A New Generic Separation Routine and Its Application in a Branch and Cut Algorithm for the Vehicle Routing Problem. Ph.D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA

[12] L. Ladányi (1996): Parallel Branch and Cut and Its Application to the Traveling Salesman Problem. Ph.D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA

[13] L. Ladányi, T.K. Ralphs, and L.E. Trotter (2001): Branch, Cut, and Price: Sequential and Parallel. *Computational Combinatorial Optimization*, D. Naddef and M. Jünger, eds., Springer, Berlin, 223–260

[14] G. Laporte and Y. Nobert (1981): Comb Inequalities for the Vehicle Routing Problem. Methods of Operations Research **51**, 271–276

[15] G. Laporte, Y. Nobert and M. Desrochers (1985): Optimal Routing with Capacity and Distance Restrictions. Operations Research **33**, 1050–1073

[16] A.N. Letchford, R.W. Eglese, and J. Lysgaard (2001): Multistars, Partial Multistars and the Capacitated Vehicle Routing Problem. Technical Report available at `http://www.lancs.ac.uk/staff/letchfoa/pubs.htm`

[17] H. Nagamochi and T. Ibaraki (1992): Computing Edge Connectivity in Multigraphs and Capacitated Graphs. SIAM Journal of Discrete Mathematics 5, 54–66

[18] T.K. Ralphs (1995): Parallel Branch and Cut for Vehicle Routing. Ph.D. Dissertation, Field of Operations Research, Cornell University, Ithaca, NY, USA

[19] T.K. Ralphs (2001): SYMPHONY Version 2.8 User's Guide. Available at `www.branchandcut.org/SYMPHONY`

[20] G. Reinelt (1991): TSPLIB—A traveling salesman problem library. ORSA Journal on Computing **3**, 376–384. Update available at `http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/`