ISYE 4133 Advanced Optimization

Vehicle Routing Problem

Nicholas Loprinzo & Karen Loscocco

November 29, 2019

Contents

1 Executive Summary

In this paper, we examine two well-known variants of the Vehicle Routing Problem (VRP): the Capacitated Vehicle Routing Problem (CVRP) and the Vehicle Routing Problem with Time Windows (VRPTW). VRPs are used in a variety of applications which motivate research in both industry and academia. We explore two approaches to solve both of these problems, a flow based approach and a set partitioning approach using column generation and a corresponding pricing problem. After running the models using Gurobi, we have come to the conclusion that flow formulations are slower to solve than the corresponding set partitioning with column generation approach for models with more than 10 nodes. While the flow formulation is concise and easy to interpret, it does not scale well to problems of real world size. Below, we include the model formulations, implementations in Gurobi, as well as our solutions and comments on the scalability of the approaches.

2 Introduction

The CVRP has a set of cities with known demands serviced by a fleet of vehicles of identical capacity that must start and return to a single depot. The objective is to minimize the total distance traveled by all vehicles while satisfying the demand of all cities.

We model this problem as an optimization problem with the following constraints:

- Each route begins and ends at the depot.
- Each city is visited exactly once.
- The total demand serviced by each route does not exceed the capacity of the vehicle.

The VRPTW is an extension of the classic VRP with the additional specification of the time intervals within which vehicles must arrive at cities. A vehicle can arrive early but must wait until the time window starts before moving on to the next city. The vehicle can arrive later than the end of the time window for a city.

We model this problem as an optimization problem with the following additional constraint:

- The time windows of the cities are observed.

We solved these problems in Gurobi, a mathematical optimization solver, which required us to translate our problems into mathematical models. We ran the Gurobi optimizer for a period of 20 hours for each of our formulations on the full set of example data. Although it continued to make progress, it did not converge to the optimal solutions during that time. However, it did identify feasible solutions. On scaled down versions of the data, it did identify optimal solutions in a reasonable amount of time using our formulations. We model these problems in the next section using two approaches.

3 Model

For our models we define a set of cities, $V = \{1, 2, \ldots, n\}$, with $\{1\}$ representing the depot for CVRP and $\{0\}$ representing the depot for VRPTW. For the formulations, we use i and j to index the set of cities V. Each city has a demand b_i, and in the VRPTW, a time window during which deliveries can be made $[a_i, b_i]$. The set of arcs, $A = \{(i, j), \forall i \forall j\}$, representing directed connections between each city, have associated distances, d_{ij}. Each vehicle has a capacity C.

The example data does not specify a number of vehicles so we do not constrain our model and instead assume that the number of vehicles allocated will be equal to the number of routes in the solution.

3.1 CVRP

3.1.1 Flow Formulation

The first approach for modeling the CVRP is a well-known flow based formulation. We define a binary decision variable, x_{ij}, and a continuous decision variable, q_i.

- x_{ij} is a binary decision where $x_{ij} = 1$ if a vehicle travels on arc (i, j), the arc connecting city i and city j, and 0 otherwise.
- q_i is the load on the vehicle required to service the cities on the route up to and including the current city i.

The integer linear programming model of the CVRP can be written as:

$$\text{Minimize} \sum_{i,j \in V} d_{ij} x_{ij} \tag{3.1}$$

$$\text{Subject to} \sum_{j \in V, j \neq i, j \neq 1} x_{ij} = 1 \quad \forall i \tag{3.2}$$

$$\sum_{i \in V, i \neq j, i \neq 1} x_{ij} = 1 \quad \forall j \tag{3.3}$$

$$\sum_{j \in V} x_{1j} = \sum_{i \in V} x_{i1} \tag{3.4}$$

$$q_i + b_j - q_j \leq (1 - x_{ij})C \quad \forall i \forall j \setminus \{1\} \tag{3.5}$$

$$x_{ij} \in \{0, 1\} \;\; \forall i \forall j, \;\; 0 \leq q_i \leq C \;\; \forall i \tag{3.6}$$

The objective function (3.1) minimizes the total distance traveled. The model constraints (3.2) and (3.3) ensure that each city except the depot is visited exactly once. Constraint (3.4) ensures that the number of vehicles leaving the depot equals the number of vehicles returning to the depot. Constraint (3.5) establishes a relationship between the load on a vehicle at a city and the load on a vehicle at that city's immediate successor. This makes sure that the sum of demands of cities in a route is less than the capacity of the vehicle. This constraint also ensures that the solution contains no cycles disconnected from the depot. The last constraint (3.6) specifies the domains of the variables.

3.1.2 Set Partitioning with Column Generation

The second approach for solving this problem involves introducing a variable for each possible route. We define a decision variable x_k and two known quantities a_{ik} and c_k.

- x_k is a binary decision to include a feasible route or not. $x_k = 1$ if route k is included in the solution, 0 otherwise.
- a_{ik} are elements of a matrix corresponding to all possible feasible routes, so a_{ik} will have 1s where route k includes city i.
- c_k is the total distance or cost of a route k.

The linear integer program becomes the following:

$$\text{Minimize} \sum_k c_k x_k \tag{3.7}$$

$$\text{Subject to} \sum_k a_{ik} x_k = 1 \quad \forall i \in V \setminus \{1\} \tag{3.8}$$

$$x_k \in \{0,1\} \ \forall k \tag{3.9}$$

The objective function (3.7) minimizes the total cost of all routes included in the solution. The model constraint (3.8) ensures that each city is included in a route. This is the master problem.

We use a pricing problem to avoid generating all feasible routes. The pricing problem determines a feasible route to add to the master problem by finding a route with the best reduced cost. This approach is called column generation.

An additional route added will have a negative reduced cost $\tilde{c}_k = c_k - \sum_i a_{ik}\pi_i$ where π_is are the dual variables associated with the constraints of the master problem.

Therefore, the pricing problem as a linear integer program is the following:

$$\text{Minimize} \sum_{i,j \in V} (d_{ij} - \pi_j) x_{ij} \tag{3.10}$$

$$\text{Subject to} \sum_{j \in V} x_{1j} = 1 \tag{3.11}$$

$$-\sum_{i\in V} x_{i1} = -1 \tag{3.12}$$

$$\sum_{j\in V, j\neq i} x_{ij} = \sum_{j\in V, j\neq i} x_{ji} \quad \forall i \tag{3.13}$$

$$q_i + b_j - q_j \le (1 - x_{ij})C \quad \forall i \forall j \setminus \{1\} \tag{3.14}$$

$$x_{ij} \in \{0,1\} \ \forall i \forall j, \ 0 \le q_i \le C \ \forall i \tag{3.15}$$

The objective function (3.10) will select a route with the best reduced cost. This equation can conceptually be explained as the original cost of the route minus the reduced cost of including node j in the formulation. If the reduced cost is greater than the original cost, then our objective value is negative. We loop solving an LP relaxation of the master problem and the pricing problem until the pricing problem can no longer generate routes with negative reduced costs. Then we solve the master problem as an IP to get a final solution. This is the column generation method. The constraints (3.11) and (3.12) ensure that the route in the solution is connected to the depot. Constraint (3.13) and (3.14) have the same meaning as the flow based formulation.

3.2 VRPTW

3.2.1 Flow Formulation

The first approach for modeling the VRPTW is a flow based formulation. We define a binary decision variable, x_{ij}, and three continuous decision variables, q_i, h_i,t_{ij}, as well as a large positive constant M.

- x_{ij} is a binary decision where $x_{ij} = 1$ if a vehicle travels on arc (i, j).
- q_i is the load on the vehicle required to service the cities on the route up to and including the current city.
- h_i is the time when a vehicle arrives at city i.
- t_{ij} is the transit time along arc (i, j). It includes the service time at city i and the travel time from city i to city j. For this formulation, $t_{ij} = \frac{d_{ij}}{3}$.
- $M_{ij} = max\{b_i + t_{ij} - a_j, 0\}$.

The integer linear programming model of the VRPTW can be written as:

$$\text{Minimize} \sum_{i,j\in V} d_{ij} x_{ij} \tag{3.16}$$

$$\text{Subject to} \sum_{j\in V, j\neq i, j\neq 0} x_{ij} = 1 \quad \forall i \tag{3.17}$$

$$\sum_{i\in V, i\neq j, i\neq 0} x_{ij} = 1 \quad \forall j \tag{3.18}$$

$$\sum_{j\in V} x_{0j} = \sum_{i\in V} x_{i0} \tag{3.19}$$

$$q_i + b_j - q_j \le (1 - x_{ij})C \quad \forall i \forall j \setminus \{0\} \tag{3.20}$$

$$h_i + t_{ij} - h_j \le (1 - x_{ij})M \quad \forall i \forall j \setminus \{0\} \tag{3.21}$$

$$x_{ij} \in \{0,1\} \ \forall i \forall j, \ 0 \le q_i \le C \ \forall i, \ a_i \le h_i \le b_i \ \forall i \tag{3.22}$$

The objective value and the first four constraints have the same meaning as the those in the CVRP formulation. Constraint (3.21) establishes a relationship between the vehicle arrival time at a city and its immediate successor. The bounds on h_i in (3.22) ensures that the time windows are observed.

3.2.2 Set Partitioning with Column Generation

The second approach for solving this problem is to use a pricing problem and the column generation method. Following the same master problem formulation as the CVRP, the pricing problem as a linear integer program is the following:

$$\text{Minimize} \sum_{i,j\in V} (d_{ij} - \pi_j) x_{ij} \tag{3.23}$$

$$\text{Subject to} \sum_{j\in V} x_{1j} = 1 \tag{3.24}$$

$$-\sum_{i\in V} x_{i1} = -1 \tag{3.25}$$

$$\sum_{j\in V, j\ne i} x_{ij} = \sum_{j\in V, j\ne i} x_{ji} \quad \forall i \tag{3.26}$$

$$q_i + b_j - q_j \le (1 - x_{ij})C \quad \forall i \forall j \setminus \{1\} \tag{3.27}$$

$$h_i + t_{ij} - h_j \le (1 - x_{ij})M \quad \forall i \forall j \setminus \{1\} \tag{3.28}$$

$$x_{ij} \in \{0,1\} \ \forall i \forall j, \ 0 \le q_i \le C \ \forall i, \ a_i \le h_i \le b_i \ \forall i \tag{3.29}$$

The objective function and constraints have the same meaning as the VRPTW flow based formulation and the pricing problem from the CVRP. Again, we loop over the master problem and pricing problem until we can generate no routes with a negative reduced cost.

4 Solution and Analysis

The next sections detail our solutions from implementing the above models in Gurobi.

4.1 CVRP

4.1.1 Flow Formulation

The flow formulation of the CVRP for the full graph was unable to converge in a 20 hour run. The gap percentage on the solution reached 34.3%, and the rate at which this gap approached 0% was decreasing. We tested the flow formulation on graphs of different sizes, and found that this formulation can solve graphs with up to 10 branches and 1 depot within a couple of seconds. Adding more branches to the graph would increase the solve time by minutes or hours. The optimal solution achieved with this formulation on a set of 10 branches and 1 depot is shown in figure (5.1). The red square represents the depot, the blue circles represent the branches and the green lines represent the arcs used. The optimal objective value for this solution is 362.

4.1.2 Set Partitioning with Column Generation

The set partitioning approach was also able to converge to an optimal objective of 362 on the same 10 node subset as the flow formulation. The set partitioning problem was slower than the flow formulation for this trial, taking about a minute to converge on the optimal solution. The optimal solution found using the set partitioning formulation is shown in figure (5.2). Notice that it is the same as the flow formulation.

While the flow formulation performed better on the smaller set, the set partitioning formulation was more well equipped to solve the full graph. After a couple of hours the set partitioning formulation had found a solution with an objective value of 1022. The optimal objective for the problem it was solving is 785. The solution can be seen in figure (5.3). Once we received a solution with an objective value of 1022, we ceased iterating on the column generation process since it was beginning to take much longer for the optimizer to find solutions with a negative reduced cost. Depending on the time and resource constraints we face in a real world application, we can adjust whether or not we would continue to run the column generation procedure to generate a better solution. However, for the current circumstance, we felt 1022 was a good solution for the time spent solving.

4.2 VRPTW

4.2.1 Flow Formulation

As in the CVRP problem, the flow formulation of the VRPTW was unable to converge in a 20 hour run. We tested the flow formulation on graphs of different sizes, and found that this formulation can solve graphs with up to 10 branches and 1 depot within a couple of seconds. Jumping to 11 branches and 1 depot added 4 minutes to the solve time. Adding more branches to the graph would increase the solve time by increasingly large margins. The optimal solution achieved with this formulation on a set of 11 branches and 1 depot is shown in figure (5.4). The optimal objective value for this solution is 301.

4.2.2 Set Partitioning with Column Generation

The set partitioning approach was also able to converge to an optimal objective of 303 on the same 11 node subset as the flow formulation. The set partitioning problem was slower than the flow formulation for this trial, taking about a 5 minutes to converge on an optimal solution. The optimal solution found using the set partitioning formulation is shown in figure (5.5). Notice that it is NOT the same as the flow formulation. This is because the pricing problem was unable to find a route with a negative reduced cost with the Gurobi model parameters set as they were. In the future, adjusting the time limit and heuristics parameters may yield better solutions in more or less time.

While the flow formulation performed better on the smaller set, again, the set partitioning formulation was more well equipped to solve the full graph. After an hour the set partitioning formulation had found a solution with an objective value of 5991, using 70 different pattern combinations to do so. The solution can be seen in figure (5.6). The optimal objective for the problem it was solving is unknown. However, we believe it to be much lower, within the 2500 to 4000 range based on the distribution of nodes on the graph. It is interesting to note that the solution no longer exhibits geographical separation, but overlap between the arcs of the routes occur.

Since the full model had 100 nodes, it took a long time for the column generation method to find routes with a negative reduced cost. We ran the program 2 times with different parameter settings on the Time limit and heuristics parameters, and each time it converged to an objective value of 5991. We assume that at this point the time needed to solve exceeded the scope of our project, and accept 5991 as a good enough solution. Depending on the time and resource constraints we face in a real world situation, we can adjust whether or not we would continue to run the column generation procedure to generate a better solution.

5 Appendix

5.1 Graphs

The following are the graphs of the outputs referenced in the above sections.

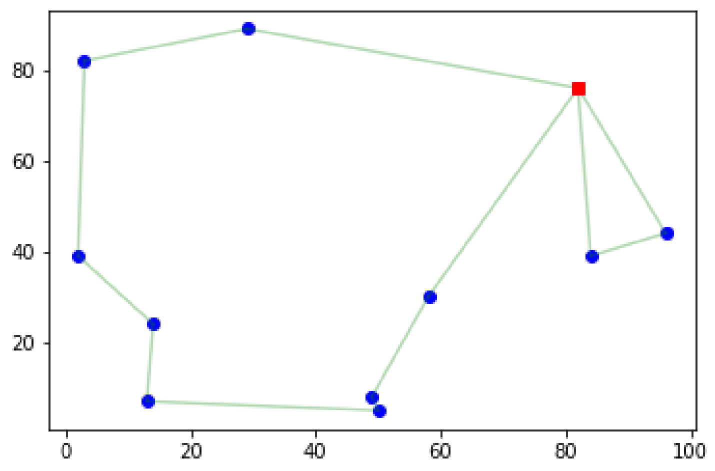

Figure 5.1: CVRP 10 Node Flow Solution

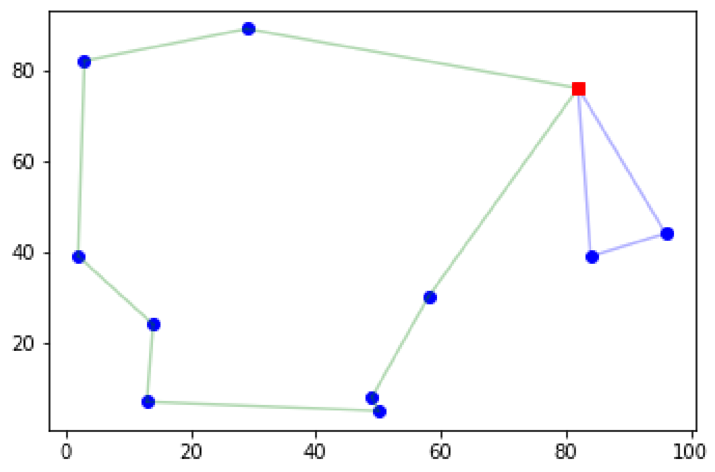

Figure 5.2: CVRP 10 Node Set Partitioning Solution

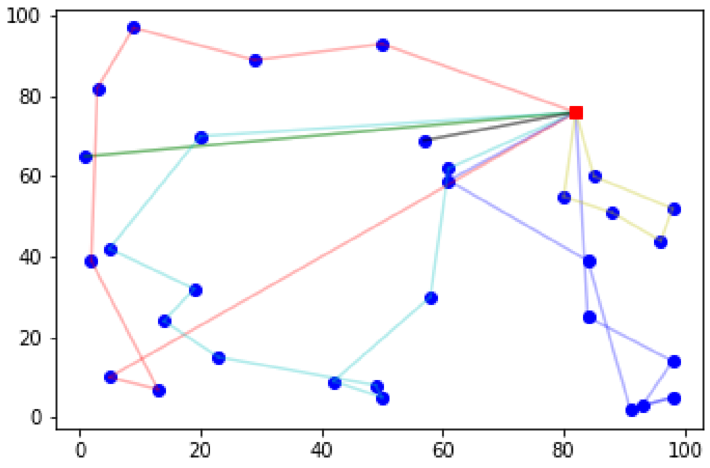

Figure 5.3: CVRP 32 Node Set Partitioning Solution

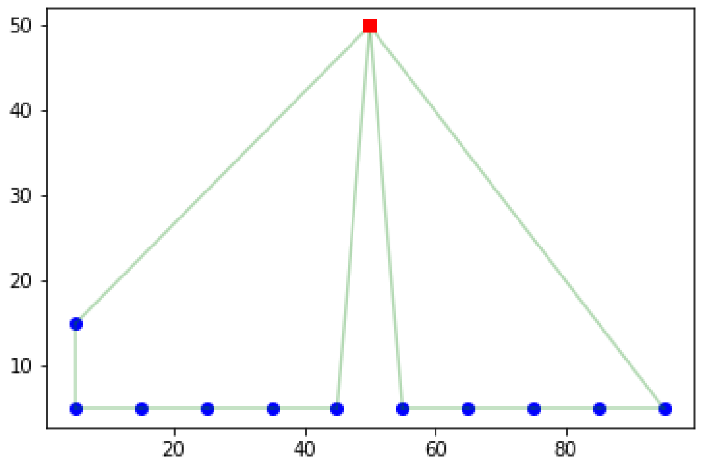

Figure 5.4: VRPTW 11 Node Flow Solution

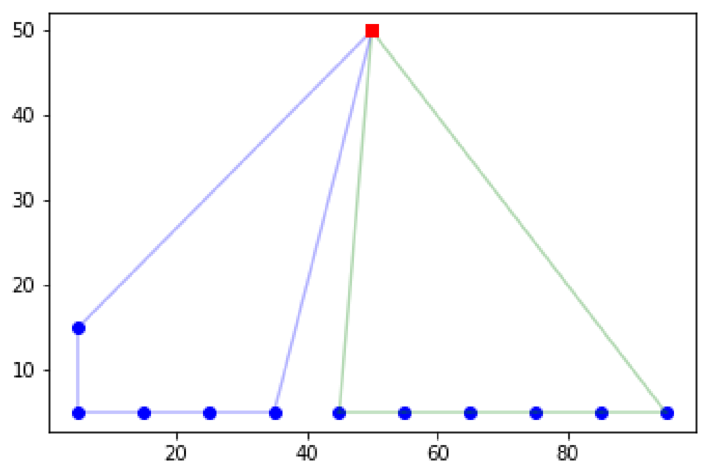

Figure 5.5: VRPTW 11 Node Set Partitioning Solution

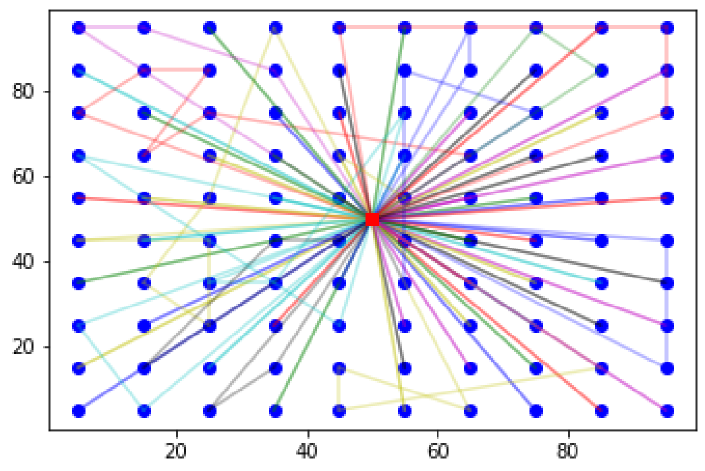

Figure 5.6: VRPTW 100 Node Set Partitioning Solution

5.2 Code

The code is submitted in a separate set of files along with this report.

References

[1] M. Goetschalckx. *Supply Chain Engineering.* Springer, New York, 2011.