Eotvos Lorand University

Applied Mathematics

Vehicle Routing Problem With Time Windows

Name:
Sevinj Bayramova

Supervisor:
Atilla Kiss

December 8, 2020

1 Introduction

Vehicle routing problems (VRP) aim to find an optimal set of routes to be used by a fleet of vehicles to visit a set of distributed customers.The classical definition of the VRP assumes that identical vehicles initially located at a depot are to deliver discrete quantities of goods to the customers.The aim is to minimize the overall cost of the route. The solution of the classical VRP problem is a set of routes which all begin and end in the depot, and which satisfies the constraint that all the customers are visited only once and all the customers must be assigned to vehicles such that the vehicle capacities are not exceeded.

The vehicle routing problem with time windows (VRPTW) is the extension of the well-known Vehicle Routing Problem. Here the service at each customer should start within a given time window and the vehicle should remain at the customer during the service. There are two types of time windows for this problem: soft and hard time windows. Time windows can be violated at a cost in soft time windows where hard time windows cannot be violated. In the latter case, a vehicle is not allowed to arrive at a customer after the latest time to begin service and if it arrives before the starting time of the time window the vehicle should wait until the customer is ready. In this case we will consider hard time window case. School bus routing, postal deliveries , bank deliveries can be real-life examples of the VRPTW.

The VRPTW is an NP hard problem. Thus, optimal solutions can be got for only small instances. For the large instances, heuristic approaches might be applied. But a heuristic approach cannot guarantee optimal solution every time

2 Background

The VRPTW is defined as follows: Let G =(N, A) be a directed graph that represents the road network. Arcs in A represent road segments, and N is the set of nodes. With each arc (i, j) A is associated a travel time t_{ij} and a travel cost c_{ij}. Let node 0 represent the depot location. Travel cost is generally taken as distance between customers i and j.A central depot where routes

are originating and terminating is denoted by d.Each customer is associated a demand q_{ij} and each service can begin at T_{ij} within the given time window a_i and b_j, earliest and latest time respectively. All the customers must be assigned to vehicles such that the vehicle capacities, Q, are not exceeded. Furthermore, if a vehicle arrives at a customer too early, it will wait.

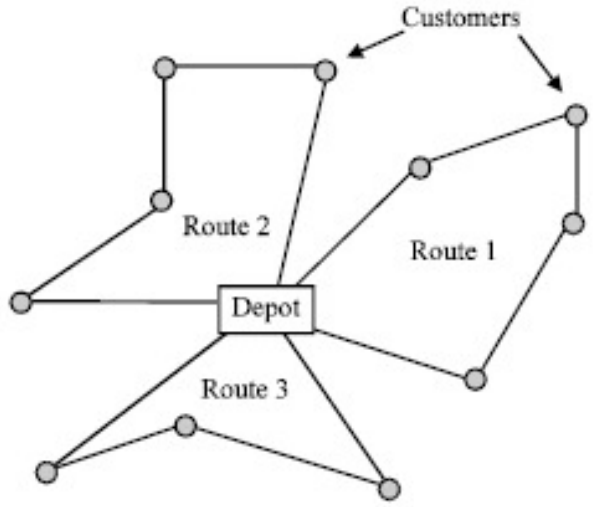

Figure 1: Vehicle Routing Problem With Time Windows

The LP relaxation of the set partitioning problem can be solved by column generation. Feasible columns are added as needed by solving a subproblem using dynamic programming. The solution obtained generally provides an excellent lower bound that is embedded in a branch-and-bound algorithm to solve the integer set partitioning problem

Dynamic programming is both a mathematical optimisation method and a computer programming method. It reduces time complexities from exponential to polynomial. It is splitting a problem into series of smaller sub-problems, solving each sub-problem and storing the solutions to each of these sub-problems in an array. The idea is to store-remember the results of subproblems.So when we need them later , we do not have to re-compute. Subproblems are smaller versions of the whole problem. Once we add solu-

tions of subproblems, we get total solution for the original problem. So, the core idea of Dynamic Programming is to avoid repeated work by remembering partial results.

One example for dynamic programming can be an algorithm for calculating the famous Fibonacci Numbers. The Fibonacci numbers, commonly denoted F_n, form a sequence, called the Fibonacci sequence, such that each number is the sum of the two preceding ones, starting from 0 and 1. That is, $F_0 = 0 \quad and \quad F_1 = 1$

$$F_n = F_{n-1} + F_{n-2} \quad for \quad n > 1$$

The beginning of the sequence is thus:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34...$$

In programming, a code for Fibonacci numbers will be as follow by using recursion.

```
def fib(n)
if i< 2
    return n
else
    return fib(n-1) + fib(n-2);
```

But if we use Dynamic programming with **memoization**

```
def fib(n):
  fib[0]=1, fib[1]=1
  for i in range (2,n+1):
        fib[i]=fib[i-1]+fib[i-2]
  return fib[n]
```

In the first code, a lot of values are calculating again at each step which is not practical. But when we use memoization we can get rid of recalculations. Memoization is a technique of storing solutions of already solved subproblems.

3 Solution Methods

The set partitioning problem tries to find routes with minimum costs in VRPWT. Minimal cost can be defined as total distance or (and) the number of vehicles used.Here we will take cost as a distance between node i and j. The set partitioning problem must satisfy some constraints mentioned above.

$$\begin{aligned} &\textbf{min} \sum_{r \in R} c_r \; x_r \\ &\sum_{r \in R} \delta_{ir} \; x_r = 1, i \in N \backslash \{d\} \\ &x_r \in \; \{0,1\}, r \in \; R \end{aligned} \tag{1}$$

Where R is the set of feasible routes,N is the set of nodes or customers, d is a central depot that vehicles are originating and terminating. δ_{ir} is constant that takes 1 if route r visits customer $i \in N \backslash \; \{d\}$ and 0 otherwise. $x_r$is the cost of route r which is defined as the sum of the costs of arcs of the route.

The branch-and-bound (B&B) algorithm is a fundamental and widely-used methodology for producing exact solutions to NP-hard optimization problems (i.e Vehicle Routing Problem). The basic idea here is enumeration all possible solutions of the problem by partitioning the total set of feasible solution space into smaller subsets of solutions in a tree structure. Those solution spaces which are not optimal are pruned off iteratively. The iteration has three main components: selection of the node to process, bound calculation, and branching.This process end when the whole tree is explored and the optimal solution is found.

Branch and Price algorithm is a generalization of LP based on Branch and Bound algorithm specially for solving difficult integer problems .The branch-and-price algorithm begins with a restricted **master problem** at the root node of the search tree that includes only a small subset of the total set of possible columns. At each node, the optimality of the LP solution is checked by solving the pricing problem. The pricing problem is used to determine whether or not there exist columns with profitable reduced cost that are missing from the current formulation. Upon the addition of columns,

the LP relaxation of the master problem is reoptimized. The algorithm iterates between solving the LP relaxation and solving the pricing problem until there are no more profitable columns to be added to the master problem. At this point, the algorithm proceeds as does the branch-and-bound algorithm. Branching occurs when no profitable columns are found.

How to identify the necessary columns to add to the model throughout the tree? In an LP model, the only columns that can improve a basic solution are ones with profitable reduced cost.Thus, in order to prove LP optimality, it is enough to find at least one column with profitable reduced cost or show that none exist. Because there are a very large number of columns , this cannot be done by enumeration.The process of finding columns to add to the existing integer program is called **column generation**.

4 Conclusion

The set partitioning problem is impractical while dealing with a huge number of variables since enumerating all related columns is difficult. That is why using column generation with branch-and-bound, which leads to a branch-and-price algorithm is far more effective. In general, For solving combinatorial optimization problems like the Vehicle Routing Problem (VRP) and its variants, the branch-and-price algorithms are considered as the most effective methods, among the exact algorithms.

5 Future Plans

In the continuation of this project, I will implement a pricing procedure and a column generation procedure which might be a part of a branch and price algorithm later.

References

[1] Martin Desrochers; Jacques Desrosiers; Marius Solomon A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows

Operations Research, Vol. 40, No. 2. (Mar. - Apr., 1992), pp. 342-354..

[2] Ben Ticha H, Absi N, Feillet D, Quilliot A, Van Woensel T. A branch-and-price algorithm for the vehicle routing problem with time windows on a road network. *Networks. 2018;1â17..*

[3] David R. Morrisona, Sheldon H. Jacobsonb, Jason J. Sauppe c, Edward C. Sewelld *Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning*

[4] Cynthia Barnhart Ellis L Johnson George L Nemhauser Martin WP Savelsbergh Pamela H Vance *BranchandPrice Column Generation for Solving Huge Integer Programs*