Vehicle Routing for Food Rescue Programs: A Comparison of Different Approaches

Canan Gunes, Willem-Jan van Hoeve, and Sridhar Tayur
Tepper School of Business, Carnegie Mellon University

1 Introduction

The 1-Commodity Pickup and Delivery Vehicle Routing Problem (1-PDVRP) asks to deliver a single commodity from a set of supply nodes to a set of demand nodes, which are unpaired. That is, a demand node can be served by any supply node. In this paper, we further assume that the supply and demand is unsplitable, which implies that we can visit each node only once. The 1-PDVRP arises in several practical contexts, ranging from bike-sharing programs in which bikes at each station need to be redistributed at various points in time, to food rescue programs in which excess food is collected from, e.g., restaurants and schools, and redistributed through agencies to people in need. The latter application is the main motivation of our study.

Pickup and delivery vehicle routing problems have been studied extensively; see, e.g., [1] for a recent survey. However, the 1-commodity pickup and delivery vehicle routing problem (1-PDVRP) has received limited attention. When only one vehicle is considered, the problem can be regarded as a traveling salesman problem, or 1-PDTSP. For the 1-PDTSP, different solution methods have been proposed, including [3,4]. On the other hand, the only paper that addresses the 1-PDVRP is by [2], to the best of our knowledge. [2] present different approaches, including MIP, CP and Local Search, which are applied to instances involving up to nine locations.

The main goal of this work is to compare off-the-shelf solution methods for the 1-PDVRP, using state-of-the-art solvers. In particular, how many vehicles, and how many locations, can still be handled (optimally) by these methods? The secondary goal of this work is to evaluate the potential (cost) savings in the context of food rescue programs. We note that the approaches we consider (MIP, CP, CBLS) are similar in spirit to those of [2]. Our MIP model is quite different, however. Further, although the CP and CBLS models are based on the same modeling concepts, the underlying solver technology has been greatly improved over the years.

2 Different Approaches to the 1-PDVRP

2.1 Input Data and Parameters

Let the set $V$ denote the set of locations, and let $O \in V$ denote the origin (or depot) from which the vehicles depart and return. With each location $i \in V$ we
associate a number $q_i \in \mathbb{R}$ representing the quantity to be picked up ($q_i > 0$) or delivered ($q_i < 0$) at $i$. The distance between two locations $i$ and $j$ in $V$ will be denoted by $d_{ij}$. Distance can be represented by length or time units.

Let $T$ denote the set of vehicles (or trucks). For simplicity, we assume that all vehicles have an equal ‘volume’ capacity $Q$ of the same unit as the quantities $q$ to be picked up (e.g., pounds). In addition, all vehicles are assumed to have an equal ‘horizon’ capacity $H$ of the same unit as the distances $d$.

### 2.2 Mixed Integer Programming

Our MIP model is based on column generation. The master problem of our column generation procedure consists of a set of ‘columns’ $S$ representing feasible routes. The routes are encoded as binary vectors on the index set $V$ of locations; that is, the actual order of the route is implicitly encoded. The columns are assumed to be grouped together in a matrix $A$ of size $V$ by $S$. The length of the routes is represented by a ‘cost’ vector $c \in \mathbb{R}^{|S|}$. We let $z \in \{0,1\}^{|S|}$ be a vector of binary variables representing the selected routes. The master problem can then be encoded as the following set covering model:

$$\begin{align*}
\min & \quad c^T z \\
\text{s.t.} & \quad Az = 1
\end{align*} \tag{1}$$

For our column generation procedure, we will actually solve the continuous relaxation of (1), which allows us to use the shadow prices corresponding to the constraints. We let $\lambda_j$ denote the shadow price of constraint $j$ in (1), where $j \in V$.

The subproblem for generating new feasible routes uses a model that employs a flow-based representation on a layered graph, where each layer consists of nodes representing all locations. The new route comprises $M$ steps, where each step represents the next location to be visited. We can safely assume that $M$ is the minimum of $|V| + 1$ and (an estimate on) the maximum number of locations that ‘fit’ in the horizon $H$ for each vehicle.

We let $x_{ijk}$ be a binary variable that represents whether we travel from location $i$ to location $j$ in step $k$. We further let $y_j$ be a binary variable representing whether we visit location $j$ at any time step. The vector of variables $y$ will represent the column to be generated. Further, variable $I_k$ represents the inventory of the vehicle, while variable $D_k$ represents the total distance traveled up to step $k$, where $k = 0, \ldots, M$. We let $D_0 = 0$, while $0 \leq I_0 \leq Q$. The problem of finding an improving route can then be modeled as presented in Figure 1.

In this model, the first four sets of constraints ensure that we leave from and finish at the origin. The fifth set of constraints enforce that we can enter the origin at any time, but not leave it again. The sixth set of constraints model the flow conservation at each node, while the seventh set of constraints (the first set in the right column) prevent the route from visiting a location more than once. The following four sets of constraints represent the capacity constraints of the vehicle in terms of quantities picked up and delivered, and in terms of