Routing and resource allocation in non-profit settings with equity and efficiency measures under demand uncertainty

Faisal Alkaabneh[a], Karmel S. Shehadeh[b], Ali Diabat[c,d,*]

[a] *Industrial & Systems Engineering, North Carolina A&T State University, Greensboro, NC 27401, United States*
[b] *Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA*
[c] *Division of Engineering, New York University Abu Dhabi, Saadiyat Island, Abu Dhabi 129188, United Arab Emirates*
[d] *Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201, United States*

Abstract

Motivated by food distribution operations for non-profit organizations, we study a variant of the stochastic routing-allocation problem under demand uncertainty, in which one decides the assignment of trucks for demand nodes, the sequence of demand nodes to visit (i.e., truck route), and the allocation of food supply to each demand node. We propose three stochastic mixed-integer programming (SMIP) models representing different performance measures important to food banks, namely maximizing efficiency, maximizing equity, and maximizing efficiency and equity simultaneously. To solve practical large-scale instances, we develop an original matheuristic based on adaptive large-scale neighborhood search. Using real-world data based on real-life instances, we conduct an extensive numerical experiment to assess the computational performance of our approach and derive insights relevant to food banks. The proposed matheuristic produces high-quality solutions quickly with an optimality gap never exceeding 4.11% on tested instances. We also demonstrate the performance of the three models in terms of service levels, food waste, and equity.

Keywords: Resource allocation; vehicle routing; food banks; humanitarian logistics; matheuristic

1. Introduction

With the recent COVID-19 pandemic and its double whammy on both the economy and healthcare sectors, an increasing number of people globally are at risk of food insecurity[1] due to the economic

*Corresponding author.
Email addresses: fmalkaabneh@ncat.edu (Faisal Alkaabneh), kas720@lehigh.edu (Karmel S. Shehadeh), diabat@nyu.edu (Ali Diabat)

[1] Food insecurity is defined as the disruption of food intake or eating patterns because of lack of money and other resources

recession (Nagurney 2021). Moreover, current political conflicts (e.g., RussiaUkraine war) have triggered a significant additional disruption in the global food supply chains and put more pressure on food security after the COVID-19 impact (Jagtap et al. 2022). Feeding America estimates that 42 million people in the United States (i.e., 1 of 8) including 13 million children (1 in 6), may experience food insecurity in 2021. These numbers represent an increase of 19.3% compared with 2019 numbers (in 2019 there were 35.2 million people in the US who were food insecure).

Food banks (i.e., non-profit organizations that primarily aim to distribute food items to people in need of food assistance) have long been the primary means to meditate food insecurity in the developed countries such as the US. Food banks provide several programs to combat food insecurity where each program has a unique channel and means of providing food items to the beneficiaries. For example, the Mobile Food Pantry Program (MFPP) is dedicated to serving populations located in remote areas or people with limited mobility, such as seniors. The MFPP is a truck loaded with food items that visit multiple locations to distribute food items to beneficiaries. We use the term "*demand nodes*," to denote the locations where beneficiaries receive food items following a predefined schedule.

MFPP managers often seek three primary logistical and operational decisions: (1) the assignment of trucks to serve demand nodes, (2) the sequence of visits to demand nodes assigned to each truck (i.e., vehicle routing), and (3) how much to allocate to each node. The former two decisions are vehicle routing problem, and the latter decision is a stochastic resource allocation problem under demand uncertainty. The truck route should not exceed the driver's work hour limitations. A primary consideration during the allocation process is equity in allocation for the different nodes while delivering overall as much food as possible (i.e., avoid food waste). In particular, food banks are interested in maximizing equity and efficiency rather than maximizing profit or minimizing cost. *Equity* refers to the status where all nodes receive similar levels of service and no node is at a disadvantage. On the other hand, *efficiency* within the context of MFPP refers to allocating as much food items as possible and minimize food waste at the end of distribution operations. Unfortunately, it is well-known that equity and efficiency are often conflicting objectives in many real-world application and within the context of MFPP in particular. For instance, allocating no food items to all nodes creates perfect equity since there is no variation in the level of service. But, on the other hand, no allocation implies that we waste large quantities of food, resulting in a total loss of efficiency.

Our work aims at developing and solving several mathematical models that help food banks run their MFPP more efficiently while maintaining equity. This paper is specifically inspired by the MFPP of the Food Bank of the Southern Tier (FBST) located in Elmira-NY, which serves more than 80 nodes across six counties. The ultimate goal of this project is to optimize the routing of MFPP trucks and food supplies allocation to demand nodes within the service network of the FBST program to best account for the variability of the demand and ensure maximizing equity and minimizing food waste (i.e., maximize efficiency).

We consider a multi-vehicle routing and resource allocation problem under demand uncertainty with equity and efficiency objectives in non-profit settings. Existing models for this problem have considered deterministic demand or focused on one aspect of the problem, often maximizing equity with a few studies focusing on the efficiency dimension. We propose a data-driven two-stage stochastic-mixed-integer programming (SMIP) models that decide on the optimal first-stage routing and allocation decisions under various performance measures important to food bank managers, namely maximizing efficiency, maximizing equity, and maximizing efficiency and equity simultaneously. We summarize our contributions as follow:

- To the best of our knowledge, and according to our literature review, see Section 2, our paper is the first to propose and analyze SMIPs for integrated stochastic routing and resource allocation for the MFPP.
- Due to the challenges of solving large instances with the SMIP, we propose a matheuristic based on an adaptive large-scale neighborhood search to obtain near-optimal solutions for large-scale instances. This matheuristic leverages the special structure of the problem. It combines a constructive heuristic, which provides an initial feasible solution in short computational times, an improvement heuristic, and solving a mixed-integer programming model.
- Using real-world data obtained from our partners at the Food Bank of the Southern Tier located in Elmira-NY, we conduct an extensive numerical experiment to assess the computational performance of our approach and derive insights into the MFPP. Our results demonstrate the trade-off between service level, efficiency, and equity. In addition, our matheuristic can solve large practical instances with an optimality gap never exceeding 4.11% within two hours for large instances, thereby enabling practitioners to use our approach.

2. Literature Review

Studies on food banks' operations are more recent in the OR literature. The first paper presenting operations research tools for food banks is the study by Bartholdi III et al. (1983). Bartholdi III et al. (1983) developed a decision support tool to solve a logistic distribution operation related to the *Meals on Wheels* program which is essentially a vehicle routing problem. Since then, there has been more attention by the OR community to study various food banks' operations such as resource allocation (Alkaabneh et al. (2021)), location-routing (Johnson et al. (2002), Ghoniem et al. (2013)), inventory managing (Davis et al. (2014)), gleaning operations (Lee et al. (2017)), food supply chain network design (Besik et al. 2022), etc.

In this work we focus on the MFPP which encompasses two major operations, the vehicle routing and the resource allocation in stochastic settings. Studies in the operations research literature considering routing allocation problems are not new (see, e.g., Kumar et al. (1995), Bassok and Ernst (1995), Berman and Larson (2001)). Nonetheless, these studies are dedicated to commercial settings where the goal is to maximize distributor's profit or minimize costs due to routing and penalty costs associated with unsatisfied demand. However, the MFPP is a service provided by the food banks to the public; therefore, it is necessary to have different perspectives on the problem. Savas (1978) argues that any public service program should have fundamental policy questions related to how well a service is performed. More precisely, Savas (1978) states that three measures - efficiency and equity - are the key indices of performance for public services. In light of such considerations, the performance of MFPP as a public service requires more focus on efficiency and equity as performance measures rather than cost and profit.

Several studies in the literature examine the problem of resource allocation/inventory management in social settings where the goal is to achieve equitable and efficient distribution of donated supplies. Motivated by the sequential resource allocation at the MFPP where the sequence of stops is fixed (i.e., the order of nodes to be visited is known upfront and not part of the planning process), Lien et al. (2014) study the sequential resource allocation with an objective function that aims to maximize the expected minimum filling ratio among a set of demand nodes. In their work, a sequence of nodes are served by a single vehicle while the order of visits is fixed and the decision is how much to allocate at each node upon arrival and revealing the actual demand. Sinclair et al. (2020) propose a simple policy that aims to "match" the ex-post fair allocation among nodes using the current available resources and "predicted" histogram of future demand. Their algorithm is simultaneously Pareto-efficient and envy-free. Orgut et al. (2016, 2018) present robust optimization models to support food banks in their inventory (resource) allocation decisions with the objective of maximizing total food distribution (i.e., efficiency) while equity is enforced as a constraint in their model. Fianu and Davis (2018) develop a model to assist food banks in distributing food supplies which they receive from random donations with the aim of maximizing equity in distribution. In their model the supplies are considered to be stochastic while demand is deterministic.

Other studies in the literature that consider the integration of routing and resource allocation are limited. To the best of our knowledge, only three studies in the literature consider the integrated problem of resource allocation/collection and routing in an integrated fashion while explicitly considering social performance measure (i.e., equity and efficiency). Eisenhandler and Tzur (2018) study "deterministic" (i.e., food supplies and demand are known in advance) problem of food rescue (food collection and distribution) as a routing resource allocation problem with the objective function of maximizing equity and efficiency simultaneously. They present an exact solution method and a large neighborhood search heuristic approach to solve large-scale instances. Eisenhandler and Tzur (2019) present a novel formulation and a powerful matheuristic that combines solving a mathematical model and a heuristic to solve rich humanitarian logistic problems including the food rescue problem at food banks. Balcik et al. (2014) extends the work of Lien et al. (2014) for the multi-vehicle version and their model incorporates the decision of vehicle routing in addition to the resource allocation decisions. It is worth mentioning that the work of Eisenhandler and Tzur (2018) and Eisenhandler and Tzur (2019) assume that the supplies to be collected and the demand to be satisfied are deterministic; on the other hand, only Balcik et al. (2014) consider random demand at the nodes which is more realistic and closer to the actual food banks operations. Our work is substantially different than Lien et al. (2014) since our model considers the multi-vehicle stochastic routing and allocation unlike the work of Lien et al. (2014) which considers only a single vehicle stochastic routing and allocation. Balcik et al. (2014) studies multi-vehicle stochastic routing allocation problem in non-profit setting focusing on maximizing equity and minimizing food waste using efficient decomposition-based heuristic; on the other hand, our work provides a two-stage integrated model. More recently, some studies focus on empirical data to analyze the preference of food banks' managers in terms of the trade-offs between equity, efficiency, and effectiveness, see Islam and Ivy (2022), Zoha et al. (2022), Hasnain et al. (2021). Hasnain et al. (2021) develop a multi-criteria optimization model that provides the decision-maker the flexibility to capture their preferences over the three criteria of equity, effectiveness, and efficiency, and explore the resulting trade-offs.

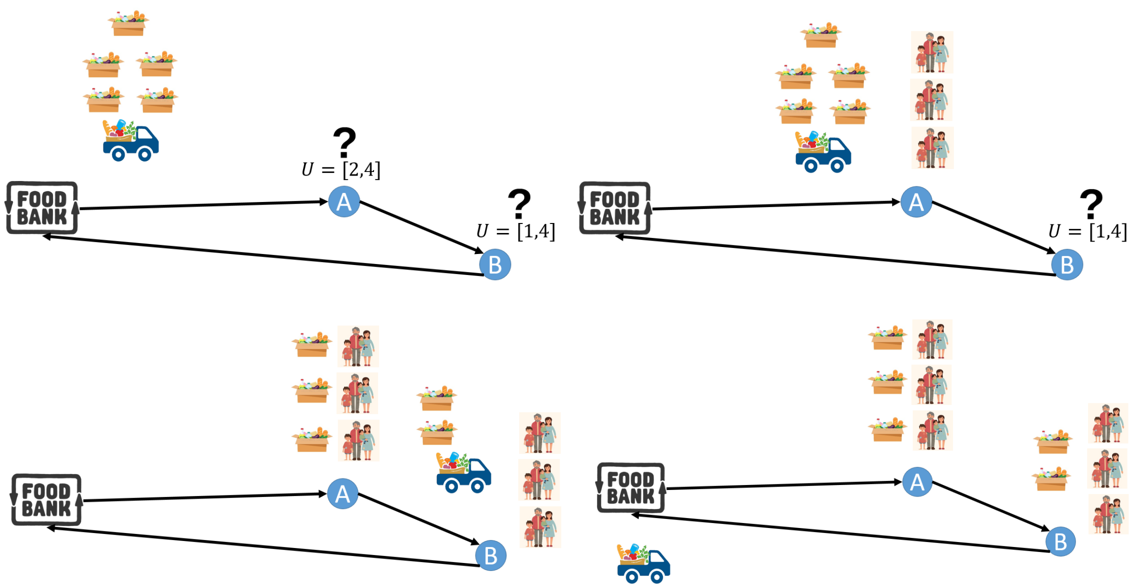


Figure 1: Example of MFPP operation.

To the best of our knowledge, solving large-scale instances of routing resource allocation problem under stochastic demand with an objective function maximizing equity and efficiency simultaneously nor a model with an objective function that maximizes efficiency and modeling equity explicitly have not been addressed before. Our study makes three main contributions. First, we introduce stochastic routing and resource allocation models to solve the MFPP under different objective functions. Second, we design a matheuristic solution procedures for this problem that utilizes certain properties and structure of its subproblem. Numerical experiments on several real-life and randomly generated data sets confirm that our matheuristic obtains high-quality solutions within short computational time. Third, we highlight the societal impact and policy insights of our work based on different performance measures that food banks care about.

3. Definitions and Assumptions

In this section, we formally define our stochastic MFPP problem and provide an illustrative example of MFPP operations (see Section 3.1). The MFPP operation is essentially a truck loaded with food items visiting demand nodes following a particular order, and at each stop, volunteers distribute food items to recipients. The location of these demand nodes is fixed and they are usually a church, a parking lot, or a public space. Food bank managers need to communicate the pickup locations and time schedule of each visit to the recipients well in-advance. The food bank publishes the pick-up times at the beginning of each year and these information are shared with the local community. One day before the distribution, the food bank coordinator decides on the number of items (food units) to load the truck given a certain set of nodes that will be visited by a truck. On the day of distribution, the volunteers load the truck with the number of units the coordinator decided. The truck starts visiting the nodes following the order of visits, and volunteers distribute food items to beneficiaries at each stop. The number of people showing up at a certain node (demand henceforth) is often random and the volunteers have full control on how many units of supplies to provide at a certain node as long as they don't over serve[2]. A shortage occurs when the node's realized demand exceeds the units pre-allocated to this node. On the other hand, extra food goes to waste if there are excess food items (i.e., the realized demand is less than pre-allocated units), which is a significant loss for a food bank operating under limited supplies. In this paper, we allow using the excess supplies from a node to satisfy the demand of any successive node within that route.

In this paper, we consider a stochastic MFPP problem. Specifically, we consider a food bank facility (node) with a set of K trucks that must serve a set of demand nodes, and accordingly, define MFPP problem on a directed service network (graph) $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A}$ is the edge set. The vertex 0 is the food bank node, and the vertices in the set $\mathcal{V}' = \mathcal{V} \backslash \{0\}$ are the demand nodes. The traveling time between two nodes i and j is $t_{i,j}$, which represent the duration of traveling along edge $(i, j) \in \mathcal{A}$. We do not assume that the traveling time matrix is symmetric. The truck driver need to get back to the food bank within L time units. The food bank have U units of supplies, and each truck k have a capacity of Q_k. The demand, d_i, of each node $i \in \mathcal{V}$ is random.

Given a set of demand nodes, we aim to find: (1) assignment of nodes to trucks, (2) a routing plan for each truck, i.e., a route that specify the movement of the truck, and (3) allocation of food items, i.e., amount of supply to pre-allocate for each node in a route. The objective is to maximize the filling rate across all nodes. As in practice, we aim to satisfy as much as demand as possible while maintaining equity among nodes. When the volunteers decide on how many units to supply a certain node, the following constraints are imposed:

- The total number of units allocated is less than or equal the recommended allocation by Feeding America.

[2] Over service refers to the case where an individual receive more units (pounds of food) than the number of pounds recommended by Feeding America.

- Equity should be maintained among different nodes implying that no node should be at a disadvantage while other nodes enjoy full demand satisfaction.

In non-profit operations, such as the MFPP, maintaining equity among recipients is very critical aspect. In the context of MFPP, *equity* refers to maintaining similar service rate among different nodes and no node is at a disadvantage. That implies that the filling ratio of all nodes should be similar, or close, to each other. The *filling ratio*, or service level, is the total allocated quantity over total demand. As in prior literature, use the deviation between the maximum filling ratio and the minimum filling ratio as a measure of *inequity.* This inequity can be controlled via the inequity parameter which we denote by α. The parameter α takes values between 0 and 1, in case α is set to 0, we have a perfect equity and setting $\alpha > 0$ imply certain degree of inequity. Perfect equity occurs when all filling ratios are the same across demand nodes.

3.1. Mobile Food Pantry Program Operation

In this section, we provide a small example to illustrate MFPP operation. Consider an MFFP with one truck and two stops (A and B). Suppose that the demand at the first stop follows a uniform distribution with a lower bound of 2 and an upper bound of 4 (i.e., $U = [2, 4]$). Moreover, suppose that the demand at the second stop follows a uniform distribution with a lower bound of 1 and an upper bound of 4 (i.e., $U = [1, 4]$), and assume that the food truck leaves the food bank with 5 units of food. Thus, the demand at nodes A and B are unknown at the planning stage, i.e., the actual value of the demand will be observed once the driver arrives at each stop (see Figure 1 part (a)). Suppose the driver's route starts at the food bank, visits A, then B before returning to the food bank. In addition, suppose that the driver observes 3 demand units at stop A (see Figure 1 part (b)). At this point, s/he needs to decide how many units to distribute. If they distribute 3 units, the filling ratio is 1 (i.e., amount distributed equals demand). Moreover, the driver leaves stop A and move to stop B with the remaining two food units in this case. If the demand at stop B is 3 units (see Figure 1 part (C)), then the driver can only distribute the remaining 2 units to B, and the filling ratio of B is 0.67 (i.e., distributed quantity is smaller than demand). Finally, the driver returns to the food bank (see Figure 1 part (d)). Note that, in this example, the food truck returned to the food bank with zero units. In other words, this feasible allocation is efficient because all the available supplies were distributed. However, this allocation is inequitable because stops A and B filling ratios are 1.0 and 0.67, respectively, leading to an inequity (highest filling ratio - lowest filling ratio) of 0.33.

4. Stochastic Optimization Model

In this section, we formulate a two-stage SMILP model for the MFPP. In the first (planning) stage, we determine (1) the assignment of nodes to trucks, (2) a routing plan for each truck, and (3) the amount of supply to pre-allocate for each node in the route of each truck. In the second stage (i.e., recourse problem), after observing the demand, we compute the filling ratio and excess at each node for each demand realization.

First, let us introduce the variables, parameters, and functions defining our first-stage SP model. We define a binary variable x_{ij}^k that is equal to 1 if and only if food truck k travels along edge (i, j). We let y_i^k be a binary decision variable indicating if node i is served by vehicle k. Let non-negative continuous decision variable q_i^k represents the amount of supply pre-allocated to node i to be delivered via vehicle k, for all i and k. Due to demand uncertainty, the actual demand at each node i may be smaller, equal, or larger than pre-allocated supplies to each node.

For each realization of the demand, we define the following second-stage variables. We define non-negative continuous decision variable f_i as the filling ratio at site i. In addition, we define a non-negative continuous decision variable w_i^k to represent the excess of supplies available on truck k after leaving at site i. Note that the excess supplies from a node can be used to satisfy the demand of any successive site within that route. In other words, in addition to the pre-allocated supply to each node, volunteers may distribute food items carried from previous nodes to satisfy the demand as much as possible. Table 1 summarizes all notation. Using these notation, we formulate the following SMILP:

$$\max_{x,y,q} \quad \mathbb{E}_{\xi}[Q(x, q, y, \xi)] \tag{1a}$$

$$\text{s.t.} \sum_{j \in \mathcal{V}, i \neq j} x_{ij}^k - \sum_{j \in \mathcal{V}, i \neq j} x_{ji}^k = 0, \quad i \in \mathcal{V}, k \in \mathcal{K} \tag{1b}$$

$$\sum_{i \in \mathcal{V}, i \neq j} x_{ij}^k = y_j^k, \quad j \in \mathcal{V}, k \in \mathcal{K} \tag{1c}$$

$$\sum_{i \in \mathcal{V}} x_{0i}^k \leq y_0^k, \quad k \in \mathcal{K} \tag{1d}$$

$$\sum_{k \in \mathcal{K}} y_i^k = 1, \quad i \in \mathcal{V}' \tag{1e}$$

$$\sum_{(i,j) \in \mathcal{A}} t_{ij} x_{ij}^k \leq L, \quad k \in \mathcal{K} \tag{1f}$$

$$\sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}: j \neq i} x_{ij}^k \leq |\mathcal{S}| - 1, \quad k \in \mathcal{K}, \mathcal{S} \subset \mathcal{S} \subset N : 2 \leq \mathcal{S} \leq N - 1 \tag{1g}$$

Table 1: Notation

Indices	
i, j	Node indices
k	Vehicle index
Parameters	
N	Number of demand nodes in the network excluding the food bank
U	Available supplies
t_{ij}	travelling time between nodes i and j
ρ	fraction of available excess in the food truck that can be allocated at a certain node $(0, 1]$
Q_k	Capacity of vehicle k
L	The maximal vehicle travel time
α	Inequity parameter
Scenario-dependent parameters	
d_i	Realized demand of node i
sa_i	Node's i share in the total demand of all nodes ($sa_i = \frac{d_i}{\sum_{j\in\mathcal{V}'} d_j}$)
ξ	Random vector containing scenario-dependent parameters $\xi = (d_1, d_2, ..., d_N)$
Scenario-independent (first stage) variables	
x_{ij}^k	binary variable that is equal to 1 if and only if vehicle k travels along edge (i, j)
y_i^k	binary variable that is equal to 1 if and only if vehicle k visits site i
q_i^k	continuous variable to denote the quantity of food distributed at site i via vehicle k
Scenario-dependent (second-stage) variables	
f_i	filling ratio of node i
γ	minimum filling ratio
w_i^k	excess of food available in truck k upon visiting site i

$$\sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}} q_i^k \le U, \tag{1h}$$

$$\sum_{i\in\mathcal{V}'} q_i^k \le Q_k y_0^k, \forall k \in \mathcal{K} \tag{1i}$$

$$x_{ij}^k, y_i^k \in \{0,1\}\ q_i^k \ge 0, \quad i, j \in \mathcal{V}, k \in \mathcal{K} \tag{1j}$$

where for a feasible $(x, y, q) \in \mathcal{X} := \{(1\text{b}) - (1\text{j})\}$ and a realization of $\xi := [d_1, \ldots, d_{|\mathcal{V}|}]^\top$:

$$Q(x, y, q, \xi) := \min -\sum_{i\in\mathcal{V}'} f_i \tag{2a}$$

$$\text{s.t. } q_i^k - d_i + w_j^k - M(1 - x_{ji}^k) \le w_i^k, \qquad \forall i \in \mathcal{V}', j \in \mathcal{V}, k \in \mathcal{K} : i \ne j \tag{2b}$$

$$(1-\rho)w_j^k - M(1 - x_{ji}^k) \le w_i^k, \qquad \forall i \in \mathcal{V}', j \in \mathcal{V}, k \in \mathcal{K} : i \ne j \tag{2c}$$

$$f_i \le \frac{\sum_{k\in\mathcal{K}}(q_i^k + w_j^k - w_i^k) + M(1 - \sum_{k\in\mathcal{K}} x_{ji}^k)}{d_i}, \qquad \forall i \in \mathcal{V}', j \in \mathcal{V} : i \ne j \tag{2d}$$

$$f_i \le 1, \qquad \forall i \in \mathcal{V}' \tag{2e}$$

$$|f_i - f_j| \le \alpha, \qquad \forall i, j \in \mathcal{V}' \tag{2f}$$

$$f_i, w_i^k \ge 0, \qquad \forall i \in \mathcal{V}, k \in \mathcal{K} \tag{2g}$$

The objective function (2a) maximizes the summation of filling ratios to achieve the largest efficiency. Constraints (1b) require that each node is arrived at from exactly one other node if visited by vehicle k. Constraints (1c) require that if node j is served by vehicle k then vehicle k visits that node. Constraints (1d) enforce that a vehicle can depart the food bank only if it is used. Constraints (1e) ensure that each node i is served by a vehicle. Constraints (1f) enforce the length of the tour of vehicle k to be less than or equal to the time limit by which the driver and the volunteers need to complete their working day. Constraints (1g) are sub-tour elimination constraints. However, for instances of realistic sizes, the number of sub-tour elimination constraints (1g) is too large to allow full enumeration and these must be dynamically generated throughout the search process. Specifically, each time the model is solved, we run minimum s-t cut algorithm on the resulting graph to detect sub-tours and add the constraints accordingly. Constraints (1h) ensure that total allocation does not exceed total supplies available and (1i) ensure that the vehicle load does not exceed its capacity.

For each scenario, constraints (2b) and (2c) calculate the remaining food excess after visiting node i and performing the distribution process upon observing the demand. Note that the food excess available on the truck can be used in part or as a whole to satisfy the demand of a node if the observed demand exceeds the pre-allocated quantity in stage 1. Parameter $0 \leq \rho \leq 1$ captures the allowed quantity to be allocated. If ρ is set to 0, then food excess from previous nodes cannot be reused for later allocations. On the other hand, if ρ is set to 1, then food excess from previous nodes should be reused completely until the observed demand is fully satisfied or there are no more excess available for allocation. Lastly, note that due to different cases of difference between supplies at stage 1 and demand and the different cases of the value of w_j, this leads to the need of having constraints (2b) and (2c), see Table 2 for more details. Constraints (2d) calculate the filling ratio of each node. Constraints (2e) ensure that distributed quantity does not exceed observed demand. Note that the waste or excess in supplies at one node is returned to the truck and can be used as available supplies at the next node, this is captured via decision variable w_i $\forall i \in \mathcal{V}'$ where $w_0^k = 0$. Constraints (2f) ensure that the deviation between maximum and minimum filling ratios does not exceed the predefined inequity threshold α. Finally, (1j) and (2g) specify feasible ranges of the decision variables. Model (1a)-(2g) aims at maximizing efficiency measured as total quantity allocated to all demand nodes while equity is maintained as a constraint, model (1a)-(2g) is called M_Eff.

We also derive models to account for different perspectives by modifying model (1a)-(2g). We next discuss these models. Lien et al. (2014) and Balcik et al. (2014) present models that focus on maximizing the minimum filling ratio (i.e., maximizing equity). To derive such model under

Table 2: Different cases for calculating w_i^k

Case	(d_i vs q_i^k)	w_j^k	w_i^k
1. A	$d_i > q_i^k$	positive	$\max\{(1-\rho)w_j^k, w_j^k - d_i + q_i^k\}$
1. B	$d_i > q_i^k$	0	0
2. A	$d_i < q_i^k$	positive	$q_i^k - d_i + w_j$
2. B	$d_i < q_i^k$	0	$q_i^k - d_i$
3. A	$d_i = q_i^k$	positive	w_j
3. B	$d_i = q_i^k$	0	0

uncertainty, we relax constraints (2f) and replace the objective function (2a) by:

$$Q(x, y, q, \gamma, \xi) := \min -\gamma \tag{3}$$

and add the following constraint:

$$\gamma \leq f_i, \qquad \forall i \in \mathcal{V}' \tag{4}$$

We call this model M_Eq as it aims at maximizing expected equity.

Lastly, the model presented in Eisenhandler and Tzur (2018) and Eisenhandler and Tzur (2019) aims at maximizing equity and efficiency simultaneously. Efficiency (i.e., total allocated quantity) can be calculated as $\sum_{i \in \mathcal{V}'} d_i f_i$, and the equity can be measured using the following linear constraints based on Anand (1983) and Mandell (1991):

$$E_{ij} \geq sa_j d_i f_i - sa_i d_j f_j \qquad \forall i, j \in \mathcal{V}' : i < j \tag{5a}$$

$$E_{ij} \geq sa_i d_j f_j - sa_j d_i f_i. \qquad \forall i, j \in \mathcal{V}' : i < j \tag{5b}$$

Constraints (5) become part of the second stage model, constraints (2f) are relaxed, and objective function (2a) is replaced by:

$$Q(x, y, q, \gamma, \xi) := \min -\sum_{i \in \mathcal{V}'} d_i f_i + \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}' : i < j} E_{ij} \tag{6}$$

We call this model M_Eq_M_Eff as it aims at maximizing expected efficiency and equity simultaneously. Note that the proposed models (i.e., M_Eff, M_Eq, and M_Eq_M_Eff) are challenging to solve primarily due to the following reasons

- If we ignore the stochastic demand and assume that the demand at each location is deterministic, and if we relax the equity constraints, the resulting problem is a Vehicle Routing Problem (VRP). VRP is a well-known problem to be NP-hard; hence, our problem is NP-hard.

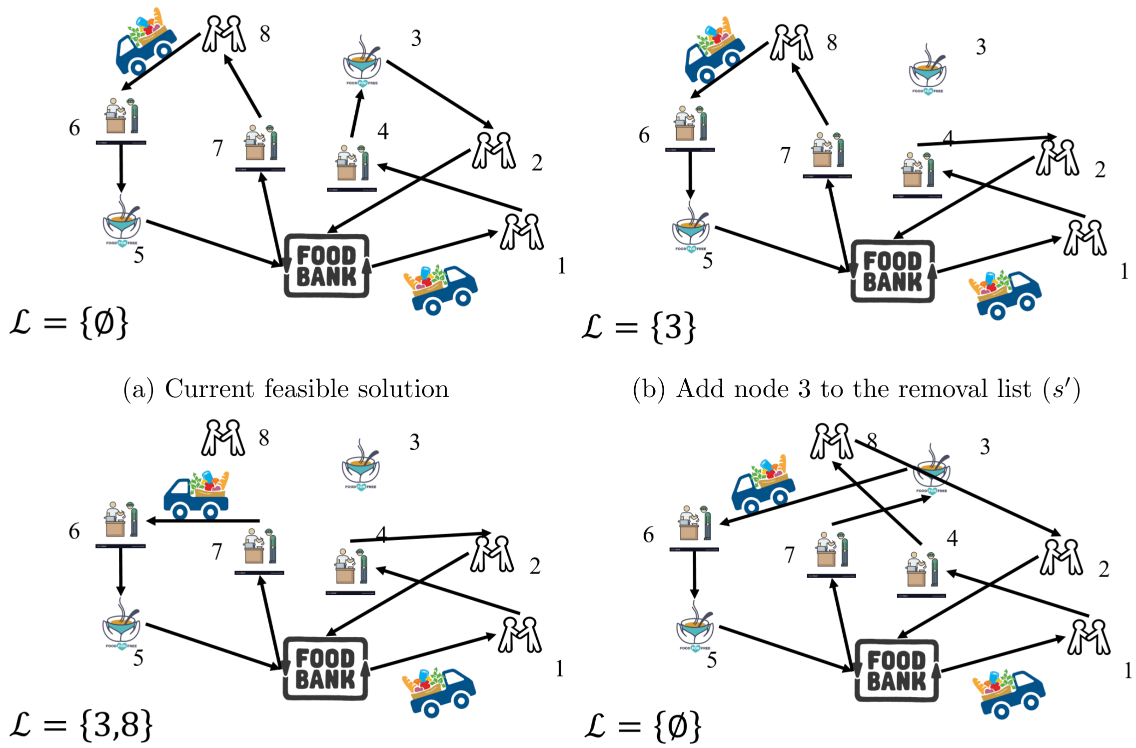
- The number of binary variables in our formulation is a function of the instance size. Specifically, the number of binary decision variables is $O(KN^2)$, where N is the number of nodes in the network, and K is the number of vehicles. Thus, the number of binary variables could grow exponentially as the size of the instance grows. It is well-known that such large-scale integer programs are challenging to solve.

5. Matheuristic for the Mobile Food Pantry Program problem

Due to the complexity of the underlying routing and stochastic allocation problems, the SMILP can solve only small instances to optimality and fail to solve large-scale real-life instances in a reasonable time. Therefore, in this section, we develop a matheuristic that finds high-quality solutions within a reasonable time. To develop this matheuristic, we leverage the following structural properties of the problem. The MFPP seeks three main decisions that can be composed into three stages, the first stage is to partition the nodes into vehicles, the second stage aims to find the optimal sequence of nodes for each vehicle route, and the third stage aims to find the optimal resource allocation across the nodes. The optimization problem at the third stage is a stochastic LP for fixed nodes partitioning and vehicle routes that is easy to solve efficiently using a standard solver.

We use the above structure to develop a three-stage matheuristic based on the Adaptive Large Neighborhood Search (ALNS) framework proposed Ropke and Pisinger (2006). Unlike the Large Neighborhood Search (LNS) framework developed by Shaw (1998), the ALNS updates the weights of the operators in an adaptive fashion based on their performance. ALNS is a suitable meta-heuristic for the MFPP problem as it demonstrates excellent performance in solving hard VRP problems under different variations, see Ribeiro and Laporte (2012), Adulyasak et al. (2014), Mjirda et al. (2014). At a high level, ALNS implements a mechanism to search within the neighbor space of a solution by calling a destroy operator, followed by a repair operator. At each iteration during the search, a certain destroy operator "destroys" part of the solution, which is later "repaired" by a repair operator. Upon evaluating the objective function value of the new repaired solution, the algorithm updates the weights of the destroy and repair operators based on the solution quality of the new repaired solution.

A pseudocode of the ALNS is shown in appendix Appendix A. An initial feasible solution s is generated by means of a construction heuristic, we describe our construction heuristic in Section 5.1. In our work, destroy operators essentially remove a node (or a set of nodes) from its (their)



(a) Current feasible solution

(b) Add node 3 to the removal list (s')

(c) Add node 8 to the removal list (s')

(d) Insert nodes 3 and 8 into new routes (s_{new})

Figure 2: Depiction of destroy and repair operators.

current route while repair operators insert a removed node into a new route. Figure 2 illustrates the high-level ideas behind the developed destroy and repair operators. The destroy operators are detailed in Section 5.2, the repair operators are described in Section 5.3, Section 5.4 describes the acceptance criteria of a new solution, and lastly operators' weight updating is provided in Section 5.5.

In our implementation of the ALNS heuristic, each solution s is an assignment of nodes to trucks and trucks routing, and $f(s)$ is the objective function value of the optimal allocation corresponding to the solution s found by solving an LP by fixing the values of all binary variables. The main motivation for an ALNS-based search heuristic for the MFPP is the fact that each iteration consists of solving a simple LP efficiently and hence the ALNS can explore large number of random neighbors. Furthermore, by solving model (1a)-(2g) to optimality for fixed routing variables while exploring a large number of neighbors, our implementation demonstrates the diversification and intensification components which are fundamental elements for meta-heuristic optimization (Glover and Samorani 2019).

5.1. Construction of an initial feasible solution

We construct the initial solution for the MFPP problem sequentially by considering one demand node at each iteration and try to find the best route to insert the selected node at each iteration until all nodes are inserted within a route. Note that any demand node can be characterized by its location (address), demand mean (μ_i), demand variance (σ_i), and nominal allocation. Nominal allocation refers to the quantity that a node receives upon solving a resource allocation problem to decide the fair share of each demand node given the trucks capacity, total supplies available for allocation, and assuming perfect equity allocation (i.e., a setting with $\alpha = 0$). The nominal allocation quantity for each node can be simply calculated as follows:

$$na_i = \min\{Q, \frac{U}{\sum_{j=1}^{N} \mu_j} * \mu_i\}, \ \forall i \in \mathcal{N}$$

Calculating the nominal allocation helps in the initial construction of routes as it provides a better idea of the truck capacity utilization. On the other hand, a route can be characterized by the set of nodes it serves, remaining capacity of the truck (given how much allocation each node receives), and the total traveling time. Our construction algorithm makes use of all these information when deciding which node to consider next and what is the best truck to assign for that node. More specifically, our algorithm calculates a score for each route when considering a node to insert as:

$$Score_{i,k} = -\delta_1 distance_{i,k} + \delta_2 rc_k + \delta_3 \frac{\sum_{j \in R(k)} \sigma_j + \sigma_i}{\sum_{j \in \mathcal{N}} \sigma_j - \sum_{j \in R(k)} \sigma_j - \sigma_i}$$

where δ_1 is the weight of importance for the distance measure, $distance_{i,k}$ is the distance from the centroid of truck's k route and node i. Note that assigning nodes to trucks does not constitute a route yet and therefore the centroid of truck's k route is assumed to be in the center of all nodes included in that route including the point of the food bank itself. The reasoning behind including the distance criteria within the weight score is due to the fact that constraints (1f) need to be satisfied. The second term incorporates the remaining capacity into $score_{i,k}$ where δ_2 is the weight of importance for the remaining capacity of truck k, rc_k is the remaining capacity at truck k. The third term accounts for variation of demand of nodes trying to spread the variance of nodes across different routes instead of packing nodes with high demand variance within the same route. A selected node i is then inserted in route k that yields the highest score. The process is repeated until all nodes are assigned trucks.

Once all nodes are assigned routes, our construction heuristic finds the sequence of nodes within each route by solving a small instance of the MFPP for one vehicle, setting $U_k = 0.8 * \sum_{i \in k} na_i$.

Our preliminary analysis suggests that the sequence of nodes within a route is fairly robust to variation in supplies as long as the supplies are less than the total demand. Therefore, we make use of this observation to find the optimal sequencing within each route independently.

The last step of our construction heuristic is to find the optimal allocation of resources to each demand node by solving the MFPP problem (1a)-(2g) with fixed routing variables. The construction heuristic returns a solution s that is feasible to all constraints in (1a)-(2g).

5.2. Destroy Operators

In this section, we present the six destroy operators that we use in our algorithm. We use $d \in \mathcal{D}$ to denote a destroy operator within the set of destroying operators $\mathcal{D}$. Some of these operators appears in the VRP literature implementing ALNS heuristic, see Ropke and Pisinger (2006), Pisinger and Ropke (2007), Demir et al. (2012), Alkaabneh et al. (2020), Alkaabneh and Diabat (2022). A destroy operator in our ALNS design is essentially a function that takes the location of each node within each route and removes a set of nodes from their locations, implying that these nodes are not visited by the food truck anymore. Therefore, a destroyed solution is no longer feasible; hence, a repair operator is needed to restore feasibility. The criteria a destroy operator follows in deciding which nodes to remove follows the definition of that destroy operator as detailed in our main manuscript. For the MFPP problem it is natural to design a destroy phase where nodes are removed from routes; hence, the destroy phase consist of a series of operations where selected nodes are removed from their routes and added to a removal list $\mathcal{L}$ as illustrated in Figure 2. We now describe the destroy operators used in our implementation:

1. Random selection: this operator starts with an empty removal list. It randomly removes p nodes from the solution. The idea of randomly selecting nodes helps diversify the search mechanism. In our implementation, the number of nodes to remove is determined using a semi-triangular distribution with a negative slope, which favors removing a small number of nodes. More specifically, p is an integer random number drawn from the interval $[1, 0.2 * |N|]$.
2. Largest demand mean removal: this operator iteratively removes nodes which belong to a predefined set of nodes with high demand mean.
3. Largest demand variance removal: this operator, similar to the Largest demand mean removal operator, iteratively removes nodes which belong to a predefined set of nodes with high demand variance.

4. Route removal: This operator removes all nodes within a selected route from the solution and add them to list $\mathcal{L}$. It randomly selects a route from the set of routes in the solution.
5. Historical knowledge node removal: this operator is inspired by the the neighbor graph removal operator used in Ropke and Pisinger (2006) and the historical knowledge node removal operator developed in Demir et al. (2012). This operator keeps a record of the food excess of node i, defined as difference between q_i^k and observed d_i, and calculated as $hr_i = (\sum_k q_i^k - d_i)^+$ at any iteration. At any iteration of the ALNS, the algorithm keeps track of the minimum excess for each node i denoted as hr_i^*. The historical knowledge operator, then, selects a node that has the highest deviation from its minimum food excess as $i^* = \arg\max_{i \in \mathcal{V}'} hr_i - hr_i^*$.
6. Zone removal: The zone removal operator removes a set of nodes from their route based on their location on the map. This operator picks a point on the map and inserts all nodes within a predefined radius from the selected point into the removal list.

5.3. Repair operators

We are now ready to present the repair operators that insert nodes from the removal list $\mathcal{L}$ until all nodes are inserted. A repair operator is a function that takes an infeasible solution violating any of the constraints (in our ALNS algorithm nodes that are not visited) and restores feasibility. Thus, a solution that is repaired is a feasible solution respecting all constraints. The criteria that a repair operator follows in deciding where each removed node is inserted follows the definition of that repair operator as detailed in our manuscript. We use $r \in \mathcal{R}$ to denote a repair operator from the set of repairing operators $\mathcal{R}$. Note that these repair operators insert the removed nodes back into existing routes if it is feasible with respect to the traveling time constraint. Each time a repair operator selects route k to insert a node from list $\mathcal{L}$ (say node i), we implement Algorithm 1 to check if adding node i to route k maintains feasible truck traveling time. Therefore, our implementation of ALNS to solve MFPP does not allow moving into a neighbor that is not feasible in terms of traveling time constraints. The main motivation of preserving feasibility during the search is due to the fact that a valid lower bound for the MFPP problem can be obtained simply by solving the problem for a single vehicle due to the pooling effect. That said, violating the truck capacity or tour length constraints will always lead to lower (i.e., better) objective function value and it might be difficult to restore feasibility in our setting.

We now briefly define the four insertion operators used in the ALNS algorithm:

- Greedy insertion: This operator repeatedly inserts a node in the best feasible route.

Algorithm 1: Repair operation

Input: A list of nodes served by vehicle k ($\mathbb{L}$), new node to be inserted i

Output: Feasibility check on inserting node i within route k (True or False)

Solve a TSP instance with an objective function to minimize the traveling time on $\mathbb{L} \cup i$

if *Total traveling time* $\leq L$ **then**

  Return True   // It is feasible to insert node i within route k

else

  Return False   // It is not feasible to insert node i within route k

- Smallest route insertion: This operator tries to insert a node in the smallest route according to the total truck load.
- Busiest route insertion: This operator tries to insert a node in one the busiest $\lceil 0.2 * K \rceil$ routes according to the total truck load.
- Random route insertion: This operator picks a route at random and tries to insert a node in that selected route.

5.4. Acceptance criteria of a new solution during the search

We use simulated annealing as a local search framework for our ALNS heuristic. Within simulated annealing, a new solution whose objective function value is worse than the current solution may be accepted with certain probability that is decreasing as the search moves forward. More specifically, if s^* denotes the best solution found during the search so far, $s_{current}$ is the current solution obtained at the beginning of an iteration, and s_{new} is a new feasible solution found at the end of an iteration upon implementing a destroy and a repair operators. The objective function value of a solution s is denoted as $f(s)$. A solution s_{new} is always accepted if $f(s_{new}) < f(s_{current})$ (for minimization problems), and accepted with a probability of $e^{\frac{f(s_{new}) - f(s_{current})}{Temp/iter}}$, where $Temp$ is the temperature. Dividing $Temp$ by the iteration counter ($iter$) serves as a cooling mechanism to reduce the temperature as the number of iterations moves forward.

5.5. Updating the weights

The last step at each iteration within the ALNS algorithm is to update the weights of the destroy and repair operators. Updating the weights of the operators gained success in addressing several

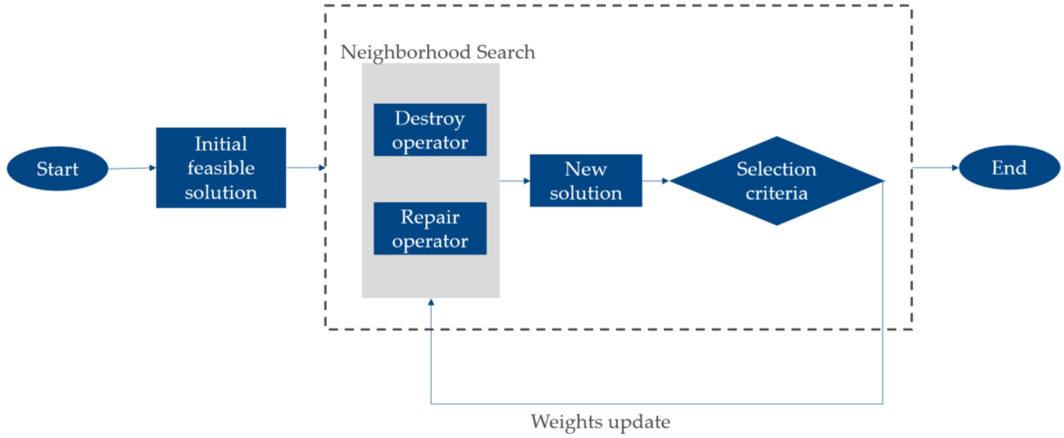


Figure 3: ALNS general framework.

complex routing and scheduling problems (e.g., Ropke and Pisinger (2006), Cordeau et al. (2010), Ruf and Cordeau (2021)). In this approach, if destroy operator d (repair operator r) was selected in iteration $iter$, then at the end of this iteration, a score η_d (η_r) is assigned to its performance. The formula for updating the value of η_d is as follows:

$$\eta_o^{iter-1} = \lambda \eta_o^{iter-1} + (1-\lambda)\psi$$

where $\lambda \in (0,1)$ is a smoothing parameter and ψ is the performance score of an operator and it is assigned as:

$$\psi = \begin{cases} \psi_1, & \text{if the new solution is better than the best solution found so far} \\ \psi_2, & \text{if the new solution is better than the current solution} \\ \psi_3, & \text{if the new solution is accepted} \\ \psi_4, & \text{if the new solution is rejected} \end{cases}$$

with $\psi_1 \geq \psi_2 \geq \psi_3 \geq \psi_4$, in our implementation we use $\psi_1 = 100, \psi_2 = 40, \psi_3 = 10$, and $\psi_4 = 1$ as in Cordeau et al. (2010). The initial weight η_o^0 at the first iteration is set to 1 for all operators. At the end of each iteration after updating the values of η, the probability assigned to each operation o is $\frac{\eta_o}{\sum_{o \in O} \eta_o}$. Figure 3 summarizes the main ALNS procedure.

5.6. Improvement Phase

Similar to the work of Archetti et al. (2017), the third phase of our matheuristic is based on using the information collected during the random search phase. Note that the complexity of the

MFPP reduces significantly if the clusters of demand nodes and routing decisions are established. As such, the third phase of our matheuristic focuses on exploring promising clusters of demand nodes and routes.

The idea of the improvement phase is to solve a new MFPP model where the decision variables are assignment of nodes to clusters (partitions) and resource allocation. Given a number of promising partitions, the new model seeks to find the best partitions and resource allocation for each node. The set of promising partitions are collected during the ALNS search. More specifically, each time there is an improvement in the current objective function or a new best global solution is found, see lines 8 and 13 of Algorithm 2 in appendix Appendix A, vehicle clusters/routes are saved as promising partitions of nodes. We introduce the following notation:

- $c \in \mathcal{C}$ the set of unique clusters collected during the ALNS execution;
- a_i^c is a parameter that takes value 1 if node $i \in \mathcal{V}'$ is served in cluster c;
- $\bar{x}_{ij}^c$ parameter equals to 1 if node i is visited before node j within cluster $c : a_i^c = a_j^c = 1$.

Note that we save the routing variables within each cluster during the search since routing and allocation variables are optimal with respect to fixed partition, see the discussion in Section 5.3. We formulate a MILP model called $findClust$ to select the best clusters and resource allocation. To this end, $findClust$ use the following variables in addition to f_i, E_{ij}, γ:

- y^c binary variable equals to 1 if and only if cluster c is selected;
- q_i^c continuous variable to denote the quantity of food distributed to node i within cluster $c : a_i^c = 1$;
- w_i^c continuous variable denoting excess of food upon visiting site i within cluster $c : a_i^c = 1$.

$$\max_{y,q} \quad \mathbb{E}_\xi[Q(q, y, \xi)] \tag{7a}$$

$$\text{s.t.} \sum_{c \in \mathcal{C}} a_i^c y^c = 1, \quad i \in \mathcal{V}' \tag{7b}$$

$$\sum_{c \in \mathcal{C}} y^c \leq K, \tag{7c}$$

$$\sum_{i \in \mathcal{V}'} \sum_{c \in \mathcal{C}} q_i^c \leq U, \tag{7d}$$

$$\sum_{i \in \mathcal{V}'} a_i^c q_i^c \leq Q y^c, \forall c \in \mathcal{C} \tag{7e}$$

$$y_i^c \in \{0,1\} \; q_i^c \geq 0, \quad i \in \mathcal{V}, c \in \mathcal{C} \tag{7f}$$

Constraints (7b) ensure that each node is served by one cluster only. Constraints (7c) enforce the number of selected clusters to be less than or equal to the number of vehicles available in service. Constraints (7d) ensure that total allocation does not exceed total supplies available and (7e) ensure that the vehicle load does not exceed its capacity. Constraints (7f) enforce that y^c variables are binary and q_i^c variables are continuous and positive. The remaining constraints have the same meaning as in the original formulation provided that x_{ij}^k are no longer binary variables, instead are replaced by $\bar{x}_{ij}^c$ and q_i^c replaces q_i^k in (2).

6. Lower Bound

Deriving a valid lower bound for the equity and efficiency maximization model can be obtained by relaxing constraints (5) and removing the second term (i.e., the equity part E_{ij}) from the objective leads to a new model that provides a lower bound for equity and efficiency maximization model. Note that constraints (1h) state that total allocated quantity never exceeds U; therefore, $-U$ is a valid lower bound for equity and efficiency maximization model. This is a much tighter bound than the LP relaxation bound. Likewise, for maximize efficiency model a valid lower bound can be obtained by relaxing constraints (2f) and (1f). In this case, the model becomes easier to solve.

7. Numerical study

In this section, we present several experiments to assess the performance of the proposed matheuristic and derive insights into the solutions of the different models. We use real-world data provided by the Food Bank of the Southern Tier (FBST), which is located in Elmira – New York, more information about the FBST data is provided in appendix Appendix B.

Note that the areas served by the FBST are mainly suburban communities with the exception of the city of Binghamton located in Broome county. Serving suburban communities implies that the network of nodes is very sparse and can be easily partitioned into groups of nodes based on spatial characteristics such as their county or proximity to the closest city. There are 21 demand nodes in Broome, 14 in Chemung, 4 in Schuyler, 24 in Steuben, 10 in Tioga, and 13 in Tompkins. We exclude Schuyler county from the analysis since it is a small county. Given that the truck driver and volunteers work at most 8 hours a day, it is impossible to visit more than 10 nodes per route since the service time at each node should be 45 minutes. Similar to Balcik et al. (2014) and Lien et al. (2014), we model the demand as Gamma random variable. Using historical data we obtained from the FBST, we compute the shape and scale of the demand distribution of each demand node. It is worth mentioning that some demand nodes, specifically senior homes, have small demand mean and low variance. On the other hand, other demand nodes have a large mean and high variance. For instance, 'Senior - Carpenter Apartments' demand node has a demand distribution of $Gamma(28.1, 1.05)$; on the other hand, demand node 'Feed Elmira - Hathorne Court' has a demand distribution of $Gamma(53.4, 3.40)$. Both demand nodes are located within the same county, highlighting that demand can be heterogeneous across different nodes even within the same county. In appendix Appendix B, we provide the list of all demand nodes, demand shape, demand scale, and longitude and latitude of each node.

We generate our testbed of problem instances using the following parameters:

- Number of demand nodes, $|\mathcal{V}'|$: 10, 13, 14, 21, 24, 30, 35.
- Value of α (the measure of inequity): 0.05, 0.10.
- Demand d_i: randomly selected integer following a gamma distribution.
- Traveling time is calculated using `Bing Maps Dev Center` data.
- Time spent at each demand node for allocation is 45 minutes including the setup, distribution, and packing.

We implemented a Monte Carlo Optimization Procedure to determine an appropriate sample size R to obtain near-optimal solutions to maximize efficiency model (1)-(2) based on its sample average approximation (SAA) (see Appendix C for details). All the code was written using MATLAB and we use Gurobi 9.1.1 as a solver. The runs were completed on a laptop computer with a 2.8 GHz Intel Core processor and 16 GB of RAM.

7.1. Comparison Between Models

In this section, we analyze and compare the solutions of the models presented in Section **??**. For illustrative purposes and brevity, we use Tioga county's data because this is a small county consisting of 10 nodes and the demand structure of the nodes within this county is very heterogeneous. For instance, Waverly demand node has a demand distribution of $Gamma(70.1, 2.75)$ while the Senior - Spring View Apartment demand node has a demand distribution of $Gamma(44.5, 0.62)$.

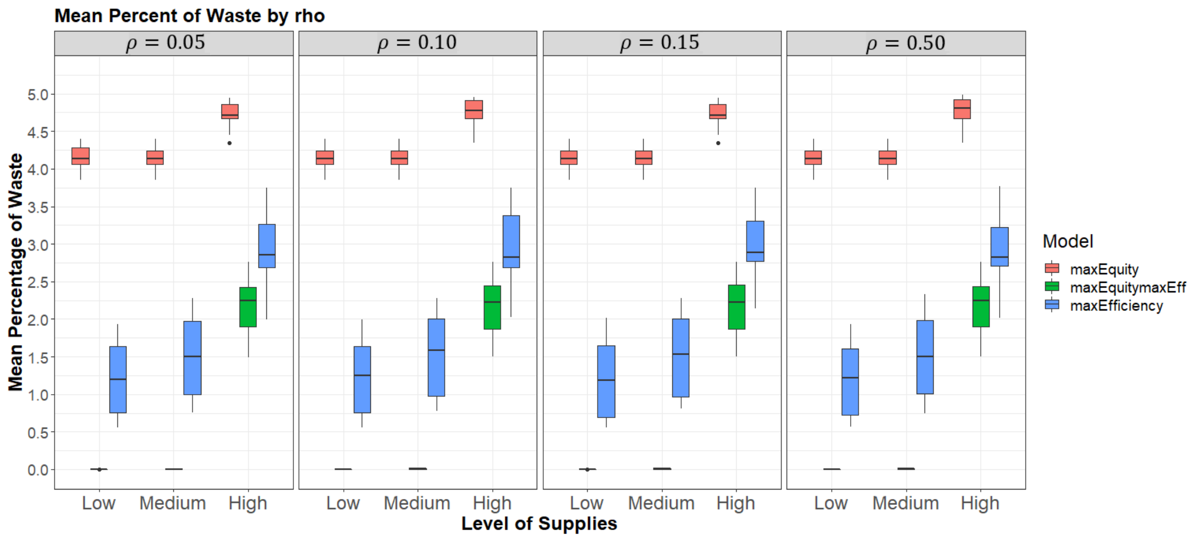


Figure 4: Comparison of the mean percentage of food waste for different values of ρ and levels of supply.

We consider four levels of $\rho = \{0.05, 0.10, 0.15, 0.50\}$, and three levels of total supply U={low, medium, high} relative to total demand. Low U equals 65% of the total demand mean, medium U equals 78% of the total demand mean, and high U equals the mean of total demand. For the M_Eff model, we consider two values for the inequity parameter α=0.05 and 0.10. For each combination of ρ, U, and α, we generate 10 data instances and use the same data to compare the models. We focus our analysis on three primary performance measures, namely equity and efficiency.

Figure 4 illustrates the percentage of food waste under different values of ρ and levels of supplies for the three models. Percentage of food waste is calculated as the difference between total supplies available for distribution (U) and total quantity allocated over total supplies available for distribution. We note that the M_Eq model yields the highest food waste percentage, and the M_Eq_M_Eff model yields the lowest food waste. These results are expected as M_Eq model focuses on maintaining similar filling ratio among the demand nodes by maximizing the minimum filling ratio at the cost of having food waste at the end of the trip. On the other hand, M_Eq_M_Eff model seeks an allocation that maximizes total allocated quantity while inequity is compromised at the expense of less value in the objective function, see the second term in equation (6).

Figure 5 illustrates the efficiency (i.e., the summation of the filling ratio of all nodes) under different values of ρ and levels of supplies for the three models. Clearly, M_Eq model yields the lowest efficiency and M_Eff yields the highest efficiency slightly better than M_Eq_M_Eff model. It is very interesting to see that M_Eq_M_Eff model achieves levels of efficiency that are very similar to maximize efficiency model but slightly less. This is due to the fact that M_Eq_M_Eff model balances equity and efficiency.

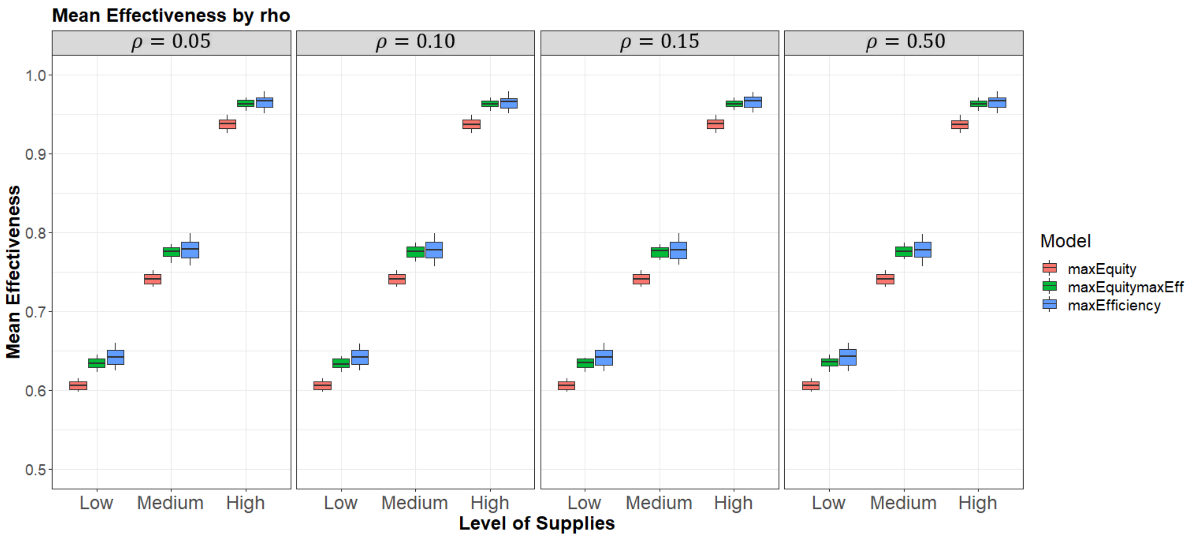


Figure 5: Comparison of the mean efficiency under different values of ρ and levels of supply.

Figure 6 illustrates the inequity under different values of ρ and levels of supplies for the three models. Inequity is calculated as the deviation between the node with the highest filling ratio and the node with the lowest filling ratio. From this figure, we observe that M_Eq model yields the best performance (zero inequity), M_Eq_M_Eff model yields excellent performance with inequity being less than 2.5% without explicitly defining that threshold, while the inequity under M_Eff model is always under the values of 0.05 and 0.10 as specified in the model. An interesting observation is that the inequity behavior of M_Eq_M_Eff model is not monotone with the levels of supplies but monotone under the other models. This is an interesting aspect to notice that maintaining equity is not always easier with an increase in supplies

Lastly, for completeness, we compare the mean traveled distance yielded by each model. Figure 7 displays the average traveled distance by the delivery trucks under different values of ρ and levels of supplies for the three models. We do not notice substantial differences between the traveled distance by the different models. Later in Section 7.3 we perform some sensitivity analysis on the traveling and service times to see how models may provide different results.

7.2. Computational Performance of the Developed Matheuristic

In this section, we demonstrate the computational performance of the developed matheuristic in terms of computational time and optimality gap. Specifically, we evaluate the performance of our matheuristic with the following characteristics:

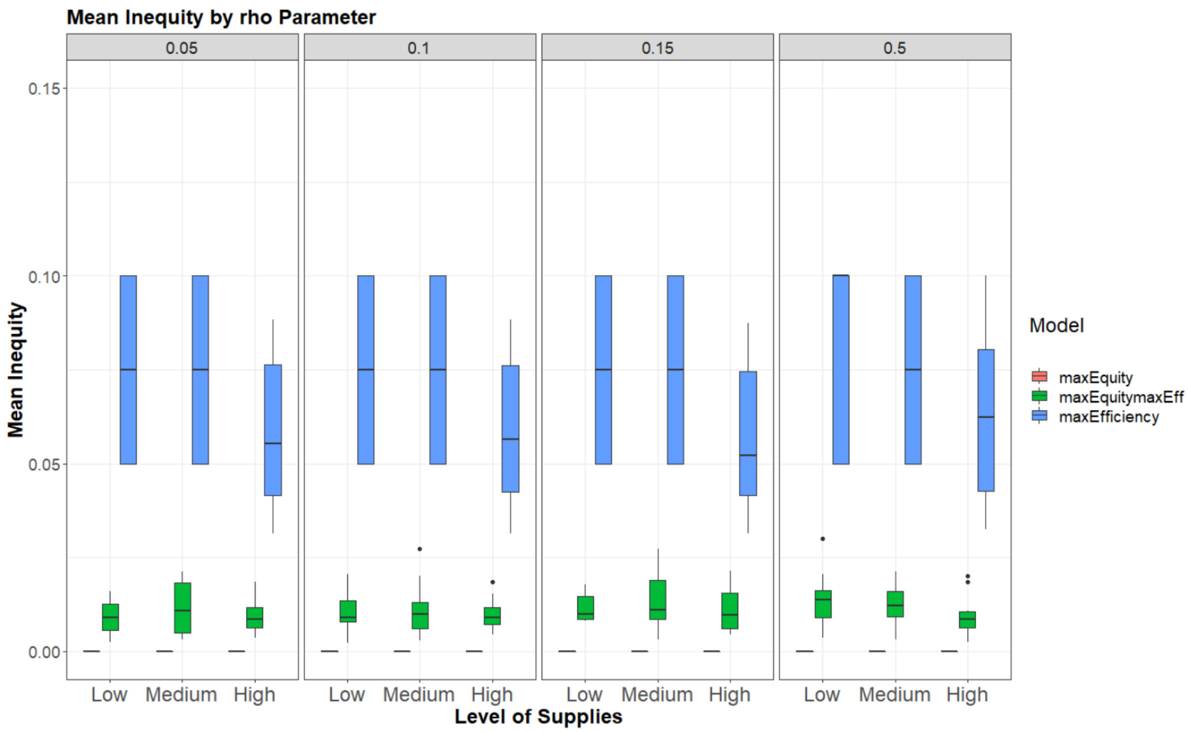


Figure 6: Comparison of the mean inequity under different values of ρ and levels of supplies

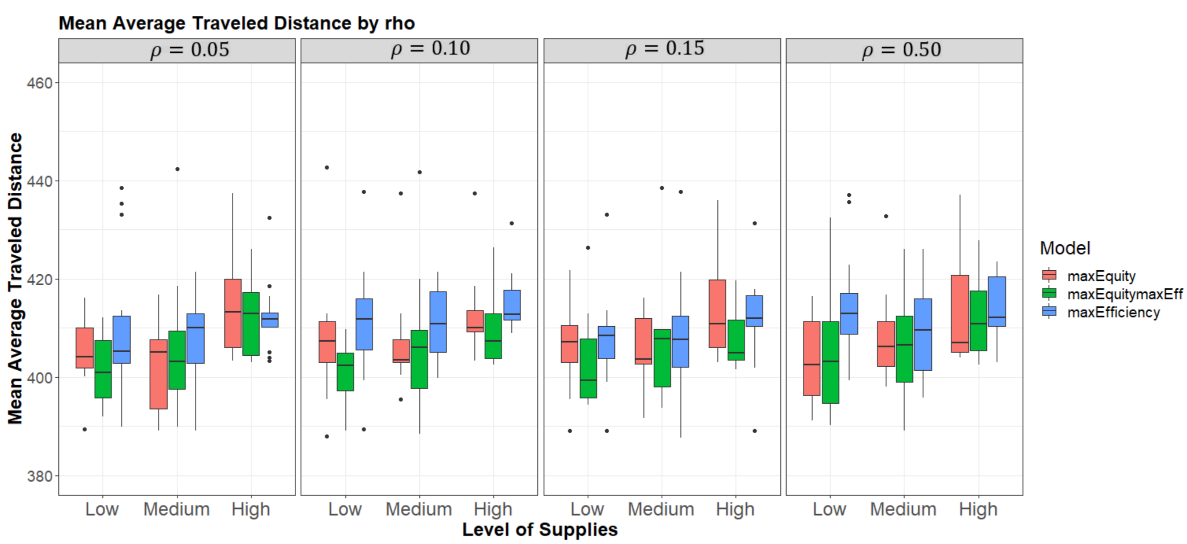


Figure 7: Comparison of the mean traveled distance under different values of ρ and levels of supplies

- The FBST data set, which consist of 5 relatively sparse networks with various sizes. Data and parameters of demand, traveling time, service time, and truck capacity are real-world data. For the total supplies (U), we follow the same procedure in Section of 7.1.

- Randomly generated instances of 30 and 35 nodes in a dense network. We randomly generate the coordinates of the nodes in a square with size 35 miles by 35 miles. The distribution of demand follows gamma distribution. We randomly divide the nodes into three groups, and demand distributions among nodes of the same group are identical. Three levels of node demand means ($\mu = \{75, 50, 25\}$) and coefficient of variance ($CV = \{1.5, 1.0, 0.5\}$) are considered: low (25,0.5), medium (50,1.0), and high (75,1.5).

Table 3 displays these computational results. We set $\rho = 0.5$ and $\alpha = 0.05$. Major columns 1, 2, and 3 show the number of demand nodes in the network, number of vehicles, and levels of supply (the setup of level of supplies follows the same procedure as illustrated in Section 7.1). Major columns 4, 5, and 6 display the computational results of each model. The first sub-column within these columns reports the computational time in seconds using the matheuristic, the second sub-column of these columns reports the negative[3] objective function value of solving the mathematical model of each model (denotes by z^{solver}), the third sub-column within these columns reports negative the objective function value (denoted by $z^{\text{MHeur.}}$), and lastly the fourth sub-column reports the gap with respect to z^{solver} (i.e., $gap = \frac{z^{\text{MHeur.}} - z^{\text{solver}}}{z^{\text{solver}}} * 100\%$). A negative gap implies that the solution obtained using the matheuristic is better than the solution found by the solver.

The computational time for solving the mathematical model is restricted to 60 minutes for all instances with $N < 20$ and 120 minutes for instances with $N \geq 20$. We repeat each experiment 5 times with newly generated data for demand, and we report on the average of the objective function and computational time. A gap value in bold font indicate that the gap is reported with respect to the lower bound discussed in Section 6 since the solver was not able to find a feasible solution within the specified computational time.

The computational results in Table 3 demonstrates the superior performance of the developed matheuristic in finding high quality solution for all models in shorter time that the solver. Specifically, the average gap ever exceeds 4.11% and on average the matheuristic finds better solutions than the one found by the solver in the fraction of time needed by the solver. More importantly, our matheuristic finds such solutions in considerably shorter computing times. The average computing time required by Gurobi is more than 5560 seconds. In contrast, our matheuristic takes 568 seconds for the equity model, 1047 seconds for the efficiency model, and 1753 seconds for the efficiency and equity model. Moreover, our matheuristic provides solutions to all instances, which is not the case

[3]Recall that all models are stated as minimization models for convenience but in reality the objectives should be maximize efficiency, maximize equity, and maximize efficiency and equity simultaneously

Table 3: Results of the Computational Results

N	K	*U*	Equity Model				Efficiency Model				Efficiency & Equity Model			
			Time	z^{solver}	$z^{\text{MHeur.}}$	*gap*	Time	z^{solver}	$z^{\text{MHeur.}}$	*gap*	Time	z^{solver}	$z^{\text{MHeur.}}$	*gap*
10	2	Low	89.30	0.63	0.64	-1.11	180.44	5.94	5.95	-0.12	168.77	500.40	500.15	0.05
10	2	Med	91.92	0.76	0.76	-1.05	181.37	7.09	7.10	-0.14	146.14	505.59	505.53	0.01
10	2	High	87.10	0.96	0.97	-0.51	155.42	8.78	8.80	-0.19	143.17	505.59	505.59	0.00
13	3	Low	299.14	0.63	0.62	0.19	514.46	7.21	7.26	-0.77	638.76	528.70	528.47	0.04
13	3	Med	280.74	0.75	0.75	0.13	444.61	8.61	8.58	0.35	612.47	634.78	634.65	0.02
13	3	High	263.81	0.96	0.96	0.13	373.77	10.76	10.74	0.20	395.74	799.21	798.21	0.12
14	3	Low	360.26	0.63	0.62	1.52	840.10	7.80	7.80	0.10	1194.28	517.85	517.92	-0.01
14	3	Med	360.07	0.76	0.75	1.39	896.90	9.33	9.32	0.06	1056.96	621.64	621.32	0.05
14	3	High	404.25	0.96	0.96	0.55	718.78	11.70	11.70	0.07	967.61	785.00	784.46	0.07
21	6	Low	244.72	0.60	0.60	0.51	429.72	11.99	11.99	0.01	1481.05	1148.71	1149.73	-0.09
21	6	Med	255.39	0.72	0.71	0.42	346.96	14.28	14.21	0.50	1326.51	1375.05	1379.36	-0.31
21	6	High	174.06	0.92	0.91	0.49	282.60	18.04	18.03	0.02	385.44	1742.11	1738.99	0.18
24	5	Low	346.15	0.60	0.60	-0.32	427.27	13.25	13.28	-0.26	1512.39	324.09	325.73	-0.51
24	5	Med	337.53	0.72	0.72	-0.51	417.25	15.73	15.80	-0.42	1143.71	389.89	391.06	-0.30
24	5	High	343.72	0.92	0.92	-0.28	412.61	19.81	19.97	-0.81	798.99	490.61	493.04	-0.50
30	6	Low	1152.47	Time	0.63	-	2069.95	Time	19.19	**21.47**	4650.72	Time	452.8	**1.81**
30	6	Med	1334.26	Time	0.76	-	2215.83	Time	22.92	**15.32**	4300.53	Time	1745.96	**1.65**
30	6	High	1103.18	Time	0.97	-	1950.74	Time	28.71	**8.07**	1530.76	Time	2199.08	**0.57**
35	7	Low	1339.33	Time	0.58	-	3103.50	Time	20.78	**20.30**	5934.76	Time	1571.52	**1.79**
35	7	Med	1431.21	Time	0.70	-	3090.82	Time	24.80	**15.07**	5168.64	Time	1888.67	**1.73**
35	7	High	1642.54	Time	0.89	-	2945.21	Time	30.81	**7.38**	3275.66	Time	2448.3	**0.61**
Max.			1642.54			1.52	3103.50			21.47	5934.76			1.81
Avg.			568.63			0.10	1047.54			4.11	1753			0.33
Min.			87.10			-1.11	155.42			-0.81	143.17			-0.51

for Gurobi, that fails to find any feasible solution for larger instances with $N = 30,\ 35$.

We also notice that instances with high levels of supplies (U) are easier to solve. This observation is consistent with the intuition that MFPP is an easy problem to solve if there are abundant supplies. Nonetheless, in reality, food banks operate under a limited budget and demand exceeds supplies.

7.3. Sensitivity Analysis and Policy Insights

In this section, we derive additional insights by studying the effect of changing some critical parameters. First, we analyze the impact of *pooling*, i.e., what happens when more nodes can be served in one route. Recall that the duration of each stop of the MFPP is set to be 45 minutes. Therefore, any increase in the service time will decrease the number of stops a truck can serve as the driver need to get back to the food bank within a time window of 8 hours. On the other hand, a decrease in service time, say from 45 minutes to 30 minutes, potentially allows visiting more nodes per route, and hence the pooling effect can be realized. To this end, in this set of experiments we assume that the service time is 30 minutes rather than 45 minutes per stop. We use the data of Chemung county as it is a medium size network. If the service time is 30 minutes, two vehicles can serve all the nodes in Chemung county. On the other hand, if the service time is 45 minutes at each node, three vehicles are needed (the base case).

Figure 8 illustrates the mean efficiency under different models and levels of supplies for two vehicles (when the service time at each node is 30 minutes) and three vehicles (when the service time at each node is 45 minutes). We observe that the efficiency performance measure of the equity model benefits the most from the pooling effect as measured by the difference from the base case of non-pooling effect. On the other hand, maximum efficiency model realizes slight increase in the efficiency due to the pooling effect. Lastly, maximum efficiency and maximum equity model does not realize significant improvement in the efficiency due to the pooling effect.

Figure 9 illustrates the mean percentage of food waste under different models and levels of supplies for two and three vehicles. We observe that the percentage of food waste under the equity model drops significantly as the pooling effect is introduced. On the other hand, maximum efficiency model realizes slight decrease in the percentage of food waste due to the pooling effect. Lastly, maximum efficiency and maximum equity model does not realize significant reduction in the percentage of food waste due to the pooling effect.

Potential benefits to food bank managers. The importance of the analysis that we provide is not only limited to identifying how each model performs under various settings. Instead, it

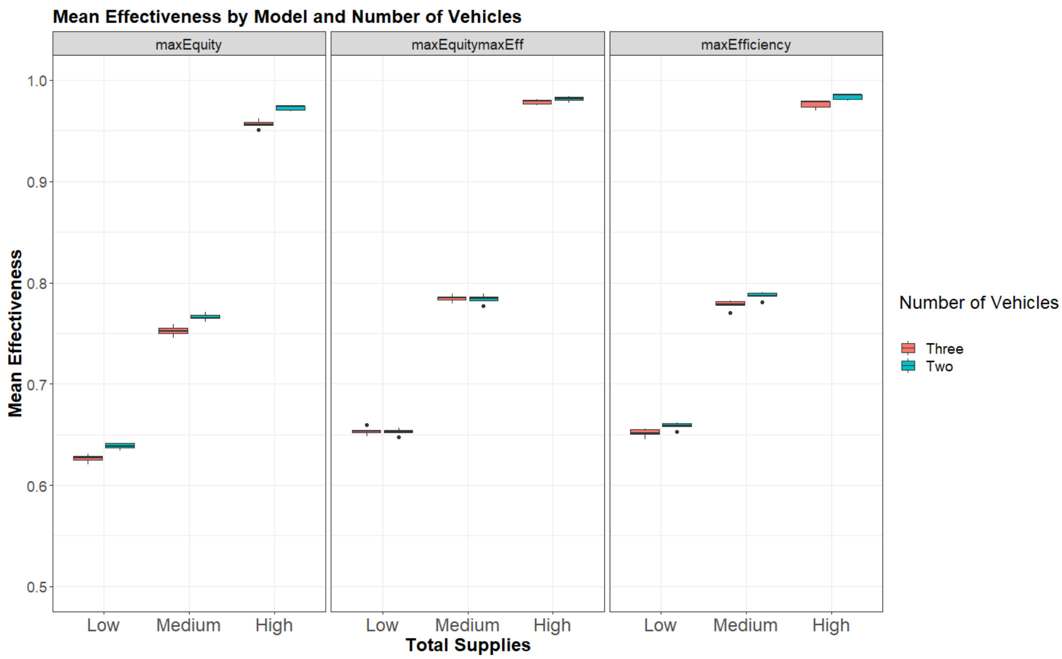


Figure 8: Mean efficiency by number of vehicles and levels of supplies

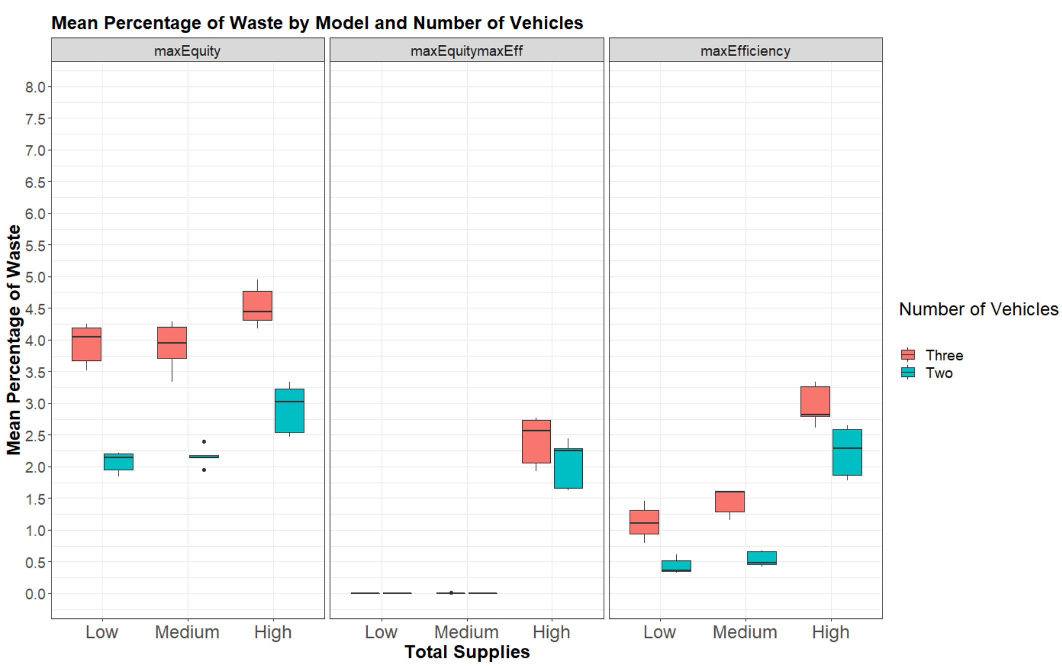


Figure 9: Mean percentage of food waste by number of vehicles and levels of supplies

highlights how using one model over another leads to substantial difference in different performance measures. For example, the performance of the maximum efficiency and maximum efficiency model are very comparable to model maximum efficiency when it comes to the efficiency performance measure while achieving superior performance when it comes to the equity performance measures too as displayed in Figure 6.

That said, it is reasonable question to ask, *can we utilize a multi-objective model that incorporates equity and efficiency parts in the objective function and achieve better performance results than M_Eq_M_Eff model?* Note that the terms in equation (6) are both in food units, what if we develop a new multi-objective model with the goal of maximizing the total number of units allocated while maximizing the minimum filling ratio. Such model will have an objective function of the form:

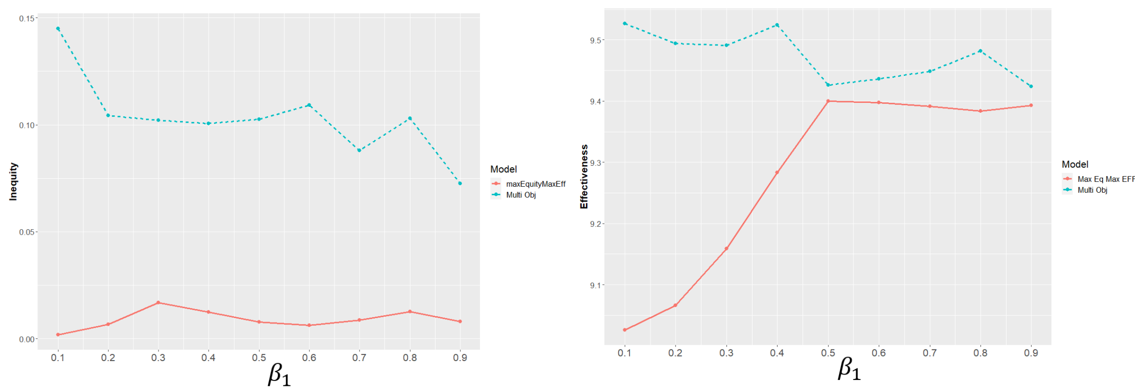
$$Q(x, y, q, \gamma, \xi) := \min \Big\{ -\beta_1 \sum_{i \in \mathcal{V}'} d_i f_i - (1 - \beta_1) U * \gamma \Big\} \tag{8}$$

The first part of (8) maximizes the total number of allocated units and the second part aims at maximizing the minimum filling ratio, the second part is multiplied by U for the sake of normalization as $\gamma \leq 1.0$, and $\beta_1 \in [0, 1]$ is the weight of importance of the efficiency performance measure. Similarly, if we modify M_Eq_M_Eff model to incorporate the weight of importance of efficiency and equity, we get the following objective function:

$$Q(x, y, q, \gamma, \xi) := \min \Big\{ -\beta_1 \sum_{i \in \mathcal{V}'} d_i f_i + (1 - \beta_1) \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}' : i < j} E_{ij} \Big\} \tag{9}$$

Figure 10 display the efficiency and equity results of models (8) and (9) labeled as Multi Obj and M_Eq_M_Eff, respectively. We note that the Multi-objective model achieves better efficiency than the M_Eq_M_Eff model especially when the value of β_1 is less than 0.5. For values of β_1 greater than 0.5, M_Eq_M_Eff model achieves slightly less results under the efficiency performance measure. On the other hand, we note that the Multi-objective model performs very poorly when considering the equity performance measure with inequity average values as high as 10.0% while the average inequity of the M_Eq_M_Eff model are always less than 2.0%. These results highlight that M_Eq_M_Eff model provides better overall results when considering both equity and efficiency simultaneously.

Another interesting point we observe is that if the decision maker wants to have more weight on the efficiency performance measure and less importance on the equity model, in that case it is better to use the Max. Efficiency model with the appropriate level of the inequity parameter α. We arrive to that conclusion since the Multi-objective model achieves better results than the M_Eq_M_Eff model for values of $\beta_1 < 0.5$ at the cost of having a higher inequity values.



(a) Equity

(b) Efficiency

Figure 10: Equity and Efficiency values of Multi-objective and M_Eq_M_Eff models.

7.4. Comparison against current practice

In this subsection we focus on comparing the performance of the optimization approach against the solution that the FBST implement in practice. The FBST routing decisions are based on first clustering the demand nodes based on spatial characteristics and then perform a greedy sequencing within each cluster by first visiting nodes with smallest expected demand based on historical data. On the other hand, the optimization approach refers to solving a problem using our developed matheuristic.

We demonstrate the results of comparing the optimization approach against the solution implemented in practice using the data from Chemung county, we note the same behavior over other counties but we report the results of Chemung county for brevity. Similar to the previous analysis, we study each instance under the settings of low, medium, and high supplies. For model M_Eff, we set the inequity parameter $\alpha = 0.05$. For any generated instance, we do not know the amount the driver will distribute for each node under each demand realization. (Here, the actual distributed quantity means the actual amount of food distributed at a given location which equals the pre-allocated amount plus additional food items (excess) carried from previous nodes.). However, if we fix the routing decisions following the routes used by the FBST, we can solve the three models to obtain the allocation decisions for any generated instance of data. On the other hand, we can obtain the solution of any generated instance by solving the corresponding model using the developed matheuristic. Once we have the solutions under any model using the optimization approach and the solutions of the FBST implementation, we can run a comparison between these two solutions in terms of food waste percentage, equity, and efficiency. In Figure 11, we begin by showing the

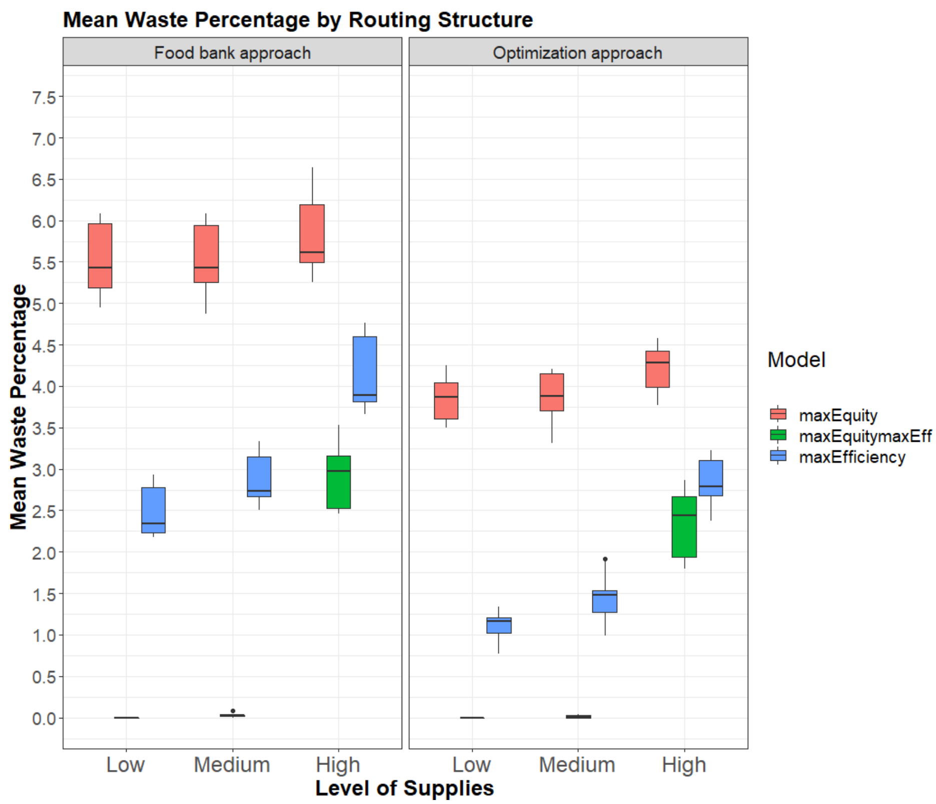


Figure 11: Comparison of the mean percentage of food waste under the optimization approach against the food bank practice.

mean food waste under the optimization approach against the FBST solution. We note that the food waste under the optimization approach is less than the food waste under the FBST routing decisions under all cases except the case of low and medium supplies of the M_Eq_M_Eff model. These results highlight the importance of routing decisions on the food waste performance measure regardless of the goal of the food bank manager (i.e., maximize equity, maximize efficiency, or maximize both).

Figure 12, displays the mean efficiency under the optimization approach against the FBST solution. We note that the efficiency under the optimization approach is higher than the efficiency under the FBST routing decisions for all tested cases. Lastly, figure 13 displays the mean inequity under the optimization approach against the FBST solution. We only report on the results of models M_Eff and M_Eq_M_Eff since model M_Eq provides 0 inequity under the optimization approach and the FBST routing. For M_Eq_M_Eff model we notice significant improvement in equity performance measure while inequity under M_Eff model using the FBST routing decisions is better than the equity achieved under the optimization approach. Using the optimization approach, the inequity

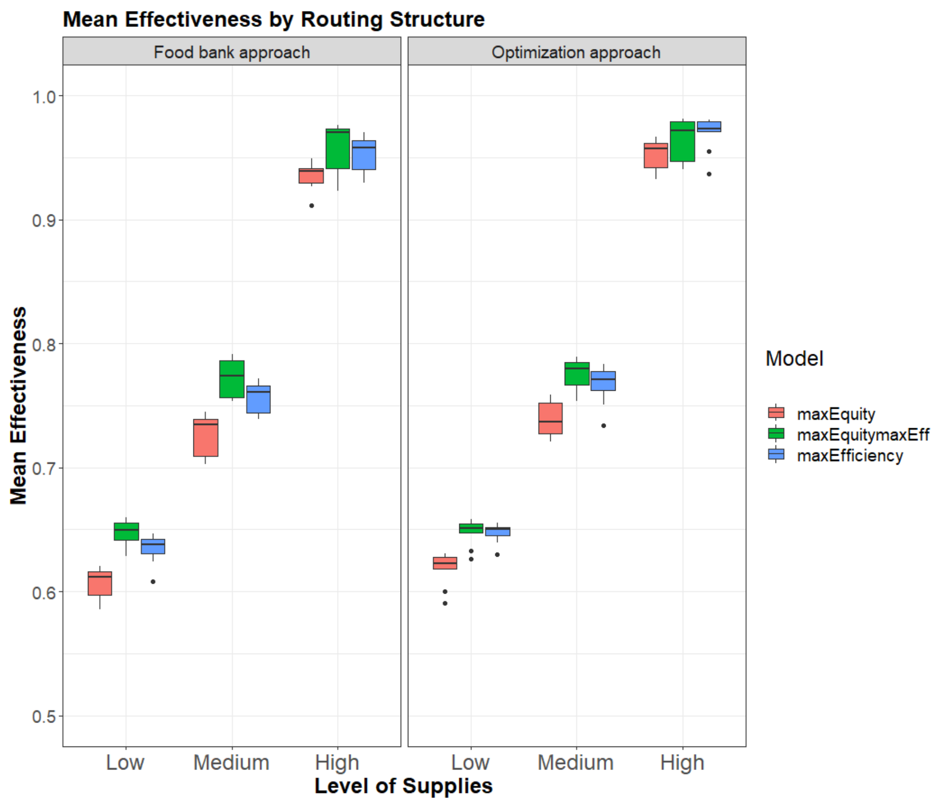


Figure 12: Comparison of the mean efficiency under the optimization approach against the food bank practice.

under M_Eff model is at 0.05 for the cases of low and medium supplies due to constraint 2f.

From Figures 11-13, we observe that the apparent importance of routing decisions on the performance measures even if the allocation is solved to optimality under each model. The food waste percentage obtained by our optimization approach improves on the FBST policies, respectively, by 4.7%. These improvements are obtained without adding any extra food resources and food bank managers would be very interested in obtaining these kinds of improvements by simply using their existing resources more carefully. Our optimization approach makes a significant step towards minimizing food waste and maximize total allocated food while maintaining equity. To illustrate the difference in routing among the different models and the one in practice, Figure 14 displays the routing under each model for Chemung county, the order of the nodes follows the same order presented in Table B.6 (for instance, node 2 refers to Erin demand node). The numeric value in red above each node represent the percentage of the average filling ratio taken across all scenarios for each node. We observe that the routing variables are substantially different under each model and a deviation in the allocation under each model.

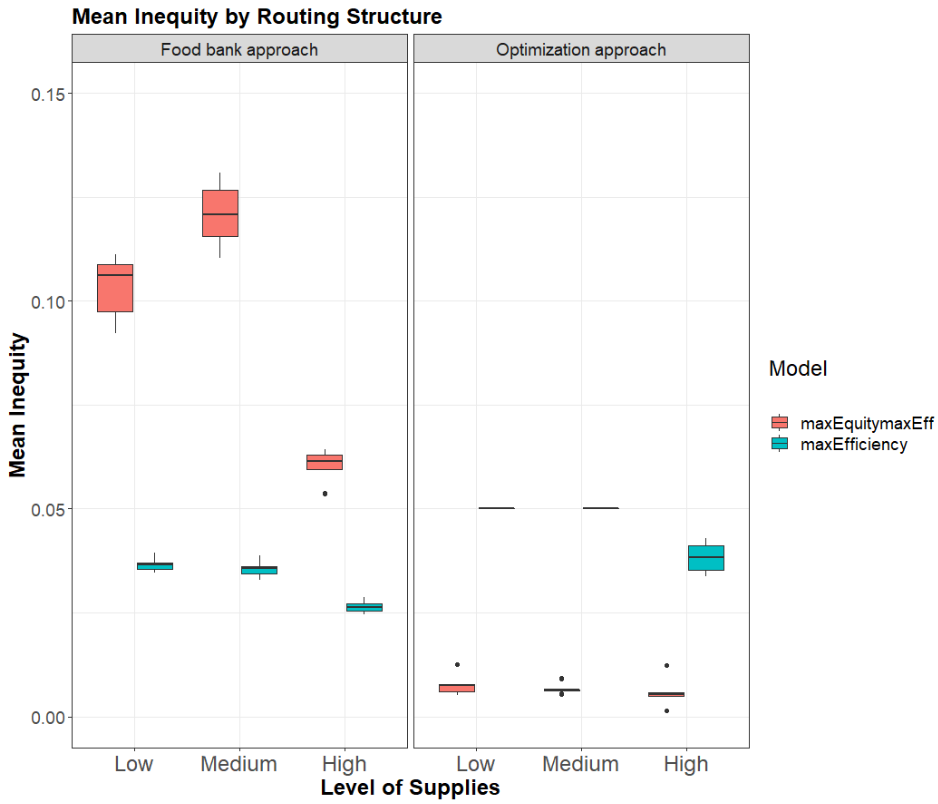
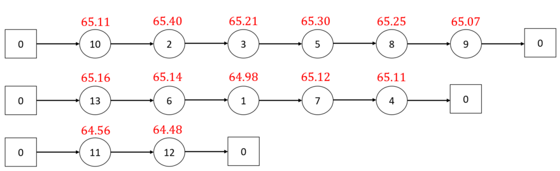
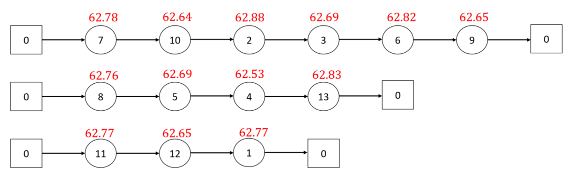


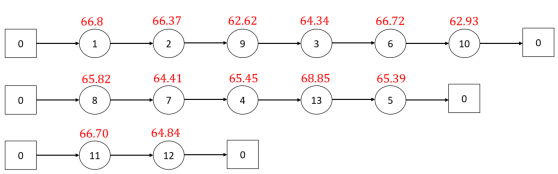
Figure 13: Comparison of the mean inequity under the optimization approach against the food bank practice.



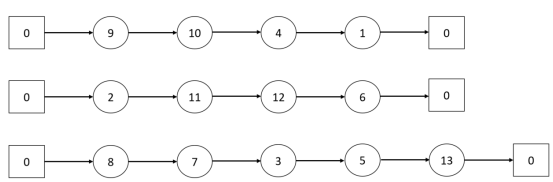
(a) M_Eq_M_Eff



(b) M_Eq



(c) M_Eff



(d) Routing used by the FBST

Figure 14: Routing decisions of each model.

8. Conclusion

In this article we address stochastic routing-allocation decisions in a nonprofit distribution system. Motivated by the MFPP operations at food banks, we study a multi-vehicle sequential allocation problem that incorporates three critical objectives for food distribution under stochastic demand: equity (maintaining similar service levels among all recipients) and efficiency (allocating as much food as possible). We develop three mathematical models focusing on these two measures and demonstrate the performance of each model under different performance measures. We show that the efficiency model achieves high level of efficiency and utilization of resource, low levels of food waste, and maintaining equity within the predefined tolerance as specified by the decision maker. On the other hand, we show that the equity model achieves near perfect equity among all recipients at the cost of high levels of food waste. Lastly, we show that the model that maximizes equity and efficiency at the same time demonstrates the best results when it comes to efficiency but at a higher levels of inequity. Furthermore, we show that routing-allocation models are very difficult to solve for large-scale practical instances. Hence, we develop a high quality matheuristic that utilizes ALNS scheme with a mathematical model to solve the developed models. Our matheuristic solves the developed models in a fraction of time when compared to the commercial solver Gurobi as well as providing high quality solutions with average gap never exceeding 4.11%.

An interesting area for future research would be to consider stochastic routing-distribution-collection decisions in non-profit setting. The FBST is located in a non-urban area where the MFPP only involves routing and distribution; however, this might not be the case for food banks located in urban areas. Based on our conversation with the Second Harvest Food Bank of Northwest North Carolina managers we find that their MFPP has different characteristics than the MFPP operated by the FBST. Specifically, the MFPP operations managed by the Second Harvest Food Bank of Northwest North Carolina involves visiting donors after they finish visiting the distribution nodes for collection/rescue of food items and then the trucks will be back to the food bank. Once the food trucks get back to the food bank loaded with food items that were rescued from donors, the volunteers at Second Harvest Food Bank of Northwest North Carolina will perform inspection, sorting, and packing of the received food items. Such variant necessitates adding the collection aspect to the studied problem under stochastic setting where the demand of nodes is not revealed until they are visited.

References

Adulyasak, Y., Cordeau, J.-F., Jans, R., 2014. Optimization-based adaptive large neighborhood search for the production routing problem. Transportation Science 48 (1), 20–45.

Alkaabneh, F., Diabat, A., 2022. A multi-objective home healthcare delivery model and its solution using a branch-and-price algorithm and a two-stage meta-heuristic algorithm. Transportation Research Part C: Emerging Technologies, 103838.

Alkaabneh, F., Diabat, A., Gao, H. O., 2020. Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. Computers & Operations Research 113, 104751.

Alkaabneh, F., Diabat, A., Gao, H. O., 2021. A unified framework for efficient, effective, and fair resource allocation by food banks using an approximate dynamic programming approach. Omega 100, 102300.

Anand, S., 1983. Inequality and poverty in Malaysia: Measurement and decomposition. The World Bank.

Archetti, C., Boland, N., Grazia Speranza, M., 2017. A matheuristic for the multivehicle inventory routing problem. INFORMS Journal on Computing 29 (3), 377–387.

Balcik, B., Iravani, S., Smilowitz, K., 2014. Multi-vehicle sequential resource allocation for a nonprofit distribution system. IIE Transactions 46 (12), 1279–1297.

Bartholdi III, J. J., Platzman, L. K., Collins, R. L., Warden III, W. H., 1983. A minimal technology routing system for meals on wheels. Interfaces 13 (3), 1–8.

Bassok, Y., Ernst, R., 1995. Dynamic allocations for multi-product distribution. Transportation Science 29 (3), 256–266.

Berman, O., Larson, R. C., 2001. Deliveries in an inventory/routing problem using stochastic dynamic programming. Transportation Science 35 (2), 192–213.

Besik, D., Nagurney, A., Dutta, P., 2022. An integrated multitiered supply chain network model of competing agricultural firms and processing firms: The case of fresh produce and quality. European Journal of Operational Research.

Cordeau, J.-F., Laporte, G., Pasin, F., Ropke, S., 2010. Scheduling technicians and tasks in a telecommunications company. Journal of Scheduling 13 (4), 393–409.

Davis, L. B., Sengul, I., Ivy, J. S., Brock III, L. G., Miles, L., 2014. Scheduling food bank collections and deliveries to ensure food safety and improve access. Socio-Economic Planning Sciences 48 (3), 175–188.

Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. European Journal of Operational Research 223 (2), 346–359.

Eisenhandler, O., Tzur, M., 2018. The humanitarian pickup and distribution problem. Operations Research 67 (1), 10–32.

Eisenhandler, O., Tzur, M., 2019. A segment-based formulation and a matheuristic for the humanitarian pickup and distribution problem. Transportation Science 53 (5), 1389–1408.

Fianu, S., Davis, L. B., 2018. A markov decision process model for equitable distribution of supplies under uncertainty. European Journal of Operational Research 264 (3), 1101–1115.

Ghoniem, A., Scherrer, C. R., Solak, S., 2013. A specialized column generation approach for a vehicle routing problem with demand allocation. Journal of the Operational Research Society 64 (1), 114–124.

Glover, F., Samorani, M., 2019. Intensification, diversification and learning in metaheuristic optimization. Journal of Heuristics 25 (4), 517–520.

Hasnain, T., Sengul Orgut, I., Ivy, J. S., 2021. Elicitation of preference among multiple criteria in food distribution by food banks. Production and Operations Management 30 (12), 4475–4500.

Homem-de Mello, T., Bayraksan, G., 2014. Monte carlo sampling-based methods for stochastic optimization. Surveys in Operations Research and Management Science 19 (1), 56–85.

Islam, M. H., Ivy, J. S., 2022. Modeling the role of efficiency for the equitable and effective distribution of donated food. OR Spectrum 44 (2), 485–534.

Jagtap, S., Trollman, H., Trollman, F., Garcia-Garcia, G., Parra-López, C., Duong, L., Martindale, W., Munekata, P. E., Lorenzo, J. M., Hdaifeh, A., et al., 2022. The russia-ukraine conflict: Its implications for the global food supply chains. Foods 11 (14), 2098.

Johnson, M. P., Gorr, W. L., Roehrig, S. F., 2002. Location/allocation/routing for home-delivered meals provision. International Journal of Industrial Engineering 9, 45–56.

Kumar, A., Schwarz, L. B., Ward, J. E., 1995. Risk-pooling along a fixed delivery route using a dynamic inventory-allocation policy. Management Science 41 (2), 344–362.

Lee, D., Sönmez, E., Gómez, M. I., Fan, X., 2017. Combining two wrongs to make two rights: Mitigating food insecurity and food waste through gleaning operations. Food policy 68, 40–52.

Lien, R. W., Iravani, S. M., Smilowitz, K. R., 2014. Sequential resource allocation for nonprofit operations. Operations Research 62 (2), 301–317.

Mak, W.-K., Morton, D. P., Wood, R. K., 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. Operations research letters 24 (1-2), 47–56.

Mandell, M. B., 1991. Modelling effectiveness-equity trade-offs in public service delivery systems. Management Science 37 (4), 467–482.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., Mladenović, N., 2014. A two phase variable neighborhood search for the multi-product inventory routing problem. Computers & Operations Research 52, 291–299.

Nagurney, A., 2021. Perishable food supply chain networks with labor in the covid-19 pandemic. In: Dynamics of Disasters. Springer, pp. 173–193.

Orgut, I. S., Ivy, J., Uzsoy, R., Wilson, J. R., 2016. Modeling for the equitable and effective distribution of donated food under capacity constraints. IIE Transactions 48 (3), 252–266.

Orgut, I. S., Ivy, J. S., Uzsoy, R., Hale, C., 2018. Robust optimization approaches for the equitable and effective distribution of donated food. European Journal of Operational Research 269 (2), 516–531.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers & operations research 34 (8), 2403–2435.

Ribeiro, G. M., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Computers & operations research 39 (3), 728–735.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science 40 (4), 455–472.

Ruf, M., Cordeau, J.-F., 2021. Adaptive large neighborhood search for integrated planning in railroad classification yards. Transportation Research Part B: Methodological 150, 26–51.

Savas, E. S., 1978. On equity in providing public services. Management Science 24 (8), 800–808.

Shapiro, A., 2003. Monte carlo sampling approach to stochastic programming. In: ESAIM: Proceedings. Vol. 13. EDP Sciences, pp. 65–73.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. In: International conference on principles and practice of constraint programming. Springer, pp. 417–431.

Shehadeh, K. S., Cohn, A. E., Jiang, R., 2021. Using stochastic programming to solve an outpatient appointment scheduling problem with random service and arrival times. Naval Research Logistics (NRL) 68 (1), 89–111.

Sinclair, S. R., Jain, G., Banerjee, S., Yu, C. L., 2020. Sequential fair allocation of limited resources under stochastic demands. arXiv preprint arXiv:2011.14382.

Zoha, N., Hasnain, T., Ivy, J., 2022. Tradeoff between geographic and demographic equity in food bank operations. In: IIE Annual Conference Proceedings.

Appendix A. ALNS algorithm for the MFPP problem

Algorithm 2: ALNS framework for the MFPP problem

Input: A feasible solution s produced by a construction heuristic and a set of ALNS parameters: $Temp$, ψ, $iterMax$, $MaxNoImprov$

Output: The best known solution s^*

Set $iter \leftarrow 0$, $s^* \leftarrow s$, $s_{current} \leftarrow s$, $\eta = [1, ..., 1]$
while $iter < iterMax$ **do**
  Select a destroy operator (d) and repair operator (r) by a roulette wheel mechanism using weights $\mathbf{w}$
  Apply destroy operator d on s and get s'
  Apply repair operator r on s' and get s_{new}
  if $f(s_{new}) < f(s^*)$ **then**
    $s^* \leftarrow s_{new}$, $s_{current} \leftarrow s_{new}$
    $NoImproveCounter \leftarrow 0$
  else
    $NoImproveCounter \leftarrow NoImproveCounter + 1$
    if $f(s_{new}) < f(s_{current})$ **then**
      $s \leftarrow s_{new}, s_{current} \leftarrow s_{new}$
    else
      $probAccept \leftarrow e^{\frac{f(s_{new}) - f(s_{current})}{Temp/iter}}$
      Generate a random number $U \sim Uniform[0, 1]$
      if $U < probAccept$ **then**
        $s \leftarrow s_{new}, s_{current} \leftarrow s_{new}$ // Accept the new solution which is worst than the current solution for diversification
      else
        $s \leftarrow s_{current}$ // Reject the new solution and use the current solution
  $iter \leftarrow iter + 1$
  Update η and calculate $\mathbf{w}$ and
  if $NoImproveCounter\%MaxNoImprov == 0$ **then**
    $s \leftarrow s^*$, $NoImproveCounter \leftarrow 0$

Appendix B. Details on the demand nodes

The FBST service area includes the counties of Broome, Chemung, Schuyler, Steuben, Tioga, and Tompkins, covering nearly 4,000 square miles. FBST serves over 21,000 people per week, and it distributed over 17.6 million pounds of food and groceries in 2020. It is worth mentioning that FBST is located in a food desert[4]. Figure B.15 shows the locations of the distribution nodes and the color of the circles reflect the average demand at each node.

[4]USDA defines food deserts as parts of the country vapid of fresh fruit, vegetables, and other healthful whole foods, usually found in impoverished areas. This is largely due to a lack of grocery stores, farmers markets, and healthy food providers.

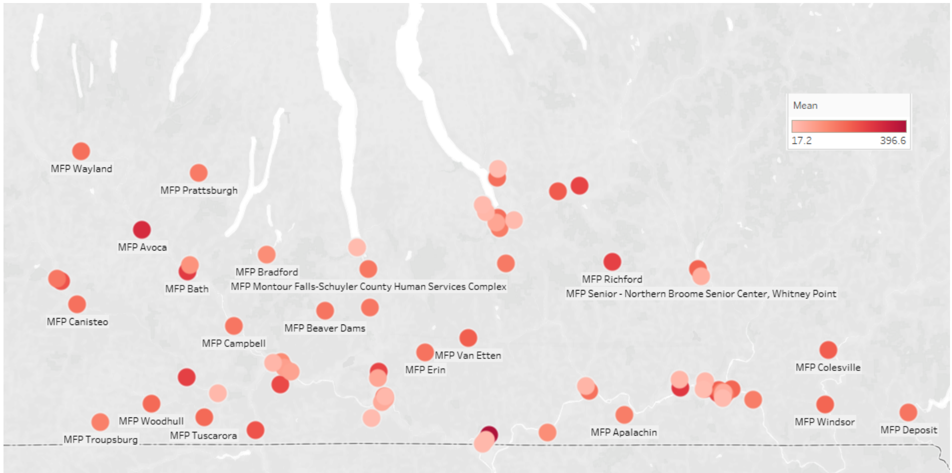


Figure B.15: Distribution of demand nodes served by the FBST

Tables B.4-B.8 display more information about the demand nodes of Tioga, Tompkins, Chemung, Broome, and Steuben counties, respectively. The first column in Tables B.4-B.8 displays the program/demand node name, the second column reports the shape parameter of demand, the third column reports the scale parameter of demand, and finally the fourth column shows the longitude and latitude on the map.

Table B.4: Details on Tioga County demand nodes.

Program name	k	θ	(long., lat.)
Apalachin	50.34	3.27	42.058463,-76.1693389
Nichols-The Creamery	45.89	3.26	42.02302,-76.371793
Owego VFW	40.00	2.12	42.1043411,-76.2615965
Richford	49.26	2.8	42.3551522,-76.2008963
Senior - Elizabeth Square, Waverly	35.81	0.79	42.001546,-76.541203
Senior - Long Meadow Senior Housing	43.99	1.02	42.1135325,-76.2704045
Senior - Springview Apartments	44.56	0.61	42.0100399,-76.533895
Van Etten	54.29	1.76	42.2080409,-76.579406
Waverly	70.15	2.74	42.0194466,-76.5236008
Tioga County Rural Ministry	60.2	2.45	42.106912, -76.265281

Table B.5: Details on Tompkins County demand nodes.

Program name	k	θ	(long., lat.)
Senior - Ellis Hollow	26.98	1.04	42.435955,-76.460374
College TC3 -College	27.89	2.10	42.502126,-76.287671
Senior - Conifer Village	19.03	1.45	42.4511294,-76.5323538
College Ithaca College	24.03	1.60	42.4199351,-76.4969643
Senior - Woodsedge Apartments	21.60	0.88	42.535151,-76.501086
Salvation Army Ithaca	48.92	1.30	42.4398066,-76.5019869
Danby	71.87	2.08	42.3522565,-76.4800051
Senior - Cayuga Meadows	26.81	0.99	42.464569,-76.5409393
Lansing	68.89	2.41	42.5183206,-76.5035538
Reach for Christ Church Freeville	53.89	2.60	42.4919905,-76.3443272
Senior - Titus Towers	77.70	1.03	42.4317058,-76.504801
Cornell University	20.47	0.80	42.450017, -76.488951
Dryden United Methodist Church	40.78	1.28	42.490688, -76.296849

Table B.6: Details on Chemung County demand nodes.

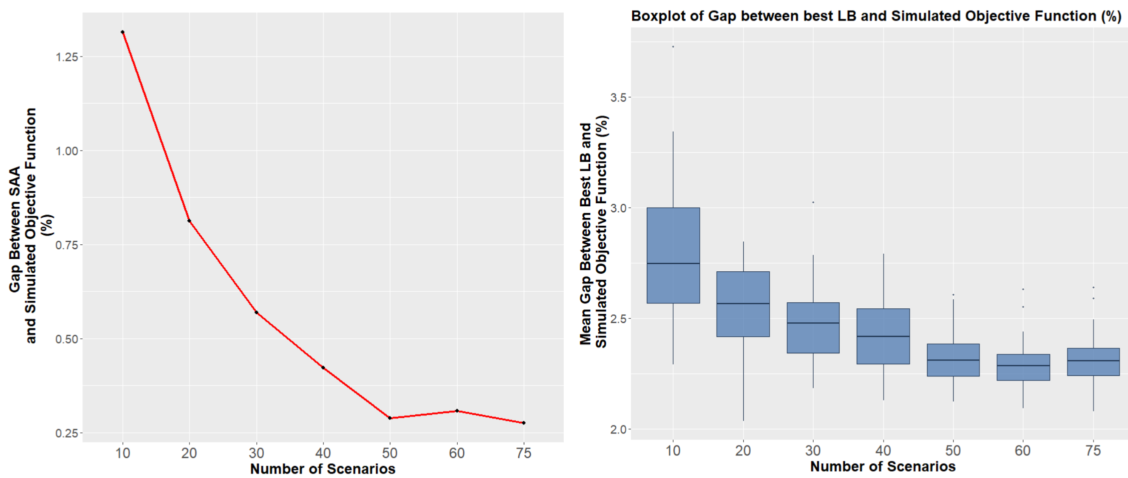
Program name	k	θ	(long., lat.)
Beaver Dams	33.25	1.04	42.2606415,-76.9559856
Erin	24.88	2.10	42.1788978,-76.6922607
Feed Elmira - Hathorne Court	53.38	1.45	42.08873,-76.800510
Millport	30.87	1.60	42.267172,-76.837356
Senior - Bragg	62.01	0.88	42.0896598,-76.7977954
Senior - Carpenter Apartments	28.11	1.30	42.0930065,-76.7984317
Senior - Flannery	44.76	2.08	42.0819957,-76.8053269
Senior - Park Terrace Congregate Apts.	39.00	0.99	42.051364,-76.832954
Senior - Villa Serene	99.30	2.41	42.1295916,-76.8158267
The Love Church	65.16	2.60	42.1412225,-76.8137998
Tioga County Rural Ministry	9.87	1.03	42.106820,-76.265460
Fenton	187.68	0.80	42.166040, -75.832960
College Corning Community College	29.54	1.31	42.148850, -77.055050

Table B.7: Details on Broome County demand nodes.

Program name	k	θ	(long., lat.)
American Legion - Binghamton	11.19	1.21	42.108036,-75.887779
Boys and Girls Club	11.53	0.99	42.1053841,-75.9213736
Colesville	13.88	1.30	42.1841905,-75.6329193
Conklin- Maines Community Center	20.44	1.10	42.0870901,-75.8309647
Deposit	26.48	1.21	42.0625771,-75.422248
Endwell United Methodist Church	24.16	0.85	42.111296,-76.02203
Family Enrichment Network	21.40	0.75	42.1096901,-75.9558181
First Assembly Of God Church	25.20	0.91	42.1022814,-75.9129518
Redeemer Lutheran Church	11.66	1.20	42.1000047,-75.9224594
Saint Mary Recreation Center	24.31	1.30	42.0988789,-75.9041391
Senior - East Hill Senior Living	29.69	1.21	42.114015, -75.872065
Senior - Harry L Apartments	39.35	0.80	42.1237339,-75.9567565
Senior - Lincoln Court	10.23	0.40	42.090217,-75.910198
Senior - Marian Apartments	20.00	0.50	42.125661,-76.025155
Senior - Metro Plaza Apartments	9.79	1.10	42.1016977,-75.9085338
Senior - North Shore Towers	33.34	1.70	42.0966132,-75.9103884
Senior - N. Broome Sen. Center	40.05	1.30	42.3269618,-75.9677778
Senior - Wells Apartments	8.56	1.20	42.1076045,-75.9607932
Whitney Point	48.35	0.45	42.3405329,-75.9765268
Windsor	95.65	0.70	42.0779615,-75.6427789
Golden Glow Fire Dept	13.00	0.50	42.106134, -76.021179

Table B.8: Details on Steuben County demand nodes.

Program name	k	θ	(long., lat.)
Troupsburg	11.19	1.21	42.0432505,-77.5456383
Lamphear Court	11.53	0.99	42.1598551,-77.0711389
Canisteo	13.88	1.30	42.272779,-77.606849
Bradford	20.44	1.10	42.3691309,-77.1081064
Campbell	26.48	1.21	42.2314129,-77.1948265
Rehoboth Deliverance Ministry	24.16	0.85	42.318273,-77.6493618
Senior - Dayspring	21.40	0.75	42.1420715,-77.0454685
Lindley	25.20	0.91	42.028142,-77.1392597
Rathbone	11.66	1.20	42.1306252,-77.3197915
Wayland	24.31	1.30	42.5685797,-77.5957241
Senior - Village Square/Manor	29.69	1.21	42.159804,-77.09139
Avoca	39.35	0.80	42.4174468,-77.4358433
Woodhull	10.23	0.40	42.0798397,-77.4111295
Senior - Addison Place Apartments	20.00	0.50	42.1001876,-77.2371924
Senior - Corning Senior Center	9.79	1.10	42.1490255,-77.0619725
Bath	33.34	1.70	42.3362172,-77.3175047
Tuscarora	40.05	1.30	42.0522655,-77.2737134
Prattsburgh	8.56	1.20	42.5272099,-77.2866
Hornell	48.35	0.45	42.3229281,-77.6593841
Senior - CFS Lakeview	95.65	0.70	42.3488781,-77.3110763
Corning Community College	10.00	0.50	42.1172594,-77.0735445
Wellsburg	25.77	2.01	42.010540, -76.726114
Chenango Forks	42.77	1.75	42.240207, -75.846290
Maine	30.01	1.74	42.194378, -76.060179



(a) `Gap` (b) `AGap`

Figure C.16: `Gap` and `AGap` values as function of sample size R.

Appendix C. Selection of the Number of Scenarios

We employed a procedure similar to the Monte Carlo Optimization Procedure to determine an appropriate sample size R to obtain near-optimal solutions to maximize efficiency model (1)-(2) based on its sample average approximation (SAA) (Homem-de Mello and Bayraksan 2014, Mak et al. 1999, Shapiro 2003, Shehadeh et al. 2021). Starting with an initial candidate value of H=10, the procedure proceeds as follows. First, we repeat the following steps for $m = 1, \dots, M,$: (1) we generate a sample of R i.i.d. scenarios of demand, (2) we solve the SAA formulation of (1)-(2) and record the corresponding optimal objective value z_R^m and optimal $[x_{ij}^k, y_i^k, q_i^k]_R^m$ (i.e., first stage decision variables), and (3) we evaluate the objective function value $z_{R'}^m$ using Monte Carlo Simulation of the optimal $[x_{ij}^k, y_i^k, q_i^k]_R^m$ with a new sample of $R' >> R$ i.i.d. scenarios of demand. Second, we compute the average of z_R^m and $z_{R'}^m$ among the K replications as $\bar{z}_R = \frac{1}{M}\sum_{m=1}^M z_R^m$ and $\bar{z}_{R'} = \frac{1}{M}\sum_{m=1}^M z_{R'}^m$. Given that we solve the SAA using our matheuristic, both $\bar{z}_{R'}$ and $\bar{z}_R$ are statistical upper bound on the optimal value of the problem, with $\bar{z}_{R'}$ representing an upper bound on $\bar{z}_R$ (largest objective value if we adopt the solution we obtain from the matheuristic). Finally, we compute the approximate gap between $\bar{z}_R$ and $\bar{z}_{R'}$ as $\texttt{Gap} = \frac{\bar{z}_{R'} - z_R}{\bar{z}_{R'}}$ and the approximate optimality gap between $\bar{z}_{R'}$ and z^* as $\texttt{AGap}_{\bar{z}_{R'}} = \frac{\bar{z}_{R'} - z^*}{\bar{z}_{R'}}$, where z^* is the best lower bound.

We implemented the above steps with an initial value of $H = 10$ scenarios, $M = 35$ replications, and an instance of 16 nodes (we observe similar results for other choices of nodes). We compute z^* as the best LB following the discussion in Section 6.

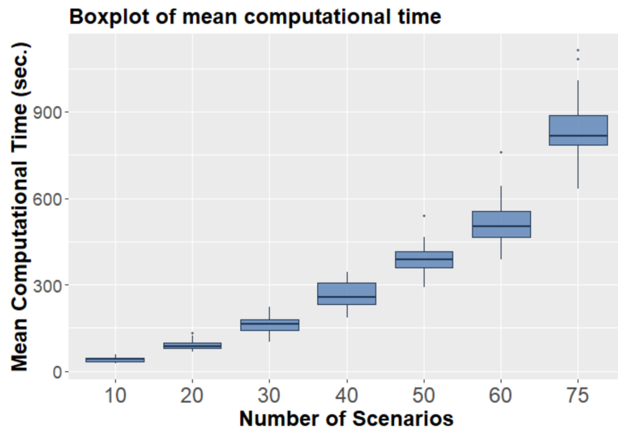


Figure C.17: Solution time as function of of sample size R

Figure C.16 presents the `Gap` and `AGap` as a function of the sample size R. From this figure, we observe that as R increases, both gap values decrease. Most importantly, `AGap` values range from 3% under H = 10 to 2.1% under H = 75, indicating that solving the SAA via our heuristic with a small sample size yield a near-optimal solution. In contrast, larger sample sizes resulted in longer solution times (see Figure C.17) without consistent and significant improvements in the `AGap` Based on these considerations, we selected $R = 40$.