The Multi-Period Humanitarian Gleaning and Distribution Problem

Ohad Eisenhandler, Michal Tzur

Department of Industrial Engineering, Iby and Aladar Fleischman Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel

ohadeize@mail.tau.ac.il , tzur@eng.tau.ac.il

September 2019

Abstract

The problem we study is motivated by the non-profit activity of a food bank, which collects unharvested fresh produce from donor farmers, processes it at a designated depot and distributes it from there to food aid agencies serving needy individuals. In this setting, the food bank needs to simultaneously determine, over a given decision horizon, both a schedule for the collection of crops from the farmers and a distribution plan of the collected food from the depot to the agencies. Clearly, there exists a strong dependency between these two activities, through the amount of available food as well as the use of common resources. A special humanitarian objective function captures the goals of the food bank, namely maximizing the total amount distributed and ensuring equity in the allocation to the different agencies. In the solution method we propose, the food bank's problem is decomposed into its collection and distribution aspects. We show how to tackle each sub-problem individually, while considering information obtained from its counterpart. The implementation of a rolling horizon framework for the problem is also discussed. Results from numerical experiments based on real life and randomly generated datasets, demonstrate the advantage of the proposed method, compared to naïve approaches and to the methods currently being used in practice.

Keywords: Vehicle Routing; Vehicle Scheduling; Humanitarian Logistics; Food Banks; Multi-period; Rolling Horizon

1. Introduction

Food insecurity is one of the most challenging social issues to cope with and it is prevalent in many countries, even where the economy is generally stable and the standard of living is high. In Israel, the home country of the organization whose activity motivated this study, official reports have found that 1 in 5 Israeli citizens suffer from food insecurity, almost half of which at a significant level of severity (Endeweld et al., 2014). The same survey also found that 55% of the nutritionally insecure families in Israel relied on assistance from their communities, especially through the support of various aid organizations, to acquire additional amounts of food and improve their nutrition. This finding emphasizes the importance of practicing food rescue, by which edible food that would otherwise be thrown away is collected and provided to individuals in need. The rescued food can be of distinct types, namely a cooked meal, any pre-packaged product or raw produce and the distinction between these categories has implications with respect to the underlying logistic process required to salvage the food. The problem we focus on in this paper is motivated by the latter type and specifically in situations where the produce has not yet been harvested by the farmer at the time of the food bank's decision making.

The act of collecting crops from fields to the benefit of the needy is commonly referred to as *gleaning*. Farmers usually allow their fields to be gleaned when the produce cannot be sold in the market, primarily because of either low revenue potential or some cosmetic impairments (that do not damage the nutritional quality of the plant). In these cases, unless the crops are collected, they are left to rot or to be plowed under. Consequently, it might be more cost-effective for farmers to have their crops gleaned than to pay costly pickers to go back through the fields. Further motivations are provided by tax benefits and by the social contribution to alleviating hunger.

The practice of gleaning dates back to Biblical times (e.g., Leviticus 23:22), when farmers were ordered to leave parts of their fields unharvested so that needy families could directly glean them to their benefit. It was a common practice for hundreds of years in Europe as well, finding vast representation in the literature and in art. Nowadays, gleaning is performed on a larger scale as part of a network of food aid agencies that receive excess crops from farmers in their vicinity. This network is typically managed by a food bank, which is a nonprofit charitable organization that acts as the central decision maker on behalf of the agencies.

To perform these operations, the food bank uses a fleet of dedicated vehicles, restricted by their capacity and by drivers' maximal permitted traveling time. The food bank determines which donor farmers to visit using any of its vehicles on each day of the decision horizon. Following collection, the produce is sent to a logistic center where it undergoes sorting and boxing, and then remains in proper storage conditions until the food bank decides to deliver it to one of the agencies served by it, using the same fleet of vehicles. The agencies, in turn, serve it to the end-beneficiaries. It should be noted that a vehicle that is assigned to a collection task (i.e., collecting crops from a field that is gleaned) cannot perform any other tasks on that day due to the constant loading of the produce onto the vehicle.

In contrast to the monetary-oriented considerations (i.e., minimizing costs or maximizing profits) usually employed in the private sector, the food bank's goals are focused on the humanitarian benefit obtained through these operations. We discuss the exact considerations and the objective function that represents them in greater detail in Section 2. The resulting problem, which we refer to as the Multi-Period *Humanitarian Gleaning and Distribution Problem* (H-GDSP for short), is motivated by the activity of *Leket Israel* (Hebrew for *Gleaning for Israel*). It is one of Israel's two major food banks, serving 175,000 individuals through 200 food aid agencies nationwide, with an emphasis on gleaning operations. Similar endeavors operate in the U.S. and in Europe. As will become clear later, the H-GDSP is meaningful only when considered over multiple periods, therefore for simplicity we sometimes refer to it as the H-GDSP and omit the multi-period characteristic from its name abbreviation.

The problem faced by the food banks that manage gleaning networks shares its characteristics with several classes of well-studied routing and/or inventory problems. All of them differ from our problem in their objective function (i.e., economic vs. humanitarian), hence in the following we shall mainly highlight the similarities and differences in terms of the decisions and constraints. In the pickup and delivery problem (PDP), a set of commodities needs to be transported between given origin and destination sites. Our problem mostly resembles the class of many-to-many PDPs, in which each commodity can be collected from and transported to many sites. However, this variation of the PDP is quite scarce in the literature, see a review in Berbeglia et al., (2007). Another major difference is that PDPs are typically solved for one period at a time, and do not need to consider the dynamics of the inventory. In contrast, the Inventory Routing Problem (IRP) focuses on supplying demands required by customers over a certain decision horizon, and considers both the transportation and inventory costs. An interested reader is referred to the review by Coelho et al. (2013). However, the typical assumption is that the depot has a sufficient amount of supply, which is inherently different from our problem, in which the amount of supply is determined by the decision maker under various constraints and is typically limited.

Another related area, the field of humanitarian logistics, which has witnessed in recent years a surge of research on the activity of food banks, focusing on the strategic (e.g., facility location) and/or operational (e.g., vehicle routing or resource allocation) decisions that need to be determined. For example, Lien et al. (2014) and Balcik et al. (2014) studied the single- and multi-vehicle versions respectively, of the problem of allocating given amounts of rescued food according to demands that are only observed upon arrival to the agencies (following a fixed route). Their objective was to maximize the lowest expected fill rate assigned to any agency. Recent studies have mostly looked into settings in which food needs to be collected from suppliers and immediately distributed to agencies. For example, Nair et al. (2017) used a goal-programming approach and a multi-stage heuristic to balance two objectives: the total transportation cost and the equity of the allocation. The periodic variant of the problem was studied in Nair et al. (2018), although without considerations of equity. Rey et al. (2018) proposed an objective function based on allocation-envy between agencies to balance the transportation costs and the inequality, and solved the problem using Benders' Decomposition and a heuristic approach. Finally, Eisenhandler and Tzur (2019a) proposed an original objective function, based on the Gini Index, to balance the total amount supplied and the equity of the allocation. They showed that given this objective function, the allocation aspect of the problem becomes easy to solve, so that an LNS approach for the routing aspect outperforms a commercial solver given the same amount of running time. A successful math-heuristic for the same problem was proposed in Eisenhandler and Tzur (2019b), based on a novel mathematical formulation, which focuses on sequences of sites within the route rather than single sites.

The assumption that the food collected from the donor suppliers can be immediately distributed to the food aid agencies holds when pre-packaged supplies are involved, which is the case for the vast majority of research on the logistic challenges of food banks. However, this is not the case for gleaning activities since they focus on fresh produce. As a matter of fact, research on gleaning from an operational perspective is generally quite scarce. Davis et al. (2016) developed predictive models to forecast donation sizes of different products distributed by a food bank, including fresh produce. Lee et al. (2017) focused on the strategic problem of determining which fixed schedule slots to offer weekly for potential gleaning requests. By simulating the stochastic processes of both donation sizes and the arrival of volunteers, they aimed to maximize the total expected volume gleaned over a given decision horizon. Lastly, Ata et al. (2019) formulated a dynamic model for the volunteer staffing problem, whose goal was to find the control policy that maximizes the total volume gleaned (minus penalties for turning down donations).

The aim of the current study is to close the gap with respect to the logistic challenges of food banks that manage gleaning operations, and focus on decision making with respect to the collection schedule and the distrbution routes. Since the gleaned produce cannot be immediately distributed, these operations give rise to a problem with special characteristics that have not been considered in previous research: (1) vehicle routes are separated to pickup (backhaul) vs. delivery (linehaul) lines, while the schedule has to meet the restrictions of both types of stakeholders, namely, the suppliers' time windows for collection and the agencies' required delivery amounts; (2) a multi-period problem whereby the decisions for the different periods are intertwined through the dynamics of the inventory at the depot. For this problem, we provide several non-trivial solution approaches based on mathematical formulations, all optimized under a humanitarian objective function. We test the described solution methods based on randomly-generated instances, as well as a real-life dataset obtained from the Israeli food bank, *Leket Israel*.

The remainder of the paper is organized as follows: in Section 2 we formally state the problem at hand and discuss a mathematical formulation. In Section 3 we suggest a solution approach which is based on decomposing the problem to its collection and distribution sub-problems, and in Section 4 we discuss the use of route-based decision variables. In Section 5 we introduce ways to compute upper bounds for the problem. In Section 6 we discuss a rolling horizon framework for the problem. In Section 7 we present our numerical analyses and finally in Section 8 we discuss our conclusions.

2. Problem Statement

In this section we provide an exact definition of the H-GDSP and present a complete mathematical formulation for it.

2.1 *Problem Setting*

We start by presenting the problem setting, including the underlying assumptions and the notation we use. A food bank is planning the gleaning operations it manages for a given horizon of days T, which we denote w.l.o.g. by $0, \ldots, |T| - 1$ (we also assume that work is performed continuously, also on weekends). The food bank, whose depot is denoted as site "0", serves a set of food aid agencies D and a set P of donor farmers (also referred to as *suppliers*) that are willing to have their fields gleaned by the food bank. A set K of capacitated vehicles is available daily, each of which may be used by the food bank for one of two purposes: (1) glean - that is, harvest and collect produce harvested by the food bank's team from donor farmers and transport it to the depot where it would undergo certain processing procedures on the following day (despite the fine distinction between *gleaning* and *harvesting*, we use both terms to represent the same activity); and (2) distribute the food following the completion of its processing, to the food aid agencies. We assume that a single product is collected and distributed.

The food bank makes its operational decisions only at the beginning of the first day of the decision horizon before any vehicle has left the depot. It is assumed that at the time of the planning, complete information is available with respect to both the donor farmers and the food aid agencies. Regarding the former, we assume without loss of generality that each supplier offers only one donation throughout the course of the decision horizon. However, this is not a restrictive assumption because any supplier with more than one donation can be modeled as several distinct suppliers, one for each donation. The information provided by each such supplier $i \in P$ includes the following elements: the amount of produce that can be collected, denoted by S_i; the time window the farmer allows the field to be gleaned, defined by the first and the last days for collection, denoted by $a_i \in T$ and $b_i \in T$, respectively (such that $a_i \leq b_i$), based on which we denote the subset of suppliers that can be visited on day $t \in T$, by $P^t \subset P$, i.e., $P^t = \{i \in P \mid a_i \leq t \leq b_i\}$; and the expiration date of the donated produce, denoted by d_i, i.e., the last day on which it can be distributed to any agency, or it becomes spoiled and needs to be destroyed by the food bank. We assume that exactly one workday is required to glean the amount donated by each supplier (this can be adjusted by the workforce the food bank uses), therefore we also note that there is no need to visit any supplier more than once.

With respect to the distribution side of the problem, we assume that all agencies are open every day and may be visited more than once throughout the decision horizon, but no more than once daily (i.e., split deliveries are not allowed). However, there is a fixed amount of food, denoted by y_i, that needs to be supplied to agency $i \in D$ whenever it is visited. This amount can vary between agencies, depending on the number of individuals they serve, denoted by n_i for agency i (no individual is served by more than one agency). It is assumed that there is an underlying optimal level of food allocated per individual, exogenously determined by the food bank and denoted by B^*, such that the fixed delivery amount for each agency i is $y_i = n_i B^*$, i.e., proportional to the size of the population served by the agency.

As mentioned, the same fleet of vehicles is used to visit both suppliers and food aid agencies. A vehicle which is assigned to perform a certain gleaning task, i.e., visit a certain supplier on a certain day cannot perform any other tasks on that day because of the constant loading of the harvested produce to the vehicle. Therefore, each vehicle will be assigned daily to one of three possible types of usage states: (1) performing a collection from a single supplier for the entirety of that workday; (2) distributing food from the depot to different food aid agencies; (3) remaining idle at the depot. A vehicle assigned to distribution tasks is restricted by two limitations: first, because of driving and work hour regulations, the total traveling time is limited to L hours a day with respect to a traveling time matrix ℓ_{ij} $(i, j \in D \cup \{0\})$, that accounts for both transportation and fixed loading or unloading times. Second, while the vehicle may leave the depot several times daily (as long as the first restriction is not violated), the sum of the fixed delivery quantities of the sites included in each such tour cannot exceed its limited capacity, denoted by C.

Food can only be distributed after it has undergone processing at the depot. Based on real-life practice, we assume that the processing lasts for exactly one day, that is, if a certain amount of produce is collected on day t, it will undergo processing on day $t + 1$, and will be available for distribution starting day $t + 2$. Therefore, at the beginning of the decision horizon, there may be initial inventory at the depot, which we define using the following notation: α_t – the amount of food that is available for distribution at the beginning of day 0 which expires at the end of day t; β_t – the amount of food which undergoes processing on day 0 (available for distribution starting the beginning of day 1) which expires at the end of day t. Because of its perishable nature, food whose expiration is on day t and that has not been distributed to any agency before the end of that day, is considered inedible and is assumed to leave the inventory so that it is no longer available for distribution from day $t + 1$ and on.

Given this setting, the food bank is required to determine two types of plans that together define a complete solution to the problem: (1) a collection plan, which specifies the subset of suppliers that are visited daily; and (2) a distribution plan, which defines the number of vehicles used for distribution daily, the subset of food aid agencies visited by each of them and the routes they are integrated into.

The food bank's goals in managing the gleaning operations and their representation in the objective function of the H-GDSP merit further discussion. Obviously, due to the humanitarian nature of the problem, budgetary considerations (i.e., costs or profits) are not the primary motivations of the food bank. Rather, it is interested in promoting the following two, possibly colliding, goals: (1) *effectiveness*, i.e., distributing as much food as possible to the food aid agencies; and (2) *equity* – maintain a degree of "fairness" in the allocation of the food among the different agencies. Note that these considerations are measured with respect to the distribution plan, however they are constrained by the decisions regarding the collection plan, such that the combination of the plans defines a complete feasible solution to the problem. Several objective functions that balance these two considerations in various settings have appeared in the literature in recent years. Among them we have adopted the approach introduced by Eisenhandler and Tzur (2019a) for a different problem with a single period setting, and we have extended it to the multi-period setting of the H-GDSP. By this approach, we now measure the effectiveness, denoted by F, as the total amount distributed to all agencies throughout the decision horizon; and measure the inequity using the Gini coefficient, denoted by G with respect to the total allocation to each agency on all days. It was shown that maximizing a combination of these two measures in a multiplicative way, namely $F \cdot (1 - G)$, leads to an objective function that is easy to compute and incorporate in a linear optimization model that promotes desirable properties of the allocation and leads to an adequate balance of the two above mentioned desirable considerations. More formally, we denote the total amount distributed to agency $i \in D$ on all days of the decision horizon as Y_i. Then, as shown in the previous study, the effectiveness can be measured as $F \equiv \sum_{i \in D} Y_i$, and the equity can be measured using the following closed-form formula, which in turn is based on Anand (1983): $1 - G \equiv 1 - \frac{\sum_{i \in D} \sum_{j \in D: j > i} |q_j Y_i - q_i Y_j|}{F}$, where q_i represents agency i's share in the total population served by all agencies, i.e., $q_i = \frac{n_i}{\sum_{j \in D} n_j}$. This leads to the objective function $Z = F \cdot (1 - G) = \sum_{i \in D} Y_i - \sum_{i \in D} \sum_{j \in D: j > i} |q_j Y_i - q_i Y_j|$, which we use in the H-GDSP.

To further clarify and illustrate the input, the decisions and the computation of the objective value of the H-GDSP, a toy instance and a sample solution are demonstrated in Online Appendix A.

While the H-GDSP includes a routing component, which is usually a complicating aspect of optimization problems, it is not a trivial generalization of traditional routing problems because of its special humanitarian objective function. However, the following claim establishes the hardness of the problem:

Claim 1.1 *The H-GDSP is strongly NP-Complete.*

Proof: See Online Appendix B.

2.2 *Mathematical Formulation*

We next provide a formulation of the H-GDSP as a mixed integer linear program (MILP) which covers all three aspects of the problem: the collection, the distribution and the inventory. We start by introducing additional notation based on the problem's input presented in the previous section:

- D^0 is the union of the food aid agency set and the depot, i.e., $D^0 \equiv D \cup \{0\}$.
- N is the set of all supplier and food aid agency sites, i.e., $N = P \cup D$
- η – an upper bound on the number of consecutive days in which food, harvested from any of the suppliers, can remain in the depot following the completion of its processing. This upper bound can be derived based on the suppliers' parameters, by computing $\max_{i \in P} \{d_i - a_i - 1\}$. To justify this expression,

note that the earliest day that food can be collected from supplier i is a_i, and the latest it can be distributed is d_i, and exactly one day is required for processing.

- $\mathbb{T}$ – the set of pairs of days in the planning horizon $t_1, t_2 \in T$, such that food with expiration date on day t_2 can be considered for distribution on day t_1, that is: $\mathbb{T} \equiv \{t_1, t_2 \in T \mid t_1 \leq t_2 - \eta + 1\}$.
- MV_D – the maximal number of delivery sites that can be visited by any vehicle in one day. The value of this parameter can be derived by solving a MILP formulation (see Eisenhandler and Tzur, 2019a).

Next, we present the decision variables we use in the formulation:

- x_{ijkt} – equals 1 if site i precedes site j ($i, j \in D^0, i \neq j$) on the route of vehicle $k \in K$ on day $t \in T$, otherwise 0.
- p_{kt} (d_{kt}) – equals 1 if vehicle k is used for collection (distribution) on day t, otherwise 0.
- v_{ikt} – equals 1 if site $i \in N$ is visited by vehicle $k \in K$ on day $t \in T$, otherwise 0.
- w_{it} – equals 1 if site $i \in N$ is visited on day $t \in T$, otherwise 0, that is, $w_{it} \equiv \sum_{k \in K} v_{ikt}$ (we recall that any pickup site can be visited at most once, and that any delivery site can be visited at most once daily).
- Q_{ijkt} – the load on vehicle $k \in K$ on day $t \in T$ when it travels from site $i \in D^0$ to site $j \in D^0$.
- I_{t_1,t_2} – the amount of produce that expires at the end of day t_2, that is available for delivery in the depot in the beginning of day t_1 ($(t_1, t_2) \in \mathbb{T}$).
- Y_i – the total amount allocated to agency $i \in D$.
- E_{ij} – used to linearize the equity part of the objective function ($i, j \in D, i < j$).

This leads to the following mathematical formulation, which we present one part after the other due to its length.

$$Maximize \;\; Z = \sum_{i \in D} Y_i - \sum_{i,j \in D, i<j} E_{ij} \tag{1}$$

The objective function (1) balances effectiveness and equity in the manner that was previously described, where the E_{ij} variables are defined in Constraints (20) and (21) below. It is optimized subject to the following constraints:

Constraints (2) - (7) focus on the assignment of site visits to days and consider the vehicle availability:

$$p_{kt} + d_{kt} \leq 1 \qquad \forall k \in K, t \in T \tag{2}$$

$$\sum_{i \in P_t} \sum_{k \in K} v_{ikt} = \sum_{k \in K} p_{kt} \qquad \forall t \in T \tag{3}$$

$$\sum_{i \in D} v_{ikt} \leq MV_D \cdot d_{kt} \qquad \forall k \in K, t \in T \tag{4}$$

$$\sum_{k \in K} \sum_{t \in T: i \in P_t} v_{ikt} \leq 1 \qquad \forall i \in P \tag{5}$$

$$\sum_{k\in K} v_{ikt} \leq 1 \qquad \forall i \in D, t \in T \qquad (6)$$

$$w_{it} = \sum_{k\in K} v_{ikt} \qquad \forall i \in N, t \in T \qquad (7)$$

Constraint (2) states that each vehicle is assigned on any given day to either collection or distribution tasks, but not both. Constraint (3) states that the number of suppliers that are visited daily is equal to the number of vehicles that are assigned to collection tasks. Constraint (4) bounds the number of food aid agencies that are visited by any vehicle daily to MV_D under the condition that the vehicle is indeed assigned to distribution tasks on that day. Constraint (5) makes sure that each supplier is visited at most once throughout the decision horizon, and Constraint (6) states that each food aid agency is visited at most once daily. Constraint (7) is used to define the value of the variables w_{it}, whose use is discussed in more detail in Section 3.

Constraints (8) – (10) focus on the sequence of stops of vehicles on the distribution side of the problem:

$$v_{ikt} = \sum_{j\in D^0} x_{ijkt} \qquad \forall t \in T, i \in D, k \in K \qquad (8)$$

$$v_{ikt} = \sum_{j\in D^0} x_{jikt} \qquad \forall t \in T, i \in D, k \in K \qquad (9)$$

$$\sum_{i\in D^0} \sum_{j\in D^0: i\neq j} c_{ij} x_{ijkt} \leq L \cdot d_{kt} \qquad \forall k \in K, t \in T \qquad (10)$$

Constraints (8) and (9) are degree balance constraints. Constraint (10) sets the traveling time constraint for each vehicle that is assigned to distribution tasks.

Constraints (11) – (13) focus on the load on the vehicle for the distribution side of the problem:

$$\sum_{j\in D^0} Q_{jikt} - \sum_{j\in D^0} Q_{ijkt} = y_i v_{ikt} \qquad \forall i \in D, k \in K, t \in T \qquad (11)$$

$$Q_{j0kt} = 0 \qquad \forall j \in D, k \in K, t \in T \qquad (12)$$

$$Q_{ijkt} \leq C \cdot x_{ijkt} \qquad \forall k \in K, t \in T, i, j \in D^0, i \neq j \qquad (13)$$

Constraint (11) makes sure that the load on the vehicle decreases by y_i after agency $i \in D$ is visited. Constraint (12) provides starting conditions for Constraint (11) to break symmetry. Constraint (13) enforces the vehicle capacity restriction. Given that any agency can be visited at most once daily, these constraints also serve to eliminate subtours.

Constraints (14) – (18) focus on the dynamics of the inventory at the depot:

$$\sum_{t_2\in T:(t_1,t_2)\in\mathbb{T}} I_{t_1,t_2} + \sum_{k\in K}\sum_{i\in P_{t_1-1}} S_i v_{i,k,t_1-1} \geq \sum_{i\in D}\sum_{k\in K} y_i v_{ikt_1} + \sum_{t_2\in T:(t_1+1,t_2)\in\mathbb{T}} I_{t_1+1,t_2} \qquad \forall t_1 \in T: t_1 \geq 1 \tag{14}$$

$$\sum_{t_2\in T:(0,t_2)\in\mathbb{T}} I_{0,t_2} + \sum_{t_2\in T:(1,t_2)\in\mathbb{T}} \beta_{t_2} \geq \sum_{i\in D}\sum_{k\in K} y_i v_{ik0} + \sum_{t_2\in T:(1,t_2)\in\mathbb{T}} I_{1,t_2} \tag{15}$$

$$I_{t_1,t_2} \leq I_{t_1-1,t_2} + \sum_{i\in P_{t_1-2}:d_i=t_2}\sum_{k\in K} S_i v_{i,k,t_1-2} \qquad \forall (t_1,t_2) \in \mathbb{T}: t_1 \geq 2 \tag{16}$$

$$I_{0,t_2} \leq \alpha_{t_2} \qquad \forall t_2 \in T: (0,t_2) \in \mathbb{T} \tag{17}$$

$$I_{1,t_2} \leq \begin{cases} \alpha_{t_2} + \beta_{t_2}, & if\ (0,t_2) \in \mathbb{T} \\ \beta_{t_2}, & if\ (0,t_2) \notin \mathbb{T} \end{cases} \qquad \forall t_2 \in T: (1,t_2) \in \mathbb{T} \tag{18}$$

Constraint (14) states that for any day considered the total opening inventory, (represented by the first addend on the left side) and the total amount undergoing processing on that day, (represented by the second addend on the left side and expressed as the amount collected on the day before the one considered) is an upper bound on the sum of the amount distributed on that day, (first addend on the right side) and the opening inventory for the following day (second addend on the right side). The constraint is formulated as an inequality in order to account for the inventory that may perish at the end of this day. Constraint (15) is the equivalent of Constraint (14) for the first day ($t_1 = 0$), where input parameters, rather than decision variables, represent the amount that is undergoing processing on that day. While Constraints (14)-(15) are aggregative over all expiration dates, the following Constraints (16)-(18) link between inventories on different days that have the same expiration date. Constraint (16) states that for each expiration date, the total amount of inventory available on a certain day cannot exceed that of the preceding day, plus the amount that enters the inventory following the completion of its processing on the day before, i.e., collected two days earlier. Constraints (17) and (18) are special cases for the first and second days of the decision horizon, respectively, whereby, similar to Constraint (15), the parameters representing the initial inventory at the depot replace the collection decision variables used in Constraint (16).

Constraints (19) – (21) focus on setting the proper objective value:

$$Y_i = \sum_{k\in K}\sum_{t\in T} y_i v_{ikt} \qquad \forall i \in D \tag{19}$$

$$E_{ij} \geq q_j Y_i - q_i Y_j \qquad \forall i,j \in D, i < j \tag{20}$$

$$E_{ij} \geq q_i Y_j - q_j Y_i \qquad \forall i,j \in D, i < j \tag{21}$$

Constraint (19) refers to the variables that are used to define the effectiveness of the allocation (i.e., the total amount allocated to each agency throughout the decision horizon). Constraints (20) – (21) linearize the absolute value expressions that define the equity of the total allocation.

Finally, Constraints (22) – (29) impose non-negativity and integrality (where necessary):

$$Y_i \geq 0 \qquad \forall i \in D \qquad (22)$$
$$I_{t_1,t_2} \geq 0 \qquad \forall t_1, t_2 \in T \qquad (23)$$
$$E_{ij} \geq 0 \qquad \forall i, j \in D, i < j \qquad (24)$$
$$Q_{ijkt} \geq 0 \qquad \forall k \in K, t \in T, i, j \in D^0, i \neq j \qquad (25)$$
$$p_{kt}, d_{kt} \in \{0,1\} \qquad \forall k \in K, t \in T \qquad (26)$$
$$w_{it} \in \{0,1\} \qquad \forall i \in N, t \in T \qquad (27)$$
$$v_{ikt} \in \{0,1\} \qquad \forall k \in K, t \in T, i \in D \cup P_t \qquad (28)$$
$$x_{ijkt} \in \{0,1\} \qquad \forall k \in K, t \in T, i, j \in D^0, i \neq j \qquad (29)$$

Formulation (1)-(29) provides a valid definition of the H-GDSP. However, it may not provide desirable solutions in practical settings due to its size (see the numerical analysis in Section 7). Therefore, while our proposed solution approach continues to be based on mathematical programming, we have applied two major changes: (1) decomposing the full problem to its constituent collection and distribution sub-problems; and (2) working with route-based rather than site-based routing decision variables. We discuss these changes, which are aimed at reducing the computational effort required to solve the problem, in the following Sections 3 and 4, respectively.

3. **A Decomposition Approach**

We refer to the mathematical model presented in the previous section as the *Full Formulation of the H-GDSP*. However, using it to obtain good feasible solutions for instances of practical sizes does not seem to be likely due to the large number of both decision variables and constraints included. This has also been confirmed by the numerical analysis presented in Section 7. Therefore, in this section we present a decomposition approach, whose main advantage is that it allows to handle smaller, and therefore easier to solve, aspects of the problem. For this purpose, we first define two types of *plans* a solution for the H-GDSP consists of:

1. A *Collection plan,* which states the subset of suppliers that are visited daily.
2. A *Distribution plan,* which states the subset of agencies that are visited daily, the assignment of these agencies to the vehicle, and the vehicle routes in which they are integrated.

We observe that a complete feasible solution to the H-GDSP consists of a collection plan and a distribution plan, and that the objective value depends only on the distribution plan. However, note that a distribution plan is admissible only if it can be accompanied by a corresponding collection plan that guarantees a sufficient amount of inventory at the depot daily. Despite this dependence, the approach we propose is based on a decomposition of the problem to its collection and distribution aspects, such that each plan is obtained separately by solving the corresponding *sub-problem*. The two sub-problems are then linked through certain elements of the solution procedure, which establish that the resulting collection and distribution plans correspond to each other, so that together they define a feasible solution to the complete problem. In what follows, we first define these sub-problems formally and discuss how to solve them, and then we explain the manner by which they are linked together.

3.1 *The Distribution Aspect*

We start with the distribution aspect of the problem, in which vehicle routes that provide food from the depot to the food aid agencies are planned for each day in the decision horizon, subject to constraints on traveling time, vehicle capacity and fixed delivery amounts to the agencies. The objective function of this optimization sub-problem is the same as the one used for the complete problem, i.e., balancing the effectiveness and the equity of the total allocation to the agencies throughout the decision horizon.

The distribution sub-problem can be formulated and solved as a MILP. This program, which we refer to as the *Distribution Formulation* (DF for short), can be obtained from the formulation of the full problem, presented in the previous section, by removing all of the decision variables and the constraints which are related to either the collection aspect of the problem or the inventory at the depot. Specifically, the DF does not include the decision variables p_{kt} and v_{ikt} for $i \in P$, nor constraints (14)-(18) above.

While this formulation is smaller in its dimensions compared to the full formulation, it excludes any restriction on the amounts that leave the depot for distribution and that are based on the collection aspect of the problem. Therefore, in most practical settings, it will yield an infeasible solution with an excessively high objective value, in which the amount to be delivered on some day (one or several) may not be available at the depot. To mitigate this issue, we next suggest as a possible course of action to add certain valid inequalities into the DF. These inequalities integrate back into the formulation some elements of the collection aspect, but without bringing back all of the variables and the constraints that were removed initially. For example, it is easy to bound the total amount distributed daily by the total amount of supplies which are available for collection up to two days before that day (to account for the processing time), although without giving room in the formulation to determine whether they should be collected or not. This can be easily stated as the following constraint:

$$\sum_{i\in D}\sum_{\tau=0}^{t} y_i w_{i\tau} \leq \begin{cases} \sum_{t_2\in T}\alpha_{t_2} & t=0 \\ \sum_{t_2\in T:t_2\geq 1}\alpha_{t_2}+\sum_{t_2\in T}\beta_{t_2} & t=1 \\ \sum_{t_2\in T:t_2\geq t}\alpha_{t_2}+\sum_{t_2\in T:t_2\geq t}\beta_{t_2}+\sum_{i\in P:a_i\leq t-2} S_i & t\geq 2 \end{cases} \qquad \forall t\in T \qquad (30)$$

Another kind of limitation that can be added refers to the fact that a vehicle that is assigned to collect any supply on a certain day cannot be used for any other task on that day. Therefore, on each day, if exactly j vehicles are used for collection, then at most $|K|-j$ vehicles may be used for distribution. This serves to simultaneously limit both the total supply that can be collected on any day and the total number of vehicles that can used for distribution on the same day. For this aim, we denote by π_t the number of suppliers that can be visited up to day $t\in T$, i.e., $\pi_t \equiv |\{i\in P: a_i\leq t\}|$; and by σ_{jt} the total potential supply up to day $t\in T$ if the $j=0,\dots,\pi_t$ largest supplies are collected, that is, the sum of their sizes plus the opening inventories. Then, the total amount distributed up to day t is at most $\sigma_{j,t-2}$, if exactly j suppliers are visited up to day $t-2$. To state this formally within the mathematical formulation, we use an additional set of decision variables. We note that the addition of these binary variables to the formulation may seem counter-intuitive given that the aim of the decomposition was to reduce the dimensions of the mathematical formulation. However, this will benefit the overall solution approach, to be presented in the following sections, in limiting its ability to come up with solutions whose objective value is excessively high, thus reducing the total computational effort required. Formally, we use the decision variable f_{jt}, which equals 1 if exactly j suppliers are visited up to day t, and then add the following constraints:

$$\sum_{j=0}^{\pi_t} f_{jt} = 1 \qquad \forall t\in T \qquad (31)$$

$$\sum_{\tau=0}^{t} y_i w_{i\tau} \leq \sum_{j=0}^{\pi_t} f_{j,t-2}\,\sigma_{j,t-2} \qquad \forall t\in T: t\geq 2 \qquad (32)$$

$$\sum_{j=0}^{\pi_t} f_{jt} + \sum_{k\in K}\sum_{\tau=0}^{t} d_{kt} \leq |K|\cdot t \qquad \forall t\in T \qquad (33)$$

Constraint (31) makes sure that the total number of suppliers that are visited daily is unique. Constraint (32) restricts the total amount distributed daily in the manner explained above, conditional upon the number of suppliers visited up to two days beforehand. Constraint (33) states that the total number of vehicles, used for collection as well as distribution, is bounded by the total number of vehicles that are available. The combination of these constraints states the trade-off in terms of vehicle usage: as the number of vehicles used for collection increases, the amount available for distribution can also increase, but this comes at the expense of the number of vehicles which can be used for distribution, and therefore on the number of agencies that can be visited.

While the addition of these constraints limits the ability of the formulation to provide infeasible solutions, it still does not guarantee the feasibility of the obtained solution. We discuss how to accomplish this in the next sub-section.

3.2 *The Collection Aspect*

The purpose of the *Collection Feasibility Sub-problem (CFS)* is to provide a feasible collection plan so as to meet a pre-defined set of daily distribution amounts. It follows the definition of the full problem, with the exception that the distribution aspect is completely removed from it, and the following aggregative parameters constrain the sub-problem:

- Y_t - the total amount that needs to be delivered to all agencies combined on day $t \in T$. The vector of all of these parameters is denoted by $\vec{Y}$.
- K_t - the maximal number of vehicles that can be used to visit suppliers on day $t \in T$. The vector of all of these parameters is denoted by $\vec{K}$.

In the next section, we discuss the manner in which the values of these parameters is set. Assume for the following discussion that they are given.

The decision problem of the CFS is to determine whether there exists a feasible collection plan that satisfies the restrictions of the parameters $\vec{K}$ and $\vec{Y}$, namely: (1) it does not violate the maximal number of visits to suppliers daily; and (2) it guarantees that daily there will be a sufficient amount of inventory at the depot to meet the required daily demand. Essentially, it incorporates elements from the well-studied classes of lot sizing and scheduling problems. However, it differs from both of them significantly: on the one hand, this problem does not include the typical cost structure of lot sizing problems, namely, fixed ordering and inventory holding costs. On the other hand, scheduling problems do not usually involve demands that occur on more than one period or inventory management decisions that are aimed at fulfilling these demands. Specifically, in the context of scheduling, the problem is most similar to discrete interval scheduling problems in which a discrete set of possible starting times (equivalent to the collection time window in our problem) is provided for each task. An interested reader is referred to Kolen et al. (2007), in which different variants, applications and algorithms in this context are surveyed. Here we only note that the maximization of the number of tasks that can be performed given a fixed number of machines, as well as the minimization of the number of machines that allow to perform all tasks, have both been proven to be NP-Hard (e.g., Spieksma, 1999; Chuzhoy and Naor, 2006). The CFS is easier from some aspects, e.g., unit-time processing, but more complex from other aspects, such as a non-stationary number of "machines" (i.e., vehicles) and inventory management considerations such as expiration. Thus, we were not able to determine its theoretical complexity.

As part of our proposed solution approach, we are required to address the Collection Feasibility Sub-problem. We tackle an instance of the problem by solving a MILP without an objective function (this MILP has no feasible solution if and only if the answer to the decision problem is that no appropriate collection plan exists). With this aim, we use a slightly different modeling approach than the one presented in the full formulation. Specifically, we use the continuous decision variables A_{i,t_3,t_4} to represent the amount that is collected from supplier $i \in P$ on day $t_3 \in T$ (such that $a_i \leq t_3 \leq b_i$) and distributed on day $t_4 \in T$ such that $t_3 + 2 \leq t_4 \leq d_i$). To represent the use of the initial inventories, we denote by B_{0,t_4,t_5} $\left(B_{1,t_4,t_5}\right)$ the amount of produce that is available for distribution at the depot on day 0 (that undergoes processing at the depot on day 0 and that will be prepared for distribution on day 1), which is distributed on day t_4 and expires on day t_5. This leads to the following formulation of the CFS:

$$\sum_{t:i\in P_t} w_{it} \leq 1 \qquad \forall i \in P \tag{34}$$

$$\sum_{i\in P_t} w_{it} \leq K_t \qquad \forall t \in T \tag{35}$$

$$\sum_{t_4=0}^{t_5} B_{0,t_4,t_5} \leq \alpha_{t_5} \qquad \forall t_5 \in T \tag{36}$$

$$\sum_{t_4=1}^{t_5} B_{1,t_4,t_5} \leq \beta_{t_5} \qquad \forall t_5 \in T: t_5 \geq 1 \tag{37}$$

$$\sum_{t_4=t_3+2}^{d_i} A_{i,t_3,t_4} \leq S_i w_{i,t_3} \qquad \forall i \in P, t_3 \in T \tag{38}$$

$$\sum_{t_5\in T} B_{0,0,t_5} = Y_0 \tag{39}$$

$$\sum_{t_5\in T:t_5\geq 1} B_{0,1,t_5} + \sum_{t_5\in T:t_5\geq 1} B_{1,1,t_5} = Y_1 \tag{40}$$

$$\sum_{j=0}^{1} \sum_{t_5\in T:t_5\geq t_4} B_{j,t_4,t_5} + \sum_{t_3=0}^{t_4-2} \sum_{i\in P:d_i\geq t_4} A_{i,t_3,t_4} = Y_{t_4} \qquad \forall t_4 \in T: t_4 \geq 2 \tag{41}$$

$$w_{it} \in \{0,1\} \qquad \forall t \in T, i \in P_t \tag{42}$$

$$A_{i,t_1,t_2} \geq 0 \qquad \forall t_1 \in T, i \in P_{t_1}, t_2: t_1 + 2 \leq t_2 \leq d_i \tag{43}$$

$$B_{i,t_4,t_5} \geq 0 \qquad \forall j = 0,1, t_4 \in T, t_5 \in T: t_5 \geq t_4 \tag{44}$$

Constraint (34) assures that each supply is scheduled to be collected on one day at most. Constraint (35) enforces the daily vehicle availability. Constraint (36) establishes the manner in which the initial inventory that is available at the depot on day 0 is distributed on different days (it is formulated as an inequality to account for the amount that is collected but not distributed since it perishes). Constraint (37) is similar with respect to the initial inventory that is ready for distribution on day 1. Constraint (38) is analogous to Constraints (36) and (37) with respect to the collection from suppliers. Constraint (39)-(41) establish that demands are satisfied on days 0, 1 and from day 2 and onwards, respectively. Constraints (42)-(44) are integrality and non-negativity constraints.

While using this formulation to solve the CFS does not provide any guarantees regarding the theoretical worst time complexity, we have observed that the CFS instances encountered in practice, even in larger H-GDSP instances, were solved in a very short time. This low computational burden can be explained by the fact that its input dimensions are low, even when the input size of the complete instance is large. This is due to the fact that the number of days, vehicles and suppliers tends to be quite small, whereas the food aid agencies, which often appear in larger numbers, are not included in the input of the CFS.

Nevertheless, we propose an enhancement that may reduce even further the computation time for determining the CFS, and additionally can provide useful insights that can be used in the solution approach for the full problem. With this aim, consider an optimization version of the CFS, in which given the same input, the objective is to determine the earliest day whereby the demand cannot be fully supplied. In the case that there exists a collection plan that can allow the given distribution plan, then any arbitrary day that is beyond the decision horizon can be returned, e.g., the first day after the end of the decision horizon, namely, day $|T|$. Note this enhancement breaks much of the problem symmetry and therefore allows obtaining a solution in a shorter time.

To solve this optimization version of the CFS, we adapt the MILP formulation of the decision version of the CFS, i.e., (34)-(44), to incorporate the new objective function. For this purpose, we add the following decision variables:

- H_t – the shortage on day $t \in T$, i.e., the amount that *cannot* be supplied on that day
- h_t – equals 1 if the demand of day $t \in T$ *cannot* be supplied in full, otherwise 0
- τ – the earliest day whose demand cannot be supplied in full (i.e., the objective value of the optimization version of the CFS)

We then add the following objective function:

$$Maximize \quad \tau \tag{45}$$

We also change Constraints (39)-(41) by subtracting the variables representing the shortage from their right hand side in the following way:

$$\sum_{t_5 \in T} B_{0,0,t_5} = Y_0 - H_0 \tag{46}$$

$$\sum_{t_5 \in T : t_5 \geq 1} B_{0,1,t_5} + \sum_{t_5 \in T : t_5 \geq 1} B_{1,1,t_5} = Y_1 - H_1 \tag{47}$$

$$\sum_{j=0}^{1} \sum_{t_5 \in T : t_5 \geq t_4} B_{j,t_4,t_5} + \sum_{t_3=0}^{t_4-2} \sum_{i \in P : d_i \geq t_4} A_{i,t_3,t_4} = Y_{t_4} - H_{t_4} \qquad \forall t_4 \in T : t_4 \geq 2 \tag{48}$$

We then add the following constraints:

$$H_t \leq Y_t \cdot h_t \qquad \forall t \in T \tag{49}$$

$$\tau \leq t + (|T| + 1 - t) \cdot (1 - h_t) \qquad \forall t \in T \tag{50}$$

$$\tau \leq |T| + 1 \tag{51}$$

$$\tau, H_t \geq 0 \qquad \forall t \in T \tag{52}$$

$$h_t \in \{0,1\} \qquad \forall t \in T \tag{53}$$

To sum up, the formulation of the optimization version of the CFS consists of the objective function (45), Constraints (34)-(38), (42)-(44) and (46)-(53). To see how this adaptation works, note that the objective function seeks to maximize τ, which is bounded by $|T| + 1$ in Constraint (51). If the supply can be met in full daily, then Constraint (49) implies $H_t = h_t = 0$ for all $t \in T$, and consequently Constraints (50) and (51) imply $\tau = |T| + 1$. By contrast, if on a certain day t' the demand is not supplied in full, then $H_{t'}$ must take a positive value, which implies $h_{t'} = 1$, and therefore by Constraint (50), τ is bounded by $t' < |T| + 1$. Therefore, the decision of the CFS is *FALSE* iff $\tau = |T| + 1$, and if it is *TRUE* then τ is lower than $|T| + 1$ and its value represents the earliest day in which the demand cannot be fully supplied.

3.3 *Linking the Sub-Problems*

The previous sub-sections have focused on the distribution and the collection sub-problems, describing how to cope with each of them separately. Next, we describe the solution procedure in its entirety and explain how the repeated interplay between the sub-problems leads to a complete feasible solution to the H-GDSP.

Consider first an optimal solution to the DF. Recall that this distribution plan is not necessarily feasible for the complete problem because there does not necessarily exist a supporting collection plan, therefore we refer to it as a candidate distribution plan. Subsequently, we denote by W the subset of visits to agencies on specific days scheduled by the candidate distribution plan, i.e., $W \equiv \{(i,t): i \in D, t \in T, w_{it} = 1\}$. Let $\mathbb{W}$ be the set of all such subsets, that is, every distribution plan that does not violate the traveling time and capacity constraints, irrespectively of the availability of inventory to fulfill the required demands. Consequently, a candidate distribution plan can take part in a feasible solution to the H-GDSP only if it can be accompanied by a corresponding collection plan. Therefore, given a distribution plan, we generate a CFS instance, whose input is based on the candidate distribution plan by computing the total demands and the number of vehicles available for daily collection. These parameters, as defined in the previous section, are derived from the value of the respective decision variables in the optimal DF solution in the following manner: $Y_t = \sum_{i \in D} \sum_{k \in K} y_i v_{ikt}$ and $K_t = K - \sum_{k \in K} d_{kt}$. By computing this, we obtain the vectors $\vec{Y}(W)$ and $\vec{K}(W)$.

Let $\Phi_{\vec{Y}(W),\vec{K}(W)}$ be a binary parameter, which represents the solution of the CFS given the vectors of required demands $\vec{Y}(W)$ and vehicle availability $\vec{K}(W)$, that is, $\Phi_{\vec{Y}(W),\vec{K}(W)} = 1$ if and only if a corresponding collection plan exists, given the subset of visits to agencies specified by W, and otherwise it equals 0. Then note that the following constraint is valid for the distribution sub-problem in terms of the full H-GDSP, since it states that only distribution plans that can be accompanied by corresponding collection plans can be chosen:

$$\sum_{(i,t) \in W} w_{it} \leq |W| - 1 + \Phi_{\vec{Y}(W),\vec{K}(W)} \qquad \forall W \in \mathbb{W} \qquad (54)$$

To see how this constraint works, consider its RHS and note that any subset of visits to agency W can be implemented iff $\Phi_{\vec{Y}(W),\vec{K}(W)} = 1$; otherwise, at most $|W| - 1$ (out of $|W|$) visits to agencies can be scheduled and therefore W is eliminated.

Furthermore, by solving the optimization version of the CFS (rather than the decision version), an even stronger version of Constraint (54) can be obtained. Recall that the optimal objective value of this problem is the earliest day in which the required demand cannot be supplied by any collection plan, denoted $\tau(W)$ given the subset of agency visits plan, W. Now, let $W_{[t']}$ denote the subset of agency visits up to day t' specified by W, i.e., $W_{[t']} \equiv \{(i,t) \in W : t \leq t'\}$. Then, note that the following constraint is also valid:

$$\sum_{(i,t) \in W_{[\tau(W)]}} w_{it} \leq \left|W_{[\tau(W)]}\right| - 1 + \Phi_{\vec{Y}(W),\vec{K}(W)} \qquad \forall W \in \mathbb{W} \qquad (55)$$

Constraint (55) is valid since if W is inadmissible because the demands on days $0, \dots, \tau(W)$ cannot be fulfilled, then any other distribution plan that shares the same subset of visits on these days is also inadmissible, regardless of the subset of visits it dictates on days $\tau(W) + 1, \dots, |T| - 1$. This constraint eliminates a possibly larger number of distribution plans, that is, any distribution plan that shares the same subset of visits as W on days $0, \dots, \tau(W)$. Therefore, Constraint (55) is stronger than Constraint (54).

Nonetheless, applying Constraints (54) or (55) directly raises technical challenges, since the number of distribution plans is theoretically exponential and so there is no practical way to explicitly include them in the formulations in full. Consequently, we implement an iterative procedure, which is described in Figure 1 below, and works as follows: we start by obtaining an optimal solution to the DF and then solve its corresponding (decision or optimization version of the) CFS. If a feasible corresponding collection plan is found, then clearly the two complementing plans define an optimal solution to the H-GDSP, and therefore the procedure terminates. Otherwise, the DF solution represents a distribution plan that is not feasible, since no collection plan can guarantee enough daily supplies at the depot in the decision horizon, and therefore it needs to be eliminated. To implement this, we add the respective Constraint (54) or (55) to the formulation and resolve. Note that this procedure may require solving the DF several times which may cause computational difficulties. They can be somewhat relieved by handling the additional constraints in a lazy manner as part of a Branch-and-Cut framework (i.e., solving the formulation in one run and verifying that no constraints are violated whenever a new integral solution is obtained), rather than resolving the updated DF "from scratch" in each iteration.

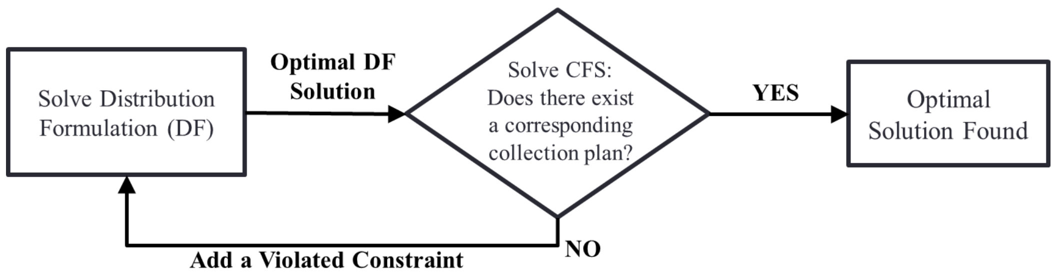


Figure 1: Procedure of the decomposition approach

Applying the above-described decomposed approach rather than the complete formulation may assist in dealing with complex instances of the H-GDSP, especially when the number of food aid agencies is high. However, as will be shown as part of the numerical analysis presented in Section 7, there are still many instances in which the former approach does not have a clear computational advantage over the latter. Therefore, in the next section we discuss an enhanced version of the decomposed approach, which allows obtaining significantly better solutions in the same amount of running time.

4. A Route-Based Approach

The previous section focused on a decomposition approach as part of which the distribution sub-problem was formulated in terms of selecting and sequencing delivery *sites* for each vehicle daily. In this section, we present an alternative approach for the distribution sub-problem, by which decisions are made with respect to complete routes, rather than specific sites.

For this purpose, we start by defining a *distribution route* r as an ordered sequence of delivery sites that starts and ends at the depot. In this regard, we denote by i_j the j-th delivery site visited by the vehicle. Depending on the context, we might also refer to a distribution route in terms of the *subset* of delivery sites that are included in the route. Therefore, the notation $i \in r$ will be used to indicate that a specific delivery site $i \in D$ is included in route r; and similarly $|r|$ will represent the number of delivery sites included in the route. Then, we denote the total amount supplied in route r as $\tilde{y}_r \equiv \sum_{i \in r} y_i$ and the total traveling time it requires as $\tilde{\ell}_r \equiv \left(\ell_{0,i_1} + \sum_{j=1}^{|r|-1} \ell_{i_j,i_{j+1}} + \ell_{i_{|r|},0}\right)$. Note that any route r is feasible only if it satisfies $\tilde{y}_r \leq C$ and $\tilde{\ell}_r \leq L$, thus we denote the set of such feasible routes by $\mathbb{R}$. Also note that by the definition of the H-GDSP, any vehicle is allowed to perform more than one route on any particular day as long as: (1) the total demand of each route does not exceed the vehicle capacity; and (2) the sum of the durations of all of the routes daily assigned to any vehicle combined does not exceed L. In this case, it does not matter which route is performed first, since this does not affect the feasibility or the objective value of the solution.

A route-based mathematical formulation can then be developed for the distribution sub-problem. For this purpose, we define the decision variables $\tilde{x}_{rkt}$ which equal 1 if route $r \in \mathbb{R}$ is assigned to vehicle $k \in K$ on day $t \in T$, and 0 otherwise. In addition to these variables, we also use the decision variables d_{kt}, w_{it}, Y_i and E_{ij}, as defined in Section 2. Solving the distribution sub-problem then amounts to maximizing the objective function (1) subject to constraints (22), (24), (26)-(27), and the following additional constraints:

$$\sum_{r \in \mathbb{R}} \tilde{\ell}_r \tilde{x}_{rkt} \leq L \cdot d_{kt} \qquad \forall k \in K, t \in T \qquad (56)$$

$$w_{it} = \sum_{r \in \mathbb{R}:\, i \in r} \sum_{k \in K} \tilde{x}_{rkt} \qquad \forall i \in D, t \in T \qquad (57)$$

$$Y_i = \sum_{k \in K} \sum_{t \in T} y_i w_{it} \qquad \forall i \in D \qquad (58)$$

$$\tilde{x}_{rkt} \in \{0,1\} \qquad \forall r \in \mathbb{R}, k \in K, t \in T \qquad (59)$$

Constraint (56) makes sure that no vehicle on any day violates the traveling time restriction. Note that since the capacity restriction applies to each individual route, it is covered by the fact that only feasible routes are initially considered and it does not need to be addressed explicitly in the formulation. Constraint (57) assigns proper values to the w_{it} variables. It also precludes split deliveries to any delivery site on any particular day by making sure it is not visited by more than one vehicle. Constraint (58) sets the proper value for the total amount allocated to each agency throughout the decision horizon, namely Y_i. Finally, Constraint (59) replaces Constraint (29) and ensures integrality of the respective decision variables. We refer to this model as the *route-based formulation* for the distribution sub-problem (R-DF for short).

The decomposition approach described in the previous section can still be applied when the R-DF is used to solve the distribution sub-problem. In this case, the optimal solution of the route-based formulation can be used to define the input of the CFS, as follows: $K_t = K - \sum_{k \in K} d^*_{kt}$ (as before) and $Y_t = \sum_{r \in \mathbb{R}} \sum_{k \in K} \tilde{y}_r \tilde{x}^*_{rkt}$. The CFS can then be solved in the same manner described above to verify that a corresponding collection plan exists. In case no such plan exists, Constraints (54) or (55) can be used to eliminate infeasible distribution plans. As for the elimination of inadmissible solutions, note that one might consider performing this elimination, when needed, based on the *routes* that are included in the distribution plan, rather than based on the *sites* that are included. However, the elimination based on sites is stronger since it eliminates distribution plans regardless of the way the sites are sequenced.

We stress that the R-DF provides a valid representation of the distribution sub-problem only if it considers *all* of the feasible routes, namely, the set $\mathbb{R}$. However, this may cause tractability issues as the cardinality of this set, and therefore the number of decision variables, grows exponentially with the number of food aid agencies. The standard approach in this case is to use a Branch-and-Price framework, by which a small subset of routes is used initially, and new routes are gradually inserted into the formulation using a column generation technique. The difficulty in implementing such an approach for the H-GDSP can be traced back to the pricing stage, aimed at identifying decision variables with potential contribution to the objective value. To be effective, the auxiliary sub-problem solved for this purpose needs to have some special structure that allows it to be solved efficiently. However, this requirement is not satisfied in the H-GDSP, which poses difficulties with respect to its implementation.

Therefore, we suggest using a different approach, by which we use a heuristically pre-generated subset of routes, denoted by $\mathcal{R} \subset \mathbb{R}$. By determining $\mathcal{R}$ in any way seen fit by the decision makers, they can control the dimensions of this formulation, which we refer to as the *restricted* R-DF, and thus handle tractability issues. A particularly beneficial consequence is that the size of the formulation does not depend on the size of the instance and specifically on the number of sites included. Another advantage of having a restricted set of routes to choose from is that this naturally reduces the amount of distribution plans that are considered as part of the iterative decomposition approach for the H-GDSP. On the other hand, this raises the challenge of designing a mechanism for the generation of a relatively small, yet sufficiently useful set of routes, such that the final obtained solution will be of high quality. In Appendix C we provide the details of the procedure we use to generate the subset of routes for our suggested solution approach. Once $\mathcal{R}$ is determined in such a fashion, the solution method described in Section 3 can be applied with no further changes, leading to a route-based decomposition approach. In Section 7, we compare the performance of the different solution approaches for the H-GDSP.

5. **Upper Bounds**

Because of the NP-Hardness of the H-GDSP, as was previously shown, it is unexpected that an optimal solution be obtained in instances of realistic characteristics within a reasonable amount of time by using mathematical programming. Therefore, in this section we discuss several methods to obtain upper bounds for the problem, which are required for the computation of optimality gaps.

The first direction we suggest for this purpose is based on valid formulations for the problem, namely the first two solution approaches we have suggested in the previous sections (which rely on the complete formulation and the decomposed site-based formulation, respectively). Note that when terminating each of these formulations after a fixed amount of time, a valid upper bound can be obtained from their respective branch-and-bound tree. Therefore, these approaches were allocated a relatively long running time (e.g., 24 hours per instance), consequently obtaining tighter bounds. We note that these prolonged running times are not aimed at obtaining improved feasible solutions, all the more so in the daily operational setting faced by the food bank in real life. The bounds that are obtained in this manner are denoted by UB_1 and UB_2, respectively.

Unfortunately, even though computing these bounds requires a significant amount of computational resources, we have observed that they result in values that are not sufficiently tight in larger instances. Therefore, we next present two additional dedicated methods to compute bounds for the H-GDSP. They rely on solving MILP formulations of distinct problems, which are obtained from the original problem by relaxing different aspects included in it, such that the resulting problem is easier to solve than the H-GDSP and its solution provides a valid bound. We note that these methods, which are described below, were designed to tackle the two complicating factors of the H-GDSP, namely, determining vehicle routes and the perishability of the inventory. Therefore, in the first method, we have relaxed the collection part of the problem as well as the perishability component, and replaced the routing aspect with decisions that are related to them but are easier to make. In the second method, we have addressed the collection aspect, but the routing part of the problem was completely relaxed, and the inventory management decisions were partially relaxed. For brevity, we next describe the sketch of these methods and leave the full formulations and procedure descriptions for Appendix D.

The first of these additional bounds, to be denoted by UB_3, is computed using an iterative procedure which works as follows: we iterate over the number of suppliers we visit throughout the decision horizon, denoted as $K_p = 1, \dots, \min\{|P|, |K||T|\}$. For each value of K_p, we make the following assumptions, aimed at making the ensuing sub-problem easier to cope with, while still retaining the validity of the resulting bound: (1) the K_p suppliers with highest supply are visited; (2) all of their supply, denoted by σ_{K_p}, is available for distribution at the beginning of the first day; and (3) the expiration date of this entire amount is on the last day of the decision horizon. Given K_p, on the distribution side of the problem, the maximal number of vehicles available for distribution throughout the decision horizon is $K_d \equiv |K||T| - K_p$. Then, instead of determining a specific route for each vehicle, we assign to it sites in a manner that satisfies a knapsack-like constraint with respect to the total traveling time. Specifically, we denote the minimal time required to include site $i \in D \cup \{0\}$ in any route by $\ell'_i \equiv \min_{j \in (D \cup \{0\}) \setminus \{i\}} \{\ell_{ij}, \ell_{ji}\}$. Then, assigning site $i \in D$ to any vehicle on any day, regardless of how it is integrated into the specific vehicle route, takes up at least ℓ'_i units of time out of the total available time, which is at most $L - \ell'_0$. Since the routing aspect is relaxed, optimizing these assignment decisions under our objective function is easier than solving the H-GDSP. Solving this formulation, which is presented in full in Appendix D, for a given value of K_p, leads to a value Z_{K_p}, which is either an optimal value under the constraint on K_p, or a respective upper bound. By maximizing over K_P, we obtain a valid bound for the H-GDSP, which we denote as $UB_3 \equiv \max_{K_p} \left\{Z_{K_p}\right\}$.

In the second additional dedicated bound, we make the following assumptions: (1) on the distribution side, we replace the traveling time limitation for each vehicle by the upper bound MV_D on the number of agencies that can be visited by any vehicle; (2) on the inventory side, we relax the perishability assumption, that is, we assume that supplies never become spoiled and only go out of the inventory when they are distributed. Under the latter assumption, all of the units in the inventory can be aggregated with no necessity to consider different expiration dates, which makes this problem easier to solve. However, the collection of supplies still needs to be scheduled under the vehicle availability limitation, and supplies only enter the inventory after their collection and their ensuing processing at the depot. The optimal solution of this problem is a valid bound for the H-GDSP, which we denote as UB_4.

Finally, we note that for each instance of the H-GDSP, we use the tightest among these bounds, i.e., $UB = \min_{i=1,\dots,4} \{UB_i\}$.

6. **A Rolling Horizon Framework**

As mentioned earlier, one of the special features of the H-GDSP is the importance of the temporal aspect in it. By its very definition, the problem requires the decision maker to perform multi-period planning, which arises primarily since the food bank needs to consider as part of its planning the following input elements from the suppliers: (1) a time window during which they are willing to have their fields gleaned; and (2) an expiration date following which their produce is considered inedible. In practice, it is often the case that the donor farmers provide information regarding their donations well in advance, and that the expiration of the produce can occur several days up to a week following its collection by the food bank. Therefore, the time range for which the outlook for the future is provided in the input of an instance, to which we refer as the *planning horizon*, can generally be quite long in realistic instances.

On the one hand, having a long planning horizon is desirable since the availability of data for a longer period can lead to improved solutions, and consequently, to improved objective value in the long run. On the other hand, a problem instance with a larger number of days also has a larger solution space, and therefore solvability issues may arise, ultimately leading to inferior solutions. For this reason, it is common practice in these cases to plan for a shorter horizon, hence referred to as the *decision horizon*, and use a *rolling horizon* procedure. To explain this, we denote the number of days in the decision horizon (i.e., the days we actively plan for) as τ, such that $\tau \leq |T|$ (we note that in this context T is the planning horizon).

Thus, in the beginning of the first day, a complete solution for days $0, \dots, \tau - 1$ is obtained, however only the decisions for the first day are implemented. In the beginning of the second day, a plan is obtained for days $1, \dots, \tau$, and this procedure goes on such that the horizon is "rolled over" every day. Note that to adequately bind the decisions made on different days, it is advisable to consider of the recent service history. For this purpose, we denote by θ the number of days before the first day in the current decision horizon, that we consider as "recent". Then, the total allocation for each agency for the purposes of evaluating the allocation consists not only of the amount allocated to it in the current planning, but also of that allocated to it during the previous θ days (given as input).

Next, we discuss the effects of choosing a value for the parameter τ, namely, the number of days considered in the current decision horizon. Clearly, as τ decreases, the computational effort required to solve any instance decreases as well. However, this setting may also entail certain difficulties, which are a result of the fact that in the H-GDSP, any donation becomes available for distribution only two days after its collection from the donor. Consequently, setting $\tau < 3$ does not make sense in the context of this problem. Moreover, for $3 \leq \tau < |T|$, using the rolling horizon framework is a heuristic approach, which may result in solutions that are not optimal in the long run. This is again the result of the fact that the distribution of any supply can occur at the very earliest at least two days after its collection, which causes no visits to suppliers to ever be scheduled in the last two days of any decision horizon. This dynamic might cause a tendency to schedule visits to suppliers sub-optimally when solving for the decision (rather than planning) horizon, and to consequently obtain poor distribution plans, which result in a poor objective value for the longer run. The difficulties which arise as part of the use of shorter decision horizons are documented in the literature as a phenomenon which has been denoted as the "end of horizon effect" (see, for example, Federgruen and Tzur, 1999; Chand et al., 2002).

Therefore, in an effort to improve the performance of the rolling horizon approach, we introduce an enhancement whose aim is to prevent a premature commitment to undesired distribution plans because of the end-of-horizon effect on the collection side of the problem. For this aim, we suggest a policy, denoted the *designation of vehicles for future collection,* by which some vehicles are not allowed to be used for distribution purposes in the last two days of each decision horizon, i.e., the days on which supplier visits do not occur because of the end-of-horizon effect. To determine the specific number of designated vehicles, we note that if it is too low it might not achieve its goal, and if it is too high, it might become too restrictive. Thus, we use the following method to determine the number of designated vehicle: given an instance to be solved as part of the rolling horizon framework, we denote by P_L (P_S) the subset of suppliers whose time windows for collection include the last (second-to-last) day of the decision horizon but not the one preceding (following) it. Similarly, we denote by P_B the subset of suppliers whose time windows for collection include both the last and second-to-last days. We also denote by K_P^L and K_P^S the number of vehicles designated for collection on the last and second-to-last days, respectively. Then, to set the values of K_P^L and K_P^S, we use a simple two-step process, which is described below in pseudo-code, and which works as follows: first, we determine in Lines 1 and 2 the vehicle designation due to suppliers that can only be visited on the respective days. The amount of designated vehicles for each of these cases is the number of suppliers that can be visited only on the respective day, or half of the number of available vehicles if the former number exceeds this. Then, in the second step we also account for suppliers available on both of these days: we designate vehicles on the last day before the second-to-last day, to prevent wastage of units that perish on the second-to-last day. Lines 4-6 define the number of additional suppliers that should be accounted for by designating vehicles for them on the last day, denoted ΔK_P^L. Lines 7-9 provide a similar definition for the second-to-last day, where the adequate value for ΔK_P^L used in Line 8 is provided either by Line 3 or by Line 5.

Method for Determining the Number of Vehicles Designated for Collection	
1	$K_P^L = \min\left\{\left\lceil \frac{1}{2}\lvert K\rvert \right\rceil, \lvert P_L\rvert\right\}$
2	$K_S^L = \min\left\{\left\lceil \frac{1}{2}\lvert K\rvert \right\rceil, \lvert P_S\rvert\right\}$
3	$\Delta K_P^L = 0$
4	**If** $K_P^L < \left\lceil \frac{1}{2}\lvert K\rvert \right\rceil$:
5	$\quad \Delta K_P^L = \min\left\{\left\lceil \frac{1}{2}\lvert K\rvert \right\rceil - K_P^L, \lvert P_B\rvert\right\}$
6	$\quad K_P^L = K_P^L + \Delta K_P^L$
7	**If** $K_P^S < \left\lceil \frac{1}{2}\lvert K\rvert \right\rceil$ **and** $\lvert P_B\rvert - \Delta K_P^L > 0$:
8	$\quad \Delta K_P^S = \min\left\{\left\lceil \frac{1}{2}\lvert K\rvert \right\rceil - K_P^S, \lvert P_B\rvert - \Delta K_P^L\right\}$
9	$\quad K_P^S = K_P^S + \Delta K_P^S$

7. Numerical Analysis

In this section we present the experiments we have performed to assess the quality of the proposed methods for the H-GDSP. The section is divided into three parts: first, in Section 7.1 we compare our suggested solution approach to the other more naïve approaches, and in Section 7.2 it is compared to real life practice; then, in Section 7.3, the rolling horizon framework is analyzed.

7.1 *Comparison of Solution Approaches*

The analysis was performed based on two datasets: first, 10 real life instances were created based on data from real life working days of the Israeli food bank "Leket Israel" in 2017, each with 4-14 suppliers, 65 welfare agencies, nine vehicles and five-day decision horizons. Donations of different crops were aggregated in terms of their weight to account for one homogeneous product (i.e., food). Second, to allow for more diversity in the input characteristics, we created an additional dataset with randomly generated instances. This dataset is composed of five basic instances, with 20, 30, 40, 60 and 100 sites in total (suppliers and agencies combined). The number of vehicles available in each of these basic instances, as well as the number of days in the decision horizon, each increase with the number of sites. The numerical parameters of these instances, i.e., the distance matrix, the capacity of the vehicles, the supply sizes and agencies' served population sizes, were all drawn from uniform distributions, such that all supplies could be collected. Each of these five basic instances were used to generate five additional instances by setting a different value for the parameter L, which represents the daily allowed traveling time per vehicle. This process has led to the creation of 30 instances, amounting to 40 instances in total (along with the real life instances), which were used as part of the analysis. The data of all of these instances are available on the webpage of the first author. The experiments discussed herein were all performed on the same computer, which featured an Intel i7-4790 3.6 GHz processor and 32 GB of RAM and operated on Windows 7 (64-bit). Coding was performed using Python 2.7, and the commercial solver used was CPLEX 12.8.

We start by looking into the values of the upper bounds for each instance, presented in Table 1, which we also use in the subsequent analysis. In the first and second columns, we describe the instance number and the associated dataset, respectively. Consequently, in Columns 3-7 we state some input characteristics, namely, the number of suppliers of welfare agencies, of vehicles, of days considered and the value of the parameter MV_D, which states the maximal number of agencies that can be respectively visited by any vehicle daily. Following this, in Columns 8-11 we present the values obtained for each type of bound presented in Section 5, defined and denoted there as UB_i for $i = 1, \dots, 4$. We note that in the majority of the small- to medium-sized instances (37 out of the 40 instances with up to 40 sites), the tightest bound, noted in boldface in the table for each instance, was obtained by one of the methods which is based on valid formulations, namely UB_1 and UB_2. Conversely, in all of the larger instances (with at least 60 sites), either UB_3 or UB_4 provided tighter bounds compared to UB_1 and UB_2. For the latter instances, UB_3 seemed to produce tighter bounds in the real life instances from Leket Israel (performed better than UB_4 in 7 out of 10 instances), whereas UB_4 was better in the randomly generated instances (performed better in 33 out of 40 instances). The instance size did not seem to have a significant effect on this trend, although the absolute improvement of UB_3 over UB_4 in the larger instances was more significant. We also recall that we use the tightest of these six four values as the effective bound for the instance, denoted by $UB \equiv \min_{i=1,\dots,4} UB_i$, for the purpose of computing optimality gaps as part of the analysis that follows.

Next, in Table 2 we focus on comparing the different solution methods for the H-GDSP. Similar to Table 1, in Columns 1-7 we specify the input parameters for each instance. Following this, in Columns 8-19 we report different measures with respect to the results of the three solution approaches described in this paper for the H-GDSP: (1) solving the full formulation with site-based routing variables; (2) using an iterative decomposition approach, in which the distribution aspect is formulated using site-based decision variables, and each distribution plan that is obtained is verified for feasibility with respect to the collection aspect; and (3) using a decomposition approach similar to (2), with the exception that the distribution aspect is formulated using route-based decision variables. Each solution method was allocated a 60-minute run for each instance. We next discuss the results from each of these approaches and analyze the differences between them (Column 20 is discussed in Section 7.2).

In Column 8, we present the actual running times (in seconds) for the first approach. It can be seen that the smaller instances were solved to optimality (therefore their run time did not reach the budget of 3,600 seconds), while mid-sized and larger instances, including the real-life instances from "Leket Israel", were not. For the second approach, we present in Columns 9, 10 and 11 the actual running times, the number of iterations that were completed and the relative improvement in the value of the feasible solution obtained by the second approach compared to the first approach, denoted by $RI_1^2 \equiv \frac{Z_2 - Z_1}{Z_1}$, where Z_i represents the value of the solution obtained by approach $i = 1,2$. It can be observed that the second approach managed to obtain optimal solutions in a smaller number of instances and required a larger amount of running time to obtain them. Furthermore, in three of the instances, no feasible solution was obtained before the allowed running time was exhausted (in such instances the relative improvement in Column 11 reads "N/A"). We note that the number of iterations tends to be larger in instances of medium size. This can be explained by the fact that smaller instances are easier to begin with and therefore require a small number of iterations, whereas larger instances require much effort for each iteration and therefore not many of them are completed by the end of the run time. As for the quality of the solutions obtained, in all of the smaller and mid-sized instances, with up to 40 sites, the first approach managed to come up with better feasible solutions. However, in the larger instances the opposite trend can be observed: in the 60-site instances, the average improvement is 5.55% (max: 22.10%); in the 100-site instances, it is 80.80% on average (max: 123.09%); and in the instances from "Leket Israel", it is 16.39% on average (max: 57.97%). These results suggest that the second approach tends to have an advantage over the first approach in larger instances, although it could also terminate its run with no feasible solution at hand. This justifies the use of our suggested third approach, as stated in Section 4.

Next, Columns 12-15 refer to the results of Approach 3 which is our suggested solution approach. In Column 12 we report its actual solution times and we note that as in the previous two methods, only in the smaller instances its run was completed before the allocated one hour was exhausted. In Column 13, we report the number of iterations that were completed during the run and note that as for the second approach, the number of completed iterations was lower in the smaller and in the larger instances than in the mid-sized ones. In Column 14, we indicate the optimality gaps reported by the commercial solver for the formulation that is solved as part of this approach, which, as explained, includes only a subset of all of the possible routes. Therefore, this gap does not represent the distance between the feasible solution at hand

and the optimal solution, but rather to the best solution that could theoretically be obtained by the actual formulation that was solved in practice. The average value of this gap was 7.85% across all of the instances that were tested, with a small number of outlying instances (4 out of 40) with gaps higher than 20%, and a majority of the instances (21 out of 40) with gaps lower than 10%. This is evidence of the fact that despite the relatively short time allocated to solving this formulation, it is close to realizing its full potential in terms of returning high quality feasible solutions. Next, in Column 15 we present the relative improvement in the value of the solution obtained by the third approach compared to the first approach (note that the improvement relative to the second approach can be derived from Columns 11 and 15). We observe that the average improvement of the third approach compared to the second approach is quite significant, standing at 47.77% in the instances from "Leket Israel". In all of the random instances with 20 sites, the optimal solutions were obtained by the third approach; and in the instances with 30, 40, 60 and 100 sites, the average improvement stands at 8.67%, 152.82%, 38.45% and 24.80%, respectively. This implies the improved ability of our suggested solution method to obtain feasible solutions which, except for two random instances, are of superior quality compared to the other two more basic approaches.

Finally, in Columns 16-18 we report optimality gaps with respect to the solutions obtained by the three solution approach, that is $OG_i = \frac{UB-Z_i}{Z_i}$, where Z_i is the value of the solution obtained by approach $i = 1,2,3$ and UB is the value of the upper bound on the optimal solution (discussed above). Since all gaps are computed with respect to the same upper bound, the optimality gaps based on the different feasible solutions can be compared directly. Therefore, we denote in boldface the lowest (tightest) of the optimality gaps for each instance, and observe that in all of the instances, except for two random ones, our suggested solution approach (i.e., the third approach) obtained the tightest gap. Furthermore, in the remaining two instances the third approach was only marginally outperformed. The average optimality gaps for the solutions from the suggested approach stand at 0.00%, 0.17%, 2.16%, 20.29% and 83.84% for the 20-, 30-, 40-, 60- and 100-site instances, respectively, and 57.19% for the instances from "Leket Israel", further demonstrating its ability to successfully solve small- and medium-sized instances. In the larger instances, the trend was inconclusive, as in some of the instances the optimality gaps were small, whereas in others they remained high. However, the analysis we have performed to assess the quality of the upper bounds suggests that in the larger instances they may not be sufficiently tight and that this may be the cause driving the optimality gaps high. Therefore, this should not be interpreted as evidence of having obtained poor quality solutions.

7.2 *Comparison to Current Practice*

In the previous section, we have shown that our suggested solution approach outperforms two alternative approaches, also based on mathematical programming. In this section we compare it to the simplistic rules of thumb currently used by the Israeli food bank "Leket Israel" to solve the H-GDSP. To this end, we have established an understanding of the principles according to which the food bank currently performs its daily operational decision making, and we have organized them as an algorithm that mimics them. This algorithm, to which we refer as the *LI algorithm,* consists of two stages. In the first, the collection decisions are determined in a way that aims to collect the larger supplies as early as possible. Following this, in the second stage, the distribution aspect is tackled, given the collection decisions determined in the first stage. The second stage was designed as a simple and quick heuristic method in an effort to make the routing decisions easy to determine in practice. We next briefly describe the sketch of these stages and note that a full pseudocode is provided in Online Appendix E.

In the first stage of the LI algorithm, visits to suppliers are scheduled, under the assumption that at most half of the vehicles are used every day for collection purposes (the remainder of the vehicles can be used for distribution). To determine which suppliers are visited every day, we go over them in descending order of their supply sizes, and schedule a visit as early as possible within their time window for collection. That is, when considering supplier $i \in P$, we start with day a_i. If there are available vehicles on that day, we set a visit, otherwise we move on to day $a_i + 1$, and so on. If there are no available vehicles on any day $a_i, \dots, b_i$, we do not visit this supplier. Following this, in the second stage of the LI algorithm, distribution routes are determined for every available vehicle on every day (i.e., that was not assigned to visit a supplier on that day). For this purpose, a route-first-assign-later approach is used, by which a "giant tour" that includes all of the agencies is determined at a strategic off-line level. Then, the daily vehicle routes follow that sequence of sites, while respecting the restrictions imposed with respect to traveling time per vehicle, capacity per route and inventory availability. The routes are constructed such that the first agency visited by each new route is the successor in the giant tour of the last agency visited by the last route constructed. The giant tour can be pre-determined in any exact or heuristic approach that optimizes a relevant measure, e.g., the total distances traversed, in which case it is equivalent to solving a Traveling Salesman Problem (TSP). The advantage of the restriction that agencies are visited sequentially according to a fixed order, other than ease of implementation, is that this balances the number of visits that each agency receives over time, therefore potentially promoting a high level of equity in the long run. A high level of effectiveness is promoted through the early scheduling of visits to suppliers with a large donation.

As part of the numerical experiment we have performed, we have also applied the LI algorithm to all of the tested instances, detailed in the previous section. The giant tour provided as input was obtained by optimally solving a TSP instance consisting of all of the agencies, as an offline step. Following this, obtaining the solution by the LI algorithm required very short running time in all of the instances (< 1 second). However, in terms of the objective value obtained, the LI algorithm was outperformed in all of the instances by our suggested solution approach. To see this, we present in Column 20 of Table 2 the relative improvement in the value of the solution obtained by our suggested solution approach compared to the value of the solution obtained by the LI algorithm, denoted $RI_{LI}^{3} \equiv \frac{Z_3 - Z_{LI}}{Z_{LI}}$, where Z_3 follows the same definition from Section 7.1, and Z_{LI} represents the value of the solution obtained the LI algorithm. The average relative improvement across all of the instances was 17.98%. Moreover, we note that the improvement tends to be higher in larger instances, namely, it was 10.74%, 17.91%, 18.13%, 23.12%, 23.92% and 15.63% for the instances with 20, 30, 40, 60 and 100 sites, and for the instances from Leket Israel, respectively. The smallest improvement obtained in any instance was 5.40%, whereas the largest one was as high as 38.24%. To conclude this discussion, we also note that in the vast majority of the instances (34 out of 40), the same subset of suppliers was visited (though possibly on different days). Therefore, the improvement in the value of the solutions obtained by our suggested approach can be associated with two aspects: (1) an improved collection schedule, which allows the total amount distributed to be higher (due to a better consideration of the perishability aspect, for example); and/or (2) an improved distribution plan, which allows the same amount of vehicles to distribute the available supplies in a preferable manner.

7.3 *Rolling Horizon Framework*

As part of our analysis of the rolling horizon framework for the H-GDSP, we have also conducted an additional numerical experiment, aimed at testing the benefit that arises from using this approach, and specifically to assess the contribution of designating vehicles for future collection. This experiment was also performed based on instances from "Leket Israel" and from a randomly generated dataset, though using different instances, created in the following manner:

a. The real-life dataset: Based on information provided from "Leket Israel", the length of the planning horizon was set to a realistic value of 10 days. Following this, and based on data regarding one month of activity, three 10-day instances were generated with 11-32 suppliers, 65 agencies and 9 vehicles each.
b. The randomly generated dataset: first, six basic instances of different sizes (11, 20, 30, 40, 60 and 100 sites in total) were randomly generated. Then, each basic instance was duplicated with 3-4 distinct values for the parameter L. To increase the diversity of the dataset, instances with 60 and 100 sites were then further duplicated to accommodate two lengths of decision horizons. This lead to a total of 28 instances, each consisting of 7-12 days of activity.

We next further discuss the horizon for which each instance was solved. First, we recall that the planning horizon is denoted by $|T|$ and it is provided as part of the input of an instance. For this "full" horizon, each instance was solved in two methods: (1) with our suggested solution approach; and (2) with the LI algorithm. Moreover, the methods described in Section 5 were used to obtain a valid upper bound value. In addition, each instance was solved using a rolling horizon framework. In this context, we follow the notation from section 6 and denote by τ the length of the decision horizon. We note that the value of τ is not part of the input of the problem, rather it is a parameter of the solution framework, to be determined by the decision maker. Based on initial calibration experiments, aimed at finding the longest decision horizon for which reasonable optimality gaps could be obtained, we have determined the following policy for the value of τ: small and medium-sized instances (with up to 40 sites in our datasets) should be solved using 5-day rolling horizons, whereas larger instances (with 60 sites or more in our datasets) should be solved using 4-day rolling horizons. After setting the value of τ in this manner, each instance of the H-GDSP was separated into a series of successive $|T| - \tau + 1$ instances, each with τ days, and the decisions were rolled over as described in Section 6. Table 3 below summarizes the different types of objective values associated with each instance.

Table 3: Objective Values for the Analysis of the Rolling Horizon Framework

Notation	Days / Decision Horizon	# Decision Horizons / Inst.	Run Time / Horizon (hrs)	Solution Method	Additional Characteristics	Feasible Solution?
$\boldsymbol{Z_f}$	$\|T\|$	1	$\|T\| - \tau + 1$	Approach 3	-	Yes
$\boldsymbol{Z_{LI}}$	$\|T\|$	1	$\|T\| - \tau + 1$	LI algorithm	-	Yes
$\boldsymbol{Z_{d-}}$	τ	$\|T\| - \tau + 1$	1	Approach 3	No designation of vehicles for future collection	Yes
$\boldsymbol{Z_{d+}}$	τ	$\|T\| - \tau + 1$	1	Approach 3	Vehicles designated for future collection	Yes
$\boldsymbol{Z_b}$	$\|T\|$	1	24	As described in Sect. 5	Upper bound	Not necessarily

At this point we turn to discuss the results of the numerical experiment, which are presented in Table 4. Columns 1 and 2 indicate the instance number and the associated dataset, respectively. Then, in Columns 3-7 we state some input characteristics, namely, the number of suppliers, welfare agencies, vehicles, days in the planning horizon, and the value of the parameter L. In Columns 8-10, we indicate the improvement in the value Z_{d-} relative to three objective values: Z_b, Z_f and Z_{LI} (as defined in Table 3). The measures reported in these columns are $RI_X^{d-} \equiv \frac{Z_{d-} - Z_X}{Z_X}$ such that $X \in \{b, f, LI\}$. We start by noting that the average optimality gap, according to Column 8, is 29.83% across all instances. Specifically, it is quite low in the 11- and 20-site instances, standing at 2.50% and 15.20%, on average, respectively; and in the random instances with 30 sites or higher, as well as in the instances from "Leket Israel", it is 34.86% on average (min: 8.49%, max: 46.94%). In Column 9, it can be observed that when smaller instances are involved, it might be advisable in certain cases to consider the full planning horizon with our suggested solution approach rather than using the rolling horizon framework. However, in all of medium and large-sized instances, the rolling horizon framework outperformed the planning horizon approach, and the relative improvements are especially significant in the large instances: 37.10%, 42.13% and 93.95%, on average, for 60- and 100-site random instances and for the instances from "Leket Israel", respectively. This trend did not seem to be correlated with the length of the planning horizon. In Column 10 we performed a similar analysis with respect to solving the single instance with the planning horizon using the LI algorithm. Since this algorithm does not suffer from the same solvability issues when longer horizons are considered, the relative improvement that the rolling horizon framework has over it are lower, standing at 1.14% on average across all instances. Specifically, the rolling horizon framework was consistently outperformed in all of the smaller instances (-4.52% on average for instances with up to 30 sites) and in the larger instances, it consistently performed worse than the LI algorithm. The conclusion that arises from this analysis is that simply using a rolling horizon framework with our suggested solution approach to obtain feasible solutions for the H-GDSP is not an advisable course of action. This can be associated with the tractability issues, as well as with the end-of-horizon effect, discussed in Section 6. Therefore, we next turn to discuss the mechanism which is designed to mitigate this effect by designating vehicles for future collection.

Thus, we next look at Column 11, which indicates the relative improvement in the value of the feasible solution obtained by using the rolling horizon framework when vehicles are designated for future collection, compared to when they are not, i.e., $RI_{d-}^{d+} \equiv \frac{Z_{d+} - Z_{d-}}{Z_{d-}}$. We start by noting that the average relative improvement across all instances was 21.86%, and that improvements were observed in all of the instances, except for one of them. The improvement tends to be more significant when larger instances are involved, standing at 62.64%, 103.03% and 133.90% for the 60- and 100-site random instances and for the instances from "Leket Israel", respectively. In the larger random instances, it can also be observed that the improvement also seems to be higher when the planning horizon is longer. Therefore, there is value in designating vehicles for future collection when the rolling horizon framework is employed.

Finally, Columns 12-14 are analogous to Columns 8-10 with respect to Z_{d+}. We do not go over the entirety of the values obtained there for brevity, since $1 + RI_X^{d+} = (1 + RI_{d-}^{d+}) \cdot (1 + RI_X^{d-})$ for $X \in \{b, f, LI\}$ and these factors were previously discussed. We shall note the following general observations: based on Column 12, the average optimality gap is lower than in Column 8, standing at 16.03%; based on Columns 13-14, the average improvement over the methods for solving the instance with the full planning horizon stands at 63.23% and 22.71% (max: 161.47% and 92.57%), respectively. In only 3 (1) out of 31 instances, the rolling horizon framework was outperformed by solving for the planning horizon with our suggested solution approach (with the LI algorithm), and in these cases the value of the solution obtained was only marginally worse ($< 1.5\%$). The conclusions that can be drawn from the combination of all of these findings is that H-GDSP instances with long planning horizons should be solved with the rolling horizon framework, and this approach should be accompanied by the suggested feature that reduces the severity of the end-of-horizon effects. When it is adequately used, the advantage over solving an instance for the long planning horizon can be significant.

8. Conclusion

This paper has focused on a logistic problem arising in the activity of food banks that manage gleaning operations. As part of this operation, a fleet of vehicles is used to collect donations of fresh produce from farmers, which are then processed in a dedicated facility so that they can be delivered to food aid agencies by the same fleet of vehicles. This setting, in which vehicle routing and scheduling as well as inventory management decisions are to be made simultaneously, under a humanitarian objective function, has not been previously studied in the literature.

The solution approach we have proposed for the problem is based on the observation that it can be decomposed into its collection and distribution aspects. Solving each of these sub-problems on its own, as part of an iterative procedure, can reduce the overall computational effort to obtain high-quality feasible solutions within a reasonable running time. The numerical analysis has confirmed that this approach has significant advantages over two different naïve approaches and over the simplistic methods currently in use. Moreover, we have shown how the use of a rolling horizon approach can help tackle tractability issues in instances with long planning horizons, while mitigating potential end-of-horizon effects.

The analysis presented in this paper opens an avenue for further research directions. First, the problem can be extended in a way that considers multiple types of food products. The significance of this variant is related to the fact that in real life, food aid agencies may have different preferences over the set of food products (e.g., tomatoes and onions are more likely to be favored compared to turnips and spinach). It is therefore often the case that the fixed quantity guaranteed to each agency whenever it is visited, included as part of the input to the problem, is generalized to a *basket* guaranteed to each agency, consisting of a fixed quantity from each of a set of product *categories* (e.g., basic vegetables, special vegetables, fruit, etc.). The problem would then require each agency to be supplied with fixed quantities of each food category every time it is visited. We note in this regard that incorporating considerations of equity over product type is a subject that also merits further investigation, though it is not relevant in the context of our problem since the basket composition is determined offline.

Second, with respect to the inventory aspect of the problem, the expiration date of products was assumed to be fixed and pre-determined. However, other perishability assumptions can be modeled, e.g., the produce becomes inedible a certain amount of days following the day it was harvested. While such an assumption does change the formulation of the collection sub-problem, it does not have severe implications on the overall approach we use for the problem. This issue becomes especially important in the multi-product setting, in which different types of produce may spoil at different rates.

Finally, another interesting direction for further investigation may be to consider settings in which complete information regarding the arrival of suppliers is not available at the time of the initial planning, contrary to the assumption we have made in this study. Instead, suppliers may arrive at random points in time throughout the decision horizon, leading to possibly dynamic or stochastic counterparts of the H-GDSP. The solution approach we have presented in this paper can well be part of the methodology used to address these variants as well.

Acknowledgments

The research was partially supported by the Israel Science Foundation (ISF), grant no. 463/15. The first author was partially supported by the Manna Center Program for Food Safety and Security, and by a scholarship from the Shlomo Shmeltzer Institute for Smart Transportation, both at Tel Aviv University.

References

1. Anand, S. (1983). *Inequality and poverty in Malaysia: measurement and decomposition.* Oxford University Press: New York.
2. Ata, B., Lee, D., and Sönmez, E. (2019). Dynamic volunteer staffing in multicrop gleaning operations. *Operations Research*, *67*(2), 295-314.
3. Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, *15*(1), 1-31.
4. Bräysy, O., and Gendreau, M. (2005). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation science*, *39*(1), 104-118.
5. Bunte, S., and Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, *1*(4), 299-317.
6. Chand, S., Hsu, V. N., and Sethi, S. (2002). Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, *4*(1), 25-43.
7. Chuzhoy, J., and Naor, J. S. (2006). New hardness results for congestion minimization and machine scheduling. *Journal of the ACM*, *53*(5), 707-721.
8. Coelho, L. C., Cordeau, J. F., and Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, *48*(1), 1-19.

9. Davis, L. B., Jiang, S. X., Morgan, S. D., Nuamah, I. A., and Terry, J. R. (2016). Analysis and prediction of food donation behavior for a domestic hunger relief organization. *International Journal of Production Economics*, *182*, 26-37.
10. Eisenhandler O. and Tzur M. (2019a). The Humanitarian Pickup and Distribution Problem. *Operations Research*, 67(1), 10-32.
11. Eisenhandler, O., and Tzur, M. (2019b). A Segment-Based Formulation and a Matheuristic for the Humanitarian Pickup and Distribution Problem. *Transportation Science, 53(5), 1389-1408.*
12. Endeweld, M., Barkali, N., Abrahamov, V., Gealia, A., and Gottlieb, D. (2014). *2012 Food Security Survey: Main Socio-economic Findings.* Working paper by the Israeli Institute for Social Security. Available via the following link: https://www.btl.gov.il/Publications/research/documents/mechkar_115.pdf
13. Federgruen, A., and Tzur, M. (1999). Time-partitioning heuristics: Application to one warehouse, multi-item, multi-retailer lot-sizing problems. *Naval Research Logistics*, *46*(5), 463-486.
14. Goel, A. (2009). Vehicle scheduling and routing with drivers' working hours. *Transportation Science*, *43*(1), 17-26.
15. Karp R.M. (1972) Reducibility among Combinatorial Problems. In: Miller R.E., Thatcher J.W., Bohlinger J.D. (eds). *Complexity of Computer Computations.* The IBM Research Symposia Series. Springer: Boston, MA.
16. Kolen, A. W., Lenstra, J. K., Papadimitriou, C. H., and Spieksma, F. C. (2007). Interval scheduling: A survey. *Naval Research Logistics*, *54*(5), 530-543.
17. Lee, D., Sönmez, E., Gómez, M. I., and Fan, X. (2017). Combining two wrongs to make two rights: Mitigating food insecurity and food waste through gleaning operations. *Food Policy*, *68*, 40-52.
18. Moin, N. H., and Salhi, S. (2007). Inventory routing problems: a logistical overview. *Journal of the Operational Research Society, 58(9), 1185-1194.*
19. Nair, D. J., Grzybowska, H., Fu, Y., & Dixit, V. V. (2018). Scheduling and routing models for food rescue and delivery operations. *Socio-Economic Planning Sciences*, *63*, 18-32.
20. Nair, D. J., Rey, D., and Dixit, V. V. (2017). Fair allocation and cost-effective routing models for food rescue and redistribution. *IISE Transactions*, 49(12), 1172-1188.
21. Rey, D., Almi'ani, K., and Nair, D. J. (2018). Exact and heuristic algorithms for finding envy-free allocations in food rescue pickup and delivery logistics. *Transportation Research Part E: Logistics and Transportation Review*, *112*, 19-46.
22. Spieksma, F. C. (1999). On the approximability of an interval scheduling problem. *Journal of Scheduling*, *2*(5), 215-227.

Table 1: Analysis of the Upper Bounds for the H-GDSP

#	Dataset	$\lvert P \rvert$	$\lvert D \rvert$	$\lvert K \rvert$	$\lvert T \rvert$	MV_D	UB_1	UB_2	UB_3	UB_4
1	LEKET ISRAEL	4	65	9	5	27	4,180.56	4,045.06	**3,937.64**	4,426.85
2		4	65	9	5	27	4,474.76	4,327.28	**3,739.63**	4,314.93
3		5	65	9	5	27	4,223.98	4,162.96	3,804.01	**3,255.66**
4		5	65	9	5	27	4,124.04	4,083.24	**3,708.56**	3,731.40
5		5	65	9	5	27	4,529.78	4,444.06	**3,906.19**	4,812.75
6		7	65	9	5	27	4,506.20	4,494.97	**3,530.40**	4,170.08
7		8	65	9	5	27	4,481.49	4,345.62	**4,121.42**	4,514.43
8		11	65	9	5	27	4,113.93	3,978.06	4,001.64	**3,063.65**
9		12	65	9	5	27	4,492.72	**4,147.62**	4,467.64	4,232.21
10		14	65	9	5	27	4,493.10	4,338.51	4,438.45	**3,241.81**
11	RANDOMLY GENERATED	4	16	2	5	3	**8.67**	**8.67**	35.52	58.61
12		4	16	2	5	5	**13.28**	**13.28**	61.79	49.25
13		4	16	2	5	6	**28.39**	**28.39**	112.23	94.76
14		4	16	2	5	8	**81.61**	**81.61**	256.31	218.66
15		4	16	2	5	10	**262.91**	**262.91**	379.66	354.8
16		4	16	2	5	12	**436.26**	**436.26**	488.46	519.65
17		5	25	3	5	3	**28.59**	**28.59**	85.63	75.61
18		5	25	3	5	7	**94.25**	**94.25**	204.33	184.43
19		5	25	3	5	11	**250.8**	530.49	294.75	288.49
20		5	25	3	5	16	**632.21**	656.49	662.38	657.51
21		5	25	3	5	21	**684.34**	802.35	734.29	731.67
22		5	25	3	5	24	**811.57**	813.32	816.23	814.26
23		8	32	3	7	4	**40.21**	**40.21**	92.61	52.48
24		8	32	3	7	8	**110.61**	**110.61**	154.26	135.26
25		8	32	3	7	13	402.18	**372.18**	384.28	375.68
25		8	32	3	7	13	402.18	**372.18**	384.28	375.68
26		8	32	3	7	18	472.30	468.55	461.35	**451.92**
27		8	32	3	7	22	528.65	523.94	**521.62**	522.46
28		8	32	3	7	28	576.43	561.36	**558.35**	584.15
29		10	50	4	7	5	1,485.23	856.35	**541.65**	606.57
30		10	50	4	7	12	1,705.86	1,448.68	1,302.91	**1,204.56**
31		10	50	4	7	20	3,368.55	2,976.36	2,994.66	**2,842.46**
32		10	50	4	7	28	5,779.25	5,146.34	**4,258.16**	4,649.31
33		10	50	4	7	36	6,048.75	5,213.68	4,993.25	**4,915.22**
34		10	50	4	7	45	6,049.22	5,446.25	5,336.64	**5,223.45**
35		20	80	5	10	6	2,408.19	5,228.36	1,664.52	**1,025.61**
36		20	80	5	10	15	6,405.94	4,779.32	4,064.12	**3,118.64**
37		20	80	5	10	20	8,586.36	6,513.28	5,028.61	**4,025.61**
38		20	80	5	10	28	10,682.25	8,433.25	**7,246.58**	7,370.20
39		20	80	5	10	42	10,686.71	9,327.84	9,243.38	**9,154.23**
40		20	80	5	10	60	10,689.06	10,256.36	10,432.25	**9,756.25**

Table 2: ***Results of the Solution Approaches for the H-GDSP***

		Input Characteristics					Approach #1	Approach #2			Approach #3				Optimality Gaps			Current Practice
#	Data Set	$\lvert P\rvert$	$\lvert D\rvert$	$\lvert K\rvert$	$\lvert T\rvert$	MV_D	T_1	T_2	$Iter_2$	RI_1^2	T_3	$Iter_3$	$\widehat{OG}_3$	RI_1^3	OG_1	OG_2	OG_3	RI_{LI}^3
1	LEKET ISRAEL	4	65	9	5	27	3,600.0	3,600.0	7	5.57%	3,600.0	5	7.52%	86.14%	191.57%	176.19%	**56.64%**	10.15%
2		4	65	9	5	27	3,600.0	3,600.0	12	57.97%	3,600.0	8	24.57%	204.50%	445.16%	245.11%	**79.04%**	10.52%
3		5	65	9	5	27	3,600.0	3,600.0	10	14.96%	3,600.0	10	9.12%	66.19%	128.37%	98.65%	**37.42%**	30.62%
4		5	65	9	5	27	3,600.0	3,600.0	4	18.01%	3,600.0	4	16.78%	54.87%	72.32%	46.02%	**11.27%**	12.04%
5		5	65	9	5	27	3,600.0	3,600.0	9	9.34%	3,600.0	9	25.35%	14.09%	107.44%	89.72%	**81.82%**	9.41%
6		7	65	9	5	27	3,600.0	3,600.0	15	4.45%	3,600.0	6	1.84%	65.11%	235.09%	220.81%	**102.95%**	27.81%
7		8	65	9	5	27	3,600.0	3,600.0	17	4.41%	3,600.0	8	31.36%	23.00%	107.04%	98.30%	**68.33%**	11.64%
8		11	65	9	5	27	3,600.0	3,600.0	11	N/A	3,600.0	8	29.60%	N/A	N/A	92.37%	**22.46%**	8.37%
9		12	65	9	5	27	3,600.0	3,600.0	8	N/A	3,600.0	7	5.83%	N/A	N/A	N/A	**40.90%**	24.78%
10		14	65	9	5	27	3,600.0	3,600.0	6	N/A	3,600.0	10	7.07%	N/A	N/A	N/A	**71.11%**	10.99%
11	RANDOMLY GENERATED	4	16	2	5	3	0.20	12.20	2	0.00%	26.21	4	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	8.78%
12		4	16	2	5	5	0.84	18.61	5	0.00%	45.21	7	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	12.60%
13		4	16	2	5	6	7.04	61.23	3	0.00%	138.34	8	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	7.77%
14		4	16	2	5	8	164.91	100.61	8	0.00%	600.24	10	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	5.40%
15		4	16	2	5	10	1,219.85	300.12	10	0.00%	215.31	12	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	19.29%
16		4	16	2	5	12	3,412.12	75.49	4	0.00%	71.28	6	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	10.58%
17		5	25	3	5	3	5.21	1,057.01	5	0.00%	123.20	4	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	8.12%
18		5	25	3	5	7	7.35	865.29	3	0.00%	432.25	6	0.00%	-1.02%	**0.00%**	**0.00%**	1.03%	35.41%
19		5	25	3	5	11	12.23	3,600.0	11	0.00%	358.41	8	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	28.40%
20		5	25	3	5	16	743.52	3,600.0	8	-13.10%	3,600.0	10	5.70%	0.00%	**0.00%**	15.08%	**0.00%**	17.97%
21		5	25	3	5	21	946.31	3,600.0	6	-8.15%	3,600.0	6	14.65%	0.00%	**0.00%**	8.87%	**0.00%**	11.16%
22		5	25	3	5	24	1,132.21	3,600.0	4	-22.55%	3,600.0	8	4.76%	0.00%	**0.00%**	29.11%	**0.00%**	6.41%
23		8	32	3	7	4	0.17	5.28	5	0.00%	3,600.0	10	0.00%	0.00%	**0.00%**	**0.00%**	**0.00%**	10.82%
24		8	32	3	7	8	2.18	21.23	8	-63.65%	3,600.0	6	3.35%	0.00%	**0.00%**	175.08%	**0.00%**	11.92%

		Input Characteristics					***Approach #1***	***Approach #2***			***Approach #3***				***Optimality Gaps***			***Current Practice***
#	**Data Set**	$\lvert P \rvert$	$\lvert D \rvert$	$\lvert K \rvert$	$\lvert T \rvert$	MV_D	T_1	T_2	$Iter_2$	RI_1^2	T_3	$Iter_3$	$\widehat{OG}_3$	RI_1^3	OG_1	OG_2	OG_3	RI_{LI}^3
25	RANDOMLY GENERATED (cont.)	8	32	3	7	13	3,600.0	3,600.0	6	-33.98%	3,600.0	13	12.55%	190.50%	208.66%	367.50%	**6.25%**	36.86%
26		8	32	3	7	18	3,600.0	3,600.0	4	-52.86%	3,600.0	20	35.63%	86.85%	92.61%	308.57%	**3.08%**	9.47%
27		8	32	3	7	22	3,600.0	3,600.0	11	-50.88%	3,600.0	8	19.90%	0.92%	2.10%	107.86%	**1.17%**	27.91%
28		8	32	3	7	28	3,600.0	3,600.0	6	0.00%	3,600.0	4	0.00%	0.00%	**2.43%**	**2.43%**	**2.43%**	11.78%
29		10	50	4	7	5	3,600.0	3,600.0	3	3.01%	3,600.0	7	0.00%	27.60%	125.61%	119.03%	**76.81%**	9.95%
30		10	50	4	7	12	3,600.0	3,600.0	4	22.10%	3,600.0	15	9.02%	99.20%	110.40%	72.32%	**5.62%**	11.37%
31		10	50	4	7	20	3,600.0	3,600.0	9	N/A	3,600.0	10	10.12%	N/A	N/A	93.45%	**7.11%**	12.28%
32		10	50	4	7	28	3,600.0	3,600.0	6	3.84%	3,600.0	12	1.38%	0.06%	21.29%	**16.80%**	21.22%	37.25%
33		10	50	4	7	36	3,600.0	3,600.0	5	5.43%	3,600.0	19	3.68%	36.17%	44.56%	37.12%	**6.16%**	29.89%
34		10	50	4	7	45	3,600.0	3,600.0	8	-6.64%	3,600.0	14	0.00%	28.42%	34.61%	44.17%	**4.82%**	37.95%
35		20	80	5	10	6	3,600.0	3,600.0	4	123.09%	3,600.0	5	8.79%	141.83%	1066.79%	423.00%	**382.48%**	12.04%
36		20	80	5	10	15	3,600.0	3,600.0	3	107.69%	3,600.0	9	10.49%	148.00%	319.60%	102.03%	**69.19%**	38.24%
37		20	80	5	10	20	3,600.0	3,600.0	1	5.69%	3,600.0	17	13.66%	51.47%	99.99%	89.23%	**32.03%**	31.86%
38		20	80	5	10	28	3,600.0	3,600.0	2	86.75%	3,600.0	8	6.11%	120.27%	144.91%	31.15%	**11.19%**	14.82%
39		20	80	5	10	42	3,600.0	3,600.0	0	N/A	3,600.0	20	4.17%	N/A	N/A	N/A	**5.74%**	9.51%
40		20	80	5	10	60	3,600.0	3,600.0	2	N/A	3,600.0	12	1.18%	N/A	N/A	38.14%	**2.38%**	37.06%

Table 4: Results of the Rolling Horizon Framework for the H-GDSP

Inst	**Dataset**	$\lvert P \rvert$	$\lvert D \rvert$	$\lvert K \rvert$	L	$\lvert T \rvert$	RI_b^{d-}	RI_f^{d-}	RI_{LI}^{d-}	RI_{d-}^{d+}	RI_b^{d+}	RI_f^{d+}	RI_{LI}^{d+}
1	LEKET ISRAEL	11	65	9	600	10	-44.54%	61.76%	9.61%	17.58%	-34.79%	90.20%	28.88%
2		24	65	9	600	10	-35.69%	107.78%	-13.61%	25.84%	-19.08%	161.47%	8.71%
3		32	65	9	600	10	-38.68%	112.29%	0.35%	17.78%	-27.78%	150.04%	18.20%
4	RANDOMLY GENERATED	3	8	2	20	7	-2.54%	-2.54%	-0.75%	2.17%	-0.43%	-0.43%	1.40%
5		3	8	2	45	7	-2.66%	-2.66%	-2.05%	1.27%	-1.43%	-1.43%	-0.80%
6		3	8	2	70	7	-2.31%	-2.31%	-0.28%	1.64%	-0.70%	-0.70%	1.36%
7		4	16	2	15	10	-11.08%	10.27%	-5.07%	7.85%	-4.10%	18.93%	2.38%
8		4	16	2	50	10	-14.62%	-2.64%	-11.26%	13.14%	-3.39%	10.16%	0.40%
9		4	16	2	80	10	-19.93%	-6.37%	-7.31%	15.26%	-7.71%	7.92%	6.84%
10		5	25	3	28	10	-46.94%	8.15%	-4.74%	10.36%	-41.44%	19.36%	5.14%
11		5	25	3	75	10	-38.38%	19.10%	-2.75%	9.75%	-32.37%	30.71%	6.73%
12		5	25	3	120	10	-32.88%	17.81%	-6.45%	8.16%	-27.40%	27.42%	1.19%
13		8	32	3	40	12	-40.04%	30.17%	12.34%	12.78%	-32.38%	46.80%	26.70%
14		8	32	3	70	12	-28.74%	18.66%	10.05%	15.85%	-17.44%	37.47%	27.49%
15		8	32	3	100	12	-28.61%	15.08%	14.24%	18.89%	-15.12%	36.82%	35.82%
16		10	50	4	40	10	-11.15%	55.56%	28.64%	1.30%	-9.99%	57.58%	30.31%
17		10	50	4	80	10	-8.49%	44.48%	19.48%	-6.77%	-14.68%	34.71%	11.39%
18		10	50	4	120	10	-21.77%	24.73%	-10.49%	24.00%	-3.00%	54.67%	10.99%
19		10	50	4	150	10	-30.62%	27.10%	5.80%	34.94%	-6.38%	71.51%	42.77%
20		10	50	4	40	12	-28.22%	42.99%	-5.51%	12.94%	-18.94%	61.49%	6.71%
21		10	50	4	80	12	-30.97%	38.81%	2.72%	26.70%	-12.53%	75.88%	30.14%
22		10	50	4	120	12	-35.04%	31.75%	10.44%	28.56%	-16.49%	69.39%	41.98%
23		10	50	4	150	12	-38.45%	31.40%	20.21%	33.84%	-17.63%	75.86%	60.89%
24		20	80	5	30	10	-40.76%	37.26%	27.65%	32.54%	-21.48%	81.92%	69.18%
25		20	80	5	85	10	-38.85%	56.33%	-24.59%	43.26%	123.96%	8.04%	43.26%
26		20	80	5	150	10	-39.59%	50.47%	-13.36%	37.37%	106.71%	19.02%	37.37%
27		20	80	5	200	10	-42.16%	27.10%	-26.10%	50.30%	91.03%	11.07%	50.30%
28		20	80	5	30	12	-44.55%	36.65%	-0.53%	45.83%	99.28%	45.05%	45.83%
29		20	80	5	85	12	-42.09%	36.42%	33.55%	44.19%	96.70%	92.57%	44.19%
30		20	80	5	150	12	-43.39%	44.12%	-26.62%	48.22%	113.62%	8.76%	48.22%
31		20	80	5	200	12	-40.91%	48.66%	1.88%	41.96%	111.04%	44.63%	41.96%

Online Appendix A: A Toy Instance of the H-GDSP

This online appendix provides a detailed toy instance of the H-GDSP, accompanied by a complete feasible solution to the problem. The network in the instance is described in Figure A.1 below, where all sites are located on a grid. It consists of a depot, represented by a rectangle numbered 0; three suppliers, each represented by a circle with a letter; and five food aid agencies, each represented by a triangle with a digit. Distances are computed based on the L_1 norm (Manhattan distances). The decision horizon consists of four days, denoted 0 to 3.

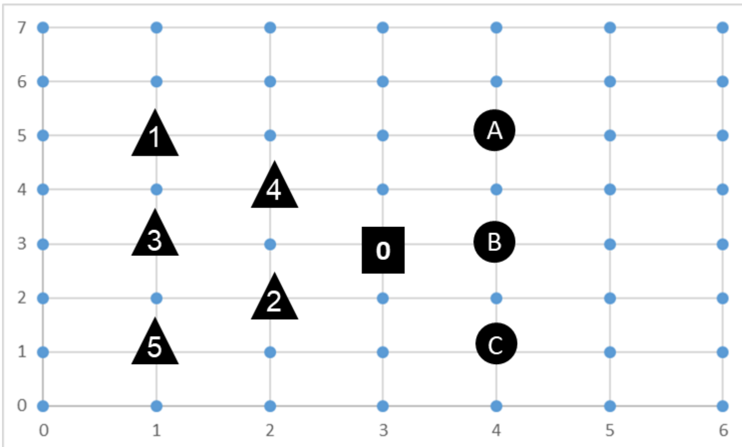


Figure A.1: The gleaning network used in the toy instance

The number of individuals served by each agency $i = 1, \ldots, 5$ is i. In addition, $B^* = 1$, therefore the fixed delivery amount for each food aid agency i is also equal to i units.

The suppliers are characterized by the following parameters:

Supplier, i	**A**	**B**	**C**
Amount Available, S_i	3	4	2
Expiration, d_i	2	3	3
Time Window for Collection (a_i, b_i)	(0,2)	(0,0)	(0,2)

The food bank has two trucks, each with a capacity of 5 units, and the maximal driving time of each truck is 8 time units. The opening inventory at the depot at the beginning of the decision horizon includes the following amounts:

Expiration Date t	**0**	**1**	**2**	**3**
Amount Available for Distribution α_t	1	2	-	-
Amount Undergoing Processing β_t	-	1	2	1

By enumerating all possible routes, we observed that while 15 distribution routes comply with the time restriction, only 8 of them also satisfy the capacity constraint (for instance, it is possible to leave the depot, visit agencies 3 and 5 and return to the depot within 8 time units, but this requires carrying 8 units, which is beyond the vehicle's capacity).

In Figure A.2, we illustrate a feasible solution for this instance. In this solution, suppliers A and B are visited on day 0; on day 1, supplier C is visited by one of the vehicles, and the other vehicle is used to visit agencies 2 and 4 (they can be visited either in one tour or in two separate tours without violating the time constraint); on day 2, one vehicle visits agencies 1 and 3, and the other visits agency 4; and on day 3, one of the vehicle visits agency 1, and the other one remains idle. The figure indicates in the first column the day considered; in the second column, we state the opening inventories, where different filling patterns represent different expiration dates (these are also differentiated by color in the online Online Appendix); in the third column, we describe the decisions to be implemented on that day by each of the vehicles: one of the vehicles is represented by full lines, and the other by dashed lines (the vehicles are also differentiated by color in the online Online Appendix).

The proposed solution is clearly feasible. Its objective value can be calculated as follows:

Agency i	**1**	**2**	**3**	**4**	**5**
Number of individuals served, n_i	1	2	3	4	5
Agency's population share, q_i	$1/15$	$2/15$	$1/5$	$4/15$	$1/3$
Fixed delivery amount per visit, y_i	1	2	3	4	5
Number of visits during decision horizon	2	2	1	1	0
Total amount delivered, Y_i	2	4	3	4	0

Hence, the effectiveness of the allocation is equal to $F = \sum_{i=1}^{5} Y_i = 13$.

The Gini index of the resulting allocation is:

$$G = \frac{\sum_{i=1}^{5} \sum_{j=i+1}^{5} |q_i Y_j - q_j Y_i|}{F} =$$

$$\frac{\left|\frac{1}{15}4-\frac{2}{15}2\right|+\left|\frac{1}{15}3-\frac{1}{5}2\right|+\left|\frac{1}{15}4-\frac{4}{15}2\right|+\left|\frac{1}{15}0-\frac{1}{3}2\right|+\left|\frac{2}{15}3-\frac{1}{5}4\right|+\left|\frac{2}{15}4-\frac{4}{15}4\right|+\left|\frac{2}{15}0-\frac{1}{3}4\right|+\left|\frac{1}{5}4-\frac{4}{15}3\right|+\left|\frac{1}{5}0-\frac{1}{3}3\right|+\left|\frac{4}{15}0-\frac{1}{3}4\right|}{13} =$$

$$\frac{0+\frac{1}{5}+\frac{4}{15}+\frac{2}{3}+\frac{2}{5}+\frac{8}{15}+\frac{4}{3}+0+1+\frac{4}{3}}{13} = \frac{5\frac{11}{15}}{13} = \frac{86}{195} \approx 0.441$$

Therefore, the objective value of this solution is $Z = 13 \cdot \left(1 - \frac{86}{195}\right) = 7\frac{4}{15} \approx 7.267$.

Figure A.2: A feasible solution for the toy instance

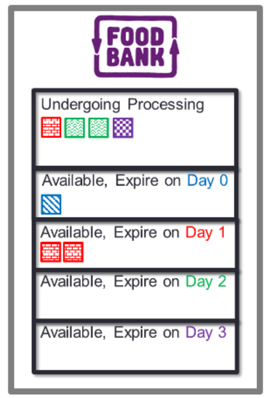
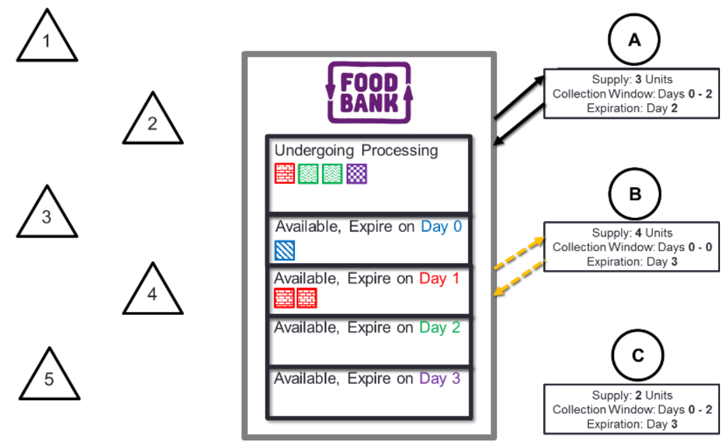
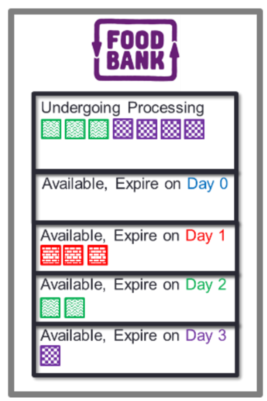
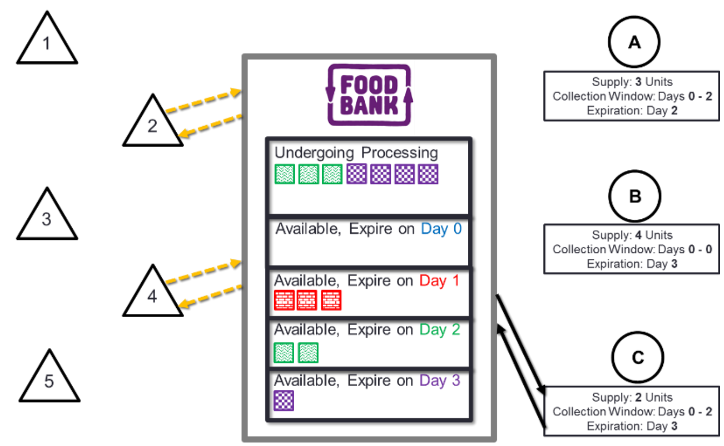
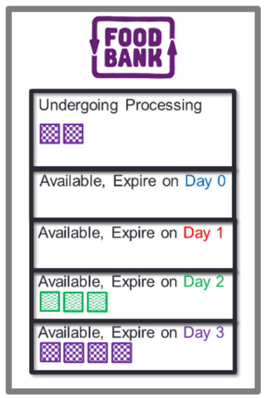
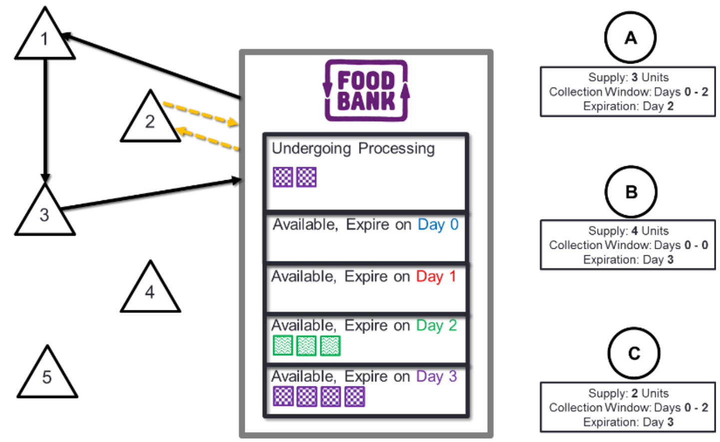
<table>
<tr><th>Day</th><th>Opening Inventory</th><th>Decisions</th></tr>
<tr><td rowspan="2">0</td><td>
</td><td>
</td></tr>
<tr><td colspan="2">Both vehicles are assigned to collect supplies</td></tr>
<tr><td rowspan="2">1</td><td>
</td><td>
</td></tr>
<tr><td colspan="2">One vehicle is assigned to collect supplies, and the other is assigned to distribute available imventory. Note that visiting agencies 2 and 4 in a single tour violates the capacity constraint, therefore two tours are required (this does not violate the time constraint).</td></tr>
</table>

Figure A.2: A feasible solution for the toy instance (cont.)

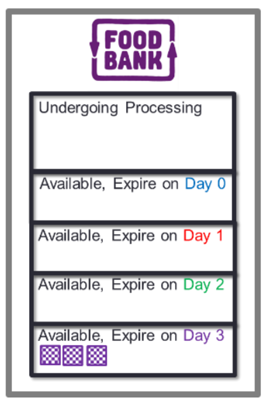
<table>
<tr><th>Day</th><th>Opening Inventory</th><th>Decisions</th></tr>
<tr><td rowspan="2">2</td><td>



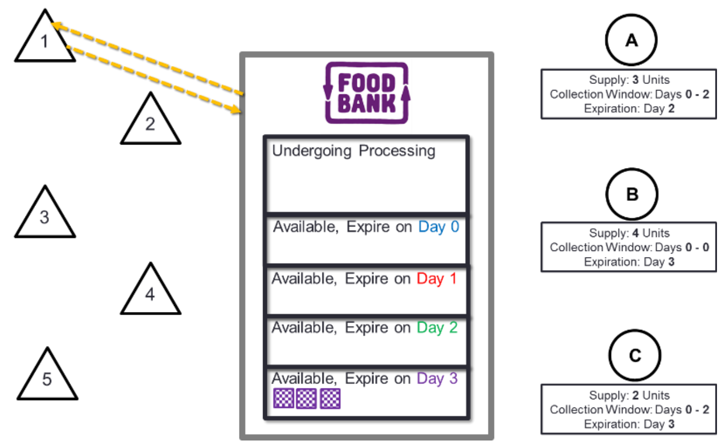
</td><td>



</td></tr>
<tr><td colspan="2">Both vehicles are assigned to distribute food inventory from the logistic center to food aid agencies.</td></tr>
<tr><td rowspan="2">3</td><td>



</td><td>



</td></tr>
<tr><td colspan="2">One vehicle is assigned to distribute food inventory from the logistic center to food aid agencies, and the other is idle</td></tr>
</table>

Online Appendix B: Hardness Proof for the H-GDSP

Claim 1.1: The decision version of the H-GDSP is strongly NP-Complete.

Proof: The decision version of the H-GDSP requires us to determine whether there exists a feasible solution whose objective value is at least as high as a given value Z^*. This problem is clearly in NP. To show that this problem is NP-Complete, we use a reduction from the Directed Hamiltonian Cycle Problem (DHCP), which is known to be strongly NP-Complete (Karp, 1972). This problem is defined as follows: "given a directed graph G with node set N, arc set A and weights c_{ij} on the arcs, does there exist a Hamiltonian cycle in G with total weight not exceeding a given value λ?". Based on the input of any DHCP instance, we construct the following instance of the H-GDSP:

- $|T| = 1$
- $P = \emptyset$
- $D = N$
- $\ell_{ij} = c_{ij}$ if $(i, j) \in A$ and $\ell_{ij} = M$ (a very large number) otherwise
- The depot is located arbitrarily located in the same location as the first food aid agency, such that $\ell_{01} = \ell_{10} = 0$ and $\ell_{0j} = \ell_{1j}, \ell_{j0} = \ell_{j1} \quad \forall j \neq 1$
- $y_i = 1 \quad \forall i \in D$
- $\alpha_1 = |D|$
- $L = C = \lambda$
- $Z^* = |D|$

This transformation is performed in polynomial time.

Then, for one side of the proof, A DHCP solution with a "YES" decision can be mapped to an H-GDSP solution in which all agencies are visited. Therefore $F = |D|$ and $1 - G = 1$ (i.e., complete equity), hence the objective value is $|D| \cdot 1 = |D|$, and the decision for H-GDSP is "YES".

For the other side, in an H-GDSP solution which represents a "YES" decision, the only way to obtain the objective value $|D|$ is by achieving simultaneously $F = |D|$ and $1 - G = 1$, since these are the highest possible levels of both the effectiveness and equity measures, respectively. Given the input of the H-GDSP instance, this can only be achieved if all agencies are visited. Hence, a directed Hamiltonian cycle is obtained. ■

Online Appendix C: Procedure for the Generation of Distribution Routes

The aim of this Online Appendix is to describe the procedure we have designed to generate the distribution routes for our suggested route-based solution method for the H-GDSP, described in Section 4. Given the desired number of pre-generated routes $|\mathcal{R}|$, This procedure consists of two stages, which are described next.

In the first stage, we construct a fixed number of routes (e.g., 30% of the specified value of $|\mathcal{R}|$) using a simple probabilistic construction heuristic, which works as follows: we start by inserting the depot to the route, then in each iteration, we add a delivery site, if possible. To do this, we create the subset of sites that can still be added to the route without violating the time (including the time to go back to the depot) and capacity restrictions, denoted D_{eff}. If no such site exists, i.e., $D_{eff} = \emptyset$, we terminate by inserting the depot again to the route. Otherwise, we assign to each site $j \in D_{eff}$ a weight $u_j \equiv \frac{y_j}{\ell_{ij}}$, where i is the current last site of the route, and then randomly select site j with probability $\frac{u_j}{\sum_{k \in D_{eff}} u_k}$. By doing so, we prioritize the insertion of sites that either do not require much traveling to or serve a relatively large number of individuals. Therefore, we refer to the routes created in this stage as being *saturated*, that is, no further sites can be sequentially added to them.

In the second stage, we generate additional routes, which we refer to as being *non-saturated*, in the sense that both the time and capacity constraints are not binding. As such, the total amount distributed by these routes may be relatively small, given the small number of agencies included in them, and therefore they may be beneficial in complementing the saturated routes that distribute larger amounts to agencies, given the total amount that is available at the depot, determined by the collection decisions. We create non-saturated routes by iteratively selecting one of the routes already included in $\mathcal{R}$ at random with equal probabilities, and then removing one of the sites included in them. Given route $r \in \mathcal{R}$, we set the probability of removing a specific site $i \in r$ as $\frac{\left(\frac{1}{u_i}\right)}{\sum_{k \in r}\left(\frac{1}{u_k}\right)}$, that is, the more "attractive" the site is (i.e., close to its predecessor or serves many individuals), it is less likely to be removed. Any new route that is generated in this manner is only added to $\mathcal{R}$ if it is not included in it already and if it is non-empty, that is, it includes any sites other than the depot. This stage terminates when $|\mathcal{R}|$ reaches the desired value, or when a certain number of consecutive iterations have concluded without a new route that could be added to $\mathcal{R}$.

Online Appendix D: Mathematical Formulations for the Upper Bounds

In this Online Appendix we present the procedures we use to compute the dedicated upper bounds for the H-GDSP, mentioned in Section 5. The definition of parameters and decision variables follows from Sections 3 and 5.

D1. *Upper Bound 3*

We recall that this bound is based on a relaxation of several aspects of the inventory management part of the problem, namely that a given amount of supply is available for distribution from the warehouse on the first day of the decision horizon, and that it all perishes only on the last day of the decision horizon. Under these assumptions, there is no need to schedule visits to suppliers or keep track of the amount of units that perish on each and every day. Moreover, the only aspect of the distribution plan that affects the feasibility of the solution, is the total traveling time constraint for each vehicle. Therefore, we assign agencies to vehicles, rather than days, as in the original problem. However, for each agency we only consider the minimal amount of time which is required to visit it, regardless of its successor in the specific route the vehicle performs in practice. This, as well as the fact the capacity of the vehicles are not taken into consideration, are relaxations with respect to the H-GDSP.

Next, we present the formulation to be solved. We recall the following notation, defined in Section 5:

- K_p - the number of vehicles used for collection
- K_d - the maximal number of vehicles used for distribution if K_p vehicles are used for collection, i.e., $K_d \equiv |K||T| - K_p$
- σ_{K_p} - the total supply of the K_p suppliers with highest supply sizes
- ℓ'_i - a lower bound on the time required to visit agency $i \in D$, i.e., $\ell'_i \equiv \min_{j \in (D \cup \{0\}) \setminus \{i\}} \{\ell_{ij}, \ell_{ji}\}$
- ℓ'_0 - a lower bound on the time required to leave the depot, i.e., $\ell'_0 \equiv \min_{j \in D} \{\ell_{0j}, \ell_{j0}\}$

Mathematical Formulation for UB3 given K_p

Decision Variables:

- h_{ik} - equals 1 if agency $i \in D$ is assigned to vehicle $k = 1, \dots, K_d$, otherwise 0

Mathematical Formulation:

$$\max \sum_{i \in D} Y_i - \sum_{i \in D} \sum_{j \in D: j > i} |q_i Y_j - q_j Y_i| \tag{60}$$

$s.t.$

$$\sum_{i \in D} \sum_{k=1}^{K_d} y_i h_{ik} \leq \sigma_{K_p} \tag{61}$$

$$\sum_{i \in D} \ell'_i h_{ik} \leq (L - \ell'_0) \qquad \forall k = 1, \dots, K_d \qquad (62)$$

$$Y_i = \sum_{k=1}^{K_d} y_i h_{ik} \qquad \forall i \in D \qquad (63)$$

$$\sum_{k=1}^{K_d} h_{ik} \leq |T| \qquad \forall i \in D \qquad (64)$$

$$Y_i \geq 0 \qquad \forall i \in D \qquad (65)$$

$$h_{ik} \in \{0,1\} \qquad \forall i \in D, k = 1, \dots, K_d \qquad (66)$$

The objective function (60) is the same one used for the H-GDSP. Constraint (61) limits the total amount distributed throughout the decision horizon based on the number of suppliers considered. Constraint (62) is the relaxed adaptation of the traveling time constraint per vehicle, namely Constraint (10). Constraint (63) sets the total amount allocated to each agency throughout the decision horizon. Constraint (64) enforces at most $|T|$ visits to every agency throughout the decision horizon. Finally, Constraints (64)-(65) are non-negativity and integrality constraints.

D2. *Upper Bound 4*

For this bound, we also relax the perishability assumption and therefore handle one type of inventory. However, as opposed to the former bound, here we schedule the collection from suppliers within their respective time windows, and as a consequence, we control the precise day each donation enters the available inventory. On the distribution side, we use any remaining vehicles every day (i.e., ones that are not used to visit suppliers on that day) to visit at most MV_D agencies.

$$\max \sum_{i \in D} Y_i - \sum_{i \in D} \sum_{j \in D: j > i} \left| q_i Y_j - q_j Y_i \right| \qquad (67)$$

$s.t.$

$$\sum_{a_i \leq t \leq b_i} p_{it} \leq 1 \qquad \forall i \in P \qquad (68)$$

$$\sum_{i \in D} d_{it} \leq MV_D \cdot (K - \sum_{i \in P} p_{it}) \qquad \forall t \in T \qquad (69)$$

$$Y_i = \sum_{t \in T} y_i d_{it} \qquad \forall i \in D \qquad (70)$$

$$\sum_{i \in D} y_i d_{i0} \leq \sum_{t \in T} \alpha_t \qquad (71)$$

$$\sum_{i \in D} y_i d_{i0} + \sum_{i \in D} y_i d_{i1} \leq \sum_{t \in T} \alpha_t + \sum_{t \in T} \beta_t \qquad (72)$$

$$\sum_{\tau=0}^{t}\sum_{i\in D} y_i d_{i\tau} \leq \sum_{t\in T}\alpha_t + \sum_{t\in T}\beta_t + \sum_{\tau=2}^{t}\sum_{i\in P} S_{i,\tau-2}\, p_{i,\tau-2} \qquad \forall t \in T: t \geq 2 \qquad (73)$$

$$p_{it} \in \{0,1\} \qquad \forall i \in P, t \in T: a_i \leq t \leq b_i \qquad (74)$$

$$d_{it} \in \{0,1\} \qquad \forall i \in D, t \in T \qquad (75)$$

$$Y_i \geq 0 \qquad \forall i \in D \qquad (76)$$

$$I_t \geq 0 \qquad \forall t \in T \qquad (77)$$

The objective function (67) is the same one used for the H-GDSP. Constraint (68) states that each supply can be collected at most once within its time window. Constraint (69) enforces the maximal number of visits to all of the agencies every day. Constraint (70) sets the total amount allocated to each agency. Constraints (71)-(73) state that the total amount that can be distributed up to every day cannot exceed the total available supplies, considering both the initial inventory and the total amount collected up to that day. Note that Constraint (71) refers to the first day ($t = 0$), Constraint (72) refers to the second day ($t = 1$), and Constraint (73) refers to the remainder of the days. Finally, Constraints (74)-(77) are integrality and non-negativity constraints.

Online Appendix E: Pseudocode of the "Leket Israel Algorithm"

In this Online Appendix, we present the full details of the "Leket Israel" (LI) Algorithm, mentioned in Section 6. The algorithm receives as input a "giant tour" that includes all of the agencies and that is used to determine the sequence according to which they are visited. The output of this algorithm includes three complementary variables:

(1) $Solution$, which defines the usage of the vehicles every day, that is $Solution[t][k]$ receives one of three possible values: 0, if the vehicle $k \in K$ is not used on day $t \in T$; "P" if the vehicle is used for collection purposes; "D" if the vehicle is used for distribution purposes.

(2) CP, which is used to define the collection plan, that is, $CP[t][k]$ indicates the supplier that vehicle k is assigned to visit on day t (only if $Solution[t][k] = "P"$)

(3) DP, which is used to define the distribution plan, that is, $DP[t][k]$ indicates the distribution routes that vehicle k is assigned to perform on day t (only if $Solution[t][k] = "D"$)

$\boldsymbol{ConstructLeketSolution}$(**Tour**)

$NumUsedVehicles[t] = 0 \;\; \forall t \in T$ *#Number of vehicles used on day t*
$Solution[t][k] = 0 \;\; \forall t \in T, k \in K$
 # Collection subproblem
For $i \in P$ in descending order of supply size, S_i:
    For $t = r_i, \ldots, b_i$:
        If $NumUsedVehicles[t] < \left\lfloor \frac{|K|}{2} \right\rfloor$:
            $Solution[t][NumUsedVehicles + 1] = "P"$
            $CP[t][NumUsedVehicles + 1] = i$
            $NumUsedVehicles[t] = NumUsedVehicles[t] + 1$
            Break *# if supplier i is scheduled, move to next supplier*
        End If
    End For
End For
 # Distribution subproblem
$Inventory[t_2] = \alpha_{t_2} \;\; \forall t_2 \in T$ *# t_2 represents the expiration*
$CurrIndex = 0$ *# the current index in the giant tour*
$curr = 0$ *# the current site in the giant tour*
$TotalInventory = \sum_{t_2=0}^{|T|-1} Inventory[t_2]$ *# the total amount available at the depot*
For $t = 0, \ldots, |T| - 1$:
    $TotalLoad = 0$ *# the total amount distributed on day t*
    For $k = NumUsedVehicles[t] + 1, \ldots, K$:
        $route = [0]$
        $TotalTravelingTime = 0$ *# the total time traveled by vehicle k*
        $CurrentLoad = 0$ *# the total load carried by vehicle k on day t*
        $next = Tour[(CurrIndex + 1) \bmod |D|]$ *# the next site considered*
        While $TotalTravelingTime + \ell_{curr,next} + \ell_{next,0} \le L$:
            While $CurrentLoad + y_{next} \le C$ **and** $TotalLoad + y_{next} \le TotalInventory$:

```
                    Append next to route
                    CurrentLoad = CurrentLoad + y_next
                    TotalLoad = TotalLoad + y_next
                    TotalTravelingTime = TotalTravelingTime + ℓ_curr,next
                    CurrIndex = (CurrIndex + 1) mod |D|
                    curr = Tour[CurrIndex]
                    next = Tour[(CurrIndex + 1) mod |D|]
                End While
                If route ≠ [0]:       # if route is non-empty, add it to the distribution plan
                    Append 0 to route
                    Solution[t][k] = "D"
                    Add route to DP[t][k]
                    route = [0]
                    CurrentLoad = 0
                    curr = 0
                End If
            End While
        End For
        # Update the opening inventories for the following day
        If t ≠ |T| − 1:
            For t_2 = t, …, |T| − 1:      # use inventory from oldest to freshest
                usage = min{TotalLoad, Inventory[t_2]}
                Inventory[t_2] = Inventory[t_2] – usage
                If t_2 = t:   # ignore inventory that expires today and is not distributed
                    Inventory[t_2] = 0
                End If
                If t = 0 :   # add β
                    Inventory[t_2] = Inventory[t_2] + β_t_2
                End If
                If t ≥ 2:   # add supplies two days after they are gleaned
                    For k in K:
                        If Solution[t − 2][k] = "P":
                            i = CP[t − 2][k]
                            Inventory[d_i] = Inventory[d_i] + S_i
                        End If
                    End For
                End If
            End For
        End If
End For
```