sensors

MDPI

Article

A Two-Stage Location Problem with Order Solved Using a Lagrangian Algorithm and Stochastic Programming for a Potential Use in COVID-19 Vaccination Based on Sensor-Related Data

Xavier Cabezas [1], Sergio García [2], Carlos Martin-Barreiro [1,3], Erwin Delgado [1] and Víctor Leiva [4,*]

[1] Faculty of Natural Sciences and Mathematics, Universidad Politécnica ESPOL, Guayaquil 090902, Ecuador; joxacabe@espol.edu.ec (X.C.); cmmartin@espol.edu.ec (C.M.-B.); edelgado@espol.edu.ec (E.D.)
[2] School of Mathematics, The University of Edinburgh, Scotland EH9 3FD, UK; sergio.garcia-quiles@ed.ac.uk
[3] Faculty of Engineering, Universidad Espíritu Santo, Samborondón 0901952, Ecuador
[4] School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362807, Chile
* Correspondence: victor.leiva@pucv.cl

Abstract: Healthcare service centers must be sited in strategic locations that meet the immediate needs of patients. The current situation due to the COVID-19 pandemic makes this problem particularly relevant. Assume that each center corresponds to an assigned place for vaccination and that each center uses one or more vaccine brands/laboratories. Then, each patient could choose a center instead of another, because she/he may prefer the vaccine from a more reliable laboratory. This defines an order of preference that might depend on each patient who may not want to be vaccinated in a center where there are only her/his non-preferred vaccine brands. In countries where the vaccination process is considered successful, the order assigned by each patient to the vaccination centers is defined by incentives that local governments give to their population. These same incentives for foreign citizens are seen as a strategic decision to generate income from tourism. The simple plant/center location problem (SPLP) is a combinatorial approach that has been extensively studied. However, a less-known natural extension of it with order (SPLPO) has not been explored in the same depth. In this case, the size of the instances that can be solved is limited. The SPLPO considers an order of preference that patients have over a set of facilities to meet their demands. This order adds a new set of constraints in its formulation that increases the complexity of the problem to obtain an optimal solution. In this paper, we propose a new two-stage stochastic formulation for the SPLPO (2S-SPLPO) that mimics the mentioned pandemic situation, where the order of preference is treated as a random vector. We carry out computational experiments on simulated 2S-SPLPO instances to evaluate the performance of the new proposal. We apply an algorithm based on Lagrangian relaxation that has been shown to be efficient for large instances of the SPLPO. A potential application of this new algorithm to COVID-19 vaccination is discussed and explored based on sensor-related data. Two further algorithms are proposed to store the patient's records in a data warehouse and generate 2S-SPLPO instances using sensors.

Keywords: heuristic algorithm; Lagrangian and semi-Lagrangian relaxations; mathematical programming; SARS-Cov2; sensing and data extraction; simple plant and uncapacitated facility location problems; XPRESS software

check for updates

Citation: Cabezas, X.; García, S. Martin-Barreiro, C.; Delgado, E.; Leiva, V. A Two-Stage Location Problem with Order Solved Using a Lagrangian Algorithm and Stochastic Programming for a Potential Use in COVID-19 Vaccination Based on Sensor-Related Data. *Sensors* **2021**, *21*, 5352. https://doi.org/10.3390/s21165352

Academic Editor: James Covington

Received: 6 July 2021
Accepted: 30 July 2021
Published: 9 August 2021

1. Introduction

The process of making decisions based on a solution to a problem, within the context of mathematical modeling, can take several stages. In each of these stages, events are defined in a such way that may be dependent on each other and on the variables that they define.

The use of random variables in the above-mentioned form of decision making may be described within a stochastic programming context. Then, the model can be represented as a sequential decision-making process, where the variables of one stage that are unknown at first are part of the input parameters of the subsequent stage.

A stochastic programming problem is a mathematical formulation with constraints where some of its components can be represented by random variables. If the problem has decisions that have to be made after the uncertainty is resolved (realization), it takes the name of the resource program [1]. The values of the random variables are known after carrying out an experiment, and decisions can be made at a first stage and at a second stage, before and after the experiment, respectively. In addition, each stage can contain a succession of decisions and they can be made over different periods of time. In this case, a multi-state stochastic programming problem should be formulated. Thus, we can say that a stochastic programming problem is multi-state if the decisions depend on realizations that occur over time. Since its introduction in the seminal papers by Dantzig [2] and Beale [3], the stochastic framework has been widely studied and applied in diverse areas, such as transportation [4], inventory management [5], drug supply [6], and industrial processes [7], among others.

Another example of the above framework is the so-called simple plant location problem, SPLP in short [8]. In the SPLP, given a set of possible locations, we must decide where to open facilities and then distribute resources demanded by certain customers/patients, minimizing the costs involved. Note that the SPLP is a combinatorial approach, where the mathematical model can be viewed as one of two stages that tries to answer two questions: (i) what facilities should be opened, and (ii) how should these facilities be located in such a way that the customer demand is met? The second question must be answered after the facilities (unknown at the beginning) are stated, that is, after the variables defined at this stage for such a decision are revealed by conducting an experiment that considers a possible future state of a second stage. Therefore, it is logical to think that it is necessary to define different forms of this state of nature that represent possible future scenarios such that the effect of uncertainty is minimized. Described in this way, the SPLP is known as a two-stage stochastic simple plan location problem, 2S-SPLP in short.

The 2S-SPLP has been widely studied, and its solution methods have been shown to be efficient. As mentioned in [1], it is important to consider that, taking advantage of the structure of the problem, the 2S-SPLP is especially beneficial in stochastic mathematical programming, and it is the main focus of the algorithmic work in this area. Here, the concept of structure refers to any characteristic in the mathematical formulation that can be used to provide a solution. One of the most-used procedures for stochastic mathematical programming that involves integer variables is the so-called integer L-shape method proposed in [9]. This method is a branch-cut algorithm that uses the L-shape that the coefficient matrix has in the stochastic formulation of the 2S-SPLP and solves subproblems obtained by the Benders decomposition. For details of this and other similar procedures, see [1,10]. Lagrangian relaxation (LR) is a less-used approach, but a full description of different applications can be found in the references previously cited. Some gradient (subgradient)-based methods, such as those used in LR, were applied over other algorithms to speed up the search for the optimum in discrete problems; see [11] as an example.

A less-known extension of the SPLP is when an order of preference, that customers have over a set of facilities to meet their demand, is considered. The SPLP with order is denoted by SPLPO in short [12,13], with the size of the instances in the SPLPO that can be solved being limited. Such an order incorporates a new set of constraints in its formulation that increases the difficulty to obtain an optimal solution. To the best of our knowledge, the use of order with two stages in the SPLP based on stochastic programming has not been considered until now. We use an algorithm that incorporates a heuristic component to handle the complexity that is added when orders of preference are included.

The motivation for our study comes from a problem related to healthcare services, where service centers must be sited in strategic locations to meet the immediate needs of patients. The current situation due to the COVID-19 pandemic makes this problem particularly relevant [14]. Let us assume that each center corresponds to an assigned place for vaccination and that each center uses one or more vaccine brands from different laboratories. Then, each patient could choose one center instead of another, because she/he may prefer the vaccine from a laboratory to be considered more reliable. This defines a ranking (order of preference) which will not necessarily be given for all of them, since a patient may not want to be vaccinated in a center where there are only non-preferred vaccine brands. In countries where the vaccination process is considered successful, the order of preference assigned by each patient to the vaccination centers is defined by incentives that local governments give to their population. These same incentives for foreign citizens are seen as a strategic decision to generate income from tourism.

The main objective of this work is to solve the problem of large instances based on a more general version of the 2S-SPLP, which we call the two-stage stochastic simple plant location problem with order (2S-SPLPO). Our secondary objective is to show a potential use of the 2S-SPLPO in COVID-19 vaccination based on sensor-related data. Indeed, we provide an algorithm that stores patient's service records in a central data warehouse using sensors. We conduct computational experiments on simulated instances to assess the performance of the 2S-SPLPO.

In the 2S-SPLPO, the customers have preferences over the set of facilities that can serve them. We consider the possibility that the order of preference given by patients is partial, that is, a patient may want to be served by only a subset of all possible facilities that may be open in some locations. In the stochastic formulation that we propose for the 2S-SPLPO, together with the variables defined to decide in which locations to open facilities, the preferences (partial or complete) are considered random variables. To solve the 2S-SPLPO, we use the accelerated dual ascent (ADA) algorithm proposed in [12,13], which solves large instances of the non-stochastic version of the SPLPO. Our algorithm is based on that given in [15] for the SPLP. The ADA algorithm uses LR and semi-Lagrangian relaxation (SLR) in an iterative procedure to approximate the solution. The SLR is a relatively new idea that exploits the structure of problems by relaxing a direction of a split equality constraint. The ADA algorithm tries to solve a Lagrangian dual (LD) problem with a subgradient method (SGM) as a first step. Then, the results of the first step are used in a second step to solve a semi-Lagrangian dual (SLD) problem with a dual ascent method (DAM). At last, a heuristic procedure is applied to accelerate the search of the optimum in a third step. For a complete description of the SGM, the reader is referred to [16–21], whereas for a description of the SLR, the reader is referred to [15,22] Applying heuristic procedures is a common practice in optimization, but without guarantee of reaching the optimum. Some creative algorithms help to finding sufficiently good solutions [23]. In the next sections, we shows how the ADA algorithm produces good solutions, comparing them to the exact solutions given by the XPRESS software.

Other two potential applications of the our approach can be seen in [24,25], where one of them is found in business-to-consumer (B2C) e-commerce. As pointed out in [25], e-commerce has logistical advantages over traditional B2C. However, although costs can be reduced considerably, due to the use of technology, the problem of distributing the products demanded to patients from centers exists still. Sometimes, patients prefer to pick up the products from these centers and, therefore, may have preferences about where to do it, as the distance to travel or travel times can influence the problem's variables. Thus, the aim of the B2C e-commerce organizations lies in finding the best way to locate the distribution centers, reducing the logistics costs related to this process and also considering the order of preference of the centers. These applications proposed in [24,25] are complemented with the potential application in COVID-19 vaccination based on the sensor-related data proposed in the present investigation.

The rest of the article is distributed as follows. Section 2 provides background on stochastic programming. Sections 3 and 4 define the SPLPO and 2S-SPLPO, respectively. In Section 5, a description of the three steps that form the ADA algorithm is provided.

In Section 6, we introduce two sensor-based algorithms that store patients' service records in a central data warehouse and that generate 2S-SPLPO instances, respectively. In Section 7, we report the results of computational experiments performed on large instances. Finally, in Section 8, some conclusions and recommendations for future studies on this topic are outlined.

2. Two-Stage Stochastic Linear Programming

In this section, we provide the background of stochastic programming, with MP standing for mathematical program. In MP 1, $\mathbb{E}_{\boldsymbol{\xi}}$ denotes the mathematical expectation on the random vector values $\boldsymbol{\xi}$, whereas $\boldsymbol{c} \in \mathbb{R}^{n_1}$, $\boldsymbol{b} \in \mathbb{R}^{m_1}$, $\boldsymbol{A} \in \mathbb{R}^{m_1 \times n_1}$, and $\boldsymbol{W} \in \mathbb{R}^{m_2 \times n_2}$ are known vectors and matrices.

Let $\Omega = \{\omega_1, \ldots, \omega_{|\Omega|}\}$ be a finite sample space of some experiment. Let $\boldsymbol{\xi}$ be a random variable over the elements of Ω with values $\boldsymbol{\xi}(\omega)$. For simplicity, we call any $\boldsymbol{\xi}(\omega)$ the scenario ω. In a stochastic program, the scenarios can be seen as states of the nature since all the unknown information could depend on them. In addition, we assume a discrete probability function expressed as

$$p(\boldsymbol{\xi}) = \mathbb{P}(\boldsymbol{\xi} = \boldsymbol{\xi}(\omega)) = \alpha^{\omega}. \tag{1}$$

The vectors $\boldsymbol{y} \in \mathbb{R}^{n_1}$ and $\boldsymbol{x} \in \mathbb{R}^{n_2}$ are the decision variables of the first and second stages, respectively. Therefore, $\boldsymbol{y}$ can be considered the decisions to be made under uncertainty, and $\boldsymbol{x}$ the corrective decisions made when uncertainty is revealed. Furthermore, $\boldsymbol{q}(\omega) \in \mathbb{R}^{n_2}$, $\boldsymbol{h}(\omega) \in \mathbb{R}^{m_2}$ and $\boldsymbol{T}(\omega) \in \mathbb{R}^{m_2 \times n_1}$ are vectors with known components after the realization of ω, with $\boldsymbol{T}(\omega)$ being usually called the technological matrix and $\boldsymbol{W}$ the fixed resource matrix. Let $T_i(\omega)$ be the i-th row of $\boldsymbol{T}(\omega)$. A random vector $\chi(\omega) = [\boldsymbol{q}(\omega), \boldsymbol{h}(\omega), T_1(\omega), \ldots, T_{m_2}(\omega)]^{\top} \in \mathbb{R}^{n_2 + m_2 + (m_2 \times n_1)}$ is implicitly defined in MP 1, whose formulation was originally stated in [2,3].

MP 1 Two-stage stochastic mathematical programming formulation with fixed resources.

$$\begin{aligned}
\min \quad Z &= \boldsymbol{c}^{\top}\boldsymbol{y} + \mathbb{E}_{\boldsymbol{\xi}}[\min \boldsymbol{q}(\omega)^{\top}\boldsymbol{x}] \\
\text{subject to} \quad \boldsymbol{A}\boldsymbol{y} &= \boldsymbol{b}, \\
\boldsymbol{T}(\omega)\boldsymbol{y} + \boldsymbol{W}\boldsymbol{x} &= \boldsymbol{h}(\omega), \\
\boldsymbol{y} &\geq \boldsymbol{0}, \\
\boldsymbol{x} &\geq \boldsymbol{0}.
\end{aligned}$$

3. Simple Plant Location Problem with Order

In this section, we define the SPLPO. Let $I = \{1, \ldots m\}$ be a set of customers, and let $J = \{1, \ldots n\}$ be a set of possible sites where facilities can be opened. Let J^i be a subset of J defined for each customer i, with $|J^i| = n_i$ and $\overline{J^i} = J \setminus J^i$. Unit costs $c_{ij} \geq 0$ are considered to supply the demand of customer i from facility j, and fixed costs $f_j \geq 0$ to open a facility at location j. We say that k is i-worse than j if the customer i prefers facility j rather than k, which is denoted by $k <_i j$. We define $W_{ij} = \{k \in J \mid k <_i j\}$ as the set of facilities k strictly i-worse than j, with its complement being denoted as $\overline{W_{ij}}$ and $W_{ij} \cup \{j\}$ as W'_{ij}. Let x_{ij} be a decision variable that represents the fraction of the demand required by the customer i and covered by facility j. Let y_j be a binary variable such that $y_j = 1$ if a facility is open at the location j, and $y_j = 0$ otherwise. We assume that each customer i classifies her/his favorite facility $j \in J^i$ with a number $p_{ij} \in \{1, \ldots, n_i\}$, where 1 and n_i are the most and least preferred, respectively. Additionally, for all $j \in \overline{J^i}$, we have that $p_{ij} = n_i + 1$. Under these conditions, the linear program for the SPLPO is given by MP 2.

MP 2 SPLPO mathematical programming formulation.

$$\begin{aligned}
\min \quad Z &= \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} f_j y_j, && (2)\\
\text{subject to} \quad \sum_{j\in J} x_{ij} &= 1, \quad \forall i \in I, && (3)\\
x_{ij} &\le y_j, \quad \forall i \in I, j \in J, && (4)\\
\sum_{k\in \overline{W_{ij}}} x_{ik} &\ge y_j, \quad \forall i \in I, j \in J^i, && (5)\\
x_{ij} &\ge 0, \quad \forall i \in I, j \in J, && (6)\\
y_j &\in \{0,1\}, \quad \forall j \in J. && (7)
\end{aligned}$$

The statement expressed in (2) says that the program minimizes a cost function related to opening a facility covering the demand. Equalities presented in (3) ensure that customer i is supplied by exactly facility j, called assignment constraints. Constraints given in (4) ensure that if customer i is supplied by facility j, then j must be opened, which are often called varying upper bounds. Inequalities stated in (5) model the customers' orders of preference [26]. Since no capacities are considered and the model minimizes the number of open facilities y, the demand of customer i can always be covered completely by one single facility. Therefore, we can guarantee that there is an optimal solution with the values of the variables x_{ij} belonging to $\{0,1\}$, even though this is not specified in the constraints; see inequalities given in (6). The family of inequalities presented in (7) makes the binary nature of the variable y_j explicit. Note that if it is more favorable to open an installation that does not belong to a particular subset J^i, customer i can be served by any open installation, since $n_i + 1$ is the worst classification given for any j.

4. Stochastic Formulation for the SPLPO

In this section, we define the 2S-SPLPO. In the context of the SPLPO, consider the experiment of asking all customers $i \in I$ to rate their preferred facilities $j \in J$. Such as that defined in MP 1, we call each result of this experiment ω and the set of all of them the sample space Ω. Recall the random variable ξ with values $\xi(\omega)$ over the events $\{\,\omega\,\} \in \Omega$ called the scenario ω. Once again, we assume that this random variable has the discrete probability function stated in (1). Under this setting and the mentioned probability function, a mathematical program for the 2S-SPLPO (complete or partial) can be written as in MP 3.

MP 3 2S-SPLPO mathematical programming formulation.

$$\begin{aligned}
\min \quad Z &= \sum_{j\in J} f_j y_j + \sum_{\omega\in\Omega} \alpha^{\omega} \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij}^{\omega} && (8)\\
\text{subject to} \quad \sum_{j\in J} x_{ij}^{\omega} &= 1, \quad \forall i \in I, \omega \in \Omega, && (9)\\
x_{ij}^{\omega} &\le y_j, \quad \forall i \in I, j \in J, \omega \in \Omega, && (10)\\
\sum_{k\in \overline{W_{ij}^{\omega}}} x_{ik}^{\omega} &\ge y_j, \quad \forall i \in I, j \in J^i, \omega \in \Omega, && (11)\\
x_{ij}^{\omega} &\ge 0, \quad \forall i \in I, j \in J, \omega \in \Omega, && (12)\\
y_j &\in \{0,1\}, \quad \forall j \in J. && (13)
\end{aligned}$$

Note that the mathematical expressions defined in (8)–(13) of MP 3 have a similar interpretation as those stated in MP 2 for each scenario ω. The second component established in (8) represents the expected value of the distribution cost over the scenarios. Consider that the first decision stage to open a facility is given by binary variable y and the distribution of the demand is given by x as the second stage. Regarding the family of constraints formulated as in (11) of MP 3, each scenario ω is represented by a $\{0,1\}$-matrix $\boldsymbol{W}(\omega)$ ($\overline{\boldsymbol{W}(\omega)}$) of dimension $(mn \times mn)$, where each row ij has a one for each element of W_{ij}^{ω} ($\overline{W_{ij}^{\omega}}$) and a zero for elements of its complement; see Figure 1. Furthermore, Ω is a finite set but it can have a big cardinality. For example, if we consider no partial preferences and since $n < +\infty$, each row of $\boldsymbol{W}(\omega)$ ($\overline{\boldsymbol{W}(\omega)}$) can be given in 2^n different ways. Thus, there are $(2^n)^{mn} = 2^{mn^2}$ possible different matrices $\boldsymbol{W}(\omega)$ ($\overline{\boldsymbol{W}(\omega)}$).

$$\begin{array}{cc} 1\left\{\begin{array}{c}1\\2\\\vdots\\n\end{array}\right. \\ 2\left\{\begin{array}{c}1\\2\\\vdots\\n\end{array}\right. \\ \vdots \\ m\left\{\begin{array}{c}1\\2\\\vdots\\n\end{array}\right. \end{array} \underbrace{\begin{bmatrix} 1&0&\cdots&1&&&&&&&&&\\ 0&1&\cdots&0&&&&&&&&&\\ \vdots&\vdots&\ddots&\vdots&&&&&&&&&\\ 1&1&\cdots&1&&&&&&&&&\\ &&&&0&1&\cdots&0&&&&&\\ &&&&0&0&\cdots&1&&&&&\\ &&&&\vdots&\vdots&\ddots&\vdots&&&&&\\ &&&&1&1&\cdots&1&&&&&\\ &&&&&&&&\ddots&&&&\\ &&&&&&&&&1&0&\cdots&1\\ &&&&&&&&&1&1&\cdots&0\\ &&&&&&&&&\vdots&\vdots&\ddots&\vdots\\ &&&&&&&&&1&0&\cdots&1 \end{bmatrix}}_{\overline{W(\omega)}\in\mathbb{R}^{(mn\times mn)}} \begin{bmatrix} x_{11}^{\omega}\\ x_{12}^{\omega}\\ \vdots\\ x_{1n}^{\omega}\\ \hline x_{21}^{\omega}\\ x_{22}^{\omega}\\ \vdots\\ x_{2n}^{\omega}\\ \hline \vdots\\ \hline x_{m1}^{\omega}\\ x_{m2}^{\omega}\\ \vdots\\ x_{mn}^{\omega} \end{bmatrix} \geq \begin{bmatrix} y_1\\ y_2\\ \vdots\\ y_n\\ \hline y_1\\ y_2\\ \vdots\\ y_n\\ \hline \vdots\\ \hline y_1\\ y_2\\ \vdots\\ y_n \end{bmatrix}$$

Figure 1. An example of the matrix $\overline{W(\omega)}$.

Note that the stochastic program formulated in MP 3 has no fixed resources since the coefficient matrix $\overline{\boldsymbol{W}(\omega)}$ given by the sets $\overline{W_{ij}^{\omega}}$ with the second-stage variables x_{ij}^{ω} is random, that is, it depends on ω.

5. A Lagrangian Algorithm for the 2S-SPLPO

In this section, we state a Lagrangian algorithm for the 2S-SPLPO. As noted in MP 3, since the mathematical formulation of the 2S-SPLPO has similar features to its non-stochastic version (SPLPO), with obvious differences due to the presence of multiple random scenarios, we suggest applying the ADA algorithm. Figure 2 shows the scheme of the ADA algorithm, which includes three modules that we have broken down into internal steps in a very general way. The SLR theory and some important results, which are exposed in the description of the ADA algorithm, can be found in [15,22], where this algorithm was introduced. The main idea of the SLR approach is to relax a direction of an equality constraint in the Lagrangian sense, keeping the other one without relaxing, and then to take advantage of the mathematical properties of the resulting programming. Next, we detail each of the steps of the ADA algorithm summarized in Figure 2.

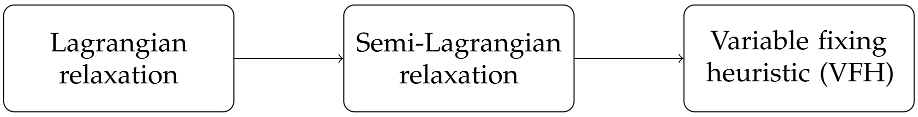

Figure 2. Scheme of steps in the ADA algorithm.

Step 1: Lagrangian relaxation

In MP 3, we relax two families of constraints: those of assignment stated in (9) and those of customer preference defined in (11), obtaining the Lagrangian program given by

$$\begin{aligned}\mathrm{LR}(\boldsymbol{\mu}^\omega, \boldsymbol{\lambda}^\omega) \quad &:= \quad \min_{(\boldsymbol{x}^\omega, \boldsymbol{y})} \sum_j f_j y_j + \sum_\omega \alpha^\omega \sum_i \sum_j c_{ij} x_{ij}^\omega \\ &\quad + \sum_\omega \sum_i \mu_i^\omega \Big(1 - \sum_j x_{ij}^\omega\Big) + \sum_\omega \sum_i \sum_j \lambda_{ij}^\omega \Big(y_j - \sum_{k \in \overline{W_{ij}^\omega}} x_{ik}^\omega\Big) \\ &= \quad \min_{(x^\omega, y)} \sum_\omega \sum_i \sum_j \Big(\alpha^\omega c_{ij} x_{ij}^\omega - \mu_i x_{ij}^\omega - \lambda_{ij}^\omega \sum_{k \in \overline{W_{ij}^\omega}} x_{ik}^\omega\Big) \\ &\quad + \sum_j \Big(f_j y_j + \sum_\omega \sum_i \lambda_{ij}^\omega y_j\Big) + \sum_\omega \sum_i \mu_i^\omega,\end{aligned} \tag{14}$$

subject to: (10), (12), (13).

In the case where a partial order is considered, for all $j \in \overline{J^i}$, the corresponding terms $\lambda_{ij}^\omega \sum_{k \in \overline{W_{ij}^\omega}} x_{ik}^\omega$ and $\sum_\omega \sum_i \lambda_{ij}^\omega y_j$ vanish in the objective function defined in (14).

As before, we are assuming that customer i ranks for each scenario her/his favorite facility $j \in J^i$ with a number $p_{ij}^\omega \in \{1, \ldots, n_i\}$, where 1 and n_i are the most and least preferred, respectively. Since in $\sum_\omega \sum_i \sum_{j \in J^i} \lambda_{ij}^\omega \sum_{k \in \overline{W_{ij}^\omega}} x_{ik}^\omega$ each λ_{ij}^ω is multiplied by a sum of x_{ij}^ω variables with p_{ij}^ω terms corresponding to $k \geq_i j$, then each x_{ij}^ω is multiplied by a sum of λ_{ik}^ω with $n_i - p_{ij}^\omega + 1$ terms corresponding to $k \leq_i j$. Thus, we can check the following:

$$\sum_\omega \sum_i \sum_{j \in J^i} \lambda_{ij}^\omega \sum_{k \in \overline{W_{ij}^\omega}} x_{ik}^\omega = \sum_\omega \sum_i \sum_{j \in J^i} x_{ij}^\omega \sum_{k \in W_{ij}^{\prime\omega}} \lambda_{ik}^\omega, \tag{15}$$

where $|W_{ij}^{\prime\omega}| = n_i - p_{ij}^\omega + 1$.

By using the expression given in (15), the objective function defined in (14) can be rewritten as

$$\begin{aligned}\mathrm{LR}(\boldsymbol{\mu}^\omega, \boldsymbol{\lambda}^\omega) \quad &:= \quad \min_{(\boldsymbol{x}^\omega, \boldsymbol{y})} \sum_j \Big[\sum_\omega \sum_i \Big(\alpha^\omega c_{ij} - \mu_i^\omega - \Lambda_{ij}^\omega\Big) x_{ij}^\omega \\ &\quad + \Big(f_j + \sum_\omega \sum_i \lambda_{ij}^\omega\Big) y_j\Big] + \sum_\omega \sum_i \mu_i^\omega,\end{aligned} \tag{16}$$

subject to: (10), (12), (13),

where $\Lambda_{ij}^\omega = \sum_{k \in W_{ij}^{\prime\omega}} \lambda_{ik}^\omega$, with $|W_{ij}^{\prime\omega}| = n_i - p_{ij}^\omega + 1$.

Similar to its non-stochastic version, with and without order, SPLP and SPLPO, respectively, the problem stated in (16), with multipliers $\boldsymbol{\mu}^\omega$ and $\boldsymbol{\lambda}^\omega$ fixed, can be easily solved by considering that

$$y_j = \begin{cases} 1, & \text{if } \sum_i \min(0, \alpha^\omega c_{ij} - \mu_i^\omega - \Lambda_{ij}^\omega) + \Big(f_j + \sum_\omega \sum_i \lambda_{ij}^\omega\Big) < 0, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$x_{ij}^\omega = \begin{cases} 1, & \text{if } y_j = 1 \text{ and } \alpha^\omega c_{ij} - \mu_i^\omega - \Lambda_{ij}^\omega < 0, \\ 0, & \text{otherwise.} \end{cases}$$

The problem stated in (16) has the integrality property, that is, its optimal value is equal to the standard linear relaxation LP (2S-SPLPO), which corresponds to the original problem without the integrality constraints. The ADA algorithm tries to solve the LD problem related to (16) established as

$$\mathrm{LD}(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega}) := \max_{(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega})} \mathrm{LR}(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega}),$$

with a SGM, due to the fact that its objective function is a piecewise linear concave function of vector multipliers $\boldsymbol{\mu}^{\omega}$ and $\boldsymbol{\lambda}^{\omega}$ over its domain. Hence, after some iterations, the multipliers obtained are used as a starting point for the second step. The SGM requires multiple times to solve the problem given in (16) , but as mentioned, it is simple to solve; see Algorithm 1, and for its step in line see (12) and also [19–21].

Algorithm 1: Subgradient method.

```
1  begin
       input  : The 2S-SPLPO approach.
       output : A lower bound for the 2S-SPLPO, multipliers $(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega})$.
2      $\texttt{iter} = 0$;
3      $\texttt{Maxiter} \in \mathbb{N}$;
4      Compute $\mathrm{LR_{ub}}$ for $\mathrm{LR}(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega})$ by applying a heuristic procedure;
5      $\beta = 2$;
6      $\texttt{k} \in \mathbb{N}$;
7      $\texttt{q} \in [0,1] \subset \mathbb{R}$;
8      $[\boldsymbol{\mu}^{\omega,\texttt{iter}}, \boldsymbol{\lambda}^{\omega,\texttt{iter}}]^{\top} = [\min_j \{ \alpha^{\omega} c_{ij} + f_j, \forall j \}, 0]^{\top}$;
9      $\mathrm{LR}^{\texttt{iter}}_{\mathrm{best}} = \mathrm{LR}(\boldsymbol{\mu}^{\omega,\texttt{iter}}, \boldsymbol{\lambda}^{\omega,\texttt{iter}})$;
10     repeat
11         Find a subgradient vector $\boldsymbol{s}^{\texttt{iter}}$ for $\mathrm{LR}(\boldsymbol{\mu}^{\omega,\texttt{iter}}, \boldsymbol{\lambda}^{\omega,\texttt{iter}})$ as
               $\boldsymbol{s}^{\texttt{iter}} = \left[ 1 - \sum_{j \in J} x^{\omega}_{ij},\ \forall i, \omega; \quad y_j - \sum_{k \in \overline{W_{ij}}} x^{\omega}_{ik},\ \forall i, j, \omega \right]^{\top} = [\boldsymbol{s}^{\texttt{iter}}_{\boldsymbol{\mu}}, \boldsymbol{s}^{\texttt{iter}}_{\boldsymbol{\lambda}}]^{\top}$
           if $\boldsymbol{s}^{\texttt{iter}} \neq \mathbf{0}$ then
12             $\alpha^{\texttt{iter}} = \frac{\beta(\mathrm{LR_{ub}} - \mathrm{LR}(\boldsymbol{\mu}^{\omega,\texttt{iter}}, \boldsymbol{\lambda}^{\omega,\texttt{iter}}))}{\|\boldsymbol{s}^{\texttt{iter}}\|^2}$;
13         end
14     until Either $\boldsymbol{s}^{\texttt{iter}} = \mathbf{0}$ or $\texttt{iter} = \texttt{Maxiter}$;
15 end
```

It is possible to prove that the vector computed in step 11 of Algorithm 1 is a subgradient of the Lagrangian problem. In our experiments, we obtain better results when $\mu^{\omega,0}_i = \min_j \{ \alpha^{\omega} c_{ij} + f_j \}$, for all $i \in I$ and $\omega \in \Omega$ instead $\mu^{\omega,0} = 0$. In step 4 of Algorithm 1, we set $\mathrm{LR_{ub}}$ by applying an upper bound heuristic for the 2S-SPLP (H2S); see Algorithm 2. First, it opens a facility with the lowest operating cost for all customers. Then, for those facilities that are not open yet, the procedure compares and chooses for each customer the most preferred facility between it and its previously assigned supplier. Consequently, the new cost is saved and the new open facility has the lowest operating cost. This is repeated until no more facilities are available.

Algorithm 2: Upper bound heuristic for the 2S-SPLP.

```
begin
    input  : A bipartite graph $G(I \cup J, E^\omega = \emptyset)$, costs $c_{ij}$ to compute the total cost (TC).
    output : An upper bound $G(I \cup J, E^\omega)$ for the 2S-SPLPO.
    Find $j^* \in J$ such that $\sum_{i,\omega} \alpha^\omega c_{ij^*} = \min\{\sum_{i,\omega} \alpha^\omega c_{i1}, \ldots, \sum_{i,\omega} \alpha^\omega c_{in}\}$;
    $J' = J \setminus \{j^*\}$, $E^\omega_m = \{(i,j^*)^\omega \mid i \in I, \omega \in \Omega\}$, $\text{TC}_{j^*} = \sum_{i,\omega} \alpha^\omega c_{ij^*}$;
    for $\omega \in \Omega$ do $E^\omega_r = E^\omega_{j^*}$;
    repeat
        for $j \in J'$ do
            for $\omega \in \Omega$ do $E^\omega_j = \emptyset$;
            for $i \in I$ and $\omega \in \Omega$ do
                Find $k_{\text{pref}}$ such that $p^\omega_{ik_{\text{pref}}} = \min[\{p^\omega_{ij}\} \cup \{p^\omega_{ik} \mid (i,k) \in E^\omega_r\}]$;
                $E^\omega_j = E^\omega_j \cup \{(i,k_{\text{pref}})^\omega\}$;
            end
            Compute $\text{TC}_j = \sum_{((i,j)^\omega \in E^\omega_j, \omega)} \alpha^\omega c_{ij}$;
        end
        Find $j_r$ such that $\text{TC}_{j_r} = \min\{\text{TC}_j \mid j \in J'\}$;
        $J' = J' \setminus \{j_r\}$;
        for $\omega \in \Omega$ do $E^\omega_r = E^\omega_{j_r}$;
        Find $j^*$ such that $\text{TC}_{j^*} = \min\{\text{TC}_{j^*}, \text{TC}_{j_r}\}$;
        for $\omega \in \Omega$ do $E^\omega = E^\omega_{j^*}$;
    until $J' = \emptyset$;
end
```

Step 2: Semi-Lagrangian relaxation

We start with a brief description of the SLR. Let the problem (P) of mathematical program be stated as

$$(\text{P}) := \min_{\boldsymbol{x}} \{ f(\boldsymbol{x}) = \boldsymbol{c}^\top \boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathbb{X} \}, \tag{17}$$

with (i) the set $\mathbb{X}$ containing integrality constraints; and (ii) $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and rational components $\boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^n$ being non-negative.

After splitting $\boldsymbol{Ax} = \boldsymbol{b}$ into $\boldsymbol{Ax} \geq \boldsymbol{b}$ and $\boldsymbol{Ax} \leq \boldsymbol{b}$, inequality $\boldsymbol{Ax} \geq \boldsymbol{b}$ is relaxed with a vector of multipliers $\boldsymbol{\lambda} \geq \boldsymbol{0}$. Then, the SLR problem related to (P) is expressed as

$$\text{SLR}(\boldsymbol{\lambda}) := \min_{\boldsymbol{x}} \{ f(\boldsymbol{x}) + \boldsymbol{\lambda}(\boldsymbol{b} - \boldsymbol{Ax}) \mid \boldsymbol{Ax} \leq \boldsymbol{b}, \boldsymbol{x} \in \mathbb{X} \},$$

with its SLD being given by $\text{SLD}(\boldsymbol{\lambda}) := \max_{\boldsymbol{\lambda} \geq \boldsymbol{0}} \text{SLR}(\boldsymbol{\lambda})$.

The following theorems summarizes the basic properties derived from the definition of the SLR.

Theorem 1 ([15])**.** *Let* (P) *be a problem as defined in* (17) *under the same conditions. Then, we have the following:*

(i) *SLR*$(\boldsymbol{\lambda})$ *is concave, non-decreasing on its domain, and* $\boldsymbol{b} - \boldsymbol{Ax}$ *is a subgradient at the point* $\boldsymbol{\lambda}$*.*
(ii) *There is an interval* $[\boldsymbol{\lambda}^*, +\infty)$ *where for each multiplier, we obtain the optimal solution of SLR*$(\boldsymbol{\lambda})$*.*
(iii) *LP*$(\text{P}) \leq$ *LD*$(\boldsymbol{\lambda}) \leq$ *SLD*$(\boldsymbol{\lambda}) = (\text{P})$*, that is, SLR*$(\boldsymbol{\lambda})$ *closes the duality gap.*

In MP 3, we semi-relax the assignment constraints defined in (9). Thus, we obtain the program formulated as

$$\begin{aligned}\text{SLR}(\gamma^{\omega}) := & \min_{(x^{\omega},y)} \sum_{\omega}\sum_{i}\sum_{j} c_{ij}x_{ij}^{\omega} + \sum_{j} f_j y_j + \sum_{\omega}\sum_{i}\gamma_i^{\omega}\Big(1 - \sum_{j} x_{ij}^{\omega}\Big) \\ = & \min_{(x^{\omega},y)} \sum_{j}\Big(\sum_{\omega}\sum_{i}\Big(\alpha^{\omega}c_{ij} - \gamma_i^{\omega}\Big)x_{ij}^{\omega} + f_j y_j\Big) + \sum_{\omega}\sum_{i}\gamma_i^{\omega},\end{aligned}$$

subject to (10), (11), (12), (13), and $\sum_{j\in J} x_{ij}^{\omega} \leq 1, \forall i \in I, \omega \in \Omega$.

The corresponding dual problem is stated as $\text{SLD}(\gamma^{\omega}) := \max_{\gamma^{\omega}} \text{LR}(\gamma^{\omega})$. We follow the same idea given in [15], in the context of the SLR applied to the uncapacitated facility location problem, to restrict our search for the vector of multipliers γ^{ω} in $\text{SLD}(\gamma^{\omega})$.

Theorem 2. *Let SLR(γ^{ω}) and $c_i^{(\omega)} = \max_j\{\alpha^{\omega}c_{ij} + f_j\}$, $\mathbf{c}^{\omega} = (c_1^{(\omega)}, \ldots, c_m^{(\omega)})$ be the maximal cost for costumer i associated with facility j and the vector of these costs, respectively. Then, we obtain $\gamma^{\omega} \in Q = [\gamma^*, +\infty)$ if $\gamma^{\omega} \geq \mathbf{c}^{\omega}$.*

Proof. If $\sum_{j\in J} x_{ij}^{\omega} = 1$ for all i and $\omega \in \Omega$, the SLR closes the duality gap. By hypothesis, $c_i^{(\omega)} - \gamma_i^{\omega} \leq 0$. If we choose j' such that $c_i^{(\omega)} = \alpha^{\omega}c_{ij'} + f_{j'}$, then $(\alpha^{\omega}c_{ij'} - \gamma_i^{\omega}) + f_{j'} \leq 0$. This inequality is true for any j since j' gives the maximum among all $\alpha^{\omega}c_{ij} + f_j$. Therefore, the event $\sum_{j\in J} x_{ij}^{\omega} = 0$ cannot happen at an optimal solution since it is always possible to set $x_{ij}^{\omega} = 1$ and $y_j = 1$ for all i and j, meeting all the constraints. □

Theorem 3. *Let SLR(γ^{ω}). For each $i \in I$ and $\omega \in \Omega$, let $\alpha^{\omega}c_i^{(1)} \leq \cdots \leq \alpha^{\omega}c_i^{(n)}$ be the sorted cost $\alpha^{\omega}c_{ij}$. Then, if $\gamma^{\omega} < \alpha^{\omega}\mathbf{c}^{\mathbf{1}}$, $\gamma^{\omega} \notin Q$.*

Proof. By hypothesis, $\alpha^{\omega}c_i^{(1)} - \gamma_i^{\omega} > 0$. Then, we have that $\alpha^{\omega}c_i^{(j)} - \gamma_i^{\omega} > 0$ for all j since we are minimizing at the optimal solution $x_{ij}^{\omega} = 0$ for all j. Therefore, the event $\sum_{j\in J} x_{ij} = 1$ cannot happen at an optimal solution and $\gamma^{\omega} \notin Q$. □

Theorems 2 and 3 limit the search of γ^{ω} to $B = \{\gamma^{\omega} \mid \alpha^{\omega}\mathbf{c}^{(\mathbf{1})} < \gamma \leq \mathbf{c}^{\omega}\}$. We use the multipliers $\boldsymbol{\mu}^{\omega}$ obtained in the previous step as starting point in this step by setting $\gamma^{\omega} = \boldsymbol{\mu}^{\omega}$. Then, based on Theorems 2 and 3, we solve the LD problem by increasing the components of γ^{ω} in each iteration of a DAM that requires solving SLR(γ^{ω}) many times; see Algorithm 3. However, this problem is harder than LR$(\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega})$. The DAM algorithm employs the following:

- By using the sorted costs $\alpha^{\omega}c_i^{(1)} \leq \cdots \leq \alpha^{\omega}c_i^{(n=|J|)}$, each component γ_i^{ω} of γ^{ω} can be either in an interval of the form $(\alpha^{\omega}c_i^{(j)}, \alpha^{\omega}c_i^{(j+1)}]$ or out of it. For the first case, there are infinite values of γ_i^{ω} that can belong to a single interval $(\alpha^{\omega}c_i^{(j)}, \alpha^{\omega}c^{(j+1)}]$. Each one of them has the same effect in the solution of SLR(γ^{ω}) since only a change from $y_j = 1$ to $y_{j+1} = 1$ and $x_{ij}^{\omega} = 1$ to $x_{i,j+1}^{\omega} = 1$ modifies the solution.
- We just need a single γ_i^{ω} representative of the intervals. As we get closer to solving the SLR, the values of the components of γ^{ω} increase. Hence, solving the SLR becomes more and more difficult. Then, it is always convenient to choose a $\gamma_i^{\omega} \in I_i$ being as small as possible, that is, at an ε distance from the lower bound of an interval.

Algorithm 3: Dual ascent method.

```
begin
    input  : A SLR($\gamma^\omega$) problem.
    output : A lower bound for SLR($\gamma^\omega$), multipliers $\gamma^\omega$.
    For all $i \in I$ and $\omega \in \Omega$, sort $\alpha^\omega c_{ij}$ as $\alpha^\omega c_i^{(1)} \leq \cdots \leq \alpha^\omega c_i^{(n)}$;
    $\varepsilon > 0$, $\texttt{iter} = 0$, $\texttt{MAXiter} \in \mathbb{N}$;
    for $i \in I$ and $\omega \in \Omega$ do
        $c_i^{(\omega)} = \max_j\{\alpha^\omega c_{ij} + f_j\}$;
        if $\gamma_i^{\omega,\texttt{iter}} < \alpha^\omega c_i^{(1)}$ then
            $\gamma_i^\omega = \alpha^\omega c_i^{(1)} + \varepsilon$;
        else if $\gamma_i^{\omega,\texttt{iter}} > \alpha^\omega c_i^{(n)}$ then
            $\gamma_i^{\omega,\texttt{iter}} = \alpha^\omega c_i^{(n)}$;
        else
            Find a $j_i$ such that $\gamma_i^{\omega,\texttt{iter}} \in (\alpha^\omega c_i^{(j_i)}, \alpha^\omega c_i^{(j_i+1)}]$;
            $\gamma_i^{\omega,\texttt{iter}} = \alpha^\omega c_i^{(j_i)} + \varepsilon$;
        end
    end
    repeat
        Use an exact method to solve SLR($\gamma^{\omega,\texttt{iter}}$);
        Find a subgradient $s^{\texttt{iter}} = \left[1 - \sum_{j \in J} x_{ij}^\omega\right]$, $\forall i, \omega$;
        if $s^{\texttt{iter}} \neq \mathbf{0}$ then
            for $i \in I$ and $\omega \in \Omega$ do
                if $s_i^{\omega,\texttt{iter}} = 1$ then
                    $j_i = j_i + 1$;
                    $\gamma_i^{\texttt{iter}+1} = \min\{\alpha^\omega c_i^{(j_i)} + \varepsilon, c_i^{(\omega)}\}$
                end
            end
            $\texttt{iter} = \texttt{iter} + 1$;
        end
    until Either $s^{\texttt{iter}} = \mathbf{0}$ or $\texttt{iter} = \texttt{MAXiter}$;
end
```

Step 3: Variable fixing heuristic

A heuristic method is used to speed up the search process to the optimum. In a run of the DAM, the procedure chooses by a cost criteria, from the solution given, a subset corresponding to a percentage ps × 100% of the best variables y_j, such that $y_j = 1$. These y are fixed to solve the 2S-SPLPO in the next iteration by an exact method. The procedure is described in Algorithm 4. In addition, the procedures described above are assembled in the ADA method stated in Algorithm 5.

Algorithm 4: Variable fixing heuristic.

```
begin
    input  : A vector $\gamma^\omega$ in a solution given by the DAM.
    output : A feasible solution for the SPLPO (probably the optimum).
    $Y_{\gamma^\omega} = \{ y_j \mid y_j = 1, j \in J \}$;
    ps $\in [0, 1]$ (a percentage of elements of $Y_\gamma^\omega$);
    sort $Y_{\gamma^\omega}$ such that $y_j \prec y_k$ if $\sum_\omega \sum_i (\alpha^\omega c_{ij} + f_j) < \sum_\omega \sum_i (\alpha^\omega c_{ik} + f_k)$;
    choose the first ps × 100% smallest elements from the sorted set $Y_{\gamma^\omega}$ to form the set $Y_{\gamma^\omega}^{\text{subset}}$;
    fix $y_j = 1, \forall y_j \in Y_{\gamma^\omega}^{\text{subset}}$;
    solve the 2S-SPLPO with an exact method;
end
```

Algorithm 5: Accelerated dual ascent algorithm.

```
begin
    input  : The 2S-SPLPO approach.
    output : A feasible solution for the 2S-SPLPO.
    $\mu_i^{\omega,0} = \min_j\{\alpha^\omega c_{ij} + f_j\}, \forall i \in I, \omega \in \Omega$, $\boldsymbol{\lambda}^{\omega,0} = \mathbf{0}$;
    run SGM($\boldsymbol{\mu}^{\omega,0}, \boldsymbol{\lambda}^{\omega,0}$) during sgm_iter iterations and find the iteration best_sgm_iter where is the best LR($\boldsymbol{\mu}^{\omega}, \boldsymbol{\lambda}^{\omega}$) value;
    $\gamma^{\omega,0} = \boldsymbol{\mu}^{\omega,\texttt{best\_sgm\_iter}}$;
    run DAM($\gamma^{\omega,0}$) during dam_iter iterations;
    run the DAM during vfh_iter iterations. In each iteration, run VFH;
    find the best solution in the searching process to obtain the best values for $x_{ij}^{\omega}$ and $y_j$;
end
```

6. Sensing Patients' and Simulated Data for the Proposed Methodology

In this section, we propose an algorithm that allows us to update patients' records and store them in a central data warehouse. Consider COVID-19 vaccination centers. Patients go to the different care points based on an order of preference. At each point, there is a local repository that stores the data of the patients who were treated at that point. However, the data must be centralized in a single database for persistent storage purposes and also for later analysis. With a certain periodicity (for example, at the end of the day), a sensor located at each local repository sends the data related to the care of each of the patients to the central warehouse. It is important to keep in mind that each vaccination center usually stores the local data for one day of care. In a general scenario, Algorithm 6 permits us to centralize, in a warehouse, the data obtained when patients are served in the facilities. In addition, to simulate data and solve instances of the 2S-SPLPO, we propose Algorithm 7 which sensorizes the database.

Algorithm 6: Approach for updating patients' records in a central data warehouse using sensors.

1: Clear the local data repository of each facility at the beginning of the operations.
2: Store a data record in the local repository for each patient and for each facility.
3: Send the patients' records to a central data warehouse by a sensor located at the local data repository of the facility at the end of the operations.
4: Store the patient's records for all facilities permanently in the central data warehouse.
5: Use the central warehouse for any data analysis.

Algorithm 7: Approach to generate and solve instances of the 2S-SPLPO with a sensor.

1: Generate data for the second scenario with the simulation algorithm.
2: Build an instance of the 2S-SPLPO with the data simulated in the previous step for the data taken in the first scenario.
3: Store the instance in the database.
4: Detect the instance of the 2S-SPLPO with a sensor in the database and dynamically build the mathematical model.
5: Call the ADA algorithm to solve the mathematical model with the sensor.
6: Receive the mathematical model and solve it with the ADA algorithm.
7: Store the output of the ADA algorithm in the database and generate tables with the execution results.

7. Computational Experiments

In this section, we show the results of computational experiments obtained after applying the ADA algorithm to a set of 2S-SPLPO (complete preferences) with two scenarios. Problems with one scenario were taken from [27], which are based on the Beasley OR-Library [28]. The second scenario was randomly generated. In order to simulate data and solve instances of the 2S-SPLPO, we use Algorithm 7 which sensorizes the database.

The experiments were carried out on a PC with Intel® Xeon® 3.40 GHz processor and 16 GB of RAM under a Windows® 10 operating system. All problems were solved with FICO XPRESS® version 8.0.

Table 1 shows the parameter settings for all steps of the ADA algorithm, which were set after an extensive experimental process. We observe that there is a positive relation between the number of iterations of the SGM and the size of the problem, with `sgm_iter` being set at a value greater than or equal to n. Due to the difficult to solve SLR(γ^{ω}) in the DAM, `sgm_iter` was set with a value less than a 10% of n.

Table 1. Parameter settings of the ADA algorithm for the indicated method. Source: the authors.

m	n	SGM				DAM		VFH	
		sgm_iter	β	k	q	dam_iter	ε	vfh_iter	ps
75	50	50	2	30	0.005	3	0.01	2	0.25
100	75	100	2	30	0.005	7	0.01	2	0.25
125	100	170	2	30	0.005	10	0.01	2	0.25
150	100	170	2	30	0.005	12	0.01	2	0.25

The names of the problems in Tables 2 and 3 use the nomenclature as given in the following example: `[10a][7550]_[2]`, where `[10a]`: The second scenario is 10% different from the first scenario `a`; `[7550]`: $m = 75$ and $n = 50$; and `[2]`: Problem 2 with first scenario `a`. In the literature on the topic (for example, in p. 145, Section 4 of [27]), the instances shown in Table 3 are considered large. Note that, although the times in many cases increase as the size of the problem increases, in some cases, this behavior is not detected. For example, in Table 3, the ADA algorithm solves the problem `100b10075_1` in 3531 seconds. However, in a larger instance, such as `100a150100_1`, the time is less (1253 seconds). Since the ADA algorithm is a heuristic, the running times can vary, even on the same problem, due to the random component that the VFH has. Therefore, we cannot affirm the exponential growth of the running times.

The results show that the ADA algorithm performs very well in all cases. Indeed, the optimum was found in most of them. Note that when the percentage of changed preferences was 10% (first six instances in Table 2), the ADA algorithm was not able to improve the XPRESS times. Nevertheless, in the remaining cases, the times were improved considerably with a couple of exceptions. From Table 3, observe that the ADA algorithm attains the optimal solution in most of the cases in much less time, with the exception of the problem `100c150100_1`, where the time was greater than that obtained by the XPRESS software. The last of the four groups corresponds to the same problems in the third group of Table 3 ($m = 150$, $n = 100$). Nonetheless, they were tested with a different parameter `sgm_iter` $= 140$. For all these six cases, the ADA algorithm improves the running times and bounds. The column headers in Tables 2 and 3 mean the following:

- Prob: Name of the problem.
- Opt: Optimal value of the problem.
- LP(P): Linear relaxation value for a problem (P).
- $\text{GAP}_\text{X} = (\text{Opt} - \text{LP})/\text{Opt} \times 100\%$: Relative gap between Opt and LP of a problem by using the XPRESS software.
- t: Time in seconds
- H2Sub: Best upper bound with the H2S.
- y: Number of opened facilities.
- SGMlb: Lower bound with the SGM.
- ADAub: Best upper bound with the ADA algorithm.
- DAMlb: Lower bound with the DAM (without the VFH).
- $\text{GAP}_\text{V} = (\text{bestUB} - \text{LB})/\text{bestUB} \times 100\%$: Relative gap between the best upper bound and lower bound of a problem by using the DAM with VHF.
- Tt: Total time in seconds.

- imp t: $((\mathrm{t_{opt}} - \mathrm{t_{ADA}})/\mathrm{t_{opt}}) \times 100\%$
- GAP: $((\mathrm{ADAub\text{-}Opt})/\mathrm{Opt}) \times 100\%$.

Table 2. ADA algorithm applied to the 2S-SPLPO with two scenarios ($m = 75$, $n = 50$). Source: the authors.

Prob	XPRESS				H2S			SGM		DAM with VFH				ADA		
	Opt	**LP**	**$\mathrm{GAP_X}$**	**t**	**H2Sub**	**y**	**t**	**SGMlb**	**t**	**ADAub**	**DAMlb**	**$\mathrm{GAP_V}$**	**t**	**Tt**	**imp t**	**GAP**
10a7550_1	1,648,853	1,200,745	27%	67	1,787,955	1	0	840,831	7	1,698,650	1,279,041	25%	60	67	0%	3.02%
10a7550_2	1,602,647	1,193,378	26%	48	1,738,071	3	0	763,664	6	1,602,647	1,286,736	20%	50	56	−17%	0.00%
10b7550_1	1,226,979	901,051	27%	48	1,335,244	10	0	334,589	5	1,226,979	953,159	22%	57	62	−28%	0.00%
10b7550_2	1,263,465	906,280	28%	53	1,299,259	10	0	372,981	7	1,271,641	970,770	24%	70	77	−46%	0.65%
10c7550_1	1,290,291	900,033	30%	46	1,388,280	9	0	591,938	7	1,290,291	971,785	25%	62	69	−48%	0.00%
10c7550_2	1,248,312	876,053	30%	47	1,365,717	13	0	557,308	6	1,248,312	936,483	25%	65	71	−51%	0.00%
25a7550_1	1,670,734	1,199,273	28%	119	1,761,050	5	0	830,624	6	1,670,734	1,284,052	23%	53	59	50%	0.00%
25a7550_2	1,606,360	1,195,482	26%	67	1,730,389	3	0	770,870	7	1,606,360	1,269,850	21%	53	60	11%	0.00%
25b7550_1	1,256,755	892,106	29%	58	1,329,801	11	0	360,026	6	1,259,217	947,822	25%	61	67	−15%	0.20%
25b7550_2	1,324,578	907,308	32%	153	1,348,842	7	0	380,921	6	1,324,578	988,646	25%	98	104	32%	0.00%
25c7550_1	1,364,933	907,957	33%	166	1,413,489	9	0	579,362	8	1,373,542	978,857	29%	96	104	37%	0.63%
25c7550_2	1,271,029	873,435	31%	84	1,291,327	13	0	578,313	7	1,271,029	958,058	25%	77	84	0%	0.00%
50a7550_1	1,689,428	1,212,042	28%	113	1,778,238	6	0	804,221	7	1,689,428	1,302,236	23%	60	67	41%	0.00%
50a7550_2	1,637,084	1,196,440	27%	86	1,744,372	5	0	760,256	6	1,637,084	1,273,267	22%	51	58	33%	0.00%
50b7550_1	1,331,207	900,654	32%	165	1,388,062	12	0	294,443	7	1,331,207	1,247,451	6%	381	388	−136%	0.00%
50b7550_2	1,307,706	896,181	31%	182	1,364,303	11	0	305,591	7	1,307,706	990,369	24%	94	101	45%	0.00%
50c7550_1	1,347,482	911,370	32%	145	1,446,345	12	0	565,472	6	1,347,482	976,268	28%	77	84	42%	0.00%
50c7550_2	1,287,982	878,852	32%	112	1,439,845	17	0	546,735	7	1,299,952	952,655	27%	71	78	30%	0.93%
100a7550_1	1,787,955	1,208,842	32%	267	1,787,955	1	0	824,731	8	1,787,955	1,293,098	28%	89	97	64%	0.00%
100a7550_2	1,683,058	1,204,739	28%	194	1737,,924	5	0	827,668	7	1,720,482	1,268,026	26%	66	72	63%	2.22%
100b7550_1	1,451,139	935,113	36%	363	1,496,648	6	6	311,032	6	1,453,678	989,340	32%	86	92	75%	0.17%
100b7550_2	1,400,184	916,167	35%	271	1,449,128	9	0	358,259	7	1,400,205	1,013,362	28%	106	113	58%	0.00%
100c7550_1	1,360,674	920,342	32%	169	1,430,298	9	0	562,495	8	1,360,674	998,681	27%	106	114	32%	0.00%
100c7550_2	1,402,514	918,696	34%	196	1,462,990	16	0	570,058	8	1,404,147	989,592	30%	102	110	44%	0.12%

Table 3. ADA algorithm applied to the 2S-SPLPO with two scenarios for $m \in \{100, 125, 150\}$ and $n \in \{75, 100\}$. Source: the authors.

Prob	XPRESS				H2S			SGM		DAM with VFH				ADA		
	Opt	**LP**	**$\mathrm{GAP_X}$**	**t**	**H2Sub**	**y**	**t**	**SGMlb**	**t**	**ADAub**	**DAMlb**	**$\mathrm{GAP_V}$**	**t**	**Tt**	**imp t**	**GAP**
100a10075_1	2,469,439	1,811,464	27%	561	2,476,632	1	1	1,644,492	31	2,476,632	1,978,083	20%	265	296	47%	0.29%
100a10075_2	2,458,870	1,805,025	27%	685	2,476,632	1	1	1,627,278	32	2,458,870	1,971,011	20%	237	270	61%	0.00%
100b10075_1	2,132,719	1,364,985	36%	17,935	2,270,467	6	1	1,112,051	37	2,132,719	1,558,555	27%	3531	3568	80%	0.00%
100b10075_2	2,163,818	1,367,450	37%	27679	2,218,215	7	1	1,160,875	43	2,163,818	1,564,516	28%	4380	4422	84%	0.00%
100c10075_1	1,978,807	1,271,848	36%	14,835	2,072,702	6	1	1,052,860	49	1,988,903	1,496,210	25%	1027	1076	93%	0.51%
100c10075_2	1,987,757	1,261,290	37%	11,567	2,118,928	9	1	1,066,388	51	1,987,757	1,452,609	27%	3152	3202	72%	0.00%
100a125100_1	3,070,535	2,416,518	21%	918	3,070,535	1	2	2,237,118	93	3,070,535	2,619,571	15%	573	666	27%	0.00%
100a125100_2	3,070,535	2,388,054	22%	1088	3,070,535	1	1	2,239,726	106	3,070,535	2,587,669	16%	702	808	26%	0.00%
100b125100_1	2,800,573	1,815,018	35%	53,666	2,850,413	5	2	1,601,481	118	2,850,413	2,078,979	27%	20,837	20,955	61%	1.78%
100b125100_2	2,820,883	1,820,001	35%	78,669	3,019,740	4	1	1,592,305	143	2,820,883	2,016,632	29%	8230	8373	89%	0.00%
100c125100_1	2,702,169	1,698,737	37%	239,967	2,866,218	10	2	1,488,148	157	2,702,169	1,990,068	26%	23,717	23,874	90%	0.00%
100c125100_2	2,716,252	1,705,149	37%	204,007	2,829,945	5	1	1,477,796	168	2,717,597	2,007,831	26%	27,442	27,610	86%	0.05%
100a150100_1	3,768,087	2,924,250	22%	1735	3,768,087	1	1	2699949	109	3,768,087	3239975	14%	905	1014	42%	0.00%
100a150100_2	3,768,087	2,918,397	23%	1819	3,768,087	1	1	2702231	111	3,768,087	3,251,719	14%	117	228	87%	0.00%
100b150100_1	3,412,417	2,179,897	36%	169,739	3,637,438	1	1	1,923,456	141	3,412,417	2,599,980	24%	21,111	21,252	87%	0.00%
100b150100_2	3,388,309	2,196,284	35%	69,508	3,637,438	1	2	1,924,300	169	3,388,309	2,679,093	21%	14792	14,962	78%	0.00%
100c150100_1	3,287,595	2,010,587	39%	502,354	3,413,288	4	2	1,768,341	185	3,413,288	N/A	N/A	N/A	N/A	N/A	3.82%
100c150100_2	3,229,424	2,012,045	38%	494,721	3,307,459	5	3	1,776,475	132	3,300,341	2,474,515	25%	119,700	119,832	76%	2.20%
100a150100_1	3,768,087	2,924,250	22%	1735	3,768,087	1	1	2,635,877	94	3,768,087	3,229,888	14%	1159	1253	28%	0.00%
100a150100_2	3,768,087	2,918,397	23%	1819	3,768,087	1	1	2,637,900	109	3,768,087	3,242,679	14%	1255	1364	25%	0.00%
100b150100_1	3,412,417	2,179,897	36%	169,739	3,637,438	1	2	1,876,775	147	3,445,585	2,607,746	24%	21,590	21,738	87%	0.97%
100b150100_2	3,388,309	2,196,284	35%	69,508	3,637,438	4	4	1,863,975	155	3,388,309	2,599,780	23%	8984	9139	87%	0.00%
100c150100_1	3,287,595	2,010,587	39%	502,354	3,413,288	4	2	1,750,243	179	3,288,348	2,457,997	25%	78,577	78756	84%	0.02%
100c150100_2	3,229,424	2,012,045	38%	494,721	3,307,459	5	3	1,718,731	185	3,230,261	2,470,331	24%	144,729	144,729	71%	0.03%

The symbol N/A in the table is used when we do not get the result.

8. Conclusions, Limitations, and Future Research

Our work was motivated from a problem related to healthcare services, where service centers must be sited in strategic locations that meet the immediate needs of patients. The COVID-19 pandemic makes this problem particularly relevant [29]. We have provided an algorithm that stores patients' records in a central data warehouse using sensors. This allowed us to propose a potential application of our new algorithm to COVID-19 vaccination based on sensor-related data. Specifically, in our investigation, we have proposed a Lagrangian-based approach for the 2S-SPLPO, called the ADA algorithm. It has three steps that are related and work consecutively to create a heuristic procedure that takes advantage of the structure of the problem.

In summary, this paper reported the following findings:

(i) Formulations for SPLPO and 2S-SPLPO with partial preferences were proposed.
(ii) Lagrangian and semi-Lagrangian structures for the 2S-SPLPO with partial preferences were introduced.
(iii) A theoretical analysis of properties of the Lagrangian and semi-Lagrangian structures for the 2S-SPLPO was presented to combine them in a procedure that approximates its solution.
(iv) Theorems 2 and 3 were stated as extensions for the 2S-SPLPO of those given in [15] for the uncapacitated facility location problem.
(v) To the best of our knowledge, there have been no algorithms proposed to solve the stochastic version of the SPLPO. The proposed algorithm is a novel approach that uses a relatively new optimization technique known as semi-Lagrangian relaxation.
(vi) The computational experiments suggested that the ADA algorithm performed satisfactorily on large instances in both search of the optimum and execution time.
(vii) Possible applications in real cases are described in the context of the COVID-19 vaccination process and B2C e-commerce.

Some limitations of our study are the following:

(i) Since the last step of the ADA algorithm is a heuristic, the optimal is not guaranteed.
(ii) The ADA algorithm is studied in the context of the 2S-SPLPO, so its use is limited to it. We suggest studying the same ideas in other location problems.
(iii) A cost–benefit evaluation should be carried out in the use of the proposed algorithm to answer the following question: is the improvement in execution times worth it, with respect to the savings obtained in the value of the objective function?
(iv) Studies and experiments with deeper parameter settings and more scenarios must be performed on larger instances.
(v) We suggest carrying out a computational experiment that allows us to determine the parameter values of the mathematical optimization model from which it is necessary to design and implement a heuristic or metaheuristic algorithm. High computational complexity problems can be solved using the heuristic approach. Some examples of this can be found in [30–32].

The authors are working on these and other issues associated with the present investigation. The corresponding findings are expected to be reported in future works.

Author Contributions: Data curation, X.C., E.D., C.M.-B.; investigation, X.C., S.G.; formal analysis and methodology, X.C., S.G., E.D., C.M.-B., V.L.; writing—original draft, X.C.; writing—review and editing, X.C., C.M.-B., V.L. All authors have read and agreed to the submitted version of the manuscript.

Funding: This research was partially supported by FONDECYT, project grant number 1200525 (V.L.), from the National Agency for Research and Development (ANID) of the Chilean government under the Ministry of Science, Technology, Knowledge and Innovation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The data analyzed are available upon request.

Acknowledgments: The authors would also like to thank the editor and three reviewers for their constructive comments, which led to improving the presentation of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Birge, J.; Louveaux, F. *Introduction to Stochastic Programming*; Springer: New York, NY, USA, 2014.
2. Dantzig, G. Linear programming under uncertainty. *Manag. Sci.* **1955**, *1*, 197–206. [CrossRef]
3. Beale, E. On minimizing a convex function subject to linear inequalities. *J. R. Stat. Soc. Ser. B* **1955**, *17*, 173–184. [CrossRef]
4. Liu, C.; Fan, Y.; Ordóñez, F. A two-stage stochastic programming model for transportation network protection. *Comput. Oper. Res.* **2009**, *36*, 1582–1590. [CrossRef]
5. Zhou, Z.; Guan, Y. Two-stage stochastic lot-sizing problem under cost uncertainty. *Ann. Oper. Res.* **2013**, *209*, 207–230. [CrossRef]
6. Rojas, F.; Leiva, V.; Wanke, P.; Lillo, C.; Pascual, J. Modeling lot-size with time-dependent demand based on stochastic programming and case study of drug supply in Chile. *PLoS ONE* **2019**, *14*, e0212768. [CrossRef]
7. Li, C.; Grossmann, I.E. A review of stochastic programming methods for optimization of process systems under uncertainty. *Front. Chem. Eng.* **2021**, *2*, 34. [CrossRef]
8. Cornuéjols, G.; Fisher, M.; Nemhauser, G. Location of bank accounts to optimize float: An analytic study of axact and approximated algorithms. *Manag. Sci.* **1977**, *23*, 789–810. [CrossRef]
9. Laporte, G.; Louveaux, F. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **1993**, *13*, 133–142. [CrossRef]
10. Kall, P.; Mayer, J. *Stochastic Linear Programming: Models, Theory, and Computation*; Springer: New York, NY, USA, 2005.
11. Ilonen, J.; Kamarainen, J.K.; Lampinen, J. Differential Evolution Training Algorithm for Feed-Forward Neural Networks. *Neural Process. Lett.* **2003**, *7*, 93–105. [CrossRef]
12. Cabezas, X.; García, S. A Lagrangian relaxation algorithm for the simple plant location problem with preferences. *arXiv* **2018**, arXiv:1805.03945.
13. Cabezas, X. Heuristic Methods for Solving Two Discrete Optimization Problems. Ph.D. Thesis, The University of Edinburgh, Edinburgh, UK, 2018.
14. Martin-Barreiro, C.; Ramirez-Figueroa, J.A.; Cabezas, X.; Leiva, V.; Galindo-Villardon, M.P. Disjoint and functional principal component analysis for infected cases and deaths due to COVID-19 in South American countries with sensor-related data. *Sensors* **2021**, *21*, 4094. [CrossRef]
15. Beltrán, C.; Vial, J.; Alonso, A. Semi-Lagrangian relaxation applied to the uncapacited facility location problem. *Computat. Optim. Appl.* **2012**, *51*, 387–409. [CrossRef]
16. Held, M.; Karp, R. The traveling salesman problem and minimum spanning trees: Part II. *Math. Program.* **1971**, *1*, 6–25. [CrossRef]
17. Held, M.; Wolfe, P.; Crowder, H. Validation of subgradient optimization. *Math. Program.* **1974**, *6*, 62–88. [CrossRef]
18. Guignard, M. Lagrangian relaxation. *Top* **2003**, 11, 151–200. [CrossRef]
19. Polyak, B.T. A general method for solving extremum problems. *Sov. Math. Dokl.* **1967**, *8*, 593–597.
20. Bagirov, A.M.; Gaudioso, M.; Karmitsa, N.; Mäkelä, M.M.; Taheri, S. (Eds.) *Numerical Nonsmooth Optimization. State of the Art Algorithms*; Springer: Cham, Switzerland , 2020.
21. Conforti, M.; Cornuéjols, G.; Zambelli, G. *Integer Programming*; Springer: New York, NY, USA, 2014.
22. Beltrán, C.; Tandoki, C.; Vial, J. Solving the p-Median Problem with a Semi-Lagrangian Relaxation. *Comput. Optim. Appl.* **2006**, *35*, 239–260. [CrossRef]
23. Doumari, S.A.; Givi, H.; Dehghani, M.; Montazeri, Z.; Leiva, V.; Guerrero, J.M. A New Two-Stage Algorithm for Solving Optimization Problems. *Entropy* **2021**, *23*, 491. [CrossRef]
24. Zhang, H.; Beltran-Royo, C.; Wang, B.; Ma, L.; Zhang, Z. Solution to the quadratic assignment problem using semi-lagrangian relaxation. *J. Syst. Eng. Electron.* **2016**, *27*, 1063–1072. [CrossRef]
25. Zhang, H.; Beltran-Royo, C.; Wang, B.; Zhang, Z. Two-phase semi-Lagrangian relaxation for solving the uncapacitated distribution centers location problem for B2C E-commerce. *Comput. Optim. Appl.* **2019**, *72*, 827–848. [CrossRef]
26. Hanjoul, P.; Peeters, D. A facility location problem with clients' preference orderings. *Reg. Sci. Urban Econ.* **1987**, 17, 451–473. [CrossRef]
27. Cánovas, L.; García, S.; Labbé, M.; Marín, A. A strengthened formulation for the simple plant location problem with order. *Oper. Res.Lett.* **2006**, *35*, 141–150. [CrossRef]
28. Beasley, E. OR-Library: Distributing test problems by electronic mail. *J. Oper. Res. Soc.* **1990**, *41*, 1069–1072. [CrossRef]
29. Rojas, F.; Leiva, V.; Huerta, M.; Martin-Barreiro, C. Lot-size models with uncertain demand considering its skewness/kurtosis and stochastic programming applied to hospital pharmacy with sensor-related COVID-19 data. *Sensors* **2021**, *21*, 5198. [CrossRef]
30. Cabezas, X.; García, S.; Salas, S. A hybrid heuristic approach for traffic light synchronization based on the MAXBAND. *Soft Comput. Lett.* **2019**, *1*, 100001. [CrossRef]

31. Ramirez-Figueroa, J.A.; Martin-Barreiro, C.; Nieto-Librero, A.B.; Leiva, V.; Galindo-Villardón, M.P. A new principal component analysis by particle swarm optimization with an environmental application for data science. *Stoch. Environ. Res. Risk Assess.* **2021**, in press. [CrossRef]
32. Martin-Barreiro, C.; Ramirez-Figueroa, J.A.; Nieto-Librero, A.B.; Leiva, V.; Martin-Casado, A.; Galindo-Villardón, M.P. A new algorithm for computing disjoint orthogonal components in the three-way Tucker model. *Mathematics* **2021**, *9*, 203. [CrossRef]