Article

The Role of Metaheuristics as Solutions Generators

Hanane El Raoui [1,2,3], **Marcelino Cabrera-Cuevas** [4] **and David A. Pelta** [2,*]

1   TICLab, ESIN—Ecole Supérieure d'Informatique et du Numérique, International University of Rabat, Rabat 11100, Morocco; hanane.elraoui@uir.ac.ma
2   Department of Computer Science and Artificial Intelligence, Universidad de Granada, 18014 Granada, Spain
3   Modeling and Mathematical Structures Laboratory, Sidi Mohamed Ben Abdellah University, Fez 30000, Morocco
4   Department of Languages and Information Systems, Universidad de Granada, 18014 Granada, Spain; mcabrera@ugr.es
*   Correspondence: dpelta@ugr.es

Abstract: Optimization problems are ubiquitous nowadays. Many times, their corresponding computational models necessarily leave out of consideration several characteristics and features of the real world, so trying to obtain the optimum solution can not be enough for a problem solving point of view. The aim of this paper is to illustrate the role of metaheuristics as solutions' generators in a basic problem solving framework. Metaheuristics become relevant in two modes: firstly because every run (in the case of population based techniques) allows to obtain a set of potentially good solutions, and secondly, if a reference solution is available, one can set up a new optimization problem that allows to obtain solutions with similar quality in the objectives space but maximally different structure in the design space. Once a set of solutions is obtained, an example of an a posteriori analysis to rank them according with decision maker's preferences is shown. All the problem solving framework steps, emphasizing the role of metaheuristics are illustrated with a dynamic version of the tourist trip design problem (for the first mode), and with a perishable food distribution problem (for the second one). These examples clearly show the benefits of the problem solving framework proposed. The potential role of the symmetry concept is also explored.

Keywords: metaheuristics; optimization; tourist trip design; vehicle routing

1. Introduction

In most 'real world' decision-making and/or optimization problems, countless decision criteria and objectives can not be evidently included in their computational model formulation; or, if included, they may increase the complexity of such models up to a point where they can not be exactly solved. Final decisions (i.e., selecting a solution to deploy) are often taken based not only on modeled objectives, but also on potentially subjective decision makers goals, biases, and preferences [1].

Many mathematical optimization algorithms are devoted to finding single optimal solutions to single-objective problems or, in the best case, to determine sets of non-inferior solutions to multi-objective formulations [2–4]. In any case, it should never be forgotten that we are, hopefully, obtaining the optimum solution for the model.

Assuming the existence of unmodeled (or hard to model) goals and parameters implies that non-conventional solving approaches are needed, not only to search the decision space for optimal solutions, but also to explore the decision region for alternative solutions with good quality. Such solutions can be sub-optimal for the model point of view, but valuable from additional perspectives.

It is here where the role of metaheuristics as solutions' generators becomes relevant in two senses. Firstly, because several runs (a single one, in the case of population based techniques) allow to obtain a set of potentially good solutions. Secondly, let us suppose a reference solution x^* for a problem P is available. Then, it is possible to define a new

optimization problem P' where the objective is to find a solution x that maximize the difference with x^* subject to $f(x^*) - f(x) \leq \gamma$, being f the objective function from problem P and γ the maximum variation in quality allowed. In other words, the aim in P' is to obtain solutions with similar quality but maximally different structure. This last approach is the so-called 'Modeling to Generate Alternatives' (MGA) approach proposed by [4–6].

Setting apart this MGA approach, the consideration of metaheuristics in the sense posed here is not being explored (to the best of our knowledge). Most of the metaheuristics publications emphasize their role as "solvers" trying to find the best possible solution for the problem at hand usually without making further analysis.

In this context, the aim of this contribution, is to illustrate the role of metaheuristics as solutions' generators in a basic problem solving framework that considers the problem formulation, its resolution, and the analysis of solutions beyond their objective function value.

Two examples are considered. The first problem is a dynamic version of the tourist trip design problem. In the static version, we have a set of points of interest (POI), each one having a visit time and a level of interest. The aim is to select a subset of points that maximizes the level of interest subject to certain available time. The problem is usually modeled as a team orienteering problem [7].

In the dynamic version, the level of interest of a POI depends on the time that it is being visited, thus adding an extra level of complexity. We will describe the many characteristics associated with a solution and how difficult it is to include them *a priori* in a model, thus requiring a different solving strategy.

The second one, is a perishable food distribution problem (basically, a vehicle routing problem) [8]. Originally modeled as a many objectives' problem, we describe a solving strategy that first solves a sub-problem and then, using the obtained solution, sets a new optimization problem with a fuzzy constraint that allows to obtain new solutions with similar quality but maximally different structure (as in the MGA approach).

The paper is structured as follows. In Section 2 the proposed basic problem solving framework is presented, together with an existing approach to rank solutions according to the user's preference. Next, Sections 3 and 4 present the two examples commented before. For each one, every step of the solving framework is illustrated with a specific dataset, and the corresponding analyses of results are presented.

In each example, a metaheuristic is used to generate a set of solutions. We should highlight here that the specific details of the implemented methods are omitted, as they are not relevant in this context: we do not want to make comparisons against other methods over a set of well-defined objectives.

Finally, Section 5 is devoted to conclusions and further work.

2. Basic Problem Solving Framework

The role of metaheuristics as solutions' generators is best seen when integrated in the following problem solving framework. Although every step is clear enough, we describe the mains aspects considered here.

Let us depart from an optimization problem P. The following steps are required to solve it.

1. **Problem description**: determine a set of solutions' features for problem P, $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$. Such features are used to assess the quality of a solution;
2. **Model formulation**: Define the subset of features to be used in P optimization model $\mathcal{F}' = \{f_i, f_j, \ldots, f_t\}$. For every feature, either a maximization or minimization goal should be established;
3. **Problem Solving**: "Solve" the problem P. Depending on the solver algorithm:

    (a)   An optimum solution s^* (or at least a reference one) is obtained. From such solution, generate new ones with similar quality but as different as possible;

    (b)   A set of solutions is obtained;

4. **Analysis of Solutions**: For every solution, calculate the corresponding values for the features in $\mathcal{F} - \mathcal{F}'$;
5. **Ranking**: Rank the solutions according to the user's preferences.

Let us illustrate these steps with the well known travelling salesmen problem (TSP). Given a solution (a route), features like the distance travelled, time of the route, fuel consumption, average length of the inter-city paths, the length of the longest path, and so on can be measured.

In a basic TSP formulation, a single goal is defined: distance minimization.

It is in Step 3 where metaheuristics comes into play. Suppose we solve the problem using a genetic algorithm, so it is easy to obtain a set of routes (Step 3.b). For example, the best m solutions from the final population are kept.

Then, in Step 4, those m solutions can be organized in a matrix M as:

	$\mathcal{F} - \mathcal{F}'$			$\mathcal{F}'$
	f_1	f_2	...	f_n
s_1	M_{11}	M_{12}	...	M_{1n}
s_2	M_{21}	M_{22}	...	M_{2n}
s_m	M_{m1}	M_{m2}	...	M_{mn}

where M_{ij} is the value achieved by solution s_i on feature f_j. Those values M_{ij} with $j \in \mathcal{F}'$ are given as the output of the optimization process, while those for $j \in \mathcal{F} - \mathcal{F}'$ are calculated after the optimization.

Finally, in Step 5 the values in every M row can be combined (for example using some aggregation function [9]) to obtain a score q_i for every solution s_i. The relevance of the objectives for the user can be expressed through a set of weights. Sorting according to q_i allows to rank the solutions.

Many options are available for this step. Here, we will use the approach in [10] that allows to select a solution of interest from a set of solutions. The user states the preferences, just providing a linear ordering of the objectives. A brief summary of the approach follows.

2.1. Ranking Solutions According to the User's Preferences

Let us depart from the solutions' matrix M. Additionally, suppose the user provides a set of weights $W = \{w_1, w_2, \ldots, w_n\}$ where if the feature f_i is more relevant than f_j for the decision maker, then $w_i \geq w_j$. It should hold that $\sum_{j=1}^{m} w_j = 1$. Under these assumptions, a score q_i for the solution s_i can be calculated as $\sum_{j=1}^{m} w_j \times x_{ij}$.

However, this approach has a problem. Let us suppose we have just three criteria and the given preference order is f_2, f_1, f_3. Then, we need the weights in such a way that $w_2 \geq w_1 \geq w_3$ with $w_1 + w_2 + w_3 = 1$. As the reader may notice, there are infinite values for w_i that verifies both conditions, and every possible set of values will give a different score for the alternative.

Instead of a single score value, the authors in [10] proposed to calculate an interval of the potential scores that a solution can attain and then sort the solutions in terms of their intervals.

The main steps are detailed below.

1. Solutions' matrix normalization: normalization is needed because different measurements should be combined;
2. Intervals calculation: the user's preferences are given as an ordinal relation among features denoted as $f_1 \succeq_p f_2 \succeq_p \ldots \succeq_p f_n$. The symbol $\succeq_p$ is to be read as "at least as preferred to". This implies that the weights are ordered as $w_1 \geq w_2 \geq \ldots \geq w_m$.
    All the potential scores that a solution s_i can attain are included in the interval denoted as $I_i = [L_i, U_i]$ with L_i, U_i are, respectively, the minimum, and the maximum scores (obtained through solving two simple linear programming problems. See [10] fo details);

3. Select a reference solution s^*: such solution s^* and its corresponding interval $I^* = [L^*, U^*]$ is the one with the greatest lower bound $\forall i, L^* \geq L_i$, thus there is no solution that always scores better than s^*;

4. Compare solutions against s^*: every solution s_i is compared against s^* using a possibility function that measures the possibility degree of a solution being better than another using their corresponding intervals.

Let us have two solutions s_X, s_Y with their corresponding non-negative intervals $X = [x_l, x_r], Y = [y_l, y_r]$ and $x_l, x_r, y_l, y_r \in R_0^+$. The possibility degree of s_X being greater than s_Y is stated in terms of $P(X \geq Y)$ as in [10] and considering a user with a neutral attitude is defined as follows [11]:

(a) if $X \cap Y = \varnothing$ (intervals do not overlap)

$$P(X \geq Y) = \begin{cases} 0 & x_r \leq y_l \\ 1 & x_l \geq y_r \end{cases}$$

(b) if $X \cap Y \neq \varnothing$ (intervals overlap)

$$P(X \geq Y) = \frac{x_r - y_l}{x_r - x_l + y_r - y_l}$$

5. Ranking of Solutions: for every solution, s_i the value $P(s_i \geq s^*)$ is calculated. Then, they are are sorted based on such possibility degree values.

3. The Time-Dependant Tourist Trip Design Problem

The tourist trip design problem (TTDP) involves two essential aspects: the selection of points of interest suitable to the tourist's preferences, and the generation of itineraries to visit these points (all or a subset of them). Although the first aspect is usually related to recommendation systems, which have to identify user preferences, the second is related to decision and optimization issues.

In general, TTDP models are based on the well-known *orienteering problem* (OP), *team orienteering problem* (TOP) and their variants for route design ([12–16].

3.1. Problem Description

The tourist trip design problem with time-dependent scores (TTDP-TDS) starts from a set N of nodes or points of interest (POIs). Each node $i \in N$ has an associated interest or score S_i and several recommendation factors f_{it} to weight the interest according to the time at which the node is visited. The travel time from node i to node j is given by t_{ij}.

On the other hand, not all nodes in the set can be visited due to there is a maximum available time (T_{max}). There is a node n_{ini} defined as the starting and ending point of the route (for example, the location of a hotel).

A solution to the problem will be a permutation of a subset of nodes (POIs), i.e., the ordered list of POIs to visit. Over such a list, it is easy to imagine the many features that could be calculated.

3.1.1. Features

- I, Overall interest;
- $\#POIs$, Number of POIs in the route;
- T_{visit}, Time spent visiting POIs;
- T_{travel}, Total walking time between POIs on the route;
- T_{route}, Overall trip duration: $T_{visit} + T_{travel}$;
- E, Route "efficiency": $100 \times (1 - T_{visit}/t_{route})$, the percentage of time in the POIs with respect to the trip time;
- Average, maximum. and minimum time spent in POIs;
- Average, maximum and minimum travel time between POIs;
- Average number of POIs visited in every time period;

- Percentage of POIs visited at the best time of the day.

Considering all of these features in a single model is, if not impossible, very difficult. Moreover, different optimization problems can be defined: Max *I*, Max #*POIs*, (Max *I*, Min #*POIs*), (Max *I*, Max *E*), etc.

In this contribution, we depart from the following model.

3.1.2. Parameters

- N—Node set (the POIs);
- n_{ini}—Index of the start/end node of the route;
- T—Number of time periods;
- S_i—Interest of node i;
- t_{ij}—Travel time between node i and node j;
- f_{it}—Recommendation factor to visit node i in period t;
- v_i—Time required to visit node i (service time);
- T_{max}—Maximum time available for the route.

3.1.3. Decision Variables

The following variables are considered:

- y_{it}, a binary variable. If the node i is visited in time period t, then $y_{it} = 1$, and 0 otherwise;
- π, a permutation of k elements (the length of route) where $\pi(i)$ represents the node that is visited at position i of the route.

3.1.4. Objective Function

$$Max \ I = \sum_{i \in N \setminus \{n_{ini}\}} \sum_{t \in T} S_i \cdot f_{it} \cdot y_{it} \tag{1}$$

subject to:

$$T_{visit} + T_{travel} \leq T_{max} \tag{2}$$

where

$$T_{visit} = \sum_{i \in N \setminus \{n_{ini}\}} \sum_{t \in T} v_i \cdot y_{it} \tag{3}$$

$$T_{travel} = t_{ini,\pi(1)} + \left(\sum_{i=1}^{k-1} t_{\pi(i),\pi(i+1)} \right) + t_{\pi(k),ini} \tag{4}$$

Although a visit to a point can span two different time periods, here the contribution of such point to the overall interest is determined by the period at which its visit starts.

3.2. Problem Solving and Generation of Solutions

We describe next the basic details of the metaheuristic used, the dataset and the way the set of solutions is generated.

3.2.1. Metaheuristic

A crossover-less evolutionary algorithm (EA) is used to solve this problem.

Each individual of the initial population is generated using a GRASP-like procedure. Initially, an initial POI is randomly selected. Subsequently, a list of candidate POIs is constructed, an element from the list is randomly chosen and then added to the solution. The process is repeated as long as the maximum time available is not exceeded.

In each generation, from the current population of *popSize* individuals, *K* parents are chosen by roulette wheel selection and incorporated into an intermediate population. Using one of the several mutation operators available, a child solution is generated from

each parent. If the new solution is feasible and better than the parent, it is incorporated into the intermediate population.

Finally, the intermediate population is sorted, and the best *popSize* elements are taken to replace the original population.

The proposed algorithm always operates with feasible solutions. Therefore, non-feasible solutions obtained by mutation are discarded.

3.2.2. Dataset

A dataset with the 50 most visited POIs of Granada city, Spain is used. The distribution of these POIs is shown in Figure 1 and the dataset is available at http://18.156.111.23/TTDP-TDS/web/ (accessed on 1 August 2021).
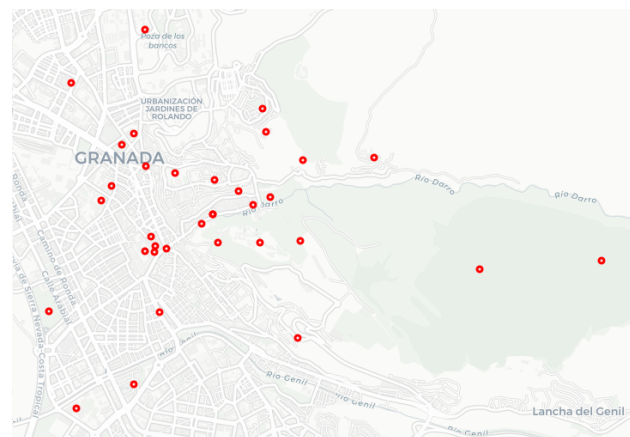
Figure 1. Map of Granada and set of points of interest.

Four time slots ($T = 4$) and a value $T_{max} = 360$ are considered. Each time slot has a duration of 90 min.

3.2.3. Generation of Solutions

The generation of solutions is done running the EA several times. More specifically, we made 40 independent runs of the EA with different parameters. Every run deals with $popSize = 50$ solutions and 100 generations. After every run, we kept the best solution, thus obtaining a set of 40 solutions.

3.3. Analysis of Solutions' Diversity

In order to explore the relation between the interest and the diversity of the solutions, we will use the Jaccard's similarity coefficient to assess how similar two solutions A, B (considered as two sets of POIs) are. The coefficient is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

It is easy to see that if routes A and B contain the same POIs, $A = B = A \cap B = A \cup B$, then $J(A, B) = 1$. On the other hand, if routes A and B do not share any element at all, $|A \cap B| = 0$, so $J(A, B) = 0$.

From the set of 40 solutions $S = \{s_1, s_2, \ldots, s_{40}\}$, we select the best one in terms of interest (s^*) and compute $J(s^*, s_i) \, \forall \, i$.

The scatter plot in Figure 2 displays the relation between similarity and interest difference for every $s_i \in S$ and s^*.

Solutions 5, 12, 14, 5 have the same interest that s^* but a different structure, with a Jaccard coefficient ≤ 0.8. There are other three solutions (3,4, 10) with a minor difference in cost (less than 2 units) and rather different from s^*, ($J < 0.7$). If the user may allow

routes with less interest (up to 6%) then all the solutions in the range of interest difference between 2 and 4 are available.
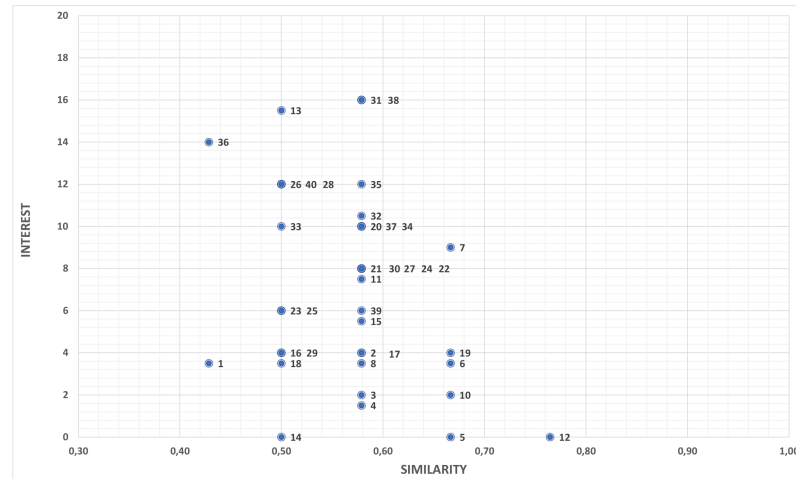
Figure 2. Similarity vs. Interest difference for every solution against s^*. The lower the values in both axis, the better the solution is.

3.4. Users Profiles and Ranking

Now, having those 40 solutions, we calculate the values of several additional features (beyond the route interest I) which are:

- I, Route Interest;
- $\#POIs$, Number of visited POIs;
- E, Route efficiency;
- T_{travel}, total walking time.

The corresponding values for every solution are shown in Table A1 (Appendix B), while Table 1 summarizes the values of the features over the 40 solutions. We can observe that a wide variety of values are available. There are solutions ranging from 11 to 15 POIs, with an interest which is (in the worst case) 18% lower than the best solution. Regarding the travel time, values from 25 min to almost an hour of walking between POIs can be considered and looking at the efficiency, it is clear that the solutions are very effective at reducing the travel time while maximizing the visit time.

Table 1. Summary of values for every feature considered, over the generated 40 solutions.

	Min	Max	Mean	Std. Dev.
Number of POIs	11	15	13.33	1.57
Interest	74.00	90.00	83.01	4.57
Travel Time	24.78	54.78	37.19	6.36
Efficiency	84.77	92.60	89.18	1.60

Now, let us define the following three profiles.

1. **Guided tour**: A tourist wanting to visit as many of the city's most interesting POIs as possible while maximizing the time available. To do so, we rank the solutions taking the interest of the solution as the main criterion, ordering the rest of features as $I \succeq_p \#POIs \succeq_p E \succeq_p T_{travel}$;
2. **Casual visit**: A visitor, who is in the city for another reason, wants to visit the most interesting places in the city but wants to have as much free time as possible during

this visit. In this case, we maximize the Effectiveness of the visit and order the rest of the criteria by $E \succeq_p I \succeq_p \#POIs \succeq_p T_{travel}$;

3. **Large groups or people with reduced mobility**: Due to the size of the groups or the mobility problems of the visitors, it is necessary to keep the journeys to be made as short as possible. In this last profile we minimize travel time between POIs T_{travel}, and we set the order of the rest of the criteria by $T_{travel} \succeq_p I \succeq_p \#POIs \succeq_p E$

For every profile, the procedure described in Section 2.1 is applied. Figure 3 shows the best 10 solutions of the corresponding rankings. The values for the reference solution (s^*) are indicated in the top row, and the columns are ordered in terms of the criteria importance. Values in red cells are worse than the reference value, while the green ones are better.

Regarding Profile 1, the reference solution $s^* = s_9$ is the top ranked alternative. Solution 4 has a slightly lower value of interest while attaining the same values in the rest of features. Nevertheless, this solution is quite different from s^*, having $J(s^*, s_4) = 0.58$.

Solutions 5, 12, and 14 have the same value of interest, but they are slightly worse in the rest of the factors.

Ref. Solution	Interest	POIs	Efficiency	Travel time
	90,00	15	90,57%	33,83
9	0,00	0	0,00%	0,00
4	-1,50	0	0,00%	0,00
14	0,00	-1	-0,41%	1,09
12	0,00	-1	-1,24%	4,39
1	-3,50	0	-0,69%	1,64
5	0,00	0	-2,96%	10,70
16	-4,00	0	-0,82%	2,69
6	-3,50	0	-1,46%	5,25
17	-4,00	0	-1,34%	4,77
8	-3,50	0	-1,77%	5,27

(**a**) Profile 1

Ref. Solution	Efficiency	Interest	POIs	Travel time
	90,57%	90,00	15	33,83
9	0,00%	0,00	0	0,00
4	0,00%	-1,50	0	0,00
37	2,03%	-10,00	-3	-9,05
1	-0,69%	-3,50	0	1,64
39	0,90%	-6,00	-3	-5,41
14	-0,41%	0,00	-1	1,09
11	-0,30%	-7,50	0	0,64
16	-0,82%	-4,00	0	2,69
6	-1,46%	-3,50	0	5,25
17	-1,34%	-4,00	0	4,77

(**b**) Profile 2

Ref. Solution	Travel time	Interest	POIs	Efficiency
	33,83	90,00	15	90,57%
37	-9,05	-10,00	-3	2,03%
39	-5,41	-6,00	-3	0,90%
27	-4,36	-8,00	-3	0,35%
9	0,00	0,00	0	0,00%
4	0,00	-1,50	0	0,00%
26	-5,33	-12,00	-4	0,62%
11	0,64	-7,50	0	-0,30%
14	1,09	0,00	-1	-0,41%
1	1,64	-3,50	0	-0,69%
30	-3,13	-8,00	-4	0,00%

(**c**) Profile 3

Figure 3. Best 15 solutions for the defined user profiles.

Regarding Profile 2, again the reference solution $s^* = s_9$ is the top ranked alternative followed by s_4. Then, solution s_{37} appears (which was not among the top 15 solutions for Profile 1). This solution has better efficiency and travel time than the reference solution, but is ranked in third place due to the interest (which is 10 units less than the reference). Solution s_{39} is also interesting because it is better in efficiency and travel time, while just having 6 units less of interest. In turn, the route uses 12 POIs instead of 15 which is a remarkable point.

Finally, for Profile 3, we observe that the reference solution is ranked in the fourth place. Solutions 37, 39, and 27 are top ranked, providing improvements in travel time

(shorter displacements between POIs) and greater efficiency. All of them had 12 POIs which nevertheless allowed to reach up to 88% of the reference interest (90 units).

So, Profiles 1 and 2 share the top two solutions. The third solution from Profile 2 does not even appear for Profile 1. None of the top 3 solutions from Profile 3 are available in Profile 1 and the third solution from Profile 3 does not appear in Profile 2.

These results clearly highlight the need of having a set of solutions to choose from instead of concentrating the efforts in obtaining one single best solution.

4. The Perishable Food Distribution Problem

Perishable food distribution has become increasingly tricky due to the features of these goods: perishables have a short shelf life, and their quality deteriorates over time during the distribution process until it is consumed. Therefore, ensuring food quality has become a major priority that must be met if customers are to be satisfied.

In a real-world setting, the distribution of perishable food entails a set of complex specifications that might be difficult to incorporate into mathematical models. Such parameters are frequently conflicting and, in some cases, unquantifiable [2,17,18].

A well-designed distribution plan that prioritizes short transit times and distances can help to preserve products quality while reducing waste. However, improving the sustainability of perishables distribution network necessitates balancing several competing goals, such as minimizing the travel costs (e.g., fuel consumption costs, refrigeration costs, the damage cost), maximize the average freshness, meeting consumers requirements to ensure their satisfaction (e.g., on-time delivery, good service level), and reducing environmental effect.

4.1. Problem Description

The problem is represented as a direct graph denoted by $G = (V, A)$, being $V = \{0, 1, \ldots, n+1\}$ the set of vertices, nodes 0 and $n+1$ represent the depot, and the remaining nodes $C = \{1, \ldots, n\}$ are the customers to serve. Each customer i should be served within a specific time slot defined by $[e_i, l_i]$.

The problem is modeled under the assumption of fleet homogeneity: all the refrigerated trucks have the same refrigeration characteristics, the same load capacity, and the same constant velocity.

When considering solutions for the perishable food distribution problem, several features can be defined.

4.1.1. Features

- Fixed, transportation and refrigeration costs;
- Damage cost;
- Average freshness;
- Quality of Service (or Service level);
- Total Tardiness.

From this set of available objectives, we deal with the following model formulation.

4.1.2. Parameters

- $C_{(i,j)}$: the transportation cost from node i to j;
- F: fixed cost associated to a vehicle;
- $T_{(i,j)}$: the travel time from i to j;
- $d_{(i,j)}$: the distance between node i and j;
- C_e: the cost per unit time for the refrigeration during the transportation process;
- U_i: the unloading time at customer i;
- C_e': the unit refrigeration cost during the unloading;
- S_i: the necessary time to serve customer i;
- q_i: the demand of customer i;
- Q: the loading capacity of trucks;

- K: the fleet of trucks;
- $[e_i, l_i]$: the time window of customer i.

4.1.3. Decision Variables

- $x^k_{(i,j)}$: a 0–1 decision variable, equal to 1 in case the truck k travels on arc (i, j), 0 otherwise;
- y^k_i: a 0–1 decision variable, equal to 1 in case the customer i is served by the truck k, 0 otherwise;
- t^k_i: a continuous decision variable representing the starting service time at customer i using the truck k.

4.1.4. Objective Function

The objective is to minimize the travel cost, calculated from: C_1, the fixed costs for using a vehicle denoted; C_2, the transportation costs and C_3 the refrigeration costs.

- **C_1, The fixed costs:** represent the vehicle's maintenance and depreciation costs.

$$C_1 = F \sum_{k \in K} y^k_0 \tag{6}$$

where k is the number of refrigerated trucks available and F represents the fixed cost related to a vehicle;

- **C_2, The transportation costs**: are proportional to the vehicle mileage. We consider only the fuel consumption cost and express it as:

$$C_2 = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} C_{(i,j)} x^k_{(i,j)} d_{(i,j)} \tag{7}$$

- **C_3, The refrigeration costs**: are calculated as

$$C_3 = C_{31} + C_{32} \tag{8}$$

where C_{31} are the refrigeration costs of the transportation process:

$$C_{31} = C_e \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x^k_{(i,j)} T_{(i,j)} \tag{9}$$

and C_{32} is the cost of energy supplied during the unloading:

$$C_{32} = C'_e \sum_{k \in K} \sum_{i \in V} y^k_i U_i \tag{10}$$

Considering the objectives described above, the problem can be formulated as a mixed integer program (MIP) giving by the following:

$$\text{Min } C_1 + C_2 + C_3 \tag{11}$$

subject to:

$$x^k_{(i,i)} = 0, \quad \forall i \in V, \quad \forall k \in K \tag{12}$$

$$x^k_{(0,n+1)} = 1, \quad \forall k \in K \tag{13}$$

$$\sum_{k \in K} y^k_i = 1, \quad \forall i \in C \tag{14}$$

$$\sum_{i \in V} x^k_{(i,j)} = y^k_j, \quad \forall j \in C, \quad \forall k \in K \tag{15}$$

$$\sum_{j \in V} x^k_{(i,j)} = y^k_i, \quad \forall i \in C, \quad \forall k \in K \tag{16}$$

$$\sum_{j \in v} x^k_{(0,j)} \leq 1, \quad \forall k \in K \tag{17}$$

$$\sum_{i \in v} x^k_{(i,n+1)} \leq 1, \quad \forall k \in K \tag{18}$$

$$\sum_{i \in C} D_i y^k_i \leq Q, \quad \forall k \in K \tag{19}$$

$$t_i + S_i + T_{(i,j)} x^k_{(i,j)} - M(1 - x^k_{(i,j)}) \leq t_j, \quad \forall i, j \in V, \quad \forall k \in K \tag{20}$$

$$e_i \leq t^k_i \leq l_i, \quad \forall i \in C, \quad \forall k \in K \tag{21}$$

$$\sum_{k \in K} y^k_0 \leq K \tag{22}$$

$$\sum_{j \in V} x^k_{(i,j)} \leq y^k_i, \quad \forall i \in V, \quad \forall k \in K \tag{23}$$

The objective (11) corresponds to the total cost minimization. Constraint (12) avoids going from a point to itself. Constraint (13) avoids going from the depot to node $n+1$ which represent the depot. Constraint (14) states that only one vehicle visits each customer exactly once. Constraint (15) and (16) ensure that each vehicle that arrives at a customer must depart for another destination. The vehicle departs from the depot 0, and returns to the depot node $n+1$ according to the Constraint (17) and (18). Constraint (19) guarantees the respect of vehicle loading capacity. The connection between the start service at a specific customer and the next one is established by Constraint (20). Constraint (21) is the time window constraint. Constraint (22) guarantees that the number of trucks departing from the depot do not exceed the available fleet of trucks K. Constraint (23) states that the vehicle must traverse the arc (i, j) in order to serve the customer j.

4.2. Problem Solving and Generation of Solutions

We describe next the basic details of the metaheuristic used, the dataset and the way the set of solutions is generated.

4.2.1. Metaheuristic

To solve the proposed problem, we use a basic implementation of a general variable neighborhood search GVNS metaheuristic. In GVNS, variable neighborhood descent (VND) is used for local search. When designing a GVNS, one should take the following decisions: the number of neighborhood structures, the order in which they will be explored, how the initial feasible solution is generated, the acceptance criterion and the stopping conditions. These elements define the configuration of the GVNS.

GVNS starts by an initialization phase in which a feasible solution is generated. The best solution is set as the first feasible solution. After selecting the neighborhood structures for the shaking stage N_s, and for local search N_k, the stopping condition is then chosen. The stopping condition corresponds to a number of iterations M that will be set in the computational experience. The shaking process, the local search, and the move decision are repeated until completing all neighborhood structures ($S = S_{max}$). In the shaking

phase a solution X' is generated randomly at the Sth neighborhood of $X^*(X' \in N_s(X^*))$. Then, the local search is performed to find a better solution X'' from X' using the N_K neighborhood structures.

4.2.2. Dataset

The proposed approach is evaluated through a dataset of 50 customers in Granada, Spain. The customers' time window takes a random value in the interval $[0, 240]$ with an interval width ≥ 45. For customer's target time, we assume that it is the midpoint of the corresponding time window. The dataset is available at http://18.156.111.23/PER-FOOD/DataPerishableFood.csv (accessed on 1 August 2021).

The refrigerated vehicles used to deliver products have a fixed cost of 25 €, and the fuel consumption is estimated to 3 €/km. The other parameters are: $P = 20$ €/unit, $\partial_1 = 0.002$, $\partial_2 = 0.003$, $C_e = 0.03$ €/unit, $C'_e = 0.04$ €/unit and $\delta = 0.8$.

4.2.3. Generation of Solutions

Although multiple runs of GVNS may allow to obtain a set of solutions, we illustrate here a different approach inspired in the so called 'Modeling to Generate Alternatives' (MGA) strategy [5,6]. MGA tries to generate a set of solutions that are quantifiable good across all the modeled goals, while remaining as different as possible from each other (in the decision space).

Let us assume that the best solution obtained for the problem is X^*, with $Z^* = F(X^*)$ as the objective value. To generate alternative solutions X that are maximally different from X^*, the following problem is addressed:

$$Min \ Jaccard(X, X^*) \tag{24}$$

Subject to:

$$X \in D \tag{25}$$

$$|F(X) - Z^*| \leq_f T \tag{26}$$

where $Jaccard(x_1, x_2)$ is the similarity coefficient used in the previous example. In the original MGA approach, the problem is stated as maximizing a dissimilarity function. The parameter T is the tolerance threshold related to the optimal objective value Z^* and should be defined by the decision maker.

We define the Jaccard's coefficient between two routing solutions as the ratio of the number of shared arcs to the number of total arcs used in both solutions. Let $y_{ijk} = 1$ if arc (i, j) from vertex i to vertex j is used by any vehicle in solution r_k, and $y_{ijk} = 0$ otherwise.

$$Jaccard(p, q) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{n} y_{ijp} \cdot y_{ijq}}{\sum_{i=0}^{n} \sum_{j=0}^{n} sign(y_{ijp} + y_{ijq})} \tag{27}$$

where $y_{ijp} \cdot y_{ijq} = 1$ iff arc (i, j) is used by both solutions, and $sign(y_{ijp} + y_{ijq}) = 1$ if any of the solutions use it. If solutions p and q are the same, the sum in the numerator will equal the sum in the denominator, and therefore $Jaccard(p, q) = 1$. On the other hand, if they are two completely different solutions with no arc in common, the numerator will equal 0, and then $Jaccard(p, q) = 0$

In second place, we need to manage the fuzzy constraint (26) (which is crisp in the original MGA approach). The membership function that represents the satisfaction degree of the fuzzy constraint is is defined as:

$$\mu(\tau) = \begin{cases} 1 & ; \tau < T \\ 1 - \frac{\tau - T}{\Delta^t} & ; T \leq \tau < T' \\ 0 & ; \tau \geq T' \end{cases} \tag{28}$$

where $T' = T + \Delta^t$ is the endurable distance threshold, $\tau = |F(X) - Z^*|$ is the cost difference between the reference solution and the obtained alternative.

Using the concept of $\alpha - cuts$ and the parametric approach [19], the fuzzy constraint is transformed into:

$$|F(X) - Z^*| \leq T + \Delta^t(1 - \alpha) \tag{29}$$

Now, the GVNS parameters, we use the following values: $\delta_1 = 0, \delta_2 = 1, \Lambda = 2, \mu = 1$. The stopping criteria M which correspond to the number of iterations, is fixed to 10. The other parameters are: $P = 20 \,€/\text{Unit}, \partial_1 = 0.002, \partial_2 = 0.003, C_e = 0.03 \,€/\text{unit}, C'_e = 0.04$ €/unit and $\delta = 0.8$.

To generate a set of alternative solutions, we start by finding a reference solution X^* by running 10 times the GVNS metaheuristic. The best solution obtained (according to Equation (12)) is considered as the reference solution X^*.

Then, using X^*, the new problem considering as objective the Equation (24) subject to the constraints (25) and (29) is solved for $\alpha \in [0, 0.2, 0.4, 0.6, 0.8, 1]$. For each value of α, a tailored version of GVNS is run 10 times.

At the end, 62 different solutions are obtained and the corresponding values for the following features are calculated: the damage cost, the average freshness, the service level, and the tardiness. The mathematical definition of these criteria can be found in Appendix A.

Table A2 (in the Appendix B) shows the results obtained for every value of α. It should be noted that several solutions per GVNS were obtained and just the different ones are reported.

4.3. Analysis of Solutions' Diversity

Table 2 summarizes the values of the features over the 62 solutions. It is clear that a wide variety of alternatives are available to choose from.

Table 2. Summary of values for every feature considered, over the generated 62 solutions.

	Min	Max	Mean	Std.Dev
Travel Cost	1167.85	1252.36	1215.43	24.97
Total damage	7530.24	8024.43	7709.90	109.80
Average freshness	20.12	22.39	21.11	0.51
Tardiness	69,137.18	216,830.61	134,649.45	36,362.44
Service level	15.96	20.31	18.10	1.07

As we explicitly generate the set of solutions minimizing the similarity with X^*, it is interesting to observe the relation between the Jaccard coefficient, and the differences in the calculated values for every solution. This information is shown in Figure 4 (for visualization purposes, just a subset of solutions are shown in every case). The Y axis reflects the difference as percentage.

When considering the distance (which is correlated with the travel cost), Figure 4a shows that all the solutions are worst in this feature. This is not surprising, as X^* has the lowest distance. There are several interesting solutions, like 10, 5, 6, and 9, that had a quite similar distance but a Jaccard similarity below 0.9. Solution 28 is also interesting because it has a low level of similarity (less than 0.45), while attaining a quite similar distance value (less than 2% of difference).

With an increment in the distance between 2% and 5%, up to eight solutions can be found with a Jaccard value between $[0.4, 0.62]$.

For the average freshness and service level, the situation is more interesting, because there are better and worse solutions than the reference one. Focusing in the average freshness (Figure 4b), those solutions with a positive value in Y are better than the reference solution. Solution 15 allows to improve up to 4% in the criteria while having a similarity value of 0.7. Other improvements between 1% and 2% can be obtained with quite different routes (solutions 47, 25, 52).

The results in terms of the service level are shown in Figure 4c. Here, solution 59 provides more than a 10% of improvement with a moderate value of similarity (less than 0.65). Several solutions providing improvements higher than 6% and similarity below 0.5 are available. In this case, even solution 37 may be interesting, as it has a minor decrement in service level but very low level of similarity.

Finally, Figure 4d shows the solution in terms of the Tardiness criteria. Here, all the solutions are better than the reference one. It is clear here that tardiness more similar than the reference solution can not be attained. Great improvements can be obtained, but with quite different solutions (look at the low values for similarity).

(**a**) Distance

(**b**) Average Freshness

(**c**) Service Level

(**d**) Tardiness
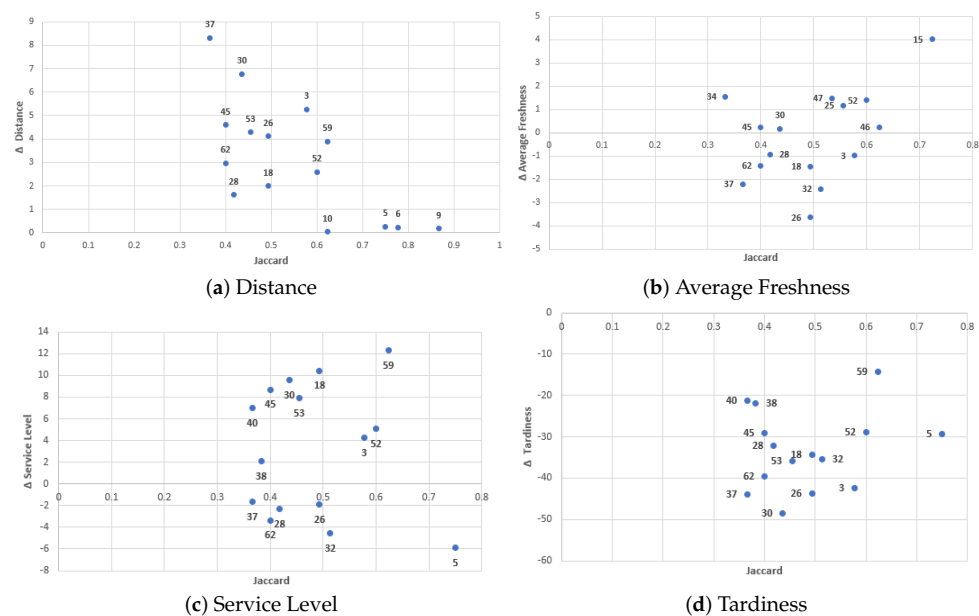
Figure 4. Variation of individual criterion and Jaccard's coefficient with respect to the reference solution. Just a subset of solutions is shown.

4.4. User Profiles and Ranking

The previous analysis is useful for detecting good solutions when analysed from just two points of view: similarity vs. a single criterion.

Here, we define three user's profiles, representing the preferences of a decision maker. As stated before, the preferences are indicated through a linear ordering of the criteria that is described below.

- Economic-centric (E-c): Travel Cost = Total Damage $\succeq_p$ Average freshness $\succeq_p$ Tardiness $\succeq_p$ Service level;
- Product-centric (P-c): Average freshness $\succeq_p$ Travel Cost $\succeq_p$ Total Damage $\succeq_p$ Tardiness $\succeq_p$ Service level;
- Customer Satisfaction-centric (C-c): Tardiness = Service level $\succeq_p$ Travel Cost $\succeq_p$ Total Damage $\succeq_p$ Average freshness.

where the symbol $\succeq_p$ should be read as "at least as preferred to".

For every profile, the procedure described in Section 2.1 to rank the solutions is applied. The best 15 solutions under each user profile are shown in Table 3–5.

Every solution is compared against the reference solution in terms of the following criteria: travel cost, total damage, average freshness, service level, and tardiness. Values in red cells are worse than the reference value, while the green ones are better.

Table 3. Ranking of solutions under the Economic−centric profile.

	Travel Cost	Total Damage	Average Freshness	Tardiness	Service Level
Reference Solution	1168.4	8024.42	21.52	134,158.6	17.45
18	1.72	−3.64	−1.46	−34.27	10.44
30	5.78	−3.29	0.17	−48.47	9.57
26	3.46	−4.90	−3.64	−43.70	−1.87
53	3.54	−4.78	−6.35	−35.80	7.88
3	4.50	−3.31	−0.99	−42.35	4.25
62	2.60	−4.22	−1.41	−39.59	−3.35
5	0.19	−5.60	−4.46	−29.38	−5.88
28	1.46	−3.61	−0.95	−32.17	−2.33
45	3.94	−3.00	0.24	−29.08	8.69
52	2.21	−2.07	1.39	−28.91	5.09
37	7.06	−4.11	−2.21	−44.00	−1.67
10	−0.04	−5.13	−6.21	−14.61	2.23
9	0.17	−3.18	−1.48	−17.28	2.45
6	0.14	−4.61	−4.48	−15.06	2.22
59	3.24	−4.77	−3.61	−14.21	12.28

Table 4. Ranking of solutions under the product-centric profile.

	Average Freshness	Travel Cost	Total Damage	Tardiness	Service Level
Reference Solution	21.52	1168.4	8024.42	134,158.60	17.45
30	0.17	5.78	−3.29	−48.47	9.57
52	1.39	2.21	−2.07	−28.91	5.09
3	−0.99	4.50	−3.31	−42.35	4.25
62	−1.41	2.60	−4.22	−39.59	−3.35
45	0.24	3.94	−3.00	−29.08	8.69
37	−2.21	7.06	−4.11	−44.00	−1.67
28	−0.95	1.46	−3.61	−32.17	−2.33
18	−1.46	1.72	−3.64	−34.27	10.44
26	−3.64	3.46	−4.90	−43.70	−1.87
47	1.47	6.07	−3.06	−18.42	4.74
32	−2.41	2.93	−2.79	−35.33	−4.58
15	4.01	2.54	−0.45	−1.24	9.68
34	1.53	6.61	−3.40	−12.16	4.29
25	1.16	5.57	−2.27	−14.17	7.58
46	0.23	3.58	−3.47	−15.53	3.11

Table 5. Ranking of solutions under the customer-centric profile.

	Tardiness	Service Level	Travel Cost	Total Damage	Average Freshness
Reference Solution	134,158.60	17.45	1168.4	8024.42	21.52
30	−48.47	9.57	5.78	−3.29	0.17
3	−42.35	4.25	4.50	−3.31	−0.99
26	−43.70	−1.87	3.46	−4.90	−3.64
18	−34.27	10.44	1.72	−3.64	−1.46
37	−44.00	−1.67	7.06	−4.11	−2.21
53	−35.80	7.88	3.54	−4.78	−6.35
62	−39.59	−3.35	2.60	−4.22	−1.41
45	−29.08	8.69	3.94	−3.00	0.24
52	−28.91	5.09	2.21	−2.07	1.39
32	−35.33	−4.58	2.93	−2.79	−2.41
28	−32.17	−2.33	1.46	−3.61	−0.95
5	−29.38	−5.88	0.19	−5.60	−4.46
40	−21.31	6.99	4.11	−4.01	−2.64
38	−21.90	2.07	3.57	−4.42	−3.11
59	−14.21	12.28	3.24	−4.77	−3.61

We can observe that the proposed approach allowed to generate a variety of alternative solutions, each of which shows a different behavior for each criterion. This diversity allows the decision maker (DM) to select the most appropriate solution, considering different perspectives and depending on his preference. Regarding the E-c profile, solution 18 is the top-ranked alternative. This solution is quite different from the reference solution X^* having a Jaccard coefficient $J(X^*, s_18) = 0.49$. s_{18} performs better than X^* in terms of service level, tardiness, and damage. On the other hand, has approximately the same travel cost as the reference solution. Solution 30 is slightly worse in terms of travel cost. However, it allows improvement in all the remaining criteria. Regarding the P-c profile, solution 30 is now the best one. This solution has a worse travel cost than the reference solution, but it is better in the remaining criteria. Solution 52 appears in the second place. Note that it was ranked in the 10th position for the E-c profile. Regarding the C-c profile, solution 30 retain the first place by improving significantly all the criteria, with a slight increase in travel cost compared to X^*.

It should be noted that solutions 26 and 30 are among the TOP-3 best alternatives for both the E-c and C-C profile. Another interesting alternative is solution 3 which maintains its place in the TOP-3 for the P-C and C-c profile. Solution 18 is the best alternative for a DM that have economic focus, and retain the third position in terms of service level improvement. However, this solution does not have interest if the focus is on the product freshness or customer satisfaction.

It is highly remarkable that the solution X^* does not appear among the top 15 solutions for all the profiles, which illustrate the ability of our proposed approach in generating performant and quite dissimilar solutions that will serve the decision maker to select the appropriate solution from his perspective.

5. Conclusions

The computational models of relevant optimization problems necessarily leave out of consideration several characteristics and features of the real world. So trying to obtain the optimum solution can not be enough for a problem solving point of view. Moreover, it is doubtful that a single solution would be able to meet all the real specifications. Therefore, it is desirable for the decision maker to have a *set of solutions* that are good enough for the modeled objective, but can perform also well when it comes to other unmodeled criteria.

In the context of a basic problem solving framework, we illustrated the role of metaheuristics as solutions' generators. In the tourist trip design problem, independent runs of an evolutionary algorithm allowed to obtain the set. In the case of perishable food distribution, a more complex strategy was used, inspired from the MGA approach.

Considering a set of solutions and different user profiles, allows to properly rank the available solutions. Using a similarity measure, such as the Jaccard's coefficient, also allowed to better understand the relation between the reference solution and the other ones, not only in the objectives space, but also in the decision space.

This opportunity to consider different perspectives will grant a decision better suited to the real context.

As future work, we will further explore MGA-like approaches, where some measurement of similarity between solutions is needed. Here, the concept of symmetry appears to be crucial as potentially symmetric solutions in the design space can be equal in the objectives space. In other words, different ways to attain the same results can be explored with a proper usage of the symmetry/anti-symmetry ideas.

Finally, connecting the solutions' analysis with multicriteria decision-making approaches will be also explored.

Institutional Review Board Statement: Not applicable.

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

Appendix A. Additional Quality Criteria for the Perishable Food Distribution Problem

The alternative solutions generated will be evaluated for a set of criteria relevant to perishable food distribution. It is expected that they can help the decision maker in selecting the most appropriate solution according to his or her preferences. In this work, we consider the four criteria described below:

Appendix A.1. Damage Cost

The quality of perishable foods degrades with the progression of time and temperature fluctuations during transportation and handling. Damage costs are incurred when product quality falls below a certain threshold. The following feature can be used to express the quality of refrigerated products: $D_t = D_0 e^{-\partial t}$, where D_t, and D_0 are, respectively, the quality of product at time t and from the depot 0. The parameter ∂ is the product's spoilage rate, which is assumed to increase as the temperature rises (Ref. [20]). As a result, we distinguish between damage cost during distribution C_{dmg}, and the damage cost during the unloading C'_{dmg} due to the temperature changes (∂ varies also). The damage cost C_{dmg} is expressed as follows:

$$C_{dmg} = \sum_{k \in K} \sum_{i \in C} y_i^k P q_i \left(1 - e^{-\partial_1 \left(t_i^k - t_0^k \right)} \right) \tag{A1}$$

where the coefficient ∂_1 reflects the spoilage rate of product when the vehicle is locked, t_i^k the arrival time of vehicle k at customer i, t_0^k the departure time of vehicle k from the depot, and y_i^k is 0–1 variable taking the value 1 in case the vehicle k is servicing the customer i, and 0 otherwise.

The damage cost during the unloading C_{42} is defined as:

$$C'_{dmg} = \sum_{k \in K} \sum_{i \in C} y_i^k P q_{in} \left(1 - e^{-\partial_2 S_i} \right) \tag{A2}$$

with q_{in} the remaining quantity of product after servicing the customer i, the necessary time to serve is customer i is S_i, and ∂_2 is the spoilage rate when the vehicle is opened.

Appendix A.2. Average Freshness

The average freshness, denoted by AVGF, is defined as [21]:

$$AVGF = \frac{\displaystyle\sum_{k \in K} \sum_{i \in C} y_i^k q_i e^{-\partial_1 \left(t_i^k - t_0^k \right)}}{\displaystyle\sum_{i \in C} q_i} \tag{A3}$$

Appendix A.3. Service Level

This criterion reflects the fact that customers may want to obtain their products at a certain time so that they can begin planning meals, for example. We use the target time as an indicator to determine the service level using the following function:

$$SL_i(t_i) = \begin{cases} 0 & t_i < e_i \\ f(t_i) & e_i \leq t_i < T_g^i \\ 1 & t_i = T_g^i \\ g(t_i) & T_g^i < t_i \leq l_i \\ 0 & t_i > l_i \end{cases} \tag{A4}$$

$SL_i(t_i)$ represents the service level ensured for customer i if we deliver his demand at time t_i. T_g^i is the target time, e_i and l_i are, respectively, the lower, and upper bounds of the time window. The function f is a non-decreasing function, while g is a decreasing function defined as follows:

$$f(t_i) = \frac{t_i - e_i}{T_g^i - e_i} \tag{A5}$$

$$g(t_i) = \frac{l_i - t_i}{l_i - T_g^i} \tag{A6}$$

A good alternative solution is the one that maximizes the following function:

$$SL = \sum_{i \in C} SL_i(t_i) \tag{A7}$$

Appendix A.4. Total Tardiness

We assume that we have a pre-defined set of customers' priorities and they must be served according to their priority level. To consider priority indexes, we define a precedence matrix P, where $P_{ij} = 1$ indicates that the customer i should be supplied before the customer j, and $P_{ij} = 0$ if customer i might be supplied after customer j.

A good solution is the one that minimize the tardiness as much as possible. Tardiness is increased when a lower-priority customer is served before a higher-priority customer, and can be expressed using the following formula:

$$\sum_{i \in C} \sum_{j \in C} t_j - t_i \tag{A8}$$

With t_j, t_i are the arrival times at customer j and i, respectively.

Appendix B. Sets of Solutions

The set of solutions for both problems, together with the calculated values for all the features, are provided here. Table A1 shows the 40 solutions for the time-dependant tourist trip design problem, and Table A2 shows the 62 solutions for the perishable food distribution problem grouped in terms of every α value

Table A1. Set of solutions for the dynamic tourist trip design problem. X^* is the reference solution.

Solution	Interest	# POIs	Efficiency	Travel Time
X^*	90.00	15	90.57%	33.83
1	86.50	15	89.88%	35.47
2	86.00	15	88.33%	41.60
3	88.00	15	88.40%	41.33
4	88.50	15	90.57%	33.83
5	90.00	15	87.61%	44.53
6	86.50	15	89.12%	39.08
7	81.00	15	87.03%	46.18
8	86.50	15	88.80%	39.10
9	90.00	15	90.57%	33.83
10	88.00	14	86.57%	48.08
11	82.50	15	90.28%	34.47
12	90.00	14	89.33%	38.22
13	74.50	14	87.54%	44.12
14	90.00	14	90.16%	34.92
15	84.50	15	84.77%	54.78
16	86.00	15	89.76%	36.52
17	86.00	15	89.24%	38.60
18	86.50	15	88.50%	40.30
19	86.00	15	86.42%	47.93
20	80.00	14	86.46%	46.98
21	82.00	13	88.43%	39.92
22	82.00	12	90.05%	32.03
23	84.00	12	89.83%	33.95
24	82.00	12	90.10%	33.50
25	84.00	13	89.58%	33.72
26	78.00	11	91.19%	28.50
27	82.00	12	90.92%	29.47
28	78.00	12	90.17%	31.08
29	86.00	14	88.32%	39.03
30	82.00	11	90.57%	30.70
31	74.00	12	86.71%	44.43
32	79.50	12	90.13%	31.75
33	80.00	11	90.53%	30.85
34	80.00	11	90.51%	30.92
35	78.00	11	88.28%	39.83
36	76.00	12	88.70%	35.68
37	80.00	12	92.60%	24.78
38	74.00	11	89.84%	33.93
39	84.00	12	91.48%	28.42
40	78.00	12	90.10%	35.15

Table A2. Set of solutions for the perishable food distribution problem in terms of every α value. X^* is the reference solution.

		Jaccard Coeff	Travel Cost	Total Damage	Average Freshness	Tardiness	Service Level
	X^*		**1168.40**	**8024.42**	**21.52**	**134,158.60**	**17.45**
	1	0.47	1168.35	8024.43	21.53	134158.60	17.45
	2	0.45	1218.24	7871.47	21.53	160,106.88	18.81
	3	0.58	1220.88	7759.02	21.31	77,340.90	18.19
	4	0.44	1211.40	7642.49	20.80	150,318.80	16.34
	5	0.75	1170.63	7574.83	20.57	94,747.56	16.42
	6	0.78	1170.01	7654.86	20.56	113,959.93	17.84
	7	0.45	1183.76	7593.93	20.96	172,670.60	19.66
	8	0.58	1184.17	7647.87	20.92	134,514.11	20.10
	9	0.87	1170.28	7769.21	21.21	110,974.37	17.88
	10	0.62	1167.85	7613.17	20.19	114,563.32	17.84
	11	0.70	1172.65	7665.73	20.62	129,750.23	17.06
$\alpha = 1$	12	0.62	1196.22	7989.30	21.64	129,427.45	18.24
	13	0.47	1242.50	7719.73	21.16	145,819.23	17.91
	14	0.44	1224.65	7530.24	20.12	179,372.30	17.95
	15	0.72	1198.04	7988.70	22.39	132,493.70	19.14
	16	0.51	1205.16	7774.37	21.70	163,477.79	20.08
	17	0.62	1236.82	7581.91	20.45	145,892.33	18.78
	18	0.49	1188.44	7732.66	21.21	88,179.91	19.27
	19	0.42	1235.29	7753.68	21.27	121,543.30	15.96
	20	0.56	1244.36	7745.72	21.81	189,232.09	17.86
	21	0.33	1239.06	7594.26	20.71	188,649.18	18.59
	22	0.26	1243.11	7713.25	21.52	149,117.63	18.95
	23	0.45	1228.98	7703.10	21.26	124,985.42	17.27
	24	0.62	1236.82	7581.91	20.45	161,864.14	18.78
$\alpha = 0.8$	25	0.56	1233.45	7842.50	21.78	115,142.75	18.77
	26	0.49	1208.76	7631.62	20.74	75,526.48	17.13
	27	0.56	1244.36	7745.72	21.81	216,830.61	15.80
	28	0.42	1185.46	7734.76	21.32	90,998.06	17.04
	29	0.53	1187.79	7670.50	20.70	115,110.98	18.45
	30	0.44	1235.87	7760.68	21.56	69,137.18	19.12
	31	0.42	1214.54	7647.11	20.82	155,266.90	16.94
$\alpha = 0.6$	32	0.51	1202.58	7800.86	21.01	86,761.52	16.65
	33	0.62	1236.82	7581.91	20.45	203,346.27	18.78
	34	0.33	1245.61	7751.62	21.86	117,838.20	18.20
	35	0.47	1200.82	7735.57	21.40	130,465.05	16.65
	36	0.38	1244.21	7543.49	20.62	113774.55	17.98
	37	0.37	1250.88	7694.87	21.05	75,133.26	17.16
	38	0.38	1210.05	7669.48	20.86	104,778.07	17.81
$\alpha = 0.4$	39	0.58	1240.43	7729.76	21.06	179,606.29	16.77
	40	0.37	1216.35	7702.60	20.96	105,564.80	18.67
	41	0.62	1235.22	7830.84	21.72	119,825.69	18.02
	42	0.47	1243.45	7605.22	21.20	175925.44	19.00
	43	0.44	1211.40	7642.49	20.80	141,176.06	16.34
	44	0.44	1224.65	7530.24	20.12	163,449.29	17.95
$\alpha = 0.2$	45	0.40	1214.36	7784.08	21.58	95,139.87	18.97
	46	0.62	1210.16	7746.02	21.58	113,320.22	17.99
	47	0.53	1239.28	7778.74	21.84	109,442.38	18.28
	48	0.58	1240.43	7729.76	21.06	179,606.29	16.77
	49	0.51	1205.16	7774.37	21.70	163,477.79	20.08
	50	0.49	1230.69	7570.80	20.39	176,408.72	18.37
	51	0.75	1170.63	7574.83	20.57	139,394.36	16.42
	52	0.60	1194.15	7858.23	21.83	95,379.98	18.34
	53	0.45	1209.71	7640.73	20.16	86,129.95	18.83
	54	0.53	1247.40	7721.07	21.20	187,736.49	17.34
$\alpha = 0$	55	0.56	1247.56	7696.99	21.08	182,320.04	18.63
	56	0.49	1252.36	7717.77	21.18	195,880.30	17.59
	57	0.47	1216.26	7955.55	21.68	126,305.39	17.12
	58	0.47	1198.73	7679.40	20.83	158,445.86	18.63
	59	0.62	1206.17	7641.84	20.75	115,100.21	19.60
	60	0.40	1208.93	7632.71	20.95	148,361.60	20.31
	61	0.40	1235.61	7747.33	21.33	125,947.14	19.89
	62	0.40	1198.73	7686.03	21.22	81,051.91	16.87

References

1. Gunalay, Y.; Yeomans, J.S. Simulation-optimization techniques for modelling to generate alternatives in waste management planning. *J. Appl. Oper. Res.* **2011**, *3*, 23–35.
2. Janssen, J.; Krol, M.; Schielen, R.; Hoekstra, A. The effect of modelling quantified expert knowledge and uncertainty information on model based decision making. *Environ. Sci. Policy* **2010**, *13*, 229–238. [CrossRef]
3. Walker, W.E.; Harremoës, P.; Rotmans, J.; Van Der Sluijs, J.P.; Van Asselt, M.B.; Janssen, P.; Krayer von Krauss, M.P. Defining uncertainty: A conceptual basis for uncertainty management in model-based decision support. *Integr. Assess.* **2003**, *4*, 5–17. [CrossRef]
4. Brill, E.D., Jr.; Chang, S.Y.; Hopkins, L.D. Modeling to generate alternatives: The HSJ approach and an illustration using a problem in land use planning. *Manag. Sci.* **1982**, *28*, 221–235. [CrossRef]
5. Loughlin, D.H.; Ranjithan, S.R.; Brill, E.D., Jr; Baugh, J.W., Jr. Genetic algorithm approaches for addressing unmodeled objectives in optimization problems. *Eng. Optim.* **2001**, *33*, 549–569. [CrossRef]
6. Zechman, E.M.; Ranjithan, S.R. An evolutionary algorithm to generate alternatives (EAGA) for engineering optimization problems. *Eng. Optim.* **2004**, *36*, 539–553. [CrossRef]
7. Gavalas, D.; Konstantopoulos, C.; Mastakas, K.; Pantziou, G. A survey on algorithmic approaches for solving tourist trip design problems. *J. Heuristics* **2014**, *20*, 291–328. [CrossRef]
8. Utama, D.M.; Dewi, S.K.; Wahid, A.; Santoso, I. The vehicle routing problem for perishable goods: A systematic review. *Cogent Eng.* **2020**, *7*, 1816148. [CrossRef]
9. Beliakov, G.; Pradera, A.; Calvo, T. *Aggregation Functions: A Guide for Practitioners*; Studies in Fuzziness and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2007. [CrossRef]
10. Torres, M.; Pelta, D.A.; Lamata, M.T.; Yager, R.R. An approach to identify solutions of interest from multi and many-objective optimization problems. *Neural Comput. Appl.* **2021**, *33*, 2471–2481. [CrossRef]
11. Liu, F.; Pan, L.H.; Liu, Z.L.; Peng, Y.N. On possibility-degree formulae for ranking interval numbers. *Soft Comput.* **2018**, *22*, 2557–2565. [CrossRef]
12. Schilde, M.; Doerner, K.F.; Hartl, R.F.; Kiechle, G. Metaheuristics for the bi-objective orienteering problem. *Swarm Intell.* **2009**, *3*, 179–201. [CrossRef]
13. Souffriau, W.; Vansteenwegen, P.; Vertommen, J.; Berghe, G.V.; Oudheusden, D.V. A personalized tourist trip design algorithm for mobile tourist guides. *Appl. Artif. Intell.* **2008**, *22*, 964–985. [CrossRef]
14. Deitch, R.; Ladany, S.P. The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm. *Eur. J. Oper. Res.* **2000**, *127*, 69–77. [CrossRef]
15. Garcia, A.; Vansteenwegen, P.; Arbelaitz, O.; Souffriau, W.; Linaza, M.T. Integrating public transportation in personalised electronic tourist guides. *Comput. Oper. Res.* **2013**, *40*, 758–774. [CrossRef]
16. Vansteenwegen, P.; Souffriau, W.; Berghe, G.V.; Oudheusden, D.V. The City Trip Planner: An expert system for tourists. *Expert Syst. Appl.* **2011**, *38*, 6540–6546. [CrossRef]
17. Brugnach, M.; Tagg, A.; Keil, F.; de Lange, W.J. Uncertainty matters: computer models at the science–policy interface. *Water Resour. Manag.* **2007**, *21*, 1075–1090. [CrossRef]
18. Matthies, M.; Giupponi, C.; Ostendorf, B. Environmental Decision Support Systems: Current Issues, Methods and Tools. *Environ. Model. Softw.* **2007**, *22*, 123–127.
19. Verdegay, J. Fuzzy Mathematical Programming. In *Fuzzy Information and Decision Processes*; Gupta, M.M., Sanchez, E., Eds.; North-Holland: New York, NY, USA, 1982.
20. Wang, S.; Tao, F.; Shi, Y.; Wen, H. Optimization of vehicle routing problem with time windows for cold chain logistics based on carbon tax. *Sustainability* **2017**, *9*, 694. [CrossRef]
21. Wang, X.; Wang, M.; Ruan, J.; Li, Y. Multi-objective optimization for delivering perishable products with mixed time windows. *Adv. Prod. Eng. Manag.* **2018**, *13*, 321–332. [CrossRef]