



Economic and Food Safety: Optimized Inspection Routes Generation

Telmo Barros¹(✉), Alexandra Oliveira^{1,2}() , Henrique Lopes Cardoso¹() ,
Luís Paulo Reis¹() , Cristina Caldeira³ , and João Pedro Machado³

¹ Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC),
Faculdade de Engenharia da Universidade do Porto,
Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
{up201405840,aao,hlc,lpreis}@fe.up.pt

² Escola Superior de Saúde do Instituto Politécnico do Porto (ESS-IPP),
Rua Dr. António Bernardino de Almeida, 400, 4200-072 Porto, Portugal

³ Autoridade de Segurança Alimentar e Económica (ASAE),
Rua Rodrigo da Fonseca, 73, 1269-274 Lisbon, Portugal
{accaldeira,jpmachado}@asae.pt

Abstract. Data-driven decision support systems rely on increasing amounts of information that needs to be converted into actionable knowledge in business intelligence processes. The latter have been applied to diverse business areas, including governmental organizations, where they can be used effectively. The Portuguese Food and Economic Safety Authority (ASAE) is one example of such organizations. Over its years of operation, a rich dataset has been collected which can be used to improve their activity regarding prevention in the areas of food safety and economic enforcement. ASAE needs to inspect Economic Operators all over the country, and the efficient and effective generation of optimized and flexible inspection routes is a major concern. The focus of this paper is, thus, the generation of optimized inspection routes, which can then be flexibly adapted towards their operational accomplishment. Each Economic Operator is assigned an inspection utility – an indication of the risk it poses to public health and food safety, to business practices and intellectual property as well as to security and environment. Optimal inspection routes are then generated typically by seeking to maximize the utility gained from inspecting the chosen Economic Operators. The need of incorporating constraints such as Economic Operators' opening hours and multiple departure/arrival spots has led to model the problem as a Multi-Depot Periodic Vehicle Routing Problem with Time Windows. Exact and meta-heuristic methods were implemented to solve the problem and the Genetic Algorithm showed a high performance with realistic solutions to be used by ASAE inspectors. The hybrid approach that combined the Genetic Algorithm with the Hill Climbing also showed to be a good manner of enhancing the solution quality.

Keywords: Planning · Scheduling · Optimization · Decision support · Vehicle routing problem

1 Introduction

Business Intelligence refers to the set of techniques and processes used by companies or organizations to convert their information into actionable knowledge which, when allied with Artificial Intelligence (AI) and Machine Learning (ML) techniques, can result in powerful decision support systems. This is the case of some governmental administrative institutions, which are currently optimizing their complex operational processes by exploiting useful knowledge extracted from a vast amount of data.

Some of these organizations have as mission the close supervision and constant assessment of a large amount and geographically sparse of other entities. In order to efficiently inspect the maximum possible entities an accurate and flexible route plan must be built taking into account several constraints inherent to the type of entities to be inspected in each operation.

In Portugal exists the Portuguese Food and Economic Safety Authority (ASAE), responsible for supervising and preventing non-compliance with the National and European legislation regarding the food and non-food sectors. Some of their duties lies in the report of detected hazards in the food chain, risk assessment and inspections to selected Economic Operators, either planned or unplanned. It is a reference entity in consumer protection, public health, safeguarding market rules and free competition by providing a public service of excellence.

ASAE's internal information system has a large dataset of portuguese Economic Operators and historic complaints. This information can be used to determine the inspection utility through the definition of risk matrices and the number and type of complaints per Economic Operator. Their data can also be used alongside external sources, using as much information as possible and enhancing, this way, the solutions and the selection of Economic Operators to be inspected.

Given an inspection utility function, the inspection routes must be generated such as to maximize the number of inspections to Economic Operators with the maximum utility. The routes that are part of the solution should be adaptable upon suggestions by the user, reflecting the nature of the work of the human inspectors. Previous research has shown that vehicle routing optimization can promote significant economic savings [10, 17, 26].

This problem was addressed in previous work [4] as a Multi-Depot Periodic Vehicle Routing Problem with Time Windows, and approached it using both exact and meta-heuristic methods. Although this was an operative solution, some details need to be added as the inclusion of distinct departure and arrival points, real opening hours of Economic Operators or dynamic inspection times. Hybrid approaches that combined the Genetic Algorithm with the Hill Climbing and Simulated Annealing were also implemented and put to the test.

The inclusion of these constrains turn the problem into a complex problem of generation and optimization of inspection routes to assess the set of selected Economic Operators. The developed models were then embedded in an Web Application that allows ASAE's collaborators to visualize geographically the Economic Operators, assign utilities and generate inspections routes.

The proposed methodologies to solve a problem that has existed for several years are an considerable advantage. Although the paper focus on the application to a specific food and economic safety context, the constraints can be easily applied in other situations without significantly changing the analyzed behavior of the algorithms.

The rest of the paper is structured as follows. Section 2, discusses the currently processes in use and data size. The formulation of the routing problem in hands and related work are given in Sect. 3. Section. 4 explains the main choices regarding the algorithms implementation and new additions from the previous version [4]. Recent results and algorithms execution comparison appear in Sect. 5. Section 6 shows the applicability of the methods by giving an overview of the developed web application. In Sect. 7 we present the conclusions and pointed some directions for future work.

2 ASAE's Data Dimension and Key Procedures

There exist more than 3,500,000 Economic Operators currently registered in ASAE's internal database. Every year, the number of customer complaints against Economic Operators, received via the complaint book or ASAE's website, surpass the 200,000. This number, which include the infractions implied by the complaints themselves and the type of targeted economic activity, will determine the utility of inspection. Consequently, complaint-targeted Economic Operators should be more likely to be included in inspection operations.

The information system in use relies on multiple data sources and platforms which, allied with the aforementioned numbers, make it difficult to automate some processes such as:

- i. the prioritization of Economic Operators that need to be supervised;
- ii. the assignment of these Economic Operators to brigades of inspectors; and
- iii. the determination of optimal inspection routes taking into account the minimization of travel distances or time.

The ASAE's organic structure covers mainland Portugal and is divided into three regional units. On a lower level there are a total of twelve operational units with a specific number of inspectors and vehicles at their disposal. The allocation and management of these resources occur at multiple levels but the ones addressed here are the:

Operational – which focuses on the generation of the brigades inspection plans, through the selection of specific Economic Operators to be inspected and the definition of starting and finishing points for each brigade; and the

Brigade – which is responsible for making minor changes in the inspection plan in real-time, due to unexpected reasons (such as the closure of an Economic Operator).

The inspections planning and the visualization of georeferenced information (Economic Operators) are two areas that can be improved. It is possible to

enhance the selection of Economic Operators that maximize the usefulness of being inspected in the inspections planning and also the generation of flexible routes for the inspectors while minimizing the spent resources.

3 Routing Problem

The three processes listed in Sect. 2 can be interpreted and solved as a single problem that falls into the family of Vehicle Routing Problems (VRP). This problem is part of the logistics operations in almost any supply chain. The approach to solve this complex combinatorial optimization problem lies in finding the set of routes with overall minimum route cost which service all the demands given: a fleet of vehicles with uniform capacity, a common depot, and several customer demands [22, 25]. One of the first problems modeled as a VRP was the *Truck Dispatching Problem* [14], which may be seen as a generalization of the *Traveling Salesman Problem* (TSP) [16, 19].

A VRP is usually modeled as a weighted graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{A}, \mathcal{C})$$

where: $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ is a set of vertices composed by the depot (vertex v_0) and various cities or customers to visit (vertex subset $\{v_1, \dots, v_n\}$); \mathcal{A} is the set of arcs joining these vertices. Each arc (v_i, v_j) , for $i \neq j$, has a non-negative distance c_{ij} associated with, and represented in the matrix \mathcal{C} . In some cases, these distances may be interpreted as travel costs or travel times. There is also a set \mathcal{K} of vehicles available at the depot. The problem's purpose is to determine the lowest cost vehicle route set, subject to the following restrictions [20]:

- i. each city (v_i) is visited exactly once by exactly one vehicle (k); and
- ii. all vehicle routes start and end at the depot (v_0).

A VRP also comprises four equations that model the objective function and its restrictions [8, 26]. Let x_{ijk} be a binary decision variable taking the value of 1 if vehicle k traverses the arc (v_i, v_j) , and 0 otherwise and c_{ij} defined as above, the minimum sum of costs of all the arcs is given by Eq. 1:

$$\min \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ijk} \tag{1}$$

With the constraint from Eq. 2 of a vertex being visited only once, with the exception of the depot [4]:

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1, \forall j > 0 \tag{2}$$

Equation 3 ensures that the number of the vehicles arriving at every customer and entering the depot is equal to the number of the vehicles leaving [4]:

$$\sum_{j=0}^n x_{ijk} = \sum_{j=0}^n x_{jik}, \forall i \geq 0, i \neq j, \forall k = 1, \dots, m \tag{3}$$

Equation 4 ensures that all vehicle depart from the depot [4]:

$$\sum_{j=1}^n \sum_{k=1}^m x_{0jk} = |K| \tag{4}$$

Even though the classical VRP may provide very useful solutions in some real life problems, the ASAE’s inspections routing problem have some constraints that are not satisfied by the definition above, namely:

- Inspections can only take place during opening hours of the Economic Operators;
- The inspections plans may have a multiple days duration;
- The departure and arrival points of each brigade vehicle can be two distinct points, since each operational unit has more than one parking spaces.
- An inspection plan is targeted to a specific economic activity or a set of economic activities, which directly affects the Economic Operators selection for inspection;
- Each operational unit is associated with a set of municipalities that make up the geographical boundary on which the its brigades can operate. This limit is well defined and must be respected without exception;

Many authors have already proposed several extensions and generalizations by adding or removing constraints leading to different route design process [6, 27]. The most relevant variations to the problem in hands are:

- i. the **Multi-Depot** (MDVRP) that considers more than one depot/place where the vehicles may start and arrive [18, 23];
- ii. the **Periodic** (PVRP) which assumes a multi-day planning with the capability to choose when to visit an Economic Operator [7]; and
- iii. the **Time Windows** (VRPTW) that takes into account a list of visiting schedules when each Economic Operator can be visited [9, 28].

Combining them leads to a Multi-Depot Periodic Vehicle Routing Problem with Time Windows (MDPVRPTW) [11]. In Fig. 1 is possible to see a solution of MDPVRPTW in a geographical representation. The solution has four complete inspection routes (set of green arrows) from three operational units (orange circles) and eleven (out of thirteen) Economic Operators inspected (blue circles).

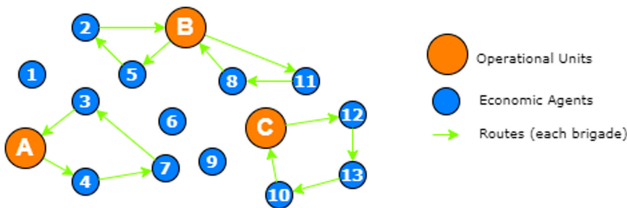


Fig. 1. MDPVRPTW representation [4]. (Color figure online)

Finally considering the problem in hands, $\mathcal{V} = \mathcal{EO} \cup \mathcal{D}$ is a set of nodes. composed of depots $d \in \mathcal{D}$ and Economic Operators $eo \in \mathcal{EO}$. $\mathcal{A} = \{(v_i, v_j)\}, i \neq j$ is the set of arcs that connect these nodes excluding the arcs that connect two depots and the matrix \mathcal{C} has the arcs travel times. There is also a set of brigades \mathcal{B} which will perform the inspections. Each brigade $b \in \mathcal{B}$ is a combination of a vehicle w (from a set \mathcal{W} of all available vehicles) and a set of inspectors I_b (a subset of \mathcal{I} of all inspectors) where $|I_b| \geq 2, \forall b \in \mathcal{B}$.

Each brigade $b \in \mathcal{B}$ has also an associated work shift $[SS_b, ES_b]$, where SS and ES are starting and ending of the work shift time, respectively, similar to Cordeau *et al.* [12] proposal.

Each Economic Operator eo has opening hours associated OH_{eo} . This is a set of time windows where they can be inspected. $OH_{eo} = \{[ot_j, ct_j]_{eo} : j = 1, \dots\}$ where ot_j and ct_j are the opening and closing times of consecutive working periods j in a day, respectively. Thus, if the arrival time at eo is not between any pair of $[ot_j, ct_j] \in OH_{eo}$, the brigade has to wait.

Every eo has also an expected time of inspection, represented by ti_{eo} , defined according to the type of their economic activity [4].

Additionally, another key factor that must be taken into consideration is the utility of inspecting a determined Economic Operator eo and can be defined as $u_{eo} \in [0, 1]$. This utility function is also known as the demand level at each node, as the importance of the node in the network, etc. [1, 2].

In this paper, the problem solving objectives are either the maximization of the global utility (defined in Eq. 5) or the maximization of the number of inspected Economic Operators in each day of inspection planning (defined in Eq. 6). The solution must satisfy the constraints for the classical VRP with Multi-Depot within the geographical limits of an operational unit, the time window of each Economic Operator and the brigade’s work shift:

$$max \sum_{b \in \mathcal{B}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{EO}} u_j x_{ijb} \tag{5}$$

$$max \sum_{b \in \mathcal{B}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{EO}} x_{ijb} \tag{6}$$

Although the study of such methods is not a recent topic, the discussion on how to achieve optimal solutions in less iterations/time is something that still remains today. The reason behind that is that the TSP, the VRP and its variant problems are NP-hard since they are not solved in polynomial time [9]. Due to their complexity, many heuristics and meta-heuristic approaches have been used to solve them as they can return solutions close to the optimum and present lower (and often adjustable) execution times, even for large data sets. Despite the problems’ difficulty, exact approaches have been developed like the branch and cut algorithm [21], but because they require a high computational power they are only feasible in small data sets. The difference between these two categories is related to the behaviour of algorithms when determining the solution [20]. To the best of our knowledge, a Tabu Search heuristic is the most explored technique to solve the Periodic Vehicle Routing Problem with Time Windows

(PVRPTW) and the Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW) separately [13].

4 Flexible Routes Generation

The methodologies implemented to solve the problem involved one exact method, three meta-heuristic methods and hybrid approaches that combined the Genetic Algorithm with two meta-heuristics to achieve the best results. The concept of inspection utility and its determination are presented in Sect. 4.1, while in Sect. 4.2 the representation of the solution is described. Then, the details of the implemented algorithms are included in Sects. 4.3 and 4.4.

4.1 Inspection Utility

The inspection utility function u_{eo} is the basis for determining optimal inspection routes. u_{eo} must be interpreted as the gain of inspecting an Economic Operator eo . The summed utilities of the inspected eo set is the component intended to be maximized when solving the routing problem. The definition of u_{eo} appears in Eq. 7 as a weighted sum of n functions $f(eo)$ defined based on some problem context criteria and taking into consideration the eo .

$$u_{eo} = \sum_{i=1}^n w_i f_i(eo) \tag{7}$$

So, for a given eo , u_{eo} retrieve a value in the interval $[0, 1]$ corresponding to the gain of inspecting that eo . This gain would be higher to eo operating in certain economic activities that pose higher risk in any of the scope inspected by ASAE. Highlighted criteria taken into account include the following eo attributes:

- i. the number and severity of **pending complaints**;
- ii. previous inspections results (its **inspections historic**); and
- iii. the operator's **economic activity**.

Focusing only on the first criteria, each eo can have a number of pending complaints received by ASAE and that have not yet been investigated or inspected. These pending complaints have associated an estimated infraction severity based on the complaint's description. This evaluation is currently being inserted manually by a collaborator but progresses have been made in the areas of automatic complaint analysis and classification of the infraction severity [3, 15]. The three severity types are shown in Table 1, and their respective weights are used to determine the severity score S assigned to each eo , represented in Eq. 8, where SC is the set of severity classes, NC the number of complaints and SW the severity weight.

$$S_{eo} = \sum_{sc \in SC} NC_{sc, eo} SW_{sc} \tag{8}$$

Table 1. Infraction severity classes and their weights.

Infraction severity - <i>sc</i>	Weight - <i>sw</i>
Crime	4
Administrative infringement	2
Other	1

The determination of this first criterion’s function was done by allaying the empirical knowledge and experience from ASAE members with the analysis of the distribution of the severity scores per Economic Operator, graphically visible in Fig. 2 (the vertical axis is in logarithmic scale). In the figure there are only represented 99% of the targeted Economic Operators which corresponds to a maximum severity score of 32. The green bars show the absolute number of Economic Operators with a given Severity Score and the orange line show the cumulative percentage.

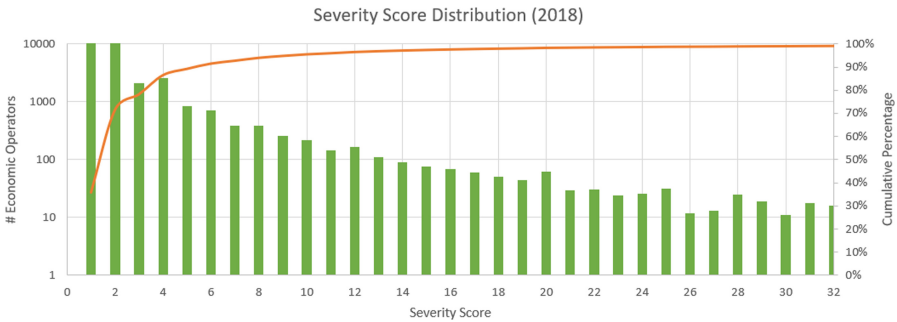


Fig. 2. Severity Scores distribution for received complaints in the year 2018. (Color figure online)

Equation 9 is an exponential function to define a higher rate of change for smaller *SS*. The rate of growth was defined with the ASAE field experts.

$$f_1(eo) = -1.3^{-S_{eo}} + 1 \tag{9}$$

4.2 Representation

The implemented methodologies described in the following sections are able to solve the optimization problem in hands but require, among other data structures, a form of representation to hold the solution routes (inspection plan). With regards to the inspection plan representation for $|\mathcal{B}|$ brigades, a set of $|\mathcal{B}| + 1$ lists is used. The first $|\mathcal{B}|$ lists include the actual inspection routes, while the last list contains a set of Economic Operators that will not be inspected in this plan. This form of representation allows us to deal with the periodic nature

of the VRP, by planning daily and not requiring all Economic Operators to be visited in one day. The Economic Operators are stored by inspection order in the $|\mathcal{B}|$ generated routes. Figure 3 shows an inspection plan geographically representation in detail by the example in Fig. 1, with four routes/brigades and thirteen Economic Operators.

This representation is used by all with the exception of the genetic algorithm, which, due to the crossover and mutation operations, requires another form of representation. This adaptation is achieved by converting the multidimensional list into a linearized single-dimensional one. For the specific representation illustrated in Fig. 3, the individual in the genetic algorithm would be [4, 7, 3, 5, 2, 11, 8, 12, 13, 10, 1, 6, 9]. To decode the individual back into the routes, each *eo* is inserted sequentially in the routes until no more *eo* fit the brigade’s work shift duration [28].

Representation		
[4,7,3], [5,2], [11,8], [12,13,10], [1,6,9]		
Brigades		Uninspected <i>eo</i>
4 → 7 → 3	5 → 2	1, 6, 9
11 → 8	12 → 13 → 10	

Fig. 3. Inspection plan representation and its meaning.

The information about the departing and arriving depots does not appear in this inspection plan representation because it is kept in the brigades data structure alongside with other information such as work shift duration and departure time.

4.3 Exact Approach

Exact methods are capable of determining the optimal solution but are well known to require more computational effort than heuristic approaches. For the problem in hands there are some real case scenarios where route generation is targeting a small set of pre-filtered Economic Operators and brigades and which allow exact methods to be employed.

The Branch and Bound algorithm is an exact methodology capable of solving optimization problems as the Vehicle Routing Problem. For this specific problem, a branching strategy that allows to traverse the whole solution space consists of adding each possible Economic Operator to each brigade at a time. The bounding occurs when the generated solution disrespects the work shift duration of any brigade.

The application of the method to the specific problem is shown in Fig. 4. The example includes three Economic Operators [1, 2, 3] with utilities of 0.05, 0.40 and 0.10, respectively, and two brigades. In the first node, every economic agent is in the set of agents not inspected by any brigade, so the utility of the

solution is 0. At each layer, an agent is removed from the last array and inserted into one of the brigades.

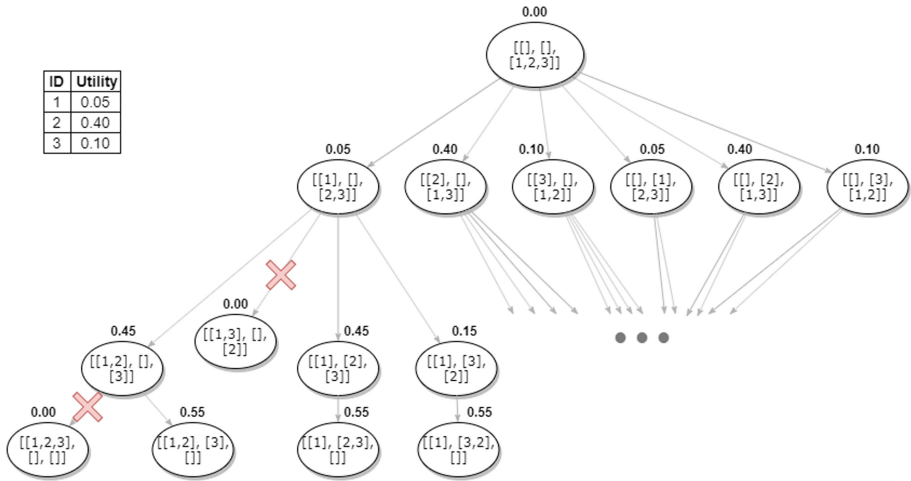


Fig. 4. Branch and Bound applied to the routing problem [4].

4.4 Meta-heuristic Approaches

Regarding meta-heuristic approaches, three different methods were developed:

- i. **Hill Climbing,**
- ii. **Simulated Annealing** and
- iii. **Genetic Algorithm.**

The main details are described in this section and these approaches allowed to obtain a sufficiently good solution in a short execution time, when compared to the Branch and Bound algorithm, for instance.

All the three meta-heuristic implementations require at least one initial solution to start. The initial solution is always valid, although it may not have a high utility. It is generated by randomly allocating Economic Operators to each brigade, ensuring that travel times and inspection durations fit in the brigade’s work shift. The remaining Economic Operators are kept in the last list, as shown in Fig. 3.

From this solution, Hill Climbing and Simulated Annealing approaches proceed to find the optimal solution out of all possible solutions (the search space). The neighborhood function for the two methodologies is based on two equiprobable operators (Fig. 5):

- *Swapping* consists of exchanging the order of two Economic Operators; swapping can occur between two operators of the same brigade, from different brigades or from one brigade and the list of uninspected Economic Operators.
- *Repositioning* consists of changing the position of an Economic Operator; just as with swapping, repositioning can also occur within the same brigade, to a different brigade or to the list of uninspected Economic Operators (removing it from being visited).

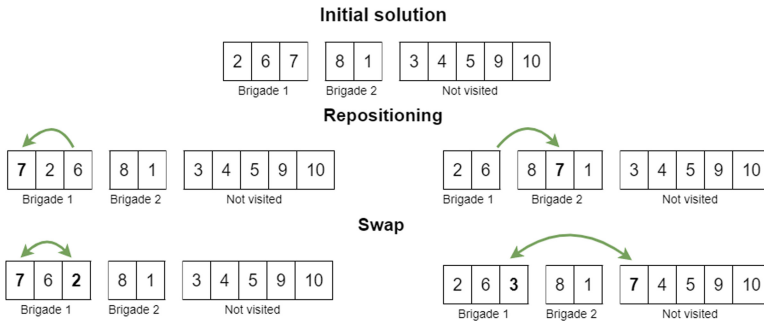


Fig. 5. Neighborhood generation operations.

The implementation of Simulated Annealing has the specificity of probabilistically accepting solutions with lower utility than the current solution. Equation 10 shows the temperature update function determined by the current iteration and total number of steps. Equation 11 shows the acceptance probability formula in this specific application of the method.

$$temperature = 1 - \left(\frac{iteration}{n \text{ of steps}} \right)^{\frac{1}{100}} \tag{10}$$

$$p = e^{\frac{new \ utility - best \ utility}{temperature}} \tag{11}$$

The probability of accepting solutions with lower utility is thus related with the differences in utility and the number of iterations already performed (which directly determines the temperature).

The Genetic Algorithm implementation followed the selection process suggested by Zhu [28] – tournament selection. The implication in this selection scheme is to give priority in mating to genetically superior chromosomes, while allowing less good entities to be selected.

Two crossover processes were implemented: PMX and heuristic crossover. The application of general-purpose crossover operations unavoidably produces invalid offspring that have duplicated genes in one string.

With Heuristic crossover, an arbitrary cut is made on both chromosomes and one of the genes immediately following the cut position is chosen to be maintained. The not selected gene is replaced or deleted to avoid duplication in the following iterations. After this decision, an iterative process begins that traverses the chromosomes and compares, in each iteration i , the distance between the gene at position i and position $i + 1$ of both progenitor chromosomes. The $i + 1$ position gene presenting lower distance to the previous gene is selected to be part of the descending chromosome. The gene that was not selected is either deleted or swapped with the selected gene. Figure 6 portrays the process with two chromosomes of size 5. c_{ij} is the distance from gene i to gene j . Two offsprings are generated, one from repositioning and the other from deletion.

Initial		Assumptions		
Parent 1	1 3 2 4 5	Parent 2	4 5 3 2 1	Cut point between 2nd and 3rd genes
Repositioning		Deletion		
Iteration 1				
Parent 1	1 3 2 4 5	Parent 1	1 3 2 4 5	Randomly selected eo: 2
Parent 2	4 5 3 2 1	Parent 2	4 5 3 2 1	
Offspring 1	2	Offspring 2	2	
Iteration 2		Iteration 2		
Parent 1	1 3 2 4 5	Parent 1	1 3 2 4 5	$c_{2,3} < c_{2,4}$ $c_{2,1} < c_{2,4}$
Parent 2	4 5 2 3 1	Parent 2	4 5 3 1	
Offspring 1	2 3	Offspring 2	2 1	
Iteration 3		Iteration 3		
Parent 1	1 4 2 3 5	Parent 1	3 2 4 5	$c_{3,1} < c_{3,5}$ $c_{1,5} < c_{1,4}$
Parent 2	4 5 2 3 1	Parent 2	4 5 3 1	
Offspring 1	2 3 1	Offspring 2	2 1 5	
...		...		

Fig. 6. Heuristic crossover example (3 iterations).

PMX is a slightly different and not so complex process as it does not take into account the distance between the Economic Operators. It starts by selecting two arbitrary points, then swapping the portions of the chromosomes between these points. In order to remove duplicate Economic Operators, a series of gene swappings are performed within each chromosome. This crossover operation originates two descending chromosomes [28].

Following the crossover process, there is a low probability of the descending chromosomes being mutated. The genetic approach encompasses four types of mutations, equally likely to occur:

- *Gene Repositioning*: a gene (Economic Operator) is randomly selected and repositioned at a random position inside the chromosome.

- *Gene swap*: two genes (Economic Operators) are randomly selected and their positions are swapped.
- *Sequence Repositioning*: a set of randomly selected consecutive genes is repositioned at a random position inside the chromosome (keeping their original sequence).
- *Sequence Swap*: a set of randomly selected consecutive genes swap their position with a gene (keeping their original sequence).

5 Results

The evaluation of the implemented methods was performed under a simulated environment using multiple tests and experiments and allowed to collect relevant information regarding the performance and solution quality of the four algorithms and hybrid approaches.

Test were performed using a set of 500 randomly selected Economic Operators from northern Portugal. The test set geographic distribution is displayed in Fig. 7. In the figure are also visible three depots, that work as points of departure and arrival of the brigades.

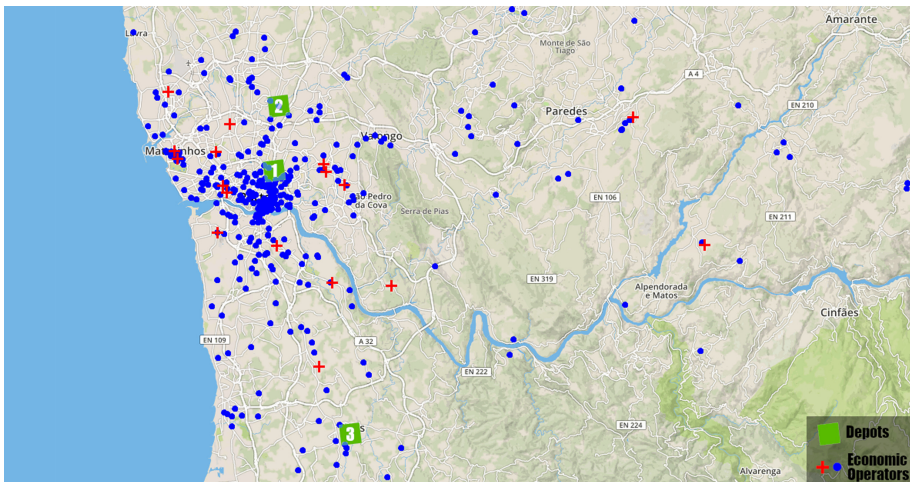


Fig. 7. Test set geographic distribution. (Color figure online)

From the 500 Economic Operators, a smaller subset of 20 operators (red crosses on the map) was randomly selected to allow the comparison of the meta-heuristic solutions with the Branch and Bound method. As such, two test cases were setup.

Some considerations about every algorithm execution:

- Brigades always start their route at the same hour and their work shift have the same duration of 8 consecutive hours.

- Economic Operators real utility or opening hours were not used; they were assigned a random utility instead and are assumed to be always open.
- Inspection time was fixed on 1 h for each Economic Operator.
- The values shown for meta-heuristic methods are always the average of 10 executions.
- For the Genetic Algorithm, executions were performed using a population of 100 individuals, with crossover probability, mutation probability and recovery percentage of 85%, 5% and 4% respectively.

Varying parameters were:

- *Objective Function*: Maximize the total utility of inspected agents, or maximize the number of Economic Operators to be inspected.
- *Brigades*: Number of brigades and departure/arrival points.
- *Stopping Criterion*: Maximum number of iterations or maximum execution time.

Varying the objective function and the number of brigades allowed us to analyze the direct effect of these variables in the algorithms performance.

The results for the referred subsets are shown in Tables 2 and 3. The test set results from Table 2 are related to the execution of the four approaches applied to the subset of 20 Economic Operators and 1 brigade that starts and finishes its shift in Depot 1. Table 3 presents the results for the algorithms execution given the full test set of 500 Economic Operators and 27 brigades. The brigades set is evenly distributed by the depots, that is, there are 3 brigades assigned to each pair of departure and arrival depot.

Each results table is split into four sections corresponding to different test cases with the number of iterations and execution times used as stop criteria. The results for the two objective functions appear in two columns: maximization of the global inspection utility ($max u_{eo}$) and maximization of the number of inspected Economic Operators ($max inspected eo$). Columns u_{eo} and eo contain the average of obtained inspection utilities and number of inspected Economic Operators, respectively. The total number of iterations executed by the methods appear in column i , while column $best i$ shows the iteration where the best solution was found. The average execution times are displayed in column t .

Each row stands for one algorithm execution: Branch and Bound (BB), Hill Climbing (HC), Simulated Annealing (SA), Genetic Algorithm with PMX crossover ($GA(P)$) and Heuristic crossover ($GA(H)$). The Genetic Algorithm was also combined with Hill Climbing and Simulated Annealing into a hybrid approach. Using the solution obtained from the Genetic Algorithm, the two meta-heuristics are applied (both executed with a stopping criterion of 100 iterations without evolution) to verify if it significantly increase the previous solution utility.

Table 2. Results for 20 eos and 1 brigade.

Algo.	max u_{eo}				max inspected eo			
	u_{eo}	best i	i	t	eo	best i	i	t
BB	5,6181	–	–	–	7,0	–	–	–
Max. i: 10000								
HC	5,6181	574	10000	0,3922	6,4	2882	10000	0,3892
SA	5,6181	624	10000	0,3983	6,2	1258	10000	0,3986
GA(P)	5,6181	7	100	1,1358	7,0	21	100	1,1557
GA(H)	5,4826	64	100	1,6443	7,0	2	100	1,6350
Max. i: 100000								
HC	5,6181	634	100000	3,8581	7,0	16283	100000	3,9408
SA	5,6181	583	100000	3,9795	7,0	19628	100000	4,0257
GA(P)	5,6181	13	1000	11,3204	7,0	43	1000	11,6694
GA(H)	5,6181	255	1000	16,6012	7,0	4	1000	16,2462
Max. t: 1s								
HC	5,6181	626	25343	–	6,8	4982	25170	–
SA	5,6181	423	24727	–	6,9	10626	24414	–
GA(P)	5,6012	11	89	–	7,0	30	86	–
GA(P)+HC	5,6181	98	198	–	7,0	30	186	–
GA(P)+SA	5,6181	90	189	–	7,0	30	186	–
GA(H)	5,5037	32	61	–	7,0	2	61	–
GA(H)+HC	5,5097	75	175	–	7,0	2	161	–
GA(H)+SA	5,5093	71	171	–	7,0	2	161	–
Max. t: 2s								
HC	5,6181	390	51267	–	6,8	10075	50451	–
SA	5,6181	652	48941	–	6,9	10344	47745	–
GA(P)	5,6181	21	176	–	7,0	31	174	–
GA(P)+HC	5,6181	21	276	–	7,0	31	274	–
GA(P)+SA	5,6181	21	276	–	7,0	31	274	–
GA(H)	5,6091	49	123	–	7,0	2	123	–
GA(H)+HC	5,6121	132	232	–	7,0	2	223	–
GA(H)+SA	5,6091	125	225	–	7,0	2	223	–

Through the global analysis of the obtained results, it is possible to conclude that they are satisfactory with multiple approaches achieving close solutions with approximately equal utilities as the exact method. The results also led to conclude that, when a lower number of iterations was given, the HC and SA could not achieve the optimal solution for the *max inspected eo* and the GA(H) for the *max u_{eo}* . By increasing the iterations limit, all of the metaheuristic approaches were capable of determining a solution with equal utility as the solution found by the Branch and Bound method.

When the algorithms were tested under the same execution time limit, HC and SA performed well in the maximization of the inspection utility. They also enhanced the solution quality obtained by the GA. For the maximization of the number of inspected Economic Operators, GA(P) and GA(H) were able to

consistently find optimal solutions and the fastest one was the heuristic crossover with an average of 2 iterations. Considering the total number of steps and the time to execute them the HC and the SA methods were the fastest to execute each iteration with very similar times, followed by the GA(P) and the GA(H).

Table 3. Results for 500 eos and 27 brigades.

Algo.	max u_{eo}				max inspected eo			
	u_{eo}	best i	i	t	eo	best i	i	t
Max. i: 10000								
HC	98,7232	9881	10000	5,5247	149,4	9119	10000	6,0015
SA	97,6994	9903	10000	5,5530	148,4	8990	10000	5,9492
GA(P)	94,9479	84	100	21,4769	151,5	49	100	22,8349
GA(P)+HC	95,9764	386	486	21,7019	151,5	49	200	22,8958
GA(P)+SA	95,6239	251	351	21,6159	151,5	49	200	22,8978
GA(H)	96,7180	69	100	84,3511	178,3	62	100	89,6848
GA(H)+HC	96,9691	219	319	84,4806	178,3	62	200	89,7489
GA(H)+SA	97,3835	235	335	84,4901	178,3	62	200	89,7475
Max. i: 100000								
HC	129,5417	98925	100000	55,0777	169,5	82586	100000	59,0404
SA	130,2098	98684	100000	54,6503	168,4	85063	100000	59,0773
GA(P)	106,4061	952	1000	216,7616	154,7	742	1000	228,8047
GA(P)+HC	106,6261	1142	1242	216,9005	154,7	742	1100	228,8688
GA(P)+SA	106,6185	1109	1209	216,8819	154,8	1005	1105	228,8688
GA(H)	102,4479	862	1000	877,2816	181,3	339	1000	845,2185
GA(H)+HC	102,9940	1178	1278	877,4526	181,3	339	1100	845,2795
GA(H)+SA	102,7018	1101	1201	877,4051	181,3	339	1100	845,2810
Max. t: 1s								
HC	72,0507	1706	1755	—	128,7	1445	1647	—
SA	71,6273	1698	1750	—	127,2	1397	1653	—
GA(P)	71,9183	4	5	—	135,7	2	4	—
GA(P)+HC	73,9384	354	454	—	136,1	24	124	—
GA(P)+SA	73,4790	235	335	—	135,9	10	110	—
GA(H)	83,0392	0	1	—	157,7	0	1	—
GA(H)+HC	83,9257	206	306	—	157,7	1	101	—
GA(H)+SA	83,6991	134	234	—	157,7	1	101	—
Max. t: 2s								
HC	80,9007	3493	3578	—	135,3	3163	3373	—
SA	82,6057	3487	3580	—	134,0	3105	3348	—
GA(P)	76,8735	8	10	—	140,4	8	9	—
GA(P)+HC	78,1521	229	329	—	140,5	12	112	—
GA(P)+SA	78,0103	200	300	—	140,5	17	117	—
GA(H)	86,1844	2	3	—	164,2	1	3	—
GA(H)+HC	86,8208	197	297	—	164,2	3	103	—
GA(H)+SA	87,4639	209	309	—	164,2	3	103	—

As expected and against to what happened in the first experiment, the methods utilities solutions obtained for the larger dataset were more scattered.

Table 3 shows that regarding inspection utility maximization, all methods were still improving their solution quality when the stopping criterion was met. HC and SA presented similar utilities and were able to always slightly improve the solution obtained by the GA. The solutions obtained by both crossovers in GA were very similar, but the Heuristic one can generally find a better solution in a earlier iteration than the PMX crossover.

Regarding the maximization of inspected Economic Operators, results followed almost the same pattern from Table 2. In general, GA(P) and GA(H) were able to find solutions with more Economic Operators inspected and almost never taking advantage of the HC/SA complementary layer. Heuristic crossover was again able to find a solution with higher utility than the other methods.

The time to perform each iteration kept the same distribution from the first test case, with HC and the SA methods executing faster, followed by GA(P) and GA(H).

6 Applicability

A crucial factor for being successful in solving sophisticated vehicle routing problems (VRP) is to offer reliable and flexible solutions [24]. Taking this into account, a Web application was designed [4,5]. It is the entry point to the visualization and interaction with the routing algorithms. The application has two main views or pages:

- i. the *Inspection Planning Form*, shown in Fig. 8, which can be used to customize route generation, and
- ii. the *Inspection Planning Overview*, shown in Fig. 9, where it is possible to see the solution and its details, also enabling editing and storing operations.

The *Inspection Planning Form* allows to input the custom conditions for a new inspection plan generation. The input variables are directly correlated with the ones specified in Sect. 3 and are the following:

- *Objective Function*: It is either the maximization of u_{eo} or the maximization of agents to be inspected;
- *Starting Date*: Starting date and time for the inspections route;
- *Meal Break*: Time interval where a 1 h break may occur;
- *Economic Activities*: List of economic activities to include in the solution (may be empty to consider all registered Economic Operators);
- *Brigade*: List of brigades composed by:
 - *Vehicle*: Vehicle to be used;
 - *Duration*: Maximum amount of time to be spent on the field performing the inspections;
 - *Inspectors*: List of inspectors (minimum of 2 per brigade);
 - *Starting/Finishing Points*: Geographic location of the starting and finishing points (the operational unit headquarter is used by default).

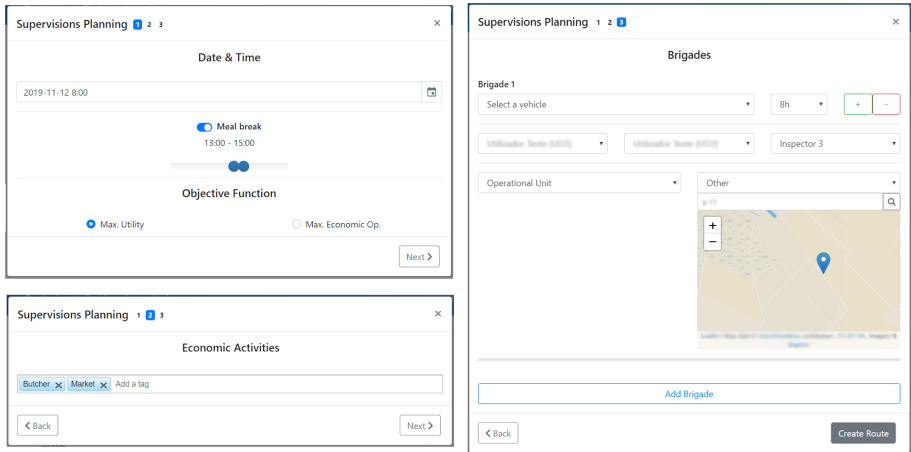


Fig. 8. Input conditions page (multiple views).

The *Inspection Planning Overview*, portrayed in Fig. 9, presents the solutions found from a geographical and chronological point of view. The routes are outlined in the map and can be selected or toggled from the left sidebar. This sidebar also contains main information about each brigade and global details of the solution. It is also possible to generate a new solution by clicking on the button “New Route”. In the bottom part there is a chronological view of the events of the selected brigade on left sidebar.

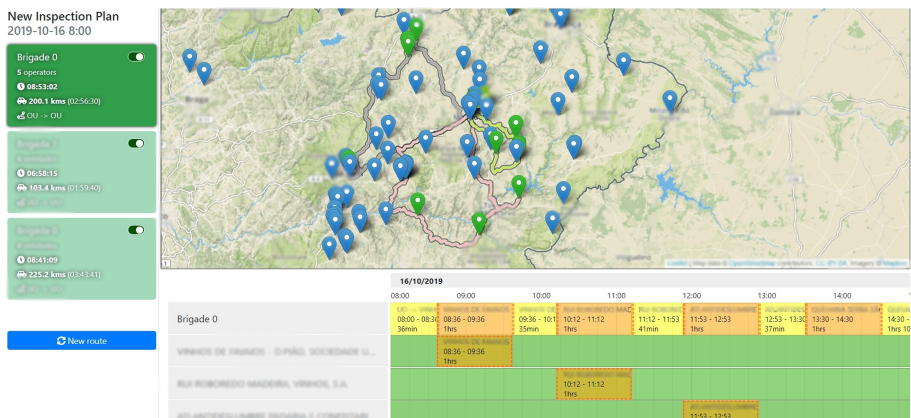


Fig. 9. Web application overview.

The developed system can work as a decision support system for inspectors by creating a daily plan for one or more brigades of inspectors and promote, this

way, a maximum number of inspections in a day or the maximization of the utility of inspected agents with an efficient use of resources. The inspection plan can be consulted by any worker and the routes are manageable by workers who suggest the additions/removals of Economic Operators. The inserted suggestions affect directly the defined utility of each Economic Operator and force the previous solution to be recalculated and presented.

7 Conclusions and Future Work

In this paper, we presented the modeling and resolution of a complex combinatorial optimization problem related with the inspections planning process of ASAE. The generation and optimization of routes is not a novel problem and the solutions presented always aim at determining the most effective route for a fleet of vehicles to visit a set of points. The constraints of the problem at hands led to classify it as an MDPVRPTW. To solve it, four base strategies were implemented: one exact, three meta-heuristic methodologies and hybrid approaches combining Genetic Algorithm with the Hill Climbing and Simulated Annealing.

Another key factor was the determination of an utility function capable of determining the inspection utility of an Economic Operator based on the number and severity of pending complaints, inspections historic and economic activity area among others.

This work is integrated in an interactive platform capable of generating flexible inspection routes targeted to support the decision of the ASAE's personnel and enhance current processes of Economic Operators selection for inspection and inspections routing. It offers functionalities that facilitate the whole process of selecting and assigning Economic Operators to the multiple brigades, respects the main constraints from the original problem and allows platform users to customize the desired inspection plan before and after the solution has been generated, through the addition and removal of Economic Operators to be inspected.

Regarding the methodologies performance, despite the fact that Heuristic crossover (in the Genetic Algorithm approach) has a higher execution time per iteration than the PMX option it was able to consistently find better solutions specially with the larger dataset and the maximization of the inspected *eo*. This phenomenon can be explained because it takes into consideration the minimization of distances when performing the crossover. For the maximization of $u_{e,o}$ Hill Climbing and Simulated Annealing were able to achieve higher utility solutions in the smaller dataset case and with a bigger execution time/number of iterations. With time limited executions in the larger dataset, Genetic Algorithm with Heuristic crossover was faster to determine greater solutions.

The development prospects for a project of this nature and scope are varied. There are at least one feature missing and more testing is necessary, specially on the field:

- Inclusion of break-time periods for inspectors and supervisors during the inspection route. The fact that this period is flexible, relative to each brigade

and can be different according to the start time of the inspection actions makes it difficult to add to the methods currently implemented, but should be taken into account in a next version of the system since it may compromise the use of the routes in a real context.

- Test the whole system in the field with ASAE’s collaboration, to extract metrics from algorithms in a real context and detect gaps that are not visible in a simulated environment without professionals in the area.

Within respect to the break time periods some progresses have been made but the work is still not complete. Regarding the second item a model was developed to track the brigades and compare their real routes with the predicted by the system to support the field-tests.

Within the scope of the project as a whole, there are development prospects that will directly and indirectly affect the module of the generation of flexible routes, such as:

- Implementation of risk matrices that will allow a determination of a more accurate inspection utility.
- The new pending complaints are already being taken into account automatically but the extraction of other new information from the ASAE system may also improve the utility definition of specific Economic Operators.
- Development of a system in constant learning of visiting times to establishments. The quality of the obtained solutions will increase by replacing the manually inserted inspection times with the categorization of average control times by economic operator’s area of activity.

Acknowledgements. This work is supported by project IA.SAE, funded by Fundação para a Ciência e a Tecnologia (FCT) through program INCoDe.2030. This research was partially supported by LIACC (FCT/UID/CEC/0027/2020).

References

1. Allahviranloo, M., Chow, J., Recker, W.: Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E Logistics and Transportation Review* (2013). <https://doi.org/10.1016/j.tre.2013.12.004>
2. Bansal, S., Goel, R.: Multi Objective Vehicle Routing Problem: A Survey. *Asian Journal of Computer Science and Technology* pp. 1–6 (2018)
3. Barbosa, L., et al.: Automatic identification of economic activities in complaints. In: Martín-Vide, C., Purver, M., Pollak, S. (eds.) *SLSP 2019. LNCS (LNAI)*, vol. 11816, pp. 249–260. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31372-2_21
4. Barros, T., Oliveira, A., Lopes Cardoso, H., Reis, L.P., Caldeira, C., Machado, J.P.: Generation and Optimization of Inspection Routes for Economic and Food Safety. In: *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pp. 268–278. SciTePress (2020). DOI: 10.5220/0009182002680278, backup Publisher: INSTICC

5. Barros, T., et al.: Interactive Inspection Routes Application for Economic and Food Safety. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S., Orovic, I., Moreira, F. (eds.) WorldCIST 2020. AISC, vol. 1159, pp. 640–649. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45688-7_64
6. Braekers, K., Ramaekers, K., Nieuwenhuysse, I.V.: The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering* **99**, 300–313 (2016). <https://doi.org/10.1016/j.cie.2015.12.007>
7. Campbell, A.M., Wilson, J.H.: Forty years of periodic vehicle routing. *Networks* **63**(1), 2–15 (2014)
8. Cardoso, S.R.d.S.N.: *Optimização de rotas e da frota associada*. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisbon (2009)
9. Caric, T., Gold, H.: Vehicle routing problem. In: Teh, Vienna, Austria (2008)
10. Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J.: Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics* **6**(1), 51–79 (2015). <https://doi.org/10.1007/s13676-014-0074-0>
11. Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J.Y., Semet, F.: A guide to vehicle routing heuristics. *Journal of the Operational Research society* **53**(5), 512–522 (2002)
12. Cordeau, J.F., Gendreau, M., Laporte, G.: A Tabu Search heuristic for periodic and multi-depot vehicle routing problems. *Networks* **30**, 105–119 (1997). [https://doi.org/10.1002/\(SICI\)1097-0037\(199709\)30:23.3.CO;2-N](https://doi.org/10.1002/(SICI)1097-0037(199709)30:23.3.CO;2-N)
13. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* **52**, 928–936 (2001)
14. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Management science* **6**(1), 80–91 (1959)
15. Filgueiras, J., Barbosa, L., Rocha, G., Lopes Cardoso, H., Reis, L.P., Machado, J.P., Oliveira, A.M.: Complaint Analysis and Classification for Economic and Food Safety. In: *Proceedings of the Second Workshop on Economics and Natural Language Processing*. pp. 51–60. Association for Computational Linguistics, Hong Kong (Nov 2019). 10.18653/v1/D19-5107
16. Flood, M.M.: The traveling-salesman problem. *Operations research* **4**(1), 61–75 (1956)
17. Hasle, G., Lie, K.A., Quak, E.: *Geometric modelling, numerical simulation, and optimization*. Springer (2007)
18. Ho, W., Ho, G.T., Ji, P., Lau, H.C.: A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence* **21**(4), 548–557 (2008). <https://doi.org/10.1016/j.engappai.2007.06.001>
19. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society* **7**(1), 48 (Feb 1956). <https://doi.org/10.2307/2033241>
20. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research* **59**(3), 345–358 (1992)
21. Letchford, A.N., Lysgaard, J., Eglese, R.W.: A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society* **58**(12), 1642–1651 (2007). <https://doi.org/10.1057/palgrave.jors.2602345>
22. Machado, P., Tavares, J., Pereira, F.B., Costa, E.: Vehicle routing problem: Doing it the evolutionary way. In: *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. pp. 690–690. Morgan Kaufmann Publishers Inc. (2002)

23. Montoya-Torres, J.R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., Herazo-Padilla, N.: A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering* **79**, 115–129 (2015). <https://doi.org/10.1016/j.cie.2014.10.029>
24. Ritzinger, U., Puchinger, J., Hartl, R.F.: A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research* **54**(1), 215–231 (2016)
25. Ruiz, E., Soto-Mendoza, V., Barbosa, A.E.R., Reyes, R.: Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. *Computers & Industrial Engineering* **133**, 207–219 (2019). <https://doi.org/10.1016/j.cie.2019.05.002>
26. Toth, P., Vigo, D. (eds.): *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics (2002). DOI: 10.1137/1.9780898718515
27. Weise, T., Podlich, A., Gorldt, C.: Solving real-world vehicle routing problems with evolutionary algorithms. In: *Natural intelligence for scheduling, planning and packing problems*, pp. 29–53. Springer (2009)
28. Zhu, K.Q.: A new genetic algorithm for VRPTW. In: *Proceedings of the international conference on artificial intelligence*. Citeseer (2000)