

Article

An Attraction Map Framework of a Complex Multi-Echelon Vehicle Routing Problem with Random Walk Analysis

Anita Agárdi ¹, László Kovács ¹ and Tamás Bányai ^{2,*}

¹ Institute of Information Science, University of Miskolc, 3515 Miskolc, Hungary; agardianita@iit.uni-miskolc.hu (A.A.); kovacslaszlo@iit.uni-miskolc.hu (L.K.)

² Institute of Logistics, University of Miskolc, 3515 Miskolc, Hungary

* Correspondence: altamas@uni-miskolc.hu

Abstract: The paper aims to investigate the basin of attraction map of a complex Vehicle Routing Problem with random walk analysis. The Vehicle Routing Problem (VRP) is a common discrete optimization problem in field of logistics. In the case of the base VRP, the positions of one single depot and many customers (which have product demands) are given. The vehicles and their capacity limits are also fixed in the system and the objective function is the minimization of the length of the route. In the literature, many approaches have appeared to simulate the transportation demands. Most of the approaches are using some kind of metaheuristics. Solving the problems with metaheuristics requires exploring the fitness landscape of the optimization problem. The fitness landscape analysis consists of the investigation of the following elements: the set of the possible states, the fitness function and the neighborhood relationship. We use also metaheuristics are used to perform neighborhood discovery depending on the neighborhood interpretation. In this article, the following neighborhood operators are used for the basin of attraction map: 2-opt, Order Crossover (OX), Partially Matched Crossover (PMX), Cycle Crossover (CX). Based on our test results, the 2-opt and Partially Matched Crossover operators are more efficient than the Order Crossover and Cycle Crossovers.

Keywords: vehicle routing problem; fitness landscape; basin of attraction; random walk



Citation: Agárdi, A.; Kovács, L.; Bányai, T. An Attraction Map Framework of a Complex Multi-Echelon Vehicle Routing Problem with Random Walk Analysis. *Appl. Sci.* **2021**, *11*, 2100. <https://doi.org/10.3390/app11052100>

Academic Editor: Panagiotis G. Asteris

Received: 20 January 2021
Accepted: 22 February 2021
Published: 27 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the most important tasks of logistics is the cost-effective delivery of the right goods, to the right place in the right time considering any other constraints. These problem domains belong to the area of the Vehicle Routing Problem (VRP), which has developed in several versions since its first version [1] in 1959. The basic VRP (which is the easiest version of the VRP) is characterized by the fact that the position of a single depot and many customers are known in advance. The demands of the customers are also fixed in advance and, in the system, there is only a single product. The number of vehicles and their capacity constraints is also given in advance. The vehicles start their routes from the depot, they visit some customers, and then return to the depot. The objective function is the minimization of the length of the route. In the following, the main variants of the VRP are detailed that contributed to the development of our own VRP model. Vehicle Routing Problem with Single Depot [2] is a VRP where the system contains a single depot. In case of Vehicle Routing Problem with Multiple Depots [3,4] multiple depots are in the system, and the vehicles start their route from one of the depots, and after visiting the customers, they return to the depot from which they started their route. In the case of the Two-Echelon Vehicle Routing Problem [5], there are not only depots and customers but also satellites. The products are transported from the depot to intermediate locations (satellites) and then from the satellites to the customers. In most of the cases, there are several types of (capacity-constrained) vehicles in the system. If there is only one type of vehicle in the system, it can be said that the system has a homogeneous fleet (Homogeneous Fleet Vehicle Routing Problem [6]). If the system contains several types of vehicles, then the system has

a heterogeneous fleet (Heterogeneous Fleet Vehicle Routing Problem [7]). If vehicles have a capacity limit for the goods to be transported, then the system is called Capacitated Vehicle Routing Problem [4,8,9]. Customers to visit can also have a time window, this is called the Vehicle Routing Problem with Time Windows [4].

One of the latest VRP trends is the Fuel-Efficient Green Vehicle Routing Problem [10]; this case prioritizes not the minimization of the length of the route, but the minimization of the fuel consumption of the vehicles as the objective function. In the case of Cumulative Vehicle Routing Problem [11], the minimization of the waiting times for the customers is the objective function. In case of Fuzzy VRP, some factors are given with fuzzy numbers [12]. Milk-run VRP [13] is also a common problem, where the products, materials, etc., are transported between the warehouse and the production line. Another new trend is the application of electric vehicles [14].

Separate algorithms are usually implemented for each VRP model and we also find examples of general heuristics [4]. The investigated system is designed to solve the following VRP model: time windows, capacitated vehicles, multi-depot, and open VRP.

In logistics, beside VRP, we can also find examples of other transportation tasks [15,16], too.

The fitness landscape analysis [17] is a method to investigate the optimization space, it consists of the following elements: the set of possible solution states, neighborhood definition and the fitness function. Although encoding does not belong directly to the concept of the fitness landscape, it plays a big role in the analysis of space. The neighbors of the actual state can be created in several ways (for example, in the case of permutation representation, the 2-opt operator is a common way). To prove that a metaheuristic is suitable for a given optimization task, fitness landscape analysis can be a good alternative. In this case, we compare the main fitness landscape analysis methods with metaheuristic operators. One way of fitness landscape analysis is the trajectory-based sampling approach. In trajectory-based sampling, samples are taken from the search space. We start from a possible state point, then create the state of one or more possible neighbors, then select one of them, which is called the current state. In the literature, the following trajectory-based sampling techniques are investigated [18]: random walk, adaptive walk, reverse adaptive walk, neutral walk, reverse neutral walk, uphill-downhill walk. During a random walk, the next state is the randomly selected neighbor of the current state. In case of an adaptive walk, only a better state is selected from the neighbors of the current state. The reverse adaptive walk is the inverse of the adaptive walk, in which case a state that is worse than the current state is selected from the neighbors of the current state. During a neutral walk, we select a neighbor that shows a fitness value that is close to the current state point. The reverse neutral walk is the opposite of neutral walk, then we select a neighbor where the difference in fitness value is as large as possible. During the uphill downhill walk, we first perform an adaptive step and then a reverse adaptive walk is performed.

This paper focuses on the analysis of the random walk in a complex Vehicle Routing Problem. The remainder of the paper is organized as follows: in Section 2, the related works are detailed. In Section 3, the Multi-Echelon Vehicle Routing Problem is presented; in Section 4, the basin of attraction map with random walk is analyzed; in Section 5, the results and discussion is detailed, and after that the summary Section follows.

2. Related Works

In this section, we present some proposals in connection with fitness landscape analysis. Examination of the optimization space is a useful component in analyzing the general behavior of the problem domain. The structure of the optimization space influences the efficiency of convergence, such as the quality of the obtained global or local optimum. Examining the fitness space can be useful for selecting the appropriate optimization methods (which operators are worth using for heuristic algorithms). One benefit of the presented methods is that it can be used as the foundation for further formal theoretical analyzes,

there is no need to perform expensive experiments (such as when comparing the results of metaheuristic algorithms with test results).

Our analyzes include Vehicle Routing Problem [19,20], Traveling Salesman Problem [21–23], Job Scheduling Problem [21], Test Function [24], Quadratic Assignment Problem [25]. The authors of [19] perform a fitness landscape analysis of the Vehicle Routing Problem. Information is analyzed from a theoretical perspective, and the operators of the Genetic Algorithm (Swap, Inversion, Insertion, Displacement, Partially Matched Crossover—PMX, Uniform Order Crossover—UOX, Cycle Crossover—CX) were chosen as the subject of their analysis. The concepts of information content, partial information content, and density-basin information are reported. The authors of [20] also perform a fitness landscape analysis of the vehicle routing problem. After presenting the mathematical model, the authors review the concepts of search space (fitness landscape). Then, in proportion to the fitness values, the averages of the distances taken from the solutions are illustrated and the distances taken from the best solution are also plotted. The authors of [22] focus on the analysis of the Memetic Algorithm in solving the Traveling Salesman Problem. After discussing each concept of fitness landscape, the authors also plot each solution in a coordinate system according to fitness values. The subject of the analysis is the distance from the optimum and the differences in fitness. Analyses are performed on benchmark data sets. The Traveling Salesman Problem and the No-Wait Job Scheduling are discussed by the authors in [21]. Fitness landscape analysis is performed as autocorrelation, time to local optimum, number of local optimums, its increase as the problem increases, distances from the optimum, the probability that we will reach the optimum. An example of the analysis of benchmark continuous functions can also be found in the literature, for example, in [24] the Differential Evolution Algorithm fitness landscape analysis is presented. Presentations of the most important concepts of fitness landscape: fitness distance correlation, correlation length, epistasis, evolvability and neutrality. The authors focused on a continuous optimization task. Benchmark functions were solved such as Sphere function, Rosenbrock function, Step function, Schwefel function, Rastrigin function, Griewangk function, Ackley function, Easom function, Schwefel's Ridge function. Schwefel function, the Easom function, the Schwefel's Ridge function had a small fitness distance correlation, which means, that a Hill Climbing algorithm can easily solve these problems. The analysis of the Traveling Salesman Problem with benchmark data sets is also presented in [23], by comparing the Iterated Local Search and Hill Climbing Algorithm. In this article, experiments were executed on benchmark data sets. They applied 3 TSP-lib instances for the simulations. The distance and running time compared to the best result known so far are compared. An example of the analysis of the Quadratic Assignment Problem (QAP) fitness landscape can also be found in [25]. In addition to presenting the task, operators are also presented as pairwise exchange or swap. In addition, the basin of attraction, the local optima network, the random walk autocorrelation function, and the autocorrelation length are also detailed. The Genetic Algorithm (GA) and Simulated Annealing (SA) were used as algorithms, Partially Matched Crossover (PMX) and Pairwise Exchange Mutation as operators. Correlation analysis is also performed, comparing the two algorithms. The number of local optima and shortest path to the optimum results are also produced. It was concluded that the GA was the stronger algorithm to solve all the studied classes of QAP instances.

3. The Multi-Echelon Vehicle Routing Problem

In this Section our Vehicle Routing Problem model is detailed. Our VRP model and test instance is based on our previous work [26]. Figure 1 illustrates the topology of our problem.

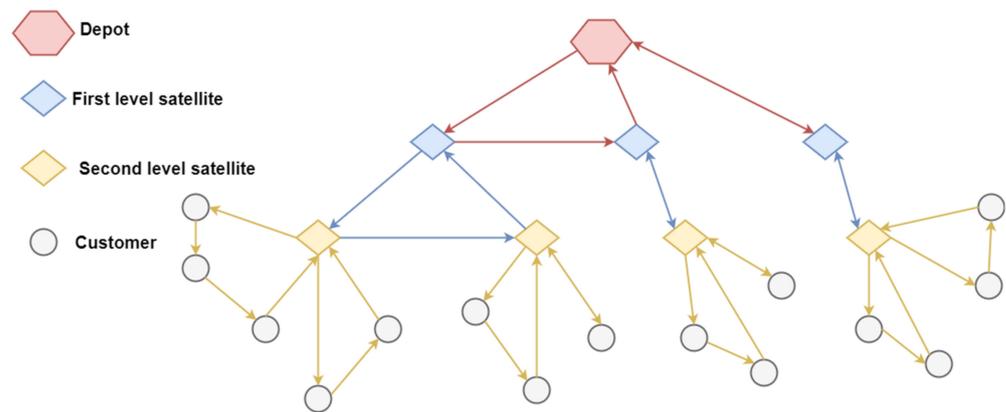


Figure 1. Multi-Echelon Vehicle Routing Problem model.

Figure 1 illustrates an example of a VRP sketch on which our own VRP example is based. Figure 1 indicates that our VRP consists of not only customers, indicated with $CU = \{cu_1, cu_2, \dots, cu_{nc}\}$ and a depot, indicated with $D = \{d_1\}$, but also satellites. However, the satellites are also different, in levels. The product is shipped from a central depot to the first level satellite $SL_1 = \{sl_{1,1}, sl_{1,2}, \dots, sl_{1,nsl_1}\}$, than to the second level satellite $SL_2 = \{sl_{2,1}, sl_{2,2}, \dots, sl_{2,nsl_2}\}$, etc., and then to the customers. The system contains a single type of product, indicated with $PR = \{pr_1\}$. We can also define the set of locations, with the union of the depot, satellites and customers: $LO = D \cup SL_1 \cup SL_2 \dots \cup CU = \{lo_1, \dots, lo_{nlo}\}$. The system also contains several vehicles, indicated with $VE = \{ve_1, \dots, ve_{nve}\}$. Our system contains several components, which can be categorized in the following way: node components, vehicle components, time components, product components, cost components, and metrics (objective function components). To formalize the components, the following notations are introduced: i, j is the node index, k is the vehicle index. The product has not got any special index, because the system contains one type of product. In the following, we use a function-oriented description of the costs associated with the main components of the VRP system in order to provide a more general formalism. The arguments of the functions are equal to the main cost parameters of the VRP.

The objective is the minimization of the length of the route ($minz_{lr}$), fuel consumption ($minz_{fc}$), vehicle rental fee ($minz_{rf}$), route time ($minz_{rt}$), unvisited customers ($minz_{uc}$), loading time ($minz_{lt}$), unloading time ($minz_{ut}$), loading cost ($minz_{lc}$), unloading cost ($minz_{uc}$), administration cost ($minz_{ac}$), quality control cost ($minz_{qcc}$) and the maximization of the route status ($minz_{rs}$). For the objective function, it is necessary to normalize the individual components of the objective function. The objective function is a combination of these components, which can be written in the following way: $Z = \{(w_{lr} \times z_{lr}^{norm})^2 + (w_{fc} \times z_{fc}^{norm})^2 + (w_{rf} \times z_{rf}^{norm})^2 + (w_{rt} \times z_{rt}^{norm})^2 + (w_{uc} \times z_{uc}^{norm})^2 + (w_{lt} \times z_{lt}^{norm})^2 + (w_{ut} \times z_{ut}^{norm})^2 + (w_{lc} \times z_{lc}^{norm})^2 + (w_{uc} \times z_{uc}^{norm})^2 + (w_{ac} \times z_{ac}^{norm})^2 + (w_{qcc} \times z_{qcc}^{norm})^2 + (w_{rs} \times z_{rs}^{norm})^2\}^{1/2}$, where z^{norm} means the normalization of an objective function component, and w means the weights of each components.

To the node component belongs the travel time between the nodes, which can be written with the following function: $TravelTime(lo_i, lo_j, ve_k)$, so the travel time varies by locations and vehicle types. The travel distance between the nodes can be written in the following way: $TravelDistance(lo_i, lo_j, ve_k)$, so the travel distance varies also by locations and vehicle types. Route status between the nodes is also defined: $RouteStatus(lo_i, lo_j, ve_k)$, it also varies by locations and vehicles. By route condition we mean the technical condition of the route (e.g., potholes) and the security of the route (e.g., robberies). The aim is to maximize the average state value of the generated path. To the vehicle component belongs the capacity constraint of the vehicle, fuel consumption and rental fee.

The vehicles have a capacity limit to the product, it can be written in the following way: $CapacityLimit(v_{e_k})$, it depends on vehicles and product (but there is only one product in the system). The vehicles have fuel consumption, which varies by vehicles; it can be written in the following way: $FuelConsumption(v_{e_k})$. In the system, there are own fleet of vehicles and also rented vehicles. The rented vehicles have rental fees, indicated with $RentalFee(v_{e_k})$. The time component indicates components in the system in connection with time, which are the followings: $LoadingTime(l_{o_i}, v_{e_k})$, it indicates, that the loading time depends on locations, and vehicles. In our system there is also unloading time, which can be written in the following way: $UnloadingTime(l_{o_i}, v_{e_k})$. The loading and unloading time depend on location and vehicle. In the locations, administration time can also arise in connection with the product; the following function indicates this component: $AdministrationTime(l_{o_i}, v_{e_k})$. In connection with product, there is only one component in our system, the product demand of each location, it can be written in the following way: $ProductDemand(l_{o_i})$. The product demand depends on location and product (but there is a single product in our system). Our system consists of the following cost components: loading cost, unloading cost, administrative cost, quality control cost. The loading cost can be written in the following way: $LoadingCost(l_{o_i}, v_{e_k})$ and unloading cost can be formulated in the same way: $UnloadingCost(l_{o_i}, v_{e_k})$. The administration cost is written with $AdministrationCost(l_{o_i}, v_{e_k})$ and quality control cost with $QualityControlCost(l_{o_i}, v_{e_k})$. All of the cost components depend on locations and vehicles.

Our Vehicle Routing Instance

In the tests, we use artificially generated data sets. The test system contains a single depot where the position is generated in interval (0, 100). The satellites in the test system can be divided into two levels. The satellites of the first level were generated with position index in (200, 300), and the positions of the second level are from (400, 500). The number of the satellite nodes is equal to 10 and the number of customers is set to 15. Each level contains two vehicles. The capacity constraints of the vehicles are set to a value between 10,000 and 50,000, and the fuel consumptions are generated between 10 and 100. As detailed above, the costs values in our system are generated from the interval (10, 50). The temporal parameter values are generated in (30, 50). The travel distances between the nodes are calculated with Euclidean distance, the travel time values are from (10, 100), and the route status values are between 100 and 500.

4. Fitness Landscape Analysis: Neighborhood Operators and Random Walk Metrics

Fitness landscape [17] definition covers the set of possible states, the definition of the neighborhood and the fitness function. Many neighborhood operators have appeared in the literature, it depends on whether the problem is a discrete or continuous optimization problem. In the following, the different neighborhood operators of our discrete optimization problem are detailed.

We can distinguish single operand and two-operand operators. In our Multi-Echelon VRP system, the operands are permutations representing the processing order of nodes. During the evaluation, each element of the permutation is lined up and assigned to a vehicle until a solution (vehicle capacity) is encountered. When the constraint is not met, we assign the nodes to a next vehicle.

2-opt [27] operator is widely used in the case of permutation representation of optimization problems (Figure 2).

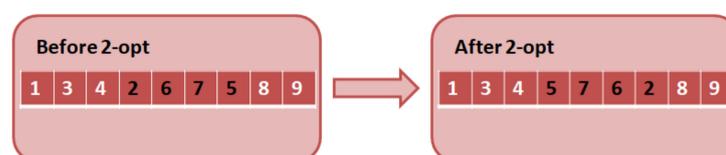


Figure 2. 2-opt operator.

In the case of 2-opt, two edges are swapped. Swapping means the selection of a given section of the permutation and the reversing of this selected section.

Within the frame of the fitness landscape analysis, we can analyze the performance of the mutation operators. For neighborhood generation the following operators can be used: cycle crossover, order crossover, partially matched crossover.

Figure 3 demonstrates a cycle crossover operator [28]. The cycle crossover locates the cycles in the path. Both parents are permutation, so the cycle crossover takes advantage of the fact that the elements are listed twice (exactly once in both parents). In the example, the (1,9) pair are taken first. The first child collects the first element of the first parent, and the second child receives the first element of the second parent. Then, the second child collects the element denoted by 9. Therefore, we have to look for the item 9 in the first parent, and the first child puts this in the right place. The second child receives the element 1. The first child has already got the element 1, so the circle is closed. The next steps can be summarized in the followings: the first child gets the other elements from the second parent, and the second child gets the other elements from the first parent.

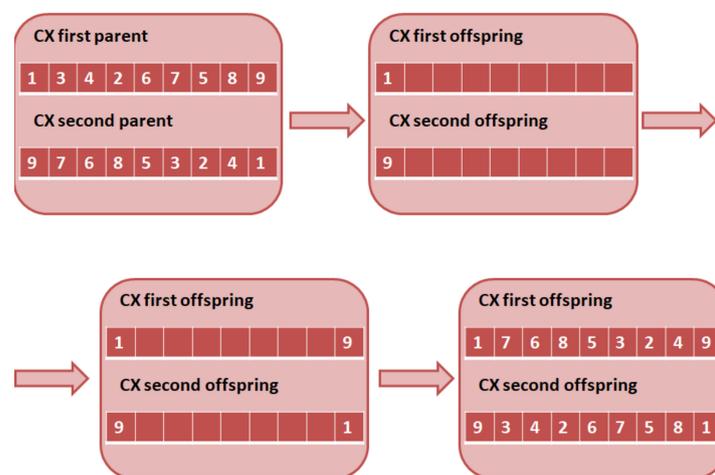


Figure 3. Cycle crossover (CX) operator.

Figure 4 shows the order crossover operator [28]. This operator uses a random fitting section. The position of the chosen fitting section must be the same for both parents. In the second child, the elements of the fitting section of the first parent are substituted with letter H (holes). Similarly, in the first child, the elements of the fitting section of the second parent are substituted with letter H (holes). The holes will be pushed up in the fitting section. After that step, the holes of the first child are substituted with the fitting section of the second parent, and the holes of the second child are substituted with the fitting section of the first parent.

Figure 5 demonstrates the partially matched crossover operator [28]. In the case of this crossover, a random fitting section is chosen. The fitting section of both parents must be in the same position. Pairs are formed from the elements of the fitting section of the two parents. Here, in pairs, the exchange operation itself takes place independently of each other. Pairs denote the elements to be exchanged, but once the elements to be exchanged have been determined, the exchange in both genes and the other gene occurs independently of each other. In our example, we can identify the following pairs: (2,8), (6,5), (7,3), (5,2). We copy parents; these copies will be the new children of the parents. These pairs are changed in the first and then the second child. In our example, we change the 2 with 8, then the 6 with 5, then the 7 with 3 then the 5 with 2.

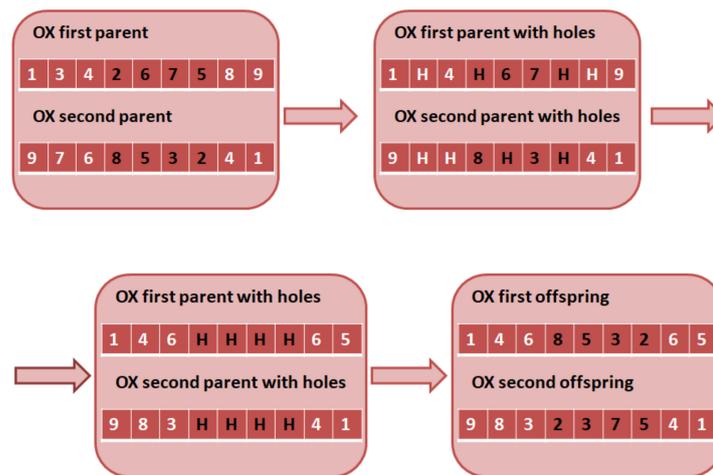


Figure 4. Order crossover (OX) operator.

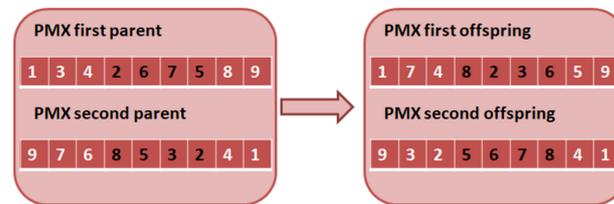


Figure 5. Partially Matched crossover (PMX) operator.

Within the frame of this operation, it is also important to measure the distance between the potential solutions. In this paper, three techniques are presented, which are the followings: fitness distance, hamming distance, and basic swap sequence distance. Fitness distance is the absolute value of the difference in fitness value between the two solutions. The hamming distance [29] is the number of differences between the representations of the two solutions. The basic swap sequence distance [30] is the number of swaps between the representations of the two solutions.

Basin of attraction are solutions, which convergence to a certain optimum [18]. The goal is to get out of this as easily as possible, because when an algorithm gets to a local optimum, if it has a hard time getting out of it, it has less chance of exploring space, thus reaching the global optimum. Our fitness landscape analysis uses trajectory-based sampling, in which a sample is taken from the search space. In this technique first, a possible state is selected, it will be first the actual state. With the neighborhood operator, we form one or more (in our case one) neighbor states, one of which will be the new current state. In the case of a random walk [18] sampling technique, the random neighbor state of the actual state will be the actual state. The neighbor can be better or worse than the actual state. We also prepare an information theory analysis of the fitness landscape, which quantifies the frequency of relative sequences between neighbors. The following important metrics can be evaluated [18]:

- **Partial Information Content:** The fitness value of each of the next neighbors is subtracted. It can be a positive number (ascending) that means 1, a negative number (descending), it means -1 , or zero (not changing), it means 0. Therefore, the values $-1, 0, 1$ are indicator values. They show whether the fitness of the neighbor is increasing, decreasing, or remaining of the same. The sequence of the indicator values is given with $s_{original}$. This procedure removes zero (non-variable) values from the sequence and then leaves only the variable slopes (i.e., descending values after ascending, as-

ending values after descending), indicated with $s_{nonzero}$. The value thus obtained is divided by the length of the original sequence.

$$M(\varepsilon) = \frac{s_{nonzero}}{s_{original}} \quad (1)$$

A value of 1 is rugged while a value of 0 is flat.

- Expected Partial Information Content: this measure requires the partial information content value. The expected partial information content can be calculated with the following way:

$$E[M(\varepsilon)] = \frac{n \times M(\varepsilon)}{2} \quad (2)$$

In the formula, n is the length of the $s_{original}$, $M(\varepsilon)$ is the partial information content. Expected partial content is not a probability variable, it shows the shape of the fitness landscape.

- Information Stability: the fitness values of each of the next neighbors are subtracted from each other. The greatest value obtained will be information stability.
- Entropy: the fitness values of each of the next neighbors are subtracted. It can be a positive number (ascending) that means 1, a negative number (descending), it means -1 , or zero (not changing), it means 0. This gives a sequence. Then, the entropy (entropy) is calculated with the following formula, where P_i is the relative frequency of the symbols:

$$-\sum_i P_i \ln P_i \quad (3)$$

- Information Content: the fitness values of each neighbor solutions are subtracted each other. It can be a positive number (ascending) that means 1, a negative number (descending), it means -1 , or zero (not changing), it means 0. This gives a sequence. Then, the information content can be calculated with the following formula, where $P_{[pq]}$ denotes the relative frequency of the consecutive symbols, where $p, q \in \{1, 0, -1\}$:

$$H(\varepsilon) = \sum_{p \neq q} P_{[pq]} \times \log_6 P_{[pq]} \quad (4)$$

- Density Basin Information: the same as information content, but the basis of the logarithm is different. The Equation of the density basin information is the following:

$$h(\varepsilon) = \sum_{p \neq q} P_{[pq]} \times \log_3 P_{[pq]} \quad (5)$$

- Regularity: the fitness values of each of the next neighbors are subtracted. These values represent regularity.
- Evolvability portrait: the fitness value of all neighbors of the current solution are divided by the number of neighbors

5. Results and Discussion

In this section, first, the search space is filtered, to find out what solutions space consists of, how they relate to each other, how far apart they are. This step is based on random sampling.

Then, the random walk, which is a trajectory-based sampling, is performed. Results of the random walk are compared with each other, then the result is also analyzed from the point of view of information theory. Operators are good if they give varied results, because then they explore the space well, the basin of attraction is small, so they can easily get out of a local optimum.

The analyzes were performed in Java with our own implementation. The test was running on a personal computer.

5.1. Scattered Filtering of the Search Space

This step means that the search space is sampled. Therefore, we randomly generate solutions and then analyze them. In the tests, we created 100 solutions at random. We chose a relatively low value, even though the task is complicated one, but the number of nodes to be visited is not very large. The following analysis methods were considered: distance of solutions, the distance of the best solution, the distance of the best solution and solutions, cost density. The following distances were analyzed: fitness, hamming, basic swap sequence. For average distances for the search space points were calculated by calculating the average distance to all other search space points. Therefore, each solution is assigned to a point-level average distance.

Figure 6a shows fitness values and average fitness distances. Fitness values range from $\approx 120,000$ to $\approx 160,000$ (fitness unit), while average fitness distances range from ≈ 5000 to $\approx 19,000$ (fitness unit). These values also show a parabolic function because small and large fitness values are paired with a large average distance, while medium fitness values are associated with a medium average distance. Figure 6b shows fitness values and average hamming distances. The average hamming distances in each case range from about 35 to 36 (hamming distance unit). Figure 7 shows fitness values and average basic swap sequence distances. The average basic swap sequence distances in each case range from about 28 to 29 (basic swap sequence distance unit). The fitness values and fitness distances from the best solution (the best of the generated solutions) show a linear function, the higher the fitness value of the solution, the greater the distance from the best solution (the best of the generated solutions). According to Figure 6b, the average hamming distances are around 35 (hamming distance unit), while according to Figure 7, the average basic swap sequence distances are around 29 (basic swap sequence distance unit). Fitness values range from $\approx 120,000$ to $\approx 160,000$ (fitness unit), and fitness distances range from $\approx 4,000$ to $\approx 36,000$ (fitness unit). In case of fitness values and hamming distance from the best solution (the best of the generated solutions), the Hamming distances range from 30 to 40 (hamming distance unit). The basic swap sequence distance from the best solution (the best of the generated solutions) range from 24 to 35 (basic swap sequence distance unit). The fitness distance from the best solution (the best of the generated solutions) and the average of the fitness distances from the other solutions are described by a parabolic function, thus, low and high distances mean high average distances from the best (best of generated solutions) solution, while medium distances mean low average distances from the best (best of generated solutions) solution. The cost density value shows that almost every solution has a different fitness value.

Table 1 presents the meaning of evaluation and Table 2 illustrates the test results.

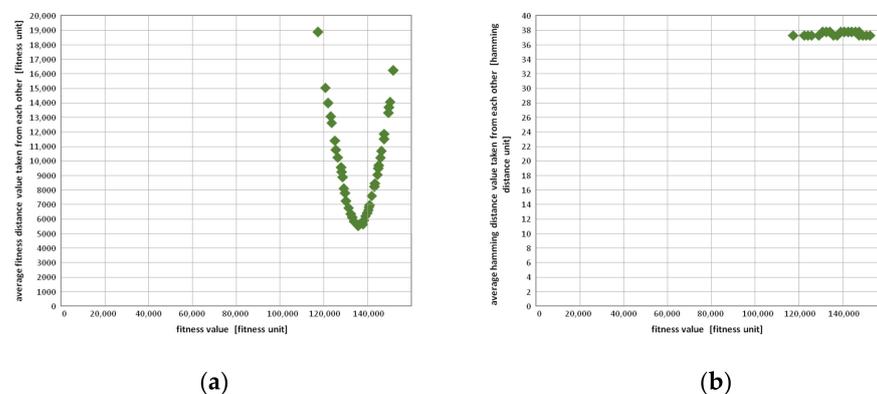


Figure 6. The fitness distance (a) and the hamming distance (b) of solutions.

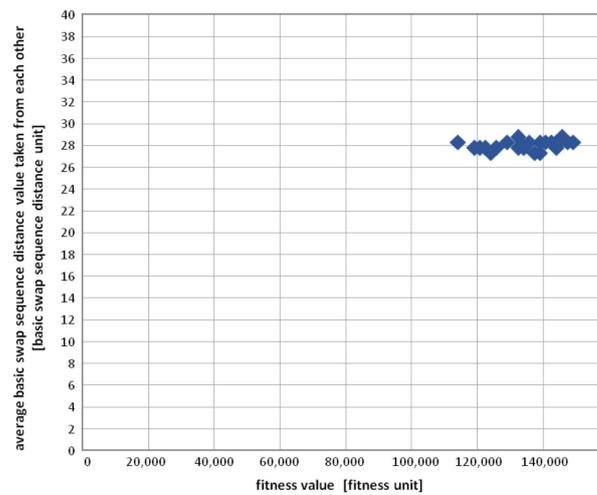


Figure 7. The basic swap sequence distance of solutions.

Table 1. Analysis of the scattered filtered search space: meaning of evaluation.

Analysis of Filtered Search Space	
Type	Optimal Value
Fitness values	Small value difference is optimal. Large value difference between lower and upper bound in case of high mountains and valleys
Average of fitness distances	
Average of hamming distances	Small lower and upper bound are optimal. At high average distances, it is more difficult for the optimization algorithm to navigate through the field because the task is complicated
Average of basic swap sequence distances	
Fitness distances of the best solution	A small difference in value is optimal. At great distances, we are far from the optimum
Hamming distances of the best solution	Small lower and upper bound are optimal. At large distances, it is more difficult for the optimization algorithm to get to the optimum
Basic swap sequence distances of the best solution	
Cost density	Small lower and upper bound are optimal. For high values, the solutions are varied

Table 2. Analysis of the scattered filtered search space: results.

Analysis of Filtered Search Space			
Type	Distance	Lower Bound	Upper Bound
Fitness values	Fitness (fitness unit)	120,000	160,000
Average of fitness distances	Fitness (fitness unit)	5000	19,000
Average of hamming distances	Hamming (hamming distance unit)	35	36
Average of basic swap sequence distances	Basic swap sequence (basic swap sequence distance unit)	28	29
Fitness distances of the best solution	Fitness (fitness unit)	4000	36,000
Hamming distances of the best solution	Hamming (hamming distance unit)	30	40
Basic swap sequence distances of the best solution	Basic swap sequence (basic swap sequence distance unit)	24	35
Cost density	-	1	3

5.2. Radom Walk Analysis: The Comparison of the Results to Each Other

The average fitness distances (where distances are calculated from the elements of the solution set) during the random walk and 2-opt operator describe a parabolic function based on Figure 8a. Fitness values range from 120,000 to 140,000 (fitness unit), while average fitness distances range from 2500 to 9000 (fitness unit). If a solution has a low or high fitness value, the average fitness distance from the other solutions is high, while if it has a medium fitness value, the average distance is low. According to Figure 8b, average hamming distances are not affected by fitness value. Average hamming distances range from 28 to 34 (hamming distance unit). The average basic swap sequence distances (Figure 9) are not affected by fitness value. The average basic swap sequence distances are between 20–26 (basic swap sequence unit).

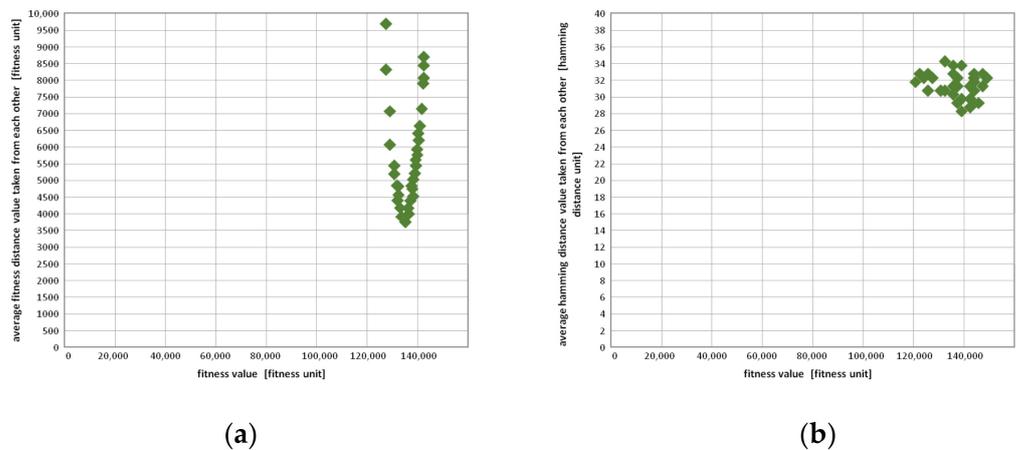


Figure 8. The fitness values and their average fitness distances (a) and the average hamming distances (b) from each other in case of 2-opt operator.

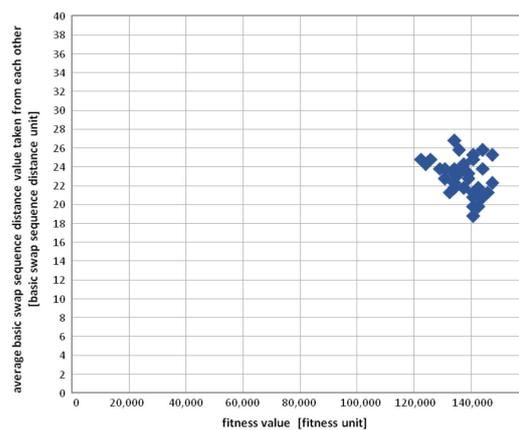


Figure 9. The average basic swap sequence distances from each other in case of 2-opt operator.

In the following, the results of order crossover operator is presented, where distances are calculated from the elements of the solution set. The average fitness distances during the random walk and order crossover operators describe a parabolic function. Fitness values range from 120,000 to 140,000 (fitness unit), while average average fitness distances range from 2000 to 10,000 (fitness unit). If a solution has a low or high fitness value, the average fitness distance from the other solutions is high, while if it has a medium fitness value, the average distance is low. The average hamming distances are not affected by fitness value. Average hamming distances range from 30 to 34 (hamming distance unit). The average

basic swap sequence distances are also not affected by fitness value. The average basic swap sequence distances are between 23–28 (basic swap sequence unit).

In the following, the results of the cycle crossover operator, where the distances are calculated from the elements of the solution set is presented. The distances describe a parabolic function. Fitness values range from 120,000 to 140,000 (fitness unit), while average fitness distances range from 1800 to 5000 (fitness unit). If a solution has a low or high fitness value, the average fitness distance from the other solutions is high, while if it has a medium fitness value, the average distance is low. The average hamming distances are not affected by fitness value. Average hamming distances range from 25 (hamming distance unit) to 30 (hamming distance unit). The average basic swap sequence distances are also not affected by fitness value. The average basic swap sequence distances are between 20–24 (basic swap sequence unit).

In the following, the results of the partially matched crossover operator, where distances are calculated from the elements of the solution set is detailed. The average fitness distances during partially matched crossover operator describes a parabolic function. Fitness values range from 120,000 to 140,000 (fitness unit), while average fitness distances range from 2000 to 7500 (fitness unit). If a solution has a low or high fitness value, the average fitness distance from the other solutions is high, while if it has a medium fitness value, the average distance is low. The average hamming distances are not affected by fitness value. Average hamming distances range from 30 (hamming distance unit) to 34 (hamming distance unit). The average basic swap sequence distances are also not affected by fitness value. The average basic swap sequence distances are between 23–27 (basic swap sequence distance unit).

In the following, the results of the 2-opt operator is presented, where the distances are calculated from the best element of the solution set. Figure 10a shows the fitness values and the fitness distance from the best solution for random walk and 2-opt. The higher the fitness value of a solution, obviously the farther away from the best solution. According to Figure 10b the hamming distances range from 2 to 38 (hamming distance unit). The greater the fitness value of a solution, the greater the hamming distance from the best solution. According to Figure 11, the basic swap sequence distances are between 1–32 (basic swap sequence distance unit). The higher the fitness value of a solution, the greater the distance of the basic swap from the best solution.

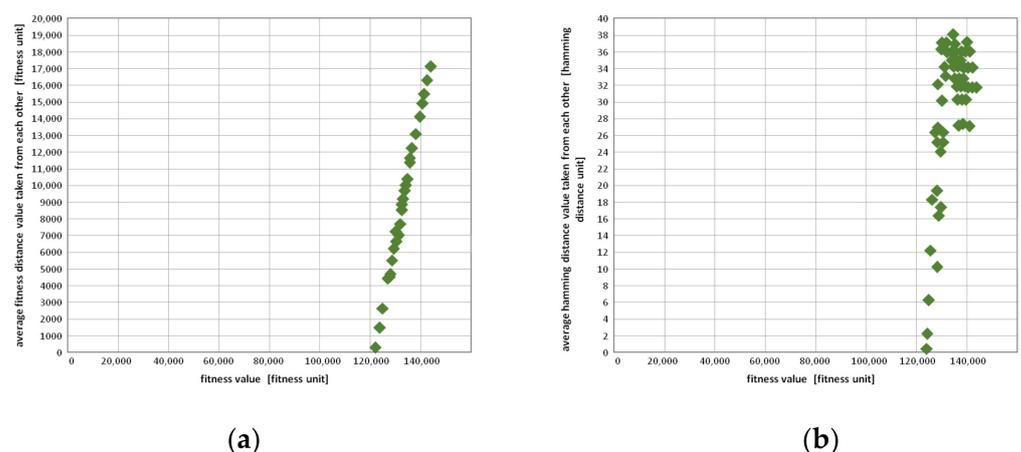


Figure 10. The fitness values and their fitness distances (a) and hamming distances (b) from the best of the solutions in case of 2-opt operator.

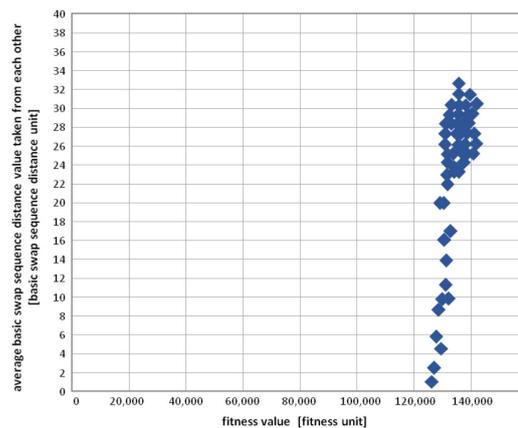


Figure 11. The basic swap sequence distances from the best of the solutions in case of 2-opt operator.

In the following, the results of the order crossover operator are detailed, where distances are calculated from the best element in the solution set. The higher the fitness value of a solution, obviously the farther away from the best solution. The hamming distances range from 2 to 40 (hamming distance unit). The greater the fitness value of a solution, the greater the hamming distance from the best solution. The basic swap sequence distances are between 2–34 (basic swap sequence distance unit). The higher the fitness value of a solution, the greater the distance of the basic swap from the best solution.

The results of the cycle crossover operator and a partially matched crossover are detailed, where distances are calculated from the best element of the solution set. These analyzes also show that the higher the fitness value of a solution, the greater the distance of the fitness from the best solution, and the hamming and basic swap sequence distances also increase as a function of the fitness value. For a cycle crossover, the hamming distance is 0–38 (hamming distance unit), while the basic swap sequence distance is 0–29 (basic swap sequence distance unit), and for a partially matched crossover, the hamming distance is 2–40 (hamming distance unit) and the basic swap sequence distance is 1–34 (basic swap sequence distance unit).

In the following, the distance from the best solution and the average of the distances from the other solutions are presented. According to Figure 12a, this is a parabolic function for 2-opt operator, so for large and small distances the average distances are large, while for medium distances the average distances are small for random walk and 2-opt. This shows that the extreme case is not in the middle of the distribution, it is located close to the elements in the center. According to Figure 12b the distance from the best solution during hamming distances does not affect the average distance here. Figure 13 shows the results for the basic swap sequence, here too the average basic swap sequence distance is not affected by the distance from the best solution to the given result. In the case of order crossover, we also get a parabolic function for fitness distances. The results for cycle crossover and partially matched crossover are similar to those for 2-opt and cycle crossover.

Figures 14 and 15 show the cost of density results, i.e., how many solutions with the same fitness value are present between each solution. Based on the results, in the case of 2-opt, almost every solution has a unique fitness value. In the case of order crossover and cycle crossover, some solutions have the same fitness value, while in the case of partially matched crossover, each solution also has a unique fitness value. Table 3. illustrates the evaluation methods of the results of random walk analysis, and Table 4. presents the summary of the results of the random walk where the solutions are compared to each other.

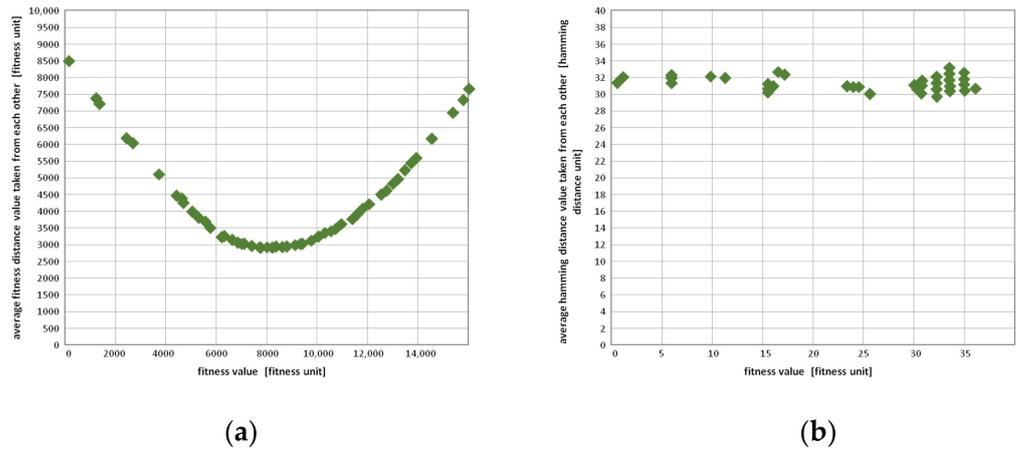


Figure 12. The distance from the best solution and the average of the fitness distances (a) and the hamming distances (b) from the other solutions in case of 2-opt operator.

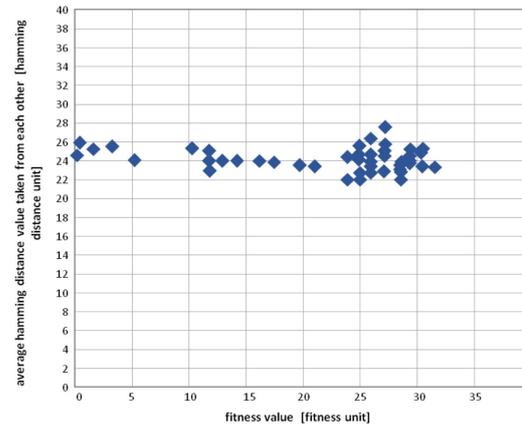


Figure 13. The basic swap sequence distances from the other solutions in case of 2-opt operator.

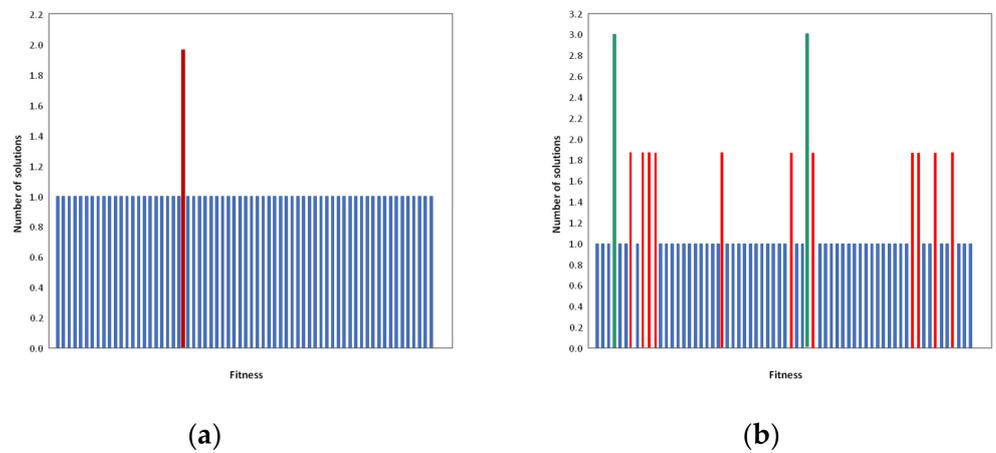


Figure 14. The cost of density in case of 2-opt (a) and order crossover operator (b).

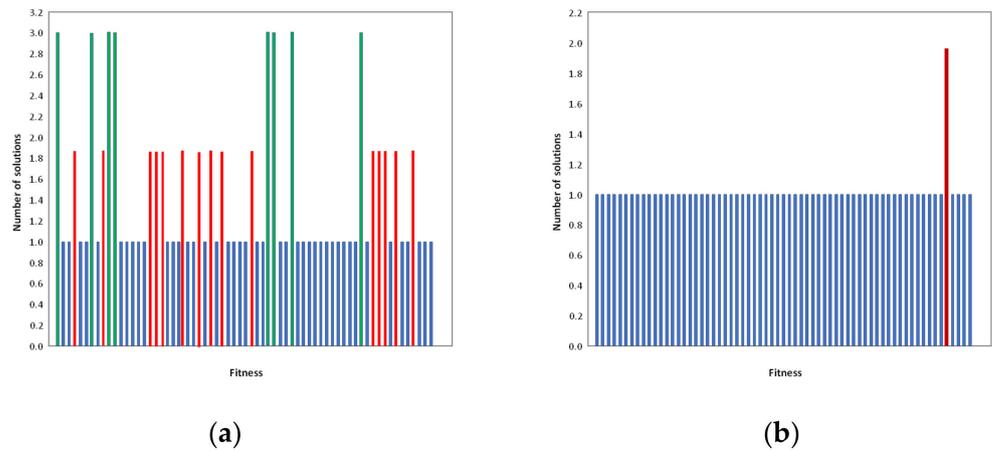


Figure 15. The cost of density in case of cycle crossover (a) and partially matched crossover operator (b).

Table 3. The evaluation methods of the results of random walk analysis.

Random Walk Analysis: The Comparison of Results to Each Other	
Type	Optimal Value
Fitness values	If the value difference between the lower and upper bound is large, the algorithm detects the space well. The lower bound value, on the other hand, must be small
Average of fitness distances	It explores space well over long distances
Average of hamming distances	
Average of basic swap sequence distances	Large upper bound and small lower bound. Then, it explores the space well (large upper bound) but also found a very good solution due to the small upper bound.
Fitness distances of best solution	
Hamming distances of best solution	At high values, it explores space well
Basic swap sequence distances of best solution	
Cost of density	Large upper bound and small lower bound. Then, it explores the space well (large upper bound) but also found a very good solution due to the small upper bound.
Fitness distance of filtered global optima	
Hamming distance of filtered global optima	
Basic swap sequence distance of filtered global optima	

Table 4. The summary of the results of the random walk where the solutions are compared to each other.

Type	Distance	Lower Bound	Upper Bound
2-opt			
Fitness values	Fitness (fitness unit)	120,000	140,000
Average of fitness distances	Fitness (fitness unit)	2500	9000
Average of hamming distances	Hamming (hamming distance unit)	28	34
Average of basic swap sequence distances	Basic swap sequence (basic swap sequence distance unit)	20	26
Fitness distances of best solution	Fitness (fitness unit)	1000	17,000
Hamming distances of best solution	Hamming (hamming distance unit)	2	38
Basic swap sequence distances of best solution	Basic swap sequence (basic swap sequence distance unit)	1	32
Cost of density	—	1	2
Fitness distance of filtered global optima	Fitness (fitness unit)	1000	18,000
Hamming distance of filtered global optima	Hamming (hamming distance unit)	30	40
Basic swap sequence distance of filtered global optima	Basic swap sequence (basic swap sequence distance unit)	22	34

Table 4. Cont.

Type	Distance	Lower Bound	Upper Bound
Order crossover			
Fitness values	Fitness (fitness unit)	120,000	140,000
Average of fitness distances	Fitness (fitness unit)	2000	10,000
Average of hamming distances	Hamming (hamming distance unit)	30	34
Average of basic swap sequence distances	Basic swap sequence (basic swap sequence distance unit)	23	28
Fitness distances of best solution	Fitness (fitness unit)	1000	16,000
Hamming distances of best solution	Hamming (hamming distance unit)	2	40
Basic swap sequence distances of best solution	Basic swap sequence (basic swap sequence distance unit)	2	34
Cost of density	—	1	3
Fitness distance of filtered global optima	Fitness (fitness unit)	0	11,000
Hamming distance of filtered global optima	Hamming (hamming distance unit)	28	38
Basic swap sequence distance of filtered global optima	Basic swap sequence (basic swap sequence distance unit)	22	32
Cycle crossover			
Fitness values	Fitness (fitness unit)	120,000	140,000
Average of fitness distances	Fitness (fitness unit)	1800	5000
Average of hamming distances	Hamming (hamming distance unit)	25	30
Average of basic swap sequence distances	Basic swap sequence (basic swap sequence distance unit)	20	24
Fitness distances of best solution	Fitness (fitness unit)	0	9500
Hamming distances of best solution	Hamming (hamming distance unit)	0	38
Basic swap sequence distances of best solution	Basic swap sequence (basic swap sequence distance unit)	0	29
Cost of density	—	1	3
Fitness distance of filtered global optima	Fitness (fitness unit)	1500	10,500
Hamming distance of filtered global optima	Hamming (hamming distance unit)	30	38
Basic swap sequence distance of filtered global optima	Basic swap sequence (basic swap sequence distance unit)	28	30
Partially Matched Crossover			
Fitness values	Fitness (fitness unit)	120,000	140,000
Average of fitness distances	Fitness (fitness unit)	2000	7500
Average of hamming distances	Hamming (hamming distance unit)	30	34
Average of basic swap sequence distances	Basic swap sequence (basic swap sequence distance unit)	23	27
Fitness distances of best solution	Fitness (fitness unit)	0	13,000
Hamming distances of best solution	Hamming (hamming distance unit)	2	40
Basic swap sequence distances of best solution	Basic swap sequence (basic swap sequence distance unit)	1	34
Cost of density	—	1	2
Fitness distance of filtered global optima	Fitness (fitness unit)	3500	17,000
Hamming distance of filtered global optima	Hamming (hamming distance unit)	30	40
Basic swap sequence distance of filtered global optima	Basic swap sequence (basic swap sequence distance unit)	22	34

5.3. Random Walk Analysis: Information Content Analysis

In this subsection, the information theory analysis of the random walk technique is evaluated. The following techniques were used: partial information content, expected partial information content, information stability, entropy, information content, density basin information, regularity, and evolvability portrait. Table 5. presents the information content analysis evaluation methods.

Table 5. Information content analysis evaluation methods

Information Content Analysis	
Method	Optimal Value
Partial Information Content	The bigger the better
Expected Partial Information Content	
Information Stability	
Entropy	The smaller the better
Information Content	
Density Basin Information	
Regularity	High values
Evolvability portrait	

The partial information content is 0.55 (partial information content unit) for 2-opt, 0.56 (partial information content unit) for order crossover, 0.47 (partial information content unit) for cycle crossover and 0.63 (partial information content unit) for partially matched crossover. Since the partial information content shows a rugged landscape for 1 and a flat landscape for 0, so in our case we get neither a flat nor a rugged landscape for all neighborhood techniques. The flattest is in the case of a partially matched crossover, so this operator may be the most efficient, while in the case of a cycle crossover it may be the most rugged, so this operator may be the worst.

The expected partial information content values are 27 (expected partial information content unit) for 2-opt, 28 (expected partial information content unit) for OX, 23 (expected partial information content unit) for CX and 31 (expected partial information content unit) for PMX.

The information stability, which shows the largest change in fitness of the neighbors, was given the following values: 7111.15 (information stability unit) for 2-opt, 5809.92 (information stability unit) for OX, 2728.38 (information stability unit) for CX and 5696.12 (information stability unit) for PMX. According to this, the largest change was for 2-opt and the smallest change was for CX.

The lower the entropy, information content, density basin information values, the better, because this means that the search is moving in the same direction (increasing or decreasing the fitness value of the neighbor of the current solutions).

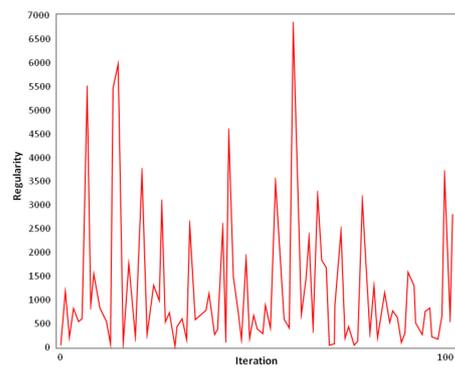
The entropy values are the followings: for 2-opt 1.5809 (entropy unit) for OX 2.5792 (entropy unit) for CX 2.9458 (entropy unit) and for PMX 2.1070 (entropy unit). According to this, the 2-opt has the smallest entropy and the CX has the largest.

The information content values are 0.4101 (information content unit) for 2-opt, 0.8252 (information content unit) for OX, 0.8831 (information content unit) for CX, and 0.6380 (information content unit) for PMX. According to this, the highest value was given by CX and the lowest by 2-opt.

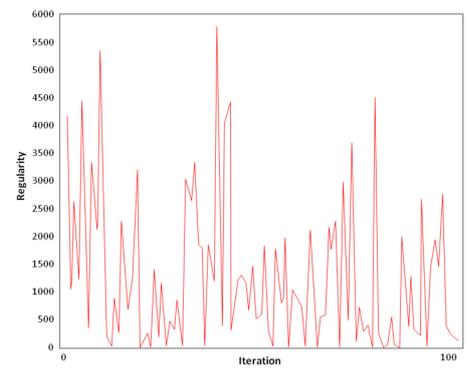
The density basin information values are 0.3285 (density basin information unit) for 2-opt, 0.2814 (density basin information unit) for OX, 0.4183 (density basin information unit) for CX, and 0.2888 (density basin information unit) for PMX.

The diagram of regularity values for each operator (2-opt, OX, CX, PMX) is shown in Figures 16 and 17 which shows ruggedness.

A graph of evolvability portrait values for each operator is shown in Figures 18 and 19, in which we also get a rugged function along the iterations. However, if we plot the values along fitness values, which are shown in Figures 20 and 21. We get a level result (function) that is independent of the fitness value. The higher the value of evolvability, the better the solution, because the more neighbors of the current solution are better than the current solution, so it is more likely to improve during the iteration. In fact, evolvability measures the probability of improvement.

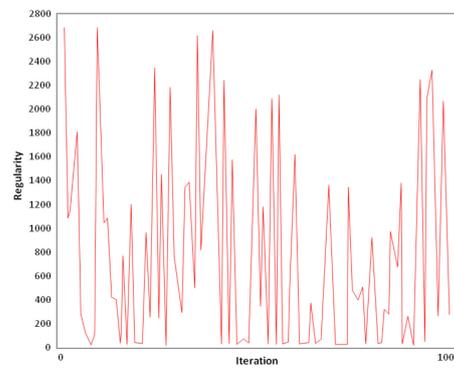


(a)

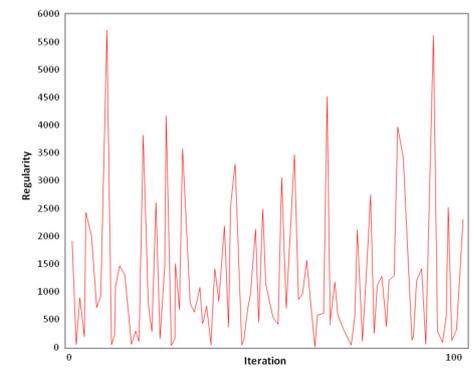


(b)

Figure 16. The regularity in case of 2-opt (a) and order crossover operator (b).

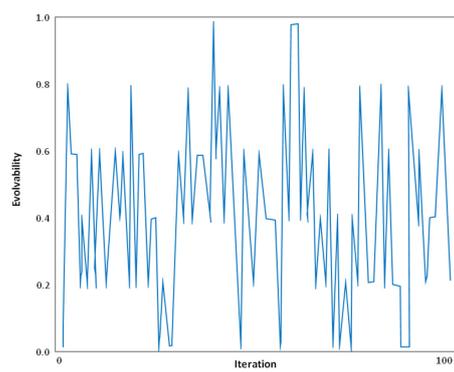


(a)

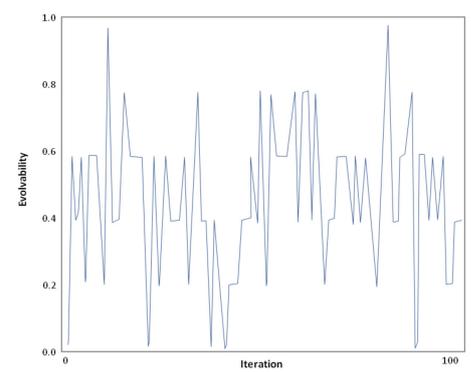


(b)

Figure 17. The regularity in case of cycle crossover (a) and partially matched crossover operator (b).



(a)



(b)

Figure 18. The evolvability along the iterations in case of 2-opt (a) and order crossover operator (b).

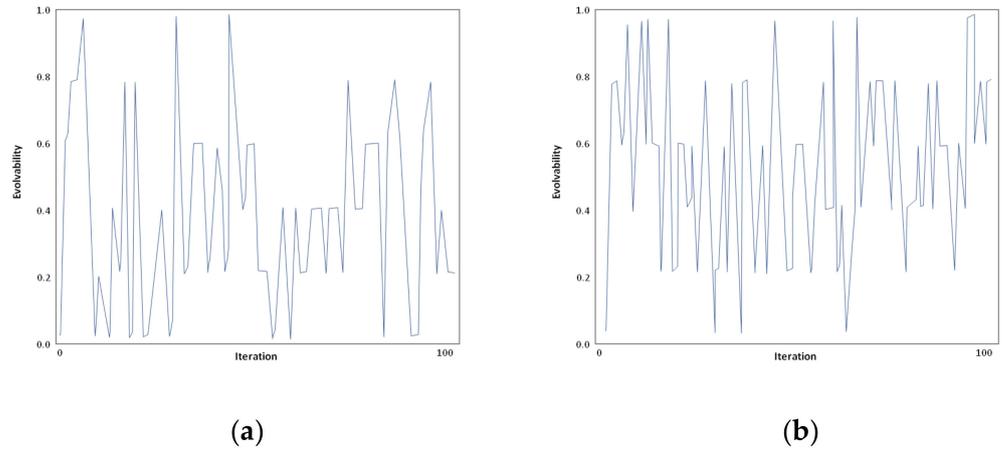


Figure 19. The evolvability along the iterations in case of cycle crossover (a) and partially matched crossover operator (b).

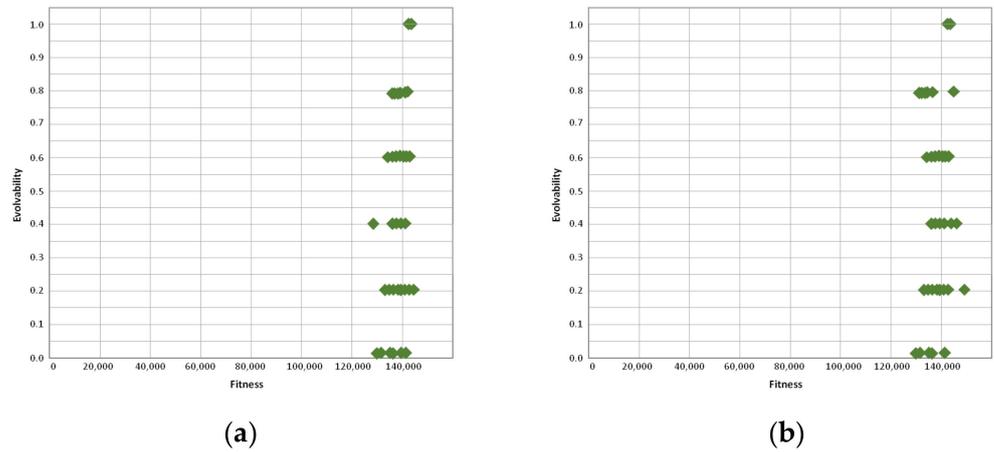


Figure 20. The evolvability in case of 2-opt (a) and order crossover operator (b).

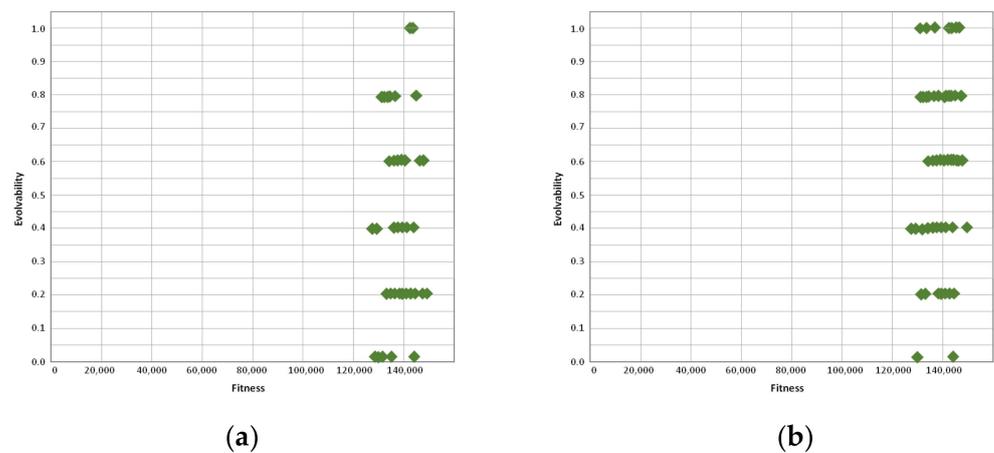


Figure 21. The evolvability in case of cycle crossover (a) and 2-opt operator (b).

5.4. Analysis of the Test Results

In the case of random walk, the average difference in fitness values of the solutions received by the operators is almost the same, the hamming and basic swap sequence

distances are very similar, in the case of order crossover and partially matched crossover the distances are slightly larger than in 2-opt and cycle crossover. Cost of density values also range from 1 to 2, so fitness values are also unique. In the case of a cycle crossover, the cost of density diagram shows that several solutions have the same fitness value, although the number of solutions with the same fitness value is only between 1 and 3. The distances from the best solution also move in large intervals, which also means that the space is well mapped by these operators.

Table 6 analyzes the efficiency of the operators (2-opt, order crossover, cycle crossover, partially matched crossover). The table shows that 2-opt and partially matched crossover (PMX) are more effective than cycle crossover (CX) and order crossover (OX).

Table 6. Summary results: basin of attraction map with random walk analysis.

Summary Results		
Analysis Method	Efficient Operator	Weak Operator
The comparison of results to each other	2-opt	
	OX	
	PMX	
Partial Information Content	PMX	CX
Expected Partial Information Content	PMX	CX
Information Stability	2-opt	CX
Entropy	2-opt	CX
Information Content	2-opt	CX
Density Basin Information	OX	CX
Regularity	2-opt	CX
Evolvability portrait	2-opt	CX
	PMX	OX

The partial information content value was the highest for the partially matched crossover, which means a flat landscape, so this technique proved to be the best during the measurement. In the case of the cycle crossover, the partial information content value is the lowest, so this technique proved to be the worst during the measurement.

The expected partial information content value is also the highest for the partially matched crossover, so this technique proved to be the best.

The information stability value is highest for 2-opt and lowest for cycle crossover. The value shows the largest change in fitness of the neighbors, which means that the higher the value for an operator, the more worthwhile it is to use. In the case of order crossover and partially matched crossover, the value between 2-opt and cycle crossover was obtained, which is almost the same for the two operators.

The lower the entropy, information content, density basin information values, the better the landscape, because this means that the search is moving in the same direction (increasing or decreasing the fitness value of the neighbor of the current solutions).

2-opt has the lowest entropy and CX has the largest. The order crossover and partially matched crossover values are nearly the same, between 2-opt and order crossover.

The 2-opt has the lowest information content value and the cycle crossover has the largest. The order crossover and partially matched crossover values are nearly the same, between 2-opt and order crossover operators.

Of the density basin information values, CX gave the highest value and OX the lowest.

According to the regularity diagram, the difference between the solutions of the cycle crossover is the largest, here we find the largest jumps. In the case of 2-opt, the jumps are the smallest.

Evolvability values are much higher on average for 2-opt and partially matched crossover than for cycle crossover and order crossover, so it is better to use 2-opt and partially matched crossover operators. The higher the value of evolvability, the better the solution, because the more neighbors of the current solution are better than the current

solution, so it is more likely to improve during the iteration. In fact, evolvability measures the probability of improvement.

5.5. Verifying our Results with Real Optimization Run

In this subsection, the results of our fitness landscape analysis technique are verified by optimization test runs. The tested operators can also be used in the Genetic Algorithm (GA), so this algorithm was selected for testing. The Genetic Algorithm maintains a population of solutions. Starting from an initial population, it continuously improves elements of the population with crossover and mutation techniques.

Table 7 presents the parameters and results of our test run. The test run was set up with OX in Test 1, PMX in Test 2, CX in Test 3, and 2-opt in Test 4. In Test 5, all three crossing operators were given equal emphasis.

Table 7. Real optimization run with Genetic Algorithm.

	Test 1	Test 2	Test 3	Test 4	Test 5
Run time (sec, average)	90.4907	80.77	81.64	85.36	74.61
Population size	60	60	60	60	60
OX rate	90%	0%	0%	10%	30%
PMX rate	0%	90%	0%	10%	30%
CX rate	0%	0%	90%	10%	30%
2-opt rate (mutation)	10%	10%	10%	50%	10%
Fitness value (average)	119,579	119,715	124,504	117,856	128,832

Since the objective is to minimize the fitness value, it can be read from Table 7 that Test 4 was the best, in addition, Tests 1 and 2 also provided high-quality results. The most efficient operator was 2-opt, but the PMX and OX operators also proved to be effective. Using the CX operator and setting equal sharing of each crossover has not been shown to be effective.

6. Conclusions

In this paper, we performed a random walk analysis of the basin of attraction map (which is one of the fitness landscape analyzation methods) for the Multi-Echelon Vehicle Routing Problem. Vehicle Routing Problems is a common optimization task in the field of logistics. The fitness landscape analysis investigates the following components: the set of possible states, the fitness function and the neighborhood. The set of possible states depends on the problem instance and the used operators. The paper presents detailed analysis of the following operators: 2-opt, order crossover, cycle crossover, partially matched crossover. Two investigation methods were used in the random walk analysis. In the first approach, we used the direct solution comparison. Here, we examined the average distances between the solutions, the distances taken from the best solution, and how many solutions have the same fitness value. The following distances were used: fitness distance, hamming distance, basic swap sequence distance. Another approach is the formal information analysis. The following methods were used in the information analysis approach: partial information content, expected partial information content, information stability, entropy, information content, density basin information, regularity, evolvability portrait. Based on the performed tests, the presented fitness landscape analysis method, the basin of attraction map, is suitable for the analysis of the efficiency of some optimization operators. Based on the measurements, the 2-opt and partially matched operators proved to be effective, and the order crossover and cycle crossover proved to be weak.

We can summarize the strengths of our approach in the followings. It provides an overview of the theoretical foundations and an abstract level analysis of the optimization task. It can be used to explore the search space for the transport task and to select the most efficient operators. The proposed methods can be used also for development and

implementation of new heuristic algorithms. Our further research direction is the extension of the analyzes to other heuristic operators.

Author Contributions: Conceptualization, L.K. and A.A.; methodology, L.K., A.A. and T.B.; software, A.A.; validation, L.K. and A.A.; formal analysis, L.K. and A.A.; investigation, L.K. and A.A.; resources, T.B.; data curation, L.K., A.A.; writing—original draft preparation, L.K., A.A. and T.B.; writing—review and editing, L.K., A.A. and T.B.; visualization, L.K., A.A. and T.B.; supervision, L.K. and T.B.; project administration, L.K. and T.B.; funding acquisition, T.B. All authors have read and agreed to the published version of the manuscript.

Funding: The described article was carried out as part of the EFOP-3.6.1-16-2016-00011 “Younger and Renewing University – Innovative Knowledge City—institutional development of the University of Miskolc aiming at intelligent specialisation” project implemented in the framework of the Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data cited in this manuscript are available from the published papers or a corresponding author.

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

References

- Dantzig, G.B.; Ramser, J.H. The truck dispatching problem. *Manag. Sci.* **1959**, *6*, 80–91. [\[CrossRef\]](#)
- Skrlec, D.; Filipec, M.; Krajcar, S. A heuristic modification of genetic algorithm used for solving the single depot capacitated vehicle routing problem. In Proceedings of the Intelligent Information Systems: IIS'97, Grand Bahama Island, Bahamas, 8–10 December 1997; pp. 184–188. [\[CrossRef\]](#)
- Nagy, G.; Salhi, S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *Eur. J. Oper. Res.* **2005**, *162*, 126–141. [\[CrossRef\]](#)
- Pisinger, D.; Ropke, S. A general heuristic for vehicle routing problems. *Comput. Oper. Res.* **2007**, *34*, 2403–2435. [\[CrossRef\]](#)
- Crainic, T.G.; Perboli, G.; Mancini, S.; Tadei, R. Two-echelon vehicle routing problem: A satellite location analysis. *Procedia-Soc. Behav. Sci.* **2010**, *2*, 5944–5955. [\[CrossRef\]](#)
- Ochi, L.S.; Vianna, D.S.; Drummond, L.M.; Victor, A. A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Future Gener. Comput. Syst.* **1998**, *14*, 285–292. [\[CrossRef\]](#)
- Yao, B.; Yu, B.; Hu, P.; Gao, J.; Zhang, M. An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot. *Ann. Oper. Res.* **2016**, *242*, 303–320. [\[CrossRef\]](#)
- Ralphs, T.K.; Kopman, L.; Pulleyblank, W.R.; Trotter, L.E. On the capacitated vehicle routing problem. *Math. Program.* **2003**, *94*, 343–359. [\[CrossRef\]](#)
- Sitek, P.; Wikarek, J.; Ruczyńska-Wdowiak, K.; Bocewicz, G.; Banaszak, Z. Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach. *Neurocomputing* **2021**, *423*, 670–678. [\[CrossRef\]](#)
- Poonthallir, G.; Nadarajan, R. A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). *Expert Syst. Appl.* **2018**, *100*, 131–144. [\[CrossRef\]](#)
- Kara, İ.; Kara, B.Y.; Yetiş, M.K. Cumulative vehicle routing problems. *Veh. Routing Probl.* **2008**, *1*, 85–98. [\[CrossRef\]](#)
- Bocewicz, G.; Banaszak, Z.; Rudnik, K.; Witzak, M.; Smutnicki, C.; Wikarek, J. Milk-run Routing and Scheduling Subject to Fuzzy Pickup and Delivery Time Constraints: An Ordered Fuzzy Numbers Approach. In Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK, 19–24 July 2020. [\[CrossRef\]](#)
- Grzegorz, B.; Izabela, N.; Arkadiusz, G.; Zbigniew, B. Reference model of milk-run traffic systems prototyping. *Int. J. Prod. Res.* **2020**, *1*–18. [\[CrossRef\]](#)
- Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Lemus-Zúñiga, L.G.; Cano, J.C.; Manzoni, P.; Gayraud, T. Centralized route-management solution for autonomous vehicles in urban areas. *Electronics* **2019**, *8*, 722. [\[CrossRef\]](#)
- Burduk, A.; Musiał, K. Optimization of Chosen Transport Task by Using Generic Algorithms. *Lect. Notes Comput. Sci.* **2016**, *9842*, 197–205. [\[CrossRef\]](#)
- Kłosowski, G.; Gola, A.; Amila, T. Computational Intelligence in Control of AGV Multimodal Systems. *Ifac-Pap.* **2018**, *51*, 1421–1427. [\[CrossRef\]](#)
- Pitzer, E. Applied Fitness Landscape Analysis. Ph.D. Dissertation, Johannes Kepler Universität Linz, Technisch-Naturwissenschaftliche Fakultät, Linz, Austria, 2013.

18. Pitzer, E.; Affenzeller, M. A comprehensive survey on fitness landscape analysis. In *Recent Advances in Intelligent Engineering Systems; Studies in Computational Intelligence*; Fodor, J., Klemmou, R., Suárez Araujo, C.P., Eds.; Springer: Berlin, Heidelberg, Germany, 2012; p. 378. [[CrossRef](#)]
19. Ventresca, M.; Ombuki-Berman, B.; Runka, A. Predicting genetic algorithm performance on the vehicle routing problem using information theoretic landscape measures. *Lect. Notes Comput. Sci.* **2013**, *7832*, 214–225. [[CrossRef](#)]
20. Czech, Z.J. Statistical measures of a fitness landscape for the vehicle routing problem. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing, Miami, FL, USA, 14–18 April 2008; pp. 1–8. [[CrossRef](#)]
21. Tayarani-N, M.H.; Prügél-Bennett, A. An analysis of the fitness landscape of travelling salesman problem. *Evol. Comput.* **2016**, *24*, 347–384. [[CrossRef](#)] [[PubMed](#)]
22. Fitness Landscapes and Graphs: Multimodularity, Ruggedness and Neutrality. Available online: <http://www-lisic.univ-littoral.fr/~jverel/talks/2tut16-verel.pdf> (accessed on 12 December 2020).
23. Fitness Landscapes of Combinatorial Problems and the Performance of Search Algorithms. Available online: <https://philippe-preux.github.io/papiers/lil-97-13.pdf> (accessed on 12 December 2020).
24. Uludağ, G.; Uyar, A.Ş. Fitness landscape analysis of differential evolution algorithms. In Proceedings of the 2009 Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, Famagusta, Cyprus, 2–4 September 2009; pp. 1–4. [[CrossRef](#)]
25. Chicano, F.; Daolio, F.; Ochoa, G.; Vérel, S.; Tomassini, M.; Alba, E. Local optima networks, landscape autocorrelation and heuristic search performance. *Lect. Notes Comput. Sci.* **2012**, *7492*, 337–347. [[CrossRef](#)]
26. Kovács, L.; Agárdi, A.; Bányai, T. Fitness Landscape Analysis and Edge Weighting-Based Optimization of Vehicle Routing Problems. *Processes* **2020**, *8*, 1363. [[CrossRef](#)]
27. Englert, M.; Röglin, H.; Vöcking, B. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. *Algorithmica* **2014**, *68*, 190–264. [[CrossRef](#)]
28. Hussain, A.; Muhammad, Y.S.; Sajid, M.N.; Hussain, I.; Shoukry, A.M.; Gani, S. Genetic algorithm for traveling salesman problem with modified cycle crossover operator. *Comput. Intell. Neurosci.* **2017**, 7430125. [[CrossRef](#)] [[PubMed](#)]
29. Zhu, K.Q. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, USA, 3–5 November 2003; pp. 176–183. [[CrossRef](#)]
30. Wang, K.P.; Huang, L.; Zhou, C.G.; Pang, W. Particle swarm optimization for traveling salesman problem. In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693), Xi'an, China, 5 November 2003; Volume 3, pp. 1583–1585. [[CrossRef](#)]