Acta Polytechnica 61(6):672–683, 2021
https://doi.org/10.14311/AP.2021.61.0672

COMPARISION OF THE WALK TECHNIQUES FOR FITNESS STATE SPACE ANALYSIS IN VEHICLE ROUTING PROBLEM

Anita Agárdi[a,*], László Kovács[a], Tamás Bányai[b]

[a] University of Miskolc, Faculty of Mechanical Engineering and Informatics, Institute of Informatics, Miskolc-Egyetemváros 3515, Miskolc, Hungary

[b] University of Miskolc, Faculty of Mechanical Engineering and Informatics, Institute of Logistics, Miskolc-Egyetemváros 3515, Miskolc, Hungary

[*] corresponding author: agardianita@iit.uni-miskolc.hu

Abstract. The Vehicle Routing Problem (VRP) is a highly researched discrete optimization task. The first article dealing with this problem was published by Dantzig and Ramster in 1959 under the name Truck Dispatching Problem. Since then, several versions of VRP have been developed. The task is NP difficult, it can be solved only in the foreseeable future, relying on different heuristic algorithms. The geometrical property of the state space influences the efficiency of the optimization method. In this paper, we present an analysis of the following state space methods: adaptive, reverse adaptive and uphill-downhill walk. In our paper, the efficiency of four operators are analysed on a complex Vehicle Routing Problem. These operators are the 2-opt, Partially Matched Crossover, Cycle Crossover and Order Crossover. Based on the test results, the 2-opt and Partially Matched Crossover are superior to the other two methods.

Keywords: Fitness state space, Vehicle Routing Problem, optimization.

1. Introduction

The Vehicle Routing Problem (VRP) is one of the best known discrete optimization tasks. During the basic VRP, the positions and demands of the customers are given in advance, and the number of vehicles and capacity limit are also known in advance. Starting from the depot, vehicles visit customers and then return to the depot. The objective of the optimization is the minimization of the distance travelled by the vehicles. Figure 1 illustrates the layout of a basic Vehicle Routing Problem.

The VRP was first introduced by Dantzig and Ramster [1] as Truck Dispatching Problem. Since then, many types and constraints of the problem have emerged as the complexity of transportation tasks has begun to increase. Below, we will introduce some types of problems. Table 1 contains the most researched type of Vehicle Routing Problem types.

In the fitness state space analysis, the key elements are the representation form of the state and the applied neighbourhood operator. The analysis can be used to show the efficiency of the selected operators of the optimisation algorithm. Below, we present several papers that investigate fitness state space analysis techniques.

Fitness state space analysis was first published by Sewall Wright [14]. He described the search space mathematically as a pair containing a set of solutions and fitness functions.

Since then, fitness state space analysis techniques have been applied to many problems, for example, Knapsack Problem [15], Timetabling Problem [16], Traveling Salesman Problem [17, 18], Flowshop Scheduling Problem [19], Quadratic Assignment Problem [20], Maximal Constraint Satisfaction Problem [21], Vehicle Routing Problem [22].

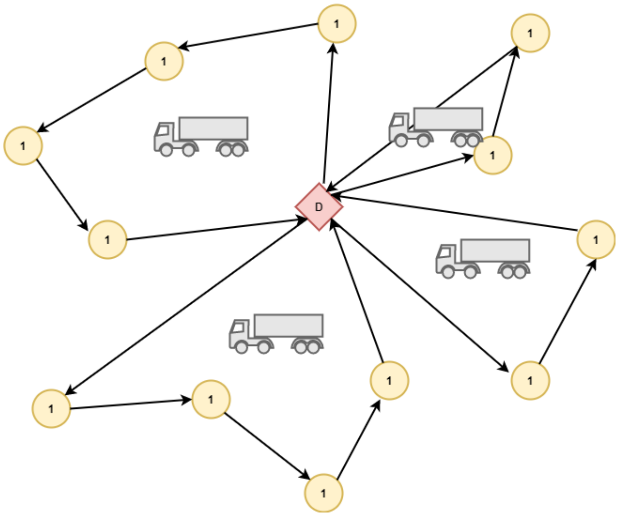

Figure 1. Example of a basic Vehicle Routing Problem (VRP)

The work [15] presents the analysis of several representation methods, such as Weight-Coding Representation, Random-Key Representation, Ordinal Representation, Binary Representation, Permutation Representation. The analysis includes the following metrics: fitness distance correlation, autocorrelation function, correlation length.

In [23], the fitness state space of the Optimum Multiuser Detection Problem is also analysed using the

Vehicle Routing Problem Type	Explanation
Time Window [2]	Customers must be visited during a specific time interval.
Multiple Time Window [3]	Customers have one or more time windows.
Soft Time Window [4]	Customers can be visited outside of the time interval (but the solution gets a penalty point).
Single Depot [5]	The system contains a single depot.
Multiple Depot [6]	The system contains multiple depots, the vehicles must start their route from one of the depots
Open Route [7]	Vehicles do not have to arrive to the depot
Inter-Depot Routes [8]	Vehicles can arrive to any of the depots
Two-Echelon [9]	The system contains intermediate locations (called satellites). The products will be shipped in the following route: depot-satellite-customer
Homogeneous Fleet [10]	The system contains one type of vehicles
Heterogeneous Fleet [10]	The system contains multiple types of vehicles
Pickup and Delivery [11]	Delivering products from the depot to the customers and collecting products from the customers to the depots.
Multiple Product [12]	The system contains multiple products
Periodic Problem [13]	Periodic visits to customers

Table 1. VRP types

following fitness state space methods: Fitness Distance Correlation, Random Walk Correlation Function.

The Timetabling Problem landscape is determined by the following techniques: fitness distance correlation, autocorrelation [16].

We can also find some examples of a fitness state space analysis of the Traveling Salesman Problem, e. g., [17, 18], where the k-opt operator is analysed by calculating the Hamming distance. The following analyses are performed by the authors: distance to global optimum, number of local optimums, probability of visiting an optimum, autocorrelation, time to local optimum, the distance between the optima. The effectiveness of Edge Recombination and Maximal Preservative Crossover (MPX) are compared.

An analysis of the Flowshop Scheduling Problem (FSP) fitness state space has also been performed, for example, by [19]. In the article, the following analytical methods are detailed: evolvability, neutral degree, neutral neighbour, autocorrelation.

An analysis of the fitness state space of the Quadratic Assignment Problem has also been investigated, for example, by the authors of [20]. The following operators are analysed: pairwise exchange, partially mapped crossover (PMX), and swap. The following fitness state space analysis techniques are performed: autocorrelation function, autocorrelation length, random walk, the basin of attraction.

In [21], the Maximal Constraint Satisfaction Problem is investigated with a random-walk and cost-density analysis. The analysis of the fitness state space of the Traveling Salesman Problem was also presented by the authors of [24]. They demonstrated the efficiencies of 2-opt (edge swap) and city swap operators. The distance from the global optimum was examined by the authors as a function of a fitness value. According to the authors, the 2-opt operator is more efficient, because in this case, the Fitness Distance Correlation is strong.

The time to the local optimum, distances from optima, autocorrelation, number of local optima, probability of getting to the optimum analysis of the Traveling Salesman Problem and the No-Wait Job Scheduling task was also detailed by the authors of [17].

The fitness state space analysis of the Quadratic Assignment Problem and Traveling Salesman Problem [25] were performed with the following techniques: fitness distance correlation, fitness cloud, autocorrelation, correlation length, information content, information stability, ruggedness, autocorrelation, correlation length. The following walk techniques are presented: random walk, adaptive walk, reverse adaptive walk, uphill-downhill walk, neutral walk.

We can find an example of an analysis of the Vehicle Routing Problem fitness state space [22], where the following operators were attempted to be analysed: Swap, Inversion, Insertion, Partially Matched Crossover, Uniform Order Crossover, Cycle Crossover, Displacement. Random walk technique with 2000 steps taken was used. The task was also examined from an information elemental point of view, with the following techniques: information content, density-basin information and partial information content.

In our earlier research, we have already investigated the fitness state space of a Multi-Echelon system [26, 27]. We performed a fitness cloud and random walk analyses of the Multi-Echelon Vehicle Routing Problem. Now, we turn to the extension of the problem, namely the investigation of the adaptive, reverse adaptive and uphill-downhill walk. The remainder of this paper is organized as follows. Section 2 contains the concept of our fitness state space analysis, such as the adaptive, reverse adaptive and uphill-downhill walks, the applied neighbourhood operators and the applied distance calculations. Section 3 contains the results and discussion. This section presents our Vehicle Routing Problem, the results of the adaptive, reverse adaptive and uphill-downhill walks, and the summary results. The last section presents conclusions and the scope of our future work.

2. Fitness state space analysis

Optimisation metaheuristics operate iteratively to find the extremum points in the object space. The body of the iteration contains the following elements [28]:

- The set of possible states.
- Neighbourhood, which is defined by an operator. From the current state with the application of a neighbourhood operator, we can get the next state.
- Objective function.
- Representation: The applicability of each operator depends on the representation.
- Transition rule: selection of the next state point from the neighbours.
- Termination criteria of the algorithm.
- Whether the initial state point is generated randomly or with some heuristics.

2.1. Adaptive, reverse adaptive and uphill-downhill walk

Analytical methods can be divided into two categories, exhaustive search and stochastic, sampling-based techniques [29].

The exhaustive search analysis gives the most complete picture of the search space. But they can only be applied to smaller problems (not suitable for larger problems), hence, it is difficult to apply this analysis in practice. The main advantage of an exhaustive search is the completeness. The search time (runtime) of the exhaustive search is high, but the search space analysis is completely traversed [29].

The most commonly used techniques in the search space analysis are the stochastic, sampling-based techniques. Their advantage over the exhaustive search is that their running time is low. However, one of the disadvantages of these methods is that we cannot get a complete systematic view of the search space. It is comparable to metaheuristics in the sense that metaheuristics search for a “relatively good” solution, while the stochastic search offers a “relatively good” mapping [29].

The basic problem of stochastic analysis methods is the selection of a representative sample. The selection of the optimal sample is a difficult task as we do not know the complete set of the solution. So far, two types of sample generation techniques have been used [29]:

- The trajectory-based sampling technique creates the path of optimization methods with continuous solution candidates.
- Discovery strategies generates scattered samples.

Sampled trajectories create a path in the search space or, in other words, a sequence of adjacent state points (solution candidates). This method is also called walk [29]. There are several walks to analyse the search space. The walk starts from a random solution candidate and uses the neighbourhood search to "walk" to neighbouring solution candidates. Depending on the type of the neighbourhood search, the following walking techniques are possible [25]:

- Random walk – a state point is randomly selected from a set of neighbours.
- Adaptive walk – the better state point (neighbour) is selected.
- Reverse adaptive walk – always selects the worst neighbour. This is the reverse of the adaptive walk.
- Uphill-downhill walk – first, an adaptive walk is performed, then, if a better fitness point (solution) is not found during the step, a reverse adaptive walk is performed until a better state point cannot be found.
- Neutral walk – a neighbour with a fitness value equal to the parent’s fitness value (with the effort to increase the distance from the starting solution) is chosen.

2.2. Neighbourhood operators

The optimization algorithm is greatly influenced by the applied search operators. In this article, the efficiency of the following operators are presented: 2-opt, Order Crossover (OX), Cycle Crossover (CX), Partially Matched Crossover (PMX).

In the case of 2-opt [30], two edges are swapped. This is the reversal of the elements of a section in the permutation. The two edges are selected randomly. Figure 2 shows a sequence. The problem is in a permutation space; an element is given with a permutation.

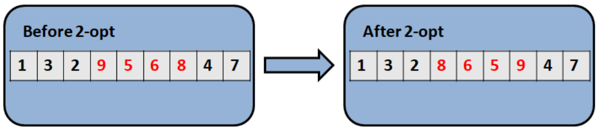

Figure 2. Example of the 2-opt operator

The Order Crossover (OX) [31] operator creates two children solutions from two parent solutions. When using the operator, we need to look for a fitting section.

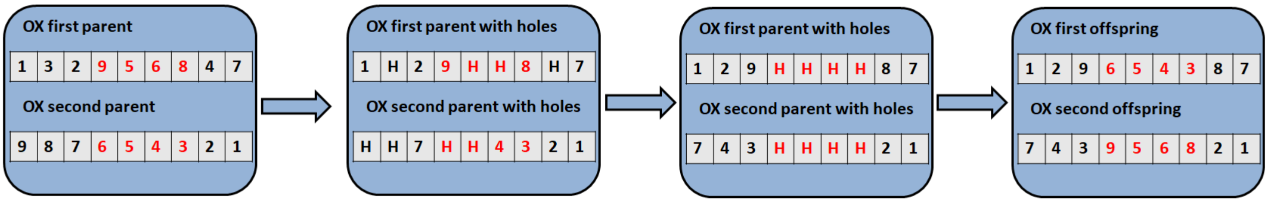

Figure 3. Example of the Order Crossover (OX) operator

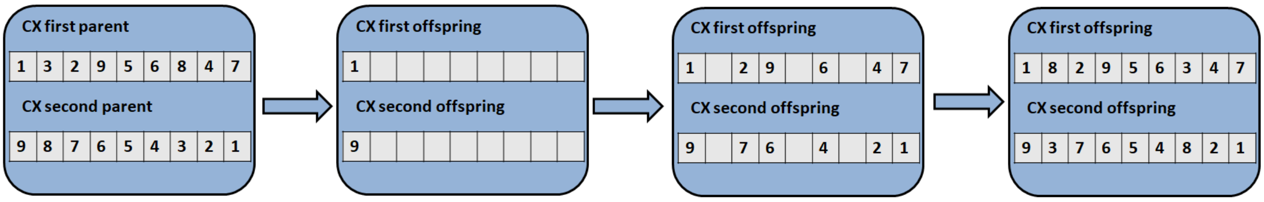

Figure 4. Example of the Cycle Crossover (CX) operator

The fitting section is selected randomly. The children will initially be the copies of the parents. Then, the elements of the fitting section of the second parent are deleted from the first child (letter H) and the elements of the fitting section of the first parent are deleted from the second child. Then, the child solutions are rearranged so that the letters H (empty spaces) are in the matching phase of the permutations. Then, the first child receives the fitting section of the second parent and the second child receives the fitting section of the first parent. Figure 3 illustrates the operations.

Cycle Crossover (CX) [32] searches for cycles. This operator also creates two child solutions from two parent solutions. Initially, the first element of the first child should be the first element of the first parent, and the first element of the second child should be the first element of the second parent. In the example, this is the pair (1,9), so the first element of the first child is 1 and the first element of the second child is 9. Then, in the example, comes the pair (9,6), they are placed in position 4, the pair (6,4) in position 6, (4,2) in position 8, (2,7) in position 3, and (7,1) in the last position. Then, (1,9) would come, but this pair has already been placed, so the cycle is closed. There are still gaps in the child solutions, they are filled so that the first child gets the right items from the second parent and the second child gets the right items from the first parent. The operator is shown in Figure 4.

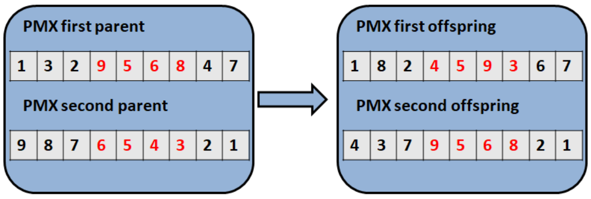

Figure 5. Example of the Partially Matched Crossover (PMX) operator

Partially Matched Crossover (PMX) [33] also creates two-child solutions from a two-parent solutions. This operator also selects the fitting section randomly. First, the child solutions will be the copies of the parents. Pairs from the fitting sections of the parents are then paired. In Figure 5, the pairs are the following: (9,6), (5,5), (6,4), and (8,3). These items are swapped in both children.

2.3. Distance calculation

We used three types of distances to describe the similarity between two solutions, these are the fitness [34], Hamming [35] and basic swap sequence [36]. The fitness distance between the two solutions is illustrated in Figure 6. In the example, the distance is 500 because the absolute value of the fitness value of the two solutions must be taken. The Hamming distance is 4, because they differ in 4 positions. The basic swap sequence distance is 1, because with a single edge swap operation, we get the other solution from one solution (the basic swap sequence distance is the minimum of the edge swap numbers).

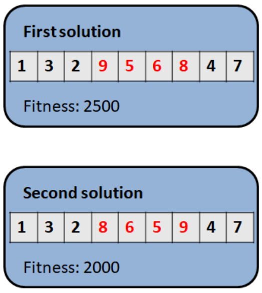

Figure 6. Distance example

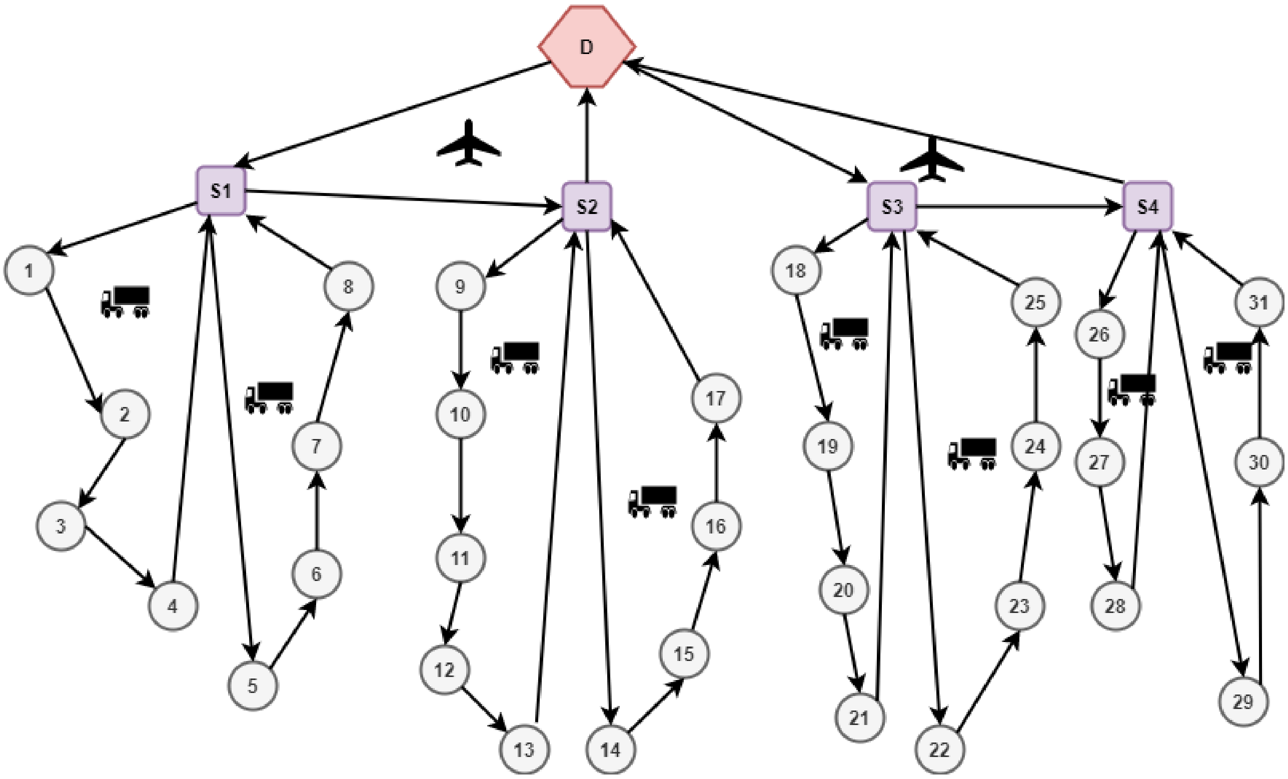

Figure 7. The investigated Vehicle Routing Problem

3. Results and Discussion

3.1. Our Vehicle Routing System

Our Vehicle Routing Problem (which is also presented in our previous works [26, 27]) is illustrated in Figure 7. The test system contains a single depot, satellites and customers. The x and y position coordinates of the depot are generated in the interval (0–100). For the first level satellites, the coordinates are generated in the interval (200–300) and the coordinates of second level satellites are in the interval (400–500). The system contains 10 first-level and 10 second-level satellites and 15 customers. The products are transported from the depots to the satellites, and then, from the satellites to the customers. The demands for the products vary by customers, and the system contains a single type of product. The system contains different types of vehicles. The vehicles differ in the capacity limit and fuel consumption The values of the capacity limits are generated in the interval (10,000–50,000), and the fuel consumption in the interval (10–100). The travel time between the customers and the route status are also important factors. Additionally, the following costs are also considered: loading and unloading time, administration time, loading and unloading cost, administrative and quality control cost. The values of these componens are generated in the interval (30–50).

The objective function of the optimization is a compound cost function having the following components and objectives: fuel consumption (minimization), route time (minimization), unvisited customers (minimization), and route status (minimization).

Our goal was to select a complex VRP system having several layers and several vehicle types with different depots for the analysis. This kind of architecture is very frequent in real life applications.

3.2. Adaptive walk

In this subsection, the results of the adaptive walk are analysed. According to Figure 8, fitness values range from 120,000 to 150,000 for 2-opt, and the average fitness distances taken from solutions range from 4,000 to 21,000. Here, we obtain a parabolic function.

The Figure 9 represents the average of the Hamming distances for 2-opt. Here, the Hamming distances are between 20–34. The same can be said for the basic swap sequence (here the distances are between 13–26). We also get a parabola-like function for fitness distances in the case of order crossover (here the fitness values are between 110,000–140,000, while the averages of the fitness differences range from 2,000–17,000), while the Hamming (here the averages of the differences are between 16–34) and basic swap sequence distances (where the averages of the distances are between 12–26) also depend on the fitness value of the solution. The cycle crossover is similar to 2-opt, here, the fitness values are between 110,000–130,000, while the averages of the distances for a fitness distance are between 2,000–15,000, while the basic swap sequence distances are in the range of 8–24. For a partially matched crossover, fitness values are between 110,000–140,000 and the average fitness differences are between 2,000–21,000. Hamming distances are between 22–36 and also depend on the fitness value. The higher the

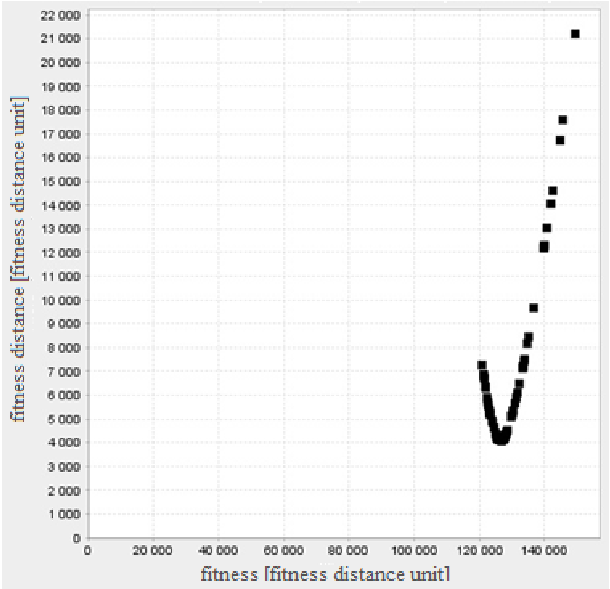

Figure 8. The relationship between fitness values and average of fitness distances to other solutions in the case of 2-opt operator

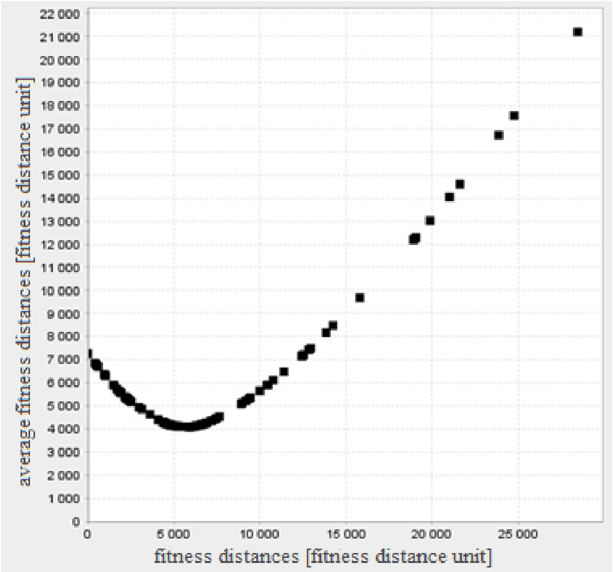

Figure 10. The relationship between the fitness distances of the best solution and the average of fitness distances to other solutions in the case of 2-opt operator

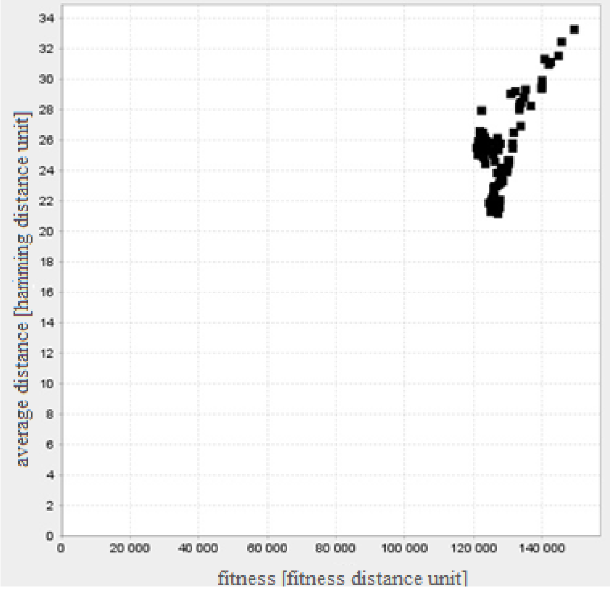

Figure 9. The relationship between fitness values and average of Hamming distances to other solutions in the case of 2-opt operator

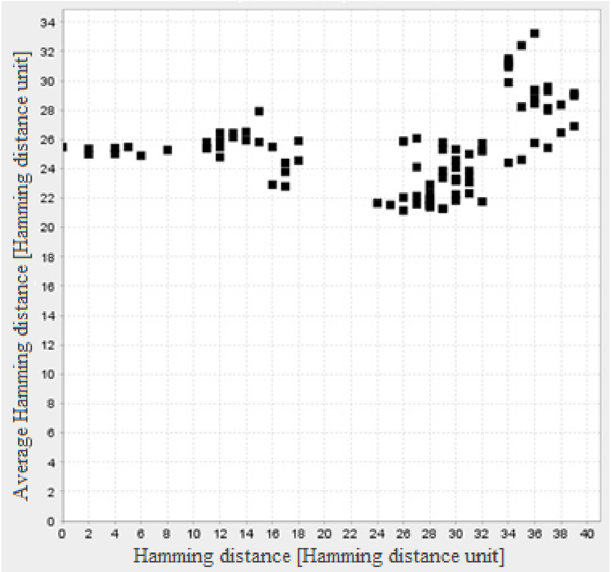

Figure 11. The relationship between the Hamming distances of the best solution and the average of Hamming distances to other solutions in the case of 2-opt operator

fitness value, the higher the averages of the Hamming distances. And we also got this result in the case of the basic swap sequence, here, the values are between 16–28.

In the case of the distance from the best solution for an adaptive walk the higher the fitness value, the greater the distances from the best solution for all operators.

The fitness distance from the best solution and the average of the fitness distances of the other solutions (Figure 10) describe a parabolic function. If the distance from the best solution is small or large, then the average distance from the other solutions is large when using a 2-opt operator.

There are no similarities between the Hamming and basic swap sequence distances (Figure 11), the distance of the best solution does not affect the average of the distances taken from the other solutions. In the case of the order crossover, if the fitness distances from the best solution are too large, the averages of the distances will also be very large, and this is also true for the Hamming and basic swap sequence distances. In the case of a cycle crossover, the averages of fitness distances are also high when the fitness

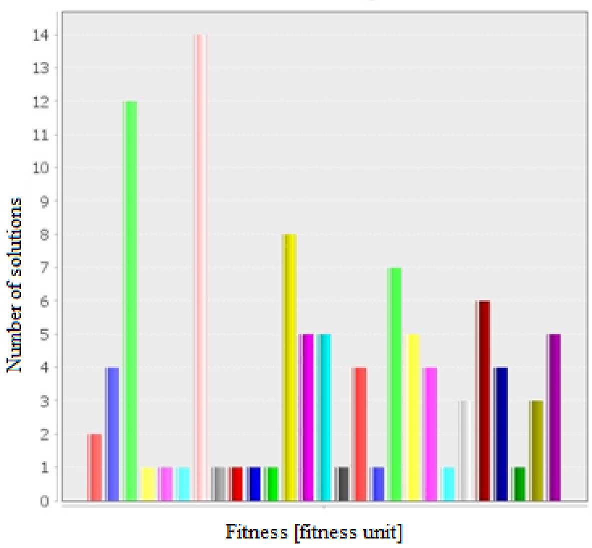

Figure 12. Cost of density in the case of 2-opt operator

distance taken from the best solution is high. The same is true for the Hamming and basic swap sequence distances. In the case of a partially matched crossover, both fitness and Hamming and basic swap sequence distances taken from the best solution also greatly affects the average distances.

In the case of 2-opt, each solution has a different fitness value, but in the case of order crossover and cycle crossover (Figure 12), some solutions have the same fitness value. In the case of a partially matched crossover, almost every solution has a unique fitness value. The summary of the adaptive walk for each operator is illustrated in the Table 2.

3.3. Reverse adaptive walk

In the following, the reverse adaptive walk solutions are analysed. The averages of fitness distances describe a parabolic function, Hamming and basic swap sequence averages decrease with increasing fitness value. Order crossover fitness distances also describe a parabolic function while Hamming and basic swap sequence distances do not depend on fitness values. The situation is similar in the case of the cycle and partially matched crossover, the averages of the difference in fitness values can be described by a parabolic function, and the distances between Hamming and basic swap sequences do not depend on the fitness value, they are located close to each other. The fitness distance from the best solution in the case of 2-opt, order crossover, cycle crossover and partially matched crossover also describes a linear function. Hamming and basic swap sequence distances occur in a large interval, from a small distance of 1–2 distances to a very large distance of 30–40 between solutions. The function of the distance from the best solution and the averages of the distances from the other solutions is as follows: the fitness distance for 2-opt is a parabolic function, but shows a prolonged, long-decreasing trend and then starts to increase slightly by the end of the scale. The Hamming and basic swap sequence distances show wider average distances for larger distances. In the case of the order crossover, the fitness distances also describe a parabolic function, only in this case, a larger increase can be observed after a small decrease. For Hamming and basic swap sequence distances, the scale of the average distances increases as the distances from the best solution increase. In the case of the cycle and partially matched crossover, we also get a parabolic function for the fitness distance, and the Hamming and basic swap sequence distances increase with increasing the distance from the best solution, and the scale of the averages of the distances also increases. The cost of density values is low, which means that the fitness values of the solutions are different during the steps. 1–2 is the cost of density for 2-opt, 1–4 for order crossover. The cycle crossover, however, provided several solutions with the same fitness value, here 1–5 is the number of identical fitness values. For the partially matched crossover, this value is 1–3, but in this case, almost all solutions have different fitness values. The Table 3 illustrates the results for a reverse adaptive walk.

3.4. Uphill-Downhill Walk

During the uphill-downhill walk, the fitness values range from 130,000 to 150,000, while the average fitness distances are between 2,500 and 9,000 in the case of 2-opt operator. This function is parabolic because small and large fitness values have a large average fitness distance, while medium fitness values have a small average distance. The averages of Hamming distances do not depend on the fitness values. They move in a relatively small interval, and the same can be said for the basic swap sequence distance. In the case of an order crossover, the fitness values of the solutions range from 120,000 to 140,000. The averages of the fitness distances taken from the solutions are large for small and large fitness values, while they are small for average fitness values, so their function is parabolic. The Hamming and basic swap sequence distances are condensed into a single point; the averages of the distances do not depend on the fitness values. Cycle crossover values are also similar to 2-opt and order crossover values, here, fitness values range from 120,000 to 140,000, fitness distances average between 2,000 and 8,500, Hamming distances range from 28 to 34, and basic swap sequence distances range from 23 to 26. Partially matched crossover results are also similar to those of the other three operators.

The distances are taken from the best solution for a 2-opt operator increase, as a function of fitness distance, while Hamming and basic swap sequence distances, however, show no correlation with fitness values, these distances differ greatly. For order crossover, fitness distances also increase linearly as a function of fitness value, and Hamming and basic swap sequence

Type	Distance	Lower bound	Upper bound
2-opt			
Fitness values	Fitness	120,000	150,000
Average of fitness distances	Fitness	4,000	21.000
Average of hamming distances	Hamming	20	34
Average of basic swap sequence distances	Basic swap sequence	13	26
Fitness distances of best solution	Fitness	0	28,000
Hamming distances of best solution	Hamming	2	39
Basic swap sequence distances of best solution	Basic swap sequence	1	30
Cost of density	–	1	3
Order Crossover			
Fitness values	Fitness	110,000	140,000
Average of fitness distances	Fitness	2,000	17,000
Average of hamming distances	Hamming	16	34
Average of basic swap sequence distances	Basic swap sequence	12	26
Fitness distances of best solution	Fitness	1,000	21,000
Hamming distances of best solution	Hamming	2	32
Basic swap sequence distances of best solution	Basic swap sequence	1	28
Cost of density	–	1	7
Cycle Crossover			
Fitness values	Fitness	110,000	130,000
Average of fitness distances	Fitness	2,000	15,000
Average of hamming distances	Hamming	12	30
Average of basic swap sequence distances	Basic swap sequence	8	24
Fitness distances of best solution	Fitness	0	18,000
Hamming distances of best solution	Hamming	2	36
Basic swap sequence distances of best solution	Basic swap sequence	1	28
Cost of density	–	1	14
Partially Matched Crossover			
Fitness values	Fitness	110,000	140,000
Average of fitness distances	Fitness	2,000	21,000
Average of hamming distances	Hamming	22	36
Average of basic swap sequence distances	Basic swap sequence	16	28
Fitness distances of best solution	Fitness	0	24.000
Hamming distances of best solution	Hamming	4	40
Basic swap sequence distances of best solution	Basic swap sequence	2	34
Cost of density	–	1	9

Table 2. Adaptive walk results

distances do not correlate with fitness values, and these distances are also in a large interval here. The results of the cycle crossover are similar to the results of 2-opt and order crossover, the fitness distances here are between 500–16,000, the Hamming distances are between 2–42 and basic swap sequence distances range from 1–34. The situation is similar in the case of a partially matched crossover, here, the fitness distances are between 0–14,000, the Hamming distances are between 6–40, and the basic swap sequence distances are between 4–32.

The cost density values move at a small interval during each operator, which means that the fitness values are different for each solution. The cost density value for 2-opt is 1–2, 1–3 for order crossover, 1–3 for cycle crossover, and in the case of partially matched crossover, 1–3. Table 4 illustrates the results for the uphill-downhill walk.

3.5. Summary Analysis

Based on the test results, we can summarize our experiments based on the following. For the walk techniques presented above, the fitness values are good in the following case: the difference between the lower and the upper boundaries is great. The lower limit should be small because the objective is to minimize the fitness value. The average of fitness, Hamming and basic swap sequence distances is good in the following case: the greater the distances, the better the algorithm maps the search space. The evaluation of the fitness, Hamming and basic swap sequence distances of the best solution is as follows: when the lower bound is small and the upper bound is large, then the operator

Type	Distance	Lower bound	Upper bound
2-opt			
Fitness values	Fitness	120,000	150,000
Average of fitness distances	Fitness	3,000	16,000
Average of hamming distances	Hamming	24	34
Average of basic swap sequence distances	Basic swap sequence	17	25
Fitness distances of best solution	Fitness	500	21,000
Hamming distances of best solution	Hamming	2	40
Basic swap sequence distances of best solution	Basic swap sequence	1	30
Cost of density	–	1	2
Order Crossover			
Fitness values	Fitness	130,000	140,000
Average of fitness distances	Fitness	800	4,000
Average of hamming distances	Hamming	28	34
Average of basic swap sequence distances	Basic swap sequence	22	28
Fitness distances of best solution	Fitness	500	6,000
Hamming distances of best solution	Hamming	6	40
Basic swap sequence distances of best solution	Basic swap sequence	6	34
Cost of density	–	1	4
Cycle Crossover			
Fitness values	Fitness	130,000	140,000
Average of fitness distances	Fitness	1,200	3,800
Average of hamming distances	Hamming	28	34
Average of basic swap sequence distances	Basic swap sequence	22	27
Fitness distances of best solution	Fitness	0	7,000
Hamming distances of best solution	Hamming	6	40
Basic swap sequence distances of best solution	Basic swap sequence	5	32
Cost of density	–	1	5
Partially Matched Crossover			
Fitness values	Fitness	130,000	150,000
Average of fitness distances	Fitness	1,500	7,000
Average of hamming distances	Hamming	29	35
Average of basic swap sequence distances	Basic swap sequence	22	28
Fitness distances of best solution	Fitness	3,000	11,000
Hamming distances of best solution	Hamming	4	40
Basic swap sequence distances of best solution	Basic swap sequence	2	34
Cost of density	–	1	3

Table 3. The results for reverse adaptive walk

explores the space well (large upper bound), but it also found a very good solution because of the small lower bound. The evaluation of the cost density is as follows: for many small values, the operator maps the space well. The objective is to create many different fitness value solutions.

Table 5 shows that the 2-opt operator is efficient, and Cycle Crossover has a weak performance.

In the case of an adaptive walk and in the case of a cycle crossover, we find the smallest average fitness distance. The average Hamming and basic swap sequence distances are also the smallest here. The cost of density value became the highest in the case of the cycle crossover, which means that we got several solutions with the same fitness value during the walk. According to the order crossover cost of density diagram, several solutions also have the same fitness value. The cost of density values is the lowest for 2-opt, so this operator produced the highest number of different solutions. Since the objective is to map the search space as well as possible, the 2-opt operator proved to be the best in this measurement experiment.

In the reverse adaptive walk analysis, we find the greatest distance between the solutions of the 2-opt operator. The Hamming and basic swap sequence distances are approximately the same for each operator. The cost of density values are low for all operators, but the lowest for 2-opt. In the case of a cycle crossover, several solutions also have the same fitness value.

The results of the uphill-downhill walk are nearly identical for each operator. The cost density values are small for all operators. During the neutral walk, we got the highest value of the average distances of the fitness values in the case of the partially matched

Type	**Distance**	**Lower bound**	**Upper bound**
2-opt			
Fitness values	Fitness	130,000	150,000
Average of fitness distances	Fitness	2,500	9,000
Average of hamming distances	Hamming	26	34
Average of basic swap sequence distances	Basic swap sequence	19	26
Fitness distances of the best solution	Fitness	1,500	15,000
Hamming distances of the best solution	Hamming	4	36
Basic swap sequence distances of the best solution	Basic swap sequence	2	30
Cost density	–	1	2
Order Crossover			
Fitness values	Fitness	120,000	140,000
Average of fitness distances	Fitness	3,000	9,500
Average of hamming distances	Hamming	30	36
Average of basic swap sequence distances	Basic swap sequence	23	28
Fitness distances of the best solution	Fitness	1,000	18,000
Hamming distances of the best solution	Hamming	2	38
Basic swap sequence distances of the best solution	Basic swap sequence	4	32
Cost density	–	1	3
Cycle Crossover			
Fitness values	Fitness	120,000	140,000
Average of fitness distances	Fitness	2,000	8,500
Average of hamming distances	Hamming	28	34
Average of basic swap sequence distances	Basic swap sequence	23	26
Fitness distances of the best solution	Fitness	500	16,000
Hamming distances of the best solution	Hamming	2	42
Basic swap sequence distances of the best solution	Basic swap sequence	1	34
Cost density	–	1	3
Partially Matched Crossover			
Fitness values	Fitness	120,000	130,000
Average of fitness distances	Fitness	2,000	8,000
Average of hamming distances	Hamming	32	36
Average of basic swap sequence distances	Basic swap sequence	24	28
Fitness distances of the best solution	Fitness	0	14,000
Hamming distances of the best solution	Hamming	6	40
Basic swap sequence distances of the best solution	Basic swap sequence	4	32
Cost density	–	1	3

Table 4. Uphill-downhill walk

	Effective	**Weak**
Adaptive walk	2-opt	CX, OX
Reverse adaptive walk	2-opt	CX
Uphill-downhill walk		

Table 5. Summary

crossover. Average Hamming and basic swap sequence distances were the greatest for the partially matched crossover operator. This means that this operator has created the highest number of different solutions. Order crossover distance values are also high but the cycle crossover and 2-opt fitness distances are low. Based on the cost density diagrams, the cycle crossover solutions are the most unchanged, but the order crossover solutions also do not change much. The 2-opt and partially matched crossover solutions vary greatly, with almost every solution having a unique fitness value.

The practical significance of the presented fitness state space lies in the selection of the appropriate optimization operators. We can assume that the search space topology of transportation problems of a similar architecture is also similar. Based on that assumption, our findings on optimal search space operators can be generalized for VRP problems of the same type.

4. Conclusions and future work

In this paper, we have presented the fitness state space analysis of a complex Vehicle Routing Problem. In our fitness state space analysis, we examined the efficiencies of four operators: 2-opt, Cycle Crossover, Order Crossover, Partially Matched Crossover. The adaptive walk, reverse adaptive walk and uphill-downhill walk techniques were used as the analytical method.

The adaptive walk receives a neighbour that is closest to the previous solution during each iteration. The reverse adaptive walk is just the opposite. During the uphill-downhill walk, the adaptive and reverse adaptive steps alternate. The walk results were analysed using fitness values, average of fitness distances, average of Hamming distances, average of basic swap sequence distances, fitness distances of the best solution, Hamming distances of the best solution, basic swap sequence distances of the best solution and cost density. Based on the test results, the 2-opt operator proved to be much more efficient than the other operators. Our further research is the investigation of other operators, such as ER, Edge-2, Edge-3, MPX, RAR, GNX operators. In addition, we plan to analyse the search space of a discrete optimization problem, which is permutation-based, such as scheduling tasks (Parallel Machines Scheduling, Job Shop, Flow Shop, etc.).

References

[1] G. B. Dantzig, J. H. Ramser. The truck dispatching problem. *Management science* **6**(1):80–91, 1959. https://doi.org/10.1287/mnsc.6.1.80.

[2] Y. Shi, Y. Zhou, T. Boudouh, O. Grunder. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup – delivery and time window. *Engineering Applications of Artificial Intelligence* **95**:103901, 2020. https://doi.org/10.1016/j.engappai.2020.103901.

[3] M. Hoogeboom, W. Dullaert, D. Lai, D. Vigo. Efficient neighborhood evaluations for the vehicle routing problem with multiple time windows. *Transportation Science* **54**(2):400–416, 2020. https://doi.org/10.1287/trsc.2019.0912.

[4] G. Li, J. Li. An improved tabu search algorithm for the stochastic vehicle routing problem with soft time windows. *IEEE Access* **8**:158115–158124, 2020. https://doi.org/10.1109/access.2020.3020093.

[5] G. Nagy, S. Salhi. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European journal of operational research* **162**(1):126–141, 2005. https://doi.org/10.1016/j.ejor.2002.11.003.

[6] P. Stodola. Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem. *Natural Computing* **19**:463–475, 2020. https://doi.org/10.1007/s11047-020-09783-6.

[7] J. Brandão. A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research* **284**(2):559–571, 2020. https://doi.org/10.1016/j.ejor.2020.01.008.

[8] T. R. P. Ramos, M. I. Gomes, A. P. Barbosa-Povoa. A new matheuristic approach for the multi-depot vehicle routing problem with inter-depot routes. *OR Spectrum* **42**(1):75–110, 2020. https://doi.org/10.1007/s00291-019-00568-7.

[9] P. Kitjacharoenchai, B.-C. Min, S. Lee. Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics* **225**:107598, 2020. https://doi.org/10.1016/j.ijpe.2019.107598.

[10] R. F. Fachini, V. A. Armentano. Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering* **148**:106641, 2020. https://doi.org/10.1016/j.cie.2020.106641.

[11] L. do C. Martins, P. Hirsch, A. A. Juan. Agile optimization of a two-echelon vehicle routing problem with pickup and delivery. *International Transactions in Operational Research* 2020. https://doi.org/10.1111/itor.12796.

[12] S. M. Nugroho, L. Nafisah, M. S. A. Khannan, et al. Vehicle routing problem with heterogeneous fleet, split delivery, multiple product, multiple trip, and time windows: A case study in fuel distribution. In *IOP Conference Series: Materials Science and Engineering*, vol. 847, p. 012066. IOP Publishing, 2020. https://doi.org/10.1088/1757-899x/847/1/012066.

[13] Y. Wang, L. Wang, G. Chen, et al. An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. *Swarm and Evolutionary Computation* **55**:100675, 2020. https://doi.org/10.1016/j.swevo.2020.100675.

[14] S. Wright. *The roles of mutation, inbreeding, crossbreeding, and selection in evolution*, vol. 1. 1932.

[15] J. Tavares, F. B. Pereira, E. Costa. Multidimensional knapsack problem: A fitness landscape analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**(3):604–616, 2008. https://doi.org/10.1109/tsmcb.2008.915539.

[16] G. Ochoa, R. Qu, E. K. Burke. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 341–348. 2009. https://doi.org/10.1145/1569901.1569949.

[17] M.-H. Tayarani-N., A. Prügel-Bennett. An analysis of the fitness landscape of travelling salesman problem. *Evolutionary computation* **24**(2):347–384, 2016. https://doi.org/10.1162/evco_a_00154.

[18] K. Mathias, D. Whitley. Genetic operators, the fitness landscape and the traveling salesman problem. In *PPSN*, pp. 221–230. 1992.

[19] M.-E. Marmion, C. Dhaenens, L. Jourdan, et al. On the neutrality of flowshop scheduling fitness landscapes. In *International Conference on Learning and Intelligent Optimization*, pp. 238–252. Springer, 2011. https://doi.org/10.1007/978-3-642-25566-3_18.

[20] F. Chicano, F. Daolio, G. Ochoa, et al. Local optima networks, landscape autocorrelation and heuristic search performance. In *International Conference on Parallel Problem Solving from Nature*, pp. 337–347. Springer, 2012. https://doi.org/10.1007/978-3-642-32964-7_34.

[21] M. Belaidouni, J.-K. Hao. An analysis of the configuration space of the maximal constraint satisfaction problem. In *International Conference on Parallel Problem Solving from Nature*, pp. 49–58. Springer, 2000. https://doi.org/10.1007/3-540-45356-3_5.

[22] M. Ventresca, B. Ombuki-Berman, A. Runka. Predicting genetic algorithm performance on the vehicle routing problem using information theoretic landscape measures. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 214–225. Springer, 2013. https://doi.org/10.1007/978-3-642-37198-1_19.

[23] S. Wang, Q. Zhu, L. Kang. Landscape properties and hybrid evolutionary algorithm for optimum multiuser detection problem. In *International Conference on Computational Science*, pp. 340–347. Springer, 2006. https://doi.org/10.1007/11758501_48.

[24] C. Fonlupt, D. Robilliard, P. Preux. Fitness landscape and the behavior of heuristics. In *Evolution Artificielle*, vol. 97, p. 56. Citeseer, 1997.

[25] E. Pitzer, M. Affenzeller. A comprehensive survey on fitness landscape analysis. In *Recent advances in intelligent engineering systems*, pp. 161–191. Springer, 2012. https://doi.org/10.1007/978-3-642-23229-9_8.

[26] L. Kovács, A. Agárdi, T. Bányai. Fitness landscape analysis and edge weighting-based optimization of vehicle routing problems. *Processes* **8**(11):1363, 2020. https://doi.org/10.3390/pr8111363.

[27] A. Agárdi, L. Kovács, T. Bányai. An attraction map framework of a complex multi-echelon vehicle routing problem with random walk analysis. *Applied Sciences* **11**(5), 2021. https://doi.org/10.3390/app11052100.

[28] E. Pitzer, M. Affenzeller, A. Beham, S. Wagner. Comprehensive and automatic fitness landscape analysis using heuristiclab. In *International Conference on Computer Aided Systems Theory*, pp. 424–431. Springer, 2011. https://doi.org/10.1007/978-3-642-27549-4_54.

[29] E. Pitzer, A. Beham, M. Affenzeller. Generic hardness estimation using fitness and parameter landscapes applied to robust taboo search and the quadratic assignment problem. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pp. 393–400. 2012. https://doi.org/10.1145/2330784.2330845.

[30] P. R. d. O. da Costa, J. Rhuggenaath, Y. Zhang, A. Akcay. Learning 2-opt heuristics for the traveling salesman problem via deep reinforcement learning, 2020. arXiv:2004.01608.

[31] A. Srivastav, S. Agrawal. Multi-objective optimization of mixture inventory system experiencing order crossover. *Annals of Operations Research* **290**(1):943–960, 2020. https://doi.org/10.1007/s10479-017-2744-4.

[32] T. Visutarrom, T.-C. Chiang. An evolutionary algorithm with heuristic longest cycle crossover for solving the capacitated vehicle routing problem. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 673–680. IEEE, 2019. https://doi.org/10.1109/cec.2019.8789946.

[33] M. A. Al-Omeer, Z. H. Ahmed. Comparative study of crossover operators for the MTSP. In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–6. IEEE, 2019. https://doi.org/10.1109/iccisci.2019.8716483.

[34] K. Li, Z. Liang, S. Yang, et al. Performance analyses of differential evolution algorithm based on dynamic fitness landscape. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* **13**(1):36–61, 2019. https://doi.org/10.4018/ijcini.2019010104.

[35] A. Shamir, I. Safran, E. Ronen, O. Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance, 2019. arXiv:1901.10861.

[36] I. Khan, M. K. Maiti. A swap sequence based artificial bee colony algorithm for traveling salesman problem. *Swarm and evolutionary computation* **44**:428–438, 2019. https://doi.org/10.1016/j.swevo.2018.05.006.