sciendo
ISSN 2083-1587; e-ISSN 2449-5999
2025,Vol. 29,No.1, pp. 187-200
Agricultural Engineering
https://agriceng.org
DOI: 10.2478/agriceng-2025-0011

AN ITERATED LOCAL SEARCH METHOD FOR DETERMINING ROUTES IN THE TRANSPORT OF RAW MATERIALS AND AGRI-FOOD PROCESSING PRODUCTS

Wojciech Misztal[a]

[a] Department of Agricultural, Forestry and Transport Machines,University of Life Sciences in Lublin, 28 Gleboka Street, 20-612 Lublin, Poland, e-mail: wojciech.misztal@up.lublin.pl, ORCID 0000-0001-6214-506X

Corresponding author: e-mail: wojciech.misztal@up.lublin.pl

ARTICLE INFO

Article history:
Received: January 2025
Received in the revised form: April 2025
Accepted: May 2025

Keywords:
transport,
optimization,
iterated local search,
heterogeneous vehicle routing problem

ABSTRACT

To acquire raw materials and distribute products, agri-food processing companies need to operate at long distances, which is associated with high energy costs and an increased negative impact on the natural environment. Many methods have been developed to reduce energy consumption in transport. One approach that stands out is optimization of transport routes, which provides benefits while incurring hardly any extra costs. In this paper, an attempt was made to modify the ILS-RVND metaheuristic by adjusting its components to the specific character of the problem being solved so that they could yield the expected results with regard to the quality of returned solutions and the time in which they were generated. A group of local search algorithms, Swap (2-1), Cross-exchange (2-1), 3-opt, Or-opt (2) and Displacement, were analyzed. The results of this analysis were used to formulate the final version of the metaheuristic. The efficiency of the algorithm was evaluated using test cases. The solutions generated by the metaheuristic produced considerable improvement in the objective function (70.99% on average) and were obtained within an acceptable time (on average 24.66 CPU seconds).

Introduction

Transport operations are performed at every stage of production, starting from the gathering of raw materials, through the movement of production components during the production process, to the distribution of goods and disposal of by-products. Goods distribution costs account for nearly 50% of the total logistics costs, while Lapinskaitė and Kuckailytė (2014) state that supply chain costs on average constitute about 55% of the total product price. For some companies working in the food sector, transport costs may constitute up to 70% of the value added to goods (Baran, 2019). This group includes agri-food-processing companies, which have to secure raw materials from farms that are often located a long distance from the production plant and from one another (especially when there is a shortage of raw materials locally). The raw materials used in agri-food processing plants are usually characterized by low durability due to the ongoing natural biological processes. They usually cannot be terminated, but can only be slowed down using appropriate procedures (Baran, 2019). Since transport is the main factor that adversely affects the quality of raw materials, companies should do their most to ensure optimal conditions for the movement of these commodities, especially in situations where an unbroken cold chain has to be ensured (Cerchione et al., 2018; Chaudhuri et al., 2018). Transport activities are characterized by a high energy consumption, which means they are also capital-intensive and exert a large negative impact on the natural environment at every stage of vehicle operation/lifecycle (Kampf, 2018; Plizga, 2021; Przywara et al., 2022). Energy consumption and the negative impact of transport on the natural environment can be reduced in different ways, with different results (Dzieniszewski et al., 2021; Lairenlakpam et al., 2017).

Optimization of transport activities can deliver substantial benefits (Stopka et al., 2019). Usually, however, it is a highly demanding and complicated process. The literature describes numerous transportation problems of various nature (Toth and Vigo, 2002). The high diversity of these problems implies that approaches designed to solve some of those problems cannot be successfully applied to others. This means that a transport problem has to be properly identified before an optimization method is applied. Also, the methods themselves must be carefully selected, as they determine the quality of the solutions returned and the time in which they are generated. The quality of solutions translates directly into the benefits to be gained in implementing them, while running time is important from the point of view of practicability. An algorithm's running time must be short enough for it to return results at the moment they are needed.

The success of the optimization process aimed at reducing the total distance (time or cost) depends on the effectiveness of the approach used. In the case of the ILS-RVND metaheuristics, the key factors are the heuristics and perturbation mechanisms it uses. The aim of the work is to conduct an analysis of selected local search algorithms, aimed at identifying those that are able to provide the highest possible quality of returned solutions without excessively extending the operation time.

In the case of the agri-food processing sector, transportation problems related to the supply of raw materials and the distribution of products are in fact one and the same problem. This problem can be described in the following way: a fleet of (usually non-identical) vehicles are available at a depot; their task is to service a certain group of geographically dispersed locations differing in demand/supply, from which they pick-up raw materials/to which they deliver finished products. When all orders have been serviced, each vehicle must return to the place from which it set off, i.e. each route (and the number of routes usually corresponds to the number of vehicles in the fleet) starts and ends at the depot. To reduce the total distance traveled (travel time or cost), the pick-up/drop-off points must be divided into groups (in the best possible way), each of which will be serviced by a different vehicle (along a different route), and the optimal order in which the individual destinations in each of these groups will be visited must be determined. A problem defined in this way is a *Vehicle Routing Problem* (*VRP*), which is a generalization of the *Traveling Salesman Problem* (*TSP*) (Absi et al., 2020; Alonso-Pecina et al., 2024; Silva et al., 2015; Toth and Vigo, 2002; Zhou et al., 2018). The *VRP* belongs to the class of NP-hard problems, for which there are no polynomial-time algorithms (Absi et al., 2020; Alonso-Pecina et al., 2024; Silva et al., 2015; Toth and Vigo, 2002). This problem has many different applications, often in areas that are very distinct from the distribution of goods (Rojas-Cuevas et al., 2018; Toth and Vigo, 2002). There are numerous variants of the *VRP* (Silva et al., 2015; Toth and Vigo, 2002), which take into account various additional constraints, such as the maximum payload capacity of the vehicles – *Capacitated VRP* (*CVRP*), time – *VRP with Time Windows*, maximum distance that a vehicle can travel – *Distance-Constrained VRP*, task type – *VRP with Backhauls*, or mixed service – *VRP with Pickup and Delivery* (Rojas-Cuevas et al., 2018; Toth and Vigo, 2002). In the standard version of the *CVRP*, a homogeneous vehicle fleet is used, but in real-life settings, companies usually run fleets made up of non-identical vehicles. Practical cases are better represented by the *Heterogeneous VRP* (*HVRP*).

The mathematical model of the *HVRP* can be formulated as follows:

Minimize function:

$$Z = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{K} d_{ij}\, x_{ij}^{k} \tag{1}$$

subject to:

$$\sum_{i=0}^{n}\sum_{k-1}^{K} x_{ij}^{k} = 1 \qquad j = 1,\dots,n$$

$$\sum_{i=0}^{n} x_{ip}^{k} - \sum_{j=0}^{n} x_{pj}^{k} = 0\,0 \qquad for \quad k = 1,\dots,K; \quad p = 0,\dots,n$$

$$\sum_{i=0}^{n} \overline{q}_{i} \sum_{j=0}^{n} x_{ij}^{k} \le Q_{k} \qquad k = 1,\dots,K$$

$$\sum_{j=1}^{n} x_{oj}^{k} \le 1\,1 \qquad k = 1,\dots,K$$

$$Y_i - Y_j + n\sum_{k=1}^{K} x_{ij}^{k} \le n - 1\,1 \qquad 1 \le i \ne j \le n, \qquad Y_i, Y_j \in R$$

$$x_{ij}^{k} = 0 \quad or \quad 1 \quad for \quad all \quad i,j,k$$

where:

n – number of pick-up points (including the depot),
K – number of vehicles,
Q_k – payload capacity of the *k*-th vehicle,
$\bar{q}_i$ – supply (in units of weight) of the i-th point (farm), with $\bar{q}_o = 0$,
d_{ij} – the shortest distance between point i and point j,
$x_{ij}^{k} = 1$if vehicle k services the route section (i, j),
$x_{ij}^{k} = 0$if vehicle k does not service the route section (i, j).

Materials and Methods

The *HVRP* has been investigated by a group of researchers whose efforts have resulted in the development of a number of methods based on various approaches, which differ in terms of efficiency, functionality, and the quality of returned solutions (Amador-Fontalvo et al. 2014; Lai et al., 2016; Liao et al., 2018; Panggabean et al., 2018; Toth and Vigo, 2002). Limitations on the applicability of these methods are a consequence of both their functionality, which encompasses only certain specific variants of the problem, as well as the complexity of the *HVRP* itself, which allows to obtain exact solutions only in the case of small problems (Liao et al., 2018; Oesterle and Bauernhansl, 2016; Panggabean et al., 2018; Toth and Vigo, 2002). The quality of returned solutions obviously affects the magnitude of the benefits gained. Unfortunately, an improvement in quality comes at the cost of prolonging the algorithm's running time (Toth and Vigo, 2002). In practical applications, a trade-off must be made between the quality of solutions and the time in which they are generated. It can be assumed that problems of larger sizes (networks containing more than 100 vertices) cannot be solved using exact methods because an exact algorithm would take too long to return a result (Toth and Vigo, 2002). An alternative approach is to use heuristic and metaheuristic algorithms, which have much shorter running times. Though they provide poorer quality results, as a rule, these solutions are close to optimal (Toth and Vigo, 2002). The great diversity of concepts on which these approaches are based can cause certain problems. For example, they can tend to generate too many routes. Other methods, on the other hand, may be hard to implement in software (Amador-Fontalvo et al. 2014; Lai et al., 2016; Panggabean et al., 2018; Toth and Vigo, 2002).

A metaheuristic that stands out among the methods developed specifically for solving *HVRP*s is *Iterated Local Search (ILS)*, which is based on the proposition that a local optimum shares some common features with the global optimum, but it is not known which features are shared, and that the global optimum is a local optimum relative to all possible local optima. This metaheuristic iterates calls to a local search routine for a set of solutions obtained by perturbing previously visited local optima (Alvarez and Munari, 2017; Labadie et al., 2016). The pseudo-code for this procedure is shown in Algorithm 1 (Penna et al., 2013).

Algorithm 1. Iterated Local Search

1. ***Procedure*** *ILS*(s_0 *← initial solution):*
2. $s^* \leftarrow$ *LocalSearch*(s_0)
3. ***while*** stopping criterion is not met*:*
4. $s^{**} \leftarrow$ *Perturb* (s^*, *history)*
5. $s^{***} \leftarrow$ *LocalSearch* (s^{**})
6. $s^* \leftarrow$ *AcceptanceCriterion* (s^*, s^{**}, *history)*
7. ***return*** s^*

The formulation of Algorithm 1 should be understood as follows: the initial solution s_0 is passed as an argument to the procedure (line 1). This solution is then submitted to a local search, as a result of which solution s^* is obtained (line 2). The *while* loop (lines 3–6) executes the instructions in lines 4–6 sequentially until a stopping criterion is met (as a rule, the loop terminates when no further improvement can be achieved). These instructions are used to: generate solution s^{**} by perturbing solution s^* (line 4); run a local search algorithm to find solution *s**** (line 5); check whether the solution obtained is better than the best solution found so far and, if so, accept it as the current local optimum (if it meets the conditions of feasibility).

An important modification of the *ILS* is *Iterated Local Search with Random Variable Neighborhood Descent* (*ILS-RVND*), which uses *inter-route* neighborhood structures (in which customers are relocated from one route to another) and *intra-route* neighborhood structures (in which customers are relocated within one route) selected in the local search phase in order to increase the search space (Alvarez and Munari, 2017; Labadie et al., 2016). The main difference in pseudo-codes between *ILS* and *ILS-RVND* relates to the instructions in line 5 of Algorithm 1. In *ILS-RVND*, line 5 contains a call to the *RVND* procedure represented by the following notation *s***← RVND (s**)* (Alvarez and Munari, 2017; Labadie et al., 2016). The pseudocode of the *RVND* procedure is shown in Algorithm 2 (Penna et al., 2013).

Algorithm 2. Random Variable Neighborhood Descent

1. ***Procedure*** *RVND(s′ ← current solution):*
2. *s″ ← s′*
3. *NL ← list of neighborhood structures*
4. ***while*** *NL≠Ø:*
5. *N ← neighborhoods randomly selected from NL*
6. *s‴ ← solution generated by neighborhood N*
7. *if f(s‴) < f(s″):*
8. *s″ ← s‴*
9. *else:*
10. *remove N from NL*
11. ***return*** *s″*

The formulation of Algorithm 2 should be understood as follows: First, the current solution *s′* (line 1) is passed to the procedure. Then, this solution is adopted as the best current solution *s″* (line 2). Next (line 3), a list of all neighborhood structures *NL* to be used by the algorithm is initialized. The *while* loop (lines 4–10) keeps executing its code until *NL* becomes empty. It repeatedly executes the instructions in lines 5–10 to: randomly select neighborhood *N* from the neighborhood list *NL* (line 5); use this neighborhood to find the best neighboring solution *s″* (line 6); check whether the value of the objective function of *s‴* is lower than the value of the objective function of solution *s″* (line 7) and, if so, accept *s‴* as the best current solution *s″* (line 8), or, if not, remove neighborhood *N* from *NL* (line 10). The instruction in line 11 is responsible for returning solution *s″* at the moment when the algorithm finishes its execution.

Critical to proper performance of the *ILS-RVND* is the appropriate selection of the component algorithms (neighborhood structures and perturbation mechanisms). Because these algorithms have a direct impact on the efficiency and effectiveness of the optimization process (Alvarez and Munari, 2017; Labadie et al., 2016), they should be selected based on a previous analysis in which excessively time-consuming and low-efficiency approaches are eliminated. This study focuses on a certain group of algorithms that can generate good-quality solutions with a relatively low computational effort. This group includes the *inter-route* algorithms *Swap (2-1)* and *Cross-exchange (2-1)* and the *intra-route* algorithms *3-opt*, *Or-*

opt (2) and *Displacement* (Alvarez and Munari, 2017; Kara et al., 2014; Pop et al., 2014; Silva et al., 2015; Toth and Vigo, 2002).

The neighborhood structures were assessed using randomly generated test cases, the parameters of which are shown in Table 1.

Table 1.
Test case parameters

Problem size (number of vertices)	Number of routes	Supply/demand	Values of cost matrix elements
10	1–3	2–10	1–10
20			
30			
40			
50			
60	1–5		
70			
80			
90			
100			

The maximum payload capacities of the vehicles were set so as to ensure the practicability of the solutions i.e. the aggregate payload was equal to or greater than the total supply of all the points serviced. Ten variants of each type of test instances were generated. Additionally, to make sure that the running time was measured correctly, each algorithm was run ten times.

Results and Discussion

All the algorithms and test case generation procedures were implemented in *Python*. The tests were carried out on a PC with i5-4590 CPU and 8GB RAM.

The results of the tests of the neighborhood structures are shown in Tables 2 and 3 and in graphs in Figs. 1 and 2, plotted on the basis of these tables.

Table 2.
Results of tests of inter-route neighborhood structures

	Improvement in objective function [%]			Running time [CPU seconds]		
Problem size	**Min.**	**Arithmetic mean**	**Max.**	**Best**	**Arithmetic mean**	**Worst**
			Swap (2-1)			
10	0	25.64	65.57	0	0	0.02
20	0	17.39	66.94	0	0	0.03
30	0	9.3	67.26	0	0.01	0.09
40	0	13.6	68.34	0	0.03	0.19
50	0	7.64	38.16	0	0.03	0.16
60	0	16.78	70.66	0	0.13	0.64
70	0	17.16	72.24	0	0.21	0.81
80	0	16.38	53.72	0	0.28	1.25

Problem size	Improvement in objective function [%] Min.	Arithmetic mean	Max.	Running time [CPU seconds] Best	Arithmetic mean	Worst
			Swap (2-1)			
90	0	17.48	71.15	0	0.38	1.34
100	0	13.95	73.35	0	0.46	2.87
			Cross-exchange (2-1)			
10	0	25.72	63.33	0	0	0.02
20	0	17.17	69.49	0	0	0.02
30	0	8.38	47.62	0	0.01	0.03
40	0	12.78	63.96	0	0.02	0.16
50	0	7.43	42.11	0	0.03	0.16
60	0	16.10	68.45	0	0.12	0.67
70	0	16.54	68.19	0	0.19	0.72
80	0	16.14	49.77	0	0.28	1.14
90	0	16.32	69.16	0	0.35	1.03
100	0	13.83	74.45	0	0.44	2.28

Table 3.
Results of tests of intra-route neighborhood structures

Problem size	Improvement in objective function [%] Min.	Arithmetic mean	Max.	Running time [CPU seconds] Best	Arithmetic mean	Worst
			3-opt			
10	16.65	20.92	46.15	0	0	0.02
20	16.49	26.31	42.98	0	0.01	0.03
30	13.7	24.84	36.42	0	0.08	0.16
40	12.37	23.00	32.33	0.02	0.24	0.48
50	11.16	21.96	34.38	0.05	0.57	1.17
60	9.96	21.79	35.16	0.03	0.85	2.37
70	8.48	21.35	32.67	0.08	1.4	4.41
80	7.65	20.32	33.72	0.08	2.53	7.43
90	5.69	19.23	33.47	0.27	3.66	11.72
100	8.28	17.10	26.62	0.45	5.61	17.71
			Or-opt (2)			
10	7.14	17.92	32.79	0	0	0
20	9	14.01	18.37	0	0	0.02
30	8.24	10.79	15.38	0	0	0.02
40	3.7	8.64	11.06	0	0.01	0.02
50	4.96	7.17	9.8	0	0.01	0.02
60	2.6	5.66	7.33	0	0.01	0.03
70	3.76	5.14	6.41	0	0.02	0.03
80	3.2	4.62	5.93	0	0.02	0.05
90	2.4	4.05	5.34	0	0.03	0.06
100	2.11	3.77	4.74	0	0.03	0.06

Problem size	Min.	Arithmetic mean	Max.	Best	Arithmetic mean	Worst
	Improvement in objective function [%]			Running time [CPU seconds]		
			Displacement			
10	6.12	18.87	39.53	0	0	0.02
20	8.57	13.78	17.98	0	0	0.02
30	8.02	11.07	14.39	0	0	0.02
40	2.78	8.41	10.84	0	0	0.02
50	4.37	7.13	9.8	0	0	0.02
60	2.78	5.63	7.67	0	0	0.02
70	3.15	4.99	6.96	0	0	0.02
80	2.87	4.6	5.56	0	0.01	0.02
90	2.61	4.07	5.13	0	0.01	0.02
100	2.73	3.78	4.66	0	0.01	0.02

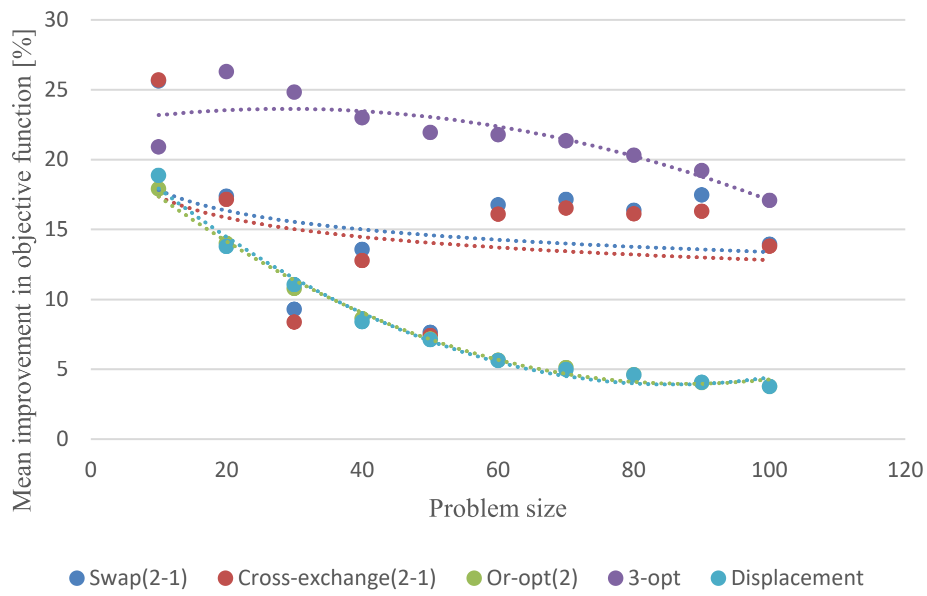

Figure 1. Mean percent improvement in the objective function value achieved by the analyzed neighborhood structures

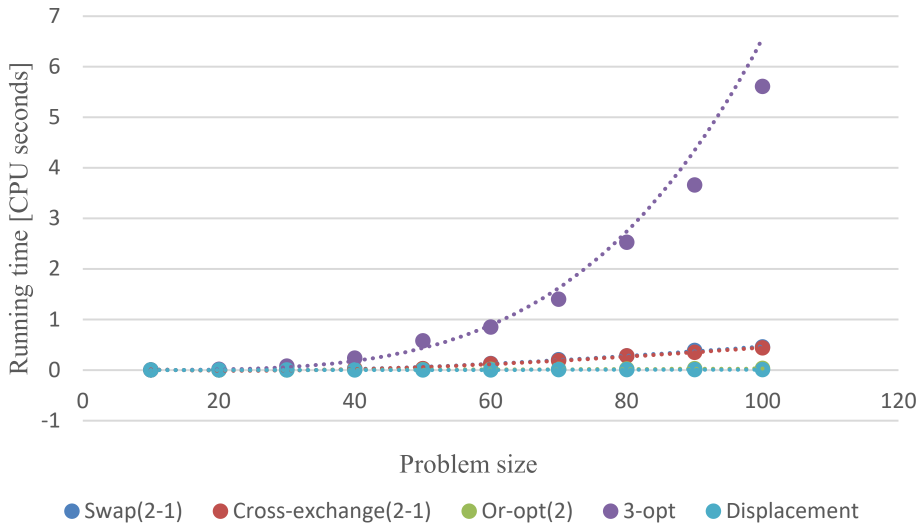

Figure 2. Mean time in which the final solutions were generated

The data presented in the tables and graphs indicate that the best quality solutions can be generated using the *3-opt*, *Swap (2-1)* and *Cross-exchange (2-1)* algorithms. This is supported by the high values of average and maximal improvement in the objective function. The values of mean improvement in the objective function averaged across all test cases were 21.36%, 15.63% and 14.21%, respectively. It is worth emphasizing that the very low minimal improvement values (a complete lack of improvement) obtained by *Swap (2-1)* and *Cross-exchange (2-1)* do not really signify that the quality of these algorithms was low. This is because it was assumed, when test cases were being generated, that the procedure developed in this study would be used to solve a broader range of problems, including cases in which only one vehicle is used (so-called *Resource-Constrained Traveling Salesman Problems*). In situations of this type, *inter-route* algorithms such as *Swap (2-1)* and *Cross-exchange* (which relocated vertices between different routes) obviously cannot initiate any changes. When these cases are excluded, however, both the minimal and the moderate improvement values turn out to be much higher. Indeed, it should be pointed out that, under certain conditions, *Swap (2-1)* and *Cross-exchange (2-1)* can generate much better solutions than *3-opt*, as evidenced by the mean maximal improvement values for these three structures: 44.85%, 43.61% and 30.91%, respectively. *Or-opt (2)* and *Displacement* performed much worse compared to the other structures. These moves yielded, on average, an 8.79% and an 8.56% improvement in the value of the objective function, respectively. On the upside, these algorithms returned solutions in a short time (even when large-sized problems were being solved). The longest they took to solve the largest problems was 0.06 CPU seconds for *Or-opt (2)* and 0.02 CPU seconds for *Displacement*. To compare, the fastest among the other algorithms (*Cross-exchange (2-1)*) had a running time of 2.28 CPU seconds. The structure that performed the worst with respect to this criterion was *3-opt*. It had the longest running time, which increased with the size of the problem, which is a significant limitation to its use.

Given the possibilities offered by the analyzed neighborhood structures with regard to the quality of the solutions they returned and the time in which they returned them, it was decided that the best option was to adjust the pool of local search algorithms to the specific character of the problem being considered, and more precisely, to the size of that problem (number of vertices in the network) and the number of vehicles used (number of routes). On this basis, we assumed that in the case of problems in which consumers were serviced by one vehicle, the set of neighborhood structures should contain the *3-opt* and *Displacement* algorithms. For transportation networks with more than one vehicle, *Swap (2-1)*, *Cross-exchange (2-1)* and *3-opt* are the best structures to use if the number of vertices in the network does not exceed 100; in other cases, *Swap (2-1)*, *Cross- exchange (2-1)* and *Displacement* should be used.

The software-implemented *ILS-RVND* metaheuristic was tested using the same test cases as the other algorithms. The results of the tests are shown in Table 5. The mean values of all measurements performed using the individual algorithms are tabulated in Table 6.

Table 5.
Results of tests of the ILS-RVND algorithm

Problem size	Improvement in objective function value [%]			Running time [CPU seconds]		
	Minimal	Average	Maximal	Best	Average	Worst
10	30	55.59	66.67	0	0.01	0.02
20	49.49	61.83	71.3	0.02	0.08	0.2
30	60.96	66.98	76.1	0.41	0.68	0.98
40	69.05	73.22	76.53	1.09	3.56	7.39
50	70.78	74.19	79.3	1.9	5.57	12.52
60	68.59	73.06	77.98	0.97	10.04	28.92
70	69.47	74.17	78.87	1.75	23.98	59.89
80	72.18	76.37	80.34	4.32	33.77	99.71
90	71.28	76.71	80.77	6.49	70.25	185.93
100	75.05	77.76	81.87	21.96	98.69	289.68

Table 6.
Comparison of means of all measurements/tests

Algorithm	Improvement in objective function value [%]			Running time [CPU seconds]		
	Minimal	Average	Maximal	Best	Average	Worst
ILS-RVND	63.69	70.99	76.97	3.89	24.66	68.52
3-opt	12.11	21.36	30.91	0.21	1.64	4.53
Swap (2-1)	0	15.63	44.85	0	0.15	0.5
Cross-exchange (2-1)	0	14.21	43.61	0	0.13	0.44
Or-opt (2)	6.87	8.79	11.04	0	0.01	0.03
Displacement	6.85	8.56	9.89	0	0	0.01

Conclusions

The results of the tests show that the software-implemented *ILS-RVND* metaheuristic generates much better quality solutions compared to each of the local search algorithms which it uses as neighborhood structures. The average improvement in the objective function yielded by *ILS-RVND* was 70.99% (mean of all measurements), a result that is over three times higher than the 20% mean improvement obtained by running *3-opt*, which ranked second on this criterion among all local search algorithms. A similar observation can be made for mean minimal and mean maximal percent improvements in the value of the objective function obtained using the proposed metaheuristic (63.69 and 76.97%, respectively). The effectiveness of this approach stems from the fact that cooperation between the individual neighborhood structures supported by the perturbation mechanism leads to considerable enlargement of the search space. The metaheuristic allows to reach areas in this space that are inaccessible to any of the algorithms run separately. Another important advantage of such cooperation is that the metaheuristic is not constrained in the way the individual neighborhood structures are when they are used separately (for example, *inter-route* algorithms cannot generate any improvement when applied to problems with one route). By appropriately selecting neighborhood structures, one can enhance the quality of the solutions and make sure that the time in which they are generated is not excessively extended, which is a natural consequence of the expansion of the procedure. In the case of the *ILS-RVND* metaheuristic proposed in this paper, running time was considerably extended (the mean time in which solutions were returned was fifteen times longer than the running time of the slowest of the local search algorithms *3-opt*). It was also longer than the mean longest running time. However, taking into account that the longest *RVND* took to solve the largest problem was only 289.68 CPU seconds and that it yielded a major improvement in the objective function, it can be said that the running time of this metaheuristics is acceptable. Accordingly, this approach can be successfully applied in practice when solving real-life problems related to obtaining raw materials or distribution of finished products in the agri-food processing sector.

The fact that a major improvement in the value of the objective function was achieved when solving test cases does not mean that the same level of improvement can be achieved for every possible problem. It should be noted that the final benefits will obviously depend on the initial state of the routes. The test problems were generated randomly, so they may be much more complex than networks generated by a human (even one using a naive approach). Depending on the complexity of a problem, the percent improvement in the objective function achieved using the proposed metaheuristic may be different. Even so, the advantage of this approach over the other algorithms is considerable, and therefore the chances for improvement it affords are high. It is also worth adding that the test cases considered in this study included unfavorable problems requiring more calculations and, thus, more time. In the case of most problems, the time in which the metaheuristic returns solutions should be much shorter (as indicated by the best running times).

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

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

Funding: This research received no external funding.

References

Absi, N., Cattaruzza, D., Feillet, D., Ogier, M., & Semet, F. (2020). A heuristic branch-cut-and-price algorithm for the ROADEF/EURO challenge on Inventory Routing. *Transportation Science, 54*(2), 313-329. https://doi.org/10.1287/trsc.2019.0961.

Alonso-Pecina, F., Hérnandez-Báez, I. Y., López-Díaz, R. E., & Cruz-Rosales, M. H. (2024). Iterated Local Search Approach to a Single-Product, Multiple-Source, Inventory-Routing Problem. *Mathematics, 12*(7), 991. https://doi.org/10.3390/math12070991.

Alvarez, A., & Munari, P. (2017). An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. *Computers & Operations Research, 83*, 1-12. https://doi.org/10.1016/j.cor.2017.02.001.

Amador-Fontalvo, J. E., Paternina-Arboleda, C. D., & Montoya-Torres, J. R. (2014). Solving the heterogeneous vehicle routing problem with time windows and multiple products via a bacterial metaheuristic. *International Journal of Advanced Operations Management, 6*(1), 81-100. https://doi.org/10.1504/IJAOM.2014.059622.

Baran, J. (2018). Supply chain management in agri-food processing enterprises. *Journal of Modern Science, 4*, 39 (4), 217-230. https://doi.org/10.13166/jms/103641.

Cerchione, R., Singh, R., Centobelli, P., & Shabani, A. (2018). Food cold chain management: From a structured literature review to a conceptual framework and research agenda. *The International Journal of Logistics Management, 29*(3), 792-821. https://doi.org/10.1108/IJLM-01-2017-0007.

Chaudhuri, A., Dukovska-Popovska, I., Subramanian, N., Chan, H. K., & Bai, R. (2018). Decision-making in cold chain logistics using data analytics: a literature review. *The International Journal of Logistics Management, 29*(3), 839-861. https://doi.org/10.1108/IJLM-03-2017-0059.

Dzieniszewski, G., Kuboń, M., Pristavka, M., & Findura, P. (2021). Operating parameters and environmental indicators of diesel engines fed with crop-based fuels. *Agricultural Engineering, 25*(1), 13-28. https://doi.org/10.2478/agriceng-2021-0002.

Kampf, R. (2018). Optimization of delivery routes using the Little´ s algorithm. *NAŠE MORE: znanstveni časopis za more i pomorstvo, 65*(4 Special issue), 237-239. https://doi.org/10.17818/NM/2018/4SI.13.

Kara, B. Y., Sabuncuoglu, I., & Bidanda, B. (Eds.). (2014). Global logistics management. CRC Press. https://doi.org/10.1201/b17845.

Labadie, N., Prins, C., & Prodhon, C. (2016). Metaheuristics for vehicle routing problems. John Wiley & Sons. https://doi.org/10.1002/9781119136767.

Lai, D. S., Demirag, O. C., & Leung, J. M. (2016). A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph. *Transportation Research Part E: Logistics and Transportation Review, 86*, 32-52. https://doi.org/10.1016/j.tre.2015.12.001.

Lairenlakpam, R., Thakre, G. D., Gupta, P., Singh, Y., & Kumar, P. (2017). Effect of different drive modes on energy consumption of an electric auto rickshaw. In 2017 IEEE Transportation Electrification Conference (ITEC-India) (pp. 1-5). IEEE. https://doi.org/10.1109/ITEC-India.2017.8333832.

Lapinskaitė, I., & Kuckailytė, J. (2014). The impact of supply chain cost on the price of the final product. *Business, Management and Education*, *12*(1), 109-126. https://doi.org/10.3846/bme.2014.08

Liao, W., Liu, L., & Fu, J. (2019). A comparative study on the routing problem of electric and fuel vehicles considering carbon trading. *International Journal of Environmental Research and Public Health, 16*(17), 3120. https://doi.org/10.3390/ijerph16173120.

Panggabean, E. M., Mawengkang, H., Azis, Z., & Sari, R. F. (2018). Periodic heterogeneous vehicle routing problem with driver scheduling. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012017). IOP Publishing. https://doi.org/10.1088/1757-899X/300/1/012017.

Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics, 19*(2), 201-232. https://doi.org/10.1007/s10732-011-9186-y.

Plizga, K. (2021). Analysis of Energy Consumption by Electric Agricultural Tractor Model Under Operating Conditions. *Agricultural Engineering, 25*, 1-12. https://doi.org/10.2478/agriceng-2021-000

Pop, P. C., Fuksz, L., & Marc, A. H. (2014, June). A variable neighborhood search approach for solving the generalized vehicle routing problem. In International Conference on Hybrid Artificial Intelligence Systems (pp. 13-24). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07617-1_2.

Przywara, A., Koszel, M., Krawczuk, A., & Sotirios Anifantis, A. (2022). The Analysis of the New Farm Tractors Market in Poland in 2010-2020 in the Context of Income Generated by Farmers' Households. *Agricultural Engineering, 26*, 65-79. https://doi.org/10.2478/agriceng-2022-0006.

Rojas-Cuevas, I. D., Caballero-Morales, S. O., Martinez-Flores, J. L., & Mendoza-Vazquez, J. R. (2018). Capacitated vehicle routing problem model for carriers. Journal of *Transport and Supply Chain Management, 12*(1), 1-9. https://doi.org/10.4102/jtscm.v12i0.345.

Silva, M. M., Subramanian, A., & Ochi, L. S. (2015). An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research, 53*, 234-249. https://doi.org/10.1016/j.cor.2014.08.005.

Stopka, O., Stopkova, M., & Kampf, R. (2019). Application of the operational research method to determine the optimum transport collection cycle of municipal waste in a predesignated urban area. *Sustainability, 11*(8), 2275. https://doi.org/10.3390/su11082275.

Toth, P. (2002). The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications. https://doi.org/10.1137/1.9780898718515.

Zhou, A. H., Zhu, L. P., Hu, B., Deng, S., Song, Y., Qiu, H., & Pan, S. (2018). Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. *Information, 10*(1), 7. https://doi.org/10.3390/info10010007.

Wojciech Misztal

METODA ITERACYJNEGO LOKALNEGO PRZESZUKIWANIA DO WYZNACZANIA TRAS W TRANSPORCIE SUROWCÓW ORAZ PRODUKTÓW PRZETWÓRSTWA ROLNO-SPOŻYWCZEGO

Streszczenie. Pozyskiwanie surowców i dystrybucja produktów wymaga od przedsiębiorstw przetwórstwa rolno-spożywczego pokonywania znacznych odległości, co wiąże się z wysokimi wydatkami energetycznymi oraz zwiększonym negatywnym oddziaływaniem na środowisko. Opracowano wiele metod mających na celu ograniczenie zużycia energii w transporcie. Jednym z wyróżniających się podejść jest optymalizacja tras przejazdów, która zapewnia znaczne korzyści przy ograniczonych nakładach. W pracy podjęto próbę modyfikacji metaheurystyki ILS-RVND poprzez dostosowanie jej elementów do specyfiki rozpatrywanego problemu w celu osiągnięcia poprawy jakości zwracanych rozwiązań oraz skrócenia czasu ich generowania. Przeanalizowano grupę algorytmów przeszukiwania lokalnego: Swap (2-1), Cross-exchange (2-1), 3-opt, Or-opt (2) oraz Displacement. Wyniki wykorzystano do określenia ostatecznej postaci metaheurystyki. Efektywność algorytmu oceniano przy użyciu przypadków testowych. Rozwiązania generowane przez metaheurystykę przynosiły znaczną poprawę funkcji celu (średnio 70,99%) i były otrzymywane w akceptowalnym czasie (średnio 24,66 sekundy CPU).

Słowa kluczowe: transport, optymalizacja, iteracyjne lokalne przeszukiwanie, problem wyznaczania tras dla heterogenicznej floty pojazdów