

COMPARISON OF ADVANCED METHODS FOR PICKING PATH OPTIMIZATION: CASE STUDY OF DUAL-ZONE WAREHOUSE

Sebo, J.* & Busa Jr., J.**

* Technical University of Košice, Letná 9, Košice, Slovakia

** Laboratory of Information Technologies at Joint Institute of Nuclear Research, Joliot-Curie 6, Dubna, Russia

E-Mail: juraj.sebo@tuke.sk, busa@jinr.ru

Abstract

Minimizing the travel distance of a picking tour is often considered an imperative factor in improving warehouse operation efficiency. This paper concentrates on the performance of the genetic algorithm (GA) method and its comparison to other routing strategies such as heuristics, the experienced warehouse picker and the brute-force algorithm under given assumptions. The set of simulations and calculations is based on an industrial case example. The results of the investigated routing strategies under given assumptions (middle size dual-zone warehouse, order size – 15 items, etc.) show the dominance of the brute-force algorithm in comparison to the experienced picker, GA and simple heuristics. It also indicates that GA is an optimization method which needs modification in dealing with picking path optimization problems and under given assumptions could generate better solutions than simple heuristics and comparable to experienced picker. The results also show quite significant sensitivity of GA results on used selection operator, size of population and number of generations. (Received in May 2020, accepted in August 2020. This paper was with the authors 1 month for 1 revision.)

Key Words: Picking Path, Optimization, Genetic Algorithm, Travel Distance, Routing Strategy

1. INTRODUCTION

Order picking is the operation of retrieving goods from specified storage locations based on customer orders [1]. Scientific literature has shown that picking, layout design, storage assignment, zoning, batching and routing are crucial aspects in order-picking operation improvement [2]. In a classical industrial warehouse, the order-picking operation optimization methods are mostly based on the traveling salesman problem (TSP). Numerous papers have dealt with approximate or exact methods inspired by the TSP to solve the order-picking problem. In some cases, factors such as numerous aisles, specific constraints, real-time computation cause that exact methods are not sufficient to solve the order-picking problem and approximate methods may be used to save the computation time [3]. The use of order batching and picking path optimization to reduce picking time and thereby increasing picking efficiency is a goal of the majority of studies [4]. This study focuses on the picking path optimization (routing policies) to reduce the picking path length (time). Minimizing the travel distance (total or average) of a picking tour is often considered as an imperative factor in reducing travel time and consequently improving warehouse operation efficiency [2]. Travel time is viewed as a waste since it costs labour hours and adds no value [5].

The objective of routing policies (picking path optimization) is to sequence the items on the pick list to ensure a good route through the warehouse. In practice, the problem of routing order pickers in a warehouse is mainly solved by using heuristics. This is also due to there not being an optimal algorithm for every layout, of not avoiding aisle congestion, etc. The S-shape method, return method, midpoint method, largest gap, and combined or composite method are the most common heuristics to determine the sequence of picks. All these methods were initially

developed for single-block warehouses, but with some modifications, they can also be used for multiple-block warehouses [2].

In solving the picker routing problem, several optimization methods have been proposed and tested. Key and Dasgupta [6] have discussed the feasibility of generating pick paths through the use of different algorithms. Cheng et al. [4] have used the ant colony optimization (ACO) algorithm in searching for the most effective traveling path for a batch. The outcomes obtained by De Santis et al. [7] have suggested that their ACO-based algorithm is generally able to provide better results than the heuristic and metaheuristic algorithms, is often able to find an exact solution and shows a very efficient computational time. However, Taherdangkoo et al. [8] have argued that while optimization algorithms such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm, Ant Colony Optimization (ACO) algorithm and Artificial Bee Colony (ABC) algorithm can give solutions to linear and non-linear problems near to the optimum for many applications, they can suffer from becoming trapped in local optima in some cases. They suggest the Stem Cells Algorithm (SCA) in overcoming the local optima problem, and the obtained results prove the superiority of the modified SCA in comparison with that of the other optimization algorithms. Finally, there have also been studies using other routing algorithms; see in [9-13].

Despite the pros and cons of the concurrent optimization algorithms, the literature [3, 14-20] has shown that the GA belongs to one of the methods which provide good results when determining a picking route. Chang et al. [15] implemented an order-picking operation mathematical model as well as a genetic algorithm to solve it. Hsu et al. [16] proposed a genetic algorithm to provide an order-picking batching method. Instead of using proximity measures and approximating the travel distances, the paper develops an optimization approach for order batching by directly minimizing the total travel distance. Wan et al. [14] used an improved genetic algorithm for the path optimization problem of the automated warehouse stocker picking operating process. As the result of demanding stockers to pick the shortest path, his fitness function is the reciprocal square of the stocker running path $f(d) = (1/d_{ij})^2$. Similarly, the current study uses genetic algorithms for the optimization of picking a path in a warehouse, while the fitness is directly related to distance and is minimized. The goal of this present study is to identify the relative improvements in shortening the travel distance generated by GA in comparison to other considered routing policies (heuristics, experienced picker and brute-force algorithm) and by this to extend the knowledge about GA presented by previous studies [3, 14-17].

2. METHODOLOGY

The problem can be described as follows. In the process of picking orders at a distribution warehouse, the order needs to be picked from shelves and collected at a depot. The picker (picking equipment) tries to find the optimal picking route to make the total travel distance the shortest.

According to the goal of the study, the performance of the GA is compared to the picking path generated by heuristics, experienced warehouse picker and brute-force algorithm. The first routing policy – heuristics – means picking items from the lowest to the highest number of a rack (in our warehouse layout from left to right), while using the central (middle) route. The study deals with a dual-zone warehouse (Fig. 1). This heuristics was selected because a barcode reader they use in the warehouse is equipped with it. The second routing policy – experienced picker – could be understood as the use of the warehouse's grid in a picker's mind or another mental structure which allows the picker to find the picking path. It is essential to mention that in this study an experienced picker is not characterized by any characteristics, excluding the time the picker had worked in the warehouse (in our case study it is a few years). The final

$$\begin{aligned}
 & \text{then } d_{ij} = |y_j - y_i| \\
 & \text{else } d_{ij} = \min(22 - y_j + |x_j - x_i| + 22 - y_i; y_j - 11 + |x_j - x_i| + y_i - 11) \\
 \text{else} & \text{ if } x_j = x_i \\
 & \text{then } d_{ij} = |y_j - y_i| \\
 & \text{else } d_{ij} = \min(11 - y_j + |x_j - x_i| + 11 - y_i; y_j + |x_j - x_i| + y_i)
 \end{aligned}$$

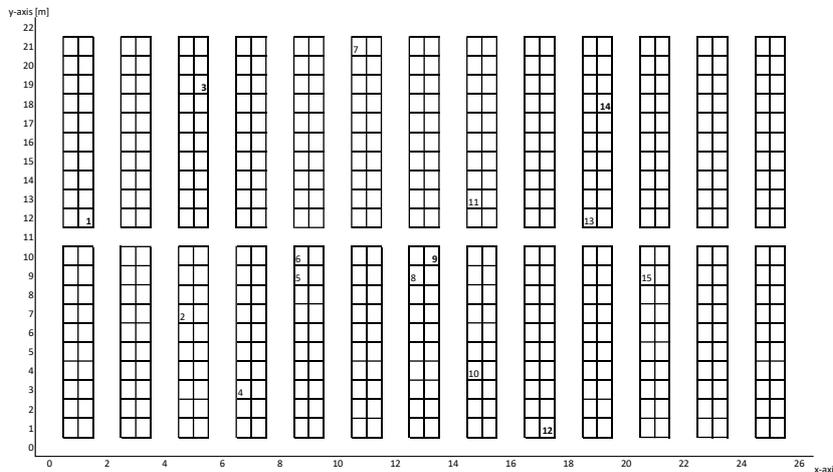


Figure 2: Model of a dual-zone warehouse with a system of x and y coordinates and order picking situation (randomly selected 15 items).

2.2 Description of brute-force routing policy

The brute-force approach consists of testing of all the possible combinations of the picking path and selecting the one with the shortest path. It is clear that for this approach to work, one needs to use distances between each pair of picking locations in a warehouse many times. Since picking locations within a warehouse do not change very often, it makes sense to calculate the distance between each two possible picking locations and store it for further use. In the case of our warehouse, about 299 picking locations are leading to a matrix of 89102 elements. The distance was calculated between each two picking locations using the Floyd-Warshall algorithm [22], which took about 0.035 s. All calculations were done on a Fedora 29 based computer with CPU Intel i7-6700 @ 3.4GHz, and as the compiler, g++ 8.2.1 was used. No parallelization was involved in the process.

Having the distances matrix, four different approaches to brute force were used. The first one created all permutations using Heap's algorithm [23], after obtaining each one, calculating the length of the obtained path, selecting and saving the shortest one. Since 15! combinations had to be tested, it was time-consuming and took about 4 h 53 min. Heap's algorithm is interesting from the point of view that each subsequent permutation differs from the previous one in a change of only two elements. Based on this and instead of calculating the whole length of the new path, we took the length of the previous one and only added the difference when compared to the previous one. When 2 points in the path change, a maximum of four sections of the path are affected. This led to a decrease in memory operations and the total calculation time dropped to 2 h 37 min. The recurrent approach was also tried. In this, a stack of all possible picking locations was taken, and one location was taken from the stack, placed to the picking path and called again function for picking and placing location points. The first implementation (testing all permutations again) ran for 7 h 40 min. In this recurrent approach, permutations are lexicographically sorted, and when, say 12 elements have been placed, and the route until now is already longer than the shortest found, it does not make sense to continue in the recurrence and the function can terminate (return to the caller). Only in 85632 cases (from a maximum of 15! = 1307674368000 cases) the program reached the end of recurrence and tested the obtained path. The calculation time was about 38 s.

2.3 Description of GA routing policy

In picking path optimization, there is a need to ensure that each item on the picking list will be picked just once, so the GA method cannot contain the same "genes" (picked items) in the "chromosome". In other words, each item can be present just once in a picking sequence. For this reason, the study used GA with randomly generated masks for creating children chromosomes in the crossover operation; see in [24]. The other characteristics of the used GA methodology (Basic GA Methodology in the current study) are as follows:

- Initial population: The initial population consists of 10 randomly generated individuals.
- Fitness calculation: Fitness is calculated as the travel distance. No transformation of the distance variable to standard fitness (i.e. higher fitness better solution) is applied. In the current case, the lower the fitness, the better the solution is. Similarly, Moon et al. [20] use the minimization of the objective function in GA methodology.
- Selection operator: Roulette wheel selection is used as the selection operator. Individuals from the previous generation are given a probability of being selected that is directly proportionate to their fitness. Each of the 10 runs of the roulette wheel selection designates one individual. Finally, it creates a "middle" population of 10 individuals which will enter the crossover operator.
- Crossover operator: The crossover operator is of uniform crossover kind. From the 10 individuals in the "middle" population, five pairs of parents are created. The pairs of parents are selected explicitly (individual 1 and individual 2, individual 3 and individual 4, etc.), which is standard when using roulette wheel selection. The crossover operator is executed through randomly generated masks; see in [24]. According to the 15 picked items, the mask consists of 15 genes. Each gene of the mask can acquire value 1 or 2 representing the parent (1 – first parent, 2 – second parent). Depending on the value (1 or 2), the gene value of the first parent or second parent will be inserted into the child's chromosome. For each pair of parents, two masks are generated, so two children are created. Finally, it makes a new generation consisting of 10 children. $P_{cross} = 1$.
- Mutation operator: From the new generation (population of 10 children), only the first four children (predetermined number) are selected for the mutation; see in [24]. The rest of the children remain unchanged. The mutation operator selects a random number of pairs of genes (in our case study from 1 to 4) and then randomly chooses the pairs for a swap. $P_{mut} = 0.4$.
- Stop condition: The algorithm will stop the generation of the new generation when it reaches the 40th generation.

The described basic GA methodology was used for the first round of simulations (10 simulations). The term simulation is understood in this study as the generation of 40 consecutive generations (from the original generation to 40th generation). Due to the rather implausible results, the methodology was modified by the incorporation of elitism (or elitist selection) and a second simulation round (10 simulations) was carried out. In this study, the elitist selection means that two individuals with the best fitness are carried over to the next generation unaltered. To test a more robust version of this GA methodology on outputs, the population was expanded to 40 units and the number of generated populations to 50 and the third simulation round (10 simulations) was executed. Accordingly, Basic Methodology with Elitist Selection and its more robust alternative are compared with other routing policies in the results part of the paper.

3. RESULTS

In this part, the solutions (the shortest paths) generated by different routing policies investigated in this study are presented.

3.1 Picking path generated by heuristics

Fig. 3 shows the order picking situation and picking path generated by heuristics. The travel distance is equal to 186 m.

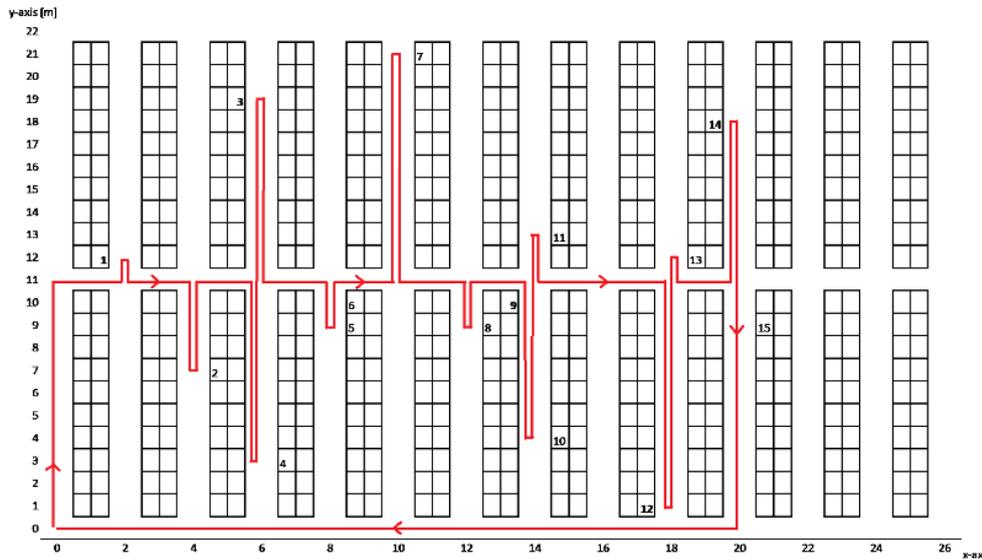


Figure 3: The order picking situation and picking path generated by heuristics.

3.2 Picking path generated by an experienced picker

Fig. 4 shows the order picking situation and picking path generated by the experienced picker. The travel distance is 162 m.

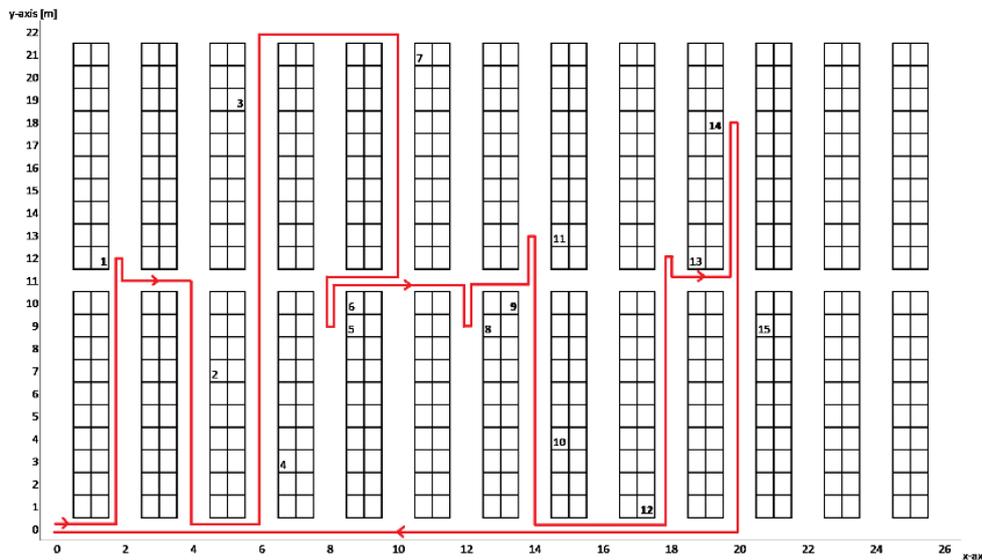


Figure 4: The order picking situation and picking path generated by the experienced picker.

3.3 Picking path generated by brute-force algorithm

The brute-force option was calculated (see the Methodology section) in a few variants with different calculation times (from a few hours to less than 1 minute). The solution with the shortest travel distance represents the following sequence of picked items (Fig. 5): 2, 1, 3, 7, 14, 15, 13, 12, 10, 9, 11, 8, 5, 6, 4. The travel distance is equal to 132 m. Within the framework of this study, this solution is considered as the global optimum for the defined problem.

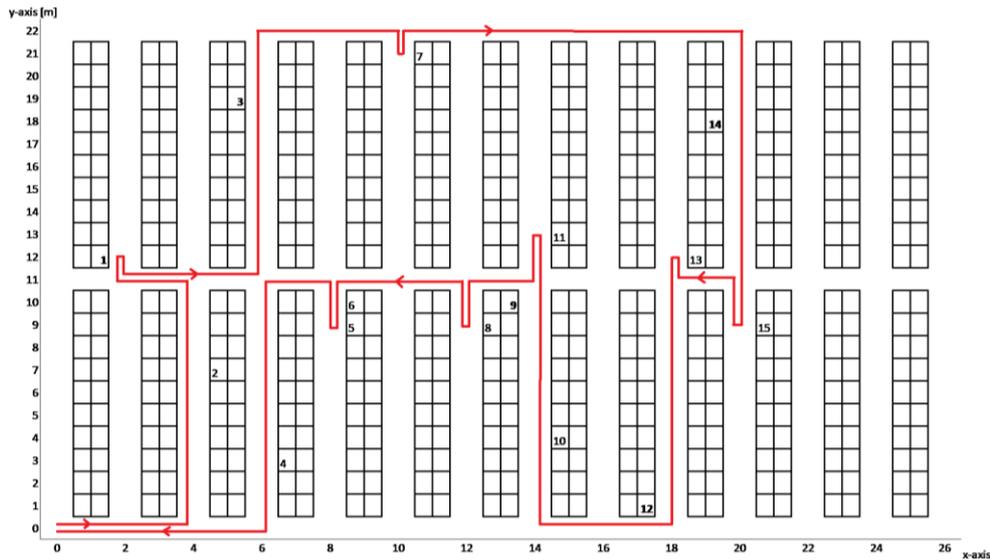


Figure 5: The order picking situation and picking path generated by the brute-force algorithm.

3.4 Picking path generated by GA

In the GA option, the Basic GA Methodology was used to search for the best solution (shortest path). Most simulations reached a solution which represents a picking path of 190 - 200 m long. The average of the best picking path generated by ten simulations is 193.2 m. The best solution was reached in the 8th simulation and is equal to a 178 m long picking path.

Given the rather implausible results of the Basic GA Methodology, there was reason to improve it. Elitist selection was used in this study to improve the Basic GA Methodology. The notion behind this was that the Basic GA Methodology was not able to keep the best solutions. By adding elitist selection to this methodology, the two best individuals from the actual generation are carried over to the next generation unaltered. This modification also decreased P_{cross} from 1 to 0.8. In the second round of simulations, this modified GA methodology was tested.

Fig. 6 and Table I show the results of the simulations from the modified GA methodology (Basic GA Methodology with Elitist Selection). In this case, the evolution of the best solution in individual simulations and their average continuously improves and achieved better results on average in comparison to the Basic GA Methodology.

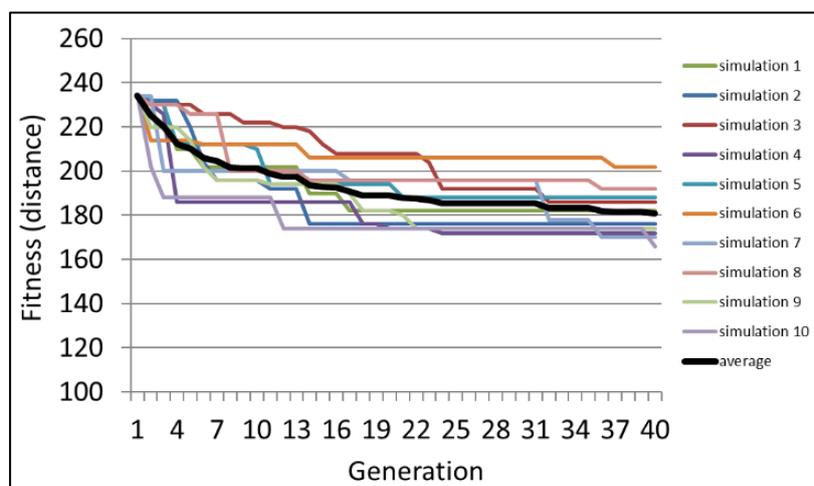


Figure 6: Evolutionary process of Basic GA Methodology with Elitist Selection (10 simulations and average).

Table I: Best solutions generated in the second ten simulations using Basic GA Methodology with Elitist Selection.

Simulation	1	2	3	4	5	6	7	8	9	10
Reached best solution [m]	182	176	186	172	188	202	170	192	174	166

The results of the second simulation show that despite the improvement of the GA with the elitist selection, its ability to improve the heuristic solution varied from a 10.75 % improvement to an 8.60 % lag behind (Fig. 7 right). On average, it can improve the heuristic solution by 2.8 %. Despite these better results, it is somewhat limited in comparison to the results of an experienced picker or brute-force algorithm (Fig. 7).

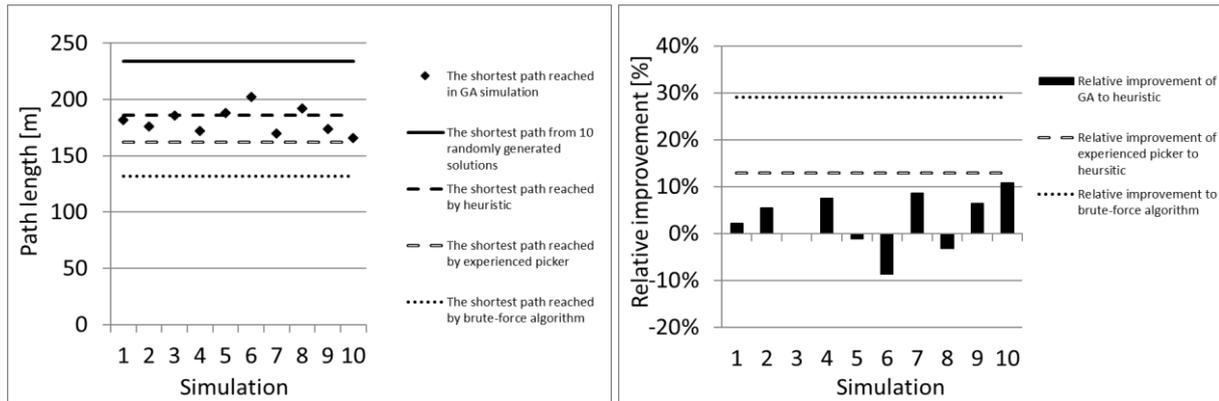


Figure 7: Comparison of the modified GA with other routing policies (shortest path – left, relative improvement – right).

To test the effects of further modifications of Basic GA Methodology with Elitist Selection on the simulation results, the population was expanded to 40 units and the number of generations to 50. P_{cross} was changed slightly to 0.95 and P_{mut} decreased to 0.1. We would call this GA methodology Basic GA Methodology with Elitist Selection – Expanded. This modification is in line with Hassanat et al. [25]. They conclude, based on relevant surveyed studies, that diversity in the initial population, high crossover rates and low mutation rates was the best situation to apply GA.

Fig. 8 and Table II show the results of the simulations from the expanded GA methodology (Basic GA Methodology with Elitist Selection – Expanded). In this case, the evolution of the best solution in individual simulations and their average continuously improves and achieved better results on average in comparison to the Basic GA Methodology with Elitist Selection.

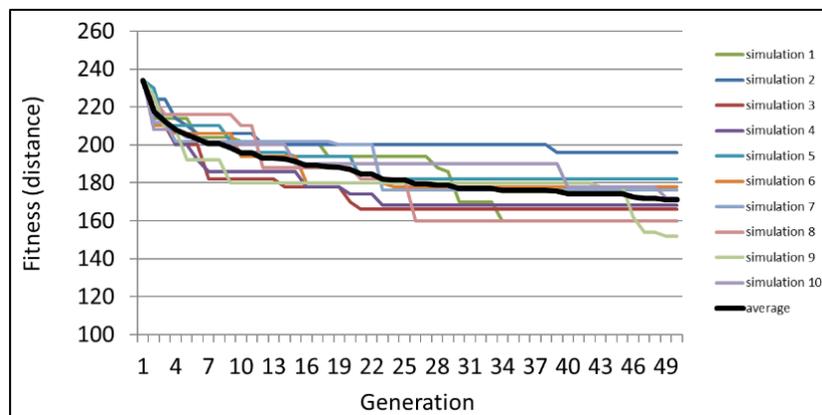


Figure 8: Evolutionary process of Basic GA Methodology with Elitist Selection – Expanded (10 simulations and average).

Table II: Best solutions generated in the third ten simulations using Basic GA Methodology with Elitist Selection – Expanded.

Simulation	1	2	3	4	5	6	7	8	9	10
Reached best solution [m]	160	196	166	168	182	178	176	160	152	172

The results of the third simulation show that despite the further improvement of the GA with elitist selection by the expansion of the population to 40 units and the number of generated populations to 50, its ability to improve the heuristic solution varied from an 18.28 % improvement to a 5.38 % lag behind (Fig. 9 right). On average, it can improve the heuristic solution by 8.06 %. Despite these better results, it is comparable only to an experienced picker, but still somewhat limited in comparison to the results of brute-force algorithm (Fig. 9). Further tests of other modification of mutation and crossover operator show lower average improvements in comparison to the heuristic as presented in the third simulation round. For example average improvement by 7.31 % for $P_{cross} = 0.5$ and $P_{mut} = 0.025$, 4.95 % ($P_{cross} = 0.95$, $P_{mut} = 0.025$), 0.11 % ($P_{cross} = 0.5$, $P_{mut} = 0.0125$) and also average deterioration by 0.22 % when $P_{cross} = 0.95$ and $P_{mut} = 0.0125$.

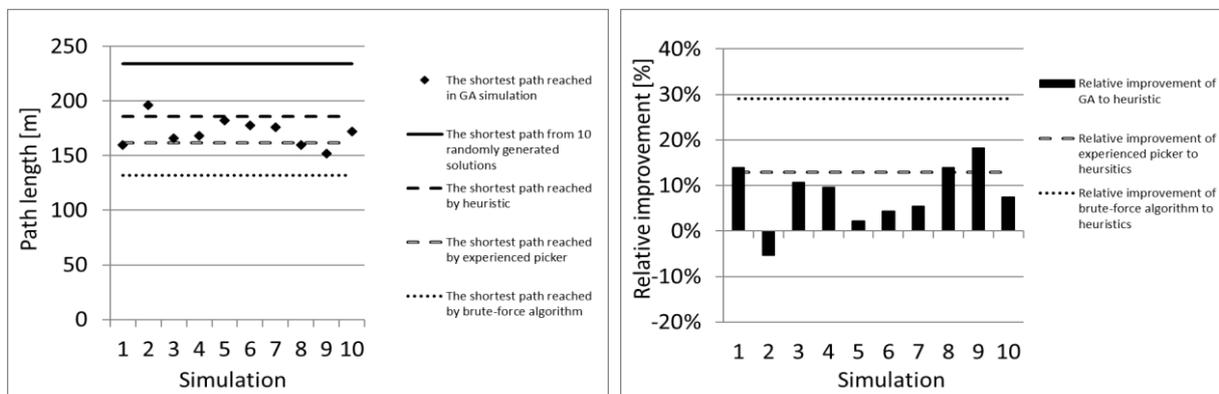


Figure 9: Comparison of the expanded GA with other routing policies (shortest path – left, relative improvement – right).

4. DISCUSSION

In the current study, the difference between the heuristic and the optimal solution is around 29 %. It is consistent with, e.g., de Koster and Der Poor [26] who have shown a reduction in the travel time per route of between 7 % and 34 % when comparing the optimal and heuristic (S-shape) solution.

It should be mentioned that the order size (15 items) is a variable with a strong influence on the complexity of the problem, including the number of potential solutions. In general, it could be assumed that with the rise of the order size, the ability of an experienced picker to make a warehouse grid in his mind would decrease. Similarly, with even a small rise in the order size, the ability of the brute-force algorithm to calculate the solution will reach its limits due to the dramatic increase in the number of potential solutions. In the simple calculation test, the following results were obtained. The calculation times in the case of 1 – 14 items were in the order of seconds or less (e.g., for 14 items, it was around 16 seconds). The calculation time for 15 items (our case) was around one minute. The calculation time for 16 items was in the order of minutes (around 8 minutes). The calculation time for 17 items was close to one hour. The calculation time for 18 items was not calculated, but it is expected to be so high that it would be rendered useless. This kind of simple sensitivity analysis indicates that with an increasing order size, the solutions generated by GA in comparison to the investigated routing strategies become more useful.

The second limitation of the study is the size of population and the number of generations. According to reasonable computing times on standard PC configuration (processor: i7, 2,80 GHz; RAM: 4 GB), which equals for the third simulation round on average 4 minutes and 20 seconds, the size of population was limited to 40 and the number of generations was limited to 50. In general, it can be assumed that there is some potential to improve the average solution if the size of population increases or more generations (e.g. 100) are generated, although this was not investigated.

5. CONCLUSION

The results of the investigated routing strategies under given assumptions (middle size dual-zone warehouse, order size – 15 items) show the dominance of the brute-force algorithm in comparison to the experienced picker, GA and simple heuristics.

GA is an optimization method which needs modification in dealing with picking path optimization problems and under given assumptions can generate better solutions than simple heuristics. In the case of used traditional GA (Basic GA Methodology), its performance (on average) lags behind all other investigated routing strategies, i.e., the heuristic, experienced picker and brute-force algorithm. In the case of used modified GA (Basic GA Methodology with Elitist Selection), in which elitist selection was incorporated, its performance (on average) is better than the heuristics. However, it still lags behind the experienced picker and brute-force algorithm. In the case of used extended GA (Basic GA Methodology with Elitist Selection – Expanded), in which the size of population was increased to 40 and the number of generations to 50, its performance (on average) is much better than the heuristics, comparable to the experienced picker. However, also this GA still lags behind brute-force algorithm. These results support the idea that modifications can improve traditional GA performance; see in [3, 20, 25]. In the current study, the incorporation of elitist selection improved the results of used traditional GA methodology by 6.53 % on average. Further extension of population size and number of generation improved the results of the traditional GA methodology by additional 4,96 % on average. By modification and extension, the results of the GA were improved by 11,49 % on average. These results show quite significant sensitivity of GA results on used selection operator, size of population and number of generations.

The sensitivity analysis indicates that if order size is only slightly increased (e.g., from 15 to 18 items) the computational times for brute-force algorithm on a standard PC will rise dramatically. Consequently, the solutions generated by GA (as well as by simple heuristics) become more useful.

ACKNOWLEDGEMENT

This contribution was supported by the research grant KEGA 002TUKE-4/2020 Implementation of intelligent technology and advanced technologies to support transformation processes and future product design.

REFERENCES

- [1] Lu, W.; McFarlane, D.; Giannikas, V.; Zhang, Q. (2016). An algorithm for dynamic order-picking in warehouse operations, *European Journal of Operational Research*, Vol. 248, No. 1, 107-122, doi:[10.1016/j.ejor.2015.06.074](https://doi.org/10.1016/j.ejor.2015.06.074)
- [2] De Koster, R.; Le-Duc, T.; Roodbergen, K. J. (2007). Design and control of warehouse order picking: a literature review, *European Journal of Operational Research*, Vol. 182, No. 2, 481-501, doi:[10.1016/J.EJOR.2006.07.009](https://doi.org/10.1016/J.EJOR.2006.07.009)

- [3] Beroule, B.; Grunder, O.; Barakat, O.; Aujoulat, O. (2017). Order picking problem in a warehouse hospital pharmacy, *IFAC-PapersOnLine*, Vol. 50, No. 1, 5017-5022, doi:[10.1016/j.ifacol.2017.08.910](https://doi.org/10.1016/j.ifacol.2017.08.910)
- [4] Cheng, C.-Y.; Chen, Y.-Y.; Chen, T.-L.; Yoo, J. J.-W. (2015). Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint order batching and picker routing problem, *International Journal of Production Economics*, Vol. 170, Part C, 805-814, doi:[10.1016/j.ijpe.2015.03.021](https://doi.org/10.1016/j.ijpe.2015.03.021)
- [5] Bartholdi, J. J.; Hackman, S. T. (2014). *Warehouse & Distribution Science*, The Supply Chain and Logistics Institute, Atlanta
- [6] Key, R.; Dasgupta, A. (2015). Warehouse pick path optimization algorithm analysis, *Proceedings of the 2015 International Conference on Foundations of Computer Science*, 63-69
- [7] De Santis, R.; Montanari, R.; Vignali, G.; Bottani, E. (2018). An adapted ant colony optimization algorithm for the minimization of the travel distance of pickers in manual warehouses, *European Journal of Operational Research*, Vol. 267, No. 1, 120-137, doi:[10.1016/j.ejor.2017.11.017](https://doi.org/10.1016/j.ejor.2017.11.017)
- [8] Taherdangkoo, M.; Paziresh, M.; Yazdi, M.; Bagheri, M. (2013). An efficient algorithm for function optimization: modified stem cells algorithm, *Central European Journal of Engineering*, Vol. 3, No. 1, 36-50, doi:[10.2478/s13531-012-0047-8](https://doi.org/10.2478/s13531-012-0047-8)
- [9] Burinskiene, A. (2015). Optimising forklift activities in wide-aisle reference warehouse, *International Journal of Simulation Modelling*, Vol. 14, No. 4, 621-632, doi:[10.2507/IJSIMM14\(4\)5.312](https://doi.org/10.2507/IJSIMM14(4)5.312)
- [10] Wang, C. L.; Li, S. W. (2018). Hybrid fruit fly optimization algorithm for solving multi-compartment vehicle routing problem in intelligent logistics, *Advances in Production Engineering & Management*, Vol. 13, No. 4, 466-478, doi:[10.14743/apem2018.4.304](https://doi.org/10.14743/apem2018.4.304)
- [11] Cao, Q. K.; Yang, K. W.; Ren, X. Y. (2017). Vehicle routing optimization with multiple fuzzy time windows based on improved wolf pack algorithm, *Advances in Production Engineering & Management*, Vol. 12, No. 4, 401-411, doi:[10.14743/apem2017.4.267](https://doi.org/10.14743/apem2017.4.267)
- [12] Gholamian, M. R.; Heydari, M. (2017). An inventory model with METRIC approach in location-routing-inventory problem, *Advances in Production Engineering & Management*, Vol. 12, No. 2, 115-126, doi:[10.14743/apem2017.2.244](https://doi.org/10.14743/apem2017.2.244)
- [13] Dujmešić, N.; Bajor, I.; Rožić, T. (2018). Warehouse processes improvement by pick by voice technology, *Technical Gazette*, Vol. 25, No. 4, 1227-1233, doi:[10.17559/TV-20160829152732](https://doi.org/10.17559/TV-20160829152732)
- [14] Wan, F. C.; Chen, D.; Wu, Y. Q. (2012). Based on genetic algorithm research of stocker picking shortest path, *Applied Mechanics and Materials*, Vol. 155, 186-190, doi:[10.4028/www.scientific.net/AMM.155-156.186](https://doi.org/10.4028/www.scientific.net/AMM.155-156.186)
- [15] Chang, F.-L.; Liu, Z.-X.; Xin, Z.; Liu, D.-D. (2007). Research on order picking optimization problem of automated warehouse, *Systems Engineering – Theory & Practice*, Vol. 27, No. 2, 139-143, doi:[10.1016/S1874-8651\(08\)60015-0](https://doi.org/10.1016/S1874-8651(08)60015-0)
- [16] Hsu, C.-M.; Chen, K.-Y.; Chen, M.-C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms, *Computers in Industry*, Vol. 56, No. 2, 169-178, doi:[10.1016/j.compind.2004.06.001](https://doi.org/10.1016/j.compind.2004.06.001)
- [17] Silva, C. A.; Sousa, J. M.; Runkler, T.; Sá da Costa, J. M. G. (2005). A logistic process scheduling problem: genetic algorithms or ant colony optimization?, *IFAC Proceedings Volumes*, Vol. 38, No. 1, 206-211, doi:[10.3182/20050703-6-cz-1902.01518](https://doi.org/10.3182/20050703-6-cz-1902.01518)
- [18] Deng, X.-Y. (2014). A parallel optimization algorithm for steel plate pick-up operation scheduling problem, *International Journal of Simulation Modelling*, Vol. 13, No. 3, 323-334, doi:[10.2507/IJSIMM13\(3\)CO11](https://doi.org/10.2507/IJSIMM13(3)CO11)
- [19] Ma, Y.; Li, Z.; Yun, W. (2015). Fast genetic algorithm for pick-up path optimization in the large warehouse system, Tan, Y.; Shi, Y.; Buarque, F.; Gelbukh, A.; Das, S.; Engelbrecht, A. (Eds.), *Advances in Swarm and Computational Intelligence, ICSI 2015, Lecture Notes in Computer Science*, Vol. 9142, Springer, Cham, 38-44, doi:[10.1007/978-3-319-20469-7_5](https://doi.org/10.1007/978-3-319-20469-7_5)
- [20] Moon, C.; Kim, J.; Choi, G.; Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints, *European Journal of Operational Research*, Vol. 140, No. 3, 606-617, doi:[10.1016/S0377-2217\(01\)00227-2](https://doi.org/10.1016/S0377-2217(01)00227-2)

- [21] Zhu, J.; Zhang, H.; Guo, J.; Zhou, L. (2015). Picking path optimization in dual zone type warehouse based on genetic algorithms, *International Journal of Economic Behavior and Organization*, Vol. 3, No. 3, 41-46, doi:[10.11648/j.ijebo.20150303.13](https://doi.org/10.11648/j.ijebo.20150303.13)
- [22] Floyd, R. W. (1962). Algorithm 97: shortest path, *Communications of the ACM*, Vol. 5, No. 6, 345, doi:[10.1145/367766.368168](https://doi.org/10.1145/367766.368168)
- [23] Heap, B. R. (1963). Permutations by interchanges, *The Computer Journal*, Vol. 6, No. 3, 293-298, doi:[10.1093/comjnl/6.3.293](https://doi.org/10.1093/comjnl/6.3.293)
- [24] Kongar, E.; Gupta, S. M. (2002). Genetic algorithm for disassembly process planning, *Environmentally Conscious Manufacturing II*, Vol. 4569, 54-62, doi:[10.1117/12.455264](https://doi.org/10.1117/12.455264)
- [25] Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Surya Prasath, V. B. (2019). Choosing mutation and crossover ratios for genetic algorithms – a review with a new dynamic approach, *Information*, Vol. 10, No. 12, Paper 390, 36 pages, doi:[10.3390/info10120390](https://doi.org/10.3390/info10120390)
- [26] De Koster, R.; Van Der Poort, E. (1998). Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions, *IIE Transactions*, Vol. 30, No. 5, 469-480, doi:[10.1080/07408179808966487](https://doi.org/10.1080/07408179808966487)