COMPUTER METHODS IN MATERIALS SCIENCE

2022, vol. 22, no. 3, 157–166
https://doi.org/10.7494/cmms.2022.3.0783

Analysis of modification of the evolutionary algorithm for sequencing production tasks

Piotr Ciepliński*, Sławomir Golak, Tadeusz Wieczorek

Department of Industrial Computer Science, Faculty of Materials Engineering, Silesian University of Technology, Katowice, Poland.

Abstract

Evolutionary algorithms are one of the heuristic techniques used to solve task sequencing problems. An important example of such a problem is the issue of sequencing production tasks. The combinatorial optimization of task sequences allows the minimization of the cost or time of a set of production tasks by reducing the components of these values which are present in the transitions between tasks.

This paper aims to analyze the influence of the production nature expressed by a set of production task parameters and a definition of the task transition cost on the effectiveness of the modification of the evolutionary algorithm based on new directed stochastic mutation operators. The research carried out included the influence of the space dimension of the task parameters, the number of levels of the value of the cost function, and a definition of this function. The results obtained allow us to assess the effectiveness of the directed mutation in task sequencing for productions of various natures.

Keywords: evolutionary algorithm, task sequencing, mutation operator

1. Introduction

The most popular representative of the problem of task sequencing is the traveling salesman problem (TSP), which is a very widely analyzed issue in the field of computer science (Kentli et al., 2013; Shi & Liu, 2021; Wan et al., 2020). However, this problem in the standard approach is a little different from the problem of sequencing production jobs. First, 'cities' in TSP are usually described with only two parameters that describe their location in a two-dimensional space. Additionally, the distance function is continuous-valued. There are some differences when it comes to sequencing production tasks. A production task is often described by more parameters, and some of these parameters are discrete ones. Also, the function of the cost of transition between tasks (most often related to retooling the production line) is usually discontinuous and has very different definitions (Kentli et al., 2013).

A representative example is steel production in the continuous casting process, shown in Figure 1. By sequencing tasks, we can minimize two values: the pause time between tasks and the financial cost of implementing the transition between tasks. Since these values are related to each other, we will refer to them by the collective term 'cost' for the sake of simplicity. Of course, there is also a cost for completing the tasks themselves, but this cost cannot be reduced by optimizing the task sequence (Ramstorfer & Delane de Souza, 2022).

* Corresponding author: piotr.cieplinski@polsl.pl
ORCID ID's: 0000-0001-9056-0367 (P. Ciepliński), 0000-0002-9325-3407 (S. Golak), 0000-0002-9310-9798 (T. Wieczorek)
© 2022 Authors. This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License requiring that the original work has been properly cited.

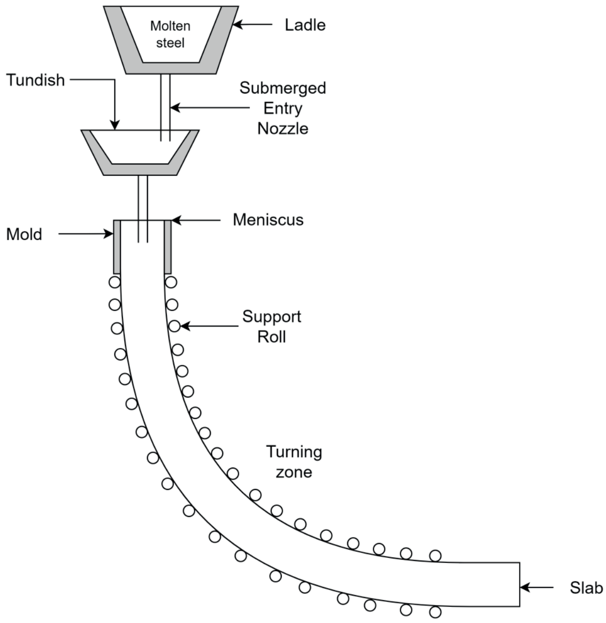

Fig. 1. Continuous steel casting process

The ingot tonnage, steel grade, and size of the ingot describes each continuous steel casting task. The first parameter is a continuous parameter that proportionally affects the constant component of the task cost. We cannot influence the constant component by sequencing production tasks. Another parameter, steel grade, is essentially a set of continuous parameters that describe the chemical composition of the steel. In the case of these parameters, we have the possibility of influencing the variable cost of task implementation related to the cost of transition between tasks. There is no additional cost of a change in steel grade when switching between grades with a similar chemical composition (Ramstorfer & Delane de Souza, 2022).

Otherwise, there is a need to remove the steel remnants from the previous smelting, which means a step (discrete) increase in the cost of transition between tasks. The cost increases abruptly again when the ladle lining has to be replaced. Thus, we have two discrete cost levels for a continuous chemical composition change. The last task parameter, the ingot format parameter, is a discrete parameter with values selected from a set of formats (e.g. round, octagonal, square, rectangular). Each format change implies a fixed cost of rebuilding the casting system output (Vijayaram et al., 2014).

Of course, there is a huge amount of variety in the types of production, and there are also productions where the parameters are continuous, and the cost function is continuous. We are then dealing with an issue analogous to the standard traveling salesman problem.

A huge number of task sequencing algorithms have been created to date, with the lion's share dedicated to the standard traveling salesmen problem. These include exact algorithms (e.g. integer linear formulations or branch-and-bound algorithms) (Laporte, 1992), the ant colony optimization algorithm (Garcia-Martínez et al., 2007), the 2-opt algorithm (Englert et al., 2014), the greedy algorithm (Bang-Jensen et al., 2004), the cover tree traversal algorithm (Zhang & Xu, 2018) or local search algorithms (Levin & Yovel, 2014). One of the most popular types of task sequence optimization algorithms are evolutionary algorithms (Gao et al., 2020). Although they often have lower performance than the others mentioned, they provide considerable flexibility in terms of the possible task parameters to be considered. We can use continuous and discrete parameters and set elements. In the case of some productions, in addition to sequencing, they can simultaneously optimize the parameters of individual tasks (not just the order of transitions between tasks). What is important in the practice of implementing task sequencing systems in production plants, they allow for a simple consideration of specific customer requirements in the optimization criterion and constraints.

The paper concerns an analysis of the influence of the parameters of a production task and a cost function definition on the effectiveness of the new modification of the evolutionary algorithm. The evolutionary algorithm is usually based on genetic operators of crossover and mutation. However, in combinatorial optimization of the sequence, basic crossover operators strongly degrade the sequence of tasks, and it is necessary to introduce elaborate mechanisms to repair these sequences, often with a low degree of efficiency. Although more advanced operators, such as partially matched crossover (PMX) or cycle crossover (CX) (Nirmala & Ramprasad, 2012), do not need a repair function, the quality of their implementation may affect experiment results. For this reason, the modification of the evolutionary algorithm described in the paper completely abandons the crossover operator in favor of the more effective version of the mutation operator. An additional advantage of such an approach is the presence in the obtained results of only the real influence of the analyzed mutation operators, without the influence of the crossover operator imperfection.

Many different implementations of mutation operators have been developed so far for the problem of task sequencing, e.g., displacement mutation (Michalewicz, 2013), exchange mutation (Banzhaf, 1990), insertion mutation (Fogel, 1988), simple inversion mutation (Holland, 1992), inversion mutation (Fogel, 1990), scramble mutation (Syswerda, 1991), inverted exchange mutation, inverted displacement mutation (Deep & Mebrahtu, 2011), twors mutation, center inverse mutation (CIM), reverse sequence mutation (RSM), throas mutation, partial shuffle mutation (PSM) (Abdoun et al., 2012). The listed mutation operators, despite their diversity, have one common feature: the determination of points in the task chain for exchanging segments or tasks is completely random.

Only recently have solutions been developed where mutations account for the cost value of connecting between tasks. Such methods that consider the cost of transition between tasks have been presented recently in the paper (Hassanat et al., 2016), where several presented mutations are based on choosing 'the worst task', defined as the task with the highest cost of transition. Mutable tasks are selected in a completely deterministic manner.

The paper analyzes a new kind of mutation developed in our department, the probability of which is related to the cost of transition between tasks. However, the location of the mutation is chosen randomly, as opposed to the completely deterministic solution mentioned. Here, a solution analogous to the selection operation in the evolutionary algorithm is used. By selecting the coefficients of selection power, it is possible to obtain a completely random (blind) mutation, a completely deterministic mutation, or a stochastic mutation characteristic for the new solution, taking into account the cost of connecting between tasks.

The main goal of this paper is to analyze the influence of the nature of scheduled production expressed by a set of production task parameters and a definition of the task transition cost on the effectiveness of the new modification of the evolutionary algorithm based on new directed stochastic mutation operators. The presented research included the analysis of the influence of the number of task parameters (solution space dimension), the level of discontinuity of the cost function and its definition on the obtained results. These investigations allow us to assess the influence of the nature of scheduled production on the effectiveness of the algorithm developed.

2. A modified evolutionary algorithm for task sequencing

The algorithm analyzed is a modification of the standard evolutionary algorithm (Fig. 2). The algorithm operates on a population of individuals which represents different versions of the task sequence (Malik, 2019).

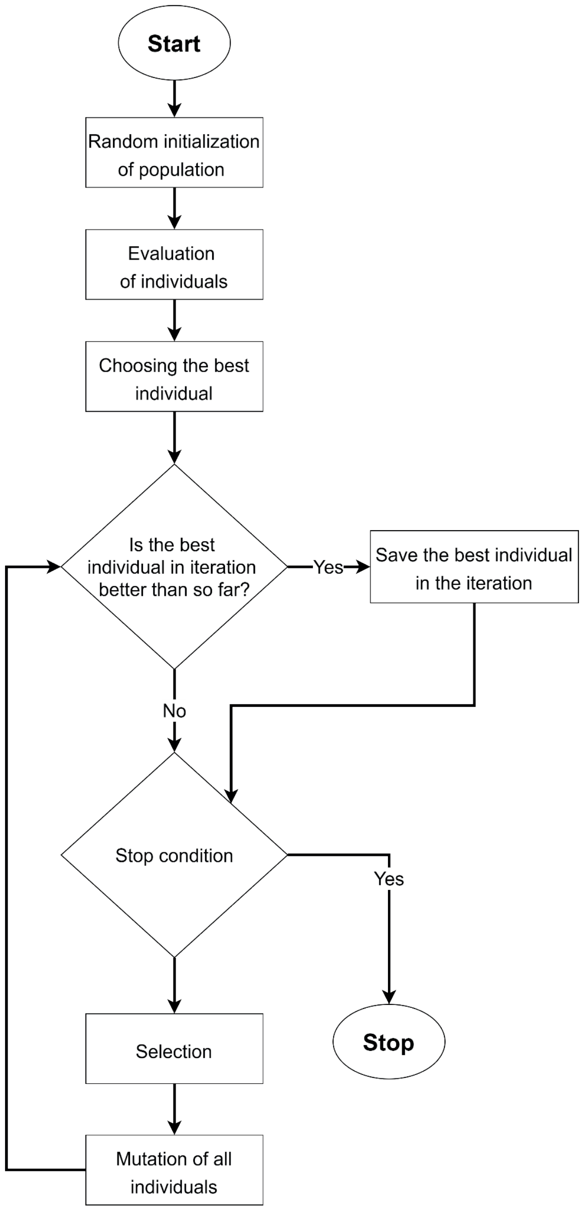

Fig. 2. Flow diagram of the evolutionary algorithm

The initial population is generated randomly. Then, the value of the fitness function is determined for each individual in the population. Before the main part of the algorithm starts, the stop condition is checked. The condition may refer to the number of iterations of the main algorithm loop, the global optimization time, and/or the achievement of the assumed value of the fitness function. However, in the case of task sequencing, we usually cannot estimate such a value. The result of the algorithm's work is the best individual found during all iterations of the algorithm.

Then, a new generation of individuals is created in the selection operation. Among the many selection methods, proportional (1) and ranking (2) selection are the most popular (Malik, 2019).

The main difference between them is that the former takes into account the exact value of the fitness function in estimating the selection probability of an individual to the new generation, and the latter is based solely on its position in the ranking.

$$P_i = \left(1 - \frac{F_i - F_{\min}}{F_{\max} - F_{\min}}\right)^a \tag{1}$$

$$P_i = \left(1 - \frac{r_i - 1}{N - 1}\right)^a \tag{2}$$

where: P_i – the probability of selection of the i-th individual; F_i – fitness of the individual; r_i – individual position in the ranking (the best individual – position 1); F_{min}, F_{max} – fitness of the best and worst individuals; N – population size; a – factor of selection strength.

Due to the most frequent use of the roulette wheel selection method, the determined probabilities are usually normalized. Then, a distribution of the frequency of individuals in the new generation is obtained which is similar to the distribution of the probability of selection.

$$P_i^{(n)} = \frac{P_i}{\sum_j P_j} \tag{3}$$

where $P_j^{(n)}$ is the normalized selection probability of the j-th individual.

The a factor allows one to control the strength of selection. In the case of a value of zero, the selection is blind, and all individuals have the same chance to pass to the new generation. In the case of assuming the infinite value of this coefficient (in practice, very large), only the best individual passes to the new generation. The selection is then completely deterministic. The value in the range (0, ∞) signifies the directed random selection. Usually, it is assumed to be equal to 1.

After selection, genetic operators modify individuals in the new generation. As mentioned in the introduction to this paper, the discussed algorithm does not use the crossover operator, because it may cause a large degradation of the task chain or needs to implement a more complicated crossover operator (Malik, 2019; Nirmala & Ramprasad, 2012). The adopted methodology allowed us to isolate only the influence of the developed and analyzed mutation operator in the results.

Before discussing the new method of mutation, it is necessary to introduce the method of coding the task sequence in the individual's chromosome. It is a cyclical graph (Fig. 3) that will be opened only after the optimization is completed at the location of the most expensive connection. In the diagram, the cost of a connection (transition between tasks) is expressed as its length. The cut connection does not have an arrow.

The figure shows an example of such a chromosome for the continuous casting of steel. Within this process, each production task has only two parameters that affect the cost of connections: steel grade and ingot format. Regardless of the number and type of task parameters, the chromosome contains the number of genes equal to the number of tasks and encoded by integers. As shown in the figure, the cost of connection is a function of the parameters of adjacent tasks – here, changes in the steel grade and/or the ingot format.

In the case of a real continuous steel casting process, the function can be defined with a set of simple rules of the type `IF GRADE1 <> GRADE2 AND FORMAT1 = FORMAT2 THEN COST = 1000.00 USD.` In the case of other types of production, this function must often be described not only by rules, but also by linear or non-linear relationships that take into account the values of the task parameters.

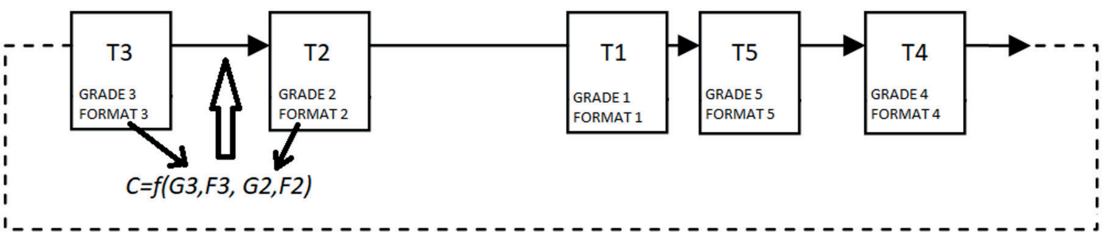

Fig. 3. Chromosome coding for the task sequence – an example for continuous casting of steel

As discussed in the introduction to the paper, in the method of mutation developed thus far, regardless of the way tasks are rearranged in their course, the location of the sequence cut is most often random (Abdoun et al., 2012; Banzhaf, 1990; Deep & Mebrahtu, 2011; Fogel, 1988, 1990; Holland, 1992; Michalewicz, 2013; Syswerda, 1991) or, as is the case in recent studies, completely deterministic (the worst connection) (Hassanat et al., 2016). Our paper analyses a mutation, the location of which is randomized but takes into account connection costs.

In the case of new mutation operators, analogically for selection methods, an idea and a mathematical description were used. Two variants were proposed: one based on the proportional selection method (4) and the ranking selection method (5).

$$P_j^{(C)} = \left(1 - \frac{C_j - C_{\min}}{C_{\max} - C_{\min}}\right)^a \tag{4}$$

$$P_j^{(C)} = \left(1 - \frac{r_j^{(C)} - 1}{M - 1}\right)^a \tag{5}$$

where: $P_j^{(C)}$ – cutting probability of the j-th connection; C_j – cost of the connection; $r_j^{(C)}$ – position of connection in the ranking (the best connection – position 1); C_{min}, C_{max} – cost of the most expensive and cheapest connection); M – the number of connections equal to the number of tasks (in the cyclic graph); a – factor of the randomness factor.

```
current_task_sequence := [T3, T2, T1, T5, T4, …]

new_task_sequence := []
chosen_indexes := []


repeat:

   x := random positive number |0,1|
   if (x < P_j) then
      chosen_indexes.push(j)
until (chosen_indexes.length == 3)


chosen_indexes.sort()


for i := chosen_indexes[0] to chosen_indexes[1]:

  new_task_sequence.push(current_task_sequence[i])

end for


for i := chosen_indexes[2] to current_task_sequence.length:

  new_task_sequence.push(current_task_sequence[i])

end for


for i := 0 to chosen_indexes[0]:

  new_task_sequence.push(current_task_sequence[i])

end for


for i := chosen_indexes[1] to chosen_indexes[2]:

  new_task_sequence.push(current_task_sequence[i])

end for
```

Fig. 4. Pseudocode for selecting three connections for mutation and the task sequence rearrangement

Factor a equals to 0 makes it possible to obtain a fully random mutation (hereinafter referred to as a blind mutation) which is well-known in the literature (Abdoun et al., 2012; Banzhaf, 1990; Deep & Mebrahtu, 2011; Fogel, 1988, 1990; Holland, 1992; Michalewicz, 2013; Syswerda, 1991). Taking a very large value (approximately infinite) results in a deterministic mutation (Hassanat et al., 2016) – always choosing the connection with the highest cost.

Of course, the methods of the mutation have not been defined in the form of formulas (4) and (5) so far. Instead, only the mechanics of their implementation are described without explicitly defining the probability value of the separation of individual connections in the sequence. Basing our mutation on an explicit probability makes it possible to take into account the cost of the connection while maintaining the randomness of the sequence break controlled by the parameter a. Such an intermediate implementation was not possible in the above-mentioned types of mutation, which were either completely random or fully deterministic.

The method of mutation is complemented by the way tasks are arranged in sequence. After determining the cut probabilities based on formulas (4) and (5) and their normalization, three cut points are selected for the cyclic task graph. The process of selecting the three connections for mutation is presented in Figure 4. Three sequence parts are obtained, and then two of them are swapped (without changing the order of tasks in these parts). After the mutation has been performed on the population, the next cycle of the algorithm begins.

The investigations described in the paper had two goals. The first goal was to determine how the degree of mutation determinism and randomness affects the effectiveness of the evolutionary task sequencing algorithm. The second was to determine the influence of the nature of sequenced production on this effectiveness. These goals were achieved on the theoretical, systematized level by assessing the impact of the dimensionality of the task parameter space, the level of discretization, and the definition of the cost function for transitions between tasks. As a result, it was possible to analyze the entire spectrum of existing types of production.

3. Experiments and discussion of the results

The aim of the experiments was to verify the hypothesis that the use of the proposed mutation operators would increase the efficiency of the evolutionary algorithm for the solution of the problem of production task sequencing for any type of production.

In order to enable the most extensive analysis, experiments were based on a virtual production model, which allows the freedom to select a set of parameters and define the cost of transition between tasks. Of course, such a model is only a certain theoretical representation of real production processes.

The research was carried out for a synthetic sequence of tasks described by a set of continuous parameters. Parameter values were normalized and ranged from 0 to 1 (uniform distribution). The adaptation function was the sum of the transition costs between all tasks in the sequence, excluding the cost of the most expensive connection. As mentioned in the previous section of this paper, after the optimization phase, the cyclic graph representing the chromosome is cut open at this location.

For the study, the cost of transition between tasks was defined based on the Minkowski distance (Singh et al., 2013):

$$C_j = \left(\frac{\sum_{k=1}^{K} \left| p_{j,k} - p_{j+1,k} \right|^b}{K} \right)^{\frac{1}{b}} \tag{6}$$

where: C_j – cost of the j-th connection; $p_{j,k}$ – the k-th parameter of the j-th task; K – number of parameters; b – is the order of the norm.

To compare the results for a different number of parameters and different exponents of the Minkowski distance, a normalizing divisor K was introduced. This made it possible to equalize the maximum cost of the connection to 1.

The research used a discretized connection cost function $C_j^{(d)}$ with a controlled number of levels:

$$C_j^{(d)} = \frac{\lfloor (d-1)C_j + 0.5 \rfloor}{(d-1)} \tag{7}$$

where d is the number of discretization levels (min. 2).

The total cost of executing a sequence (the fitness function for the optimization) is defined by the formula:

$$C = \sum_{j}^{N} C_j^{(d)} - \max{}_j C_j^{(d)} \tag{8}$$

where: C – total cost of a sequence; N – count of connections between tasks.

Note that after optimizing according to the formula, the cost of the most expensive link is subtracted since this is where the cyclic chromosome graph will always be cut open.

The first experiment concerned the influence of the number of parameters of a task on the effectiveness of the proposed adaptive mutation. The results were compared with the classic blind mutation and with a completely deterministic mutation. In the case of the adaptive mutation, a variant modeled on proportional and ranking selection was taken into account. In the experiment, the cost of transition between tasks was defined based on the Euclidean distance (exponent 2 at the Minkowski distance). The cost function was a continuous function. The analysis was performed for a task sequence with a size of 1000 and the number of iterations of the evolutionary algorithm equal 1000 and 2000. The series of experiments carried out showed that for 2000 iterations, all of the analyzed algorithms achieved the minimum value of the criterion, and we can observe the stabilization of the solution. To illustrate the dynamics of optimization, we also present the state of the optimization in the middle of this period (for 1000 iterations).

As can be seen in Figure 5, the increase in the number of parameters describing the task increases the achieved criterion value. The deterioration of the optimization conditions is related to the increase in the statistical distance between tasks caused by the increase in the dimension of parameter space. This phenomenon has not been fully eliminated by the divisor proposed in the cost definition (6). For the analyzed range of the number of parameters, it can be observed that the values of the criteria for each type of mutation tend to its individual asymptote. However, it can be concluded from this experiment that regardless of the number of parameters describing the task for sequenced production, the new mutation operators allow us to get a lower global cost (criterion). The ranking mutation is distinguished here by its effectiveness. The completely deterministic mutation turned out to be the worst and in this case this is likely due to the increased risk of being stuck at a local minimum during optimization. This is confirmed by the lack of improvement between the level of the criterion after 1000 (Fig. 5a) and after 2000 (Fig. 5b) iterations of the optimization process.

The second experiment concerned the impact of the number of levels of values of the cost function on the effectiveness of the analyzed mutations. This allows one to reflect on different types of production. For example, in continuous steel casting, any change in ingot format always generates a fixed single rebuild cost. In turn, in the case of production based on machining, changes in tool settings often translate into a proportional, continuous increase in the time cost of transition between tasks. As in the previous experiment, the Euclidean definition of the cost of transition between tasks was adopted, and the research was carried out for a sequence of 1000 tasks.

The analysis started with the lowest possible number of levels of the cost function value. Two levels of cost mean only two possibilities: There is no cost, and there is a cost of transition (as for a format change in the steel casting process). The analysis was completed at 64 levels because there were no further changes for each type of mutation. It can be assumed that a further increase in the number of levels until a continuous function is obtained will no longer affect the obtained criterion value.

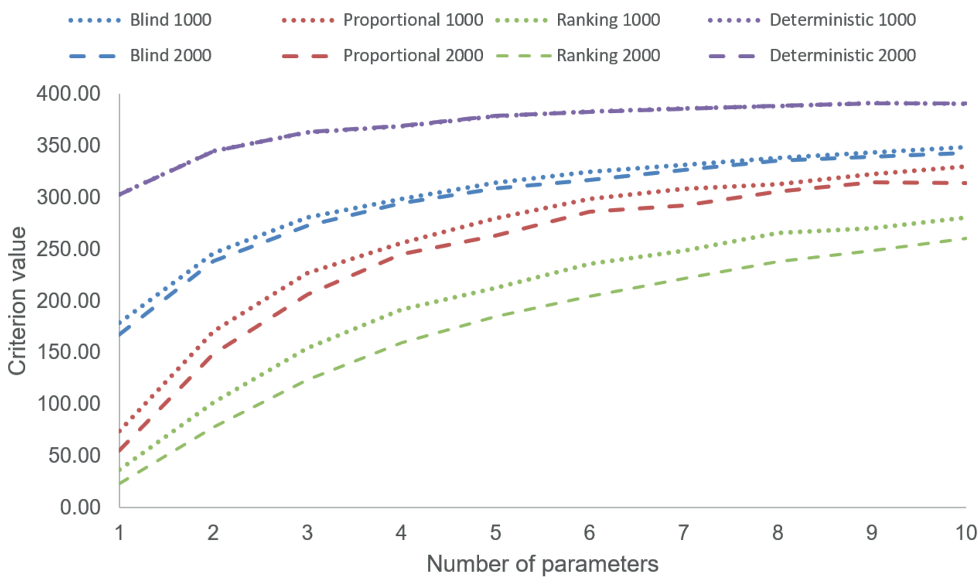

Fig. 5. Influence of the number of parameters (1–10) of the production task on the effectiveness of various types of mutation after 1000 and 2000 iterations

This time (see Fig. 6), apart from minor disturbances, we did not observe any influence of the tested factor. At the same time, the order of the efficiency of individual mutations has not changed. The new mutation operators made it possible to obtain the lowest cost of executing the task sequence.

In order to broaden the comparative analysis of different types of mutations, the third experiment was based on the study of the impact of a distance definition on the transition cost between tasks. The basing of the definition of cost on Minkowski distance allowed control of the b-exponent of this measure (6). This made it possible to include in the analysis the distance that favors changes in large parameters (the Euclidean distance, b equal to 2), the distance that proportionally treats the differences of individual parameters (the Manhattan distance, b equal to 1), and the distance that favors small parameter changes (with the b exponent 0.5). Of course, the analysis carried out is purely theoretical. However, it increases the representativeness of its results for various actual productions.

As shown in Figure 7, the order of effectiveness of the various mutation operators has not changed. In the investigated range of the exponent b we are dealing practically with a linear dependence of effectiveness on its value. This suggests a linear increase in the statistic distance between tasks as a function of the value of the coefficient b. The latter study confirms the superiority of adaptive mutation operators over operators previously described in the literature.

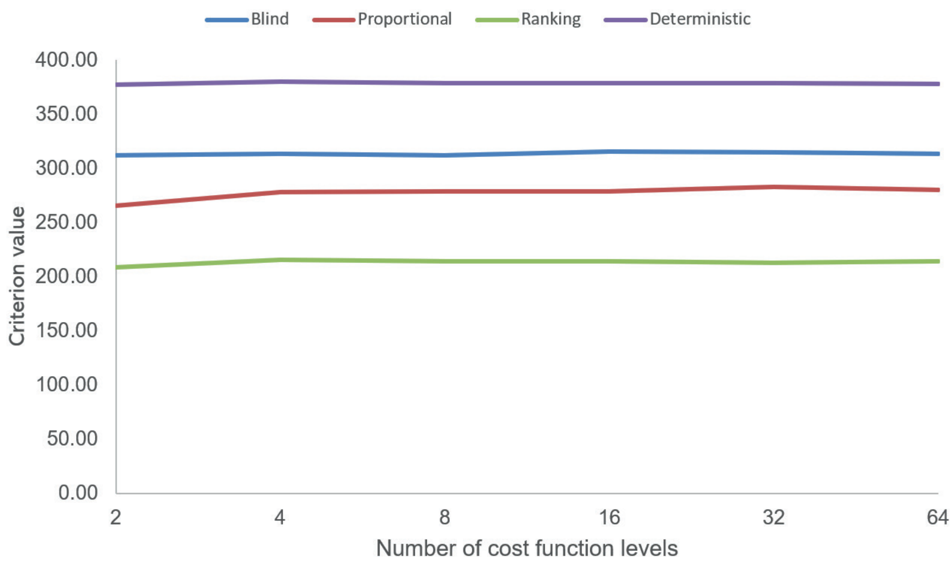

Fig. 6. Influence of the number of levels of cost functions (2–64) on the effectiveness of various types of mutations

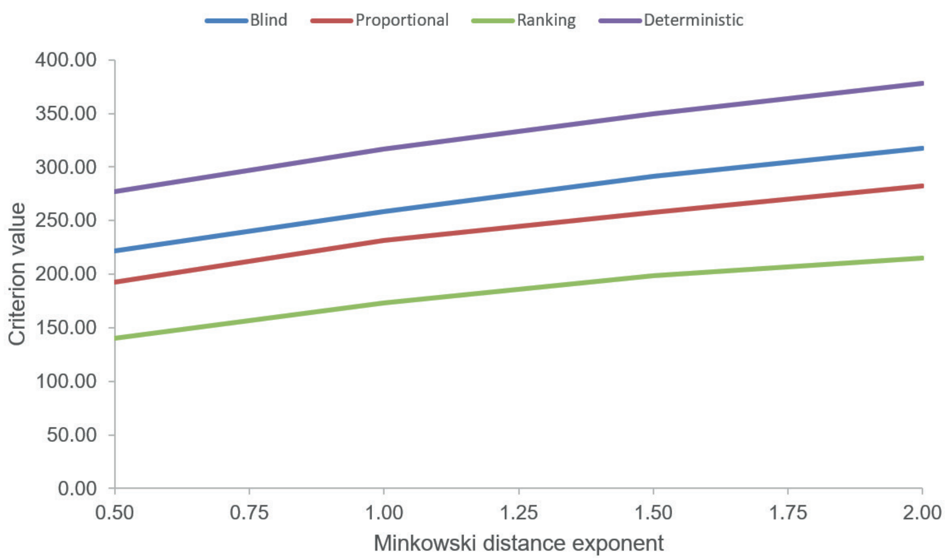

Fig. 7. Influence of the cost definition (the Minkowski distance exponent) on the effectiveness of various mutations

4. Conclusions

The article presents the concept of new mutation operators dedicated to evolutionary task scheduling, with a theoretical analysis of the impact of the nature of production on the efficiency of these operators. The aforementioned nature of the production was expressed here by the structure of parameters describing the production task and the definition of the transition cost function between tasks. The results of quite a detailed analysis confirmed unequivocally that the proposed mutation operators allow for obtaining a much lower cost of implementing a sequence of production tasks in comparison to classical random and deterministic methods and under various conditions.

The analysis carried out showed that, depending on the parameters of the tasks and the definition of the cost function:

– the new ranking mutation allowed for a reduction of the global cost in the range of 19% to 86% in relation to the most popular, blind mutation;
– the new proportional mutation allowed for a reduction of the global cost in the range of 5% to 67% in relation to the blind mutation;
– the ranking mutation allowed for a reduction of the global cost in the range of 28% to 92% in comparison to the fully deterministic mutation;
– the proportional mutation allowed for a reduction of the global cost in the range of 15% to 82% in comparison to the deterministic mutation.

The breadth of the analysis carried out allows us to assume that, in the case of real production tasks, these new mutation operators will also show their advantage.

The presented study deliberately abandoned the use of the crossover operator, focusing on the mutation operator. However, it seems advisable to also continue research in the direction of applying the idea of a directed but random selection of the point of intersection of the task sequence for the crossover operator.

References

Abdoun, O., Abouchabaka, J., & Tajani, C. (2012). Analyzing the performance of mutation operators to solve the travelling salesman problem. *International Journal of Emerging Sciences*, *2*(1), 61–77. https://doi.org/10.48550/arXiv.1203.3099.

Bang-Jensen, J., Gutin, G., & Yeo, A. (2004). When the greedy algorithm fails. *Discrete Optimization*, *1*(2), 121–127. https://doi.org/10.1016/j.disopt.2004.03.007.

Banzhaf, W. (1990). The "molecular" traveling salesman. *Biological Cybernetics*, *64*(1), 7–14. https://doi.org/10.1007/BF00203625.

Deep, K., & Mebrahtu, H. (2011). Combined mutation operators of genetic algorithm for the travelling salesman problem. *International Journal of Combinatorial Optimization Problems and Informatics*, *2*(3), 1–23. https://www.redalyc.org/pdf/2652/265219635002.pdf.

Englert, M., Röglin, H., & Vöcking, B. (2014). Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. *Algorithmica*, *68*(1), 1295–1304. https://doi.org/10.1007/s00453-013-9801-4.

Fogel, D.B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics*, *60*(2), 139–144. https://doi.org/10.1007/BF00202901.

Fogel, D.B. (1990). A parallel processing approach to a multiple travelling salesman problem using evolutionary programming. In L.H. Canter (Ed.), *Proceedings of the Fourth Annual Symposium on Parallel Processing* (pp. 318–326). IEEE Orange County Computer Society.

Gao, K., Huang, Y., Sadollah, A., & Wang, L. (2020). A review of energy-efficient scheduling in intelligent production systems. *Complex & Intelligent Systems*, *6*(2), 237–249. https://doi.org/10.1007/s40747-019-00122-6.

García-Martínez, C., Cordón, O., & Herrera, F. (2007). A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. *European Journal of Operational Research*, *180*(1), 116–148. https://doi.org/10.1016/j.ejor.2006.03.041.

Hassanat, A.B., Alkafaween, E.A., Al-Nawaiseh, N.A., Abbadi, M.A., Alkasassbeh, M., & Alhasanat, M.B. (2016). Enhancing genetic algorithms using multi mutations: Experimental results on the travelling salesman problem. *International Journal of Computer Science and Information Security*, *14*(7), 785–801. https://arxiv.org/ftp/arxiv/papers/1602/1602.08313.pdf.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology*. The MIT Press. https://doi.org/10.7551/mitpress/1090.001.0001.

Kentli, A., Dal, V., & Alkaya, A.F. (2013). Minimizing machine changeover time in product line in an apparel industry. *Textile and Apparel*, *23*(2), 159–167.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(3), 231–247. https://doi.org/10.1016/0377-2217(92)90192-C.

Levin, A., & Yovel, U. (2014). Local search algorithms for multiple-depot vehicle routing and for multiple traveling salesman problems with proved performance guarantees. *Journal of Combinatorial Optimization*, *28*(4), 726–747. https://doi.org/10.1007/s10878-012-9580-x.

Malik, A. (2019). A study of genetic algorithm and crossover techniques. *International Journal of Computer Science and Mobile Computing*, *8*(3), 335–344.

Michalewicz, Z. (2013). *Genetic algorithms + data structures = evolution programs*. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-662-03315-9.

Nirmala, G., & Ramprasad, D. (2012). Innovative PMX – CX- operators for GA TO TSP. *International Journal of Scientific and Research Publications*, *2*(10). https://www.ijsrp.org/research-paper-1012/ijsrp-p1094.pdf.

Ramstorfer, F., & Delane de Souza, M. (2022). Reduction of incompatible intermixing of different steel grades in continuous casting by optimizing the casting sequence. *International Journal of Cast Metals Research*, *35*(1–3), 43–50. https://doi.org/10.1080/13640461.2022.2078550.

Singh, A., Yadav, A., & Rana, A. (2013). K-means with three different distance metrics. *International Journal of Computer Applications*, *67*(10), 13–17. http://doi.org/10.5120/11430-6785.

Shi, Y., & Liu, S. (2021). Very large-scale neighborhood search for steel hot rolling scheduling problem with slab stack shuffling considerations. *IEEE Access*, *9*, 47856–47863. https://doi.org/10.1109/ACCESS.2021.3058328.

Syswerda, G. (1991). Scheduling optimization using genetic algorithms. In L. Davis (Ed.), *Handbook of Genetic Algorithms* (pp. 332–349). Van Nostrand Reinhold.

Vijayaram, T.R. (2014). Metallurgy of continuous casting technology. *International Journal of Manufacturing & Industrial Engineering*, *1*(1), 17–36.

Wan, Y., Zuo T.-J., Chen, L., Tang, W.Ch., & Chen, J. (2020). Efficiency-oriented production scheduling scheme: an ant colony system method. *IEEE Access*, *8*, 19286–19296. https://doi.org/10.1109/ACCESS.2020.2968378.

Zhang, H., & Xu, Y. (2018). Online covering salesman problem. *Journal of Combinatorial Optimization*, *35*(3), 941–954. https://doi.org/10.1007/s10878-017-0227-9.