

# SUITABLE CONFIGURATION OF EVOLUTIONARY ALGORITHM AS BASIS FOR EFFICIENT PROCESS PLANNING TOOL

JANKOWSKI, T.

**Abstract:** *The paper presents the application of an evolutionary algorithm to searching for the optimal sequence of the assembly of parts. Based on a proper representation of assembly sequences and a set of geometrical, topological and technological constraints, connected with the attributes of a product, it is possible to create an evolutionary model. Through proper control of the evolution process in a model, based on the appropriate selection of parameters, it is possible to achieve good results in a short period of time. Although the evolutionary algorithm does not guarantee the obtaining of optimal solutions, it has been proven, based on sample simulations, that such solutions are obtained in a repeated manner. The application of the presented evolutionary approach enables creating fast assembly sequence planning tools, indispensable in tactical planning and operational control of manufacturing processes.*

**Key words:** *evolutionary algorithms, process planning, optimization*



**Authors' data:** Ph.D. Eng. **Jankowski**, T[omasz], Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, tomasz.jankowski@pwr.edu.pl

**This Publication has to be referred as:** Jankowski, T[omasz] (2015). Suitable Configuration of Evolutionary Algorithm as Basis for Efficient Process Planning Tool, Chapter 12 in DAAAM International Scientific Book 2015, pp.135-142, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-05-1, ISSN 1726-9687, Vienna, Austria

DOI: 10.2507/daaam.scibook.2015.12

## **1. Introduction**

As flexible and computer-integrated manufacturing systems and manufacturing assisting computer systems are being intensively developed, effective algorithms are needed to solve many problems involved in the planning and scheduling of production tasks. One of the first things which should be done when designing an assembly process is to determine the order in which the particular parts enter the assembly process, i.e. to plan the assembly sequence. The planning consists in generating different assembly sequence variants and evaluating them in order to determine the optimal sequence. The generated potential solutions are evaluated on the basis of an analysis of the geometrical, topological, technological and physical data on the product to be assembled. As most combinatorial optimization problems, the assembly sequence planning problem belongs to a class of problems difficult to solve in a reasonable time, which means that their exact solution by means of algorithms with polynomial complexity is most probably impossible. The imprecise terms: “difficult to solve” and “reasonable time” are explained as follows. The number of potential assembly sequences depends exponentially on the number of parts in the assembled product and it is equal to the number of all possible permutations of the set of parts  $N$  ( $N \geq 2$ ), i.e. it amounts to (factorial)  $N!$ . The factorial function increases at a rate by orders of magnitude higher than  $N$ ,  $N \cdot \log N$ ,  $N^2$  and all the functions  $N^K$  for a certain specified criterion  $K$ . An algorithm for which the order of magnitude of the time needed to execute it is upper bounded by a polynomial function is called a polynomial time algorithm and is referred to as a reasonable algorithm. Similarly an algorithm which in the worst case requires a superpolynomial or exponential execution time is called unreasonable. Among algorithmic problems one can distinguish problems having reasonable solutions, i.e. obtained in polynomial time. Such problems are referred to as easily solvable. Problems which have only unreasonable solutions, i.e. obtained in exponential time, are referred to as difficult to solve. The existing exact algorithms of nonpolynomial complexity (as a rule based on the branch and bound method) solve only problems of relatively small size in a reasonable time, which is unacceptable in practical applications. Currently the research is concentrated on searching for effective approximate, probabilistic or heuristic algorithms. The measure of their effectiveness is runtime and accuracy defined by the distance of the generated solution from the optimal solution (Harel, 1992), (Nowicki, 1999).

Considering the increasingly wider use of computer assistance in decision problem solving, it is necessary to employ effective and fast algorithms in production planning and control. This especially applies to the levels of tactical planning connected with preparing the system for carrying out specific tasks in a short time and to operational control connected with the current functioning of the production system (Lebkowski, 2000), (Sawik, 1992), (Sawik, 1996).

## **2. Evolutionary models for assembly sequence planning problem**

In the author’s opinion, the conceptual evolutionary model presented in this section will make it possible to effectively solve the assembly sequence planning

problem. The flow chart presented in figure 1 shows the functional structure of the model, detailing two areas:

- the physical space (solution space), containing information about the structure of the product to be assembled, the way of modelling the assembly sequence, the adopted optimization criteria and other needed input data on the product or the assembly process;
- the evolutionary space (model space), containing procedures for creating an initial population, procedures for the iterative repetition of the evolutionary optimization process and definitions of termination conditions as well as definitions of the many versions of the crossover, mutation, reproduction and succession genetic operators.

This paper highlights the importance of the proper configuration of the evolutionary algorithm, using as an example the crossover operation and the different ways of carrying it out. The major components of the evolutionary algorithm are described in detail in the publications (Jankowski, 2008), (Jankowski & Jedrzejewski, 2010).

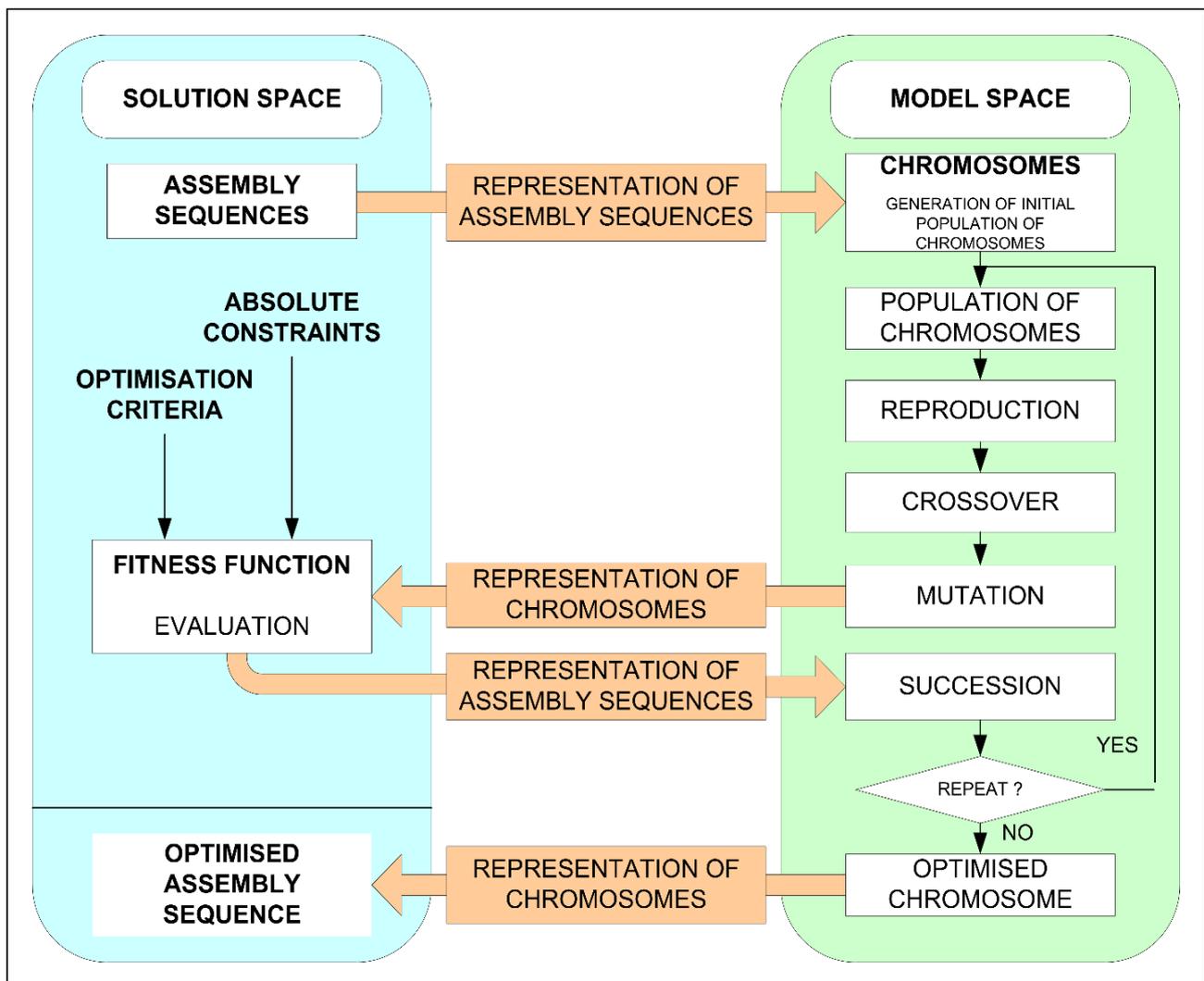


Fig. 1. Flow chart of evolutionary model for assembly sequence planning

### 3. Configuration of evolutionary algorithm

A critical stage in the preparation of the evolutionary algorithm for operation is its tuning, i.e. the optimal setting of such quantities as: the way of carrying out reproduction and succession, the kind of crossover, the kind of mutation, the population size, the crossover probability and the mutation probability, having a decisive influence on the operation of the algorithm and on the quality of the obtained results. Many time-consuming experiments need to be carried out in order to determine the optimal values of so many quantities having a bearing on the operation of the algorithm (Pawlak, 1996), (Pawlak, 1999). The quantities can be divided into:

- discrete quantities, called decision variables,
- continuous quantities, called parameters.

On the basis of an analysis of the literature on the subject the following group of variants of implementing decision variables was selected:

- three ways of carrying out the reproduction operation: roulette, tournament and rank;
- three ways of implementing the crossover operator: with ordering (OX), with partial mapping (PMX) and cyclic (CX);
- two ways of carrying out of the succession operation: trivial and elite.

Eighteen algorithm configuration variants for simulation studies of the effectiveness of the particular configurations were determined for the selected group of decision variables.

The independent development of evolutionary algorithms in the different research centres has contributed to the development of different schemes of implementing the crossover operator being the subject of the analyses presented below. One of the sources of the differences were: the number of parent individuals, being the source of data for the crossover operation, and the number of descendant individuals produced as a result of crossing-over. In this research the most often used scheme, according to which two parent individuals take part in the crossover and as a result two descendant individuals are obtained, was adopted.

- Partial mapping crossover (PMX)

The PMX crossover operator, designed by Goldberg and Lingle (Goldberg & Lingle, 1985), creates a descendant by selecting a subsequence from one of the parents, leaving unchanged the order and positions of as many elements of the other parental solution as possible.

- Ordering crossover (OX)

The ordering crossover operator, proposed by Davis (Davis, 1985), produces a descendant solution by selecting a subsequence from one of the parents, leaving unchanged the mutual order of the elements taken from the other parent.

- Cyclic crossover (CX)

The CX crossover operator, proposed by Oliver (Oliver et al., 1987), creates descendant solutions in such a way that each element of the descendant and its position come from one of the parents.

The travelling salesman problem (very popular in testing combinatorial optimization algorithms) was used as the test problem. The problem complexity was

determined for 36 points (cities) where the size of the space of solutions amounts to  $36! \approx 3,7 \cdot 10^{41}$ . The shortest route (equal to 360), being the optimal solution, was determined for the generated plan of the cities and subsequently used to assess whether the algorithm could find the optimal solution. The test problem optimal value was also used as an additional condition for stopping the algorithm once the optimum was reached.

#### 4. Analysis of obtained results

Figures 2-7 show a comparison of the results obtained for the analyzed ways of implementing the crossover operator (OX, PMX, CX) for different configurations of the reproduction and succession operators.

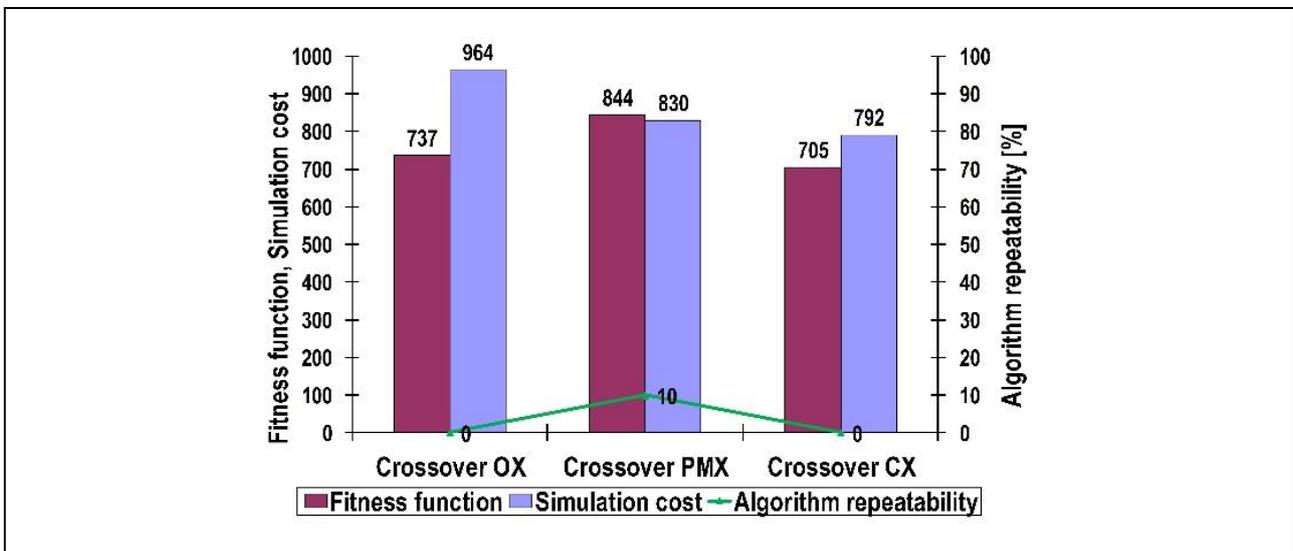


Fig. 2. Comparison of crossover operators in configuration with roulette reproduction and trivial succession

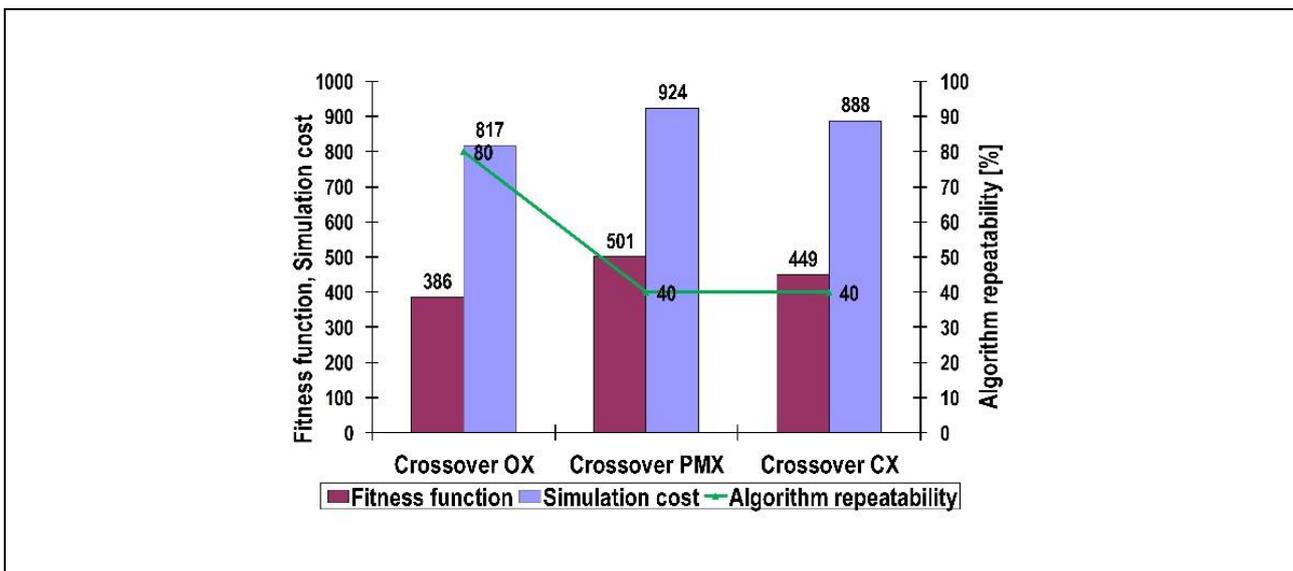


Fig. 3. Comparison of crossover operators in configuration with roulette reproduction and elite succession

It is hard to draw conclusions directly from the graphs shown in figs 2 and 3, where the algorithm would not reach, or would reach with a relatively low repeatability, optimal solutions. Some ambiguity is visible in fig. 2. The configuration with PMX, the only one from the compared configurations which reached the optimal value once, is characterized by the worst average of the solution quality results. For the further interpretation of the results the measures of their dispersion were used. For this purpose standard deviations for the solution quality value and the simulation cost, being the mean values for a sample of ten experiments, were calculated in order to determine the random uncertainty of the results. The PMX crossover results show an extent of uncertainty several times larger than for the other crossover operators (OX and CX). This indicates that despite the high solution quality average (844,1±228,3) and the large dispersion of the results, the algorithm managed to reach the optimal value in one of the measurements. As regards the simulation cost, the mean value (which, if no optimal values are reached, shows the number of the algorithm iteration in which the last improvement in the evaluative function occurred) was the highest for the OX crossover operator. As regards uncertainty, the lowest value (by nearly an order of magnitude lower than for the other operators) was obtained for the OX crossover, which means a much better kurtosis than in the case of the other crossover operators. As regards the configuration with roulette reproduction and elite succession (fig. 3), the results are better for the previous configuration, but no one hundred per cent repeatability in reaching the optimum was achieved. As opposed to the previous graph, the graph in fig. 3 shows much better results in optimum reaching repeatability. Thanks to the use of the OX operator a repeatability of 80% was achieved, which is a twice better result than the one obtained using the PMX/CX operator. This is also reflected in the mean value of the solution quality function and in its uncertainty.

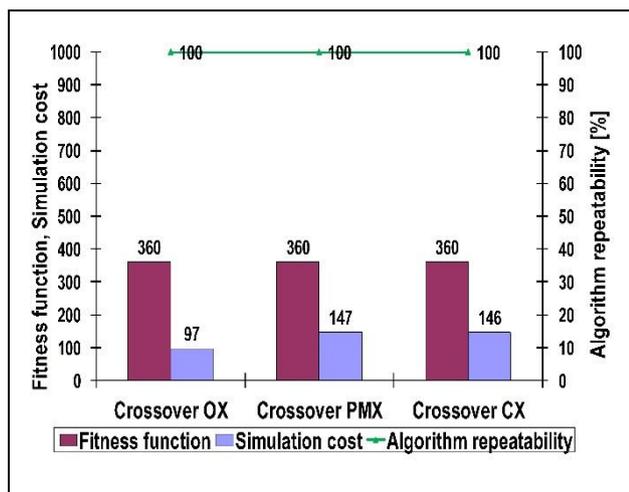


Fig. 4. Comparison of crossover operators in configuration with tournament reproduction and elite succession

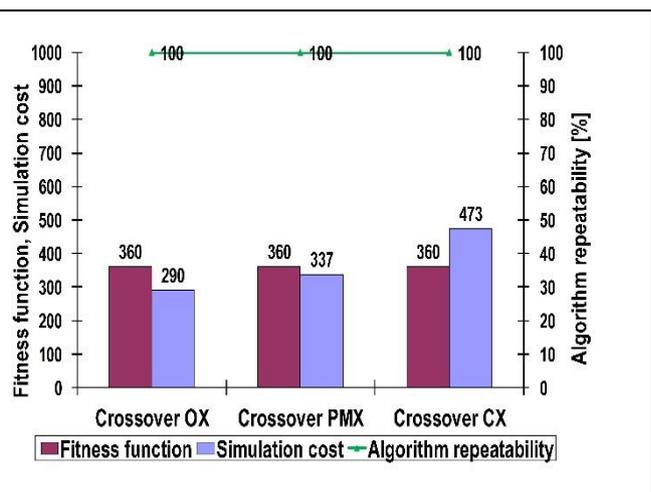


Fig. 5. Comparison of crossover operators in configuration with rank reproduction and trivial succession

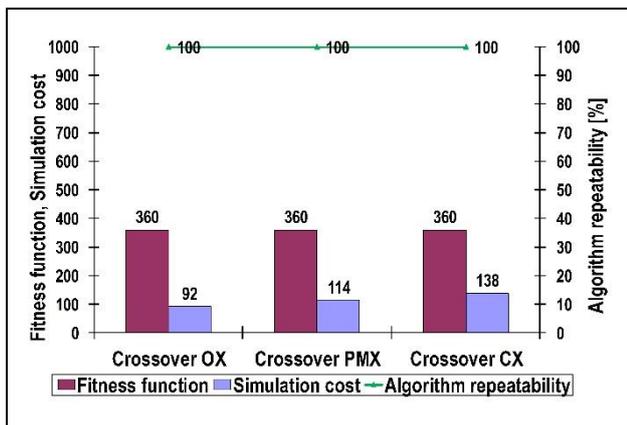


Fig. 6. Comparison of crossover algorithms in configuration with tournament reproduction and elite succession

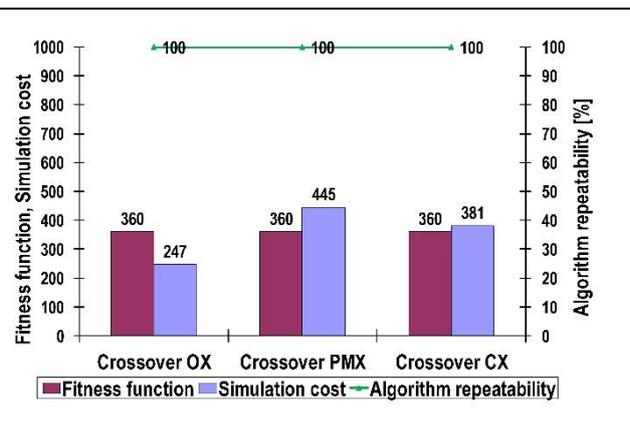


Fig. 7. Comparison of crossover operators in configuration with rank reproduction and elite succession

The results presented in figures 4-7 are much easier to interpret. All the four configurations would score the optimal result with one hundred per cent repeatability, regardless of the crossover used, whereby evaluation in this case practically comes down to the assessment and comparison of the simulation cost needed to obtain the results. In each of the configurations the OX crossover would entail the lowest simulation cost. The cost of simulations run using the PMX operator or the CX operator was by 24-80% higher depending on the configuration of the reproduction and succession operators. It would take the longest to reach the optimum when the CX operator was used, except for the single case of the configuration with rank reproduction and elite succession (fig. 7) when the simulation cost for this operator was lower than the one for the PMX operator.

## 5. Conclusion

The aim of the analysis was to determine the optimal configuration (different variants of crossover operator) of an evolutionary algorithm to solve combinatorial optimization problems. The tests were performed on the example of the traveling salesman problem, for which it was known the optimum solution. The same group of optimization problems may also include assembly sequence planning problem, although it has much more restrictions.

Numerical studies of the effect of the selected ways of implementing the crossover operator, generally considered to be the best for the permutational encoding of combinatorial optimization problems, to which the assembly sequence planning task and the travelling salesman route planning test task belong, have been carried out.

In all the configuration variants (except for the ones using the roulette reproduction) for all the three ways of carrying out the crossover operation, i.e. with ordering (OX), with partial mapping (PMX) and cyclic (CX), optimal solutions were obtained with one hundred per cent repeatability of algorithm operation, with some differences in the simulation cost. The OX crossover comes out best, achieving the optimal solution at the lowest simulation cost in each of the variants. When the optimal

solution was obtained using the PMX crossover or the CX crossover, the simulation costs was by a few tens of per cent higher.

Thanks to the numerical simulations the most effective configuration of the OX crossover operator and the values of the crossover and mutation probability parameters for which the algorithm generates optimal solutions with one hundred per cent repeatability at the lowest simulation cost have been determined.

In the next stage of analysis, it is planned to use an algorithm with the optimal configuration operators. Function to evaluate solutions to the traveling salesman problem will be replaced with a function to evaluate solutions for assembly planning problem. The most important thing will be to construct a function taking into account all the constraints assemble the parts into a finished product.

## 6. References

- Goldberg D. E., Lingle R. (1985). Alleles, Loci, and the TSP, *Proceedings of the First International Conference on Genetic Algorithms*, J. J. Grefenstette, Ed., pp. 154-159, Lawrence Erlbaum Associates
- Harel D. (1992). *The thing about the essence of computer science: algorithmics*, Wydawnictwa Naukowo-Techniczne, Warszawa (in Polish)
- Davis L. (1985). Applying Adaptive Algorithms to Epistatic Domains, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 162-164
- Jankowski T. (2008). Memetic algorithm for assembly sequence planning, *Journal of Machine Engineering*, vol. 8, no. 3, pp. 77-90
- Jankowski T., Jedrzejewski J. (2010). Hybrid evolutionary optimization in efficient task planning", *Sustainable production and logistics in global networks: 43rd CIRP International Conference on Manufacturing Systems*, W. Sihn and P. Kurlang, Ed., pp. 884-891
- Lebkowski P. (2000). *Methods of computer-aided mechanical assembly in flexible manufacturing systems*, Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Krakow (in Polish)
- Nowicki E. (1999). *Tabu search method in production scheduling problems*, Prace Naukowe Instytutu Cybernetyki Technicznej Politechniki Wrocławskiej. Monografie nr 27, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (in Polish)
- Oliver, I. M., Smith, D. J., Holland, J. R. C. (1987). A Study of Permutation Crossover Operators on the Travelling Salesman Problem, *Proceedings of the Second International Conference on Genetic Algorithms*, J. J. Grefenstette, Ed., pp. 224-230, Lawrence Erlbaum Associates
- Pawlak M. (1996). The search for the best decision variables values of the evolution program, *Materialy I Krajowej Konferencji pt.: Algorytmy Ewolucyjne*, pp. 94-105, Murzasichle
- Pawlak M. (1999). *Evolutionary algorithms as a tool for production scheduling*, Wydawnictwo Naukowe PWN, Warszawa (in Polish)
- Sawik T. (1992). *Discrete optimization in flexible manufacturing systems*, Wydawnictwa Naukowo-Techniczne, Warszawa (in Polish)
- Sawik T. (1996). *Production planning and control in flexible assembly systems*, Wydawnictwa Naukowo-Techniczne, Warszawa (in Polish)