Journal of Machine Engineering, Vol. 8, No. 3, 2008

evolutionary computing, assembly,
optimisation, process planning

Tomasz JANKOWSKI[1]

MEMETIC ALGORITHM FOR ASSEMBLY SEQUENCE PLANNING

The paper presents the application of a memetic algorithm to searching for the optimal sequence of the assembly of parts. Such approach is based on the use of an algorithm connecting two methods of global and local search in order to increase the effectiveness of the conducted optimisation process. 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.

1. INTRODUCTION

The competitiveness of the undertakings in a manufacturing process is more and more often dependent on its elasticity and the ability to fulfil the needs of a client. The ability of quick reactions to the market needs, indispensable in such conditions, forces the use of computer-aided methods of designing of such manufacturing processes, which bring special organisational and economical benefits.

It will be dependent on the proper elaboration of the assembly's technological process and its adequate realisation, whether the product (machine, mechanism) will fulfil the tasks set by its designer and what will long the length of its exploitation period will be. Therefore the assembly, so important and laborious in the construction process of machines and devices, should be properly prepared and realised.

One of the main tasks of designing the technological process of assembly is to determine the most suitable order of completing parts (assemblies). The goal of assembly sequence planning is the determination of different variants of the order of assembly operation realisations and their assessment, allowing for the determination of the optimal sequence. The assessment criterion of created assembly plans is, eventually, total cost of the

[1] Wroclaw University of Technology, Institute of Production Engineering and Automation, ul. Łukasiewicza 5, 50-371 Wrocław, Poland, e-mail: tomasz.jankowski@pwr.wroc.pl

assembly process. Considering geometrical, topological and technological restraints, the achievement of the optimal assembly plan is very difficult from the combinatorial point of view. The amount of possible assembly plans depends exponentially from the amount of parts in a product. Proper selection of the assembly plan considerably limits the number of reorientations of a base part, the amount of tool changes, grouping of tasks in multi-task operations and the simultaneous attachment of many parts, eliminating some operations, because of which the amount of needed assembly devices is limited, and eventually also production costs [7, 10, 15].

The shortening of designing of such a complex process as assembly and the assurance of obtaining effective solutions during particular design stages requires the use of computer systems as design aids.

Conducted research shows the percentage shares of different tasks realised by a technologist in the entire designing time, according to the traditional (manual) method of designing the technological process [16]:

- making technological decisions 15%
- data analysis and calculations 40%
- preparation of documentation 45%

According to the presented research, as much as 85% of the technologist's work time is spent on the completion of tasks which do not require making decisions, but very laborious at the same time. The application of Computer Aided Process Planning - CAPP makes it possible to automate the process of carrying out task which do not require making decisions, which considerably increases efficiency, as well as shortens preparation and commissioning times.

2. EVOLUTIONARY OPTIMISATION

2.1. EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EA) are the approach to using evolution mechanisms, known from nature, in order to solve optimisation problems (Fig. 1) First clear analogy of EA to the natural evolution is the existence of population – in nature understood as a group of coexisting individuals of different species, while in the EA as a group of different possible solutions of a problem. Evolution is nothing else than the formation of species better and better fitted to the environment and the disappearance of the least fitted. The analogy of EA to nature mechanisms is simple: assuming that different solutions of a given problem apply to the individuals, and establishing a certain measure of their fitting (so called assessment function), then, with the use of adequate methods of generating new solutions, it is possible to evolve a population consisting of "good" solutions, that is preferred by the assessment function. Which generation methods? Here EA follow the example of nature. Such solutions are treated as genotypes of organisms, and these, as it is well known, in connection with the operations of crossovers and mutations, constitute the essence of how evolution works. As a remainder: crossovers consist in the transfer of genotype fragments to progeny, coming from both parents, while mutation consists in random changes of a new genotype. From the crossovers it is expected to cause the creation of progeny possessing good characteristics of both parents, while from mutations to enable discovering good, formerly unknown characteristics. Naturally the operation of EA does not end on the one-time generation of progeny [1,5,11].

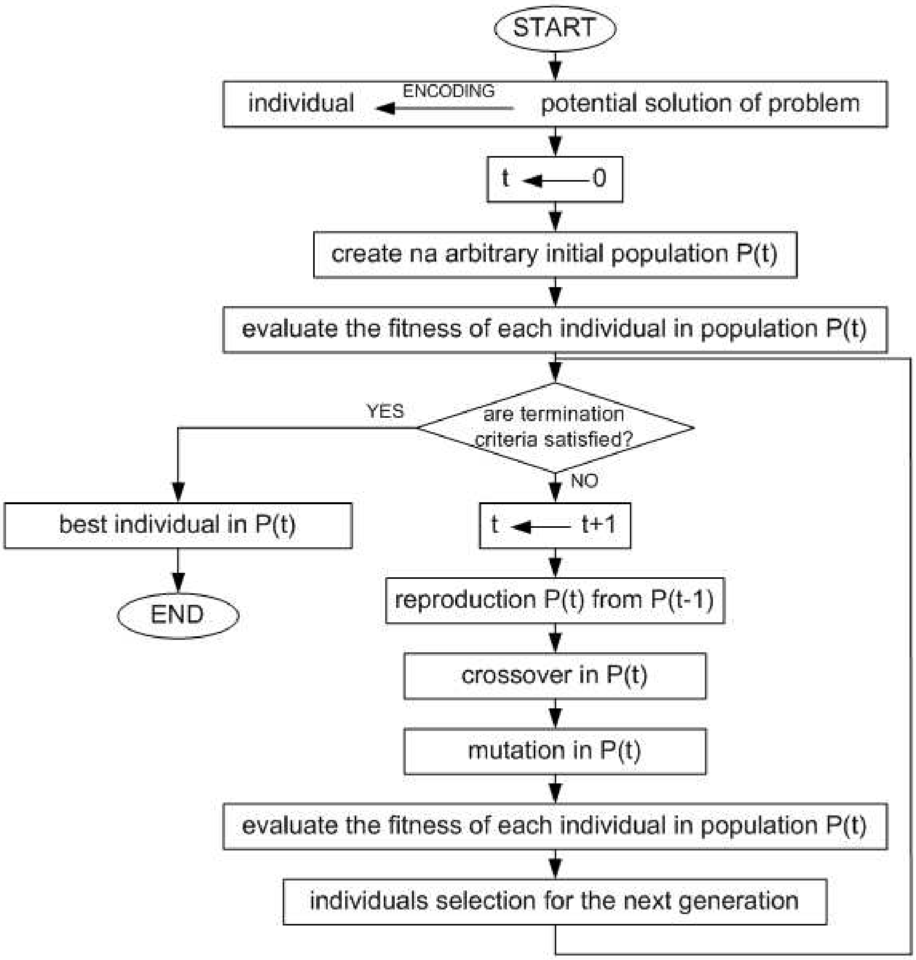

Fig. 1. The flowchart of the evolutionary algorithm

2.2. MEMETIC ALGORITHMS

Memetic algorithms constitute a group of stochastic heuristics for the optimisation with a global character, which connect the global character of an evolutionary algorithm with the precision of a method of local search, in order to improve the quality of obtained solutions (Fig. 2).

Optimisation methods connecting the evolutionary algorithm with a precise local search are differently named in the available literature, such as: a hybrid evolutionary algorithm, the Baldwin evolutionary algorithm, the Lamarck evolutionary algorithm,

algorithm of evolutionary – local search, etc. In 1989 Pablo Moscato [12,13] in his works for the first time used the term of *memetic algorithm* to describe the whole group of algorithms, which to the operation of a simple algorithm introduce one or more operations of improving the solutions with the use of a precise local search.

Based on the analyses of the operation of EAs and the achieved results of solving different optimisation problems it is possible to name few main reasons of hybridisation with precise methods of local optimisation [6]:

- Solving complex optimisation problems can be divided to smaller tasks, realised in successive stages by precise methods or specialised heuristics. Such method may sometimes require additional operations transforming data from EA to a suitable form, which allows using the knowledge of the problem by other methods in successive stages and vice versa. The application of a hybrid of the most effective methods, in order to improve the quality of achieved solutions by the introduction of additional operations realising precise local search inside the EA, eliminates superfluous operations.
- There is no universal method for solving all optimisation problems, which is confirmed by the results of research and theoretical analyses. EA are also not the ideal method for the use in solving all optimisation tasks. Most often EA are used for problems with a high computational complexity with itemised restraints. In such case the use of specialist operators, based on, for example, method of local search, can increase the effectiveness of the optimisation process. As it is shown by conducted experiments, the use of hybrid EAs yields better results than any of the methods separately.
- EAs are characterised by very good results in fast search of proper areas of optimum search, i.e. so-called exploration. It is much worse with the precise localisation of optimum solutions, i.e. so-called exploitation. The application of hybridisation of AE with the method of precise local search may enable increasing the effectiveness of the optimisation process by obtaining better solutions and decreasing operation time.
- In many practical problems it is required to fulfil a certain amount of limitations, resulting from the definition of a problem. In such case it is possible to take advantage of the local search as a mechanism for fixing solutions, which do not meet imposed limitations.
- One of hybridisation motives is also the concept of Dawkins on the subject of the idea of memes in the cultural evolution, presented for the first time in a book called “The Selfish Gene” [2]. According to Dawkins’ opinion, a meme is a basic unit of the cultural transmission, which is imitation. Memes are subject to the same rules, which apply to the replication of genes, however the process of meme replication has a distinct character. First of all, unlike the biological evolution, the cultural evolution progresses much faster. Second of all, genes compete for the resources of natural environment, while memes compete for the resources of the human brain’s memory. Memes are for example melodies, ideas, circulating expressions, fashion of clothes, rules of good manners.

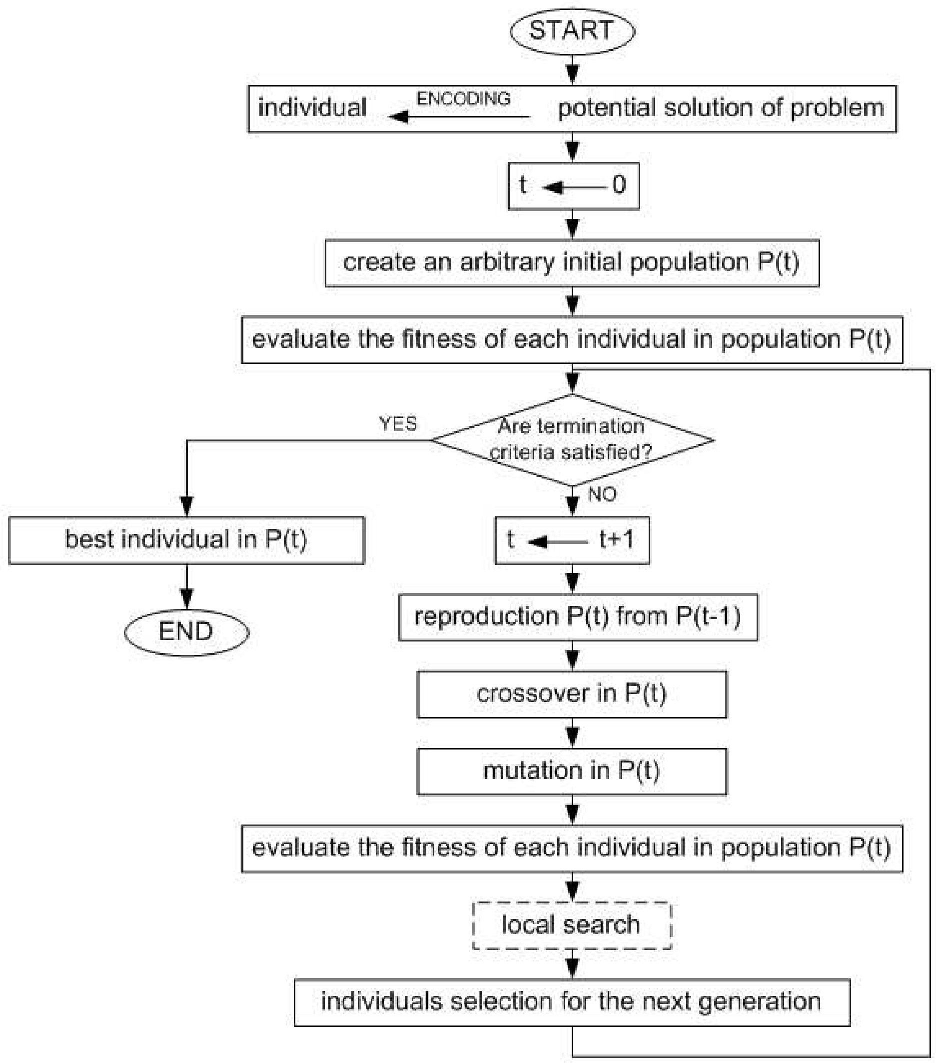

Fig. 2. The flowchart of the memetic algorithm

3. EVOLUTIONARY APPROACH TO SOLVING ASSEMBLY SEQUENCE PLANNING PROBLEMS

The application of evolutionary algorithms in the process of assembly sequence planning was realised according to the evolutionary approach shown on a diagram (Fig. 3). Detailed descriptions of particular problems will be found in the successive sections [8,9].

3.1. THE REPRESENTATION OF A CHROMOSOME

One of the most important decisions concerning the use of evolutionary algorithms is the definition of the method for representing the real problem in the algorithm (coding). Such task is completed by defining the mapping between a point in the space of solutions (assembly sequence) and a point in the space of representations (chromosome). Among methods from the class of evolutionary algorithms there is no unequivocal solution of this problem. In genetic algorithms, the basic form is the binary representation; evolutionary strategies are based on the representation of real numbers, while genetic programming mainly uses trees describing the structure of programs. In the presented work, the integral presentation has been used. A chromosome consists of a sequence of natural numbers representing particular parts of the assembled unit or finished product.

In order to ensure proper operation of the algorithm, making it possible to assess the obtained results, a suitable form of data is required, describing geometrical, topological and technological characteristics of the analysed product or assembly. For this purpose a graph is created, representing the characteristics of a product (assembly) and based on it, matrices are created describing: existing connections between parts, geometrical restraints of the assembly order, possible directions of the assembly and available end-effectors for particular parts in an assembly.

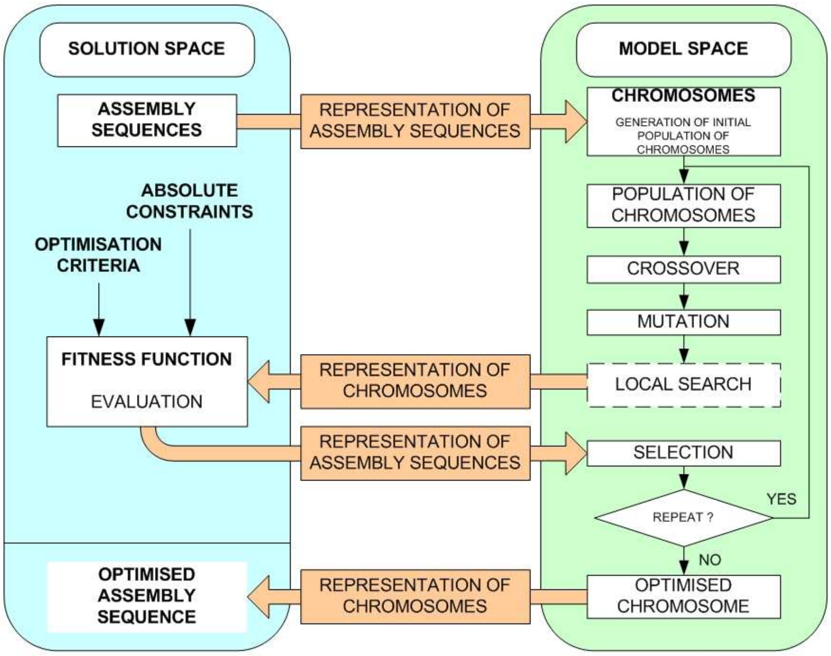

Fig. 3. The evolutionary approach for assembly sequence planning problem

3.2. THE INITIATION OF THE INITIAL POPULATION

Initiating of the population is used to generate the populations of a given size, from which the algorithm can start the evolution process. Two methods are used for initiating the preliminary population: in a random manner or by creating the chromosomes initially optimised by means of other methods. The effectiveness of the evolutionary process can be significantly improved by a high quality initial population and its proper size. It must be however remembered that in case of creating well adjusted chromosomes by means of other methods, there is a risk that the evolution process will be directed to a local optimum. In the presented solution the initial population is created in random manner. The size of this population is one of the algorithm's parameters.

3.3. CROSSOVERS

The task of a crossover parameter is to recombine fragments of chromosomes (parents) and in this way to create new chromosomes (progeny). There are at least few methods of realising the crossovers for the assumed integer representation. In the first step, the realisation of such operator consists in the random selection of a part from the parent's chromosome and the duplication of it to the chromosome of progeny, retaining its position. In the second step, the rest of chromosome elements of the first parent are copied to the progeny chromosome, but with retaining their order of appearance in the chromosome of the second parent. The insertion of elements, copied in the second step to the chromosome of progeny, begins from the first position after the fragment copied in the first step. In the analogous manner, the creation of the second descendant is carried out, based on the same pair of chromosomes (parents).

3.4. MUTATION

The mutation operator is responsible for random introduction of changes in a chromosome. In this example, a variation of mutation based on the change of the position of genes (swap mutation) has been used. It consists in the random selection of two elements in a chromosome and swapping their position.

3.5. SELECTION

The selection (carried out in reproduction and succession phases) results in channelling the algorithm in the direction of results improving the former. The task of reproduction is the selection of parents for genetic operations, the result of which is the creation of a temporary population, on which the operations of crossovers and mutations are carried out, leading to the creation of progeny population. The succession consists in creating new base population to the successive generation, based on individuals found in the progeny population and the old base population.

3.6. FITNESS FUNCTION

The fitness function allows assessing the degree of how the chromosomes are fit to the environment, that is the meeting of restraints resulting from the geometrical, topological and technological description of the assembled unit and the technological process of assembly. First assessment criterion is the verification of meeting the geometrical restraints by the chromosome, necessary to create the realisable sequence. In the next step, the assessment for the realisable sequences takes place, based on the optimisation criteria, which apply to the minimisation of reorienting the base unit and the minimisation of end-effector changes in the assembly robot. Such sequence optimisation enables minimising the summary time needed for the assembly of the entire unit or product, which directly influences the cost of the entire assembly.

3.7. LOCAL OPTIMISATION – TABU SEARCH METHOD

Local optimisation algorithms (LOA) (improvement algorithms) have been known in the literature for a long time in combinatorial optimisation tasks. General principle of operation of the LOA comes down to the iterative improvement of the goal function value for a certain solution, assumed as the starting solution. Starting solution can be generated by means of a different algorithm or chosen randomly. In the presented paper, starting solution for the local optimisation algorithm is generated by the evolutionary algorithm. A set of potential modification of a solution is connected to the concept of contiguity, i.e. a set of solutions, "close" to the base solution in a given iteration. In every iteration, a contiguity set is generated, from which the solution with the highest value of a goal function is chosen (for maximisation problems). The algorithm finishes its operation when it does not find the solution with a higher goal function value than the base solution. Presently, research conducted on the LOA introduces many new methods and techniques modifying, mentioned earlier, the classic operation scheme of LOA, making the methods described on such base very effective. One of such methods is a search method with forbiddances, called the Tabu method [3,4,14].

4. INVESTIGATION OF THE INFLUENCE OF THE GENETIC OPERATION CONFIGURATION ON THE OPERATION EFFECTIVENESS AND OPTIMISATION POSSIBILITIES

A very important stage of preparing the algorithm is its tuning, that is the definition of optimal values for characteristic quantities, such as: methods of realisation of reproduction, crossovers and mutations, intensiveness of succession, probability of crossovers and mutation, size of population, and others. Mentioned quantities, because of their character, can be divided to:

- continuous, which can be named as parameters;
- discontinuous; which can be named as decisive variables.

Based on the selected and presented in Chapter 5 ways of realisation of genetic operators, for the assumed path coding of combinatorial optimisation problems, 18 variants of the evolutionary algorithm configurations were selected (Table 1). For each of the selected variants, simulation investigations were carried out in order to define the characteristics of the algorithm behaviour for different values of crossing probability and mutation probability parameters.

Based on preliminary investigations, values of parameters for carrying out these simulations were set:

- size of population: 200;
- maximum amount of iterations: 1000;
- probability of crossovers: 0; 0,05; 0,1; 0,15; 0,2; 0,25; 0,3; 0,35; 0,4; 0,45; 0,5; 0,55; 0,6; 0,65; 0,7; 0,75; 0,8; 0,85; 0,9; 0,95; 1;
- probability of mutations; 0; 0,02; 0,04; 0,06; 0,08; 0,1; 0,12; 0,15; 0,2; 0,25; 0,3; 0,35; 0,4; 0,45; 0,5; 0,55; 0,65; 0,75; 0,85; 0,95.

Table 1. Variants of investigated algorithm configurations

Variant no.	Reproduction	Crossovers	Succession
1	Roulette	OX	Trivial
2	Roulette	OX	Elite
3	Roulette	PMX	Trivial
4	Roulette	PMX	Elite
5	Roulette	CX	Trivial
6	Roulette	CX	Elite
7	Tournament	OX	Trivial
8	Tournament	OX	Elite
9	Tournament	PMX	Trivial
10	Tournament	PMX	Elite
11	Tournament	CX	Trivial
12	Tournament	CX	Elite
13	Rank	OX	Trivial
14	Rank	OX	Elite
15	Rank	PMX	Trivial
16	Rank	PMX	Elite
17	Rank	CX	Trivial
18	Rank	CX	Elite

In the tuning process, a travelling salesman problem has been used, which is very popular in investigating the combinatorial optimisation algorithms. The complexity of a problem was defined for 36 cities, where the size of a searched solution space is $36! \approx 3{,}7*10^{41}$. For the generated plans of cities the shortest path was determined by means of analytical methods, which is the optimal solution, and then used for assessing if the algorithm is capable of finding the optimal solution. Additionally, the knowledge of the optimal value for the investigated test task was used as an additional condition for stopping the operation of the algorithm in the moment of reaching the optimum.

For each set of parameters, 10 independent simulations were carried out and then the average value of results was determined. Based on the obtained results for each of the configurations, a set of three characteristics in function of two variables – probabilities of crossovers and mutations – was determined:

- quality of obtained result: value of fitness function, that is length of the shortest found travelling salesman route;
- simulation cost; number of algorithm iterations needed to find the best reached solution;
- repeatability of the algorithm operation; amount of obtained optimal results in 10 carried out simulations.

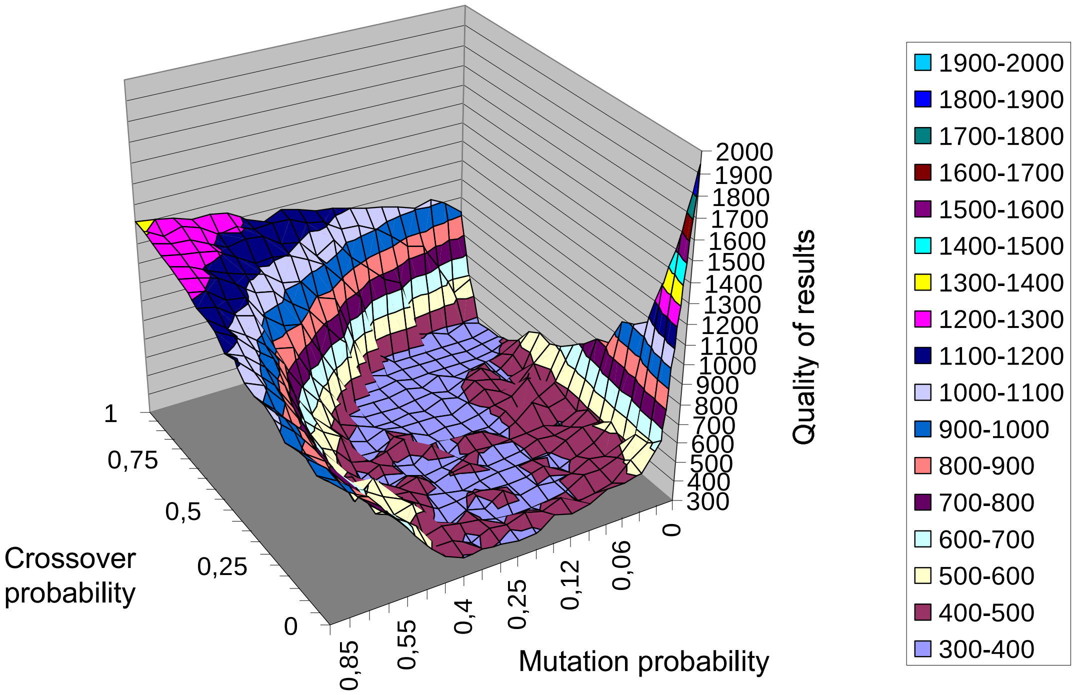

Fig. 4. Characteristic of the quality of obtained results

Figure below shows the example set of operation characteristics of the algorithm configured based on the range reproduction, OX crossovers and trivial succession (variant no. 13). Figure 5 shows the quality characteristic of obtained results. Such graph allows verifying in which spots the algorithm reaches best results and whether these solutions are optimal. If the algorithm does not reach solutions equal to the optimal solution, then how large is the difference between the obtained and optimal solution. Fig. 6 shows the simulation cost characteristic, that is the amount of algorithm iterations, by which the last quality improvement of the obtained result occurred. If, during simulations, the optimal solution was obtained, the simulation cost is defined by the amount of iterations, after which such solution was obtained. In case in simulations the optimal solution was not achieved, the iteration number, by which the last solution quality improvement was achieved allows speculating about the possibilities and the pace of algorithm convergence. Fig. 7 shows the characteristic applying to the repeatability of the algorithm operation. Such graph defines how many times the algorithm found the optimal solution in 10 independent simulations. This allows assessing the repeatability, with which the algorithm can achieve optimum with defined parameter values. Presented set of characteristics makes it possible to assess the particular configurations of the algorithm and to analyse the influence of such configurations on the algorithm properties.

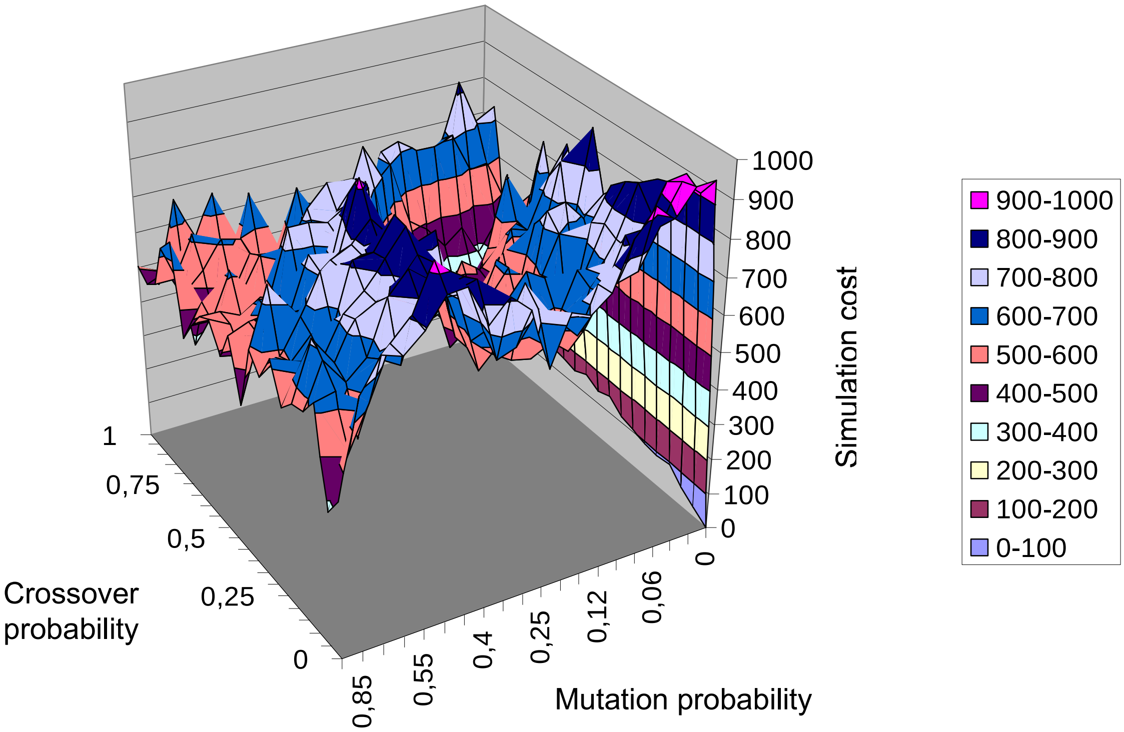

Fig. 5. Simulation cost characteristic

5. EXAMPLE OF THE USE OF THE EVOLUTIONARY APPROACH

For the verification of the formerly presented evolutionary approach, an example of the assembly sequence planning for a gear pump PZ-10P is presented (Fig. 8). This pump consists of 34 components. After designing and implementing particular components of the earlier mentioned evolutionary approach, a series of simulations was conducted. In the course of consecutive simulations, the parameterisation of the algorithm was achieved, which aimed for such selection of parameters (probabilities) controlling the evolution process, as to achieve the highest possible quality of obtained results in the possible shortest time. As a result of parameterisation, the following values of parameters were determined: population size – 100, crossover probability – 0.7, mutation probability – 0.33. Then independent simulations were conducted for various sizes of the initial population using the algorithm with the defined parameters, which as a result enabled obtaining the solution with the highest assessment function values in form:
7-6-13-12-10-11-14-15-17-16-19-18-20-21-23-22-24-32-31-34-33-28-30-25-27-29-26-8-9-5-4-3-1-2.

The course of assessment function value changes for the best result and the mean value for all chromosomes in the population is shown on a graph (Fig. 9).

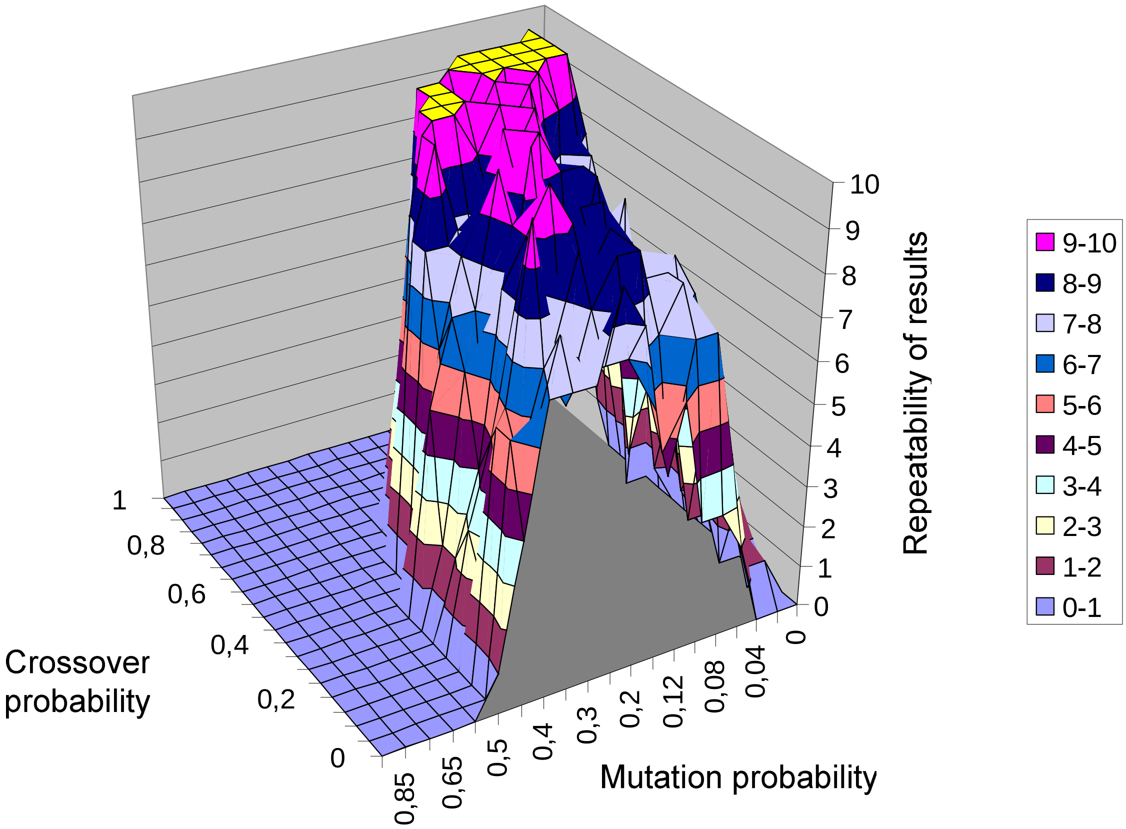

Fig. 6. Algorithm operation repeatability characteristic

6. CONCLUSIONS

Intensive development of elastic and computer-integrated manufacturing systems and computer aided manufacturing systems requires effective algorithms for determining the assembly sequence. Definite majority of these problems and generally combinatorial optimisation problems belongs however to the class of NP-hard. Therefore it will be impossible to solve them precisely with the use of algorithms of the polynomial complexity. That is why at the present moment the main direction of research is searching for effective approximate algorithms. The measure of their effectiveness is their operation time and precision, defined as the distance of the generated solution to the optimal solution based on the assumed quality criterion. Based on research, which results were partially presented in hereby article, it can be stated that the memetic algorithm is a “strong” tool for applying to the assembly sequence planning. Although they do not guarantee obtaining the optimal solution, conducted simulation research shows that with the use of proper working parameters the algorithm reaches the optimal solution in a repeatable manner.

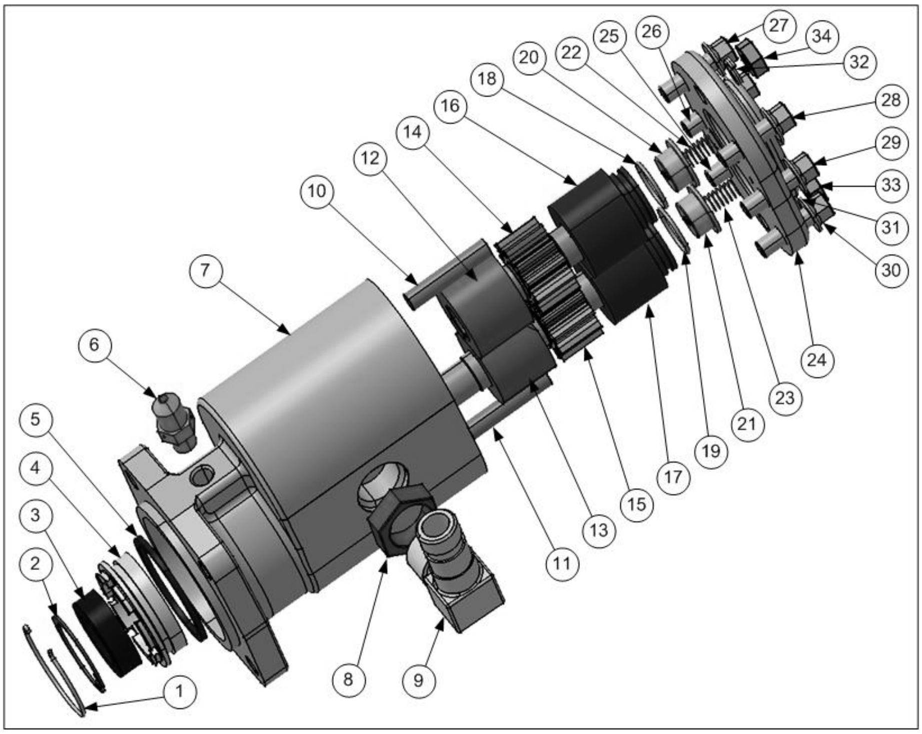

Fig. 7. Gear pump PZ-10P

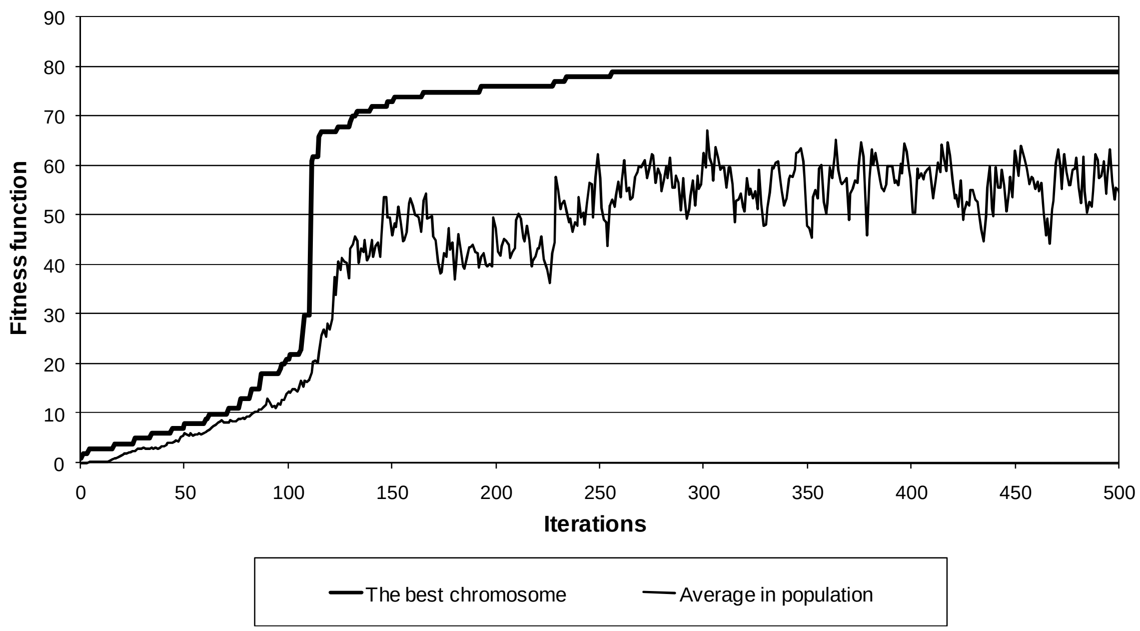

Fig. 8. Course of assessment function changes in the evolutionary optimisation process

REFERENCES

[1] ARABAS J., *Wykłady z algorytmów ewolucyjnych,* WNT, Warsaw, 2001.
[2] DAWKINS R., The S*elfish Gene,* Oxford University Press, 1976.
[3] GLOVER F., *Tabu Search-Part I,* ORSA Journal on Computing, Vol. 1, No. 3, 1989.
[4] GLOVER F., LAGUNA M., *Tabu Search,* Kluwer Academic Publishers, 1997.
[5] GOLDBERG D., *Genetic Algorithms In Search, Optimization, and Machine Learning,* WNT, Warsaw, 1995, (in Polish).
[6] HART W. E., KRASNOGOR N., SMITH J. E., *Recent Advances in Memetic Algorithms,* Springer-Verlag, Berlin Heidelberg, 2005
[7] IZDEBSKI K., *Modelling and simulation of assembly processes,* Wydawnictwo Politechniki Białostockiej, Białystok, 2002, (in Polish).
[8] JANKOWSKI T., REIFUR B., *Planning of assembly sequences based on hybrid evolutionary algorithms*, Archives of Civil and Mechanical Engineering, 2005, Vol. 5, no. 2, 61-69.
[9] JANKOWSKI T., REIFUR B., *Evolutionary algorithm for assembly sequence planning,* in Proceedings on VI International Conference on Artificial Intelligence. AI-19 '2004 (research-development-applications). [Ed. Mieczysław Kłopotek, Jerzy Tchórzewski. Siedlce, 22-23th September 2004]. Siedlce: Publ. House of University of Podlasie, cop. 2004, 73-80.
[10] ŁEBKOWSKI P., *Metody komputerowego wspomagania montażu mechanicznego w elastycznych systemach produkcyjnych.* Wydawnictwa AGH, Cracow, 2000.
[11] MICHALEWICZ Z., *Genetic Algorithms + Data Structures = Evolution Programs,* WNT, Warszawa, 1999, (in Polish).
[12] MOSCATO P., *On Evolution,Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms,* Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, 1989.
[13] MOSCATO P., NORMAN M. G., *A Memetic Approach for the Traveling Salesman Problem Implementation of a Computational Ecology for Combinatorial Optimization on Message-Passing Systems,* in Parallel Computing and Transputer Applications, M. Valero, E. Onate, M. Jane, J. L. Larriba, and B. Suarez, eds., The Netherlands: IOS Press, Amsterdam, 1992, 177-186.
[14] NOWICKI E., *Tabu serach metod in job scheduling problems,* Scientific Papers of the Institute of Technical Cybernetics of the Wrocław University of Technology, No. 101, Wrocław, 1999, (in Polish).
[15] PAWLAK M., *Algorytmy ewolucyjne jako narzędzie harmonogramowania produkcji,* Wydawnictwo Naukowe PWN, Warsaw, 1999.
[16] THOMAS K. K., FISCHER G. W., *Integrating CAD/CAM software for process planning applications,* Journal of Materials Processing Technology, No. 61, 1996, 87-92.