

DESIGNING A LAYOUT USING THE MODIFIED TRIANGLE METHOD, AND GENETIC ALGORITHMS

Ficko, M. & Palcic, I.

University of Maribor, Faculty of Mechanical Engineering, Smetanova 17, 2000 Maribor, Slovenia

E-Mail: mirko.ficko@um.si, iztok.palcic@um.si

Abstract

This paper describes the use of genetic algorithms (GA) for solving the facility layout problem (FLP) within manufacturing systems' design. The paper considers a specific heuristic layout planning method, known as the modified triangle method. This method seeks for an optimal layout solution based on the degrees of flows between workstations. The search for an optimal solution is extremely time-consuming and not suitable for larger systems; therefore we have developed a system based on evolutionary computation. Our paper presents a system, based on GA, and the results obtained regarding several numerical cases from the literature. We propose the usage of this system by presenting numerous advantages over other methods of solving FLP for problem representation and evolutionary computation for solution search.

(Received in September 2012, accepted in June 2013. This paper was with the authors 2 months for 2 revisions.)

Key Words: Facility Layout Problem, Triangle Method, Genetic Algorithms

1. INTRODUCTION

The FLP is an optimization problem that tries to make layouts more efficient by taking into account various interactions between facilities and material-handling systems whilst designing layouts [1, 2]. Sule and Islier indicated that approximately 30-75 % of a product's cost could be attributed to material handling and transportation [3, 4]. An effective layout could reduce costs considerably, including material-handling and transportation costs.

Kusiak and Heragu and Heragu used the term "facility" to represent a machine, workstation, inspection-station, washing-station, locker-room, rest area or any manufacturing or support facility [5, 6]. The FLP is primarily concerned with finding an optimum arrangement of a set of facilities in any layout subject to certain qualitative or quantitative constraints. The FLPs, like most facility design and planning problems, are computationally non-polynomially difficult [1]. Various computer techniques have been developed to support FLP optimization.

Decision making is a very important process, present in various stages of product design [7], workplace design [8] and manufacturing often supported with intelligent algorithms, whose foundation is human cognition of particular field [9]. FLP optimization is even more important in the production environments served by robots, because the complexity of their tasks requires complex robots, thus increasing their potential for failures [10].

Evolutionary computation has shown interesting potential in many different fields, one of them being the design and organisation of manufacturing systems. The optimisation strategies used, based on a population of solutions and often inspired by nature, have led to an increasing number of researchers publishing articles that address several types of problems encountered within the area of manufacturing systems. The designs, configurations and operations of these systems often involve complex decisions, where evolutionary algorithms can play an important role [11].

In this paper, emphasis is placed on the use of evolutionary algorithms, namely genetic algorithms (GA) in order to those tackle problems relating to FLPs, especially layout design. The paper is organised as follows. Introductory background about GA is recalled as well as their usages in the field of layout design and other FLP. A specific layout design heuristic method is presented, based on the work of Schmigalla, known as the “modified triangle method” [12]. We have developed a GA-based system to solve complex problems of layout design, based on the Schmigalla method. The performance of this GA-based system was analysed for several numerical cases. Finally, several future research perspectives are pointed out.

2. GENETIC ALGORITHMS AND THE FACILITY LAYOUT PROBLEM – LITERATURE REVIEW

The term “evolutionary computation” is used to describe computer-based problem solving that uses computational models of evolutionary processes as key elements during its design and implementation [1]. First GAs applications started about 35 years ago, after a presentation of the idea of evolutionary computation by Rechenberg in his work “Evolutionary strategies” [13]. GAs are search algorithms based on mechanisms of natural selection based on Charles Darwin’s theory of evolution and stochastic search techniques [14-16]. They have a high built-in degree of randomness in order to escape from local optima and inferior regions of the solution space. GAs use a highly problem-dependent data structure known as an “organism” to represent candidate solutions. Through parallel processing on a population of randomly generated chromosomes, it speeds up the whole search procedure [1]. GAs were applied to parameter optimization by de Jong [17]. Bäck et al., and Spears et al. noted that GAs traditionally use a more domain independent representation than certain other evolutionary algorithms [18, 19].

GAs search for the best solution within a search space consisting of all possible and feasible solutions. The emphasis is on searching, since they do not use any constructive methods that are based on detailed knowledge of the problem. An environment and organisms have to be created for the solving of problems using GA. The environment is defined by a target function. Organisms represent the solutions to the problems; in fact organisms are coded solutions and not solutions themselves. The solving of an optimization problem starts with a set of randomly created possible solutions, and is called generation. After evaluation the greater probability of cooperating in the selection and variation operations is assigned to those organisms that represent a more accurate solution of the problem. The selection operation assures the survival of the fitter individuals of the population and their advance into the next iteration, called generation. The variation operation has an effect on one or more parental organisms, and their offspring are created from them. After completing selection and variation, a new generation is obtained that is also evaluated. This process is repeated until the termination criterion of the process is fulfilled. This can be a prescribed number of generations or a sufficient quality of the solutions [20].

The use of GAs to solve FLPs has gained popularity over the last twenty years. Tam introduced the coding of layouts as a string of characters of finite length, and used a slicing tree structure to represent a layout as a chromosome of string of characters [21]. He used a clustering algorithm to reflect the proximity requirements of interacting facilities. Gau and Meller introduced a GA with a slicing tree [22], and tried to improve upon the algorithm of Tam [21].

Tate and Smith used GAs to solve quadratic assignment problems (QAPs), and unequal area layout problems [23]. They used the non-binary coding of layouts into chromosomes and gave more importance to mutation in order to generate new solutions. They could not

implement an effective crossover as it disrupted the structural integrity of the parents. Suresh et al. developed another GA to solve the QAP [24] and tried to eliminate some of the shortcomings of the GA proposed by Tate and Smith [23].

Tam used a slicing tree representation to solve the unequal area layout problem using geometric constraints [25]. Azadivir and Wang used a GA with a slicing tree structure whilst using simulation as a means of evaluating the objective function [26]. Wu and Appleton used a slicing tree structure to represent a layout as a chromosome consisting of three strings of characters: facilities string, cutting levels string that represents the aisles, and orientation string [27]. A repair procedure was used to make sure that the chromosome represented a feasible layout. Al-Hakim's paper [28] modifies the representation of Tam and Chan [25] and introduces a new preserving operation, referred to as transplanting, that manages to produce feasible offspring. It also studies the applicability of other genetic operations such as diagonal crossover and cloning in generating feasible offspring.

Islier implemented a GA with a multi-criteria objective function that considered the load, shape, deviation, etc. of each facility [4]. The chromosomes used, however, needed frequent repairs to ensure valid layouts.

There is also vast literature on the dynamic facility layout problem (DFLP) that assumes that the material-handling flows can change over time. This in turn might require layout rearrangement. An extensive literature review was made by Balakrishnan and Hung Cheng [29], two authors who have dealt with this problem in numerous papers. In their paper Balakrishnan and Hung Cheng developed a GA that differs from the existing implementation in three ways: firstly a different crossover operator, secondly, uses of mutation, and thirdly use of a new generational replacement strategy to help increase population diversity [30]. Their work was further improved in Balakrishnan et al., where they created a hybrid GA [31]. Dunker et al. addressed the non-QAP for solving DFLP [32]. Some researches are oriented toward solving the DFLP with fuzzy goals and constraints defined in different spaces [33].

Kochar et al. presented a specific algorithm, called the heuristically operated placement evolution (HOPE) technique for solving the single-floor FLP [34]. They considered the departments of both the equal (QAP problems) and the unequal sizes (non-QAP problems). Another paper that reports on the findings of research into the design of a GA solving modified version of the quadratic assignment formulation of equal and unequal-sized facilities layout problems, is by Tavakkoli-Moghaddain and Shayan [35]. Lee et al. proposed an improved GA to derive solutions for facility layouts that are to have inner walls and passages [36].

Hamamoto proposed a GA approach with an embedded simulation model that allows the user to select those objectives that are important for each particular layout design throughout the pharmaceutical industry [37]. They focused on those layout designs that reduce material handling cost and fulfil the specified throughput rate. Polajnar et al. used simulation modelling to optimize transport costs in several flexible machining centres [38].

3. DESCRIPTION OF MODIFIED TRIANGLE METHOD

The modified triangle method by Schmigalla [12] belongs to the heuristics layout planning methods. Its main characteristic is that a mesh of equilateral triangles is built (where the vertices represent workstations), based on minimising the material-handling cost-function. It is mostly used to find a reasonably good solution for a theoretical layout that is later adjusted to the real requirements. This method is extremely useful when designing new production systems (facilities) or when analysing existing production systems that are confronted with changes during the production processes. There are several presumptions regarding this method:

- The sizes of the workstations are irrelevant, thus all workstations are equal in size and are represented by vertices within the triangular mesh;
- The distances between neighbouring workstations are equal as the triangles are equilateral;
- The cost per distance travelled when making a journey between workstations is constant.

In order to construct this triangular mesh, the following data is needed:

- The list of workstations;
- The degrees of the material flows between workstations (e. g. number of loads per year);
- The directions of the material flows between workstations (production routings).

All these data can be constructed within a flow record chart. The directions of the material flows are unimportant. Both degrees of material flow from one workstation to other workstation (and vice versa) are summarised and written above the diagonal within the flow record chart (matrix).

The FLP addressed here is the assignment of N workstations to M locations within a manufacturing plant. During the manufacturing process, material flows from one workstation to the next workstation until all the processes are completed. The objective of solving the FLP is, therefore, to minimise the total material-handling cost of the system. The following notations are used during the development of the objective function:

- F_{ij} – amount of material flow amongst workstations i and j ($i, j = 1, 2, \dots, N$).
- C_{ij} – unit material-handling cost between the locations of workstations i and j ($i, j = 1, 2, \dots, N$).
- D_{ij} – distance between the locations of workstations i , and j , measured as the number of triangular line-segments.
- C – total cost of material-handling system.

The total cost function is defined as:

$$C = \sum_{i=1}^N \sum_{j=1}^N F_{ij} C_{ij} D_{ij} \quad (1)$$

The evaluation function considered in this paper is the minimisation of material-handling cost, which is the criterion most researchers prefer to apply when solving layout problems.

The procedure proposed by Schmigalla [12] for placing workstations within the triangular mesh is as follows:

- Step 1 – selection of the first two workstations: the pair of workstations with the highest degrees of material flow are sought within the flow record chart. It is reasonable to place workstations with the highest degree of material flow next to each other. This pair is then placed into our triangular mesh as the first (base)-line segment.
- Step 2 – search for the third workstation: all workstations that have not yet been placed in the layout position are looked at, and a sum of degrees of material flow is calculated for each workstation using the pair of first selected workstations. The selected workstation is placed next to the first pair in the mesh, with the aim of constructing the first triangle. There are two possibilities for where to place this workstation. Two additional pairs of workstations emerge when the first triangle has been constructed.
- Step 3 – search for the next workstation: all the workstations that have not yet been placed in the layout position are examined, and a sum of the degrees of material flow for each workstation is calculated using the all pairs of newly-emerged workstations within the triangular mesh. The workstation with the highest sum regarding degree of material flow is sought and then placed into the triangular mesh next to the pair where this highest degree of material flow emerged.
- Step 4 – repeat step 3 until all workstations are placed within the triangle mesh.

As seen from the procedure, the aim is always to place together those workstations that have the highest degrees of material flow. By doing so minimisation of the material-handling cost function is directly considered by trying to minimise the distances between these workstations.

The procedure of calculating, selecting, and placing workstations within the triangle mesh is fairly simple, but only for low numbers of workstations. The calculation for more than 10 workstations becomes extremely time-consuming. This is, however, not the biggest problem. The biggest problem is the fact that the degrees of material flow, and especially the calculating of its sums between workstations and the pairs of already placed workstations within the mesh, start to repeat themselves. When selecting a new workstation for placing within the triangular mesh, several possibilities can emerge (several equal, yet highest values for the degrees of material flow). This means that there are more options for building a layout. There can be more than one candidate (workstation) to be selected next or / and there are several possibilities regarding already emerged pairs where new candidate can be placed. Even when there are less than 10 workstations, a large number of possible layouts can emerge that make these cases extremely difficult to solve. When the number of workstations increases the number of possible solutions also goes up. The aim is, of course, to find the best solution from all the possible solutions. The best solution is the one with the lowest material-handling costs. To put it other way, the optimal solution is the one where the individual distances between those all pairs of workstations that have mutual material flows, are equal to one triangular line-segment.

As already mentioned, solving these problems is extremely difficult. Therefore, a system for solving this problem is proposed using the GA that uses efficient problem representation, as made by Schmigalla [12].

4. DESCRIPTION OF THE GA-BASED SYSTEM

This presented system uses evolutionary computation for the searching of an optimal facility layout. This system represents the adaptation of a GA method to the actual problem of FLP, where the layout space and correlations (degree of flows) between workstations is represented by a triangular mesh as by the Schmigalla method [12]. The system is presented in Fig. 1. In the first step, three types of information have to be inputted into the system. The first type relates to the generation of mesh, the second is problem definition (flow record chart), and the third the parameters of the GA. From this point on, the automated procedure takes over. The creation of an initial population randomly creates a new generation; the only limitation used is the fact that the randomly-generated organisms have to present a feasible solution. After creating an initial population of organisms (solutions), the value of the cost function is calculated for all of them. The solution with the best value is graphically represented, together with cost-function value for human interactive control over its evolution (design process). The evolution starts after the first population is created. Evolution is a loop where generation follows generation. New generations are created out of the current generation with the help of selection, reproduction, crossover, and mutation operations. Selection is an operation which ensures the creation of new organisms from better-than-average organisms. The selection of a genetic operation is based on tournament selection probability. After the new population is created, the fitness is applied and the stop criterion is checked. The stop criterion is the fixed number of generations, since the optimal value of the cost-function is unknown in advance.

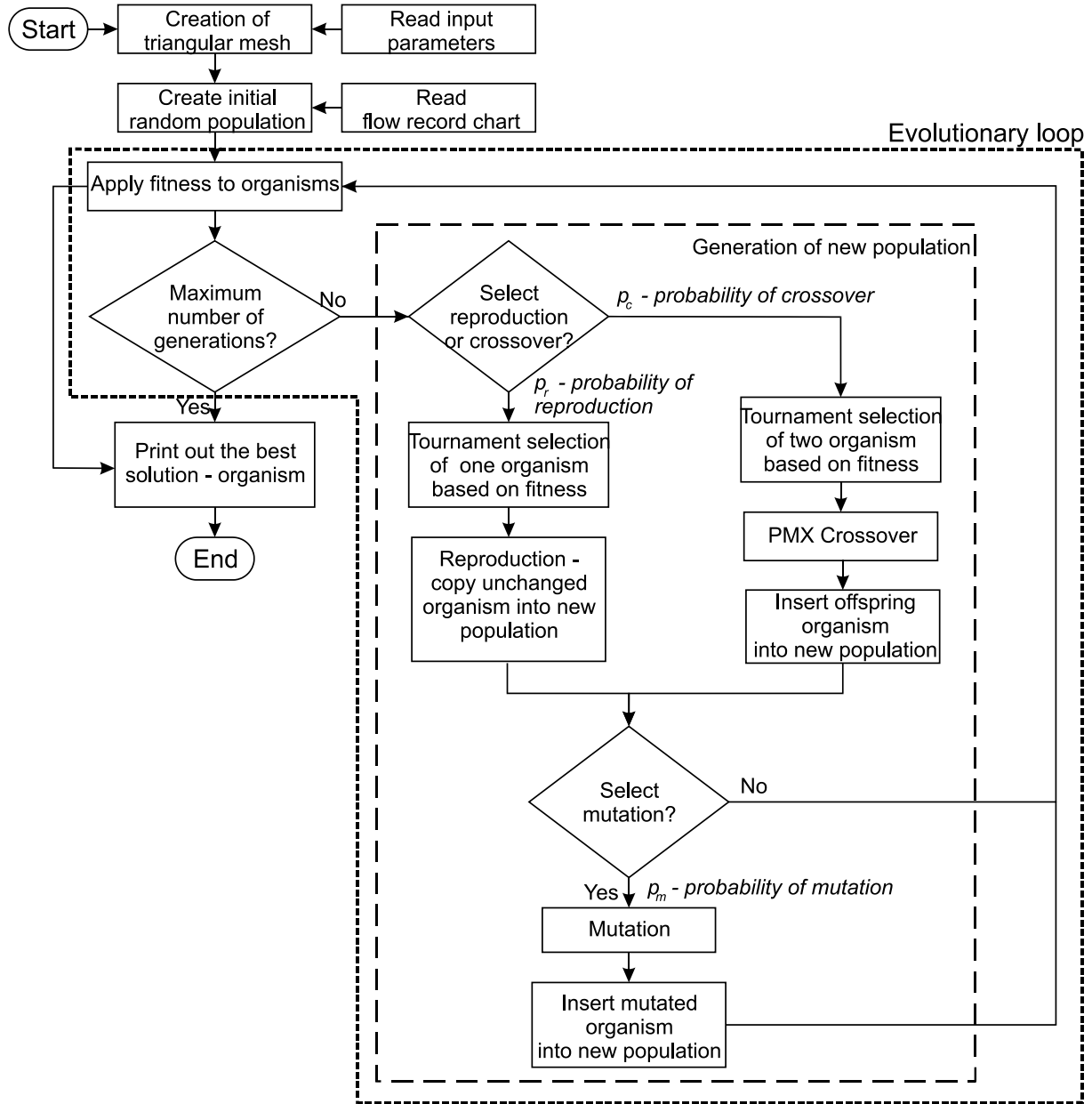


Figure 1: Model of genetic algorithm for triangular mesh based layout.

4.1 Coding

In order to solve using GAs, the possible solution has to be coded to the appropriate form. In other words, the solutions must be suitably-coded into organisms. Permutation coding was chosen since it ensures the simple, natural coding of organisms and, therefore, a fast computational speed. The modified triangle method by Schmigalla [12] uses the representation of layout solutions with the mesh based on equilateral triangles. Workstations are set to the mesh's vertices. The method constructively forms the solution, which can grow in any direction. Therefore, by solving the problem of optimal layout within a triangular mesh by a GA the mesh size has to be defined. When defining the mesh few limitations have to be taken into account. If N is the number of workstations for placing, I is the number of columns, and J is the number of rows, then for the feasibility of the solutions, the following relationship has to be true:

$$N \leq I \cdot J = M \quad (2)$$

This method of representing solutions enables ‘pre’ forming of the final solution. If the actual available space for the layout has some limitations, such as e.g. limitations regarding the maximal number of rows, it can already be included in this phase during designing. In extreme case we can force creation linear (one-row) layout:

$$I = N \text{ and } J = 1 \quad (3)$$

It is beneficial to keep the number of rows and columns as low as possible since the solution converges quicker and the feasibility of solution is improved. If there is no limitation regarding the space shape (we have available unconstrained space), unnecessary large layout space should be avoided. In this case we suggest that the number of columns equals the number of workstations and the number of rows is only half of the number of workstations:

$$I = N \text{ and } J = N / 2 \quad (4)$$

Eqs. (3) and (4) are respectively representing the extreme cases of size of layout space. In first case it is limited to the maximum and in the second case is large enough to get global optimal solution for the given number of workstations.

Fig. 2 presents a representation of the available space using a mesh of equilateral triangles, in rows and columns.

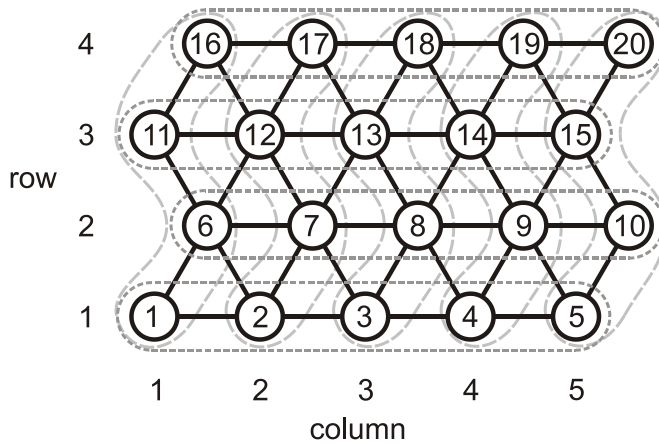


Figure 2: Space for workstation layout represented by a mesh of equilateral triangles.

The variant of permutation coding includes the designation of a workstation combined with its location (vertex). Vertices could be represented by coordinates or by designations, as shown in Fig. 2. Transformation between vertex designation P and placement in columns (i) and rows (j), is as follows:

$$P_{i,j} = I \cdot (j - 1) + i \quad (5)$$

An organism is a set of vertex designations (alleles) and the location of the vertex within an organism (locus) which carries the workstation's designation, as shown in Fig. 3.

This type of coding is very simple and therefore fast for computation. Many genetic operators exist for this type of coding [39].

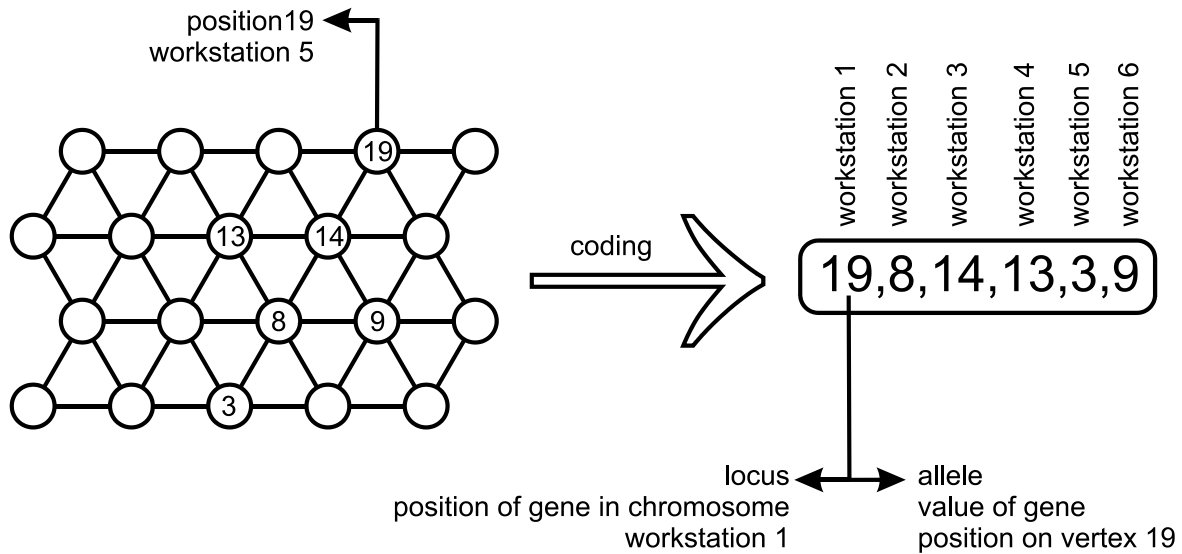


Figure 3: Coding of organism.

4.2 Evolutionary and genetic operators

The evolutionary operations of selection, reproduction, and the genetic operation of crossover and mutation, were used in this model. Selection is the motive force during the development of GAs. It leads to the searching into of those search space areas promising the best solutions. In this system, tournament selection was used with a tournament count of 2 [40]. A greater number of tournaments, in this case, would be unfavourable since the evolution in most cases sinks into the local optimum. In next step organisms obtained by selection were the subject of reproduction, crossover or mutation. The probabilities of reproduction p_r , of crossover p_c and mutation p_m are input parameters set before the evolution starts.

During reproduction, the unaltered selected organisms are forwarded to the next generation. The crossover operation takes place on two organisms at a time out of population by combining their genes. The crossover ensures the exchange of genes amongst organisms. However, the crossover and mutation operations create new organisms (solutions) with major differences: the crossover uses genetic material from selected organisms and recombines it, whilst the mutation adds new genetic material. In the presented case, this means: the crossover operation creates a new layout as a combination of two existing ones, and mutation creates a new layout by placing workstations on those vertices that are unoccupied. Therefore, the crossover operation only uses the vertices of existing solutions (Fig. 4), whilst the mutation adds new vertices to the solutions (Fig. 5).

For crossover operation a variation of partial-mapped crossover (PMX) was used which was proposed by Goldberg and Lingle [41]. In view of the operation, PMX is a modification of one or two-point crossover but additionally applies a mapping relationship to repair offspring that have duplicate genes. Proposed coding namely does not allow duplicate genes in organism, because this would present an unfeasible solution with two workstations on same place. The crossover works as follows (Fig. 4):

- Randomly selects two cut points on both parents. In order to create an offspring, the substring between the two cut points in the first parent replaces the corresponding substring in the second parent.
- The inverse replacement is applied outside of the cut points, in order to eliminate duplicates.

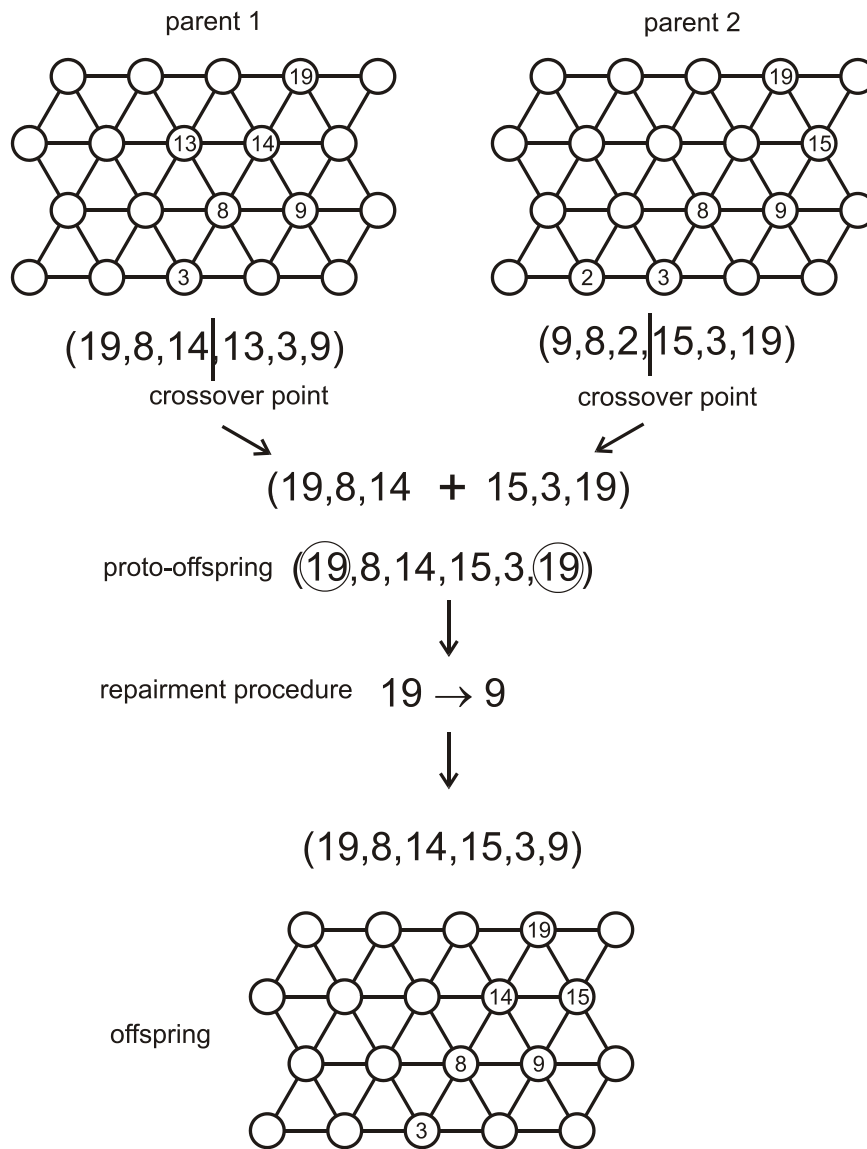


Figure 4: Partial-mapped crossover (PMX).

Mutation, on the other hand, introduces new genetic material into the organisms at the level of the genes. The aim of mutation is the conservation of variety within the population, thus avoiding premature convergence (to a suboptimal solution). In general, mutation [39] replaces the genes in the organisms by those lost genes omitted from the evolution but able to contribute to better results in other organisms. It also introduces into the population those genes which were absent in the initial generation. Simple mutation was used in the presented system. In fact the mutation represents the movement of one randomly selected workstation from one vertex-to-randomly-selected free vertex, as shown in Fig. 5.

In-depth research of evolutionary and genetic operators was avoided when defining this system. The selection and creation of these operators was made with respect to computational simplicity. The used operators ensure, together with selected coding, the creation of offspring that represent feasible layouts. By use of this system, it is necessary to determine the factors of probabilities p_c and p_m wisely. The use of sole-selection leads to a filling of the entire population by copies of the best individuals and, therefore, no improvement. The use of only selection and crossover leads to a good but sub-optimal solution because it is limited to only existing genetic material. Only when using mutation that implies random motion within the

search space, the system together with selection and crossover approaches leads to a good solution in reasonable time.

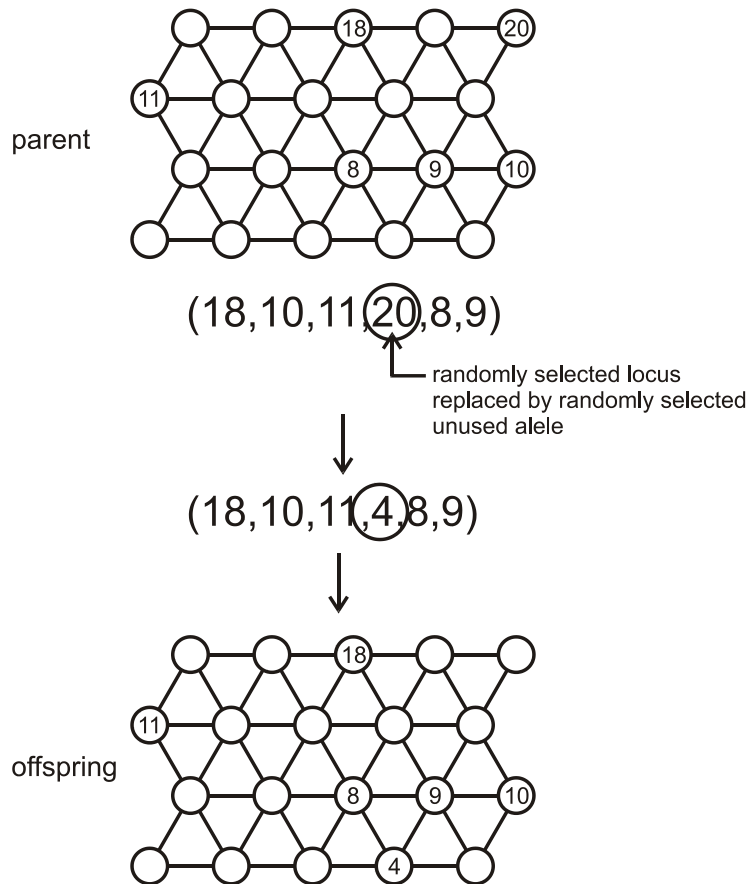


Figure 5: Applied mutation.

5. EXAMPLES

The presented system was intensively evaluated on two cases, already solved by Schmigalla [12]. The first case consisted of 9 workstations and was proven to be efficiently solved by Schmigalla method. The From/To matrix of this case is shown in Table I. The second case was more complex regarding the flow record chart and number of workstations. It consisted in total of 11 workstations [42]. After this evaluation which provided very positive feedback, the system was evaluated with an even bigger problem, consisted of 25 workstations in total [43]. This case proved the usability and capability of the presented system on larger, complex cases. The chosen size of the mesh or the number of nodes used for the placing of workstations, was large enough as such, that it did not represent any limitation. For determination of the mesh size the Eq. (4) was used. In the first case (with 9 workstations) the size of mesh was intentionally for testing purposes set on 9×9 . If it were necessary to avoid the space limitation completely, the number of rows and columns could be set as equal to the number of workstations. On the other hand, in an extreme case the layout in one row could be obtained. In this case the shape of layout was not predefined and a search was made for the global optimum for the layout within the open space. The search-space is therefore limited, although very large. Namely 9 workstations can be set, on the 9×9 mesh on approximately 10^{17} different ways.

Table I: From/To matrix for the first case (with 9 workstations).

From / To	1	2	3	4	5	6	7	8	9
1	0	90	170	140	0	10	0	0	0
2	0	0	90	0	0	0	0	0	0
3	0	0	0	180	100	0	40	50	0
4	0	0	0	0	210	10	30	0	0
5	0	0	0	0	0	90	80	180	0
6	0	0	0	0	0	0	0	110	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	40
9	0	0	0	0	0	0	0	0	0

At the beginning of the testing, the true potential of this system was clear, as when searching for an optimal layout for around 10 workstations, the solutions converged within a few seconds. Even in those cases consisting of several ten-workstations, the converged solutions were obtained in few minutes. Good or even optimal solutions were obtained at virtually all runs. The in-depth analysis of solutions obtained after several ten-runs produced, for the first case, the same result as the best known solution, and in second case an even better solution than the case author [42]. During the solution analysis the fact that the same solution can actually be presented in different ways had to be taken into consideration as it can be namely rotated or mirrored. There are six different representations of the same solution.

After a few runs of the system for solving a case with 9 workstations, the solutions were obtained as shown in Fig. 6. The obtained solutions were comprised of those solutions presented in literature and two alternatives, which lie within 2.5 % margin of the optimal solution. After a short analysis the core solution can be obtained, as presented using bolder lines in Fig. 6, together with those parts of solution that are less important for the target function. Fig. 6 presents best layout for the case with 9 workstations on the left, whilst in the middle, and on the right, two possible (alternative) solutions are presented. The values of the total transportation cost are written at the bottom. As can be seen the alternatives are quite different from the best solution.

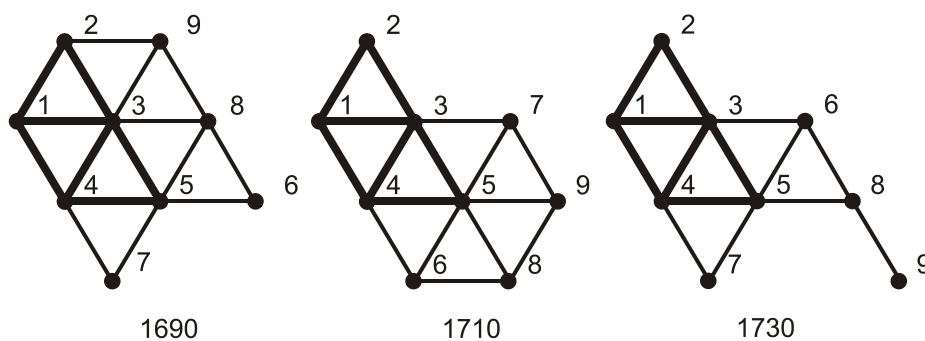


Figure 6: Three layout solutions for the case with 9 workstations.

For the case with 11 workstations an even better solution was obtained than by the author who solved this case using the Schmigalla method, as shown in Fig. 7. In this case, the system found even more similarly good solutions within the 2.5 % margin of the value target function.

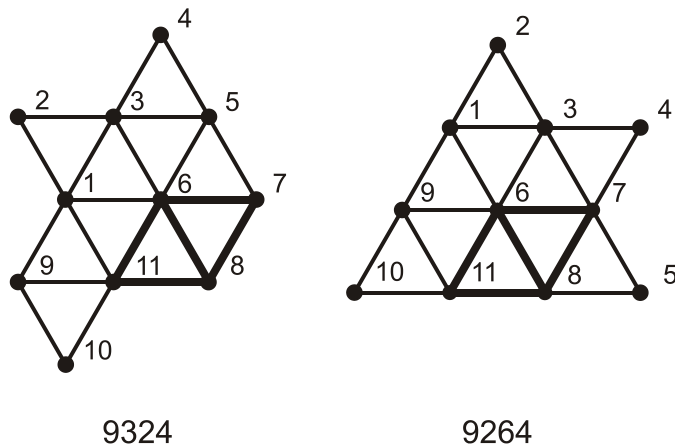


Figure 7: The best solution of case author [39] on the left and the best solution found by the system on the right.

Searching for the best layout in the case of 25 workstations took more time. Despite this fact, the solutions converged as expected. There was no value for the good (or optimal) solution, because this case was not solved by the Schmigalla method, as were the former two. 276 runs (with 500 generations in each run) were made in order to ensure as good a solution as possible. The solution converged very well, at every run of the evolution. Fig. 8 shows the value changes of best solution during the evolution. Changes of A series of solutions were obtained, where the worst one had a 17 % higher value of cost function than the best one. This test proved the capacity of the presented system; it is easy to imagine using this system on cases consisting of many more workstations than our largest tested case.

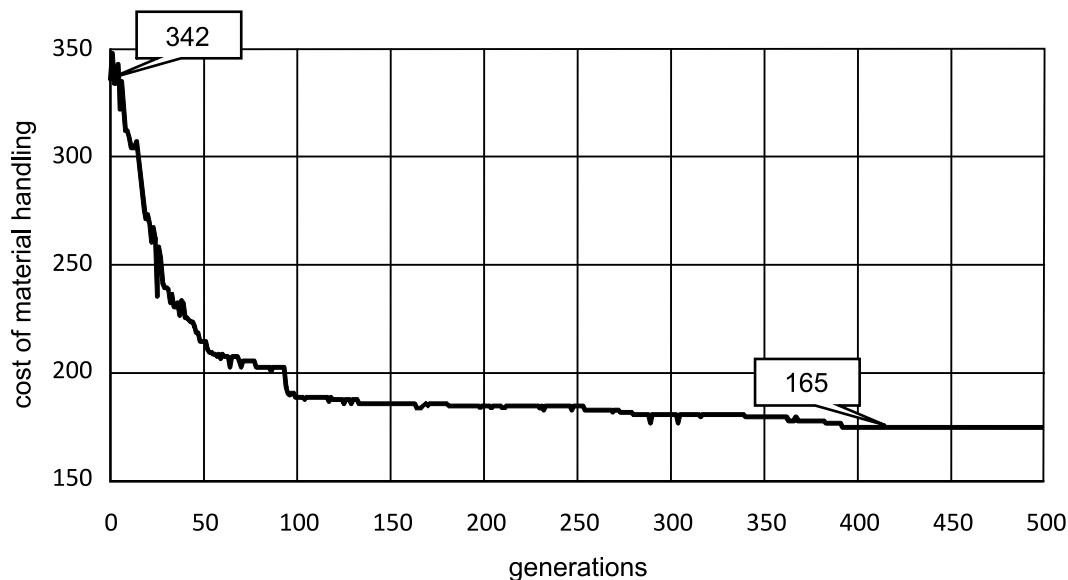


Figure 8: Fitness function values of best solution during the evolution.

6. CONCLUSION

The presented system follows the solving of FLP problem based on Schmigalla method; it uses a similar presentation to the problem but the building of the solution does not follow any deterministic principles. Namely, with larger problems the set of building rules becomes impractical and does not ensure good results. For the creation of a good solution we used

evolutionary computation which, together with a simplification of the Schmigalla representation and a robust algorithm, opens-up possibilities of searching for optimal layouts for extremely large problems. The presented system enables a fast and robust approach to a presented problem.

We have to note here that a good solution in industrial practice is often not a mathematically optimal solution but a good technological solution based on a mathematical solution. The most important from the criteria is the total length of the travel path with a time period, but however this is not the sole criteria. Often this is the only criteria when searching for an optimal solution and other, mostly technological, criteria are added later within the phase of creating a real layout by humans. The involvement of significant technological criteria within the phase of searching for an optimal solution is unfruitful since it is often hard to describe the mostly subjective wishes of the system-user. Only the user knows most of the design goals and creates an altered solution that is based on the optimal solution. At this point, the set of solutions created by evolutionary computation presents their strengths. The user can choose from a set of similarly good solutions according to all the technological and environmental limitations. At this point, we see the possibility of proceeding with research that would ensure the creation of good solutions, and provide knowledge of those elements of a good solution that can support humans by the creation of a real layout.

There are several future challenges that will be addressed using the presented GA-system. We will add the real sizes of workstations within our system and apply the possibility of different transportation costs amongst each pair of workstations. Another idea is to include the types of machines, and find optimal solutions based on placing the same types of workstations together (design of process layout). As we have already mentioned, we are also interested in finding the core solution for possible layouts – the part of the layout that is extremely important for the target function. This core-solution is a base solution of the optimal layout that cannot be changed if we want to keep transportation costs to a minimum.

The primary motivation for our work was to design a tool for generating ‘good’ and ‘robust’ layouts that could be used by a decision maker to apply subjective judgment. We have used the evolutionary strength of GAs to develop such a tool. Its strength lies in the fact that it provides the designer with a number of ‘good’ solutions.

REFERENCES

- [1] Shayan, E.; Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure, *International Journal of Production Research*, Vol. 42, No. 19, 4055-4067, [doi:10.1080/00207540410001716471](https://doi.org/10.1080/00207540410001716471)
- [2] Hammadi, M.; Choley, J. Y.; Penas, O.; Louati, J.; Rivière, A.; Haddar, M. (2011). Layout optimization of power modules using a sequentially coupled approach, *International Journal of Simulation Modelling*, Vol. 10, No. 3, 122-132, [doi:10.2507/IJSIMM10\(3\)2.183](https://doi.org/10.2507/IJSIMM10(3)2.183)
- [3] Sule, D. R. (1994). *Manufacturing Facilities: Location, Planning, and Design*, PWS, Boston
- [4] Islier, A. A. (1998). A genetic algorithm approach for multiple criteria facility layout design, *International Journal of Production Research*, Vol. 36, No. 6, 1549-1569, [doi:10.1080/002075498193165](https://doi.org/10.1080/002075498193165)
- [5] Kusiak, A.; Heragu, S. S. (1997). The facility layout problem, *European Journal of Operational Research*, Vol. 29, No. 3, 229-251, [doi:10.1016/0377-2217\(87\)90238-4](https://doi.org/10.1016/0377-2217(87)90238-4)
- [6] Heragu, S. S. (1997). *Facilities Design*, PWS, Boston
- [7] Kaljun, J.; Dolsak, B. (2012). Ergonomic design knowledge built in the intelligent decision support system, *International Journal of Industrial Ergonomics*, Vol. 42, No. 1, 162-171, [doi:10.1016/j.ergon.2011.11.009](https://doi.org/10.1016/j.ergon.2011.11.009)
- [8] Polajnar, A.; Vujica-Herzog, N.; Buchmeister, B.; Jevsnik, S. (2012). Strains and stresses of workers caused by exposure to noise, *Collegium Antropologicum*, Vol. 36, No. 3, 899-909

- [9] Sancin, U.; Dobravec, M.; Dolsak, B. (2010). Human cognition as an intelligent decision support system for plastic products' design, *Expert Systems with Applications*, Vol. 37, No. 10, 7227-7233, [doi:10.1016/j.eswa.2010.04.005](https://doi.org/10.1016/j.eswa.2010.04.005)
- [10] Dev Anand, M.; Selvaraj, T.; Kumanan, S. (2012). Fault detection and fault tolerance methods for industrial robot manipulators based on hybrid intelligent approach, *Advances in Production Engineering & Management*, Vol. 7, No. 4, 225-236
- [11] Pierreval, H.; Caux, C.; Paris, J. L.; Viguier, F. (2003). Evolutionary approaches to the design and organization of manufacturing systems, *Computers & Industrial Engineering*, Vol. 44, No. 3, 339-364, [doi:10.1016/S0360-8352\(02\)00195-X](https://doi.org/10.1016/S0360-8352(02)00195-X)
- [12] Schmigalla, H. (1995). *Fabrikplanung: Begriffe und Zusammenhänge*, Fachbuchverlag, Leipzig
- [13] Rechenberg, I. (1974). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog Verlag, Stuttgart
- [14] Holland, J. H. (1962). Outline for a logical theory of adaptive systems, *Journal of the ACM*, Vol. 9, No. 3, 297-314, [doi:10.1145/321127.321128](https://doi.org/10.1145/321127.321128)
- [15] Holland, J. H. (1975). *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor
- [16] Sedlar, D.; Lozina, Z.; Vucina, D. (2011). Comparison of genetic and bees algorithm in the finite element model update, *Transactions of FAMENA*, Vol. 35, No. 1, 1-13
- [17] De Jong, K. (1975). *An analysis of the behavior of a class of genetic adaptive systems*, Doctoral Dissertation, University of Michigan, Ann Arbor
- [18] Bäck, T.; Hammel, U.; Schwefel, H.-P. (1997). Evolutionary computation: comments on the history and current state, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 3-17, [doi:10.1109/4235.585888](https://doi.org/10.1109/4235.585888)
- [19] Spears, W. M.; De Jong, K. A.; Bäck, T.; Fogel, D. B.; de Garis, H. (1993). An overview of evolutionary computation, Brazdil, P. (Ed.), *Machine Learning: ECML-93*, Springer-Verlag, Berlin, 442-459
- [20] Udhayakumar, P.; Kumanan, S. (2010). Task scheduling of AGV in FMS using non-traditional optimization techniques, *International Journal of Simulation Modelling*, Vol. 9, No. 1, 28-39, [doi:10.2507/IJSIMM09\(1\)3.139](https://doi.org/10.2507/IJSIMM09(1)3.139)
- [21] Tam, K. Y. (1992). Genetic algorithms, function optimization, and facility layout design, *European Journal of Operational Research*, Vol. 63, No. 2, 322-346, [doi:10.1016/0377-2217\(92\)90034-7](https://doi.org/10.1016/0377-2217(92)90034-7)
- [22] Gau, K.-Y.; Meller, R. D. (1999). An iterative facility layout algorithm, *International Journal of Production Research*, Vol. 37, No. 16, 3739-3758, [doi:10.1080/002075499190022](https://doi.org/10.1080/002075499190022)
- [23] Tate, D. M.; Smith, A. E. (1995). A genetic approach to the quadratic assignment problem, *Computers & Operations Research*, Vol. 22, No. 1, 73-83, [doi:10.1016/0305-0548\(93\)E0020-T](https://doi.org/10.1016/0305-0548(93)E0020-T)
- [24] Suresh, G.; Vinod, V. V.; Sahu, S. (1995). A genetic algorithm for facility layout, *International Journal of Production Research*, Vol. 33, No. 12, 3411-3423, [doi:10.1080/00207549508904881](https://doi.org/10.1080/00207549508904881)
- [25] Tam, K. Y. (1998). Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings, *International Journal of Production Research*, Vol. 36, No. 12, 3253-3272, [doi:10.1080/002075498192058](https://doi.org/10.1080/002075498192058)
- [26] Azadivar, F.; Wang, J. (2000). Facility layout optimisation using simulation and genetic algorithms, *International Journal of Production Research*, Vol. 38, No. 17, 4369-4383, [doi:10.1080/00207540050205154](https://doi.org/10.1080/00207540050205154)
- [27] Wu, Y.; Appleton, E. (2002). The optimisation of block layout and aisle structure by a genetic algorithm, *Computers & Industrial Engineering*, Vol. 41, No. 4, 371-387, [doi:10.1016/S0360-8352\(01\)00063-8](https://doi.org/10.1016/S0360-8352(01)00063-8)
- [28] Al-Hakim, L. (2000). On solving facility layout problems using genetic algorithms, *International Journal of Production Research*, Vol. 38, No. 11, 2573-2582, [doi:10.1080/00207540050031922](https://doi.org/10.1080/00207540050031922)
- [29] Balakrishnan, J.; Cheng, C. H. (1998). Dynamic layout algorithms: a state-of-the-art survey, *Omega*, Vol. 26, No. 4, 507-521, [doi:10.1016/S0305-0483\(97\)00078-9](https://doi.org/10.1016/S0305-0483(97)00078-9)
- [30] Balakrishnan, J.; Cheng, C. H. (2000). Genetic search and the dynamic layout problem, *Computers & Operations Research*, Vol. 27, No. 6, 587-593, [doi:10.1016/S0305-0548\(99\)00052-0](https://doi.org/10.1016/S0305-0548(99)00052-0)

- [31] Balakrishnan, J.; Cheng, C. H.; Conway, D. G.; Lau, C. M. (2003). A hybrid genetic algorithm for the dynamic plant layout problem, *International Journal of Production Economics*, Vol. 86, No. 2, 107-120, [doi:10.1016/S0925-5273\(03\)00027-6](https://doi.org/10.1016/S0925-5273(03)00027-6)
- [32] Dunker, T.; Radons, G.; Westkamper, E. (2005). Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem, *European Journal of Operational Research*, Vol. 165, No. 1, 55-69, [doi:10.1016/j.ejor.2003.01.002](https://doi.org/10.1016/j.ejor.2003.01.002)
- [33] Naadimuthu, G.; Liu, Y. L.; Lee, E. S. (2012). Multi-period production planning under fuzzy conditions, *Advances in Production Engineering & Management*, Vol. 7, No. 1, 61-73
- [34] Kochhar, J. S.; Foster, B. T.; Heragu, S. S. (1998). HOPE: A genetic algorithm for the unequal area facility layout problem, *Computers & Operations Research*, Vol. 25, No. 7-8, 583-594, [doi:10.1016/S0305-0548\(97\)00100-7](https://doi.org/10.1016/S0305-0548(97)00100-7)
- [35] Tavakkoli-Moghaddain, R.; Shayan, E. (1998). Facilities layout design by genetic algorithms, *Computers & Industrial Engineering*, Vol. 35, No. 3-4, 527-530, [doi:10.1016/S0360-8352\(98\)00150-8](https://doi.org/10.1016/S0360-8352(98)00150-8)
- [36] Lee, K.-Y.; Han, S.-N.; Roh, M.-I. (2003). An improved genetic algorithm for facility layout problems having inner structure walls and passages, *Computers & Operations Research*, Vol. 30, No. 1, 117-138, [doi:10.1016/S0305-0548\(01\)00085-5](https://doi.org/10.1016/S0305-0548(01)00085-5)
- [37] Hamamoto, S. (1999). Development and validation of genetic algorithm-based facility layout – a case study in the pharmaceutical industry, *International Journal of Production Research*, Vol. 37, No. 4, 749-768, [doi:10.1080/002075499191508](https://doi.org/10.1080/002075499191508)
- [38] Polajnar, A.; Buchmeister, B.; Leber, M. (1995). Analysis of different transport solutions in the flexible manufacturing cell by using computer simulation, *International Journal of Operations & Production Management*, Vol. 15, No. 6, 51-58, [doi:10.1108/01443579510090336](https://doi.org/10.1108/01443579510090336)
- [39] Gen, M.; Cheng, R. (1997). *Genetic algorithms and engineering design*, Wiley, New York
- [40] Miller, B. L.; Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise, *Evolutionary Computation*, Vol. 4, No. 2, 113-131, [doi:10.1162/evco.1996.4.2.113](https://doi.org/10.1162/evco.1996.4.2.113)
- [41] Goldberg, D. E.; Lingle, R. (1985). Alleles, Loci and the Traveling Salesman Problem, *Proceedings of the First International Conference on Genetic Algorithms*, Pittsburgh, 154-159
- [42] Prêt, U. (2012). Fabrikplanung (Übung), from: <http://www.uwe-pret.de/fabrikplanung/skripte/stapel3/getriebe.pdf>, accessed on 04-01-2012
- [43] Ficko, M.; Brezovnik, S.; Klancnik, S.; Balic, J.; Brezocnik, M.; Pahole, I. (2010). Intelligent design of an unconstrained layout for a flexible manufacturing system, *Neurocomputing*, Vol. 73, No. 4-6, 639-647, [doi:10.1016/j.neucom.2009.06.019](https://doi.org/10.1016/j.neucom.2009.06.019)