genetic algorithm, simulation modeling, manufacturing system

*Tomasz WIŚNIEWSKI, Szymon RYMASZEWSKI**

THE USE OF SIMULATION AND GENETIC ALGORITHM WITH DIFFERENT GENETIC OPERATORS TO OPTIMIZE MANUFACTURING SYSTEM

Abstract

The article depicts an evolutionary approach to simulation based optimization of a typical manufacturing system. Genetic algorithm with four different variants of genetic operators (crossover operator and type of selection) is compared to find the best optimization method. A comprehensive discussion of the genetic algorithm results obtained from the simulation model was also presented.

1. INTRODUCTION AND A LITERATURE REVIEW

Profit from operating activities is an obvious aim of any manufacturing system. Therefore, optimization of production systems increases their productivity, flexibility and generally financial benefits. The aim of this paper is to compare the genetic algorithm of different types of genetic operators (selection method and crossover operator) for the optimization task in a typical manufacturing system. Exact algorithms within a reasonable time frame can only solve small problems. Thus, heuristic and metaheuristic algorithms (especially evolutionary algorithms) have been widely applied to solve problems in complex manufacturing systems [1]. The computer simulation environment Arena from Rockwell has been used for this purpose. Typical model of the manufacturing system will be optimize using Visual Basic (VBA) implemented with genetic algorithm.

---

* West Pomeranian University of Technology, Szczecin,
Department Of Information Systems Engineering, 71-210 Szczecin, Żołnierska 49,
e-mail: twisniewski@wi.zu.edu.pl, srymaszewski@wi.zu.edu.pl

A simulation modelling as an evaluative tool for stochastic systems has facilitated the ability to obtain performance measure estimates under any given system configuration [2]. Simulation is a very powerful tool often used in the design phase of manufacturing systems. Performance of various layout alternatives can be studied using simulation. Important features of the simulation indicated in [3] are: a possibility to study production systems without interfering with the real production system, and the ability to compress time, depending on your needs.

Genetic algorithm (GA) is the universal tool for combinatorial optimization problems. It belongs to evolutionary algorithms and have been applied to a variety of function optimization problems. GAs were shown to be highly effective in searching a large, complex response surface even in the presence of difficulties such as high dimensionality, multimodality and discontinuity. Many evolutionary algorithms have been developed in literature and implemented to solve manufacturing problems, due to the qualitative character of the variable and scale of the problem. This methodology is used in many fields such as manufacturing, engineering, business, science, etc [4].

The GA is good enough tool that is used not only to the optimization of the production systems, but also in forecasting such as energy consumption [5] and job shop scheduling problems [6]. The publication of Paul and Chanev [4] shows how, in practice, one can use computer simulations with GA to optimize the real production system. They created a simulation model of foundry, depending on control parameters such as the number of cranes, furnaces, and the number and size of cars. The use of optimization using a simple GA caused that the system has become less expensive and it did not generate waste. Entriken, Vössner [7] used the simulation with GA to optimize the production line of printed circuit boards. Zhang and Gen [6] presented a multistage operation-based GA, simplifying the chromosome representation to apply crossover and mutation operators in an optimal strategy. Cakar and Yildirim [8] propose a neuro-genetic decision support system coupled with simulation to design a job shop manufacturing system to obtain the optimal amount of resources in each workstation in conjunction with the right dispatching rule to schedule.

A popular control parameter analyzed in manufacturing systems is the size of the buffer. Its performance affects the allocation of the production line and the cost of production in progress. Storing too much Work-In-Progress will also be a cost associated with the freezing of capital. So it is important to optimally determine the value of the buffer in order to ensure maximum system performance with the lowest possible cost of Work-In-Progress. Can, Beham and Havey [9] devoted a whole article to the problem of buffer allocation and optimization of this problem by using GA and simulation. Many authors examine different variants of GA [1]. Eskandari et al. [10] presented an improved GA applied to the multicriteria optimization problems of simulation modeling. Konstam et al. [11] indicated the effectiveness of distributed GAs with multivariate crossover in optimization of function with a large number of independent variables. Their results showed that this algorithm has the unique potential to optimize such a function.

In this paper the multi-product manufacturing system is analyzed, i.e. parts for different kinds of final products are processed at the same time within a single manufacturing system. Each workstation in the system is preceded by a buffer. The criterion adopted in optimization task is profits derived from production. Control parameters during the optimization process are the size of buffers and amounts of resources (people, machines) in a given workstation. The study will use four variants of the genetic algorithm which differ from each other in types of crossover operator and the type of selection. Two types of crossover operator will be applied, and two types of selection methods. Each operator will be examined with any type of selection which will allow to compare four different types of genetic algorithm, and select the most suitable to the task of this kind.

The rest of the paper is organized as follows. Built simulation model with the control parameters was presented in Section 2. Section 3 contains a general scheme of optimization with a descriptions of criterion function and genetic algorithm. Section 4 is an analysis of the results obtained in simulations. Section 5 contains a summary of the discussion.

2. PRESENTATION OF THE MODEL

A typical manufacturing system was modeled to show the opportunities of using computer simulation. The model was constructed in Rockwell Arena simulation environment. There are five workstations and three types of product. Each product in the model passes through the system according to different sequences, which results in that the workstations have not equally utilization. Each workstation is preceded by a buffer of different capacity. An important parameter for the workstation is the amount of resources (capacity). Best value for this parameter for each of the workstation will be searched during the optimization process. Capacity values for individual workstation may range from 1 to 4. Customers orders' arrival patterns are exponential (100 pieces of each type of product a day). Components wait before first buffer, which do not generate the cost of Work-In-Progress. If only space in first buffer is available component goes through it technological itinerary and since then starts to generate the cost of Work-In-Progress (0.5 units per hour). Processing times for each workstation, which is different for each type of product, has been shown in

Table 1. Utilization costs is set at 60 units and cost of idle time is set at 50 units. Utilization costs is higher because of the assumption that while work the material is consumed.

Tab. 1. Processing times for each workstation (min.)

	Station 1	Station 2	Station 3	Station 4	Station 5
Product 1	10	8	10		10
Product 2	10	12	10		14
Product 3	10			12	12

2.1. Control parameters

Buffer sizes and resource capacity at each workstation have been selected as control parameters that could be used to optimize manufacturing system performance. Choice was dictated due to the fact that these are the parameters that you can easily change in practice compared to the others, and they allow significant improvement of the system through small changes. The parameters are discrete – different size of the buffer before each workstation takes integer values in the range from 1 to 32. While resource capacity at each workstation takes integer values in the range from 1 to 4. Properly selected system control parameters allow to affect the amount of Work-In Progress, eliminate the bottlenecks and provide the possibility of adjusting the productivity of the system. Number of different combinations of all these parameters is: 32^5 (buffer size) x 4^5 (resource capacity) = 34 359 738 368 combinations. It means that there are so many different settings of the system using only these indicated parameters, that will cause a very time-consuming calculations. Therefore, to find the optimal settings the GA will be used. Simulation runs parameters were selected in order to assure reliable results, hence the length of each replication was set at 30 days and 16 working hours a day. It can be assumed that this corresponds to a month of work in two shifts, 7 days a week. Each simulation run consists 3 replication and warm-up time is set to 1 day.

3. GENETIC ALGORITHM

Genetic algorithm (GA) belong to heuristic method that is frequently used with simulation-based optimization. GAs are based on the mechanisms of natural evolution. In this paper we compare four types of GA with different kind of selection and crossover. Algorithm was implemented in VBA. It changes control parameters of the model and the fitness function value is returned for each created solution. A general structure of the developed algorithm is described below:

STEP 1 Initialization (Create a population with random chromosome)
STEP 2 Fitness function evaluation (Evaluate a function value for all chromosome in population)
STEP 3 Stop condition (If true then end)
STEP 4 Selection (Select parents for offspring in next population)
STEP 5 Crossover (Crossover select parent and create offspring for next population)
STEP 6 Mutation (Go to STEP 2) (Mutation gene in offspring and create new population)

In GA solution is called chromosome which is a set of genes (one gene is one variable). A set of chromosome is a population. The main principle of operation in GA is created new population by using previous population of chromosomes and genetic operators. In this case the chromosome is encoded binary. Each parameter is encoded by the corresponding number of genes, what is shown in table 2. First five genes corresponding with buffer size before workstations and next are corresponding to the resource capacity at workstations.

Tab. 2. Chromosome encoding

Buffer 1	Buffer 2	Buffer 3	Buffer 4	Buffer 5	Cap1	Cap2	Cap3	Cap4	Cap5

Step 1 Initialization

The first step in the action of GA is generating, in a random way, a new population of chromosomes. Generate of new population occurs only once when the simulation is starting. The population select randomly *k* individuals with random genes. The value of each gene – 0 or 1 occurs with probability ½.

Step 2 Fitness function evaluation

A discrete-event simulator is used to evaluate the fitness function of each chromosome, which is calculated as a mean value obtained from set of replications. In our case the value of fitness function is maximized. The fitness function is given by equations 1 and 2.

$$CF = \sum_{i=1}^{n}[(A_i - B_i) \cdot C_i] - D - E \tag{1}$$

$$B_i = \sum_{i=1}^{n}\sum_{j=1}^{k}(W_{ij} \cdot K_j) \tag{2}$$

where: i – type of product (1,…n),
k – number of workstation (1,…k),
CF – fitness (criterion) function,
Ai – vector of selling prices of i-th product,
Bi – vector of unit costs of i-th product,
Ci – vector of quantities of i-th product,
D – idle cost,
E – cost of Work-In-Progress,
Kj – vector of costs for j-th workstation,
Wij – matrix of production time posts for i-th product at j-th workstation (table 1).

Step 3 Stop condition

The GA stops after constant number of generations. In our case – 100 generations. Until stopping condition is not fulfilled, the algorithm moves to the next step.

Step 4 Selection

GAs are inspired by nature, and therefore are more likely to draw individuals in order to have a better fitness function value. The selection process involves the selection of chromosomes to be parents of the offspring in the next population. In this paper the algorithm used a two different selection strategy, roulette wheel and tournament selection. The roulette wheel selection (proportional selection) operator is developed by Holland [12] and is used in many GA studies. The principle of this type of selection is to assign the probability of selecting to the each individual in each population. Chromosomes with better value of fitness function have better genetic material so they should go into the next population. The probability of drawing is formed by the following scheme:

$$\Pr_j = \frac{CF_j}{\sum_{j=1}^{k} CF_j} \tag{3}$$

where: Pr_j – probability of drawing j-th chromosome,
j – index of a chromosome in the population (1,..,k),
CF_j – fitness function value for j-th chromosomes.

Second type of selection is tournament. First step is draw a sets of chromosomes from current population. In each set the tournament between individuals is executed. Individual with the best value of fitness function win the tournament. This chromosome with best genetic material goes to the next population. Additionally the algorithm applies an elitist strategy, which means that the chromosomes with the best fitness function value go to the next population. This avoids losing the best genetic material.

Step 5 Crossover

After selection, a crossover operation is carried out. This procedure generates offsprings from the selected parents. The offspring are a combination of both parents. In this paper we are used two types of this genetic operator. First is one point crossover. It creates an offspring by copying the genes from 1 to *cp* from first parent, where *cp* is a random crossover point, and genes from *cp* to last gene form second parent. The second offspring is formed from the remaining genes. One point crossover is shown by pseudocode:

1. Select random point of crossover *cp{1,…,n-1}*
2. for $i = 1$ to *cp* do
3. *offspring1(i) = parent1(i)*
4. *offspiring2(i) = parent2(i)*
5. end do
6. for $i = cp+1$ to *n* do
7. *offspring1(i) = parent2(i)*
8. *offspring2(i) = parent1(i)*
9. end do

where *n* is the length of chromosome

Second type of crossover is multivariate crossover. In this type the chromosome is divided into vectors of parameters and crossing these parameters depending on the random factor. This is a pseudo code for multivariate crossover type:

1. Each parent chromosome is divided into a p_{ij} chains and offspring chains are of_{ij}, *q* is a number of parameters contained in the parent chromosome, *i* is the number of individuals and *j* is the number of parameter.
2. for $j=1$ to *q* do
3. if $Rnd \leq cp$ then
4. for $i = 1$ to *cp* do
5. $of_{1j}(i) = p_{1j}(i)$
6. $of_{2j}(i) = p_{2j}(i)$
7. end do
8. for $i = cp+1$ to *n* do

```
      of1j(i) = p2j(i)
      of2j(i) = p1j(i)
    end do
  Else
    for i=1 to pij length
      of1j(i) = p1j(i)
      of2j(i) = p2j(i)
    end do
  end if
end do
```

Step 6 Mutation

After selection and crossover operations all obtained offsprings may be subjected to mutation operation. For set probability one gene (from chromosome) is selected for mutation, then its value is reversed to the contrary. After mutation offsprings and champion chromosome created a new population and the algorithm returns to step 2.

4. RESULTS ANALYSIS

Simulation study was carried out with a presented earlier typical model of manufacturing system and different combinations of genetic operations: selection parameter and crossover in GA. Number of individuals in the population is 30, and the amount of replication of the population is 100 (amount is result of large simulation time). The probability of mutation is 0,035. Here are the four analyzed combinations of the GA:

1 – proportional selection, one-point crossover
2 – proportional selection, multivariate crossover
3 – tournament selection, one-point crossover
4 – tournament selection, multivariate crossover

Progress of GA will be presented for each combination. Figures from 1 to 4 showed the average criterion function (CF) for each generation and obtained the best value of CF. In each case repetiton with the best average value of CF gave the best result for the maximum value of CF. For each combination of selection and crossover was carried out three independent repetitions. In one repetition was performed 3000 simulation runs (30 individuals x 100 generation) with parameter values dependent on the chromosome. In figures 1 to 4 bottom line on the graphs represents average values of CF for entire population in 3 independent runs of GA. Top line indicates maximum value of CF from these runs.

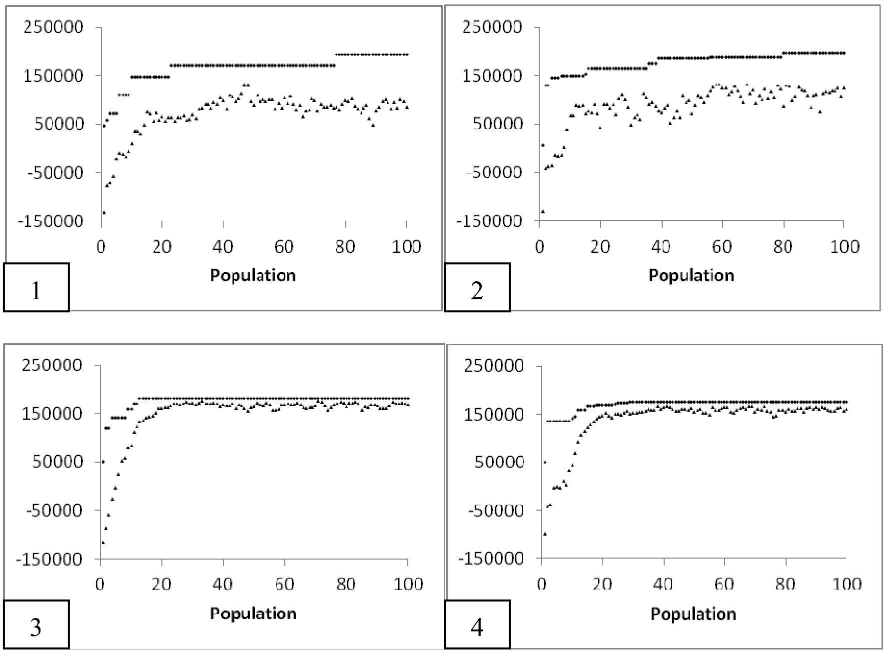

Fig. 1 – 4. The average and maximum value of CF for each algorithm combination [source: own study]

The results obtained for combinations 1 and 2 (Figures 1 and 2) showed the diversity of individuals in the population, as the average value for the entire population is significantly different from the maximum result obtained in all repetitions performed for these settings. The average value of CF, does not stabilize with each successive generation, which again shows the wide variety of individuals in successive generations. Wide variety of average values in successive generations gives a chance to not get stuck in local extremes. In the initial 20 generations it can be seen the considerable increase in the average value of CF. The large difference between the average and maximum value of CF leads to the conclusion that the best individuals was not often preferred as the parent. All three replications gave the average results which oscillating in the same bounds, which means similar adaptation of each population. Average values of CF obtained for the combinations 3 and 4 (figures 3 and 4) are close to the maximum values of CF obtained for these combinations. This proves a good adaptation of the population. Average values of CF significantly increased in the first 20 generations, which indicates a continuous improvement in the adaptation of the population. The average values close to the maximum result means frequent choice the best individual as a parent.

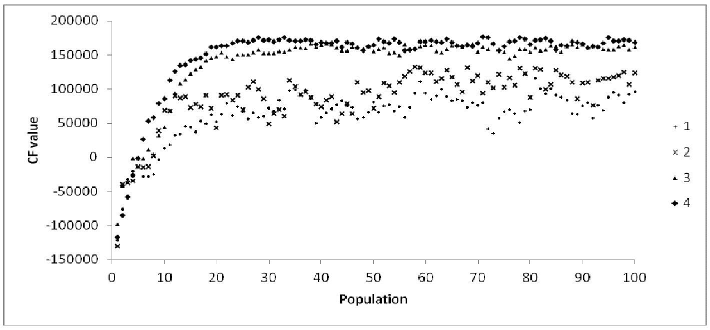

Fig. 5. The best average value of CF for each combination (symbols from 1 to 4 indicate best average value of CF from each combination) [source: own study]

Figure 5 is a comparison of the average value of the best average value of CF for each combination. Combination 1 gave the worst average result. It is very unstable and indicates the wide variety of individuals. Combination 2 gave a slightly better result, the average values are higher. Combination 3 gave a much improvement in terms of the average value of CF, is significantly higher than combinations 1 and 2, and slightly worse than the combination 4. The results showed that the average population adaptation depends on the method of selection and type of crossover. If we want to obtain a population where most individuals are well adopted and the average value of CF is close to the maximum value it should be used tournament selection method, which explicitly promotes individuals with better adaptation than the proportional selection, which gives worse average value of CF, but provides a greater variety of individuals with much different adaptation. When it comes to crossover, the average values indicate the minimum advantage multivariate crossover. If we assume that much greater impact on the average value of CF has the selection method, the results for combination 2 and 4 are better than the results for combination 1 and 3.

Figures form 6 to 9 show the value of CF for all individuals in successive generations of the GA populations for combination 1 to 4. For combination 1 and 2 populations contain a very diverse individuals as can be seen by a large range of values which CF reaches across generations. By way of elitism in any population there is an individual with the maximum CF value achieved in the previous populations, so that there is a chance of crossing the best individuals.

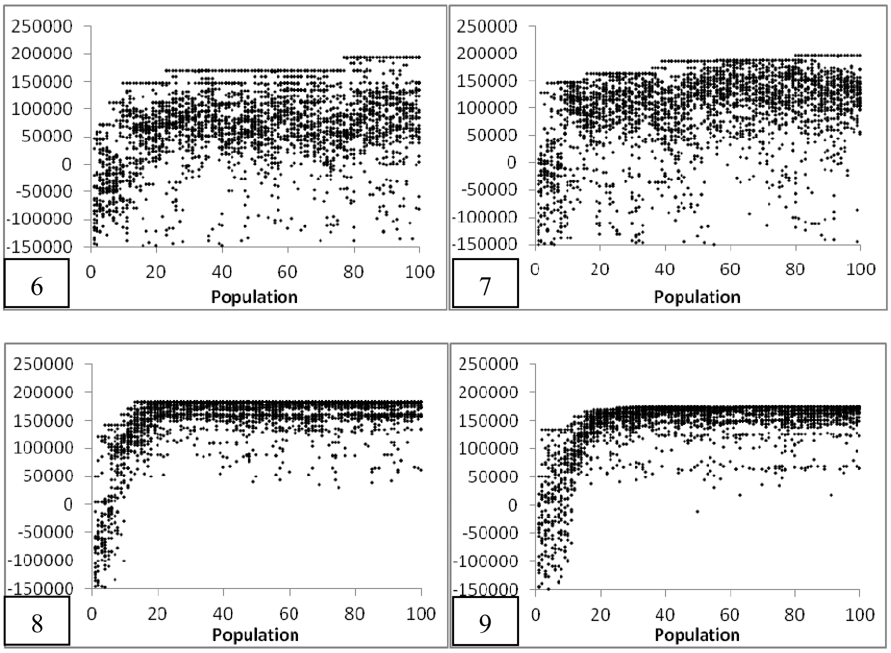

Fig. 6 – 9. CF for individuals in the one population for each combination [source: own study]

Although in the combination 2 with the multivariate crossing it can be seen that more individuals approach the maximum result of that particular repetition. In figures 8 and 9 a large concentration of CF values for each population can be seen. Most of the CF values are close to the maximum value. The concentration of results in a fairly narrow range already takes place in the first 20 generations, which may mean that chances of getting a better result for the next population is decreasing.

From the observation of all individuals in the population it can be concluded that proportional selection gives more diverse populations, and the tournament selection results in faster convergence of CF. Crossover operator does not have much influence on the result, one can assume that in this example a random factor has a strong influence on the maximum result obtained depending on the type of crossover.

5. SUMMARY

In the article, the optimization of typical manufacturing system using computer simulation was executed. For optimization genetic algorithm with different variants of genetic operators was used. Executed analysis shown that suitable buffer

allocation and selection of resource capacity increase manufacturing system performance which gnerates higher profits. Various combinations of GA give different results. Proportional selection provides greater diversity in the given population. With tournament selection, the better value of whole population adaptation can be achieved. The crossover does not cause apparent effect, but it can be seen that the multivariate crossover operator will reduce the range in which there is the majority of CF values.

The analysis of the problem showed that it is worthwhile to conduct research towards the optimization of manufacturing systems by computer simulation and GAs. From the diversity of the obtained results one can observed how significantly a single parameter affects the profits received from the manufacturing system.

REFERENCES

[1] GOLDBERG D.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, 1989.

[2] SCOTT L.R., HARMONOSKY C.M.: *An improved simulated annealing simulation optimization method for discrete parameter stochastic systems.* Computers & Operations Research, 32, 2005, p. 343–358.

[3] BANKS J., CARSON J.S., NELSON B.L., NICOL D.M.: *Discrete-Event System Simulation*. Prentice Hall Inc., New Jersey, 2001.

[4] PAUL R.J., CHANEV T.S.: *Simulation optimisation using a genetic algorithm.* Simulation Practice and Theory 6, 1998, p. 601–611.

[5] AZADEH A., TARVERDIAN S.: *Integration of genetic algorithm, computer simulation and design of experiments for forecasting electrical energy consumption*. Energy Policy, 35, 2007, p. 5229–5241.

[6] ZHANG H., GEN M.: *Multistage-based genetic algorithm for flexible job shop scheduling problem.* Complex Int., 11, 2005, p. 223–232.

[7] ENTRIKEN R., VÖSSNER S.: *Genetic Algorithms with cluster Analysis for Production Simulation.* Proceedings of the 1997 Winter Simulation Conference, p. 1307 – 1314.

[8] CAKAR T., YILDIRIM M.B.: *A neuro-genetic approach to designand planning of a manufacturing cell.* Journal of Intelligent Manufacturing, 16, 2005, p. 453–462.

[9] CAN B., BEHAM A., HEAVEY C.: *A Comparative study of Genetic Algorithm Components in Simulation-Based Optimisation*. Proceedings of the 2008 Winter Simulation Conference, p. 1829-1837.

[10] ESKANDARI H., RABELO L., MOLLAGHASEMI M.: *Multiobjective Simulation Optimization using an Enhanced Genetic Algorithm.* Proceedings of the 2005 Winter Simulation Conference, p. 833-841.

[11] KONSTAM, A.H., HARTLEY, S.J., CARR, W.L.: *Optimiaztion in a Distributed Proccesing Environment using Genetic Algorithm with Mutivariate Crossover*. Proceeding of the 1992 ACM annual conference on Communications, p. 109-116.

[12] HOLLAND J.K.: *Adaptation in Neural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI., 1975.