International Journal of Industrial Engineering and Management (IJIEM), Vol. 7 No 3, 2016, pp. 129-134
Available online at www.iim.ftn.uns.ac.rs/ijiem_journal.php
ISSN 2217-2661

UDK: 51-7 : 519.8

Weaknesses of Genetic Algorithms with Precedence Preservative Crossover and Mutation in Disassembly Sequence Optimization

Juraj Šebo
Faculty of Mechanical Engineering, Technical University of Košice, Letná 9, Košice, Slovak Republic,
juraj.sebo@tuke.sk
Miriam Šebová
Faculty of Economics, Technical University of Košice, Letná 9, Košice, Slovak Republic

Received (14.09.2015.); Revised (18.01.2016.); Accepted (22.03.2016)

Abstract

The paper is focused on genetic algorithms(GA) with precedence preservative crossover and their practical use in disassembly sequence optimization. It tries to find out if a genetic algorithm is a viable means in disassembly sequence optimization and particularly to identify what its weaknesses and limits are. In the first part of the article the well-known genetic GA methodology is described. The second part identifiesthe practical problems and limitsof the methodology in experiments withmobile phone disassembly optimization. The paper identifies the need for the simplification of product structure diagrams in some cases as well as the need to set the rules for the selection of genes for mutation. Furthermore, based on the experiments, itfavours the idea of regular incorporation of a mutation operator in GA, i.e. mutation in every generation. From the experiments,it also seems that GA operators in the methodology function in the way that they prevent higher uniformity of individuals in a few subsequent generations, andthat it is not significantly better on average to leave the algorithm run longer (e.g. until the 40th generation) in comparison to the original "stop" conditions of the methodology.

Key words: *genetic algorithm, optimization, precedence relationships, disassembly, mobile (cell)phone*

1. INTRODUCTION

In everyday life, we are often faced with complex problems for which solutionshave to be found. These solutions should be optimal in some sense, or as close to optimum as possible. Most problems can be called optimization problems whose solution can be found within a set of potential solutions. For more complex problems it is necessary to develop methods, which take chance into account. The behaviour of these methods is quite chaotic and they solve specified problems in another way every time. These methods include, for example, simulated annealing and genetic algorithms (GA)[1].

A genetic algorithmisa search algorithm based on the mechanism of natural selection and genetic principles.

The evolutionary principles applied in nature are the conceptual model for a genetic algorithm. There are populations of individual species which are composed of individuals with different characteristics. These characteristics are initially encoded in the genes that form larger units called chromosomes. New individuals are formed by a crossover and usually have a random part of the genes from the first and second parent. In particularly exceptional cases, there may bean accidental change of any gene in a chromosomecalled mutation. This could be favourable or unfavourable to the further development of the species. Depending on itscharacteristics, each of the offspring has agreater or smaller ability to withstand natural selection and create the next generation. The selection process is repeated over and over again in the course of improving the genetic characteristics of the species. Thus,evolution in nature continues to run [2].

The optimization algorithms like GA search the optimal solution in a space of solutions. The search in a space of solutions usually requires balancing two contradictory requirements [3]:

- thefocus of searching in promising areas,
-the most complete search of the entire space of solutions.

By setting the parameters of a GA, it is possible to adjust the balance between the focus on promising areas and searching through the major parts of the space of solutions.Another important feature of a GA is that it works with the entire population of potential solutions while other methods always works with only one point of aspace of solutions, with one solution.In

order to solve a specific problem using a GA it is necessary to meet the following requirements:
- To find a suitable representation of potential solutions to the problem (appropriately select chromosome "format");
- To find a way to create an initial population of chromosomes that represent acceptable solutions;
- To establish a fitness (evaluation) function, which enables the algorithm to decide which individual is "better" and which is "worse";
- To select or create appropriategenetic operators that affect the formation of new offspring;
- To appropriately set various parameters used in the GA (population size, probability of applying genetic operators, etc.)[3].

Based on the presented issues, reviewed articles and own experiments, the paper will try to answer the question as to whether a genetic algorithmwith precedence preservative crossover is a viable meansin disassembly sequence optimization.

2. DESCRIPTION OF GENETIC ALGORITHM METHODOLOGY FOR DISASSEMBLY SEQUENCE OPTIMIZATION

In the relevant literature, the disassembly planning issue usually includes the disassembly sequence optimization (see [4][5][6][7] [8][9][10][11][12][13][14][15][18] [19]). Some articles are focused on non-standard optimization techniquessuch as genetic algorithms (e.g. [6][16]).

The following part is focused on a description of the genetic algorithm methodology with the precedence preservative crossover based on thearticle by Kongar and Gupta (see [6]). It describes this method through the steps of its application in the disassembly sequence optimization of a mobile phone [17]:

1. Disassembly ofa mobile phone. During and after the disassembly,write down the disassembly directions, methods of removal and demand for the components. Creation ofa product structure diagram and identification of precedence relationships of components.
2. Creation of the original population of 10“individuals”that all individuals respect precedence relationships and are different.
3. Calculation of the fitnessof the current generation (in the first round the current generation is equal to the original generation).
4. Choose the five individuals with the highest fitnessvalue,clonethem and receive 10 “individuals” (10 parents).
5. Application of the genetic operators (crossover and mutation) on the current population and thus creationof a new generation (10 children).
6. Testingif the "stop" conditions (serial number of generation and the “ratio” indicator) are met. If yes.then stop the generation of the new generations.
7. The “individual” with the highest fitness value across all generations is considered as the optimal disassembly sequence.

3. IDENTIFICATION OF PROBLEMS AND LIMITS OF THE GA METHODOLOGY THROUGH THE PRACTICAL APPLICATION ON DISASSEMBLY SEQUENCE OPTIMIZATION

Based on experiments with multiple applicationsof the describedmethodology on the disassembly sequence optimization of mobile phones, the following problems and limits have been found.

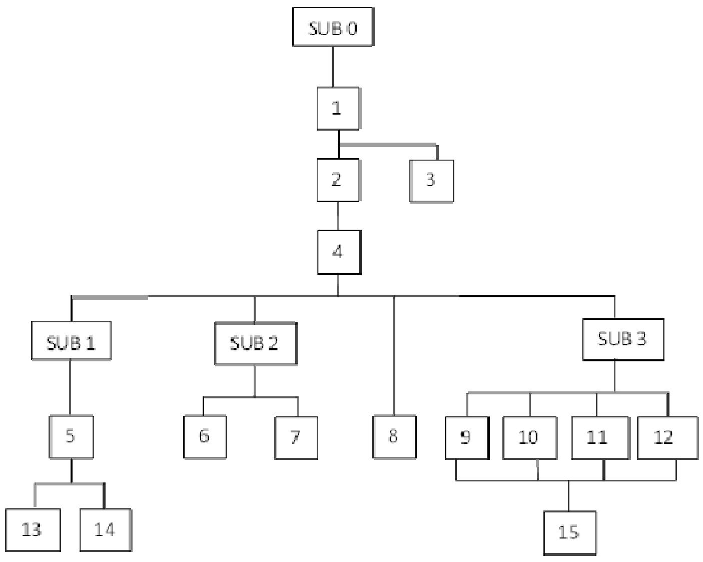

Figure 1. Product structure of Siemens C60

The first problem was identified in step 1 of the methodology. In construction of the product, a structure diagram is not possible to design the situation for the next disassembly step (resp. reaching a lower level on the diagram). There is a need to disassemble the subassembly, not individual components. In other words, in order to reach a lower level in the diagram, there is a need to disassemble a particular component, not subassembly. If the real product had such a structure, from which there would be the need to disassemble a subassembly, one possibility would be to simplify/modify the product structure diagram in a way, that the subassembly would be considered as a component. In the experiment withthe disassembly of the mobile phone Siemens C60, it is meantto simplify the real product structure (Figure 1). In reality,component 2 is composed of two components (front cover and rubber keyboard) so it is a subassembly.

The second problem arises in step 5 of the methodology.The application of mutation in a GA with precedence preservative crossover is a bit complicated in the GA application, if there is a need to preserve precedence conditions also after mutation. One reason isthat it is not possible to mutate that gene (component) in any way,just change to another gene (component). It requires using only mutation, which is switching two genes (components) position, because each individual represents a sequence of components and every component canonly be once in the sequence. It implies that before mutation can be runthe algorithmhas to know if genes (components) can be mutated and if so,

which genes (components) they could be interchanged with. It was found in individual experiments that some components could not be mutated, because if they were, precedence conditionswould not be maintained. In respect of this, the question ishow to identify genes (components) which could be switched without breaking the precedence conditions?
The authors have found that one “gene identification rule” exists, which could be incorporated before the mutation operator. The rule is thatthe genes (components), which are at the same vertical level of the product structure diagram and are under the same node of product structure diagram could switch (a node is equal to a component in the product structure diagram). On the basis of this rule it was possible to createtwo groups of switchable genes for the Siemens C60 mobile phone (group 1 (5,6,7,8,9,10,11,12) and group 2 (13,14)), which were used in the experiments with the GA methodology.
The methodology also requires a random setting of the number of mutated genes (components). Based on the fact there are,in the case of the Siemens C60,10 switchable genes (5 possible pairs of genes) the random number generator was set so that theminimum of mutated genes was equal to 1 and the maximum of mutated genes equal to 5.
After these settings,it was possible to incorporate the mutation operator in the following way. Before mutation in each generation,a number of mutated genes (components) (resp. pairs of genes (components)) was set from 1 to 5 and then the mutation operator was applied on the first 4 individuals (children) (theapplication of the mutation on the first 4 individuals is proposed in original Kongar and Gupta methodology (see [6]).) This number of mutated individuals is fixed for all generations[17].
The third problem is connected to step 6 of the methodology. The maximum and average fitness in its development from the first to the last generation does not improve continuously and does not reach the same maximum (resp. global maximum) in alloptimization attempts (Figure 4 and 5, Table 1 and 2).(The optimization attempts mean10 independent executions of the GA algorithm on an identical original population of disassembly sequences of the Siemens C60 mobile phone). Although this is the natural characteristic of the genetic algorithm, the intention was to find a way of how to improve the outcomes of the methodology – the maximum fitness value reached.
Previously, the author’s research had focused on the mutation operator. In order to find out how the mutation operator affects the GA methodology outcomes, experiments/calculations with a changed method of incorporation of the operator in the algorithm were executed. In the experiments three modifications (scenarios) of the GA were set and traced.The first scenario (crossover and mutation in every second generation), second scenario (crossover and mutation in first five generations) and third scenario (only crossover (no mutation)) (see [17]). It was found, that these described modifications of mutation had rather a positive influence on the development of average fitness but a rather negative influence onthe reached maximum fitness. Because the final effectiveness of the GA methodology is dependent on the maximum fitness reached, it was concluded that the findings favour the idea of the suitability of a regular incorporation of a mutation operator in the GA[17].
Secondly, the original methodology was tested as to how it would behave in multiple optimization attempts. The results are describedin Tables 1 and 2 as well as in Figures 2 and 3.

Table 1. Maximum fitness values in multiple optimization

Number of attempt	Maximum fitness value	Number of generations in which the maximum fitness was reached
1.	1645	2.
2.	1965	4.
3.	2145	2.
4.	1945	6.
5.	1965	15.
6.	1825	28.
7.	1495	1.
8.	1705	4.
9.	1945	2.
10.	1765	4.
Average	1840	6,80

In Table 1,it can be seenthat the methodology is ableto improve the original solution (1495 = maximum fitness in the original population) in most of the independent attempts, but the final maximum fitness reached is not globally optimum. It is also visible that there is quite a big difference between the overall maximum reached in the whole 10 attempts (2145), to average the maximum of independent attempts (1840).
In Table 2,the variability of the methodology output in 10 executed attempts is analysed. It is possible to see that the average improvement of maximum fitness in 10 executed attempts is about 23%. The biggest improvement was achieved in the 3rdattemptand was more than 43%. The lowest improvement is in 7thattempt and is equal to 0, what means no improvement. Half of the attempts achieved an improvement in the range of 22-32 %.

Table 2. Difference of the maximum fitness value reached in individual attempts to the maximum fitness value of original population (1495)

Number of trial	Maximum fitness value	Absolute difference*	Relative difference* [%]
1.	1645	150	10,03
2.	1965	470	31,44
3.	2145	650	43,48
4.	1945	450	30,10
5.	1965	470	31,44
6.	1825	330	22,07
7.	1495	0	0,00
8.	1705	210	14,05
9.	1945	450	30,10
10.	1765	270	18,06
Average	1840	345	23,08

In Figure 2 (development of maximum fitness in multiple attempts), itis possible to see that in some parts of the chart the maximum fitness values are quite stable. These parts of stability of the maximum fitness bring us to idea to check, if there are not too many uniform individuals, which could influence the ability of the methodology to find new individuals with higher fitness. There was an intention to find out if there is an accumulation of identical individuals in these parts of the chart (these parts represent generations with the same maximum fitness). In other words,to find out how many identical (unvaried) individuals with the same maximum fitness value (resp. how many identical individuals in general) are in these generations. For this analysisthe last (10th) attempt was chosen.

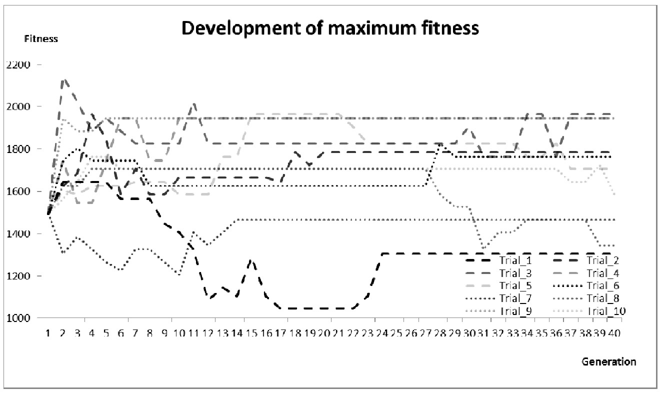

Figure 2. Development of maximum fitness in multiple optimization attempts

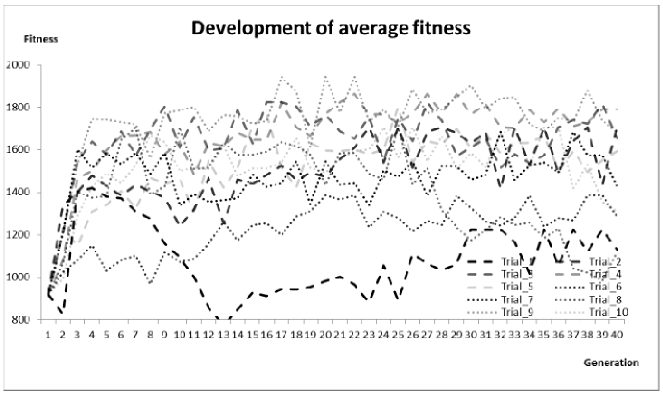

Figure 3. Development of average fitness in multiple optimization attempts

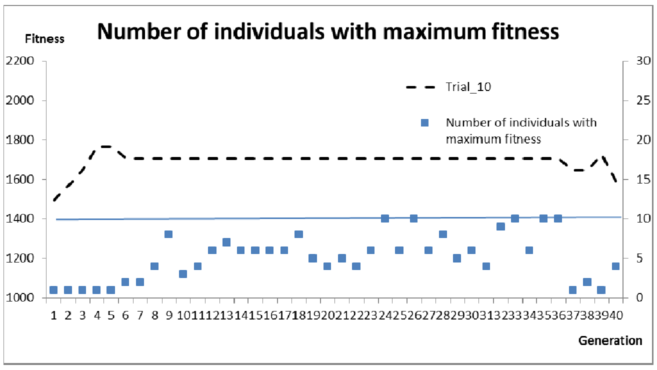

Figure 4 Stability of maximum fitness and number of individuals with maximum fitness in the10th attempt

From Figure 4 it is possible to see the chart of maximum fitness development in the 10th attempt with quite a large part (range of generations), in which the maximum fitness is stable. It seems from the graph that there is some relationship between the number of individuals with maximum fitness in a particulargeneration and stable development of maximum fitness. The problem is that there could be different individuals with equal fitness, so there is a need to analyse if individuals are identical and not only their fitness. Thequestion addressed was how many of the individuals with maximum fitness are identical or quasi-identical.

In a more precise look at the individuals with maximum fitness in each of the 40 generations (Figure 5) it can be seen that there is no visible relation between the development of the number of identical (or quasi-identical) individuals with maximum fitness and stable character of maximum fitness in the chart area from the 6th to 36th generation. From this it can be assumed that GA operators in the methodology function in a way that they prevent uniformity of individuals for more then a few subsequent generations.

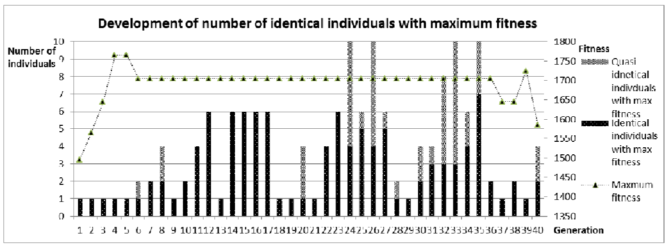

Figure 5. Development of number of identical and quasi-identical individuals with maximum fitness in the10thattempt

Thirdly,an analysisof the outcomes of the GA methodology under different “stop” conditionswas carried out. Different “stop” conditionsmean theoriginal “stop” conditions (stop when the first of the “stop” conditions is met) and modified “stop” conditions (leave algorithm run until it reaches the predefined number of generations (in this case 40))(Original “stop” conditions mean the“stop” conditions in the Kongar and Gupta methodology (the “ratio” indicator and number of generations). The “ratio” indicator is the ratio of the average fitness value of the new generation to the average fitness value of the old generation. If the “ratio” indicator is smaller or equal to 1,0005, the algorithm should stop creating new generations [6].)

Table 3 shows that in the original “stop” conditions, the algorithm usually stops after the first few generations (on average at the 5th generation). It also shows that in most of the attempts (8 from 10), the reached maximum fitness value is the same for both variants of the “stop” conditions. Only in 2 attempts from 10 a better maximum fitness value was reached in the modified “stop” conditions. The relative improvement of the maximum fitness in particular attempts was 22,43 % (attempt 5) and 1,11 % (attempt 6) and on average 2,11 % for all attempts.

The conclusion is that in executed optimization attempts it was not significantly better,on average, to leave the algorithm run under modified “stop” conditions (i.e. until the 40th generation) in comparison to the original “stop” conditions.(Figure 6 shows, for all attempts,a development of the “ratio” indicator until the algorithm stops.)

Table 3. The maximum fitness values under original “stop” conditions in comparison with modified “stop” conditions

	Original “stop” conditions			Modified “stop” conditions (run until 40th generation)	
Number of trial	Maximum fitness value	Number of generation in which was reached maximum fitness	Number of the last generat-ion	Maximum fitness value	Number of generation in which was reached maximum fitness
1.	1645	2.	2.	1645	2.
2.	1965	4.	5.	1965	4.
3.	2145	2.	5.	2145	2.
4.	1945	6.	9.	1945	6.
5.	1605	2.	3.	1965	15.
6.	1805	3.	4.	1825	28.
7.	1495	1.	5.	1495	1.
8.	1705	4.	4.	1705	4.
9.	1945	2.	5.	1945	2.
10.	1765	4.	6.	1765	4.
Average	1802	3,00	4,80	1840	6,80

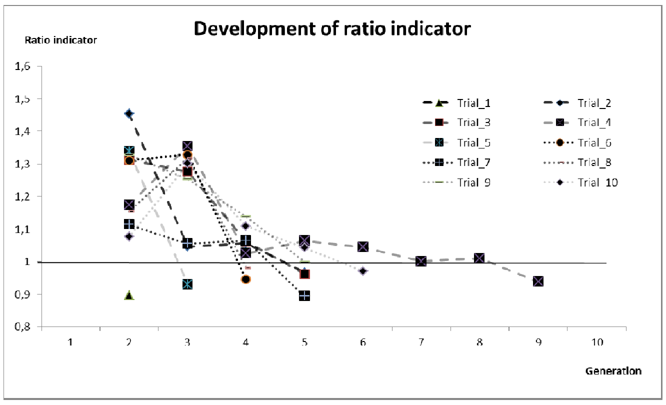

Figure 6. Development of ratio indicator in multiple optimization attempts

(The whole analysis was made with the unchanged (identical) original population).

4. CONCLUSION

From the three highlighted problems of the described GA methodology, the first problem can be considered as methodology-relevantand problemstwo and three as GA-relevant. The first problem is that the product structure diagram used in the methodology is not able in some cases to exactly represent the real structures of subassemblies and components in aproduct. In orderto be able to use the methodology for some products, product structure diagram needs to be simplified in a way thatproblematic subassembly is taken asa component. The second problem isthat the system of selecting group(s) of switchable genes is not set. So as to be able to use the methodology,there is a need for some selection system of the group(s) of switchable genes (components), e.g. based on the assumption, thatit is possible to switch the genes (components), which are at the same vertical level and under the same node of product structure diagram. The third problem, connected with the nature of GA, is that the fitness function does not reach same maximum (resp. global maximum) in every attempt. In searching for betteroutcomes of the methodology,few experiments with modifications in the mutation operator, with multiple optimization attempts and with the modification of „stop“ conditions were made.

The experiment with the changed mutation frequency throughoutthe generations has shown that decreasingthe mutation frequency has a rather negative influence on the reached maximum fitness. As such, it is possible to favour the idea of regular incorporation of a mutation operator in the GA, i.e. mutation in every generation.

From the analysis of the number of identical individuals with maximum fitness in the chart with stable maximum fitness it can beassumed that the GA operators in the methodology function in a way that prevent higher uniformity of individuals in a few subsequent generations, so they are not a visible cause of the stable character of maximum fitness in some attempts.

The results of the optimization attempts havealso shown that it is not significantly better,on average, to leave the algorithm to run longer (e.g. until the 40th generation) in comparison to running it until theoriginal “stop” conditions of the methodologyare met.

ACKNOWLEDGEMENT

This article was supported by the state grant agency KEGA research grant 029TUKE-4/2016 “Educational and training workplace of innovation development and implementation of business processes and systems”.

5. REFERENCES

[1] Pošík, P. (2000). Genetic Algorithms. Retrieved 2015, from http://labe.felk.cvut.cz/~posik/pga/theory/ga-theory.htm

[2] Teda, J., &Lehocký, Z. (2005). Genetic algorithms and their application in practice.Retrieved 2015, from http://programujte.com/clanek/2005072601-geneticke-algoritmy-a-jejich-aplikace-v-praxi/.

[3] Kvasnička, V. (1998). Genetic algorithm.Retrieved 2015, from http://www2.fiit.stuba.sk/~kvasnicka/NeuralNetworks/5.prednaska/GA_background.pdf.

[4] Huang, H.-H., Wang, M.H. and Johnson, M.R. (2000) 'Disassembly Sequence Generation UsingaNeural Network Approach', Journal of Manufacturing Systems, vol. 19, no. 2.

[5] Moore, K.E., Gungor, A. and Gupta, S.M. (1998) 'Disassembly process planning using Petri nets.', Proceedings of ISEE International Symposium on Electronics and the Environment , 88-93..

[6] Kongar, E., & Gupta, S. M. (2001). Genetic Algorithm for Disassembly Process Planning. Proceedings of the SPIE International Conference on Environmentally Conscious Manufacturing II, (pp. 54-62). Newton (Massachusetts).

[7] Dong, J. and Arndt, G. (2003) 'A review of current research on disassembly sequence generation and computer aided design for disassembly', J. Engineering Manufacture , vol. 217, no. Part B, pp. 299-312..

[8] Santochi, M., Dini, G. and Failli, F. (2002) 'Computer Aided Disassembly Planning: State of the Art and Perspectives,', CIRP Annals - Manufacturing Technology, vol. 51, no. 2, pp. 507-529.

[9] Tang, Y., Zhou, M.C., Zussman, E. and Caudill, R. (2002) 'Disassembly Modeling, Planning and Application, ', Journal of Manufacturing Systems, vol. 21, no. 3...

[10] Lambert, A.J.D. (1999) 'Linear programming in disassembly/clustering sequence generation', Computer and Industrial Engineering , no. 36, pp. 723-738.

[11] Lee, H.B., Cho, N.W. and Hong, Y.S. (2010) 'A hierarchical end-of-life decision model for determining the economic levels of remanufacturing and disassembly under environmental regulations', Journal of Cleaner production, no. 18, pp. 1276-1283.

[12] Youssif, M.M., Alkadeem, R.A. and El Dardiry, M.A. (2011) 'Incorporating ergonomic factors in disassembly sequence planning ', Alexandria Engineering Journal , vol. 50, no. 3, pp. 213–217.

[13] Tian, G., Liu, Y., Tian, Q. and Chu, J. (2012) 'Evaluation model and algorithm of product disassembly process with stochastic feature', Clean Techn Environ Policy , no. 14, pp. 345–356.

[14] Wiendahl, H.-P., Seliger, G., Perlewitz, H. and Bürkner, S. (1999) 'A general approach to disassembly planning and control', Production Planning & Control, vol. 10, no. 8, pp. 718 –726.

[15] Cosič, I. and Lazarevič, M. (2012) Technologijedemontažeproizvoda, Novi Sad: Fakultettehničkihnauka.

[16] Shimizu,Y., Tsuji,K., and Nomura,M.(2007) 'Optimal disassembly sequence generation using a genetic programming'International Journal of Production Research, vol. 45, no. 18-19, p. 4537–4554.

[17] Šebo, J., Szabóová, V. and Kováč, J. (2015) 'Testing the genetic algorithm suitability for disassembly sequence optimization in a case of recycling of obsolete mobile phones' International Journal of Industrial Engineering and Management, vol. 6, no. 3, pp. 93-99.

[18] Vukelic, D., Ostojic, G., Stankovski, S., Lazarevic, M., Tadic, B., Hodolic, J. and Simeunovic, N. (2011) 'Machining fixture assembly/disassembly in RFID environment ', Assembly Automation, vol. 31, no. 1, pp. 62-68.

[19] Debevec, M., Simic, M. and Herakovic, N. (2014) 'Virtual factory as an advanced approach for production process optimization', International Journal of Simulation Modelling, vol. 13, no. 1, pp. 66-78.

Nedostaci genetskih algoritama sa prvenstvom konzervacije ukrštanja i mutacija prilikom optimizacije sekvenci u demontaži

Juraj Šebo, Miriam Šebová

Primljen (14.09.2015.); Recenziran (18.01.2016.); Prihvaćen (22.03.2016.)

Rezime

*Rad se fokusira na genetske algoritme (GA) sa prvenstvom konzervacije ukrštanja i njihovu praktičnu primenu prilikom optimizacije sekvenci u procesu demontaže. Cilj rada je određivanje saznanja da li je genetski algoritam održivo sredstvo optimizacije sekvenci u procesu demontaže, kao i da identifikuje šta su njegove slabosti i ograničenja. U prvom delu članka poznata GA metodologija je koncizno opisana. Drugi deo identifikuje praktične probleme i ograničenja metodologije u eksperimentima optimizacije demontaže mobilnog telefona. Pored toga, ovaj rad identifikuje potrebu za pojednostavljenjem dijagrama strukture proizvoda u nekim slučajevima, kao i potrebu za postavkom pravila prilikom izbora gena mutacije. Nadalje, na osnovu eksperimenata, favorizuje se ideja redovne inkorporacije operatora mutacija u GA, tj. mutacija u svakoj generaciji. Takođe, na osnovu eksperimenata, utvrđeno je da GA operateri u metodološkoj funkciji da se spreči veća uniformnost pojedinaca u nekoliko narednih generacija, nije značajno bolje u proseku da bi algoritam trajao duže (npr do 40. generacija) u poređenju sa originalnim "stop" uslovima metodologije***.**

Ključnereči: genetski algoritam, optimizacija, prednost odnosa, demontaža, mobilni (mobilni) telefon