International Journal of Industrial Engineering and Management (IJIEM), Vol. 6 No 3, 2015, pp. 93-99
Available online at www.iim.ftn.uns.ac.rs/ijiem_journal.php
ISSN 2217-2661

**UDK: 621-771
568.567**

Testing the Genetic Algorithm Suitability for Disassembly Sequence Optimization in a Case of Recycling of Obsolete Mobile Phones

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

Veronika Szabóová
Čaňa, Slovak Republic

Juraj Kováč
Faculty of Mechanical Engineering, Technical University of Košice, Letná 9, Košice, Slovak Republic

Received (26.02.2015); Revised (16.07.2015); Accepted (01.09.2015)

Abstract

There are a number of studies of product disassembly sequence optimization based on different methods (linear programming, genetic algorithms, petri nets, neural networks, case-based reasoning, etc.). The article is focused on disassembly sequence optimization based on genetic algorithms. We test the suitability of genetic algorithm (GA) on disassembly sequence optimization of mobile phones . Through modification of "stop" rules and the mutation operator we analyze alternative outcomes of the GA. We found some limitations of regular "stop" rules and some difficulties in practical application of the GA. Better results were reached under reduced "stop" rules only to a number of generations (in our case 40). Difficulties in practical application are related to time demanding adjustment of the mutation operator, because of needed specification of switchable genes, and to limitations of product structure representation. On the basis of our findings, we do not consider GA as viable mean of optimization in conditions where there is a need to optimize disassembly of a large number of different products in a short time period.

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

1. INTRODUCTION

The research in disassembly has been intensively developing in the last two decades. There are a number of articles, which study the disassembly of products from different perspectives. From the articles we selected the following group of issues that are often researched in context of product disassembly:

- Design for disassembly [1] [2] [3] [4] [5],
- Disassembly planning (including economic optimization) [6] [7] [8] [9] [2] [10] [11] [12] [13] [14] [15],
- Projecting of disassembly system [16] [2] [10] [17] [18] [19],
- Technological aspects of disassembly [20] [21] [22] [23] [24],
- Reverse logistic [25] [26] [27] [28] [29],
- Uncertainty [16] [2] [12].

From these articles two core statements of respected authors deserve mentioning, as they are describing the position of disassembly in current research and practice. "The practice of industrial recycling and remanufacturing is growing in importance due to increasing environmental and economic pressures. Industrial recycling and remanufacturing involves product disassembly…"[10]. "The conclusions have not been consistent in both academic and industrial circles whether systematic product disassembly as an end-of-life (EoL) paradigm is a viable means to achieve a closed loop economy" [1].

There are different types of Waste Electric and Electronic Equipment (WEEE). Mobile phones are the equipment without a long history, and therefore recycling/reuse strategies are only just developing. We identify these following factors that are making mobile phones a „hot" issue in product reuse/recycling:

- A dramatic increase of production of mobile devices (last years more then 1,2 billion/year),
- Low awareness of people for recycling of mobile phones (according to one survey 74 % of people do not think about recycling mobile phones, 50 % of people is unaware that it is possible to recycle mobile phones, and only 3 % of people recycle their mobile phones)
- The short lifetime of mobile devices (estimated lifetime of usage around 1-2 years, technical lifetime around 10 years),

- Extended producer responsibility (currently incorporated in EU legislation),
- Valuable materials inside mobile phone (gold, silver, copper, etc.),
- Environmental trends in manufacturing industry (e.g. Environmentally Conscious Manufacturing, Closed Loop Supply Chain).

If we search the mobile phone research areas, we can find in time frame 2000-2010 following issues (prepared on the base of [30]):

- Life-cycle assessments of mobile phones and the entire mobile phone network (including e.g.: energy and cost assessments of the mobile phone life-cycle, in particular, their end-of-use management,),
- Basic assessments of material composition and energy requirements,
- Product remanufacturing (with focus on: production and operations management issues) - industrial engineering and management view,
- Economic and/or environmental assessments as addition to industrial engineering research,
- Environmental assessment (including e.g.: environmental footprint analysis, leaching assessments of mobile phones, assessment of releases due to mobile phone incineration),
- Manufacturers' guidelines on environmentally sound cell phone end-of-use management.

Based on issues presented, literature, additional information and thoughts, we will try to answer the question: if disassembly is a viable means in reuse/recycling of mobile phones. In this article we are focusing on pros and cons of genetic algorithm use in disassembly sequence optimization.

2. PRACTICAL APPLICATION OF GENETIC ALGORITHM (GA) ON DISASSEMBLY SEQUENCE OPTIMIZATION OF MOBILE PHONE

In studied literature the disassembly planning issue usually includes the disassembly sequence optimization. There are different algorithms presented for optimization of disassembly sequence. A number of articles are related to linear programing use in disassembly optimization (e.g. [11] [12]). Some articles are focused on other non-standard optimization techniques as genetic algorithms (e.g. [8]), petri nets (e.g. [7] [31]) and neural networks (e.g. [6]). There are other approaches to disassembly planning such as e.g. case-based reasoning, as well ([32]).

We have done a few tests of disassembly sequence optimization with different optimization methods (e.g. linear programing, heuristic methods) in the past. There are some optimization methods that are relatively easy to use (e.g. [12] [33]) and gives reasonable results for one specific mobile phone. In the following we focus on practical application of genetic algorithm with the maintaining precedence relationships. Methodology is based on an article of Kongar and Gupta ([8]). This method of genetic algorithm we apply on disassembly sequence optimization of mobile phone Siemens C60.

In the practical application of GA we have followed the methodology in these steps:

1. To determine the number and type of parts we have experimentally disassembled the mobile phone to the components. After disassembly of the components we created a product structure diagram, from which we identified disassembly order precedence, relationships and conditions. During and after the disassembly we wrote down disassembly directions (+/- x, y, z), identified methods of removal of the components (D-destructive, N-non-destructive) and also identified whether or not is demand for the components (D-demand, N- no-demand) in terms of the material from which they are made.
2. We created an original population of 10 "individuals"("individual" is in our sense equal to chromosome composed from 15 genes) so that each "individual" had a different order of "genes"(gene in our sense is part of chromosome, which could take the values 1, 2, 3, … ,15, that represent specific component of mobile phone) ,i.e. that no two "individuals" in the original population were the same, and respect order preference conditions.
3. We calculated the fitness (Fitness function in our case is depending on demand for component, directional changes in disassembly and disassembly method changes. Higher fitness of "individual" means better disassembly sequence it represents) of the current generation. (Remark. In the first round the current population = original population.)
4. We chose the five individuals with the highest level of fitness from the current population which we clone and thus receive 10 "individuals" (10 parents).
5. We applied the genetic operators (crossover and mutation) on current population and thus we created a new generation (10 children).
6. We test the "stop" conditions (serial number of generation and the "ratio" indicator). When we met one of the stop conditions we discontinue the process of generating new generations. (Remark. Here, we have made a change in the original approach, we do not respect the "ratio" indicator, i.e. we leave algorithm to run until the 40th generation is met).
7. The "individual" (i.e. specific order disassembly sequence) of components) with the highest achieved value of fitness across all generations represents the optimal disassembly sequence.

Step 1:

In the beginning, we did experimental disassembly of the mobile phone Siemens C60 and on the basis of this we created a product structure (Figure 1) where we marked particular components by numbers from 1 to 15. (Remark: In the product structure there is implemented one simplification. After releasing the screws (component number 4) we receive the 3 subassemblies and 1 component. We do not assume the disassembly order of these subassemblies and components.)

During and after experimental disassembly we have identified disassembly directions (+/- x, y, z), methods of removal of the components (D-destructive, N-non-destructive) and demand for the components (D-demand, N- no-demand). The identified data are presented in the Table 1.

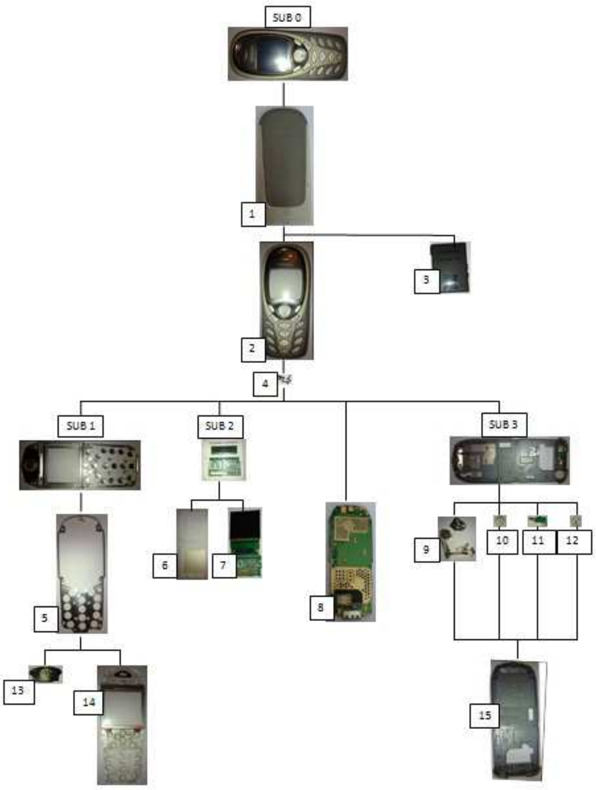

Figure 1. Product structure of Siemens C60

Table 1. Identified demand, disassembly directions and disassembly methods of Siemens C60 components

Component num.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
Demand for the component	D	D	D	N	N	N	N	D	N	N	N	N	N	N	D
Disassembly direction of the component	-x	+y	-y	+y	+y	-y	-y	-y	+y	+y	+y	+x	-y	+y	+y
Methods of removal of the component	N	N	N	N	N	N	N	N	N	N	D	N	N	N	N

Step 2 and 3:

In these steps we have created an original population of 10 individuals and calculate the fitness (Table 2).

Table 2. Original population and its fitness values

"Gene" position / „Individuals"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Fitness
Individual 1	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15	915
Individual 2	1	2	3	4	5	6	7	13	14	11	10	12	9	8	15	675
Individual 3	1	2	4	5	13	14	8	6	7	12	11	10	9	3	15	535
Individual 4	1	3	2	4	9	10	11	12	15	5	13	14	6	7	8	1175
Individual 5	1	3	2	4	8	6	7	10	9	12	11	15	5	13	14	1495
Individual 6	1	2	4	8	5	6	7	13	14	12	11	10	9	3	15	615
Individual 7	1	2	4	11	10	12	9	8	6	7	5	3	13	14	15	695
Individual 8	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15	1195
Individual 9	1	2	3	4	12	11	10	9	5	13	14	8	6	7	15	795
Individual 10	1	3	2	4	8	5	13	14	6	7	9	12	11	10	15	1115

Step 4:

In this step we have chosen (from Table 2) five individuals with the highest fitness, which we clone and thus receive 10 "individuals" (10 parents) (Table 3).

Table 3. Adjusted initial generation composed of five individuals with the highest fitness from the original generation and their clones

"Gene" position / „Individuals"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Individual 1	1	3	2	4	8	6	7	10	9	12	11	15	5	13	14
Individual 2	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Individual 3	1	3	2	4	9	10	11	12	15	5	13	14	6	7	8
Individual 4	1	3	2	4	8	5	13	14	6	7	9	12	11	10	15
Individual 5	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15
Individual 6	1	3	2	4	8	6	7	10	9	12	11	15	5	13	14
Individual 7	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Individual 8	1	3	2	4	9	10	11	12	15	5	13	14	6	7	8
Individual 9	1	3	2	4	8	5	13	14	6	7	9	12	11	10	15
Individual 10	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15

Step 5:

In step 5 we applied crossover and mutation on the "individuals". First we applied crossover. We chose 5 pairs of parents randomly, and through two randomly generated "masks" (Remark. These "masks" ensure maintaining of precedence relationships.) we create two children for each pair of parents (Table 4). This way we created a new generation (Table 5) on which we can further apply mutations.

Table 4. Crossover application scheme (Example of crossover of parents "Individual 2" and "Individual 7")

Parent "individual 2"	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Parent "individual 7"	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Crossover "masks"															
Mask for child 9	2	2	2	2	2	1	1	2	1	1	1	1	1	1	2
Mask for child 10	2	2	1	1	1	1	1	2	1	2	2	2	1	2	1
Children from crossover															
Child 9	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Child 10	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15

Table5. New generation received after crossover

"Gene" position / „Individuals"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Child 1	1	3	2	4	8	5	9	13	14	10	6	11	12	15	7
Child 2	1	3	2	4	9	8	10	11	12	15	5	13	14	6	7
Child 3	1	3	2	4	9	8	10	5	13	14	6	7	12	11	15
Child 4	1	3	2	4	9	10	11	8	12	15	5	13	14	6	7
Child 5	1	3	2	4	6	7	8	5	13	14	12	9	11	10	15
Child 6	1	3	2	4	6	7	8	5	13	14	12	11	9	10	15
Child 7	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15
Child 8	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15
Child 9	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Child 10	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15

On this new generation (Table 5) we applied the mutation operator. This was a most complicated part of application of GA. In this case, we could only use mutation, which is switching two genes (components) position, because of the character of our chromosome/"individual" (sequence of components). We cannot just exchange one gene (component) by another gene (component), because we have to include all genes (components) in each chromosome/"individual". Before we can run a mutation we have to know if genes (components) could be mutated and if yes, which components they could be interchanged with. We did found that some of components could not be mutated, because if we mutate them we will not maintain precedence relationships. Considering this idea, we can switch genes without breaking the rules of precedence relationships if the genes are at the same level of product structure. We created 2 groups of switchable genes (Table 6). In addition, we randomly set a number of mutated genes (or pairs of genes) for each generation. Based on the fact that we have 10 switchable genes (means 5 possible pairs of genes) we set in random number generator minimum of mutated genes equal to 1 and maximum of mutated genes equal to 5. This way we have created new generations after mutation (Table 7)

Table 6. Groups of switchable genes (components)

Group of components 1	5	6	7	8	9	10	11	12
Group of components 2	13	14						

Table 7. The new generation received after crossover and mutation (random number of mutated genes = 3 in this example)

"Gene" position / „Individuals"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Child 1	1	3	2	4	6	7	9	14	13	10	8	11	12	15	5
Child 2	1	3	2	4	9	12	10	11	8	15	5	14	13	7	6
Child 3	1	3	2	4	9	8	7	5	14	13	6	10	11	12	15
Child 4	1	3	2	4	9	10	8	11	12	15	5	14	13	7	6
Child 5	1	3	2	4	6	7	8	5	13	14	12	9	11	10	15
Child 6	1	3	2	4	6	7	8	5	13	14	12	11	9	10	15
Child 7	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15
Child 8	1	2	3	4	5	13	14	6	7	8	9	10	11	12	15
Child 9	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15
Child 10	1	3	2	4	6	7	8	5	13	14	12	11	10	9	15

Step 6:
After mutation we have continued with (resp. have gone back to) the calculation of the fitness (Step 3), until the number of generation is equal to 40. We use this modification of "stop" conditions of genetic algorithm methodology presented by Kongar and Gupta ([8]), because we would like to trace behavior of the "individual´s" fitness and the "ratio" indicator. (The "ratio" indicator is together with number of generation considered as "stop" condition for GA. 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, GA should stop.)
Step 7:
The "individual" with the highest achieved value of fitness across all generations is individual 9 in 20th generation (Table 8). The optimal disassembly sequence is presented in Table 9.

Table 8. The "individual" with the highest achieved fitness value

"Gene" position / „Individuals"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Fitness
Individual 9	1	3	2	4	9	8	12	10	11	15	5	14	6	7	13	1755

Table 9. The optimal disassembly sequence of mobile phone Siemens C60

1.	Component 1	(back cover)
2.	Component 3	(battery)
3.	Component 2	(front cover)
4.	Component 4	(screws)
5.	Component 9	(antenna)
6.	Component 8	(motherboard)
7.	Component 12	(SIM card holder)
8.	Component 10	(microphone)
9.	Component 11	(vibration engine)
10.	Component 15	(inner plastic part)
11.	Component 5	(metal frame)
12.	Component 14	(plastic frame)
13.	Component 6	(LCD frame)
14.	Component 7	(LCD)
15.	Component 13	(speaker)

3. ANALYSIS OF MODIFICATION OF "STOP" RULES AND MUTATION OPERATOR IN THE PRACTICAL APPLICATION OF GA

The next graphs (Figure 2 and 3) present the development of maximum and average fitness and the development of the "ratio" indicator. On the graphs we can see trends, which are in case of average fitness upward and in the case of maximum fitness very little downward oriented and its values oscillate between values (1400 and 1600) except 20th generation (1755). Also we can see that with the development of new generations the trend line of the "ratio" indicator is little downward oriented and the indicator oscillate between values (0,8 and 1,2) except 2nd generation (1,275).
Based on the presented graphs, we can analyze the outcomes of GA under different "stop" rules. If we stop GA (resp. creation of new generations) according to the general "stop" rule (stop when first "stop" condition is met) we would in our case stop the algorithm in the 6th generation (the "ratio" indicator is equal to 1 it means smaller than 1,0005) and receive the maximum fitness value of all generations equal to 1595. If we do not stop the algorithm according to the general "stop" rule and leave it to run further (in our case to the 40th generation) we will receive the maximum fitness value of all generations equal to 1755 (reached in the 20th generation).
From this practical application of GA we can conclude, that in our optimization case it is better to leave algorithm run further, even if the "ratio" indicator is smaller or equal to 1,0005.

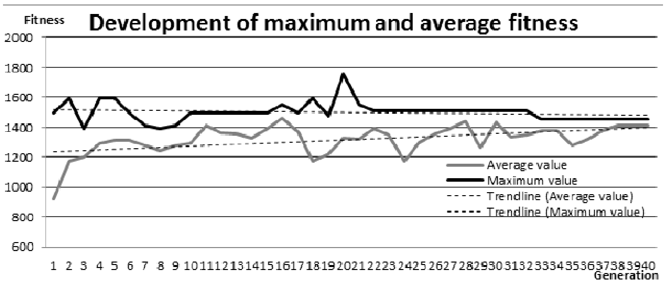

Figure 2. Development of the fitness values (Base scenario: crossover and mutation)

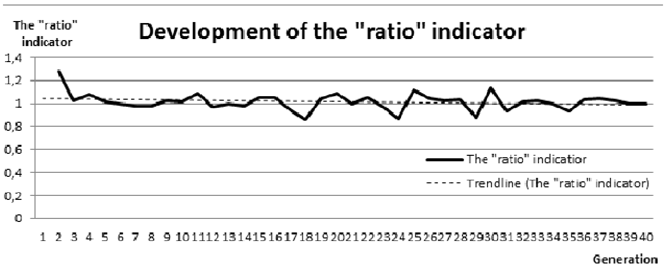

Figure 3. Development of the "ratio" indicator (Base scenario: crossover and mutation)

On the way to find out how the mutation operator affects the GA outcomes, we have changed the method of incorporating this operator in the algorithm. Based on this, we have set and traced three additional modifications (scenario) of GA:

- First scenario: crossover and mutation in every second generation
- Second scenario: crossover and mutation in first five generations
- Third scenario: crossover (no mutation)

The results of the base scenario in comparison to alternative scenarios (with modified mutation operator) are presented on the following graphs (Figure 4, 5 and 6). As we can see from graph (Figure 4) the highest average fitness value is reached in the first scenario, although with quite regular oscillations of values between 1400 and 1700.

The average fitness value of the second and third scenario has rather large increases in the beginning and no further oscillation, but a stabilization at certain values (1635 for 2nd and 1555 for 3rd scenario). The basic scenario has most of the average values below alternative scenarios values, and similarly to the 1st scenario it has quite regular oscillation mostly between 1200 and 1450.

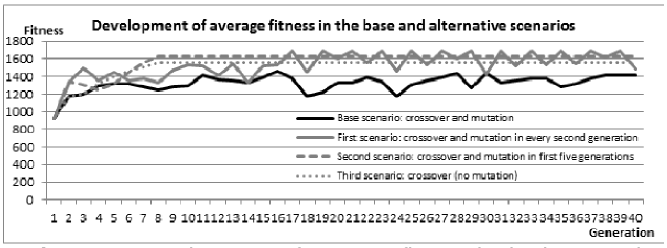

Figure 4. Development of average fitness in the base and alternative scenarios

In the next graph (Figure 5) we can see, that the highest maximum fitness value (1755) is reached by base scenario, while most of the base scenario values oscillate between 1400 and 1600. The maximum fitness value of first, second and third scenario has different values in the first generations and from 5th (resp. 6th) generation they have stable values (1695 for 1st, 1635 for 2nd and 1555 for 3rd scenario).

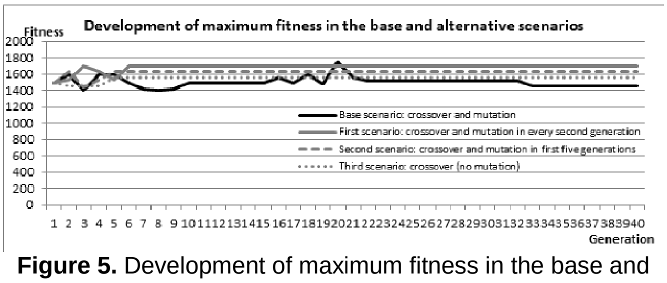

Figure 5. Development of maximum fitness in the base and alternative scenarios

In the last graph (Figure 6), describing development of the “ratio” indicator in different scenarios, we can see, that there is, in first few generation in each scenario a dramatic decrease of the indicator, but then in second and third scenario it is stabilized on the value of 1 (because of equal average fitness values in the subsequent generations) and in base and first scenario it is quite regularly oscillating between 0,8 – 1,2.

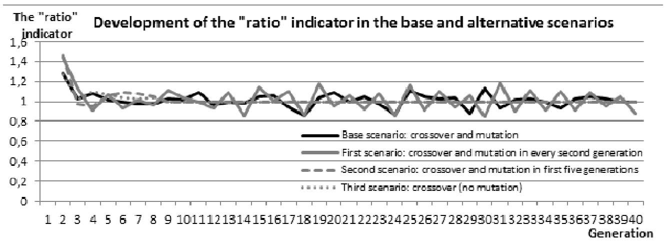

Figure 6. Development of the “ratio” indicator in the base and alternative scenarios

4. CONCLUSION

We discovered that maximum fitness value (1755) was reached in base scenario (crossover and mutation), second highest (1695) in first scenario (crossover and mutation in every second generation), third highest (1635) in second scenario (crossover and mutation in first five generations and the smallest value (1555) in third scenario (crossover (with no mutation)). These findings support the idea of the suitability of regular incorporation of mutation operator in the GA.

We further discovered, that the base scenario has reached the highest fitness value (so the best disassembly sequence) without regular “stop” conditions settings (based on Kongar and Gupta methodology) but with modification of “stop” conditions in a way, that we do not discontinue when first “stop” condition is met. We continue until “stop” condition based on number of generation is met (in our case 40).

During the application of the above described methodology of GA we face some difficulties. Firstly, the application of GA includes time demanding adjustment of the mutation operator, because of needed specification of switchable genes (in our case two groups of switchable genes). Secondly, there are some problems with this product’s structure representation, which has its limitations (e.g. there are more options where the battery could be placed in the product structure, which will have impact on GA).

Based on presented graphs and findings from application of GA with maintaining of precedence relationships on the mobile phone disassembly sequence optimization we can conclude, that the genetic algorithm (based on Kongar and Gupta methodology) gives reasonable outcomes, but could be improved (e.g. by modification of “stop” conditions). Since the algorithm needs quite demanding adjustment to the specific product, we doubt its viability in conditions where there is a need to optimize disassembly of high number of different products in short time period (e.g. in a recycling/disassembly center). Further research opportunity in this issue we see in tracing influence of pre-set values in fitness function on GA outcomes.

ACKNOWLEDGEMENT

The paper is the result of the project implementation: University Science Park TECHNICOM for Innovation Applications Supported by Knowledge Technology, ITMS: 26220220182, supported by the Research & Development Operational Programme funded by the ERDF.

5. REFERENCES

[1] Duflou, J.R., Seliger, G., Kara, S., Umeda, Y., Ometto, A. and Willems, B. (2008) 'Efficiency and feasibility of product disassembly: A case-based study', CIRP Annals - Manufacturing Technology, no. 57.

[2] 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.

[3] Muránsky, J., Badida, M. and Hricová, B. (2011) Ekodizajn v strojárstve, Košice: Elfa.

[4] Huanga, C.-C., Liangb, W.-Y., Chuangc, H.-F. and Changa, Z.-Y. (2012) 'A novel approach to product modularity and product disassembly with the consideration of 3R-abilities', Computers & Industrial Engineering, , vol. 62, no. 1, pp. 96–107..

[5] De Napoli, L., Rizzuti, S. and Rocco, C. (2012) 'Comparison of environmental characteristics of functionally equivalent devices by weighted product method', Proceedings of International Design Conference, DESIGN 2012; Dubrovnik, Croatia, 777-786.

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

[7] 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..

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

[9] 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.

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Kopacek, B. and Kopacek, P. (1999) 'Intelligent disassembly of electronic equipment', Annual Reviews in Control, no. 23, pp. 165-170.

[17] Kováč, J. and Mareš, A. (2009) Projektovania demontážnych systémov na báze využívania videoanalýzy a cad modelov, SjF, TUKE

[18] Cosič, I. and Lazarevič, M. (2012) Technologije demontaže proizvoda, Novi Sad: Fakultet tehničkih nauka.

[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.

[20] Basdere, B. and Seliger, G. (2003) 'Disassembly Factories for Electrical and Electronic Products To Recover Resources in Product and Material Cycles', Environmental science & technology, no. 37, pp. 5354-5362.

[21] Seliger, G., Skerlos, S.J., Basdere, B. and Zettl, M. (2003) 'Design of a Modular HousinPlatform to accommodate the remanufacturing of multiple cellular telephone models', 3rd International Symposium on Environmentally Conscious Design and Inverse Manufacturing, EcoDesign '03, 243 – 250.

[22] Bailey-Van Kuren, M. (2006) 'Flexible robotic demanufacturing using real time tool path generation', Robotics and Computer-Integrated Manufacturing , no. 22 , pp. 17–24.

[23] 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.

[24] Svetlík, J. (2014) 'Connecting interface for construction of modular structures', 13th International Conference on Industrial, Service and Humanoid Robotics, ROBTEP 2014; , Strbske Pleso, Slovakia.

[25] Škapa, R. (2005) Reverzní logistika, Brno: Masarykova univerzita v Brne, Ekonomicko-správní fakulta.

[26] Shih, L.-H. (2001) 'Reverse logistics system planning for recycling electrical appliances and computers in Taiwan Resources', Conservation and Recycling , no. 32, pp. 55–72.

[27] Assavapokee, T. and Wongthatsanekorn, W. (2012) 'Reverse production system infrastructure design for electronic products in the state of Texas', Computers & Industrial Engineering , no. 62 , pp. 129–140.

[28] Sibe, A.A., Nickel, S., Saldanha-da-Gama, F. and Verter, V. (2012) 'Multi-period reverse logistics network design', European Journal of Operational Research, no. 220, pp. 67-78.

[29] Dat, L.Q., Linh, D.T.T., Chou, S.-Y. and Yu, V.F. (2012) 'Optimizing reverse logistic costs for recycling end-of-life electrical and electronic products', Expert Systems with Applications , no. 39, pp. 6380–6387.

[30] Gayer, R. and Doctori Blass, V. (2010) ' The economics of cell phone reuse and recycling', International Journal of Advanced Manufacturing Technologies , no. 47, pp. 515-525.

[31] Zussman, E., C, Z.M. and Caudil, I.R. (1998) 'Disassembly Petri Net Approach to Modeling and Planning Disassembly Processes of Electronic Products', Proceedings of the 1998 IEEE International Symposium on Electronics and the Environment, ISEE – 1998, , New York.

[32] Veerakamolma, I.P. and Gupta, S.M. (2002) 'A case-based reasoning approach for automating disassembly process planning', Journal of Intelligent Manufacturing, vol. 13, no. 1, pp. 47-60.

[33] Meacham, A., Uzsoy, R. and Venkatadri, U. (1999) 'Optimal Disassembly Configurations for Single and Multiple Products, ', Journal of Manufacturing Systems, vol. 18, no. 5, pp. 311-322.

Testiranje pogodnosti generičkog algoritma za optimizaciju redosleda demontaže u slučaju recikliranja zastarelih mobilnih telefona

Juraj Šebo, Veronika Szabóová, Juraj Kováč

Primljen (26.02.2015); Recenziran (16.07.2015); Prihvaćen (01.09.2015)

Rezime

Postoje studije o optimizaciji redosleda demontaže koje se temelje na različitim metodama (linearno programiranje, generički algoritmi, petri mreže, neuronske mreže, studije slučaja, itd.). Ovaj rad se fokusira na optimizaciju redosleda demontaže zasnovanu na generičkim algoritmima. Testirali smo pogodnost generičkog algoritma (GA) na optimizaciju redosleda demontaže mobilnih telefona. Putem modifikacije „stop“ pravila i operatera mutacije, analizirali smo alternativne rezultate GA. Pronašli smo neka ograničenja redovnih „stop“ pravila i neke poteškoće u praktičnoj primeni GA. Bolji rezultati su postignuti sa redukovanim „stop“ pravilima samo u određenom broju generacija (u našem slučaju 40). Teškoće u praktičnoj primeni vezane su za vreme, zahtevajući regulaciju operatera mutacije zbog potrebnih specifikacija promenljivih gena, kao i za ograničenja prezentovanja strukture proizvoda. Na osnovu naših pronalazaka, ne smatramo da je GA održivo sredstvo optimizacije u uslovima u kojima postoji potreba da se optimizira demontaža velikog broja različitih proizvoda u kratkom vremenskom periodu.

Ključne reči: *demontaža, optimizacija, generički algoritam, mobilni telefon*