DESIGN TASKS SCHEDULING USING GENETIC ALGORITHMS

Tamás RICK* Márk HORVÁTH** and Tibor BERCSEY*

*Institute of Machine Design
Faculty of Mechanical Engineering
Budapest University of Technology and Economics
H–1521 Budapest, Hungary
e-mail: rick.tamas@gszi.bme.hu
**Faculty of Sciences
Vrije Universität Amsterdam
De Boelelaan 1081a
NL-1081 HV Amsterdam
e-mail: cyber@inf.elte.hu

Received: March 21, 2006

Abstract

This paper deals with a modelling and optimization method that is capable of the product structure based optimization of design processes. The processes were modelled with Design Structure Matrix. Guided stochastic search techniques were applied when finding the optimal task schedule. The impact of the probability of mutation and crossover and the two different selection processes on searching was examined. Further investigations were completed in order to multi-object optimize the time and cost of design processes.

Keywords: product development process, genetic algorithm (GA), design structure matrix (DSM), task scheduling, learning rate.

1. Introduction

The product and this way the production process determines the economic prosperity of a company. Constructional design plays an important role in the manufacturing process. Not only the quality requirements but also the necessary time and costs criteria are considered when the efficiency of the process is determined [4]. The base of planning constructional design processes is that the tasks necessary for the design of a given product are determined exactly and in detail. This requirement has not been fulfilled yet, since the models used nowadays, mentioned later, allow only rough design task description. Most often only sequential task order is applied, and the tasks that can be completed simultaneously are neglected. For this reason, the costs and time can only be estimated roughly. A further problem is that they do not allow iterations (modifications, redesign), which occur often during design, due to their sequential structure.

2. Modelling

Several methods, procedures and models have been developed in the fields of operations research, organization, management, information technology and engineering sciences for manufacturing and innovation processes from the strategic design of the product to the introduction to market.

Simple relation models, methods such as the ERM (Entity Relationship Model) are used to describe the structure of processes, while models applied in network planning are Petri nets, PERT (Program Evaluation und Review Technique), CPM (Critical Path Method) and MPM (Metra Potential Method).

The development process is often illustrated with the Nassi – Schneidemann, PAP and Gantt diagram, which is wide spread in the scheduling tasks in project management.

The Popp type generalized decision net supports the decision steps of production and models the development process with activities, decision points and stochastic nodes as well as forecasts the progress.

Hierarchic relation models, graphical description methods like the SADT are applied in activity and data modelling. Up-to-date information technological or artificial intelligence methods such as neural networks are also more and more frequently used.

The first generation model of product development that only contained the classical steps was worked out at NASA in the 1960s. In Europe the outstanding representatives of the German theoretical design trend, i.e. R. KOLLER, K. H. ROTH, G. PAHL and W. BEITZ worked out process plans of uniform structure for the purpose of development and constructional design [5]. These were the basis of VDI Richtlinie 2221 and 2222 [15], which are considered to be a transition to the second generation and are applied frequently in the industry.

The second generation models focused on activities, preserved the rigid sequential separation of phases in the development process and separated the phases with exact decision points. Several theories, such as the integrated product development proposed by Ehrenspiel, quality standard ISO 9000, axiomatic design theory (Suh) and general design theory (Tomiyama) have been worked out on the basis of these activity oriented models. This way the autogenetic design theory (Bercsey-Vajna) as well as design systems that provide primarily theoretical and methodological support, such as TRIZ/ARIZ (Altschuller) and the contradiction oriented WOIS (Linde-Hill) have evolved based on the analogy between the evolution of natural and technical systems,

The third generation phase gate models that allow the overlapping of phases and activities and the flexible transformation of the process model dependant on the task, company and risk were developed from the second generation models. Decisions in the model also depend on the optimal process of the whole development project. Two applications of these third generation models are QS9000 and VDA 4.3.

The development trend of methods inevitably show that there is an increasing need for the decomposition of processes to the smallest available details, for the consideration of the costs, resources and time of the process and for the dynamic optimization of the process according to these parameters.

Processes can be rather long and costly, so finding optimal schedules is crucial in product success over competitors. There are many possible objectives when considering a project precedence problem. These include minimizing process cost, minimizing variation of resource profiles, or minimizing project duration. In particular, minimizing project duration or makespan is of strategic significance in the stage of product planning for product development problems.

The designer usually requires an aid for effective design since it requires overall knowledge, checking and control of all processes and activities. This aid should model the series of activities, all possible parameters (e.g. time, cost) and the environment as well. If the environment changes, the model should react considering all the intervention possibilities.

Process elements, the logical relations of which are defined in rules, are applied in process modelling and description. The following techniques are available for process description [5]:

- Flowchart
- Sequence diagrams
- Multiple Activity diagram
- Process diagrams.

The most wide spread, standardized process modelling method is the SADT (Structured Analysis and Design Technique) [8], which is one of the Multiple Activity diagram techniques. SADT is a graphical method and is similar to data flow and structure diagrams, although it is more general and uniform. It has disadvantages since it is vast, difficult to handle and to modify (see *Fig. 1*) due to graphical and hierarchical mapping (only a limited number of boxes can be illustrated on the given sheet size).

The IDEF (International DEFinition language) [15] family has been formed on the basis of modelling tool SADT so that process modelling of different purposes can be carried out. There are four versions considered to be important in design processes. Their modelling tools differ to a certain extent, since the purpose of the description is also different but the principle is the same as in the SADT model.

IDEF0 is used to produce a function model which is a structured representation of the functions of a manufacturing or design system or environment and of the information and objects which interrelate to those functions.

IDEF1 is used to produce an information model which represents the structure of information needed to support the functions of a manufacturing or design system or environment.

IDEF2 is used to produce a dynamics model which represents the time varying behaviour of functions, information, and resources of a manufacturing system or environment.

An IDEF3 process description organizes the network of relations between situations in a specified scenario. IDEF3 descriptions are developed from two different

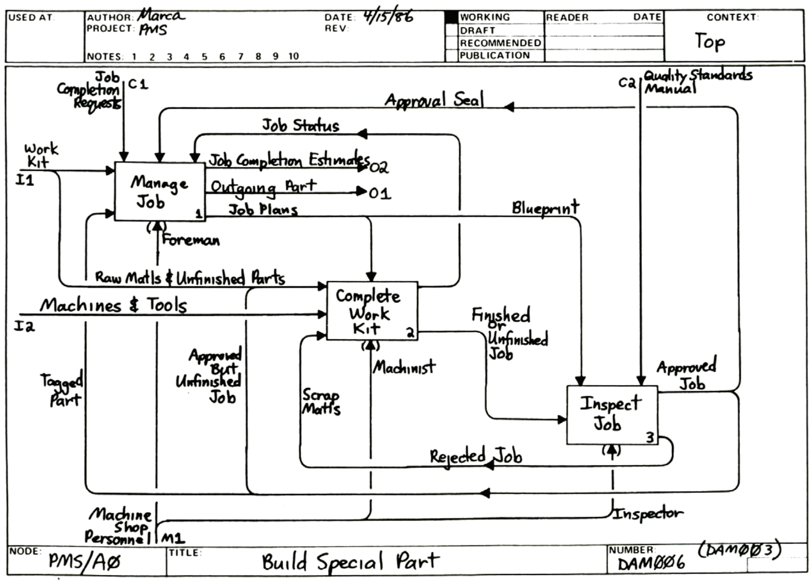

Fig. 1. An example of SADT [8]

perspectives: process-centered and object-centered. Because these approaches are not mutually exclusive, IDEF3 allows cross-referencing between them to represent complex process descriptions.

These versions can be integrated but then their usage and computerization becomes difficult. Since handling the costs, time and resources is important from the aspect of design processes, these factors should be presented in a way that makes exact description and optimization possible. This task can be solved with using the Design Structure Matrix (DSM).

The adaptation and conversion of process models do not result in data loss, and they can be transformed back to their original state exactly from DSMs (see *Figs. 2* and *3*).

3. Design Structure Matrix

STEWART [12] has used DSM for the first time to describe informational and organizational relations.

The DSM method is based on the fact that the sequence of activities can be rearranged on the basis of the relations among the design processes of the product elements. This way the whole process is easier to understand and becomes shorter (cheaper).

A matrix with the following parameters was used in the description of the relation among the structural elements of the product to be designed:

The main structural elements (S.E.) of the product A_i $(i = 1, 2, \ldots, n)$ define the matrix shown in *Fig. 3*. The elements of the diagonal represent themselves, hence a_{ij}=0 $(i = j)$. The other elements of matrix A reveal the relations between the main structural elements.

If the structure elements provide information for A_i, a_{ij}=1 otherwise a_{ij}=0 meaning that there is no connection between elements A_i and A_j. If a_{ij}=1 and $i < j$ are valid for one element in the matrix, it is supposed to be above the diagonal and refers to a feed forward relation. While if $i > j$, the element is below the diagonal and refers to feedback or to a cycle. In case of a cycle the number of supposed cycles based on the current sequence can be given (see *Fig. 3*).

If the description of the method is applied, *Fig. 2* can be transformed into the matrix revealed in *Fig. 3* and vice versa.

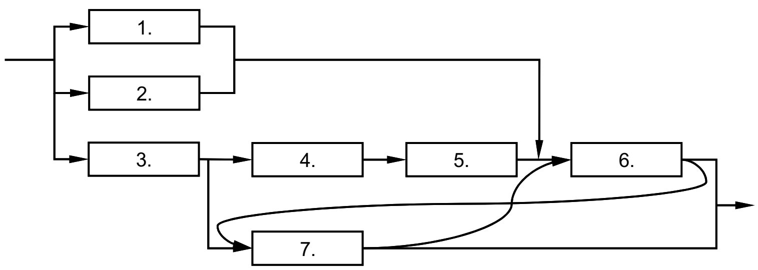

Fig. 2. Process blocks

Further information, such as the time and cost of design, can be assigned to the matrix elements. These pieces of information are shown in *Fig. 4*, in the second (time) and third (cost) columns of the matrix.

The relations plotted in DSM can be transformed into graphical form in the way revealed in *Fig. 3*.

A_i	1	2	3	4	5	6	7
1	*					1	
2		*				1	
3			*	1			1
4				*	1		
5					*	1	
6						*	1
7						1	*

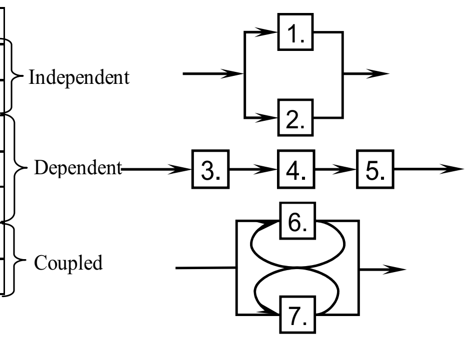

Fig. 3. Interpretation of relations

When the matrix is produced, the number and 'size' of feedbacks (containing more elements) is huge due to the precedence, hence more time is required and the costs are also higher. An example is shown in *Fig. 4*, in the 20-8 and 18-5 column-row combinations.

From the aspect of information flow it is rather disadvantageous if the cycles cross. This results in increased costs and chaotic events as well during the planning process, due to information redundancy and uncertainty. An example for this phenomenon is revealed in *Fig. 4*, where cycle 20-14 crosses cycle 17-10.

The aim is to produce a sequence of DSM elements in which the number of feedbacks and crossovers is minimal, while costs and required time is also decreased.

This task requires an optimizing algorithm that can also solve robust tasks as well and is capable of fast optimization when there are more, weighted aims. Hence genetic algorithms (GA) have been chosen for the optimization task [10]. Since the applicability of GAs depends on the type of the task, preliminary examination is necessary for the correct setting of algorithm parameters (mutation and crossover probability, selection procedures).

S.E	T	C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	30	30									1													
2	40	20	8																					
3	10	60																		1				
4	10	60														1								
5	10	60								1														
6	60	10	4		2					1												1	1	
7	40	20														1								
8	60	10		8					8		2													
9	20	40				7				2														
10	20	40								8						1								
11	40	20						8																
12	30	30											6											
13	30	30												3										1
14	20	40													8									
15	30	30								4						8								
16	20	40															8		7					
17	30	30										8						7		8				
18	40	20					6												8		8			
19	60	10																		8		2		
20	40	20								7						4					2			
21	20	40																8						
22	20	40																						

Fig. 4. DSM example

4. Genetic Algorithms

A genetic algorithm is a search method that first of all maps a wide range of allowed solutions randomly. After an evaluation it chooses the ones from the range that are most capable of living and it recombines and mutates them in a way that the results are almost optimal solutions. Genetic algorithms use the principles of biological evolution during stochastic searching and optimization [7, 9].

4.1. Coding

Genetic algorithms usually work with the parameters to be optimized in a coded form (e.g. binary or gray coding) and not directly but in case of precedence optimization this is not effective [2, 3]. In this case a gene of an 'individual' (a given sequence, a solution of the search space) consists of the numbers of structural elements (chromosome) in an uncoded way.

4.2. Selection

The efficiency of two selection methods was examined among the individuals of the initial population, which is formed randomly, after evaluation. 'Better Half' [1] selection was one of these methods, where the better half of the population is selected to operate further genetic operators. The other type is 'Tournament' [13], where the capability of living of the two individuals chosen randomly is compared and the better one makes it to the next step.

4.3. Crossover

During crossover the genetic information is interchanged between the two individuals and a new individual is formed. The algorithm carries out crossover with a so called position based crossover method [13]. This means that the algorithm chooses chromosomes randomly from one parent and these are rewritten in the child's gene in the selected places. The remaining places are filled with the other parent's elements in a way that the sequence is checked and the first chromosome not present in the child is placed in the child's first free chromosome place (see *Fig. 5*).

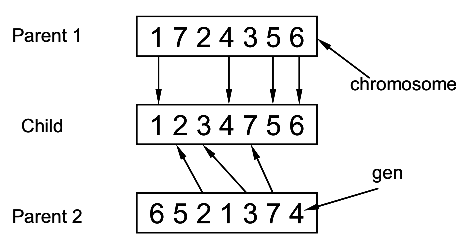

Fig. 5. Position based crossover

4.4. Mutation

During mutation the algorithm chooses two chromosomes randomly in the child produced in a crossover and swaps the values of the chromosomes. This is the order-based mutation [12].

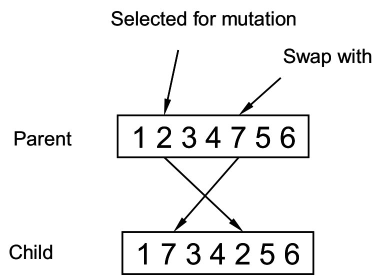

Fig. 6. Order based mutation

In case of sequence design tasks, the application of genetic operators can cause problems since the individuals created this way may contradict the dependencies set previously. This is a significant issue in the field of production technology when process plans are created. Constructional design is different, since if a given iterative task order is started at the ‘wrong’ place according to the optimal sequence, maximum one more iteration will be carried out, and this way time and costs increase, and the individual becomes worse but not yet unviable. During mutation and crossover the quantity of iterations may change since a relation below the diagonal can move above the diagonal in case of an adequate sequence, hence the task will become sequentially and simultaneously accomplishable (*Fig. 4* Task 1-Task 2; Task 2-Task 1).

4.5. Evaluation

The different evaluation methods are detailed in Chapter 5.

5. Examinations

Position-based crossover and order-based mutation were used in our examinations. The impact of mutation, crossover parameters and the two different selection processes were examined (Chapter 5.1). With the help of the experience gained a multi object optimization task was solved in order to find a cost and time optimum (Chapter 5.2).

5.1. Testing Parameter Values and Selection Processes

First of all the impact of the change in the ratio of the probability of mutation and crossover (0.1/0.9; 0.2/0.8; 0.25/0.75; 0.3/0.7; 0.4/0.6) was examined simultaneously with the efficiency of the 'Better Half' and the 'Tournament' selection in case of different matrix dimensions (10, 12, 16, 22).

Evaluation

During testing 1 time unit was uniformly assigned to the elements of the matrices. The aim of searching was to produce a sequence of structural element design the turnaround time of which is minimal.

Turnaround time was calculated on the basis of *Eq.* (1) and the cycles were also taken into consideration.

$$f(t) = T_\Sigma = \sum_{i=1}^{n} t_i \tag{1}$$

In this equation f(t) denotes the fitness function the minimum of which is to be found, T_Σ is the calculated time value, t_i is the time dedicated to one structural element, while n stands for the number of structural elements.

The value of the function increases if the number of iterations is high, since all the elements in the cycle have to be completed again, and that takes time and money. This is the reason for a search into a sequence that contains the fewest iterations of the smallest size.

Test Results

12 optimization processes were carried out during the tests. Efficiency was examined in the following way: the population number where the individual with the best fitness value appears was registered. *Table 1* involves the average values (Av.) and the standard deviation (s) of the 12 optimizations.

Table 1. Optimization results

Mut. rate/ Crosso. rate	0.1/0.9				0.2/0.8				0.25/0.75				0.3/0.7				0.4/0.6			
	Tourna-ment		Better Half		Tourna-ment		Better Half		Tourna-ment		Better Half		Tourna-ment		Better Half		Tourna-ment		Better Half	
Matrix dim.	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s	Av.	s
10	22	14	15	6	19	12	32	30	21	7,5	20	10	13	8	32	18	17	3	14	4
12	47	10	45	8	28	11	36	15	40	9	45	12	47	14	38	18	38	14	43	12
16	63	13	60	10	36	5	33	4	42	8	40	10	51	14	49	19	49	7	51	10
22	700	100	750	100	380	20	400	20	220	18	320	20	200	15	300	15	180	31	250	30

Fig. 7 illustrates the average values (shown in *Table 1*) of two selected combinations (0.2/0.8 and 0.4/0.6). The efficiency (the population value where the best

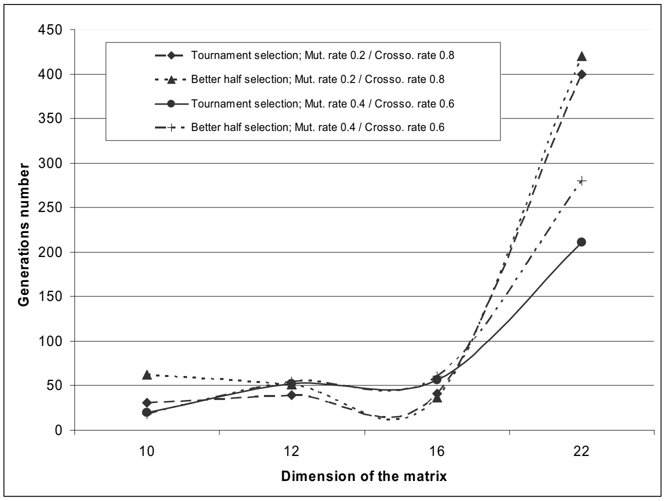

Fig. 7. Efficiency of selection processes

fitness was achieved) of the two selection procedures are graphed as a function of the matrix dimension in case of the above mentioned two mutation/crossover combinations.

The number of relations in the matrices was determined on the basis of the dimension so that the almost same filling was provided. The number of relations was set to be one and a half time the matrix dimension and $^3/_3$ of it was set to be feedback. The population size was 20 during the tests.

5.2. *Multi Object Search, Learning Rate*

The optimization of design processes requires optimization according to time and cost. Hence, the evaluation process was modified in a way that both the time and costs can be taken into consideration simultaneously. During the development the fact that the customer requirements may differ for these two aims was kept in focus. For this reason a weighing factor was introduced for both parameters. This way the cost or the time can gain different importance.

A further learning rate was introduced in the cycles so that the model approximates reality better. The application of this rate is verified because the steps that occur more times in the cycles require less and less time and expense since the a priori knowledge necessary for the solution also decreases.

The time and costs combined with the learning rate are calculated as the sum

S.E	T	C	12	11	8	3	21	18	17	18	19	20	10	16	6	13	14	8	7	9	2	1	4	22
12	30	30		1																				
11	40	20			1																			
8	60	10				1	1					1						1				1		
3	10	60								1														
21	20	40						1																
18	20	40							7					1										
17	30	30						7		8			1											
18	40	20							8		6				1									
19	60	10								6		2												
20	40	20									2						1	1						
10	20	40															1	1						
16	30	30															1	1						
6	10	60																1						
13	30	30	3																					1
14	20	40														6								
8	60	10																	1	2	1			
7	40	20															1							
9	20	40																2					1	
2	40	20																				1		
1	30	30																		1				
4	10	60															1							
22	20	40																						

Fig. 8. Optimized DSM 1

of a geometrical series (2) in our model:

$$F_{\Sigma} = \sum_{j=1}^{n} X_j \frac{1 - L_{rate}^{n}}{1 - L_{rate}} \tag{2}$$

where $F_{\sum}$ is the total time/cost, X_j is the time/cost of designing a structural element, L_{rate} is the learning rate and n is the number of cycles.

5.3. *Evaluation*

The applied fitness function f(c,t) (3) is determined with the help of the calculated time and costs and its components may have different weights. The calculation was carried out in the following way:

$$f(c, t) = w_c \cdot \sum_{k=1}^{m} F_c + w_t \cdot \sum_{i=1}^{m} F_t \tag{3}$$

where f(c,t) is the fitness function, the minimum of which is to be found, w_c/w_t is the weight of the cost/time value, F_c/F_t is the cost / time value calculated with *Eq.* (2) and *m* is the number of the structural elements.

5.4. *Algorithm Parameters*

The following parameters were used in the algorithm:

- Mutation probability: 0.4
- Crossover probability: 0.6
- Learning rate: 0.95
- Population size: 20
- Number of generations: 1000

5.5. Test Results

The examined task consists of 22 structural elements, contains 24 feedbacks and 16 crossovers. The initial sequence requires 12066 time units and 12020 cost units. The initial matrix can be seen in *Fig. 4*, where S.E. denotes the number of structural elements, T is the design time and C is the cost of planning.

The impact of the weighing factors was studied during the optimization process. The values of weighing were the same, i.e. $w_c = w_t = 0.5$ during the first optimization. *Fig. 8* shows the optimized sequence, while *Fig. 9* reveals the optimization process. The cost demand decreased to 4476 units, while the time demand to 5359 units as a result of the optimization.

The weight values were the following during the second optimization: $w_c =$ 0.7; w_t =0.3. *Fig. 10* shows the optimized sequence, while *Fig. 11* reveals the optimization process. The cost demand decreased to 4450 units, while the time demand to 5370 units as a result of the optimization.

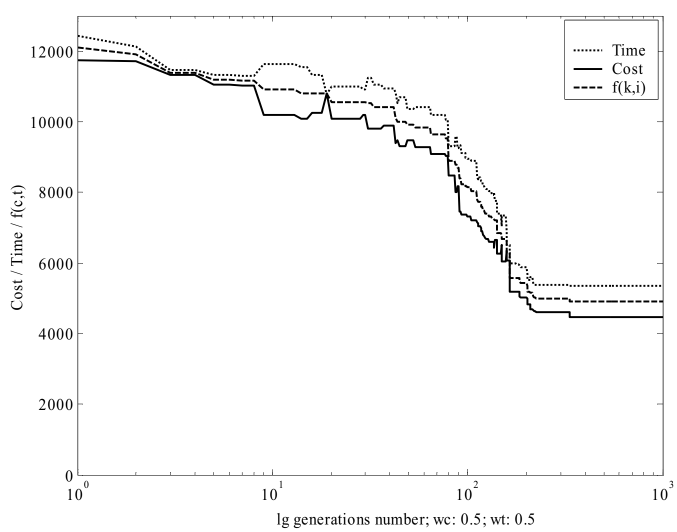

Fig. 9. Optimization process 1

S. E	T	C	12	11	6	21	3	16	17	18	19	20	10	15	5	13	14	8	7	9	2	1	4	22
12	30	30		1																				
11	40	20			1																			
6	60	10				1	1					1						1				1		
21	20	40						1																
3	10	60								1														
16	20	40							7					1										
17	30	30						7		8			1											
18	40	20							8		6				1									
19	60	10								6		2												
20	40	20									2						1	1						
10	20	40															1	1						
15	30	30															1	1						
5	10	60																1						
13	30	30	3																					1
14	20	40														8								
8	60	10																	1	2	1			
7	40	20															1							
9	20	40																2					1	
2	40	20																				1		
1	30	30																		1				
4	10	60															1							
22	20	40																						

Fig. 10. Optimized DSM 2

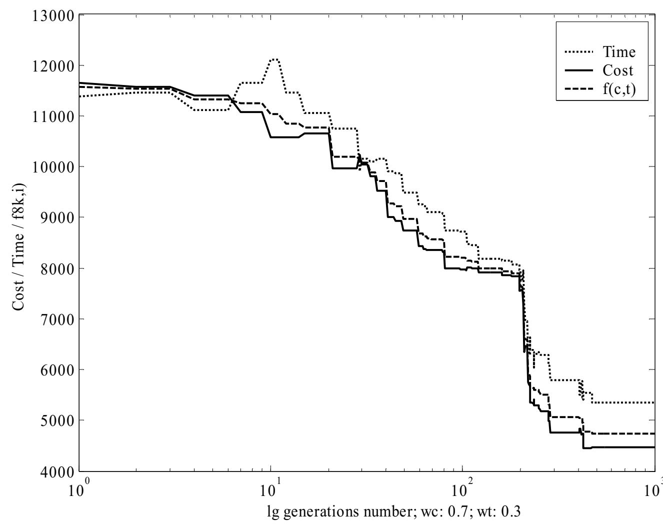

Fig. 11. Optimization process 2

5.6. Conclusions

The examinations revealed inevitably that efficiency of the 'Better Half' selection is much worse than that of the 'Tournament' selection in case of tasks of large dimension. The tests also showed that if the average fitness value of the population reaches that of the best individual, the algorithm sticks, since only the same individuals are present and the one-point crossover is not efficient. Search is only done due to the mutation probability and its small value slows down the process very much.

This was experienced in case of the 'Better Half' selection since there only the better half of the population takes part in offspring production. Thus, it converges more quickly than the 'Tournament' selection but excludes the individuals which have worse fitness values but may contain sequences that could be positioned in a right place during crossovers.

The examinations also proved that it is worth choosing a higher mutation/crossover ratio than the usual 0.2/0.8 if the dimension of the matrix increases. Since the algorithm implemented by us used only one-point mutations and crossovers, the trial of 2-point mutation and crossover is also planned in a further research project. A module that changes the value of mutation dynamically is also to be introduced in order to sustain the selection pressure.

The conclusion that the simultaneous application of 0.4/0.6 mutation/crossover probability ratio and 'Tournament' selection is the best process in case of genetic algorithms created for decimally coded precedence tasks.

In order to understand the operation of the algorithm it should be noted that if the target function is chosen adequately, the algorithm arranges the related elements into subprocesses (and this way decreased the time and costs of the process) without using a separate clustering algorithm (see *Fig. 8*, the part between elements 12-13 and 14-22). The conclusion can be that the developed method provides a possibility for adequate design process optimization drawn. Furthermore, it can co-work with the standardized systems and can be converted back and forth.

The system efficiency is to be increased with the help of the probabilities introduced, hence the system will be able to provide an adequate solution even when a new product is introduced in a way that the weakest relation are not taken into consideration. Presently total man hours are used in the calculation of the target function, since this is optimal concerning the sequence but the process cannot be positioned in time. Real time can be calculated if the optimized DSM is converted into a network plan. This makes it possible to assign the resources to the tasks on the basis of the scheduled process, the requirements of the project.

Acknowledgement

The research was supported by application OTKA T032474.

References

[1] ABRAMSON, D.– LEWIS, A.– PEACHEY, T. – FLETSHER, C., An Automatic Design Optimization Tool and its Application to Computational Fluid Dynamics, *Proc. ACM/IEEE SC2001 Conf., Denver, CO*, 2001.

[2] ALTUS, S. S.– KROO, I. M.– GAGE, P. J., A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems, Proc. ASME, 1995, pp 95-141.

[3] BLOEBAUM, C. L., An Intelligent Decomposition Approach for Coupled Engineering Systems, Proc. Fourth AIAA/AF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Cleveland, OH, 1992.

[4] DONALD, G. R., Die neuen Werkzeuge der Produktentwicklung, Carl Hanser Verlag, München, Wien, 1998.

[5] EHRLENSPIEL, K., Integrierte Produktentwicklung: Methoden für Prozessorganisation, Produktentwicklung und Konstruktion, Carls Hanser Verlag, München, Wien, 1995.

[6] GAREY, M. R.– JOHNSON, D. S., Computers and intractability: A guide to the theory of NP-completeness, W. H. Freeman & Co., New York, 1979.

[7] HOLLAND, J., Adaptation in Natural and Artificial Systems, MIT Press, Cambridge, MASS, 1975.

[8] MARCA, D. A.– MCGOWAN, C. L., *Structured Analysis and Design Technique: SADT*, McGraw-Hill, New York, 1988.

[9] RECHENBERG, I., *Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipen der biologischen Evolution*, Friedrich Frommann Verlag, Stuttgart, 1973.

[10] ROGERS, J. L., DeMAID/GA - An Enhanced Design Manager's Aid for Intelligent Decomposition, *Proc. 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Seattle, WA , September 1996*, No. 96-4157.

[11] STEWARD, D. V., *System0020 Analysis and Management: Structure, Strategy and Design*, Petrocelli Books Inc., 1981.

[12] SYSWERDA, G., *Schedule Optimization Using Genetic Algorithms*, Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.

[13] BACK, T., Generalized Convergence Models for Tournament- and Selection, *Proc. Sixth Int. Conf. on Genetic Algorithms*, L.J. Eshelman ed., Morgan Kaufmann Publishers, San Francisco, 1995.

[14] VDI-RICHTLINIE 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, VDI-Verlag, Düsseldorf, 1986.

[15] www.Idef.Com