DESGIN STRUCTURE BASED TASKS SCHEDULING USING GENETIC ALGORITHMS

István Gromma, Tamás Rick,
Institute of Machine Design Faculty of Mechanical Engineering
Budapest University of Technology and Economics
H–1111 Budapest, Műegyetem rkp. 3., Hungary
e-mail: rick.tamas@gszi.bme.hu

Abstract: *This paper deals with a modeling and optimization method that is capable of the product structure based optimization of design processes. The processes were modeled with Design Structure Matrix. Since the precedence task is a whole NP one, guided stochastic search techniques were applied. The impact of the probability of mutation and crossover and the two different selection processes on searching was examined.*

Key words: *product development process, genetic algorithm (GA), design structure matrix (DSM), 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, hence optimal resource management are essential criteria when the efficiency of the process is determined [4]. 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). Hierarchic relation models, graphical description methods like the SADT are applied in activity and data modeling. 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.

2. MODELLING

Process elements, the logical relations of which are defined in rules, are applied in process modeling and description [5]. The most wide spread, standardized process modeling method is the SADT (Structured Analysis and Design Technique) [9], 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 due to graphical and hierarchical mapping (only a limited number of boxes can be illustrated on the given sheet size). 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).

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

Further information, such as the time and cost of design, can be assigned to the matrix elements. These pieces of information are shown in Figure 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 Figure 3.

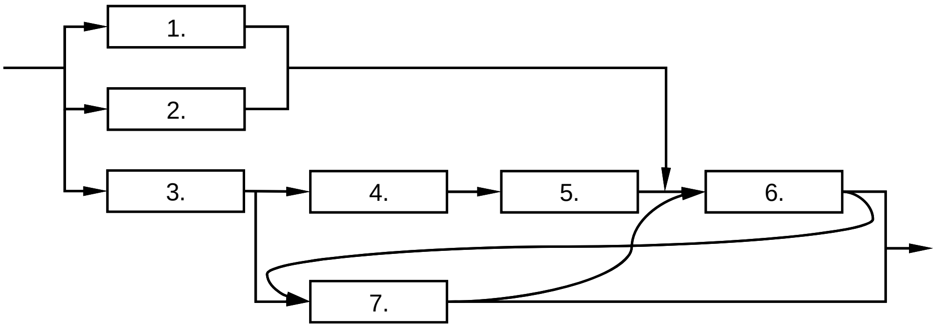
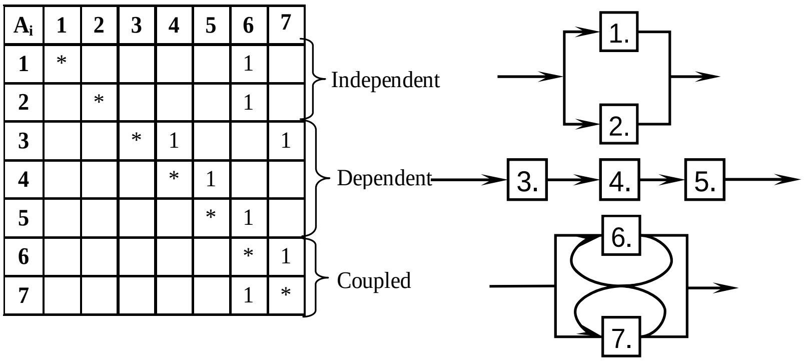


Fig. 2 Process blocks



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 Figure 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 Figure 4, where cycle 20-14 crosses cycle 17-10.

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	50			■															1				
4	10	50				■										1								
5	10	50					■			1														
6	50	10	4		2			■		1												1	1	
7	40	20							■							1								
8	50	10		8					6	■	2													
9	20	40				7				2	■													
10	20	40								8		■				1								
11	40	20						8					■											
12	30	30											5	■										
13	30	30												3	■									1
14	20	40													6	■								
15	30	30								4						8	■							
16	20	40															6	■	7					
17	30	30										8						7	■	8				
18	40	20					6												8	■	6			
19	50	10																		6	■	2		
20	40	20								7						4					2	■		
21	20	40																8					■	
22	20	40																						■

Fig. 4 DSM example

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. It has been shown earlier that the scheduling problem subject to precedence and resource constraints is NP-Hard [6] [7], leaving exact methods time consuming and inefficient at solving large problems and real-world applications. 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 [11]. 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).

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 [8, 10].

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 “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” [14], 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.

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 [13].

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 was examined.

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).

5.1. 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 Equation (1) and the cycles were also taken into consideration.

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

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.

5.2. 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.

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

Table 1 Optimization results

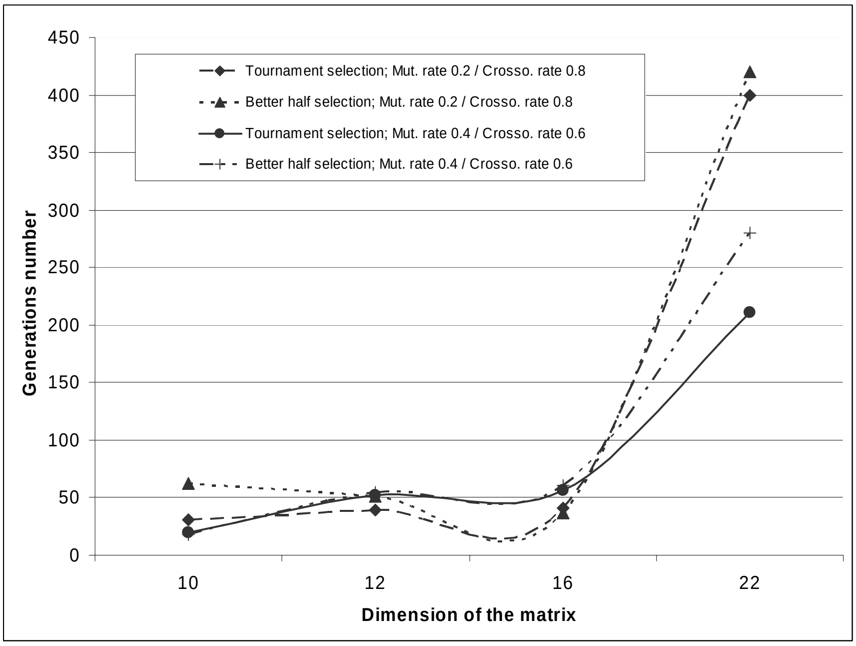


Figure 7 Efficiency of selection processes

Figure 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 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 combination.

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 ¾ of it was set to be feedback. The population size was 20 during the tests.

6. CONCLUSION

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.

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.

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. The research supported by application OTKA T032474.

7. REFERENCES

[1] Abramson, D., Lewis, A., Peachey, T., Fletcher, 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, ASME Paper 95-141., 1995.

[3] Bloebaum, C.L.: An Intelligent Decomposition Approach for Coupled Engineering Systems, Proc. of the Fourth AIAA/AF/NASA/OAI Symp. on Multidisciplinary Analysis and Opt., Cleveland, OH. 1992.

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

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

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

[7] Hartmann, S. A competitive genetic algorithm for resource-constrained project scheduling. Naval Research Logistics 45:733-750., 1998.

[8] Holland, J.: Adaptation in natural and artificial systems, MIT Press, Cambridge, Mass 1975

[9] Marca, D. A.; McGowan, C. L.: Structured analysis and design technique: SADT, McGraw-Hill, New York 1988

[10] Rechenberg, I.: Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipen der biologischen Evolution, Friedrich Frommann Verlag, Stuttgart 1973

[11] Rogers, J., L.: DeMAID/GA - An Enhanced Design Manager's Aid for Intelligent Decomposition, 6th AIAA/USAF/NASA/ISSMO Symp. on Multidisciplinary Analysis and Optimization, Seattle, WA , September 4-6, 1996a. AIAA paper No. 96-4157.

[12] Steward, D., V.: System Analysis and Management: Structure, Strategy and Design, Petrocelli Books Inc. 1981.

[13] Syswerda, G.: Schedule Optimization Using Genetic Algorithms, Handbook of Genetic Algorithms, Van Nostran Reinhold, New York, 1990.

[14] Back, T.: Generalized Convergence Models for Tournament- and Selection, In Larry J. Eshelman, editor, Proc. Sixth Int. Conf. on Genetic Algorithms, 2-8. Morgan Kaufmann Publishers, San Francisco, 1995.

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