

# COMPARISON OF DIFFERENT APPROACHES TO THE CUTTING PLAN SCHEDULING

PETER BOBER

## 1 INTRODUCTION

Cutting plan is a program for a NC cutting machine (e.g. plasma cutting) based on which multiple parts are cut out of a single piece of sheet. The production engineering department has a set of cutting plans for each cutting machine at hand. The set of plans increases gradually in size along with extending of the assortment and with purchasing of new cutting machines. Designing a cutting plan takes some time, and hence it is impossible to develop customized cutting plans to comply with immediately needed assemblage of parts (parts production plan). A specific part is appearing on one or more cutting plans in varying numbers. Usually, a cutting plan is being cut as a whole, and thus some “extra” parts are also often manufactured that are transferred to the stock for future use. Hundreds of parts should be scheduled for machining with many possible combination of cutting plans. The combination with the minimum surplus parts and/or minimum machining time is needed to be selected. It should be stressed that the criteria function can depend on actual market situation or multiple objectives exist. Therefore practical optimization algorithm should reflect these requirements. Otherwise multiple algorithms must be used.

The task being resolved falls into the combinatorial optimization (CO) issues class where multiple optimization methods and hundreds of algorithms exist. The methods and algorithms are used across different application areas. Grossmann et al. (2002), Gutin et al. (2003), Pentico (2007), and Sergienko et al. (2009) try to classify methods and types of CO problems.

One group of methods is Mathematical programming (Linear and Mixed-Integer Programming, Nonlinear programming). The presented problem is special type of multi-set multi-cover program. Rajagopalan and Vazirani (1993) proposed a parallel approximation algorithm but without constraints, which exist in our problem. Other algorithm published by Cal'egaria et al. (2001) and latest exact algorithm of Hua et al. (2010) have different objective function. If we omit general methods like exhaustive search or branch and bound than we didn't find an exact or approximation algorithm which can be used directly for our purpose in literature.

Another group consists of stochastic methods (e.g. local search, climbing algorithm, random search, taboo search, genetic algorithms, simulated annealing,

quantum annealing, swarm optimization, etc.). Each of these methods can be adapted for presented problem. Genetic algorithms (GA) are often used for scheduling problems as well as other engineering systems (Zalzala & Fleming, 1997). Slak et al. (2011) presented quite complex integrated model with production planning and scheduling function. On the basis of ERP system it schedules the activities on the shop floor in real time. GA is used for optimization of task schedule. Zhang and Ishikawa (2004) proposed a hybrid GA to improve the performance of the standard GA for solving CO problems with time-varying parameters.

One more group of optimization methods are heuristics and metaheuristics. These methods combine different approaches to obtain better performance in optima search. Jaszkiwicz (2002) proposed genetic local search which may be interpreted as a standard GA working on set of local optima. Teghem et al. (2000) adapted the simulated annealing for multi-objective CO problem and propose an interaction with the decision maker during optimization process. Bubeník (2004) proposed a simulation techniques for work-order scheduling which depends upon the availability of shop-floor data and the response time of the evaluation process.

From the literature survey it results that the question is not in existence of an algorithm itself but which one is applicable for the presented problem from practical point of view. Many authors (e.g. Dhaenens, 2010) offer time and efficiency measurement of their algorithms for benchmark instances. However it is hard to estimate time requirements for particular problem type and size. Therefore the experimental implementation of different algorithms should be done to answer the question. We choose three methods. One is an exact method, which can find optima for a reduced problem size and enables to check other methods. The second method belongs to group of GA which have potential to solve also big CO problems. Last method falls into group of heuristic methods and constructs a solution upon a set of rules.

## 2 PROBLEM FORMULATION

The core of the problem is selecting the specific machine, specific cutting plan, and defining the number of sheets to be cut to manufacture required quantity of parts in order to minimize or maximize an objective function. The objective function usually includes the total manufacturing time, costs and other criteria such as decreasing the number of surplus parts. An example of the outcome of an optimization we need for a production is presented in Table 1.

Formal definition of the problem is represented by triplet  $(X, P, f)$ , where:

- $X$  is the solutions space,
- $P$  is the solution constraints,
- $f$  is the objective (fitness) function.

Solution is represented by  $n$ -tuple  $\bar{x} = (a_1, a_2, \dots, a_n)$ , where  $a_i$  is the number of sheets cut according to the cutting plan  $i$ , and  $n$  is the number of available cutting plans. Constraint  $P$  determines the feasible solutions space  $Y \subseteq X$  from all possible combination of  $a_i$  values, and it is expressed as follows: the time capacity  $C_k$  of any machine  $k$  will not be exceeded.

Table 1 – Example of a daily production plan (outcome from optimization)

Daily production plan		Date:			
No.	Machine	Cutting plan	Number of pieces	Material	Drawing number
1.	Laser L4030S	2245	5	235656	264
2.	Plasma HD 4070	2485	10	286262	321
3.	...				

### 2.1 Objective function

The objective (fitness) function determines how good the solution is and it can reflect various facts such as:

1. The shortest period of time needed to produce the planned quantity.
2. At respecting required precision, selection the most cost-favorable machine.
3. Minimizing the number of surplus (extra) parts.

In this paper a single aggregate objective function was chosen. It is determined as weighted sum of individual objectives:

$$f(\bar{x}) = \sum_{i=1}^I w_i M_i(\bar{x}) \tag{1}$$

where  $\bar{x}$  is the solution,  $M_i(\bar{x})$  is value of the metrics that reflects meeting of objective  $i$  and  $w_i$  is the weight of objective  $i$  expressing its importance. An arbitrary combination of criteria can be included in the objective function. Non-linear metrics that allow taking account constraints where used:

$$f(\bar{x}) = w_1 M_1(\bar{x}) + w_2 M_2(\bar{x})$$

$$M_1(\bar{x}) = \begin{cases} \frac{T_p}{C_T} & \text{if time capacity is not exceeded} \\ 10 \cdot \frac{T_p}{C_T} & \text{otherwise} \end{cases} \tag{2}$$

$$M_2(\bar{x}) = \begin{cases} \frac{N_s}{N_p} & \text{if plan fulfilled} \\ 10 \cdot \frac{N_s}{N_p} & \text{otherwise} \end{cases}$$

where

$T_p$  is sum of processing times of machines,  
 $C_T$  is total time capacity of all machines,  
 $N_s$  is number of surplus parts produced over the plan,  
 $N_p$  is sum of parts according the plan,  
 $w_1$  and  $w_2$  is weight of the time criteria and the number of parts criteria respectively ( $w_1=0.1, w_2=0.9$ ).

If any machine exceeds its time capacity or there is not sufficient number of produced parts of some type then penalization coefficient 10 is used. The best value is “minimum” (the objective function is a penalization function).

The information to evaluate the objective function is as follows:

1. Production plan.
2. Operational time to prepare for cutting and cutting time of single sheet.
3. Time capacity of each machine.

This data can be obtained by measuring or evaluating from CAD software according cutting plan parameters what can be find in Fabian et al. (2009).

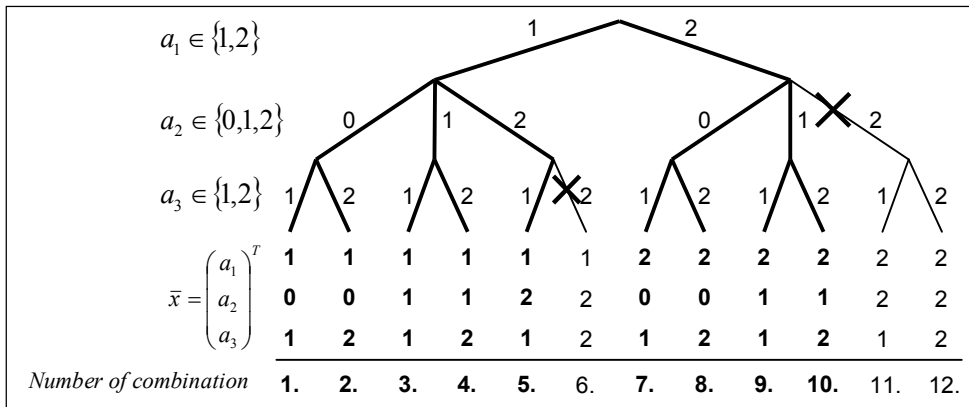


Figure 1 - The tree representation of a simple combinatorial problem.

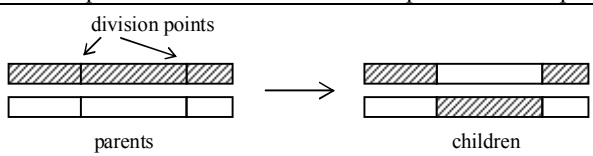
### 3 BACKTRACKING ALGORITHM

Direct approach to the combinatorial issue optimization is the brute-force or exhaustive search method, which enumerate the objective function for every solutions and finds the best one. The back-tracing algorithm (BA) named by D. H. Lehmer and early used by Golomb and Baumert (1965) can reduce number of calculation by incrementally building candidates to the solutions, and abandons each partial candidate  $\bar{x}$  (“backtracks”) as soon as it determines that  $\bar{x}$  cannot possibly be completed to a valid solution. Advantage of the method is that in principle it always finds the optimum. The disadvantage remains in huge calculations that restrict its practical use for limited size problems only.

Illustrated in Fig. 1 is a tree-shaped example of the introduced combinatorial problem. Each single solution is presented by triplet  $\bar{x} = (a_1, a_2, a_3)$ . Some

branches fail meeting the constraints  $P$  and hence are removed (combinations 6, 11 and 12). Thus, size of the tree decreases same as the time needed for searching. The solution constraints check is run within each node of the tree. The objective function is only evaluated for leaves of the tree. Within the algorithm used the tree has been developed recursively. A back-tracing algorithm for introduced problem was developed by author (Bober, 2007).

Table 2 – Proposal of genetic algorithms for planning of a daily production plan

Coding of the solution in chromosome	Chromosome is a chain of numerals $a_i$ of solution $\bar{x} = (a_1, a_2, \dots, a_n)$ , where $a_i$ stands for numbers of pieces of sheet cut according plan $i$ .
Creating new generation	The elite of the preceding generation is passed to a new one. Other individuals emerge by crossover. Probability that an individual solution will proceed to the crossover process is proportional to its fitness function. The mutation is performed after crossover with predetermined probability.
Multipoint crossover	
Mutation	During mutation, values of some genes are randomly altered.
Objective (fitness) function	$f(\bar{x}) = w_1 M_1(\bar{x}) + w_2 M_2(\bar{x})$ , see equation (2)

#### 4 GENETIC ALGORITHMS

The problem resolving technique by use of evolution principle was introduced by Holland in 1972 but it were only Goldberg works of the late 80s that triggered broader utilization of GA in a variety of fields (see Goldberg, 1989). Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. Solution is such a combination of properties (string structure) at which the objective (fitness) function acquires the extreme. The method does not guarantee finding the best solution but in practice “a solution close to the optimum” commonly suffices.

When applying GA to solving of a specific problem it is necessary to:

1. Define coding of the solution into the string structure (here vector ) called chromosome. Individual members of the chain are called genes.
2. Propose operators for selection, reproduction, crossover and mutations of solutions.
3. Define the objective (fitness) function.
4. Determine the GA parameters.
5. Select the first generation of random solutions.

Combinatorial problems are usually extensive but GA and other discrete optimization methods of these days are sufficiently powerful as well. For a

number of issues, speed of the GA can be further increased as suggested in by Duque et al. (2008).

Selection of parent solution into crossover and mutation operators, defining of the fitness function and selection of other parameters depend on the issue being resolved, and hence there is no strict procedure to specify them. Thus, it is possible to develop varying ways that differ in the subsequent computer implementation. Shown in Table 2 is a proposal for designing these for cutting plans as author explained in (Bober, 2008).

## 5 HEURISTIC ALGORITHM

Heuristic algorithm reduces combinatorial complexity by use of the rule for selection of a cutting plan. The method does not guarantee finding the extreme of the objective function but practically it is suitable for a lot of problems.

Cutting plans are rated by function  $v$ , which represents the highest priority of the part appearing on the cutting plan. The part priority equals the number of cutting plans on which is the part appearing (the lower the value the higher the priority). Plans with the lowest  $v$  will be included into the plan first. Value of  $v=1$  for a cutting plan means that some part is produced solely in accordance with this single cutting plan, and hence it must be scheduled into production. Sequential procedure will secure selection of cutting plans for the overall production plan. If the value  $v$  is identical for more cutting plans then the cutting plan with the lowest value of the additional criteria will be selected. The criteria is calculated as a weighted sum of the processing time and of the number of extra parts produced, similar to equation (2).

## 6 COMPARISON OF METHODS

The methods were compared on two problems. Size of the first problem is reduced to 7 parts, which dramatically reduces the number of combinations to  $7.39 \times 10^{10}$ . This smaller problem serves for verification of the functionality of methods. In the second case, plan is prepared for 130 various parts that are commonly produced in a single day. Estimated number of combination is  $10^{63}$  in this case. Investigated in individual methods were: the ability to find the best solution, value of the objective function, and the execution time necessary to arrive to a solution. Programming language Delphi Pascal was chosen to implement methods.

Table 3 indicates that each method used was successful in finding the best solution for reduced size problem. Table 4 shows results for full size problem. Back-tracing algorithm is not applicable for such big problem. Values for GA are average values of the objective function and execution time as the nature of the algorithms is stochastic, and calculations consisted of five runs at varying initial conditions.

*Table 3 – Comparison of methods for a reduced problem (7 parts, 27 cutting plans)*

	Method		
	BA	GA	HA
Optimum found	Yes	Yes	Yes
Objective function	0.21	0.21	0.21
Execution time	26 h 56 min*	95 s *	10 s **

\* Intel Core2 1.8GHz

\*\* AMD Athlon XP3500+ 2.2 GHz

*Table 4 – Comparison of methods for full size problem (130 parts, 285 cutting plans)*

	Method		
	BA	GA	HA
Optimum found	-	?	?
Objective function	-	Avg. 0.785 Worst run 0.789 Best run 0.77	0.792
Execution time	-	2 h 32 min * (average) 112 h * (best solution)	3 min 30 s **

\* Intel Core2 1.8GHz

\*\* AMD Athlon XP3500+ 2.2 GHz

## 7 DISCUSSION

The below conclusions follow from the facts included in Tables 3 and 4:

1. All three methods are able to find the optimum solution for a small size problem.
2. Satisfactory but not optimum solutions can be found for full size problem only by the use of GA and of heuristic algorithm.
3. GA can find better solution.
4. Heuristic algorithm is fastest though from its nature it follows.

Heuristic algorithm is closely connected with the objective function. Hence, at any change to the objective function the algorithm must be reworked. If the problem nature does not require altering the objective function, the heuristic algorithm would be the best one to be applied. In practice, however, it happens that the objective function changes in dependence on external circumstances. Again, GA are more appropriate in this case the on the assumption that execution time will not be unacceptable long.

## 8 CONCLUSIONS

The properties of three methods of solving a specific scheduling problem of scheduling cutting plans were the subject of our research. These methods are: backtracking algorithm, genetic algorithms, and heuristic algorithm.

Results of experiments suggest that genetic algorithms are time consuming but are able to find better solutions than the heuristic algorithm. Moreover, they are not dependent on the selected objective function. Heuristic algorithm is the fastest one but from its nature follows that it is closely connected with the objective function, and therefore if objective function is changed the algorithm must be changed as well. If the scheduling by use of genetic algorithms issue is solvable in a sufficiently short period of time then it is, from practical view, more appropriate than the heuristic algorithm. Furthermore genetic algorithms find better solution. The back-tracing methods can supply results only with scheduling issues of small size.

## ACKNOWLEDGMENTS

This work was supported by Scientific Grant Agency of the Ministry of Education of the Slovak Republic and of Slovak Academy of Sciences, project VEGA 1/0006/10.

## REFERENCES

- Bober, P (2007), "Optimization of cutting plans for more machines by searching of solution tree", *Journal "Quality Innovation Prosperity"*, vol. 11, no. 2, p. 9-17 (In Slovak).
- Bober, P (2008), "Optimization of Cutting Plans Using Genetic Algorithms", *Journal "Quality Innovation Prosperity"*, vol. 12, no. 2, p. 1-10, (In Slovak).
- Bubeník, P. (2004), "A scheduling system for minimizing the cost of productions", *Strojníški Vestník - Journal of Mechanical Engineering*, vol. 50, no. 5, ISSN 0039-2480, p. 291-297.
- Cal'egaria P., Guideca, F., Kuonena, P., Nielsen, F. (2001), "Combinatorial optimization algorithms for radio network planning", *Theoretical Computer Science* 263, p. 235–245.
- Dhaenens, C., Lemesre, J., Talbi, E.G. (2010), "K-PPM: A new exact method to solve multi-objective combinatorial optimization problems", *European Journal of Operational Research*, vol. 200, no. 1, p. 45-53, doi:10.1016/j.ejor.2008.12.034.
- Duque, T., Goldberg, D. E., Sastry, K. (2008), "Enhancing the Efficiency of the ECGA", TR No. 2008006, Retrieved on 3.11.2010 from <http://www.illigal.uiuc.edu/pub/papers/IlliGALs/2008006.pdf>.



- Fabian, M., Spišák, E., Šeminský, J., Dovica, M. (2009), “Anticipation of Cutting Surface Quality from Pre-Set CAM Parameters”, *Ovidius University Annual Scientific Journal*, vol. 11, no. 1, p. 19-24.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, ISBN 0-201-15767-5.
- Golomb, S. W., Baumert, L. D. (1965), “Backtrack Programming”, *Journal of the ACM*, vol. 12, no. 4, p. 516-524, DOI: 10.1145/321296.321300.
- Grossmann, I. E., Heever, S. A., Harjunkoski, I. (2002), “Discrete Optimization Methods and their Role in the Integration of Planning and Scheduling”, *AIChE Symposium Series*, no. 326, vol. 98, p. 150-168.
- Gutin, G., Vainshtein, A., Yeo, A. (2003), “Domination analysis of combinatorial optimization problems”. *Discrete Applied Mathematics*, vol. 129, no. 2-3, p. 513-520, DOI: 10.1016/S0166-218X(03)00359-7.
- Hua, Q-S., Wang, Y., Yu, D., Lau, F. C. M. (2010), “Dynamic programming based algorithms for set multicover and multiset multicover problems”. *Theoretical Computer Science*, vol. 411, no. 26-28, p. 2467-2474, DOI: 10.1016/j.tcs.2010.02.016.
- Jaszkiewicz, A. (2002), “Genetic local search for multi-objective combinatorial optimization”, *European Journal of Operational Research*, vol. 137, no. 1, p. 50-71, DOI: 10.1016/S0377-2217(01)00104-7.
- Pentico, D. W. (2007), “Assignment problems: A golden anniversary survey”, *European Journal of Operational Research*, vol. 176, no. 2, p. 774-792, DOI: 10.1016/j.eor.2005.09.014.
- Rajagopalan, S., Vazirani, V. V. (1993), “Primal-dual RNC approximation algorithms for (multi)-set (multi)-cover and covering integer programs”. In *SFCS '93 Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science*, Washington, DC, p. 322 – 331, DOI: 10.1109/SFCS.1993.366855.
- Sergienko, I. V., Hulianytskyi, L. F., Sirenko, S. I. (2009), “Classification of applied methods of combinatorial optimization”, *Cybernetics and Systems Analysis*, vol. 45, no. 5, p. 732-741.
- Slak, A., Tavčar, J., Duhovnik, J. (2011), “Application of Genetic Algorithm into Multicriteria Batch Manufacturing Scheduling”, *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 57, no. 2, p. 110-124, DOI:10.5545/sv-jme.2010.122.
- Teghem, J., Tuytens, D., Ulungu, E. L. (2000), “An interactive heuristic method for multi-objective combinatorial optimization”, *Computers & Operations Research*, vol. 27, no. 7-8, p. 621-634, DOI: 10.1016/S0305-0548(99)00109-4.
- Zalzala, A. M. S., Fleming, P. J. (editors) (1997), *Genetic algorithms in engineering systems*, London: Institution of Electrical Engineers, ISBN 0 85296 902 3.

Zhang, H., Ishikawa, M. (2004), “A solution to combinatorial optimization with time-varying parameters by a hybrid genetic algorithm”, in *International Congress Series*, vol. 1269, Brain-Inspired IT I. Invited papers of the 1st Meeting entitled Brain IT 2004, August 2004, p. 149-152,  
DOI: 10.1016/j.ics.2004.05.019.

---

#### **ABOUT THE AUTHOR**

**Peter Bober**, PhD. is assistant professor at the Department of Electrical Engineering and Mechatronics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia  
e-mail: peter.bober@tuke.sk