

A GA-BASED CONSTRAINT SATISFACTION MODEL FOR GENERATING OPTIMAL PROCESS PLANS

Amr I. Shabaka and Hoda A. ElMaraghy*
*Intelligent Manufacturing Systems Centre,
Industrial, University of Windsor, Canada,
{shabaka, hae}@uwindsor.ca*

In this paper a new process planning approach is proposed in order to achieve the lowest cost using Genetic Algorithms. It considers all process planning parameters simultaneously while simplifying the problem formulation and reducing the computational complexity. A new approach that guarantees operations with specific constraints will be clustered together is proposed. The method introduces the use of a string of continuous variables to represent a process plan. A new method, which insures that any randomly generated chromosome will result in a feasible process plan, is presented. The problem formulation is described and illustrated with an example. The presented optimal process planning method can also aid the part/ machine assignment activities.

1. INTRODUCTION

The Society of Manufacturing Engineers defines Process planning as "the systematic determination of the methods by which a product is to be manufactured economically and competitively". This is achieved by utilizing the available system capabilities and resources. In the past 3 decades there has been increasing efforts to develop and improve CAPP systems that aid human process planners. Altng and Zhang (1989) and H. ElMaraghy (1993) provide comprehensive literature surveys on CAPP. Process planning is a multi-decision making activity that determines the operations selection and operations sequencing which involve a great deal of manufacturing data. Operation selection and sequencing are important manufacturing activities (Reddy *et al.*, 1999).

The work reported in this paper presents a new process planning optimization model using GAs to obtain the optimum machining sequence, machine assignment and assign tools to operations. GAs are used because development of various feasible plans and identifying the best solution has been proven to be NP-complete (Reddy *et al.*, 1999). To achieve an optimal process plan, operation selection and operation sequencing have to be carried out simultaneously (Zhang *et al.*, 1999). In process planning, when using GAs, the choice of objective function and how to obtain a feasible process plan are important.

* *Corresponding Author*

Dereli and Filiz (1999) and Reddy *et al.* (1999) formulated the objective function using a reward/penalty matrix. This meant having to develop a penalty matrix for every new part. Also only operation sequencing can be considered in this approach. Zhang and Nee (2001), Li *et al.* (2002) and Ong, *et al.* (2002) minimized cost based on a multi-objective function, provides a more comprehensive and general solution compared to using the penalty matrix because it not only provides the sequence of operations, the output will also include operation selection.

Generating a feasible process plan is a challenge due to the randomness used in GAs. Most works (Dereli and Filiz, 1999; Reddy *et al.*, 1999; Li *et al.*, 2002; Ong, *et al.*, 2002) generate the 1st initial population randomly. A test is then carried out to select the feasible process plans (Reddy *et al.*, 1999) or repair the infeasible ones (Li *et al.*, 2002). When the crossover or mutation operators are used, testing and repairing is carried out but the child chromosome will contain different characteristics than both parents which defeats the purpose. Tang *et al.* (2004) used the partial precedence graph sorting technique to translate a given chromosome into a feasible sequence. However, the method does not know the limits of each variable as they change from one sequence to another.

ElMaraghy and Gu (1987) first introduced the concept of clustering parts and features according to their tolerance datums and inspection requirements for task planning of CMM machines. Shabaka and ElMaraghy (2005) proposed a method for clustering operations with tolerance and logical constraints into operation clusters. Although some research used tolerances to create precedence relations between operations, forcing operations, which have tight tolerance to be processed on the same machine has not been proposed earlier in process planning. This is important because it is much cheaper to perform operations with tight tolerances on the same machine as it would reduce the required number of highly capable machines.

Other optimization techniques apart from GAs are Simulated Annealing (Ong, *et al.*, 2002), Petri nets (Kiritsis and Prochet, 1996), Graph theory (Ciurana *et al.*, 2003) and Taiber (1996) applied a set of modified algorithms from the field of combinatorial search problems.

2. PROBLEM FORMULATION

Given a part with NOP number of operations, the approach proposed by Shabaka and H. ElMaraghy (2006) is used to obtain the formed OCs and the matrix for machines capable of producing each OC depending on the combination of TADs used (referred to as TAD Odd Used). This data is used as input to find the optimal process plan, which reduces the cost of machine usage, machine change, tool usage, tool change and setup change.

2.1 Nomenclature

<i>OP</i> : Operation	x,y : index for OP #, $x,y=1,\dots,NOP$	d : index for TAD
<i>NOP</i> : # of <i>OPs</i>	i,j : index for <i>OC</i> #, $i,j=1,\dots,NOC$	
<i>OC</i> : Operation cluster	t : index for tool #, $t=1,\dots,\#$ of tools	
<i>NOC</i> : # of <i>OCs</i>	m : index for machine # $m=1,\dots,\#$ of machines	

2.2 Inputs

The following input parameters and information that are assumed to be available.

NOP	Number of Operations
$CM(m)$:	Array containing the cost of using machine m
$CT(t)$:	Array containing the cost of using tool t
$OPP(x,y)$:	Operation precedence relation between OP_x and OP_y
$OPTAD(x,d)$:	Matrix containing the possible TADs for each operation

The following are obtained from the above inputs using the approach proposed by Shabaka & ElMaraghy (2006). They are also the inputs to the optimization model.

$NOPC(i)$:	Array containing the number of operations per operation cluster i
$OCP(i,j)$:	Operation Cluster precedence relation between OC_i and OC_j
$NOCO(i)$:	Array containing the number of OC odds for each OC
$OCO(z_i,d)$:	Matrix that contains the required TADs to manufacture the different odds of OC_i where z_i is the index for $NOCO(i)$
$MCAP_i(od, m)$:	Matrix containing list of machines capable of producing the different odds for OC_i

3. MATHEMATICAL MODEL

3.1 Decision Variables

<i>OC Sequence</i>	$OCS = \{oc_1, oc_2, \dots, oc_{NOC}\}$, where oc_i is OC taking the i^{th} sequence
<i>Odd Numbers Used</i>	$OD = \{od_1, od_2, \dots, od_{NOC}\}$, where od_i is the TADs odd number used for oc_i
<i>Machines Sequence</i>	$MS = \{m_1, m_2, \dots, m_{NOC}\}$, where m_i is the machine type assigned to the OC in the i^{th} position of the sequence
<i>Operation Sequence</i>	$OPS = \{op_1, op_2, \dots, op_{NOP}\}$, where op_x is the OP taking the x^{th} position in the sequence of Ops
<i>TAD Sequence</i>	$OPS = \{op_1, op_2, \dots, op_{NOP}\}$, where op_x is the OP taking the x^{th} position in the sequence of Ops
<i>Tools Used</i>	$TS = \{t_1, t_2, \dots, t_{NOP}\}$, where t_x is tool type assigned to op_x

3.2 Objective Function and Constraints

The objective function to minimize the total cost (TC):

$$\text{Min TC} = \text{MUC} + \text{TUC} + \text{MCC} + \text{TCC} + \text{SCC} \quad (1)$$

Machine Usage Cost (MUC): cost of using each machine for every operation cluster

$$MUC = MCI \times \sum_{i=1}^{NOC} CM(m_i), \text{ where MCI is the machine cost index} \quad (2)$$

Tool Usage Cost (TUC): cost of using every tool for each operation

$$TUC = TCI \times \sum_{x=1}^{NOP} CT(t_x), \text{ where TCI is the tool cost index} \quad (3)$$

Machine Change Cost (MCC): cost of changing machine between two consecutive operation clusters

$$MCC = MCCI \times \sum_{i=1}^{NOC-1} [\Omega(MS(i), MS(i+1))], \text{ where } \Omega(a, b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{if } a = b \end{cases} \quad (4)$$

$MCCI$ is the machine change cost index

Tool Change Cost (TCC): Cost of changing a tool between operations.

$$TCC = TCCI \times \text{Total Number of Tool change within the same machine} \quad (5)$$

$TCCI$ is the tool change cost index

Setup Change Cost (SCC): Cost of changing a setup between operations.

$$TDCC = TDCCI \times \text{Total Number of TAD change within the same machine} \quad (6)$$

$TDCCI$ is the TAD change cost index

Subject To:

Precedence Constraint for Operations and Operation Clusters:

$$OCP(oc_i, oc_j) = 0 \quad \forall i > j \quad (7)$$

$$OPP(op_x, op_y) = 0 \quad \forall x > y \quad (8)$$

Operation Clusters Are Assigned Only Once:

$$OC_i \neq OC_j, \quad \forall i \neq j \quad (9)$$

Operation should only be assigned once

$$OP_x \neq OP_y, \quad \forall x \neq y \quad (10)$$

Machine Capabilities:

$$MCAP_i(od_i, m_i) = 1 \quad \forall i \quad (11)$$

Operations with Tolerance Constraints are Assigned to the same OC

$$OPC(oc_i, op_x) = OPC(oc_i, op_y) \quad \forall OPP(op_x, op_y) = 2 \quad \forall i, x, y. \quad (12)$$

Operations with Logical Constraints are Assigned to the same OC

$$OPC(oc_i, op_x) = OPC(oc_i, op_y) \quad \forall OPP(op_x, op_y) = 3 \quad \forall i, x, y. \quad (13)$$

Decision Variable Domain Constraints

$$\text{Operation Cluster Sequence: } oc_i \in \{1, 2, \Lambda, NOC\} \quad \forall i = 1, 2, \Lambda, NOC \quad (14)$$

$$\text{Odd Numbers Used: } od_i \in \{1, 2, \Lambda, NOCO(oc_i)\} \quad \forall i = 1, 2, \Lambda, NOC \quad (15)$$

$$\text{Machines Sequence: } m_i \in \{1, 2, \Lambda, NM\} \quad \forall i = 1, 2, \Lambda, NOC \quad (16)$$

$$\text{Machine Configuration Sequence: } c_i \in \{1, 2, \Lambda, NMC(m_i)\} \quad \forall i = 1, 2, \Lambda, NOC \quad (17)$$

$$\text{Operation Sequence: } op_x \in \{1, 2, \Lambda, NOP\} \quad \forall x = 1, 2, \Lambda, NOP \quad (18)$$

$$\text{TAD Sequence: } td_x \in \{1, 2, \Lambda, NTAD(x)\} \quad \forall x = 1, 2, \Lambda, NOP \quad (19)$$

$$\text{Tools Used: } t_x \in \{1, 2, \Lambda, 6\} \quad \forall x = 1, 2, \Lambda, NOP \quad (20)$$

4. GENETIC ALGORITHM METHOD

4.1 Proposed Real-Coded Approach for String Representation

Discrete GAs were used in process planning, but the number of feasible alternatives varied depending on the operation sequence used. In addition, when two feasible chromosomes perform crossover the resulting string could be infeasible.

The use of continuous domain variables solves these problems as it permits dealing with varying domain sizes while maintaining equal probabilities of selecting

each alternative (Youssef & H. ElMaraghy, 2006). In addition, this facilitates the use of the proposed constraint satisfaction approach (Section 4.2) for manipulation of the generated solutions in terms of crossovers and mutations to ensure producing feasible solutions. The process plan is expressed in a domain of continuous variables ranging between 0 and 1 to guarantee the satisfaction of specified constraints.

4.2 Decoding of Variables and Constraint Satisfaction Approach

Oper Clust Seq.	0.11	0.54			0.59	0.82		
TAD Odd Used	0.36	0.74			0.93	0.34		
Machine	0.55	0.61			0.17	0.22		
Operation Seq	0.13	0.42			0.37	0.90		
Tool Used	0.71	0.56	0.32	0.85	0.32	0.52	0.19	0.5

Figure 1 – String representation of the encoded process plan

Decoding is the translation of any of the produced encoded solution strings to a full process plan. The encoded string has five sets of variables as shown in Figure 1. The size of the first 4 variables is equal to NOC and the size for the *Tool Used* string is equal to NOP . Therefore, the number of variables for any given problem is equal to $4 \times NOC + NOP$.

4.2.1 Decoding the Operation Cluster Sequence and Operation Sequence

Each variable in the *Oper Clust Seq* string determines the selected feasible sequence of OCs. The number of feasible OCs at a specific point in the sequence is obtained by checking *OCP* after omitting the OCs that have already been sequenced in the *Oper Clust Seq* string to find the number of OCs that have no preceding OC. The feasible OCs are numbered in order starting from 1. This number determines the OC to use. The value of the continuous domain is multiplied by the total number of feasible OCs, then rounded up to the nearest integer which will in turn represent the order of the OC to select in the current sequence. This method guarantees equal probability of selection for all the possible feasible OC and OP sequences and guarantees that a feasible sequence is always generated. The same approach is used for decoding the *Operation Sequence*.

4.2.2 Decoding the TAD Odd Used, Machine Used and Tool Used

The 0-1 value in the *TAD Odd Used* is multiplied by the number of possible TAD combinations for the OC corresponding to it then rounded up to the nearest integer which will in turn represent the TAD combination number used. The same approach is used for decoding the *Machine Used* and *Tool Used*.

5. CASE STUDY

The proposed GA optimization model was applied to part ANC-101, which is the CAM-I 1986 test part widely used in literature (Li *et al.* 2002, and Ong *et al.* 2002). Figure 2 shows the part, which contains 12 features. Refer to Shabaka & H. ElMaraghy (2006) for the precedence graph and TAD for each operation.

Inputs to the proposed method are divided into two types. The first type of input is that related with all the costs and part information. The values in Table 1 represent cost units. The second type of inputs are the OCs and capable machines for each TAD odd for every OC (Figure3) which are obtained from the approach proposed by Shabaka & H. ElMaraghy (2006).

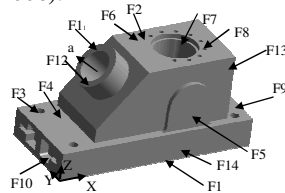


Figure 2 – Case Study ANC-101

Table 1 – Cost information used (Ong *et al.*, 2002)

ID	Type	Cost	ID	Type	Cost
M1	3-axis machine	100	C1	Drill 1	7
M2	3-axis CNC	200	C2	Drill 1	5
M3	4-axis CNC (A rotation $\pm 135^\circ$)	300	C3	Drill 1	3
M4	4-axis CNC (A rotation $+90^\circ$)	290	C4	Drill 1	8
M5	4-axis CNC (B rotation $\pm 120^\circ$)	320	C5	Tapping Tool	7
M6	5-axis CNC	450	C6	Mill 1	10
	MCCI	1000	C7	Mill 2	15
	TDCCI	120	C8	Mill 3	30
	TCCI	15	C9	Ream	15
			C10	Boring Tool	20

OC	TAD Odd				OC	TAD Odd	
	1	2	3	4		1	2
1	M1,M2,M3, M4,M5,M6				7	M5, M6	M1,M2,M3, M4,M5,M6
2	M1,M2,M3, M4,M5, M6				8	M5,M6	
3	M6	M1,M2,M3, M4,M5, M6			9	M3, M4,M6	
4	M5,M6	M1,M2,M3, M4,M5, M6			10	M5, M6	
5	M3,M6	M3,M6	M3,M4,M6	M1,M2,M3, M4,M5, M6	11	M1,M2,M3, M4,M5, M6	
6	M1,M2,M3, M4,M5, M6						

Figure 3 – Capable machines for each TAD odd for every OC

6. RESULTS AND DISCUSSIONS

A MATLAB[®] toolbox was developed for the proposed method. Figure 4 shows the convergence curves using different population sizes. The number of generations used was 150. The cross-over and mutation operators were each applied 6 and 12 times respectively per generation in this work.

The process plan with the least cost reached has a total of 5125 cost units. Figure 5 shows the corresponding optimal process plan representation. The 20 operations

are grouped into 11 clusters, therefore, there are 64 variables. The output shows that only two machines are used, M6 and M1. The number of tool changes is equal to 14 and the number of TAD change is equal to 4.

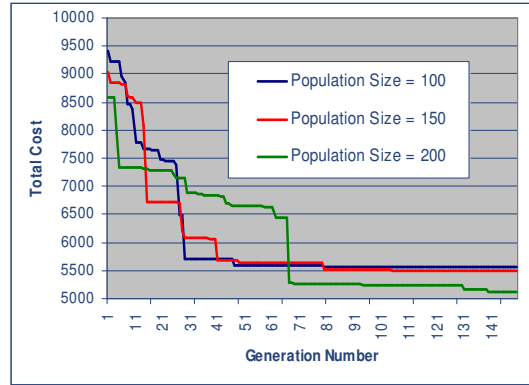


Figure 4 – Convergence Curve for different population size

Oper Clust Seq.	1	8	4	7	10		9		3	2		5		6	11					
TAD Odd Used	1	1	1	1	1		1		2	1		4		1	1					
Machine	6	6	6	6	6		6		1	1		1		1	1					
Operation Seq	1	13	4	12	18	14	15	16	17	3	2	6	5	7	8	9	10	11	19	20
TAD	+Z	+X	+X	-X	-X	-a	-a	-a	-a	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	+Z	+Z
Tool Used	6	2	6	6	6	3	9	10	7	2	7	7	7	3	9	10	1	5	9	10

Figure 5 –Representation of optimal reached process plan

The ability to guarantee that certain operations will be clustered together on the same machine is a powerful characteristic of the proposed approach because in practice for operations that have tight tolerances, as it reduces the cost of re-setting and re-fixturing and the required number of highly accurate machines. The computation time required was on average 1 min/run on a Pentium 4 2.6 GHz PC with 512 MB memory. This is a reasonable time considering the large solution space containing 64 variables with over 860 constraints.

7. CONCLUSIONS

In this paper a new process planning approach was proposed for choosing the following parameters: operation selection (Machine, Tool and TAD selection) and operation sequencing. All the parameters were considered simultaneously in the optimization model in order to achieve the lowest cost.

A novel procedure was developed and utilized for ensuring the generation of feasible process plans. It is based on mapping of the decision variables from their original discrete domain into a continuous domain of variables, which not only guarantees the generation of feasible process plans but also addresses the problem of having variable domain size. In addition, the proposed method produces solution strings that are easy to manipulate using different types of operators, such as crossovers or mutations, without violating the constraints or changing the size of the

solution string as in traditional methods. Also the proposed method guarantees that operations that have related tolerance or logical constraints are clustered together and manufactured on the same machine. A new process plan representation was developed accordingly to represent both the OC and OP strings.

A toolbox was developed using MATLAB[®]. A case study was presented to demonstrate the use of the developed model and the constraint satisfaction procedure. The tool is flexible in the sense that the tolerance and logical constraints can be relaxed to produce traditional process plan with no pre-assigned OCs while taking advantage of the continuous domain method. This method could serve as a tool in aiding the machine assignment/selection activities.

8. ACKNOWLEDGMENTS

The support from Canada Research Chairs (CRC) program and Natural Sciences and Engineering Research Council (NSERC) of Canada is acknowledged.

9. REFERENCES

1. Altung L., and Zhang, H. "Computer Aided Process Planning: the state-of-the-art survey". *Int J Prod Res*, 1989; 27(4): 553-585.
2. Ciurana, J., G.-Romeu, M.L. and Castro. R. "Optimising process planning using groups of precedence between operations based on machined volumes". *Engineering Computations*. 2003; 20(1): 67-81.
3. Dereli, T. and Filiz, I.H. "Optimisation of Process Planning Functions by Genetic Algorithms". *Computers & Industrial Engineering*, 1999; 36: 281-308.
4. ElMaraghy, H.A. "Evolution and Future Perspective of CAPP". *CIRP Annals*, 1993; 42(2):739-751.
5. ElMaraghy, H.A. and Gu, P. "Expert system for inspection planning". *CIRP Annals*, 1987; 36(1):85-89.
6. Kiritsis, D. and Prochet, M. "A generic Petri net model for dynamic process planning and sequencing optimization". *Advances in Engineering Software*. 1996; 26:61-71.
7. Li, W.D., Ong, S.K. and Nee, A.Y.C., "Hybrid Genetic Algorithm and Simulated Annealing Approach for the Optimization of Process Plans for Prismatic Parts". *Int J Prod Res*. 2002; 40(8):1899-1922.
8. Ong, S. K., Ding, J. and Nee, A.Y.C. "Hybrid GA and SA Dynamic Set-Up Planning Optimization". *International Journal of Production Research*. 2002; 40(18): 4697-4719.
9. Reddy, S.V.B., Shunmugam, M.S. and Narendran, T.T., "Operation Sequencing in CAPP Using Genetic Algorithms". *International Journal of Production Research* 1999; 37(5):1063-1074.
10. Shabaka, A.I. and ElMaraghy, H.A. "Matching Products Processing Requirements and Machine Structure Capabilities in RMS". 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Institute for Machine Tools and Industrial Management, Technical University of Munich, Garching, Germany; September 2005: 22-23.
11. Shabaka, A.I. and ElMaraghy, H.A. "Generation of Machine Configurations Based on Product Features". *International Journal of Computer Integrated Manufacturing*, 2006 (In Press)
12. Taiber, J.G. "Optimization of process sequences considering prismatic workpieces". *Advances in Engineering Software*. 1996; 25:41-50.
13. Tang, L., Yip-Hoi, D. M., Wang, W. and Koren, Y., "Concurrent Line-Balancing, Equipment Selection and Throughput Analysis for Multi-Part Optimal Line Design". *The International Journal for Manufacturing Science & Production*. 2004; 6(1-2): 71-81.
14. Youssef, A.M.A. and ElMaraghy, H.A. "Modelling and Optimization of Multiple-Aspect RMS Configurations". *International Journal of Production Research*. Special Issue on Advances in Evolutionary Computation for Design and Manufacturing Problems, 2006 (In Press).
15. Zhang, Y. F., Ma, G. H. and Nee, A. Y. C. "Modeling Process Planning Problems in an Optimization Perspective". *Proc. of the 1999 IEEE Int. Conf. on Rob. & Aut. ; 1999:1764-1769.*
16. Zhang, Y. F. and Nee, A.Y.C. "Application of Genetic Algorithms and Simulated Annealing in Process Planning Optimization". *Computational Intelligence in Manufacturing Handbook*. edited by Wang, J. and Kusiak, A., CRC Press; 2001.