



Optimisation of process planning functions by genetic algorithms

Türkay Dereli*, İ. Hüseyin Filiz

Department of Mechanical Engineering, University of Gaziantep, 27310 Gaziantep, Turkey

Abstract

This paper introduces optimisation modules of a process planning system called OPPS-PRI (Optimised Process Planning System for PRismatic parts) which has been developed together with its interfaces to provide a complete CAD/CAM integration. Primary objective of this work is to develop an intelligent and integrated CAD/CAM system for shop-floor use that can be used by an average operator and to produce globally optimised results (process plans and part programs). For this purpose, in this work, an attempt has been made to include the impact and potential of artificial intelligence (AI) in process planning applications and to optimise all events in an integrated CAD/CAM environment. GAs were extensively used in the development of process planning facilities and in the optimisation issues, in order to include profits of AI techniques into the system. © 1999 Elsevier Science Ltd. All rights reserved.

1. Introduction

Computer aided process planning (CAPP) is considered as the key technology for computer aided design and manufacturing (CAD/CAM) integration and consists of the determination of processes and parameters required to convert a block into a finished product. A huge amount of CAPP systems have been reported in the literature. However, only a few of them have intended to provide globally-optimised process plans [1]. In addition, there have not been so many researches for prismatic components as compared to those for the turning processes.

* Corresponding author. Tel.: +(342)-3601200; fax: +(342)-3601100.
E-mail address: dereli@gantep.edu.tr (T. Dereli)

This is mainly due to problems in geometrical representation of the 3D parts which have often complex shapes and also intricate nature of cutting mechanism in milling.

Artificial intelligence (AI) has the largest impact on the recent advances in CAD/CAM integration. Genetic algorithm (GA) being *one of the most popular combinatorial algorithms and AI techniques*, is a search technique for solving optimisation problems based on the mechanics of the survival of the fittest. GAs have been successfully applied to various optimisation problems, such as the travelling sales person (TSP) problem, Boolean satisfiability, space allocation, job-shop scheduling, etc. [2]. A detailed information on the use of GAs for engineering design and optimisation can be found in Ref. [3].

This paper presents the development of optimisation modules of a process planning system for prismatic parts; called OPPS-PRI (Optimised Process Planning System for PRismatic Parts). OPPS-PRI is implemented on a PC. Its goal in the first place is to integrate CAD and CAM with corresponding interfaces while taking *optimality* in each stage of planning endeavour into account. GAs are extensively utilised for optimising the process planning functions. It is mainly composed of the modules prepared for the following functions;

- interfacing CAD to CAPP: feature recognition based on STEP (STandard for the Exchange of Product modelling data) standard,
- determination and elimination of critical regions on the component,
- selection of an optimum workpiece size,
- determination of machining operations and set-up planning,
- operations sequencing and/or optimisation of sequence of operations,
- optimisation of cutting parameters (conditions),
- selection of cutting tools and auxiliary tooling,
- optimisation of index positions of cutting tools on tool magazines,
- interfacing CAPP to CAM: CNC code generation and verification.

These modules are grouped into five divisions. The first division consists of the modelling platform and feature recogniser. The second includes usual process planning works. The third involves optimisation of process planning events. The fourth division offers a utility on *design for manufacturing* (DFM). The last division is devoted to the CAPP/CAM interface.

A typical session that can be traced within the OPPS-PRI is as follows. After the component is modelled on a CAD platform and the STEP file of the component is obtained, the machining features on the component are recognised. An optimum workpiece size is selected from standard workpiece database. The type of machining operations for each feature of the component is determined correspondingly. Machining operations are collapsed into set-ups. They are first sequenced using the machinability rules. The cutting tools as well as the other auxiliary tooling are selected from respective tool libraries. The sequence of operations is optimised, based on a user-selected sequencing criterion like safety or minimum tool change. The machining parameters (speed, feed, depth of cut, and number of passes for each operation) are optimised. Critical regions between the features of the component are checked using the DFM module in order to determine whether they are machinable or not, under the specified machining conditions. If any problem exists, it is eliminated. Optimal positions of the cutting tools on the tool magazine of a selected machine tool are found. The most usable result of the

system is a part program executable on a vertical machining centre (VMC) which is generated with the use of the CAD/CAM data base prepared by up-stream modules of the system.

The OPPS-PRI has a modular structure. For each of the modules specified, a program has been written in C programming language. The modules have been combined under a supervisory program. The developed package can work either as a stand-alone system or might be integrated with other process planning systems developed for prismatic parts, since both ends of the OPPS-PRI were left 'open'. Among the modules described, in this paper, the emphasis is given to the optimisation modules involved in the third division which includes the optimisation of; (a) *sequence of operations*, (b) *index positions* and (c) *cutting parameters*. These modules mainly attempt to enhance the performance of the optimisation issues of a process planning system. The methodologies used in the development of the optimisation modules are discussed in the following sections. The rest of this paper is organised as follows. Section 2 introduces the state-of-the art of the optimisation work and procedures in process planning. Genetic operators are explained briefly in Section 3. Sections 4–6 present the GA-based systems developed for finding optimal *sequence of operations*, *index positions* and *cutting parameters*, respectively, while demonstrating their methodologies with practical examples. Discussion and conclusions are left to Section 7.

2. State of the art

Optimisation of corporate activities in *computer integrated manufacturing* (CIM) and process planning is one of the foremost targets of *intelligent manufacturing systems* (IMSS), since it is believed that only those industries capable of making effective productions would withstand international competition in the next millennium.

Determination of the optimal cutting parameters is considered as an indispensable stage in process planning. The effective optimisation of these parameters affects dramatically the cost and production time of machined components. Although the importance of using optimal cutting parameters was identified in the early 1900s, the advance in the development optimisation strategies has been very slow, since the problem is too complex due to the *nonlinear dependence* of machining variables. Therefore, the literature in the domain of the optimisation of machining operations has not been so inclusive. It has been also recognised that the progress in developing constrained optimisation systems for milling operations has been even slower than for turning operations [4].

Use of many methods has been reported in the literature to solve the optimisation problem for cutting parameters. These methods include the use of nomograms, graphical methods, linear programming, geometric programming, dynamic programming, search procedures, feasible directions, and AI [5]. Computer aided mathematical programming techniques and numerical search techniques were generally used in the past. AI-based optimisation techniques have come into view recently. Most of the works using AI have been carried out in the last four or five years.

Direct search methods include function evaluations and comparisons only. Gradient methods need values of function and its derivatives, and their computerisation are also problematic. They are more difficult than the direct search methods, but they can yield more accurate

solutions for the same computational effort. Derivative-based mathematical optimisation techniques are actually not manageable for optimising functions of discrete variables. Dynamic programming which may be applied to problems whose solution involve a multistage decision process, can handle both continuous and discrete variables. Contrary to many other optimisation methods, it can yield a global optimum solution. However, if the optimisation problem involves a large number of independent parameters with a wide range of values (*as in the case of optimisation of cutting parameters*), the use of dynamic programming is limited [6]. As the number of variables and constraints increases, the optimum has a tendency to grow flatter with less probability that the realisable optimum will be a mathematical optimum, and hence, computational effort increases considerably. Geometric programming is a useful method that can be used for solving nonlinear problems subject to nonlinear constraints, especially if the objective function to be optimised is a posynomial with fractional and negative exponents, while the constraints may be incorporated in the solution techniques. It is more powerful than other mathematical optimisation techniques when the problem is restricted by one or two constraints [7]. However, if the degree of difficulty increases, the formulated problem might be more complicated than the original problem. Geometric programming can only handle continuous variables [8]. Somlo and Nagy [7] pointed out that as the number of constraints increases in large-scale problems, other optimisation techniques should be employed in conjunction with the geometric programming. Sönmez et al. [4] have recently developed a system for the constrained optimisation of cutting parameters to be used in the multipass plain and face milling operations using dynamic programming and geometric programming. They reproached about the long execution times needed for good scores of the objective function.

The solution to the optimisation problems which include real-valued variables can be obtained using numerous methods. However, each method has its own profits and hindrances. There is no efficient all-purpose optimisation method available for nonlinear programming problems. The computational time and cost involved in the determination of the optimal parameters commonly depend on the complexity or simplicity of the model. Some models can produce accurate solutions by making rigorous computation which is not economic in terms of the computational time and cost. Sometimes, the solutions from these models may not be optimal. Some other models may develop solutions far from the optimum in a fast manner. Therefore, a compromise between the high accuracy of a rigorous solution and low accuracy of an oversimplified solution should be made [9]. This middle course may be achieved using GAs which are easy to implement and also powerful to search large solution spaces.

The optimisation problem for sequence of operations is similar to the optimisation problem for index positions of cutting tools to be used on the tool magazines of CNC machine tools. Use of numerous strategies has been notified for determining an optimal sequence of operations. These techniques include the use of integer programming, branch&bound, dynamic programming and evolutionary techniques [10]. Solution spaces to be considered in these optimisation problems are very large, since there are many possible alternative solutions, although the solution space is reduced by the use of feasibility constraints. It is too difficult to search effectively such large spaces using dedicated search strategies. Consideration of all applicable constraints results in difficulties in the formulation and solution of the problem. Therefore, evolutionary search techniques *which often require less effort to search the large solution spaces* are generally preferred [10].

GA is a search strategy ideally suited to parallel computing and most effectively applied to problems in which small changes result in *very nonlinear behaviour* in the solution space [11]. GAs are able to search very large solution spaces efficiently by providing a concise computational cost, since they use probabilistic transition rules instead of deterministic ones. They are easy to implement and increasingly used to solve inherently intractable problems called NP-hard problems. The optimising routines to handle NP-hard problems increase quickly with increasing problem size. Therefore, more emphasis is given on the development of heuristic procedures which usually do not claim for reaching either a local or global optimum and on obtaining near optimal solutions within a reasonable computation time. This results in the restriction of the search space in some way, leaving some parts totally untouched. Although GAs are heuristic procedures themselves, they test a wealth of samplings from different regions of the search space for fitness simultaneously, and sort out and exploit regions of interest very quickly [12]. It has been proved that the TSP problem *which can be referred to either a combinatorial optimisation problem or a NP-complete problem* cannot be solved by deterministic algorithms within an acceptable time, since it has numerous local minima. Some traditional optimisation methods like exhaustive search method, greedy method, and dynamic programming, have been applied to this problem. They were either too time-consuming or too difficult to find an acceptable solution. GAs are well suited to solving complicated and multi-variable optimisation problems [13].

Although *simulated annealing (SA)*, *tabu search (TS)* and GAs are originally developed for the combinatorial optimisation problems, they have been also used with success in numerical optimisation problems as well [14]. The detailed discussion on the consideration of GAs as valid approaches to numerical optimisation and the reasons can be found in Refs. [15,16].

3. Genetic operators

In GA terminology, a candidate solution is represented by a sequence of numbers and/or characters known as a *chromosome* or string. Each element in the string is called a *gene* and represents a process variable. A selected number of strings is called a *population* and the population at a given time is a *generation*. A typical GA is composed of several genetic operators such as *crossover*, *inversion* and *mutation*. There are also other types of genetic operators that yield good results. Genetic operators operate on the *genes* to replace their place within the chromosome. In the following examples, a gene is abbreviated by ‘G’ in the chromosomes.

Simple crossover involves two parents and crossover points are selected randomly. If two parents to be used for generating new chromosomes are; {Parent 1: G1-G2-G3-G4-G5} and {Parent 2: G5-G3-G1-G4-G2} and a crossover point was chosen randomly as 2; this produces the following children: {Child 1: G1-G2-|G1-G4-G2} and {Child 2: G5-G3-|G3-G4-G5}. From the example above, it is obvious that using simple one-point crossover produces undesirable results, and therefore, a modified crossover operator was used, referred to as PMX (*partially matched crossover*) [11] or sometimes LOX (*linear order crossover*) [17]. PMX is actually a method of reproduction that arose to deal with TSP problem. Under PMX two parents are randomly picked from the population, and two crossover points are randomly chosen. These

two points define where the crossover is to take place. The genes between the crossover points are replaced between the parents and the children are generated. The example below illustrates how PMX operator works. If the same parents (Parent 1 and Parent 2) are used for generating new chromosomes with PMX or LOX and two crossover points were chosen randomly as 2 and 4; this produces the following children: {Child 3: G1-G2-|G1-G4-|G5} and {Child 4: G5-G3-|G3-G4-|G2}. These intermediate children are not valid, since some of the genes appears more than once and others do not appear at all. To eliminate this problem, the children go through a verification process that produces valid chromosomes from the invalid children, making sure that the genes between the crossover points are not changed and each gene appears once and only once in a chromosome. The final result is: {Child 3: G3-G2-|G1-G4-|G5} and {Child 4: G5-G1-|G3-G4-|G2}.

Inversion operates on a single parent. It reverses the order of the element between two randomly chosen points in the parent: {Parent 5: G1-|G2-G3-|G4-G5}. Assuming that the two random inversion points are 1 and 3, the child generated by the inversion operator on the parent is: {Child 5: G3-G2-G1-G4-G5}.

Mutation operation involves a single parent. An index into the parent is randomly picked, and the gene at that position becomes the first gene in the new chromosome. From this picked position on, the parent is wrapped around to produce the child. This operation keeps some of the parent characteristics. If the parent is: {Parent: G1-G2-|G3-G4-G5}, pick position is 2; this operator produces: {Child: G3-G4-G5-G1-G2}.

4. Optimisation of sequence of operations

This section describes the strategy behind an optimisation system developed for determining optimal sequence of machining operations based on either *minimum tool change* (MTC) or *minimum tool travelling distance* (MTTD) or *safety* (based on either geometric constraints or strength). Combinations of these criteria might also be used. Among the three alternative objective functions, GA-based optimisation system gives best response to the safety criterion.

Input to the optimisation system includes an explicit CAD data base for each component. This data base is obtained from the feature recognition module of the OPSS-PRI. Features and their accompanying operations (determined in the operation selection module) are sequenced for each set-up (determined in the set-up planning module). The datum of the features are described in 2D. A *reward/penalty matrix* called REPMAX for each set-up is automatically prepared by a *reward/penalty generator* according to the selected criterion by using the corresponding rules structured in the system. Feature sequencing is performed using a GA to find an optimal sequence which is the one that has the least total penalty or largest total reward. It takes the list of features and then generates an initial population of sequences. Strings from *branch&bound* algorithm are also fed into the initial population to enhance the production of good sequences. Genetic operators are used in the generation of the new sequences.

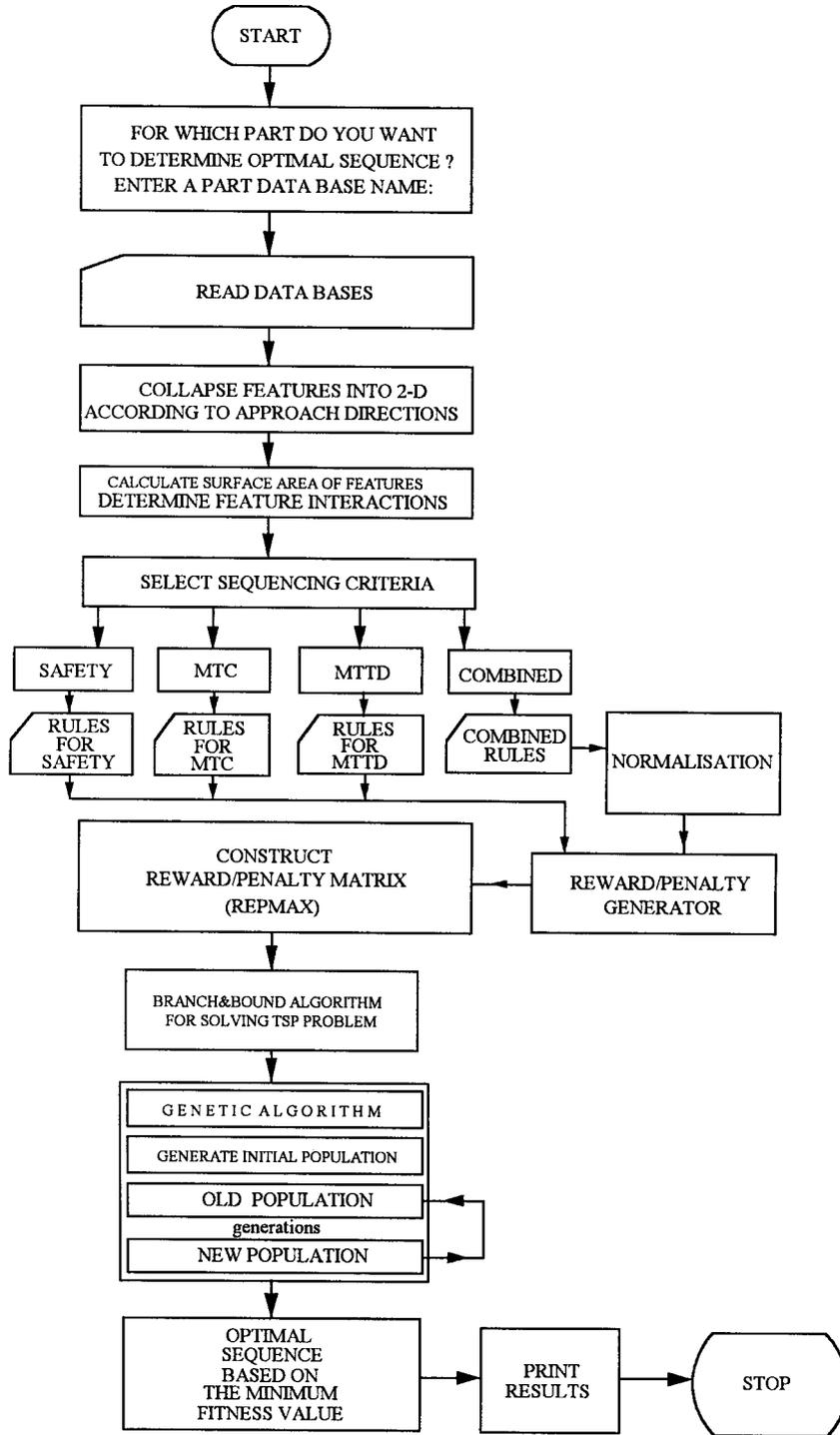


Fig. 1. Flowchart of the GA-based optimisation system for sequence of operations.

4.1. Proposed approach

The approach we adopt is based on a GA in which the initial population is fed with closed-end alternative solutions obtained by *branch&bound* methodology. Simplified flowchart of developed system is given in Fig. 1. The selected tools for the machining operations can be accessed from the tooling module of OPPS-PRI. The features are listed as collapsed according to approach directions (set-ups) into 2D, in the CAD data base. The physical interactions as well as relationships and associations between features are then determined. Intersections or interfeature relations are considered when two or more feature volumes that physically interact or overlap with each other, whereas an association can be considered when there are two features of the same type or similar surface area on the part [11]. The next stage is the selection of criterion to be used in the optimisation. For each type of criterion, a *rule base* is prepared. Some of the rules for the safety criterion included within the system are as follows;

- Rule 1: if ⟨two features partially intersect⟩, then ⟨machine the feature with smaller surface area first⟩
- Rule 2: if ⟨two features are nested inside one another⟩, then ⟨machine the top-most feature first⟩
- Rule 3: if ⟨there is an edge cut on the part⟩, then ⟨it should always be machined last⟩

Construction of the REPMAX based on the rules of selected criterion is an important stage of the methodology. Each rule has a penalty or reward (negative penalty). The REPMAX for each set-up is automatically prepared by a reward/penalty generator. The penalty scale used in the preparation of REPMAX is as follows; if precedence of operations such as *i* and *j* satisfies the sequencing rules, according to degree of satisfaction of rules, a reward is given. If precedence of operations does not satisfy the sequencing rules, according to degree of non-satisfaction of the rules, a positive penalty is given. For example, an edge cut must normally be

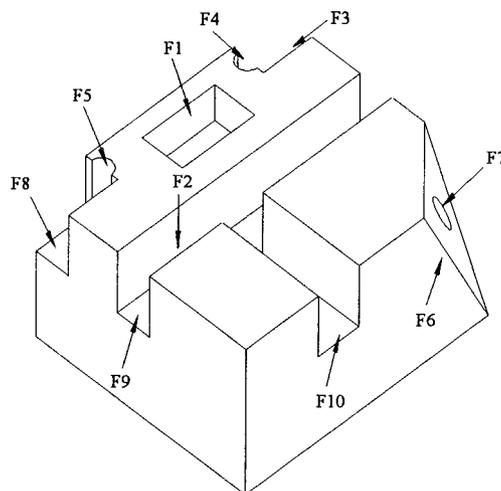


Fig. 2. Isometric view of sample part.

machined last due to safety criteria. In a precedence relationship of two features, if an edge cut is to be machined first, its penalty would be a highly positive penalty in the REPMAX. At this stage, *branch&bound* algorithm takes the REPMAX as the input and gives *closed-end* solutions (strings of sequence of features in which the relationship between the last and first operation is also considered) to be used in the initial population of GA together with other randomly generated strings.

Number of strings from *branch&bound* algorithm is equal to total number of features on the component. Other members of initial population of candidate sequences is generated by a random number generator which is initialised with the *randseed* parameter. Total number of strings in the initial population is taken as '200' in this study. The random number generator is also used to select the mutation points that are used to generate the initial population. New sequences can be generated by using the genetic operators in different combination of generation cycles. A *fitness function* is then used to evaluate the goodness of each sequence in terms of penalties or rewards specified in the REPMAX between the feature pairs.

An example part is illustrated in Fig. 2. CAD data base of the part is given in Table 1. The CAD data base includes the type and label of features, dimensions (width, radius, length and depth) of features and the distances *dx* and *dy* (in *x*- and *y*-axis) between the main datum on the body which is always at the lower-left corner and individual datum of the features.

The datum for hole features are located on their centre coordinates, while they are located on the lower left corner of the pockets and slots. For step features, they are considered to be on the inner corner of the feature. Fig. 3 illustrates how to measure the distances (*dx* and *dy*) between a hole and a rectangular pocket and the main datum in *x*- and *y*-axes.

In this example, *safety* is selected as the sequencing criterion. Consider the precedence relationship between F6 (E-CUT) and F7 (B-HOL). The sequence from 7 to 6 violates the rule-3; '*edge cut would always be machined last*'. So such a sequence is not preferable and it must be punished by a highly positive penalty in the REPMAX. As can be seen in Table 2, the relation

Table 1
CAD data base of the sample part shown in Fig. 2

Feature no.	Feature type	<i>dx</i> (mm)	<i>dy</i> (mm)	Width or radius (mm)	Length (mm)	Depth (mm)	Projected area, (mm) ²	Volume (mm) ³
F1	R-PKT (rectangular pocket)	50	90	45	30	20	1350	27,000
F2	T-HOL (thru hole)	35	70	7.5	–	25	176.71	4418
F3	B-STP (blind step)	110	110	20	30	25	600	15,000
F4	T-HOL (thru hole)	110	120	6	–	35	113.1	3960
F5	T-HOL (thru hole)	30	120	6	–	35	113.1	3960
F6	E-CUT (edge cut)	110	0	40	40	30	1600	48,000
F7	B-HOL (blind hole)	125	15	5	–	25	78.54	1960
F8	B-STP (blind step)	30	110	20	30	25	600	15,000
F9	T-SLT (thru slot)	0	60	20	140	25	2800	70,000
F10	B-SLT (blind slot)	45	0	35	60	25	2100	52,500

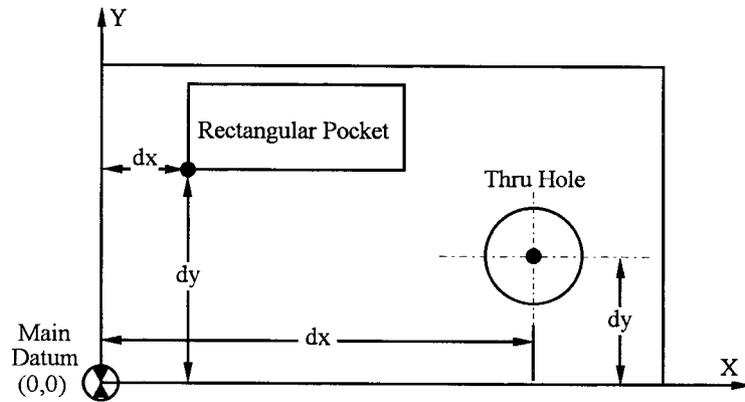


Fig. 3. Distances between datum of the features and the main datum.

between 6 to 7 (not 7 to 6) that is REPMAX (6,7), is described by a positive penalty (105). Other relations, and their penalties prepared by the reward/penalty generator are also shown in Table 2. Notice that REPMAX (7,6) is equal to -95 which promotes the sequences in which F7 is before F6. Penalties of no relations are taken as 5 due to nature of the algorithm. In an anticipated optimal sequence, F5 must be machined before F8, F4 must be machined before F3, F2 must be machined before F9, F7 must be machined before F6, for satisfying the safety rules.

The strings of sequences for the example part found by the *branch&bound* algorithm upon inputting the REPMAX is given in Table 3. The next stage is the execution of GA. The first step is to randomly generate an initial population of sequences. The results obtained by the *branch&bound* algorithm as shown in Table 3 are also included in the initial population. A section of the initial population is shown in Table 4.

New chromosomes (children) are then generated from the initial population (parents) by using the PMX operator. The children obtained from the parents are shown in Table 5.

Table 2

Reward/penalty matrix (REPMAX) of the sample part shown in Fig. 2

FN	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	∞	5	35	25	25	5	5	35	45	5
F2	5	∞	5	5	5	5	5	5	-45	5
F3	-25	5	∞	85	5	5	5	5	5	5
F4	-15	5	-75	∞	5	5	5	5	5	5
F5	-15	5	5	5	∞	5	5	-75	5	5
F6	5	5	5	5	5	∞	105	5	5	5
F7	5	5	5	5	5	-95	∞	5	5	5
F8	-25	5	5	5	85	5	5	∞	5	5
F9	-20	55	5	5	5	5	5	5	∞	-15
F10	5	5	5	5	5	5	5	5	25	∞

Table 3

The sequences from branch&bound algorithm for the sample part shown in Fig. 2

Sequence of features	Closed-end fitness value	Open-end fitness value
7-6-4-3-5-8-1-2-9-10	-310	-315
5-8-1-2-9-10-7-6-4-3	-310	-315
4-3-5-8-1-2-9-10-7-6	-310	-315
2-9-10-7-6-4-3-5-8-1	-310	-315
8-1-2-9-10-7-6-4-3-5	-310	-235
9-10-7-6-4-3-5-8-1-2	-310	-265
10-7-6-4-3-5-8-1-2-9	-310	-295
6-4-3-5-8-1-2-9-10-7	-310	-215
3-5-8-1-2-9-10-7-6-4	-310	-235
1-2-9-10-7-6-4-3-5-8	-310	-285

PMX operator requires two crossover points (shown by “*”s in Table 4) which are randomly chosen.

Generation of child chromosomes 201 and 202 from the parent chromosomes 1 and 2 is as follows. Two random crossover points (i.e. 9 and 10) are picked along the sequence (by “*”). The *cutting tools (genes)* at the crossover points are switched between the parents and then

Table 4

Randomly generated initial population of sequences

Parents—chromosome no.	Sequence of features
1	1-2-5-3-10-7-6-4-9-*8*
2	10-3-9-1-2-8-4-5-6-*7*
3	8-9-*5-6*-1-3-10-7-2-4
4	5-3-*8-6*-2-7-10-1-4-9
Strings from the branch&bound algorithm	7-6-4-3-5-8-1-2-9-10
	5-8-1-2-9-10-7-6-4-3
	4-3-5-8-1-2-9-10-7-6
	2-9-10-7-6-4-3-5-8-1
	8-1-2-9-10-7-6-4-3-5
	9-10-7-6-4-3-5-8-1-2
	10-7-6-4-3-5-8-1-2-9
	6-4-3-5-8-1-2-9-10-7
	3-5-8-1-2-9-10-7-6-4
	1-2-9-10-7-6-4-3-5-8
197	5-7-6-2-*3*-4-1-9-10-8
198	6-7-8-5-*4*-1-2-3-9-10
199	4-7-*5-8-9*-2-10-6-3-1
200	8-6-*9-10-4*-1-7-3-5-2

Table 5
Sequences generated by the PMX operator

Child—chromosome no.	Sequence of features
201	1-2-5-3-10-8-6-4-9-7
202	10-3-9-1-2-7-4-5-6-8
203	5-9-8-6-1-3-10-7-2-4
204	8-3-5-6-2-7-10-1-4-9
397	5-7-6-2-4-3-1-9-10-8
398	6-7-8-5-3-1-2-4-9-10
399	5-7-9-10-4-2-8-6-3-1
400	10-6-5-8-9-1-7-3-4-2

children are generated. We have {Parent 1: 1-2-5-3-10-7-6-4-9-8} and {Parent 2: 10-3-9-1-2-8-4-5-6-7}. The uniform crossover operator produces the following children; {Child 201: 1-2-5-3-10-7-6-4-9-7} and {Child 202: 10-3-9-1-2-8-4-5-6-8}. These intermediate sequences are not valid, since some of the features appear more than once (7 in *Child 201* and 8 in *Child 202*). The children are validated and modified to produce valid sequences from the invalid children, making sure that the features at the crossover points are not changed and feature appears only once in a sequence. The final result is then: {Child 201: 1-2-5-3-10-8-6-4-9-7} and {Child 202: 10-3-9-1-2-7-4-5-6-8}. All the chromosomes in the initial population are matched *two by two*, and new population is generated using the PMX operator as described above. We have now 400 chromosomes; 200 from the initial population and 200 from the new population. At this point, an *objective function (fitness function)* is used to measure the goodness of both parents and child chromosomes in terms of penalties or rewards specified in the REPMAX of the part. The *fitness function* ($f_{\text{chromosome}}$) for each chromosome can be expressed mathematically by the following equation;

$$f_{\text{chromosome}} = \sum_{i=1, j=2}^{i=n-1, j=n} \text{REPMAX}[\text{gene}[i]][\text{gene}[j]] \quad (1)$$

where; n is the total number of genes in a chromosome or in other words is the total number of features on the component, and gene is the vectorial representation of genes in a single chromosome. For example, the fitness value of *Parent-1 that involves the operations {1-2-5-3-10-7-6-4-9-8} assigned to the sequence numbers {1-2-3-4-5-6-7-8-9-10}* is calculated as follows;

$$\begin{aligned} & \text{REPMAX}[[\text{gene}[1]],[\text{gene}[2]]] \quad (= \text{REPMAX}[1][2] = 5) \quad + \quad \text{REPMAX}[[\text{gene}[2]],[\text{gene}[3]]] \\ & (= \text{REPMAX}[2][5] = 5) \quad + \\ & \text{REPMAX}[[\text{gene}[3]],[\text{gene}[4]]] \quad (= \text{REPMAX}[5][3] = 5) \quad + \quad \text{REPMAX}[[\text{gene}[4]],[\text{gene}[5]]] \\ & (= \text{REPMAX}[3][10] = 5) \quad + \\ & \text{REPMAX}[[\text{gene}[5]],[\text{gene}[6]]] \quad (= \text{REPMAX}[10][7] = 5) \quad + \quad \text{REPMAX}[[\text{gene}[6]],[\text{gene}[7]]] \\ & (= \text{REPMAX}[7][6] = -95 \quad + \end{aligned}$$

$$\begin{aligned} & \text{REPMAX}[[\text{gene}[7]],[\text{gene}[8]]] \quad (= \text{REPMAX}[6][4]=5) \quad + \quad \text{REPMAX}[[\text{gene}[8]],[\text{gene}[9]]] \\ & (= \text{REPMAX}[4][9]=5) \quad + \\ & \text{REPMAX}[[\text{gene}[9]],[\text{gene}[10]]] \quad (= \text{REPMAX}[9][8]=5) = \text{Fitness value} \quad (= -45) \end{aligned}$$

The next step is to put all 400 chromosomes in a descending order starting from the one which has the most negative cumulative fitness. Based on the fitness values, the next generation (current population) is formed from the newly generated sequences and old population such that it includes 80% of good positioning sets and 20% of bad positioning sets among 400 chromosomes in due order. At this stage, each chromosome in the new population is mutated and reproduced randomly using the mutation and inversion operators, respectively. The random number generator is used to select the mutation and inversion points. Finally, the order of the chromosomes in the new population is re-mixed before the PMX re-operates on the genes. The iterations are continued by this way. GA seeks to find the *chromosomes* with the most negative cumulative fitness. As the execution of the genetic algorithm reaches to certain number of iterations, the better sequences with the least fitness values dominate in the population and the system eventually converges to an optimal solution. The number of iterations can be specified by the user or the system automatically stops, if the solutions cannot be improved for a cycle of generations.

For this relatively simple example, the GA finds optimal solutions which have a fitness value of -315 (more than 50 solutions) using 3 to 10 iterations. Some of these solutions have been already given by the branch&bound algorithm. As shown in Table 3, the open-end fitness value of the best four strings are also equal to (-315). However, for highly difficult components including many interacting features, the proposed GA can find more optimal sequences than branch&bound algorithm. For instance, for a particular component with highly interacting features, the REPMAX is given in Table 6. The best string of the branch&bound algorithm is {8-6-3-5-9-1-10-7-4-2} which has an open-end fitness value of -340 (see Table 7). However, the optimal sequence found by the proposed GA is; {1-10-7-4-2-9-8-6-3-5} or {1-6-3-5-9-8-10-7-4-2} that has an open end fitness value of -345. This result can be achieved by performing on the average 40 iterations completed in 4 min. It is worth noting that the total number of iterations required to reach an optimal value is fluctuating from 1 to 100, as the

Table 6
Reward/penalty matrix prepared for a highly difficult part

FN	1	2	3	4	5	6	7	8	9	10
1	∞	-30	-20	-10	-10	-10	-10	-10	-10	-10
2	50	∞	-10	-5	-5	-5	-5	-5	-10	-20
3	40	10	∞	-30	-50	-10	5	10	15	20
4	45	-45	-5	∞	4	40	-25	60	-30	-10
5	100	40	60	55	∞	5	20	-10	-20	5
6	100	70	-50	-15	40	∞	40	20	-5	10
7	60	5	30	-25	5	5	∞	-15	30	20
8	90	30	-5	10	-5	-35	80	∞	15	-25
9	-75	10	-5	5	-40	40	20	-90	∞	-5
10	80	30	-20	45	75	25	-30	30	35	∞

Table 7

The sequences from branch&bound algorithm based on Table 6

Sequence of features	Closed-end fitness value	Open-end fitness value
9-1-10-7-4-2-8-6-3-5	-345	-325
6-3-5-9-1-10-7-4-2-8	-345	-310
10-7-4-2-8-6-3-5-9-1	-345	-335
3-5-9-1-10-7-4-2-8-6	-345	-295
4-2-8-6-3-5-9-1-10-7	-345	-320
7-4-2-8-6-3-5-9-1-10	-345	-315
8-6-3-5-9-1-10-7-4-2	-345	-340
1-10-7-4-2-8-6-3-5-9	-345	-270
5-9-1-10-7-4-2-8-6-3	-345	-295
2-8-6-3-5-9-1-10-7-4	-345	-300

execution time varies from 1 to 10 min, since the iteration process is based on a random number generator.

5. Optimisation of tool index positions (tool indexing time)

Determination of optimal positions of cutting tools on the *automatic tool changer* (ATC) or turret magazine of a CNC machine tool is an important task for reducing total non-machining time since profits are only generated when the machine is *cutting* and for achievement of optimal process plans. The problem can actually be considered as the minimisation of the total indexing time. *Indexing* can be broadly described as the process of automatic tool positioning

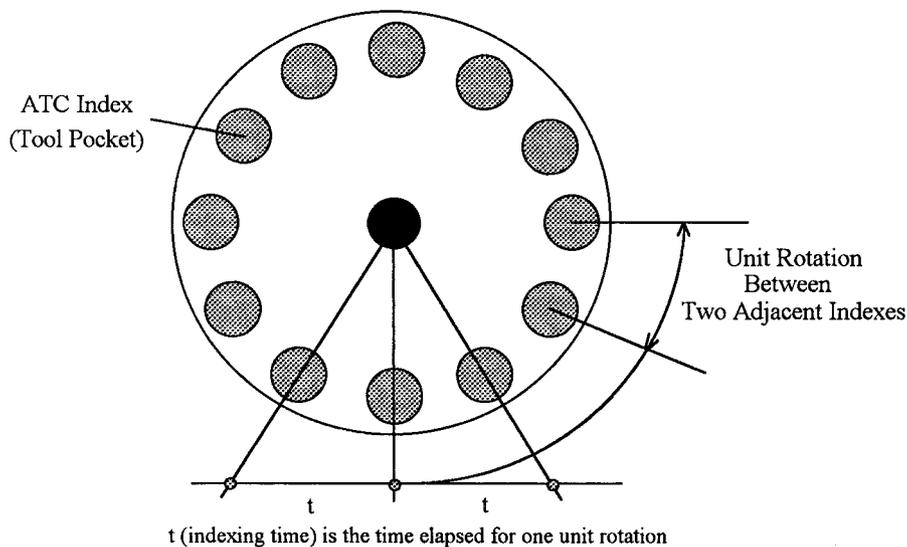


Fig. 4. ATC magazine and indexing time.

and/or changing on the ATC or turret magazine of CNC machine tools, when the cutting tools are called within the part program. However, its definition depends on the type of apparatus (such as disk, turret, drum, or chain types) used for the tool changing or indexing, as turrets are used on CNC lathes and turning centres, and ATCs are used on CNC milling machines and milling centres. Chain type tool magazines are generally used in machining centres. *ATC indexing time (from tool to tool or from pocket to next pocket)* can be defined as the time elapsed in which an ATC magazine can move between two neighbouring stations, pockets or tools as illustrated in Fig. 4. The indexing time is sometimes called the *magazine rotating speed* in machine tool catalogues. Especially for those machines that cannot provide a fast tool-indexing capability, it is extremely important to decrease the total tool-indexing time which directly affects total non-cutting time. Although machine tool manufacturers have recently equipped their machines with superior and faster turrets and ATCs, tool-indexing time still can be reduced by applying an effective *tool arrangement policy (or index allocation policy)* on the tool magazines in order to increase the time in cut [18].

In this work, a GA-based optimisation system has been developed for allocating the optimal index positions on the tool magazine to the specified cutting tools. Position selection is performed using a GA which leads to the least total tool-indexing time. It takes a list of cutting tools characterised with certain numbers assigned to machining operations, together with the total number of positions available on the ATC magazine, and the catalogue value of the indexing time specified in the manuals of CNC machine tools, as the input. The type of ATC or turret magazine such that whether it has a *uni-directional* or *bi-directional* tool indexing capability, is also considered. The methodology used in the optimisation of ATC indexing time is similar to that used in the optimisation of sequence of operations based on a GA which is presented in the previous section. The only difference between the two is the use of different objective functions (fitness functions) to be minimised. Instead of the objective function given by Eq. (1) which is used to minimise the total rewards/penalties for a given chromosome (set of operations), a simple objective function is used in this case in order to minimise the *total indexing time* for a given chromosome (set of cutting tools). The value of the objective function can easily be calculated by multiplying the total number of *unit rotations*

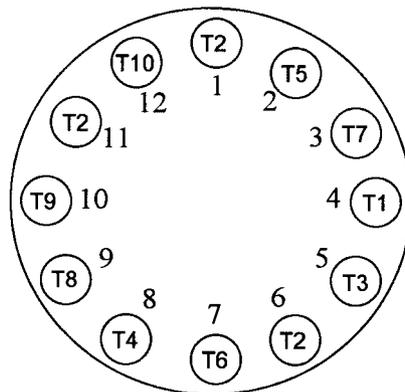


Fig. 5. A typical tool arrangement on a 12-station ATC.

Table 8

A chromosome equivalent to the representation given in Fig. 5

ATC index positions	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Chromosome of cutting tools	T2	T5	T7	T1	T3	T2	T6	T4	T8	T9	T2	T10

between the indexes of the tool magazine with the catalogue value of *ATC indexing time*. The cutting tools are represented as the genes in the chromosomes. Fig. 5 shows a typical arrangement of cutting tools on a 12-station ATC. The representation of this arrangement in the GA as a typical chromosome of cutting tools is given in Table 8.

New chromosomes of cutting tools are generated from the initial population (parents) by using the genetic operators. Fitness values of the chromosomes are then calculated, and based on the fitness values; the next generation is formed from the newly generated chromosomes of cutting tools and the old population. As the iterations of the GA continues, the better tooling arrangements (chromosomes) with lower total indexing time dominate and the system finally converges to an optimal positional set of cutting tools.

In this work, the *index allocation problem* is handled using *three phases* in terms of the relation between the total number of cutting tools employed and the total number of available index positions on the ATC magazine of the machine tool to be used;

- Phase 1—the number of cutting tools is *equal to* the number of index positions.
- Phase 2—the number of cutting tools is *smaller than* the number of index positions,
 - (a) without duplicated tools or,
 - (b) with duplicated tools.
- Phase 3—the number of cutting tools is *higher than* the number of index positions.

The overall aim is to minimise the total manufacturing cost by reducing the tool operating or tooling cost with the use of different tool indexing policies like loading *duplicate tools* on the tool magazines. If the problem falls into Phase 1, there is no need to duplicate the cutting tools in the tooling set to avoid the second ATC set-up which increases the total non-machining time considerably. If the total number of the cutting tools that are assigned for fully machining a component, are smaller than the total number of available index positions on the ATC of a machine tool (Phase 2), then the effect of the duplicated tools on a possible decrease in the tool indexing time should be tested. For example, certain cutting tools can be duplicated on the ATC, so in the *chromosome of cutting tools in GA* as well. In case Phase 3, the problem is somewhat different, so it changes to selecting the cutting tools to be used (shifted) in the second set-up. It should also be noticed that there may be other *sub-phases* between the three tool set-up phases specified above. For instance, for Phase 2(b) where the duplicated tools are used in such a way that no unloaded index is left on the ATC. However, there is another case where the optimal arrangement of cutting tools may require an ATC organisation in which an index (or more than one index) is left unloaded.

A machine tool data base is prepared by using the manuals of several CNC machine tools and integrated to the system. When the user selects the machine tool, parameters like tool capacity of the ATC, type of ATC movement and standard indexing time are captured and fed into the optimisation software. When the system is executed in the OPSS-PRI, all necessary

information regarding to machining operations and their associated cutting tools are prepared by the operation selection/sequencing module and tool selection module, respectively. For ease of managing, both the machining operations and corresponding cutting tools are characterised by numbers. Their accompanying labels or specifications are also stored in the memory.

5.1. An example

An example optimal sequence of operations together with the cutting tools selected for each operation *which are found in the upstream modules of the OPPS-PRI* is given in Table 9. The other inputs are; tool capacity of ATC (= 10 tools), type of ATC movement (= bi-directional) and indexing time between two adjacent tool stations on the ATC (= 0.20 s).

The problem described here falls into Phase 1 where the total number of cutting tools employed is equal to the total number of index positions available on ATC. Therefore it is not possible to duplicate any tool on the ATC. The total number of operations is found to be 15. The set-ups of cutting tools proposed by GA based optimisation software for different number of iterations are given in Table 10.

Note that GA converges rapidly to an optimal solution; 25 unit rotations of ATC or in other words a total turret indexing time of 5 ($= 25 \times 0.20$) s. The above problem is also asked to more than 10 average technicians, workers and operators. Average value of total unit rotations of ATC and indexing time obtained from this quiz is equal to 30 and 6 s, respectively. Even this is a small-size problem, the gain is 1 s per component to be produced. For a batch of 40,000 parts, the total gain is about 11.11 h. The slower the ATC (the higher value of indexing time from tool to tool), the higher the gain is. It should be noticed that rotating speed of the machine tool magazine is also important. If the size of the problem increases, the gain obtained from GA will also increase.

6. Optimisation of cutting parameters

Determination of optimal cutting parameters *including number of passes, depth of cut for each pass, speed, and feed applicable for selected cutting tools* is a vital stage in process planning, since the economy of machining operations plays an important role in increasing productivity and competitiveness. Since CNC machine tools are extensively employed in manufacturing industry, *economic machining* has gained a great importance. As everyone accepts, CNC machine tools have eliminated the auxiliary tooling and reduced the set-up times considerably. However, it is not possible to run the CNC machine tools effectively and economically without using *optimised machining parameters*.

Table 9
Cutting tools assigned to machining operations

Operations (ordered)	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15
Cutting tools	T1	T2	T3	T4	T5	T6	T3	T6	T7	T3	T8	T9	T6	T9	T10

Table 10

Proposed solutions for index positions by GA-based optimisation system

Itr. no.	No. of iterations/ elapsed time	Positions on ATC										Fitness value: no. of unit rotations	Fitness value: total indexing time
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
1	1/6 s	T9	T8	T5	T6	T7	T3	T4	T2	T1	T10	27	5.4 s
2	2/12 s	T9	T8	T5	T6	T7	T3	T4	T2	T1	T10	27	5.4 s
3	3/17 s	T9	T8	T5	T6	T7	T3	T4	T2	T1	T10	27	5.4 s
4	6/26 s	T7	T6	T9	T10	T8	T1	T2	T4	T5	T3	26	5.2 s
5	10/35 s	T10	T8	T9	T6	T7	T3	T5	T4	T2	T1	25	5.0 s
6	15/50 s	T7	T3	T6	T8	T9	T10	T5	T4	T1	T2	25	5.0 s
7	30/80 s	T10	T9	T8	T7	T6	T3	T4	T5	T2	T1	25	5.0 s
8	100/233 s	T2	T5	T4	T3	T7	T6	T9	T8	T10	T1	25	5.0 s

Many works have been done to optimise the cutting parameters. They rely on methods which neither guarantee the optimal solutions nor provide clearly defined economic characteristics of the optimisation problem [19]. Most of them used a single constraint ‘*power*’ in their optimisation strategies while ignoring other constraints like ‘*surface finish*’ [5]. Therefore, there is still a need to develop an optimisation system for determining the optimal values of cutting parameters for milling operations. In this work, for the optimisation of multipass milling operations, a GA-based system called *Cutting Parameters Optimisation System* (CPOS) is developed. It is an integrated module of the OPPS-PRI.

CPOS was initially constructed on a methodology using the geometric programming and dynamic programming techniques. It was restricted only to face and plain milling operations. One of the disadvantages of this initial system was its low execution speed due to large empirical mathematical models involved in the algorithm and the large amount of derivatives contained in the formulation of the problem. A detailed discussion on the previous state of this system can be found in Ref. [4]. CPOS is then modified to enhance its *performance*. The modification includes the replacement of the implementation tool; using a GA instead of geometric programming which increases the processing speed.

CPOS has a multi-pass optimisation strategy incorporating several technological constraints such as power, surface finish, speed, feed limitations, etc. It makes use of two methods called *volume sectioning* and GA. Although the problem of optimisation of cutting parameters is different from the *optimisation of sequence of operations* and the *optimisation of ATC indexing time* discussed in the previous sections, the structure of the GA is similar to those prepared for the two previous events. However, in this case, two *real-valued variables* of cutting parameters, namely; *feed-rate* (f) and *cutting speed* (V), should simultaneously be controlled within their specified ranges. Therefore, in this case, *binary genetic chromosomes* are used to represent the feed-rate and cutting speed values. The ranges of two parameters are also different from each other, hence normalisation of chromosomes are necessary. A typical binary chromosome of *feed-rate* and *cutting speed* is given in Table 11.

Table 11
A typical chromosome of feed-rate and cutting speed

	(f)—feed-rate section											(V)—cutting speed section										
Positions	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10	11
Binary Rep.	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁸	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁸	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Chromosome	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Value	1024											1										
Divider	1000											10										
Normalised value	1.024 mm/tooth											0.1 m/min										

6.1. Optimisation model

Three basic cutting parameters are usually considered in the optimisation of milling operations. They are; optimal number of passes and depth of cut for each pass, cutting speed(s) and feed rate(s). Among these, *depth of cut* is the dominant parameter which is determined by the upstream modules of the OPPS-PRI like feature recognition or operation selection module. Although it is preferable to machine the features or volumes in single pass in order to reduce the cost and time of machining (if possible), this is usually unachievable. The reason for this is that the machining operations are constrained by not only the cutter but also by other constraints imposed by the machine tool (by the characteristics of the feed drive and main drive systems of machine tools) and the workpiece. This means that although the *total depth of cuts* for machining the features are determined in the upstream modules of the process planning, they may require more than one pass, due to unavailability of the cutting tools that can provide the required cutting edge or due to the low power capacity of available machine tool. It should be noticed that the *multi-pass* scheme inherently includes the single-pass optimisation combinations, as well.

6.2. Objective functions

CPOS finds optimal cutting parameters based on the user-selected objective function such as *minimum production cost* or *minimum production time*. The former minimises the unit cost (C_u) of an operation, whereas the latter minimises the unit time (T_u) required to perform an operation. Here, only the formulation of the first one is given. The unit cost for an operation can be represented by the sum of four cost terms; cost of raw material, cost of set-up, cost of machining and cost of tool changing. *Set-up cost* and *machining cost* are the sum of the corresponding labour and overhead costs. On top of labour and overhead cost, the cost of the cutting tool is added in the case of the *tool changing cost*. However, *tool changing cost* is rationalised with the machining time divided by tool life, since a cutting tool would have been replaced before the machining operation takes place. So, the unit cost for an operation can be represented by Eq. (2). Nomenclature used in Eq. (2) can be found in Table 12.

$$C_u = c_{mat} + (c_1 + c_o)t_s + (c_1 + c_o)t_m + (c_1t_{tc} + c_t + c_ot_{tc})\left(\frac{t_m}{T}\right) \tag{2}$$

Table 12
Nomenclature used in the formulation of unit cost (C_u)

Symbol	Defines	Unit
c_l, c_o	Labour cost, overhead cost	\$/min
c_{mat}, c_t	Cost of machining, raw material and a cutting tool	\$
C_u	Unit cost	\$
T	Tool life	min
t_m, t_s, t_{tc}	Machining time, set-up time, tool changing time	min

6.3. Constraints

Following constraints are considered in the optimisation of cutting parameters; machine power, surface finish, available feed rates on the machine tool, available spindle speeds on the machine tool and cutting force.

6.4. Optimisation methodology

The optimisation problem discussed above involves four variables which are; the *number of passes* and the corresponding *depth of cut*, *cutting speed* and *feed rate* for each pass required to machine a component. The values of these four variables should be determined such that combination of them will optimise the selected objective functions. The methodology proposed herein has two stages. In the first stage, tentative number of passes and depth of cut(s) to be removed are determined through a method called *volume sectioning*, while the cutting speed and feed-rate for each pass are calculated/optimised by using a GA in the second stage.

The *volume sectioning* can be considered as a multi-stage decision process in which each of single-stage optimisation problem can be stated such that the volume to be cut is divided into possible sections (depth of cuts). The decision variable in *volume sectioning* is the *depth of cut* a_j to be taken in the i -th pass, which is represented as $a(i, j)$. Total depth of cut a_t is divided into N equal sections. The minimum increment of depth of cut is therefore equal to: a_t/N . This increment should always be smaller than the maximum depth of cut and higher than the minimum depth of cut allowed for a machine tool workpiece system; a_{max} and a_{min} , respectively. The *volume sectioning* procedure applied to multipass milling operations can be summarised as follows in conjunction with Fig. 6. The total depth of cut to be machined (a_t), the minimum (a_{min}) and maximum (a_{max}) allowable depth of cuts and the number of sections (N) are the inputs to the volume sectioning procedure. Among them, the selection of a proper number of sections for the problem has the extreme importance, since higher precision, i.e., selecting a higher number, will increase the execution time, although more effective optimal values are calculated for the objective function. The value of N should be selected always by making a compromise between the execution time and precision. The thickness of each section (a_t/N) is called unit depth of cut (section). The problem is to find all the alternative possible passes that are composed of certain number of unit sections. Fig. 6 shows a pass distribution in which total depth of cut is divided into 5 sections (N) and a_{max} and a_{min} are equal to 4 sections and 1 section. Here, thickness of each section can be considered as 1 mm. The

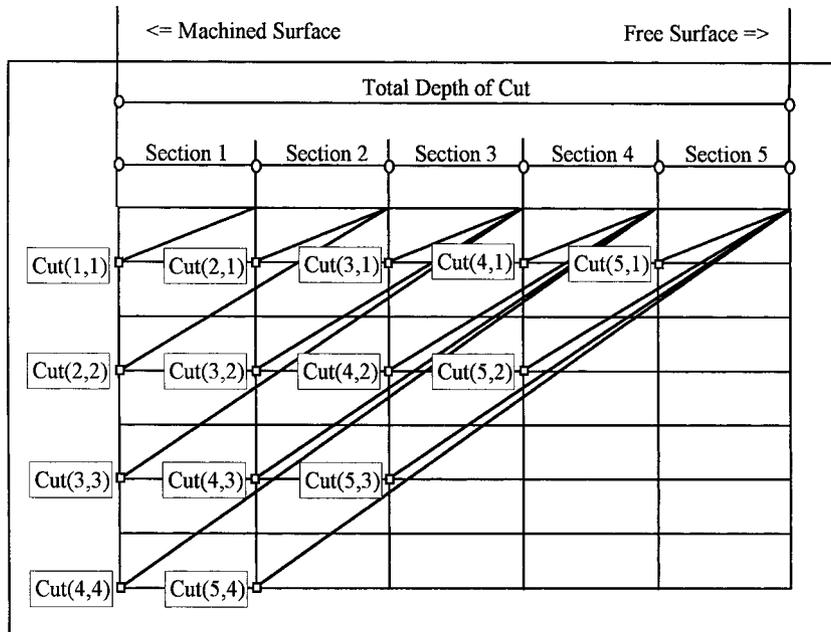


Fig. 6. Schematic representation of sectioning strategy ($N = 5$, $a_{\max} = 4$, $a_{\min} = 1$).

procedure is started from the 5th section. The stock can be machined from the outer end of the 5th section to the inner ends (right ends) of 4th, 3rd, 2nd sections, as shown in Fig. 6. Notice that a cut (5,5) including 5 sections from the outer end of the 5th section to the inner ends of the 1st section is not possible, since the maximum allowable depth of cut is 4 in terms of sections.

The same procedure is continued successively until reaching to the first section and all possible cuts are stored for future use. To reach to the 1st section from the 5th section, we have several alternative cutting strategies. These are;

- Cut(5,4) + Cut(1,1)
- Cut(5,3) + Cut(2,2)
- Cut(5,3) + Cut(2,1) + Cut(1,1)
- Cut(5,2) + Cut(3,3)
- Cut(5,2) + Cut(3,2) + Cut(1,1)
- Cut(5,2) + Cut(3,1) + Cut(2,2)
- Cut(5,2) + Cut(3,1) + Cut(2,1) + Cut(1,1)
- Cut(5,1) + Cut(4,4)
- Cut(5,1) + Cut(4,3) + Cut(1,1)
- Cut(5,1) + Cut(4,2) + Cut(2,2)
- Cut(5,1) + Cut(4,2) + Cut(2,1) + Cut(1,1)
- Cut(5,1) + Cut(4,1) + Cut(3,3)
- Cut(5,1) + Cut(4,1) + Cut(3,2) + Cut(1,1)

- Cut(5,1) + Cut(4,1) + Cut(3,1) + Cut(2,2)
- Cut(5,1) + Cut(4,1) + Cut(3,1) + Cut(2,1) + Cut(1,1)

In milling processes; for example, Cut(1,1) and Cut(2,1) are equal because workpiece is fixed while cutter turns as opposed to the turning. As a result, we have;

- Cut(5,4) = Cut(4,4)
- Cut(5,3) = Cut(4,3) = Cut(3,3)
- Cut(5,2) = Cut(4,2) = Cut(3,2) = Cut(2,2)
- Cut(5,1) = Cut(4,1) = Cut(3,1) = Cut(2,1) = Cut(1,1)

Therefore, the number of alternative cutting strategies are reduced to 6. These are given in Table 13. If a_{\max} were given equal to the total depth of cut (5 sections), a single pass strategy (Cut(5,5)) would also be considered as an alternative solution with those given in Table 13.

After the cutting strategy alternatives are determined by the *volume sectioning procedure* discussed above, the developed GA finds the optimal values of feed-rate and cutting speed which minimise the objective function for each pass. When they are found for each pass, the cutting strategy (i.e. strategies given in Table 13) that leads to minimal objective function value for multi-pass operation is selected as the optimal strategy. The number of passes, depth of cut for each pass, feed-rate and cutting speed values associated with the optimal strategy are the optimal cutting parameters to be used in machining.

In the optimisation of the feed-rate and cutting speed, GA uses the objective functions as the fitness functions to measure the goodness of the chromosomes. New chromosomes of *feed-rate* and *cutting speed* are generated from the initial population by using the genetic operators discussed in the previous sections. Fitness values (unit costs or unit times) of the chromosomes are then calculated, and based on the fitness values, the next generation is formed from the newly generated chromosomes of *feed-rate* and *cutting speed*, and the old population. As the iterations of the GA continues, the better cutting parameters that minimises the objective function based on the selected criterion dominate and the GA converges to an optimal set of cutting parameters.

Table 13
Alternative cutting strategies

Cutting strategy no.	No. of sections to be cut in each pass				
	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5
1	1	4	–	–	–
2	1	1	3	–	–
3	1	2	2	–	–
4	1	1	1	2	–
5	1	1	1	1	1
6	2	3	–	–	–

6.5. Case study

A slot as shown in Fig. 7 is to be machined on a CNC VMC. It is required to find optimal values of *feed-rate* (f) and *cutting speed* (V) based on the objective functions of *minimum production cost* and *minimum production time* and using the constraints discussed above. Specifications of the required parameters and values of the constants are given in Table 14. The information on the tooling is also provided in Table 15.

The optimum cutting parameters found by the proposed volume sectioning and GA methodology as well as the catalogue values are tabulated in Table 16. The associated objective function values are given in Table 17. The possible cutting strategies with their total objective-function results are given in Table 18. As can be seen from Table 17 and Table 18, *in which the objective function values based on optimum machining parameters found by CPOS are given in comparison with those from handbook recommendations* [20], considerable cost or time savings have been achieved with the optimal parameters in all cases.

For the *minimum production time*, the best cutting strategy is found to be (1-1-1-1-1, i.e., five passes of 1 mm) whereas for the *minimum production cost*, a single pass strategy (5 mm) results in the better improvement. It should also be noted that these improvements can further be increased by using higher number of iterations in GA cycle and loosening the feed range. For example, when allowable feed range is relaxed to (0.050–0.600 mm/tooth) and 250 iterations are used, the more effective values of feed and cutting speed can be found. However, in this case the processing time is increased considerably. The values of machining parameters and their ratings for a 5 mm depth of cut (single pass) are shown in Tables 19 and 20, respectively.

It is worth pointing out that most of the reported systems have used their own mathematical models for the optimisation of machining parameters to be used in *milling processes*. Many works have omitted some optimisation constraints like *surface finish* and considered removing of stocks using only *single-pass strategy*, or were limited to handle only simple milling operations like face milling, due to the lack of exponents, constants or empirical formulae for some other types of milling operations. Each has used a different data base of material properties, tooling data, constants and exponents, etc. Since it is difficult to find a completely similar study using different approaches in the literature, we tried to compare our results with

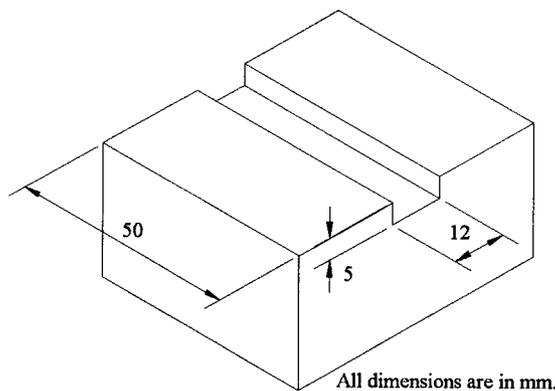


Fig. 7. A slot.

Table 14
Specifications and constants

Specification	Value
c_{mat}	0.25 \$
c_o	1.45 \$/min
c_l	0.45 \$/min
t_s	2 min
t_{tc}	0.5
Machine tool	VMC
Machine power	5.5 kW
e (efficiency)	0.95
Material type	Leaded steel
Material hardness	225 BHN
N (no. of sections)	5 mm
a_{max}	5 mm
a_{min}	1 mm

Table 15
Tooling data

Tool type	Shear yield strength (MPa)	Diameter (mm)	Cutting length (mm)	No. of teeth	Price (\$)	Shank diameter (mm)	Helix angle (°)	Lead angle (°)	Clearance angle (°)
HSS, end-mill	1000	12	40	4	10	10	45	0	5

Table 16
Handbook values [20] and optimum machining parameters found by CPOS when: feed range = 0.050–0.300 mm/tooth and no. of iterations in the GA = 50

Depth of cut (mm)	CPOS					
	Catalogue values		Based on minimum production cost		Based on minimum production time	
	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)
1	25.0	0.100	51.2	0.224	96.0	0.272
2	25.0	0.100	51.5	0.248	38.4	0.241
3	20.0	0.100	38.0	0.254	15.2	0.252
4	20.0	0.100	25.6	0.254	11.2	0.256
5	20.0	0.100	25.6	0.255	25.3	0.142

Table 17
Objective function values

Depth of cut (mm)	Catalogue		CPOS	
	C_u (\$)	T_u (min)	C_u (\$)	T_u (min)
1	0.4441	0.2337	0.0990	0.0279
2	0.4413	0.2337	0.0922	0.0632
3	0.5551	0.2921	0.1165	0.1526
4	0.5551	0.2921	0.1709	0.2038
5	0.5551	0.2921	0.1703	0.1627

Table 18
Cutting strategies for machining with their ratings

Cutting strategy (pass distribution) (mm)	Catalogue		CPOS	
	T_u (min)	C_u (\$)	T_u (min)	C_u (\$)
1-4	0.5258	0.9992	0.2317	0.2631
1-1-3	0.7595	1.4433	0.2084	0.3009
1-2-2	0.7011	1.3267	0.1543	0.2834
1-1-1-2	0.9348	1.7736	0.1469	0.3892
1-1-1-1-1	1.1685	2.2205	0.1395	0.4950
2-3	0.5258	0.9964	0.2158	0.2087
5 (single pass)	0.2921	0.5551	0.1627	0.1703

handbook recommendations [20] as shown in Tables 16–20, as it is usually performed by the previous studies. In this way, the percentage cost save and time save obtained by the developed optimisation system against the handbook values can be calculated. CPOS described in this paper has made significant improvements over the handbook values as illustrated in Tables 16–20. Many tests have been performed to check the performance of the optimisation system by using different types of workpiece materials and sizes. CPOS provided results better than

Table 19
Optimum machining parameters for 5 mm depth of cut (no. iterations = 250) when: feed range = 0.050–0.600 and no. of iterations in the GA = 250

Depth of cut (mm)	Catalogue values		CPOS			
			Minimum production cost		Minimum production time	
	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)
5	20	0.10	24.8	0.512	25.8	0.486

Table 20
Ratings of parameters shown in Table 19

	Catalogue		CPOS	
	T_u (min)	C_u (\$)	T_u (min)	C_u (\$)
5 (single pass)	0.2921	0.5551	0.0466	0.0882

handbook recommendations in all cases. The average values of percentage improvement on the *production cost* and *production time* are about 38 and 45%, respectively, and are never below 30%.

7. Discussion and conclusion

Optimisation of all process parameters is one of the important duties of the CAPP systems. Most of the optimisation systems related to process planning applications have been developed as off-line systems such that they cannot be used as integrated modules within process planning packages. Therefore, optimisation systems need to be integrated with CAPP systems. The impact of AI techniques on the optimisation of CAPP functions has been proven by many research projects. The potential and the power of AI is very great and it is believed that with the exploitation of AI methods, it will be possible to increase the capabilities of the IMSs. GA is promoted as one of the promising AI technologies to be used for solving nonlinear and combinatorial problems involved in process planning.

In this paper, the methodologies used for the development of the three GA-based systems responsible for optimisation of sequence of operations, optimisation of ATC-index positions and optimisation of cutting parameters are presented. They can be used as stand-alone systems or as integrated optimisation modules within OPPS-PRI which has been implemented on a VMC. The developed systems have been used in small and medium-sized manufacturing industries making batch production of spare parts for the textile industry in Gaziantep city. The methodologies reported in this paper can also be used for the optimisation of other process planning functions like *set-up planning* with little modifications. The values obtained by the developed optimisation modules have been tested for various components within the OPPS-PRI and positive results have been obtained.

GAs have the advantage of rapid reaching to the region which includes the global optimum due to their parallel structure. However, the most important drawback of the GA is that it is easily trapped in local optima. A mixed methodology can be used to increase the performance of the GA, by coupling the parallel computing ability of GAs with the advantages of the SA which attempts to escape local optima. The computational cost of GAs can be reduced by adopting an *artificial selection mechanism* in addition to the common natural selection mechanics. It can also be reduced by using *adaptive penalty approaches* to handle the optimisation constraints and to provide a way of evaluating how close is an infeasible solution from the feasible region within the solution space to be searched.

With GA-based optimisation systems developed in this work, it would be possible to increase machining efficiency by the use optimal cutting parameters, sequence of operations, and positioning sets on tool magazines, and to contribute to the success of the manufacturing industry. This will lead to increased utilisation of CNC machine tools. The increased use of GAs will probably enhance the performances of future process planning systems.

Acknowledgements

The authors wish to thank the referees and the editor for their suggestions which have substantially improved the manuscript. The authors are also grateful to the *Research Fund of University of Gaziantep* and (*TUBITAK*)-*Foundation of Munir Birsal* for their partial financial supports to the research programmes of ‘*Development of a Process Planning and Integrated Manufacturing System for Prismatic Parts*’ under project number MF.96.09 and ‘*Development of a Process Planning System for Prismatic Parts*’, respectively.

References

- [1] Kayacan MC, Filiz IH, Sönmez AI, Baykasoglu A, Dereli T. OPPS-ROT: an optimised process planning system for rotational parts. *Computers in Industry* 1996;32:181–95.
- [2] Hong L, Guangzhou Z, Zongkai L. A system of optimizing nesting with analogical learning mechanism. *Computers and Industrial Engineering* 1997;32(4):713–25.
- [3] Gen M, Cheng R. Genetic algorithms and engineering design. USA: John Wiley & Sons, 1997.
- [4] Sönmez AI, Baykasoglu A, Dereli T, Filiz IH. Dynamic optimization of multipass milling operations via geometric programming. *International Journal of Machine Tools and Manufacture* 1999;39(2):297–320.
- [5] Rad-Tolouei M, Bidhendi IM. On the optimization of machining parameters for milling operations. *International Journal of Machine Tools and Manufacture* 1997;37(1):1–16.
- [6] Roy S, Ghosh S, Shivpuri R. Optimal design of process variables in multi-pass wire drawing by genetic algorithms. *Journal of Manufacturing Science and Engineering* 1996;118(2):244–51.
- [7] Somlo J, Nagy J. A new approach to cutting data optimization: advances in computer aided manufacture. North-Holland Publication Co, 1977.
- [8] Humphreys KK. Jelen’s cost and optimization engineering. Inc, USA: McGraw-Hill, 1991.
- [9] Prasad AVSRK, Rao PN, Rao URK. Optimal selection of process parameters for turning operations in a CAPP system. *International Journal of Production Research* 1997;35(6):1495–522.
- [10] Usher JM, Bowden RO. The application of genetic algorithms to operation sequencing for use in computer aided process planning. *Computers and Industrial Engineering* 1996;30(4):999–1013.
- [11] Kamhawi HN, Leclair RS, Philip CL. Feature sequencing in the rapid design system using a genetic algorithm. *Journal of Intelligent Manufacturing* 1996;7:55–67.
- [12] Ulusoy G, Serifoglu SF, Bilge U. A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. In: *Proceedings of First Symposium on Intelligent Manufacturing Systems*, Sakarya, TR, 1996. p. 438–61.
- [13] Chen CJ, Tseng CS. The path and location planning of workpieces by genetic algorithms. *Journal of Intelligent Manufacturing* 1996;7:69–76.
- [14] Karaboga D. Design of fuzzy logic controllers using tabu search algorithm. In: *Proceedings of Biennial Conference of The North American Fuzzy Information Processing Society*, University of California, Berkeley, USA, 1996.

- [15] Michalewicz Z, Janikow CZ. Genetic algorithms for numerical optimization. *Statistics Computing* 1991;1:75–91.
- [16] Schaffer JD, Eshelman LJ, Offutt D. In: Rawlins G, editor. *Spurious correlations and premature convergence in genetic algorithms*. San Mateo, CA: Morgan Kaufmann, 1991. p. 102–12.
- [17] Lee CY, Choi JY. A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights. *Computers Operations Research* 1995;22(8):857–69.
- [18] Dereli T, Baykasoglu A, Gindy NNZ, Filiz IH. Determination of optimal turret-index positions of cutting tools by using genetic algorithms. In: *Proceedings of the Second International Symposium on Intelligent Manufacturing Systems*, 6–7 August, Sakarya, TR, 1998. 2. p. 743–750.
- [19] Ostafiev V, Globa A. Integrated end milling development. *Annals of the CIRP* 1984;33(1):29–32.
- [20] Hertel Co., Germany Milling Technique, Catalogue 1994.