ELSEVIER

Robotics and Autonomous Systems 33 (2000) 155–167

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Allocating optimal index positions on tool magazines using genetic algorithms

Türkay Dereli [a,*], İ. Hüseyin Filiz [b]

[a] *Department of Industrial Engineering, University of Gaziantep, 27310 Gaziantep, Turkey*
[b] *Department of Mechanical Engineering, University of Gaziantep, 27310 Gaziantep, Turkey*

Received 1 January 1999; received in revised form 1 May 1999; accepted 1 December 1999

Abstract

This paper presents an optimisation system software developed for the determination of optimal index positions of cutting tools on the automatic tool changer (ATC) or turret magazine of CNC machine tools. Position selection is performed using a genetic algorithm (GA) which takes a list of cutting tools assigned to certain machining operations together with total number of index positions available on the ATC or turret magazines and, catalogue value of indexing time as the input, and then randomly generates an initial population of position sets (chromosomes). New chromosomes are generated using genetic operators: *crossover*, *reproduction* and *mutation*. A fitness function is used to evaluate the goodness of each chromosome in terms of *total tool-indexing time*. Based on the fitness values, the next generation is formed from the newly generated sequences and old population. As the iterations are continued, the better sequences with higher fitness values (lower total-indexing times) dominate and the system converges to an optimal positioning set. The system is implemented in C programming language and on a PC. It can be used as stand-alone system or as an integrated module of a process planning system called OPPS-PRI (*O*ptimised *P*rocess *P*lanning *S*ystem for *PRI*smatic parts) developed for prismatic parts. © 2000 Published by Elsevier Science B.V.

Keywords: Automatic tool change; Indexing time; Optimisation; Process planning; Genetic algorithm

1. Introduction

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 and for achievement of optimal process plans for reducing total non-machining time, since profits are generated only when the machine is *cutting*.

Indexing can broadly be described as the process of automatic tool positioning 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. In this work, only the turret and ATC magazines are considered. Since there is no need to load or unload the cutting tools after fixing them to the turret magazine of a lathe, *turret-indexing* can only be referred to the positioning of turret and *turret-indexing time* (*from tool-to-tool* or *from face-to-face*) can be defined as the

* Corresponding author. Tel.: +90-342-360-1200/ext. 2601; fax: +90-342-360-1100.
E-mail address: dereli@gantep.edu.tr (T. Dereli).

0921-8890/00/$ – see front matter © 2000 Published by Elsevier Science B.V.
PII: S0921-8890(00)00086-5

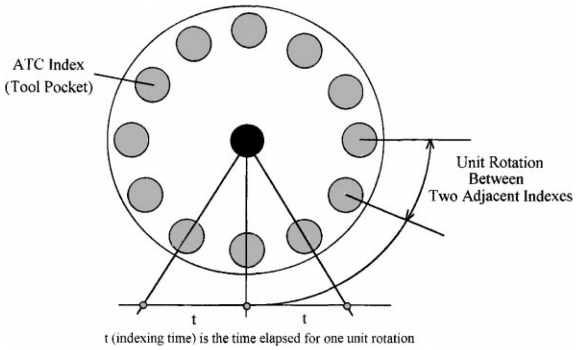


Fig. 1. Turret indexing time.

time elapsed in which a turret magazine can move between two neighbouring stations or tools as illustrated in Fig. 1. Cutters must first be carried between the ATC(s) and spindle(s) in milling. Therefore, an indexing operation on the ATC of a CNC milling machine includes three sub-processes; unloading a used cutter from the spindle, positioning the ATC magazine for a new cutting tool and loading the new cutting tool to the spindle. *ATC-indexing time* can generally be defined as the time elapsed in which all of the three sub-processes described above are performed. However, if the loading and unloading time are excluded, *ATC-indexing time* (*from tool-to-tool or from pocket to next pocket*) refers to the time elapsed in which an ATC magazine can move between two neighbouring index or tool pocket, as in the case of turret magazines.

The indexing time for both turret and ATC magazines is sometimes referred as the *magazine rotating speed* in machine tool catalogues. The typical values of ATC-indexing time for CNC milling machines are higher than those of turret-indexing time given for CNC turning machines. This is not only due to involvement of the loading and unloading in each tool change, but also due to the heavier cutters on the ATCs as compared to turrets.

HITEC Turning Centre produced by Hitachi Seiki [8] has a turret indexing time of 0.1 s for a 12-station turret. Compact Universal CNC lathe Hi-Eco10 produced by Hwacheon [2] has a higher value: 0.2 s for a 10-station turret. TAKSAN TMC-700 VMC (Vertical Machining Centre) which has a 12-station ATC provides a magazine rotating speed (time) of 0.69 s at 50 Hz [14]. They all offer a *bi-directional* indexing in which the nearest path between the index stations (current station and target station) on the magazine can automatically be determined.

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 can still be reduced by applying an effective *tool arrangement policy* (*or index allocation policy*) in order to increase the cutting time. In this work, a GA-based optimisation system is developed for allocating the optimal index positions on the tool magazine to the specified cutting tools. It can be used as a stand-alone system for both CNC turning and milling machines. It has also been integrated to a process planning system called OPPS-PRI [4] which is implemented on a VMC, which aims in the first place to integrate CAD and CAM with its interfaces, taking into account *optimality* in each stage of planning endeavour. GAs are generally used for optimising the process planning events. An overall structure of the OPPS-PRI including the system developed in this work (OTIP) is shown in Fig. 2.

A typical session in the OPPS-PRI is as follows. After the component is modelled on a CAD (computer aided design) platform and the STEP (STandard for Exchange of Product modelling data) file of the component is obtained, the machining features on the component are recognised. An optimum workpiece size is selected from standard workpiece database. Type of machining operations for each feature of the compo-

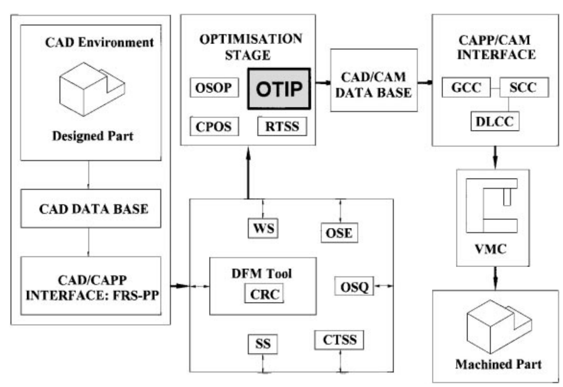


Fig. 2. Overall structure of the OPPS-PRI [4].

nent is determined correspondingly. Machining operations are collapsed into set-ups. They are 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 sequencing criteria like safety and 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 DFM (design for manufacturing) module in order to determine whether they are machinable or not, under the specified machining conditions. If any problem exists, it is eliminated. The most useable result of the system is a part program executable on a VMC which is generated with the use of the CAD/CAM database prepared by up-stream modules of the system.

Among all the modules of OPPS-PRI shown in Fig. 2, in this paper, the emphasis is given to the *optimal pocket allocation policy* on the ATC or turret magazines. The other modules are left out of the content of this paper. The whole system has been developed in C programming language. When it is executed, a pop-up/pull-down menu appears on the screen. It involves seven menus, namely; File, CAD-interface, Plan, Optimisation, DFM-tool, CAM-interface, and About, as shown in Fig. 3.

This paper has the main goal of underlining the importance of the determination of optimal index positions of cutting tools on the magazines for reducing

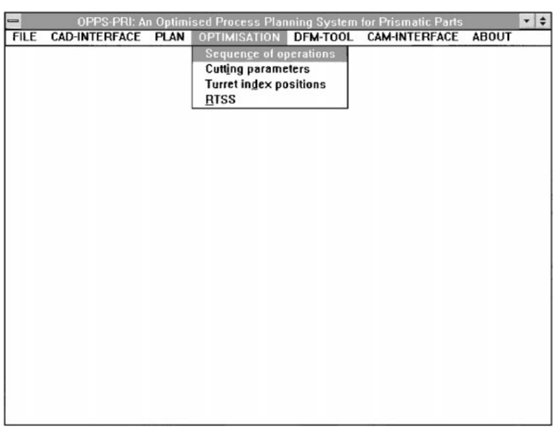


Fig. 3. Main menu of the OPPS-PRI (Optimisation is selected).

non-cutting time in manufacturing and presents an implemented methodology based on a genetic algorithm (GA). Following this section, the rest of this paper is organised as follows. A brief explanation of GAs and their operators are given in Section 2. Section 3 provides previous work using GAs. Approach to the problem of finding optimal tool position sets on the ATCs and the methodology are then presented in Section 4. The use of methodology is demonstrated with an example in Section 5. The importance of optimisation of corporate activities in CIM (computer integrated manufacturing) and use of artificial intelligence (AI) techniques for solving CAD/CAM problems are discussed and finally, conclusions are drawn in Section 6.

2. Genetic algorithms and operators

GAs are the most popular approach among combinatorial algorithms like Tabu (Taboo) search (TS), simulated annealing (SA). They are robust search algorithms based on the mechanics of natural selection and genetics. Generation of the initial population of strings is done randomly. New chromosomes can be generated using *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 in the chromosomes is abbreviated by “G”.

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 above example, 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) [7] or sometimes LOX (linear order crossover) [11]. PMX is actually a method of reproduction that arose to deal with travelling sales person (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 appear 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 1: 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 1: G1-G2-|G3-G4-G5}, pick position is 2; this operator produces: {Child 6: G3-G4-G5-G1-G2}.

3. Background

Optimisation of corporate activities in 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. 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. 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 needed [16].

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 [10]. GAs are able to search very large solution spaces efficiently by providing a lower 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 for fitness a wealth of samplings from different regions of the search space simultaneously, and sort out and exploit regions of interest very quickly [15]. 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 [1].

AI has the largest impact on the recent advances in CAD/CAM integration. GA being *one of the most popular combinatorial algorithms and AI techniques*, is a robust 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 TSP problem, space allocation, job-shop scheduling, etc. [9]. Some of the outstanding work using GAs found in the CAD/CAM literature is summarised as follows.

Kamhawi et al. [10] proposed an approach based on a genetic algorithm for feature sequencing in a rapid design system. The proposed system provides a machinist with safe and near-optimal feature sequence while considering "safety" and "tool change" as conflicting criteria. Dereli et al. [5] developed a GA based system to optimise the sequence of features which correspond to at least a single or more operations required to produce prismatic components by feeding the initial population with TSP strings to make the convergence to the optimal solution faster. Usher and Bowden [16] applied GA approach to operation sequencing for use in computer aided process planning (CAPP). Their system determines optimal or near-optimal operation sequences for parts of varying geometry. It has been shown that a GA is a viable means for searching the solution space of operation sequences providing a computational time of the order of a few seconds. Vancza and Markus [17] used a GA for operation sequencing in which each chromosome is represented by elements corresponding to feature states that are produced by machining operations. The use of genetic operators like crossover and mutation resulted in about 10% in feasible sequences as the operators violated some of the precedence constraints. Derek and Debasish [3] applied a coding strategy in GA for sequencing of parallel machining operations, where combinations of interacting work holding and tool holding devices are used. Their coding methodology allows the generation of valid operation sequences through the use of well known genetic operators such as mutation and crossover. Roy [13] developed a method based on an adaptive micro genetic algorithm (μGA) for optimal design of process variables in multi-pass wire drawing. By using the μGA, the difference between maximum and minimum effective plastic strains in the end product is minimised, so also the total deformation energy in a multi-pass wire drawing process. The μGAs are different from simple GAs in terms of the number of strings in population. Population size for GAs being currently used in the optimisation problems ranges from 30 to 200. However, it is enough to randomly generate a population of five strings in μGAs. The application of GAs to job sequencing problems was introduced by Öztemel and Düğenci [12]. Several problems are solved by both GAs and well known heuristics based methods. The results and their comparison were also presented. It has been proven that GAs are more successful than the others, especially when large numbers of jobs and machines are considered. Yokota [18] formulated an optimal weight design problem of a gear for a constrained bending strength of gear, torsional strength of shafts and each gear dimension as a nonlinear programming problem, and solved it directly by keeping the nonlinear constraint by using an improved genetic algorithm.

The inclusive information on the different applications (i.e. on engineering design and optimisation) of GAs can be found in [6,7].

4. Allocation policy and methodology

In this work, the *index allocation problem* is handled in *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 or turret 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,
(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 Phase 3, the problem is somewhat different, so it changes to selecting the tools to be used (shifted) in the second set-up. The example arrangements of cutting tools on a 12-station ATC for each phase are given in Figs. 4–7, respectively.

It should also be noticed that there might 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.

GA is an effective tool for solving the problem of determining optimal positions of cutting tools on the ATC magazines. It is very difficult to solve the problem using *Brute-force Search* algorithm, since there are many alternatives to be controlled and they require a long time. For instance, there are about 479 millions of alternative tooling set-ups that can be used on a 12-station ATC and a positioning set of 12 cutting tools. As the total number of cutting tools employed in machining and the ATC capacities increase, the benefit to be obtained from the right tool arrangement would have a greater importance. Even 1 s saved in the tool indexing time would reduce the total production time considerably, especially in high-volume production. For example, if 1 billion parts are to be produced per year, 1 s saved in the tool indexing time will cause a gain of about 12 working days.

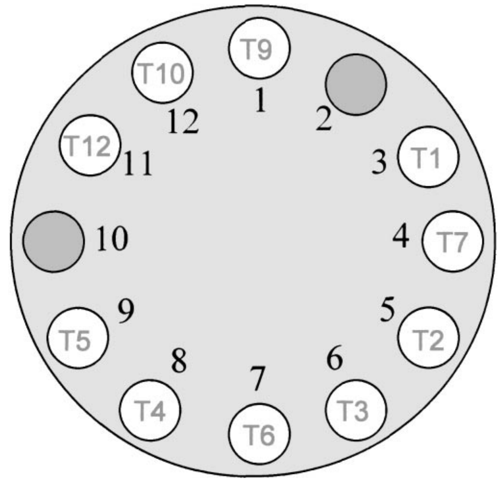


Fig. 5. ATC arrangement: number of tools < number of indexes (Phase 2(a)).

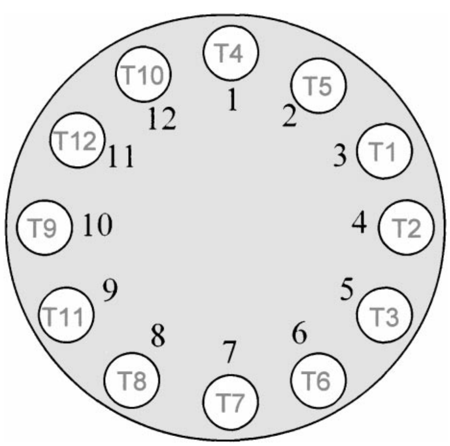


Fig. 4. ATC arrangement: number of tools = number of indexes (Phase 1).

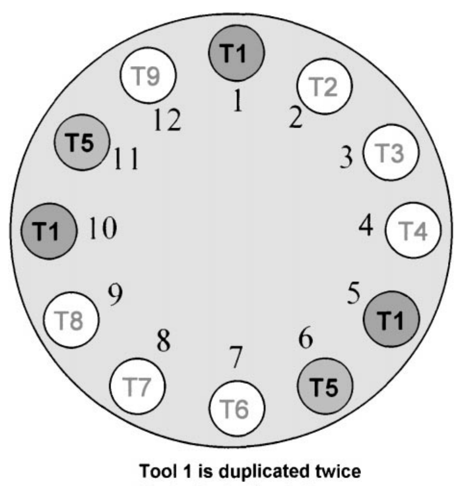


Fig. 6. ATC arrangement: number of tools < number of indexes (Phase 2(b)).

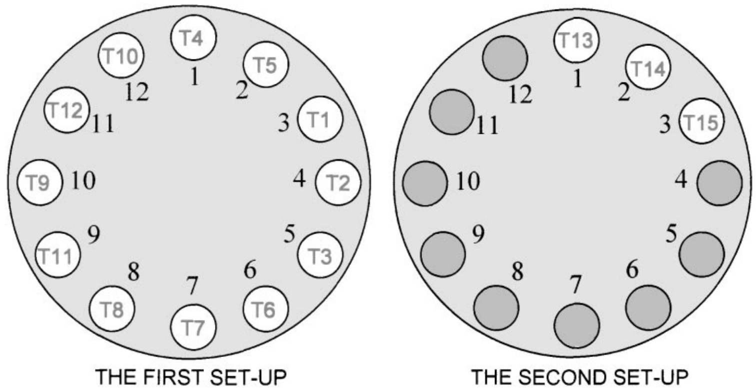


Fig. 7. ATC arrangements: number of tools > number of indexes (Phase 3).

This paper presents a methodology proposed for the determination of optimal ATC (or turret) index positions of cutting tools that are assigned to certain machining operations. Position selection is performed using a genetic algorithm that leads to the least total tool indexing time. The algorithm 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 or turret magazine, and the catalogue value of their indexing time specified in the manuals of CNC machine tools as the input. The type of ATC or turret magazine such as whether it has a *uni-directional* or *bi-directional* tool indexing capability, is also considered. The first step in the GA is to randomly generate an initial population of positioning sets (chromosomes). The cutting tools are represented as the genes in these chromosomes. Fig. 8 shows a typical arrangement of cutting tools indexed on a 12-station ATC. The representation of this arrangement in the GA, i.e. a typical chromosome of cutting tools for a 12-station ATC is given in Table 1.

Initial population of candidate position sets (chromosomes) is then generated by a random number generator, which is initialised with the randseed parameter. Total number of chromosomes in the initial population is taken as 200 in this study. A section of the initial population is shown in Table 2.

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 3.

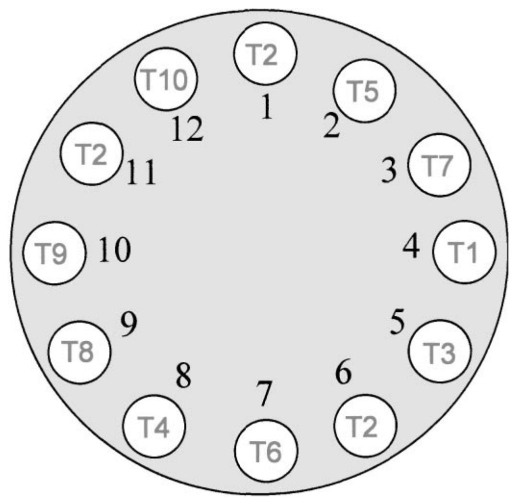


Fig. 8. A typical tool arrangement on a 12-station ATC.

Table 1
A typical chromosome of 12 cutting tools

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

Table 2
Randomly generated initial population of cutting tools

Chromosome no.	Tools at index: 1-2-3-4-5-6-7-8-9-10
Parent 1	1-2-5-3-10-7-6-4-9-*8*
Parent 2	10-3-9-1-2-8-4-5-6-*7*
Parent 3	8-9-*5-6*-1-3-10-7-2-4
Parent 4	5-3-*8-6*-2-7-10-1-4-9
⋮	⋮
Parent 197	5-7-6-2-*3*-4-1-9-10-8
Parent 198	6-7-8-5-*4*-1-2-3-9-10
Parent 199	4-7-*5-8-9*-2-10-6-3-1
Parent 200	8-6-*9-10-4*-1-7-3-5-2

Table 3
Positioning sets generated by the PMX operator

Chromosome no.	Tools at index: 1-2-3-4-5-6-7-8-9-10
Child 201	1-2-5-3-10-8-6-4-9-7
Child 202	10-3-9-1-2-7-4-5-6-8
Child 203	5-9-8-6-1-3-10-7-2-4
Child 204	8-3-5-6-2-7-10-1-4-9
⋮	⋮
Child 397	5-7-6-2-4-3-1-9-10-8
Child 398	6-7-8-5-3-1-2-4-9-10
Child 399	5-7-9-10-4-2-8-6-3-1
Child 400	10-6-5-8-9-1-7-3-4-2

PMX operator requires two crossover points (shown by asterisks in Table 2) 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 asterisks). The *cutting tools* (*genes*) at the crossover points are switched between the parents and then 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 each chromosome (set of cutting tools assigned to the certain index positions) and in order to minimise the *total ATC indexing time* for a given chromosome. Based on the operation sequence given as the input to the system, the value of the objective function for a chromosome can be calculated by multiplying the total number of unit rotations (between two adjacent index) of ATC magazine due to the order of the genes (cutting tools) in the chromosome by the indexing time (magazine rotating speed from one ATC index to next ATC index). The fitness function for each chromosome (f) can be expressed mathematically by the following equation:

$$f = \sum_{\substack{i=1,\, j=2;\\ \text{while: } i=i+1,\, j=j+1}}^{\substack{i=\#\_\text{of}\_\text{total}\_\text{operations}-1,\\ j=\#\_\text{of}\_\text{total}\_\text{operations}}} |[\text{INO}[\text{gene}[\text{opr}[i]]]] - [\text{INO}[\text{gene}[\text{opr}[j]]]]|, \tag{1}$$

where gene, opr and INO are the vectorial representations of genes (cutting tools or cutting tool numbers) in a single chromosome, associated machining operations (operation no) and index numbers available on an ATC magazine, respectively. The index numbers of each cutting tool in a chromosome are found. The differences between the index numbers of subsequent cutting tools are calculated and then totalled to find the total number of unit rotations (fitness value) for each chromosome. Absolute differences are always considered in calculating the number of unit rotations required from current tool to target tool. It should also be noticed that if the ATC has a bi-directional indexing capability, the difference between two index numbers must be calculated such that its value should be equal to or smaller than the half of the ATC capacity. For a 10-station/bi-directional ATC magazine, the maximum number of unit rotations from one tool to another is equal to or smaller than 5. In this case, if the ATC magazine has a magazine rotating speed (from index to next index) of 1 s, and the sequenced operations {O1-O2-O3-O4-O5-O6- O7-O8-O9- O10-O11- O12} require the cutting tools {T10-T9-T3-T5-T4-T6-T7-T8-T2-T1-T10-T8}, respectively, then the fitness value of chromosome-201, i.e. set of cutting tools {1-2-5-3-10-7-6-4-9-8} assigned to index numbers {1-2-3-4-5-6-7-8-9-10} on the ATC, is calculated as follows:

$$\text{INO[gene[opr[1]]} - \text{INO[gene[opr[2]]} = \text{INO[10]} - \text{INO[9]} = |5-9| = 4$$

$$\text{INO[gene[opr[2]]} - \text{INO[gene[opr[3]]} = \text{INO[9]} - \text{INO[3]} = |9-4| = 5$$

$$\text{INO[gene[opr[3]]} - \text{INO[gene[opr[4]]} = \text{INO[3]} - \text{INO[5]} = |4-3| = 1$$

$$\text{INO[gene[opr[4]]} - \text{INO[gene[opr[5]]} = \text{INO[5]} - \text{INO[4]} = |3-8| = 5$$

$$\text{INO[gene[opr[5]]} - \text{INO[gene[opr[6]]} = \text{INO[4]} - \text{INO[6]} = |8-7| = 1$$

$$\text{INO[gene[opr[6]]} - \text{INO[gene[opr[7]]} = \text{INO[6]} - \text{INO[7]} = |7-10| = 3$$

$$\text{INO[gene[opr[7]]} - \text{INO[gene[opr[8]]} = \text{INO[7]} - \text{INO[8]} = |10-6| = 4$$

$$\text{INO[gene[opr[8]]} - \text{INO[gene[opr[9]]} = \text{INO[8]} - \text{INO[2]} = |6-2| = 4$$

$$\text{INO[gene[opr[9]]} - \text{INO[gene[opr[10]]} = \text{INO[2]} - \text{INO[1]} = |2-1| = 1$$

$$\text{INO[gene[opr[10]]} - \text{INO[gene[opr[11]]} = \text{INO[1]} - \text{INO[10]} = |1-5| = 4$$

$$\text{INO[gene[opr[11]]} - \text{INO[gene[opr[12]]} = \text{INO[10]} - \text{INO[8]} = |5-6| = 1$$

Total unit rotations $= 33$
Total indexing time $= 33 \times 1 = 33$ s

The next step is to put all 400 chromosomes (200 parents and 200 child) in an ascending order starting from the one, which has *the least total indexing time* as 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 operates on the genes. The iterations are continued by this way. GA seeks to find the *position set* with the least cumulative fitness (total indexing time). 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.

A machine tool data base is prepared by using the manuals of several CNC machine tool manufacturers and integrated to the system. When the user selects the machine tool, parameters like tool capacity of the ATC, type of ATC movement and indexing time are captured and fed into the optimisation software.

When the GA-based optimisation system is used alone, the sequence of machining operations and cutting tools assignment to that operations can interactively be provided to the system by the user from the keyboard or they can be achieved from a previously created text file. When the system is executed in the OPPS-PRI, all necessary information regarding 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. An example

An example part including 12 features is shown in Fig. 9. The abbreviations used for the part features are tabulated in Table 4. The operations assigned to the part features by the operation selection module of the OPPS-PRI are given in Table 5. Tool selection module of the OPPS-PRI, provides candidate tools (up to 3)

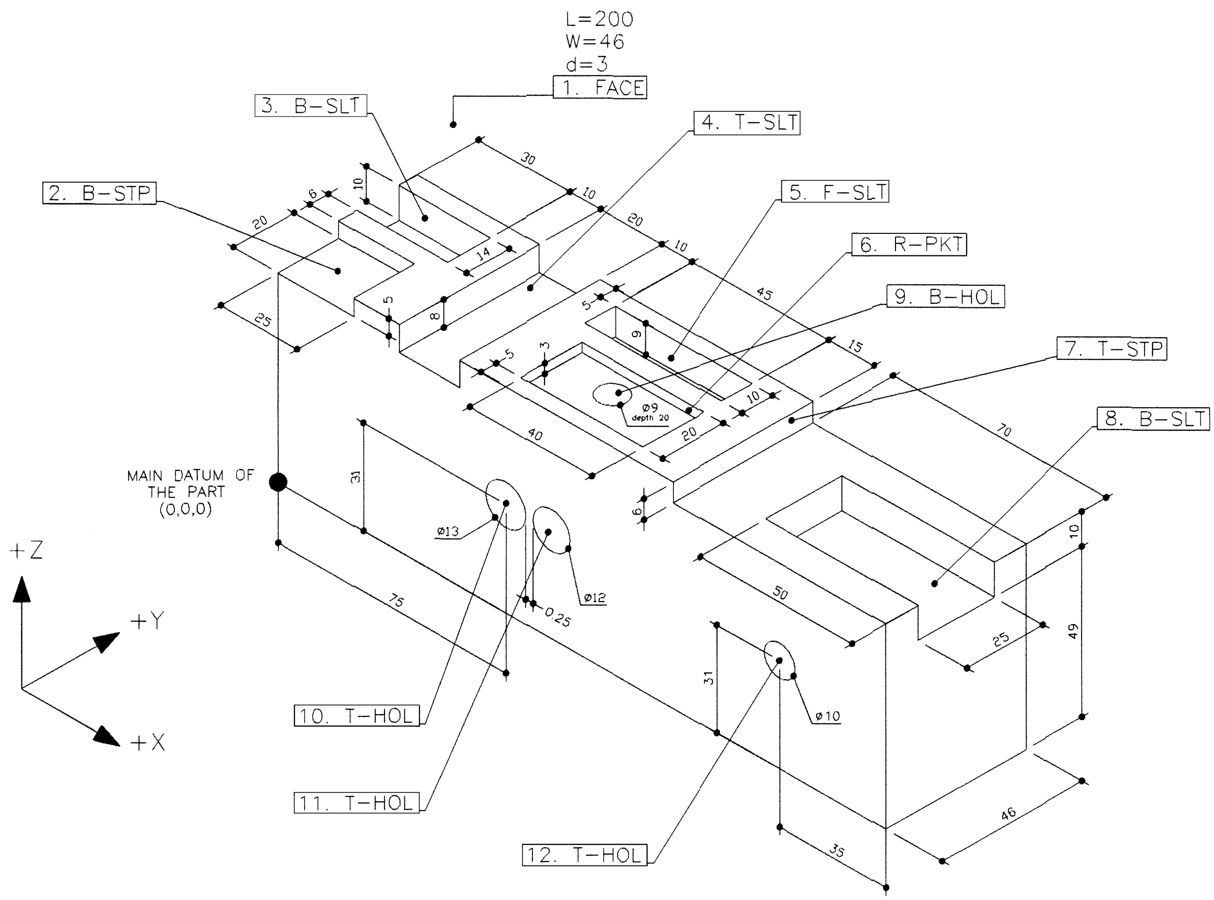


Fig. 9. An example part.

Table 4
List of features on the example part and their acronyms

Feature no.	Type	Abbreviations
1	Face	–
2	Blind step	B-STP
3	Blind slot	B-SLT
4	Thru slot	T-SLT
5	Face slot	F-SLT
6	Rectangular pocket	R-PKT
7	Thru step	T-STP
8	Blind slot	B-SLT
9	Blind hole	B-HOL
10	Thru hole	T-HOL
11	Thru hole	T-HOL
12	Thru hole	T-HOL

Table 5
Operations assigned to the features of the example part

Set-up no.	Operations	Operation	Corresponding feature
1	O1	Face milling	FACE (F1)
1	O2	Step milling	T-STP (F7)
1	O3	Slot milling	B-SLT (F8)
1	O4	Slot milling	T-SLT (F4)
1	O5	Slot milling	B-SLT (F3)
1	O6	Slot milling	F-SLT (F5)
1	O7	Step milling	B-STP (F2)
1	O8	Pocket milling	R-PKT (F6)
1	O9	Centre drilling	B-HOL (F9)
1	O10	Twist drilling	B-HOL (F9)
2	O11	Centre drilling	T-HOL (F10)
2	O12	Centre drilling	T-HOL (F11)
2	O13	Centre drilling	T-HOL (F12)
2	O14	Twist drilling	T-HOL (F10)
2	O15	Twist drilling	T-HOL (F11)
2	O16	Twist drilling	T-HOL (F12)

Table 6
Cutting tools assigned to (optimal) operations

Operation seq.	Operation no.	Cutting tools
1	O1	T1
2	O2	T1
3	O3	T5
4	O4	T4
5	O5	T2
6	O9	T9
7	O10	T7
8	O8	T3
9	O6	T3
10	O7	T3
11	O11	T9
12	O12	T9
13	O13	T9
14	O14	T10
15	O15	T8
16	O16	T6

for the operations and a single tool is assigned for each operation.

The sequence of operations is then optimised based on *minimum tool-change criterion*. The optimal sequence of operations together with the cutting tools selected for each operation is given in Table 6. The detailed information on the cutting tools can be found in Table 7.

The last input to the system involves the machine tool specifications, which can either be obtained from the machine tool data base or can be entered from the keyboard by the user. In this example, the target machine is assumed to be TAKSAN TMC-700 VMC, which has a bi-directional ATC with a holding capacity of 16 cutting tools and an indexing time of 0.69 s.

The problem described here falls into Phase 2(a) where the total number of cutting tools employed is smaller than the total number of index positions available on the ATC. Therefore, some cutting tools can also be duplicated in the tooling set-up of ATC magazine. At this stage, the user is warned whether he/she has spare tool(s)/tool holder(s) to be used for duplicating any tool or not. If available, the specified tool(s) can be used more than once. However, it should be noticed that tooling is expensive, so its profits should be tested carefully.

The proposed alternative position sets for the problem are tabulated in Table 8. The resulting configuration on the ATC is also depicted in Fig. 10.

Notice that GA converges rapidly to an optimal solution; 13 unit rotations of ATC or in other words a total turret indexing time of 8.97 s per part. The above problem is also asked to 10 average technicians, workers and operators. Average value of total unit rotations of ATC and indexing time obtained from this quiz is equal to 20 and 13.80 s, respectively. Even in this small sized problem, the gain is 4 s per component to be produced. For a batch of 40 000 parts, the total gain is about 44.44 h. The slower the ATC (the higher value of indexing time from tool-to-tool), the higher is the gain. 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.

Table 7
Tooling Information

Cutter	Type	Cutter	Adaptor	Insert
T1	Modern	ISCAR F90SD-D63-22-12	BT 40 KBM 22	SDTM 1205 PDR-HQ
T2	Solid	–	BT 40 PB 444 E	–
T3	Solid	–	BT 40 VT 10	–
T4	Modern	ISCAR E90A-D20-CS20	BT 40 VT 20	APKT 1003 PDTR-76
T5	Modern	ISCAR E90SP-D25-W25-10	BT 40 VT 25	SPMT 100408 TR-HQ
T6	Solid	–	BT 40 PB 444 E-10	–
T7	Solid	–	BT 40 PB 444 E-9	–
T8	Solid	–	BT 40 VT 12	–
T9	Solid	–	BT 40 MM B12 with (0–10 mm mandren)	
T10	Solid	–	BT 40 MM B16 with (4–16 mm mandren)	

Table 8
Proposed position sets

No. of iterations		Positions on ATC (with no duplications)										Fitness:	Fitness:
Itr. no.	Elapsed time (s)	1	2	3	4	5	6	7	8	9	10	No. of unit rotations	total index time (s)
1	1/5	T3	T7	T4	T5	T1	T2	T9	T10	T8	T6	20	13.80
2	2/12	T8	T6	T10	T3	T9	T2	T7	T1	T4	T5	19	13.11
3	3/15	T1	T5	T4	T7	T3	T9	T2	T10	T8	T6	15	10.35
4	20/50	T1	T5	T4	T3	T2	T7	T9	T10	T8	T6	15	10.35
5	30/80	T1	T5	T4	T2	T7	T3	T9	T10	T8	T6	13	8.97

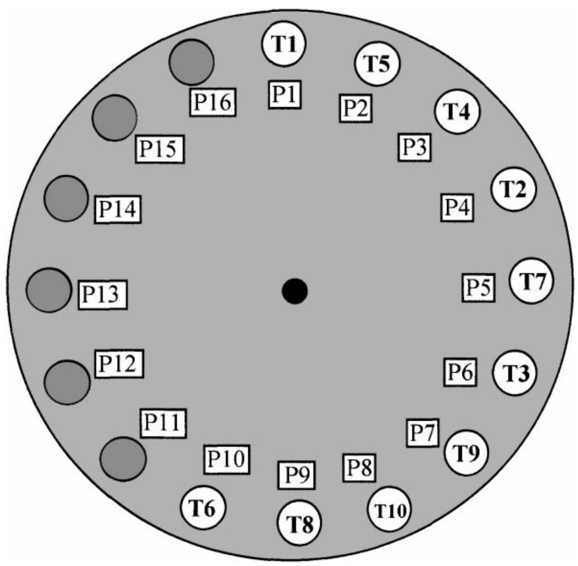


Fig. 10. Tool positions on the ATC for the example part.

6. Discussion and concluding remarks

Recent researches and developments in manufacturing area have the objectives of increased productivity and cost effectiveness by integrating many activities within a CIM system which is heading towards the Factory of the Future. For this purpose, an advanced production system called IMS is oriented towards the needs of the 21st century and designed to maintain and improve the vitality of *manufacturing sector* keeping it as the cornerstone of all economic activities and ensuring that manufacturing remains an attractive industrial area. IMS takes intellectual activities in manufacturing and improves productivity by systematising, *optimising* and flexibly integrating those corporate activities. Therefore, the optimisation of corporate activities in CIM and CAPP is one of the greatest targets of IMSs, since it is believed that only those industries capable of *making effective productions* would withstand international competition in the next millennium. AI based techniques like GAs are designed for capturing, representing, organising, and utilising knowledge by computers, and hence play an important role in *intelligent manufacturing*. AI techniques shorten the reaction time of a manufacturing system.

In this paper, a GA-based system developed for the optimisation of turret index positions of cutting tools to be used on the turret or ATC magazine of the CNC machine tools, is presented. It has been recognised that a small saving in the total turret indexing time will cause a considerable increase in cutting time by decreasing the non-cutting time, especially for high-volume production. The developed optimisation system can be used for both ATC magazines and turrets on CNC machine tools. It can be used as a stand-alone system. It has been also integrated to a process planning system called OPPS-PRI which has been implemented on a VMC. The developed system has been used in small and medium-sized manufacturing industries making batch production of spare parts for textile industry in Gaziantep city, with positive results. However, it has been recognised that for large-scale problems, GAs are somewhat slow and occasionally cannot converge to a global optimum.

With the developed optimisation system in this work, it would be possible to reduce the time spent in the indexing on tool magazines, and to contribute to the success of the manufacturing industry. This will lead to increased utilisation of CNC machine tools and maximisation of CNC productivity which are commonly considered as ultimate goals of CIM. The methodology described in this work can also be adopted to 5-axis machining centres, Flexible Manufacturing Systems and Flexible Manufacturing Cells which use chain type of tool magazines on which a huge amount of cutting tools are to be controlled.

References

[1] C.J. Chen, C.S. Tseng, The path and location planning of workpieces by genetic algorithms, Journal of Intelligent Manufacturing 7 (1996) 69–76.

[2] Compact Universal CNC Lathe Hi-Eco10, Manual, Hwacheon, 1993.

[3] Y.H. Derek, D. Debasish, A genetic algorithm application for sequencing of operations in process planning or parallel machining, IIE Transactions 28 (1996) 55–68.

[4] T. Dereli, Development of a process planning system for prismatic parts, Ph.D. Thesis, Department of Mechanical Engineering, University of Gaziantep, Turkey, 1998.

[5] T. Dereli, İ.H. Filiz, G. Görür, O. Yilmaz, Determining optimal sequence of machining operations using conflicting processing criteria and a genetic algorithm, in: Proceedings of the13th ISPE/IEE International Conference on CAD/CAM, Robotics and Factories of the Future, Vol. 1, Colombia, 1997, pp. 319–326.

[6] M. Gen, R. Cheng, Genetic Algorithms and Engineering Design, Wiley, New York, 1997.

[7] E.D. Goldberg, Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[8] HITEC Turning Centre Manual, Hitachi-Seiki Co., Ltd., 1996.

[9] L. Hong, Z. Guangzhou, L. Zongkai, A system of optimising nesting with analogical learning mechanism, Computers and Industrial Engineering 32 (1997) 713–725.

[10] H.N. Kamhawi, R.S. Leclair, C.L. Philip, Feature sequencing in the rapid design system using a genetic algorithm, Journal of Intelligent Manufacturing 7 (1996) 55–67.

[11] C.Y. Lee, J.Y. Choi, A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights, Computers Operations Research 22 (1995) 857–869.

[12] E. Öztemel, M. Düğenci, Job sequencing by genetic algorithms, in: Proceedings of the First Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey, 1996, pp. 428–437.

[13] S. Roy, Optimal design of process variables in multi-pass wire drawing by genetic algorithms, Journal of Manufacturing Science and Engineering 118 (1996) 244–251.

[14] TAKSAN TMC-700 VMC, Manual, Taksan, A.Ş., Kayseri, Turkey, 1995.

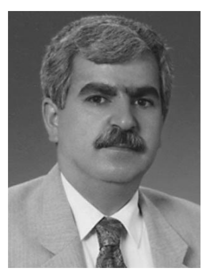
[15] G. Ulusoy, S.F. Şerifoğlu, U. Bilge, A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles, in: Proceedings of the First Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey, 1996, pp. 438–461.

[16] J.M. Usher, R.O. Bowden, The application of genetic algorithms to operation sequencing for use in computer aided process planning, Computers and Industrial Engineering 30 (1996) 999–1013.

[17] J. Vancza, A. Markus, Genetic algorithms in process planning, Computers in Industry 17 (1991) 181–194.

[18] T. Yokota, A Solution Method for optimal weight design problem of the gear using genetic algorithms, in: Proceedings of the 22nd International Conference on Computers and Industrial Engineering, Cairo, Egypt, 1997, pp. 423–426.

Türkay Dereli received his B.Sc. degree in Mechanical Engineering at the Middle East Technical University (METU) in 1992, and his M.Sc. and Ph.D. degrees in Mechanical Engineering from the University of Gaziantep, Institute of Natural and Applied Sciences, in 1994 and 1998, respectively. From 1992 to 1998, he was with the Department of Mechanical Engineering at the University of Gaziantep, first as a Research Assistant (from 1992 to 1996) and later as an Instructor (from 1996 to 1998). He is presently an Assistant Professor in the Industrial Engineering Department of the University of Gaziantep. Dr. Dereli has co-authored a text book on CAD and more than 25 technical papers. His current research interests include CAD/CAM, process planning, feature recognition, CNC part programming and artificial intelligence applications. He is a fellow of the Turkish Chamber of Mechanical Engineers (MMO), Turkish Mechanical Design and Production Society (MATIM) and Turkish Society of Information (TBD).



İ. Hüseyin Filiz is a Professor of Mechanical Engineering at University of Gaziantep in Turkey. He received his B.Sc. and M.Sc. degrees in Mechanical Engineering at Middle East Technical University (METU), Turkey, in 1973 and 1975, respectively. In 1981, he received his Ph.D. degree from the University of Manchester, Institute of Science and Technology (UMIST), UK. He has held various faculty positions at the University of Gaziantep. He is presently the president of the same university. Professor Filiz has authored and co-authored five text books and more than 50 technical papers on the design of machine tools, the design of machine elements and CAD/CAM. He is a member of the Turkish Chamber of Mechanical Engineers (MMO) and Turkish Mechanical Design and Production Society (MATIM).