MultiCraft

International Journal of Engineering, Science and Technology
Vol. 3, No. 1, 2011, pp. 183-190

INTERNATIONAL JOURNAL OF ENGINEERING, SCIENCE AND TECHNOLOGY
www.ijest-ng.com

©2010MultiCraft Limited. All rights reserved

Hybrid genetic algorithm for minimizing non productive machining time during 2.5 D milling

Arun Kumar Gupta[1]*, Pankaj Chandna [2] & Puneet Tandon [3]

[1]Department of Mechanical Engineering, National Institute of Technology Kurukshetra,Haryana, INDIA*
[2] Department of Mechanical Engineering, National Institute of Technology Kurukshetra,Haryana, INDIA
[3]PDPM Indian Institute of Information Technology, Design &Manufacturng Jabalpur, MP, INDIA
[*]Corresponding Author: e-mail: arun_gupta@engineer.com, Tel +91-1262-274069

Abstract

Minimization of non-productive time of tool during machining for 2.5 D milling significantly reduces the machining cost. The tool gets retracted and repositioned several times in multi pocket jobs during rough machining which consumes 15 to 30% of total machining time depending on the complexity of job. The automatic CNC program commonly generates contour parallel tool path. Optimization of tool path length during non-productive time can be modeled on Traveling Salesman Problem (TSP) which belongs to family of non-deterministic polynomial (NP) hard problem. In the present work, a Hybrid Genetic Algorithm (HGA) has been proposed to optimize the non-productive tool path in which the initial seed solution is generated by special heuristic and combined with random initial solution generated by simple genetic algorithm (SGA). A defined performance index known as Relative percentage deviation (RPD) has been used for analyzing the results by varying the size of the jobs. From the analysis, it is found that HGA shows superiority over SGA for same computation time limit as the stopping criteria.

Keywords: Hybrid Genetic Algorithm (HGA), 2.5D Milling, Tool path optimization, Non-productive time, CNC Machining.

1. Introduction

Now-a-days, industry is using advanced planning for 2.5 D machining to increase productivity of high-speed machining and for mass production. The automatic tool path planning is being done using hi-end CAD/ CAM software for majority of complex jobs of 2.5D milling. The jobs are being modeled on CAD software which is being used to generate the CNC codes in conjunction with Computer Aided Manufacturing (CAM) packages. The movement of tool is synchronized with the help of these CNC codes. Total time spent during 2.5 D milling primarily comprises of productive and non-productive time. The time spent by the tool during chip formation is known as productive time whereas the time spent during repositioning of the tool is known as non-productive time.

Lot of work has been reported for minimizing the productive time by optimizing the cutting parameters (Dereli *et al*, 2001), type of tool paths to be involved (Yao and Gupta, 2004), tool selection and tool sequencing (Zhang and Ge, 2008), however, very limited number of researchers have considered non-productive time. In case of rough machining, the tool gets retracted and repositioned several times depending on complexity of the jobs. The non-productive time consumes 15 to 30% of total machining time and varies with complexity of the job that for the machining of multi pocket jobs by small diameter tool (Souza *et al*, 2001 and Castelino *et al*, 2003). Therefore minimization of non-productive time becomes important in case of multi pockets jobs (Oysu and Bingul, 2009).

The tool moves from one region to next rapidly and visits once in a meticulous region. The automatic program generally generates contour parallel tool path which has same entry and exit point. Hence, the problem can be formulated as a generalized Traveling Salesman Problem (TSP) which belongs to nondeterministic polynomial (NP) hard problem. Khan *et al*, (1999) identified firstly the problem of minimizing non-productive time in 2.5 D milling.. They formulated the problem as TSP and applied simulated annealing algorithm based on stochastic search.

Huang *et al*, (1997) proposed a genetic algorithm for welding sequence problem by formulating the problem as travelling salesman problems. Kolahan and Liang (1999) applied Tabu-search approach to minimize the cost in hole making processes by optimizing tool travel scheduling, tool switch scheduling, tool selection, and machining speed specification. Kolahanand Liang (2000) also proposed the algorithm for minimization of non-productive time for drilling, pocketing and face milling operations. The algorithm was tested for the job size of 50, 100, 150, and 200 nodes with stopping limit of 600s and showed better results especially for the large size jobs. Souza (2001) minimized the machining time for 2.5 D milling by optimizing the cutting parameters and the tool selection. Castellino *et al*, (2003) proposed a heuristic method to solve TSP with precedence constraints and Sequential Ordering Problem (SOP) for optimizing non-productive tool path movements in laser cutting operation. Onwubolu and Clerc (2004) proposed new heuristic approach, particle swarm optimization, for minimizing the operating path of CNC drilling operations. Whereas Ghaiebi and Solimanpur (2007) optimized tool airtime (non-productive time) and tool switching time for hole-making operations by formulating a 0–1 non-linear mathematical model and solved using ant colony algorithm. Kentli and Alkaya (2009) developed record-to-record travel algorithm hybridized with simulated annealing for the bolt assembly and drilling sequences.

Tawfik *et al*, (2007) formulated optimal machining time for small-hole EDM-drilling problem as Traveling Salesman Problem (TSP) and proposed Guided Fast Local Search (GFLS) technique. Qudeiri *et al*, (2007) optimized productive and non-productive time during 2.5 D problem formulated as TSP using GA. Oysu and Bingul (2007) proposed different Genetic Algorithm (GA) with preprocessing path produced better solution for the problem considered for tool path optimization problem during 2.5D milling. .

For the solution of traveling salesman problem, Ulder *et al*, (1991) solved eight classical TSP problems ranging in size from 48 to 666 cities with the order crossover based genetic algorithm and the Lin-Kernighan hill-climbing heuristic. Dhingra and Chandna (2010a) minimized multi criteria SDST flow shop scheduling including weighted sum of total tardiness, total earliness and makespan by developing special heuristic based on hybrid genetic algorithm in which initial feasible sequence has been obtained by special heuristic similar to NEH (Nawaz *et al*., 1983). Dhingra and Chandna (2010b) extended the work to evaluate the performance of the proposed modified heuristic based genetic algorithm for Bi-criteria flow shop scheduling problems by implementing computational analysis. Oysu and Bingul (2009) extended the work by hybridizing the Simulated Annealing SA with GA and concluded that hybrid SAGA produced better solutions than standard SA and GA for the 88 and 344 nodes 2.5 D milling problem.

The present work is focused on the formulation of the model and proposes hybrid genetic algorithm approach to solve the non productive time problem. Special heuristic similar to NEH (Nawaz *et al*, 1983) has been proposed for obtaining initial Seed Solution which combined with random initial population of simple genetic algorithm (SGA) is called as Hybrid Genetic Algorithm (HGA). The proposed HGA has been applied for optimizing the non-productive tool path. A defined performance index known as Relative Percentage Deviation (RPD) has also been used for comparative analysis by varying the job size.

2. Problem Description

In the present work, non-productive time for 2.5D milling has been optimized by reducing unwanted movements of tool during rapid travelling. The tool path generation techniques have widely used contour-parallel tool path. Every contour consists of single retraction and entry point. These retraction and entry points of a contour coincide with each other and are termed as nodes. During milling the tool has to travel from one node to the next node rapidly and perform productive and non-productive movements alternatively. To formulate the problem of non-productive time minimization some assumptions, parameters and fitness functions are considered and described below:

2.1 Parameters:

p	*Index for nodes*	$p=1,2,3........n$
p_{ij}	*Distance from one node to another node*	$p_{ij}>0$
Ps	*Population Size*	$Ps=20,40,60,80,100$
K	*Non looping constraint*	$K=1 or 0$
l_p	*Length of productive tool path*	$l_p \geq 0$
l_{np}	*Length of non productive tool path*	$l_{np} \geq 0$
f_p	*Productive feed*	$f_p>0$ *(mm per minute)*
f_{np}	*Non productive feed*	$f_{np}>0$ *(mm per minute)*
T	*Total machining time*	T *is in minutes*
s	*Optimum sequence index*	$s=1,2,3........r$
x,y	*Co-ordinates of nodes*	
d_{ij}	*Distance between nodes p=i and p=j.*	d *is in mm*

2.2 Assumptions:

- Single tool has been used.
- The left over area where tool could not enter has not been reconsidered.
- The speed of tool for non-productive movement is constant.
- The contour-parallel tool path has same entry and exit point which is called as node.

2.3 Fitness Function: Fitness function considered in this work given by equation (1) is sum of product of distances between nodes and non looping constant for all possible combinations. The optimization of non-productive time problem can be considered as the NP-hard problem. Let 'p' be the node index (entry/exit point of a particular region of tool path). The tool moves from one to next region by rapid movement and visits a meticulous region once in whole operation. Therefore K_{ij} non looping constraint with value 1 or 0 and from equation (2(a), 2(b)) sum of these non looping constraints remains equal to one. If tool is moving from node 'i' to' j' than K_{ij} = 1 otherwise K_{ij} = 0. Total length of non productive tool path (Oysu and Bingul., 2009) is given as:

$$\min(l_{np}) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} K_{ij} \quad (1)$$

$$\sum_{i=1}^{n} K_{ij} = 1 \quad 2(a)$$

$$\sum_{i=1}^{n} K_{ji} = 1 \quad 2(b)$$

Total machining time depends upon productive and non-productive machining time. It is calculated by relation (3) as under:

$$T = \left[\frac{l_{np}}{f_{np}} + \frac{l_p}{f_p} \right] \quad (3)$$

From equation (4) Optimum tool path length can be calculated as follows:

$$l_{np} = \sum_{s=1}^{n} \sqrt{(x_s - x_{s+1})^2 + (y_s - y_{s+1})^2} \quad (4)$$

where x and y are the coordinates of nodes arranged in optimum sequence. The optimum sequence is generated with the help of Simple Genetic Algorithm and Hybrid Genetic Algorithms.

3. Genetic Algorithm

Genetic Algorithms (GA) perform the search by solution recombination and belong to the group of Evolutionary Algorithms (EA). As the name suggests, they imitate the process of evolution and use the inspiration of natural selection, recombination and mutation to direct the search process. The set of solutions is called population in the perspective of Genetic Algorithm. Consequently, a single solution of this population is called individual or chromosome. Chromosomes are made of genes and the values of a gene are called alleles. A gene is a property of a solution, which can take values of a predefined domain. The position of a gene is called locus. If a gene has taken a certain value it is called allele. So, the genes of a chromosome describe the genotype of an individual. A phenotype describes a certain individual with all its states, i.e. all genes have fixed alleles. Genetic algorithms are an optimization technique based on natural evolution. They include the survival of the fittest idea into a search algorithm, which provides a method of searching which does not need to explore every possible solution in the feasible region to obtain a good result. Genetic algorithms are based on the natural process of evolution. In nature, the fittest individuals are most likely to survive and mate; consequently the next generation should be fitter and healthier because they were bred from healthy parents. The same idea is applied to a problem by first 'guessing' solutions and then combining the fittest solutions to create a new generation of solutions, which should be superior than the previous generation.

3.1 Simple Genetic Algorithm (SGA): A Genetic Algorithm (GA) uses probabilistic selection as a base for evolving a population of problem solutions. An initial population is created and subsequent generations are generated according to a pre-specified breeding and mutation methods inspired by nature. GA generates initial population randomly according to constraints mentioned. The best solution is selected from the population as evaluated by fitness function. This best solution is termed as elite solution and is further make crossover and mutate with remaining population. The new population again goes through same process and is repeated to calculate the best solution as explained in the following steps.

Step 1: Randomly generation of population as per population size.

Step 2: The algorithm then creates a sequence of new population. At each step, the algorithm uses individuals in current generation to create the next population. To create the new population, the algorithm performs the following steps:

a. Scores each member of the current population by computing fitness i.e. minimizing.
b. Select members, called parents, based on their fitness.
c. Some of the individuals in the current population that have lesser fitness are chosen as *elite*. These elite individuals are conceded to the next population.
d. Produces offspring from the parents. Offspring are produced either by combining the vector entries of a pair of parents—crossover or by making random changes to a single parent—mutation.
e. Replaces the current population with the children to form the next generation.

Step 3: The algorithm stops when computational time reaches 600 seconds.

3.2 Hybrid Genetic Algorithm (HGA): The quality of result generated by GA is dependent on initial seed solution to a great extent. If the initial solution is generated randomly, the quality of result might be inferior in same stopping limits. SGA have the drawback of inferior results as the size of the problem increases and hence for achieving better solution in a very reasonable time, special heuristic based HGA is developed for optimization of non-productive machining time problem. The initial seed solution for HGA is obtained by special heuristic as shown in Figure 1 and is combined with randomly generated population (Ps-1) as per GA procedure which is shown in Figure 2.

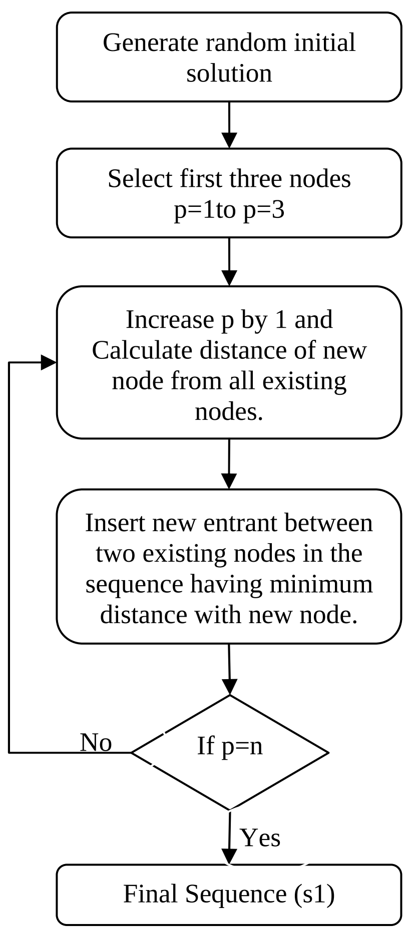


Figure 1. Special Heuristic

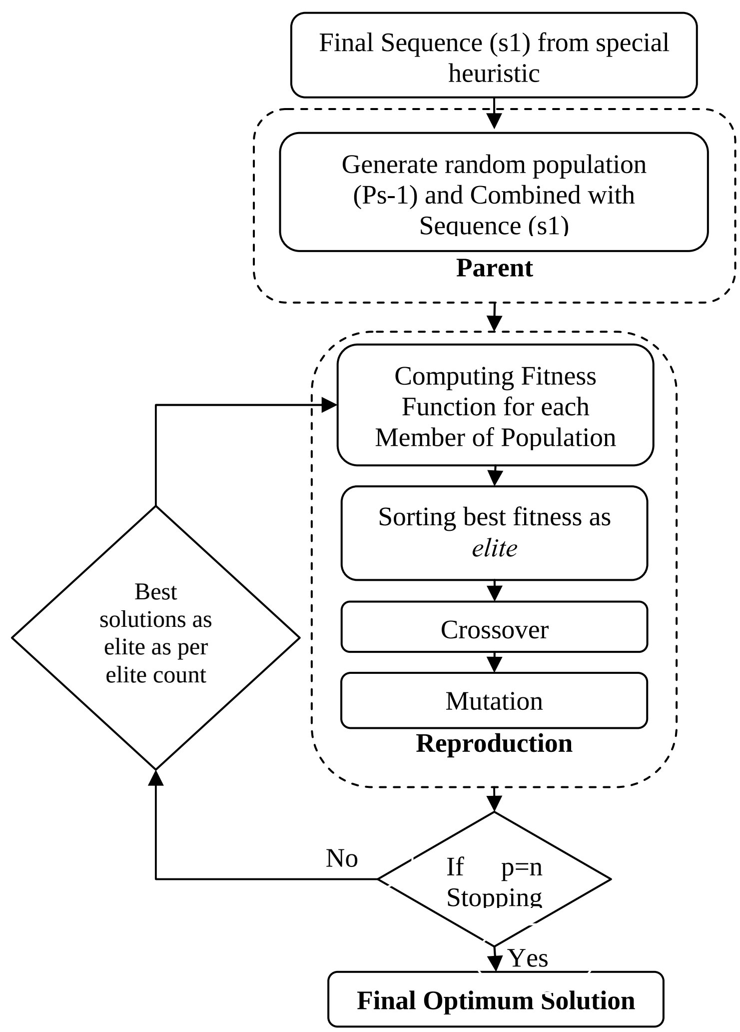


Figure 2. Hybrid Genetic Algorithm

Detailed steps for special heuristics are explained as under:

Step1: Generate initial solution randomly.

Step2: Select first three nodes from the node index p=1 to p=3. Three nodes are selected as total distance traveled by the tool is independent of sequence of these three nodes.

Step3: Select next node from node index (Increment p by 1) and compute distance of new node from all existing nodes given by equation (5).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

Step4: Insert new entrant between two existing nodes in the sequence having minimum distance with the new node.
Step5: Go to Step3 and Repeat the process till p = n.
Step6: Obtain sequence s1.

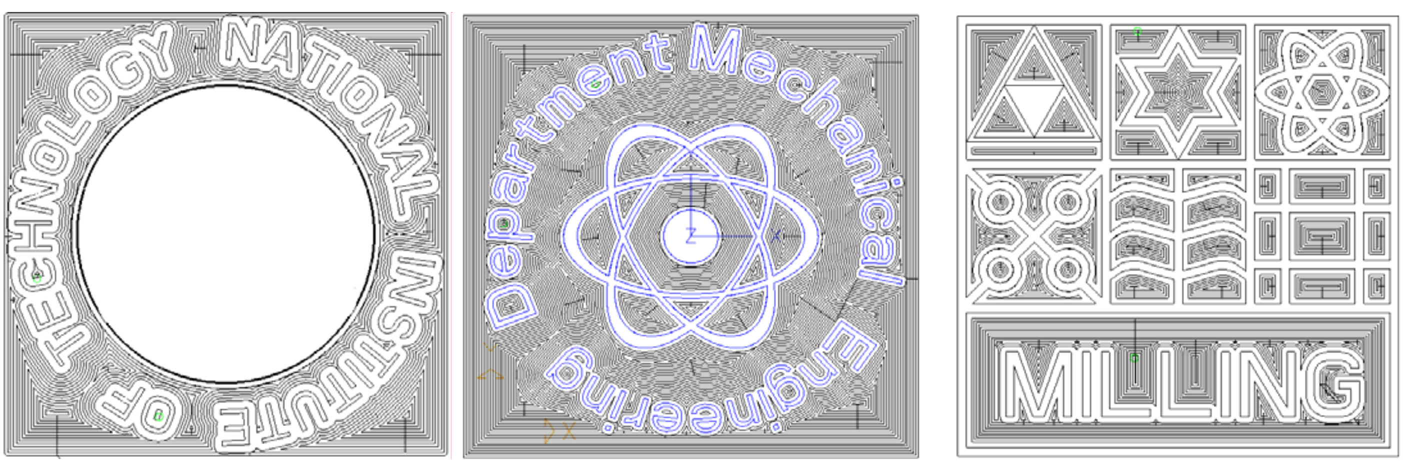
3.3Parameters Settings: After comparing experimentally, it is found that the following parameters as shown in table 1 give better results for non-productive machining time problem to run SGA and HGA. As the algorithm is being tested over three different sized node problems therefore varying population sizes are considered. The stopping criterion of 600 seconds is being considered for both algorithms for fair comparison. However, if stopping criterion is maximum number of generations then HGA always produces better results than SGA as time of initial heuristics is not been included in HGA since it starts from better initial solution obtained from the heuristics. Hence, for fair comparison among SGA and HGA, computational time of both the GAs should be same.

Table 1. Parameters for Genetic Algorithm

Parameter	Value
Population Size	20, 40, 60, 80, and100
Crossover Function	Partially Matched Crossover (PMX)
Mutation Function	Reciprocal Exchange (RX)
Elite Count	2
Crossover Fraction	0.8
Mutation Fraction	0.15
Stopping Condition	600 seconds

4. Results and Discussions

In this study, three types of jobs (easy, medium and hard) have been considered to optimize non productive time for 2.5D milling. The problem of milling has been segregated on the basis of number of retraction points. The 275, 150 and 85 retraction points (nodes) is treated as hard, medium and easy problems respectively and have been shown in Figure 3. The computational experiments have been performed on a personal computer with 2.00 GHz Core^{TM} Duo T5800 Pentium Processor and 2 GB of RAM.



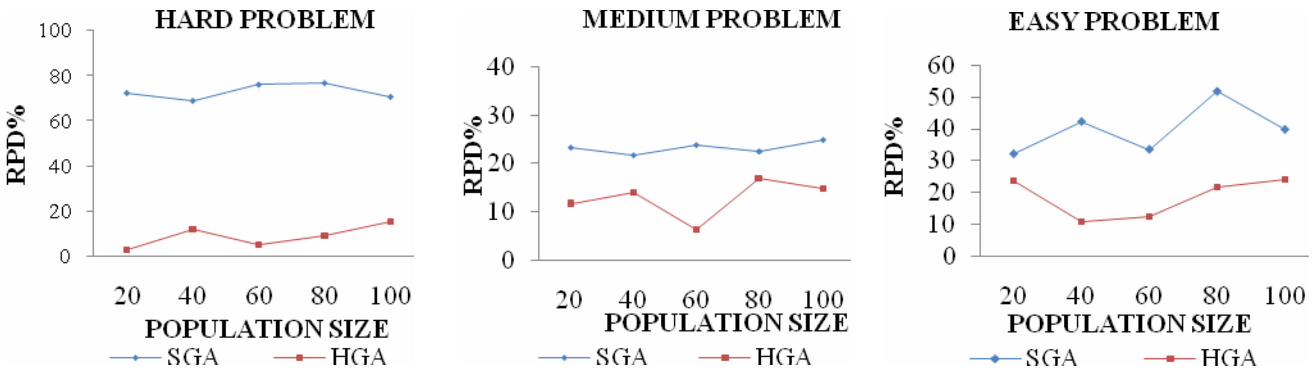
(a) **(b)** **(c)**

Figure 3. Different Size Problems Considered for Non Productive Machining Problem
(a) (275 Nodes) **(b)** (150 Nodes) **(c)** (85 Nodes)

The code is developed in MATLAB for simple Genetic Algorithm (SGA) and Hybrid Genetic Algorithm (HGA). The results obtained by proposed HGA are compared with SGA for minimization of non-productive time during milling of different size jobs with same stopping limit of time (600s).The effect of population size has also been analyzed on solution quality. Retraction points of contour parallel tool path have been calculated for all the three jobs. The feed rate during non-productive movements is taken to be 760 mm per minute. Initial Solution obtained from the special heuristic is used as a seed solution and combined with a set of other population for finding near to optimal solution known as HGA. The results of HGA and SGA are being compared with the help of relative percentage deviation (*RPD*) by relation defined as:

$$\text{Relative percentage deviation (RPD)} = \frac{Method_{sol} - Best_{sol}}{Best_{Sol}} X100` \quad (6)$$

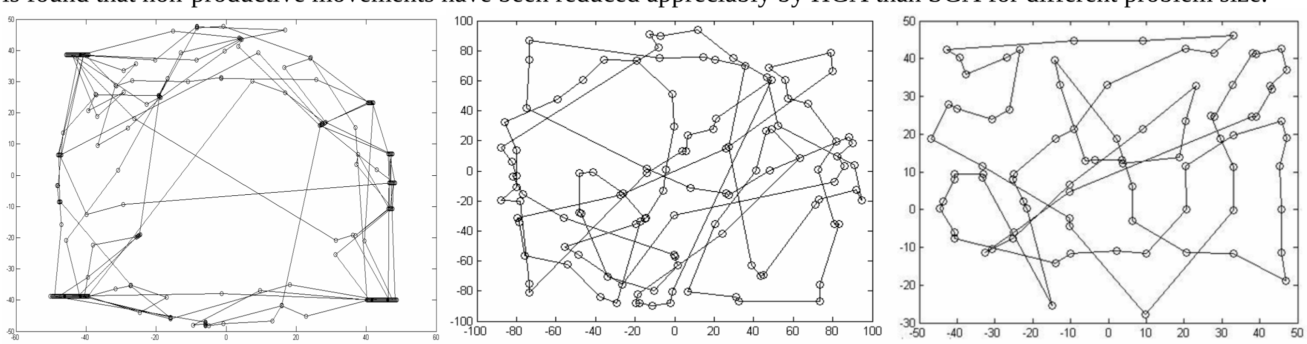
$Best_{sol}$ can be found among the results obtained by running GA five times for a particular job and $Method_{sol}$ is final average solution given by the algorithm for all the five runs. Results obtained by running HGA and SGA on 275 nodes, 150 nodes and 85 nodes have been shown in Figure 4. The performance of proposed HGA gives better results than SGA for all the three problems with RPD of 3.00 %, 6.42% and 11.01 % respectively. It has also been found that, performance of HGA varies with increase in population size.



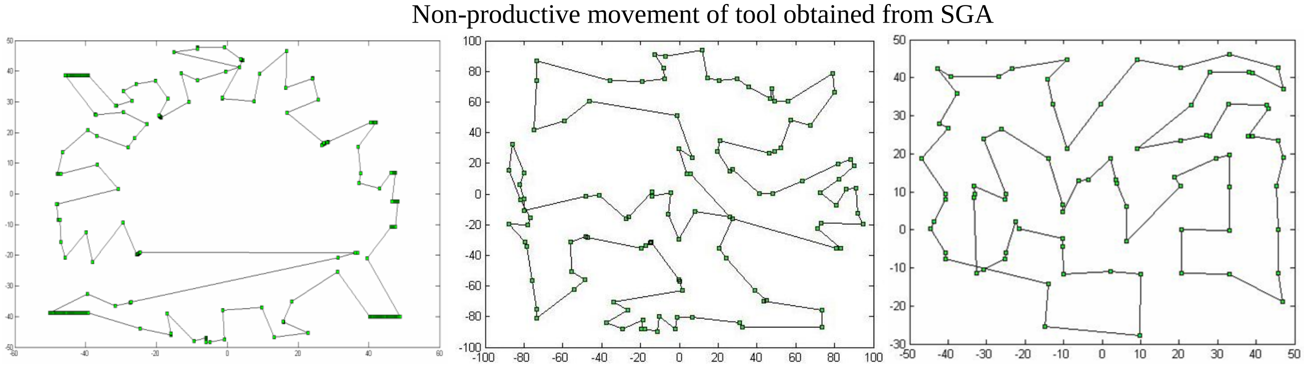
(a) (b) (c)

Figure 4. Relative percentage deviation (RPD) for three problems of size = **(a)** (275 Nodes) **(b)** (150 Nodes) **(c)** (85 Nodes)

The result obtained by HGA and SGA have been shown in Figure 5 in terms of non-productive movements of tool. Therefore, it is found that non-productive movements have been reduced appreciably by HGA than SGA for different problem size.



Non-productive movement of tool obtained from SGA



Non-productive movement of tool obtained from HGA

(a) (b) (C)

Figure 5. Non Productive Movement of Tool during Machining, Problem Size= **(a)** (275 Nodes) **(b)** (150 Nodes) **(c)** (85 Nodes)

5. Conclusions

Minimization of non-productive time of tool during machining for 2.5 D milling significantly reduces the machining cost. The tool gets retracted and repositioned several times in multi pocket jobs during rough machining which consumes 15 to 30% of total machining time depending on the complexity of job. In the present work, Special heuristic based Hybrid Genetic Algorithm (HGA) has been proposed and compared with Simple Genetic Algorithm (SGA). Three types of problems have been considered for comparative analysis with simple genetic algorithm (SGA). From the analysis it has been found that HGA performed well for all three easy, medium and hard sized problems. However, as number of nodes increases HGA proves to be more effective for minimizing non-productive machining time. Hence, it may be concluded that the performance of the HGA discovered the absolute superiority for large size problems. The result also shows that the HGA gives the better optimum solution at a population size of 60 for all the problem size. The work might be further extended for machining problems, where multi tools are being used with varying speed during non productive movements. Proposed HGA can be applied on 3D contour problems also which are very important in current scenario. Designing optimal parameters using design of experiments approach for HGA can also be the scope for further analysis in tool path optimization.

References

Castelino K., Souza R. D and Wright P. K., 2003. Tool-path Optimization for Minimizing Airtime during Machining, *Journal of Manufacturing Systems*, Vol. 22, No. 3, pp. 173-180.

Dereli T., Filizz I.H. and Baykasoglu A. 2001, Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms. *International Journal of Production Research*, Vol. 39, No. 15, pp. 3303-3328

Dhingra A., Chandna P., 2010a. A Bi-criteria M-Machine SDST Flow Shop Scheduling using Modified Heuristic Genetic Algorithm, *International Journal of Engineering, Science and Technology*, Vol. 2, No. 5, pp. 216-225.

Dhingra A., Chandna P.,2010b. Hybrid Genetic Algorithm for SDST flow shop scheduling with due dates, A case study *International Journal Advanced Operations Management*, Vol. 2, No. 3/4, pp. 141-161.

Ghaiebi H. and Solimanpur M., 2007. An ant algorithm for Optimization of hole-making operations. *Computers & Industrial Engineering*, Vol. 52, pp. 308–319.

Han G. C. and Na S. J., 1999. A Study on Torch Path Planning in Laser Cutting Processes Part 2. *Cutting Path Optimization Using Simulated Annealing*, Vol. 1 No.1 pp. 62-70.

Huang M., Hsieh C., Arora J.S., 1997. A genetic algorithm for sequencing type problems in engineering design. *InternationalJournal for Numerical Methods in Engineering*, Vol. 40, pp. 3105-3115.

Kentli A. and Alkaya A. F., 2009. Deterministic Approach to Path Optimization Problem, *Ozean Journal of Applied Sciences*, Vol. 2, No. 2, pp. 149–157.

Khan W.A., Hayhurst D.R. and Cannings C., 1999. Determination of optimal path under approach and exit constraints. *European Journal of Operational Research*, Vol. 117, No.2, pp. 310-325.

Koeing O. and Jouaneh M., 2005. Minimization of Airtime in Cutting and Welding Applications. *International Conference on Robotics and Automation, Barcelona*, pp. 3300-3305.

Kolahan F. and Liang M., 2000. Optimization of hole-making operations: A Tabu-search approach, *International Journal of Machine Tools & Manufacture*, Vol 40, No 12, pp. 1735-1753.

Kolahan F. and Liang M., 1999. Tabu Search Approach to Optimization of Drilling Operations, *Computers & Industrial Engineering*, Vol 31, No. 1&2, pp. 371-374.

Nawaz, Muhammad E. Jr, Ham E.E. and Inyong, 1983. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, *International Journal of Management Sciences,* Vol. 11 No.1, pp. 91-95.

Onwubolu G. C. and M. Clerc, 2004. Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research*, Vol.42, No.3, pp.473 – 491.

OysuC.andBingul Z., 2007. Tool path optimization using genetic algorithms. *Conference on Genetic and Evolutionary Methods, Las Vegas, USA*, pp. 120–126.

Oysu C. and Bingul Z., 2009. Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining, *Engineering Applications of Artificial Intelligence*, Vol. 22,pp. 389-396.

Qudeiri J.A., Yamamoto H. and Ramli R., 2007. Optimization of operation sequence in CNC machine tools using genetic algorithm, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 1, No.2, pp. 272–282.

Souza R. D., Wright P. and Skquin C., 2001. Automated Micro planning for 2.5-D Pocket Machining, *Journal of Manufacturing Systems*, Vol. 20, No. 4, pp. 228-296

Ulder N.L.J. and Aarts E.H.L., 1991. Genetic local search algorithm for traveling salesman problem, *Parallel Problem Solving from Nature*, Vol. 496, pp. 109-111.

Yao Z. and Gupta S. K., 2004, Cutter path generation for 2.5D milling by combining multiple different cutter path patterns, *International Journal of Production Research*, Vol. 42, No. 11, 2141-2161.

Zhang Y. and Ge L., 2008. Selecting optimal set of tool sequences for machining of multiple pockets, *International Journal of Advance Manufacturing Technology*, Vol. 42, pp. 233-241.

Biographical notes

Arun Gupta is a research scholar in the Department of Mechanical Engineering at NIT Kurukshetra Haryana INDIA. He has obtained Bachelors degree from Maharshi Dayanand University Rohtak Haryana INDIA in 2002, Masters Degree from Panjab Engineering College Chandigarh INDIA. He has guided many to undergraduate & post graduate students on projects thesis. His areas of interest include: CAD/CAM, Metaheuristics, Manufacturing Process, Computer integrated manufacturing etc.

Pankaj Chandna is presently serving as an Associate Professor in the Department of Mechanical Engineering at the National Institute of Technology Kurukshetra Haryana, India. After graduating in Mechanical Engineering from NIT Kurukshetra (formerly REC Kurukshetra) in 1989, he received his Masters in 1991 and his PhD in 2003 from Kurukshetra University, Kurukshetra. He is supervising 8 PhD research scholars and has supervised 20 MTech dissertations and 15 UG projects. He has more than 35 research papers in international/national journals and conferences to his credit. His current research interest areas include sequencing and scheduling, ergonomics, supply chain management, inventory management, machine learning, etc

Puneet Tandon is serving as a Professor in the discipline of Mechanical Engineering as well as Design Programme at IIITDM Jabalpur (India). His current research interests are CAD/ CAM/ CAE, product development and advances in design and manufacturing. He has more than 90 publications in International referred Journals and International / National Conferences to his credit.