International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:8, No:6, 2014

Minimization of Non-Productive Time during 2.5D Milling

Satish Kumar, Arun Kumar Gupta, Pankaj Chandna

***Abstract*—**In the modern manufacturing systems, the use of thermal cutting techniques using oxyfuel, plasma and laser have become indispensable for the shape forming of high quality complex components; however, the conventional chip removal production techniques still have its widespread space in the manufacturing industry. Both these types of machining operations require the positioning of end effector tool at the edge where the cutting process commences. This repositioning of the cutting tool in every machining operation is repeated several times and is termed as non-productive time or airtime motion. Minimization of this non-productive machining time plays an important role in mass production with high speed machining. As, the tool moves from one region to the other by rapid movement and visits a meticulous region once in the whole operation, hence the non-productive time can be minimized by synchronizing the tool movements. In this work, this problem is being formulated as a general travelling salesman problem (TSP) and a genetic algorithm approach has been applied to solve the same. For improving the efficiency of the algorithm, the GA has been hybridized with a noble special heuristic and simulating annealing (SA). In the present work a novel heuristic in the combination of GA has been developed for synchronization of toolpath movements during repositioning of the tool. A comparative analysis of new Meta heuristic techniques with simple genetic algorithm has been performed. The proposed metaheuristic approach shows better performance than simple genetic algorithm for minimization of non-productive toolpath length. Also, the results obtained with the help of hybrid simulated annealing genetic algorithm (HSAGA) are also found better than the results using simple genetic algorithm only.

***Keywords*—**Non-productive time, Airtime, 2.5 D milling, Laser cutting, Metaheuristic, Genetic Algorithm, Simulated Annealing.

I. Introduction

MILLING is a versatile and useful machining operation due to its capability of manufacturing the complex geometric surfaces with reasonable accuracy and surface finish for most of the manufacturing industries. The flexibility and adaptability of milling process have been highly improved with the introduction of high speed computer numerical controlled milling machines. Hence, while manufacturing mechanical components, milling is still one of the most essential metal cutting processes. Increasing profits and production quality by optimization of machining parameters are the vital issues in the present competitive environment. The economy and quality of production are highly influenced by machining parameters including cutting speed, feed rate, width of cut (step over), depth of cut and tool diameter etc. The total time consumed during milling operation includes, the tool productive machining time & non-productive machining time. The minimization of this non-productive machining time becomes more significant during mass production with high-speed machining. When the smaller tools often have to remove a number of small regions, the non-productive marching time becomes particularly important [2]. Nearly 15% of the total machining time is being consumed during non-productive movements [14] for most of the jobs and varies with the complexity of the jobs.

When the tool moves from one to next region by rapid movement and visits a meticulous region once in whole operation, the entry and exit points of spiral tool path (contour parallel toll path) coincide with each other and termed as nodes. Reference [2], [7] have converted this type of non-productive machining time problem into general Traveling Salesman Problem. Reference [9] introduced ant colony system, a distributed algorithm that is applied to the TSP. Similarly Reference [4] also converted the problem of minimization of the total processing cost for hole-making operations into traveling salesman problem. As there are (n-1)/2! possible solutions of these type of problems, therefore the problem is considered as NP-hard problem [2] and the complexity of problem is subject to the number of nodes. The traveling salesman problems have been solved by many researchers using the probabilistic approaches such as genetic algorithm [10], simulated annealing [16], ant colony optimization [8], [9], a memetic-clustering-based evolution strategy etc. For instance, Reference [12] applied GA with partially modified crossover (PMX) with the tour notation and found that this algorithm gave a tour whose length was found 10% percent smaller than the random solution for 33 city problem. Similarly, Reference [10] solved eight classical TSP problems ranging in size from 48 to 666 cities with a genetic algorithm with order crossover OX and the Lin-Kernighan hill-climbing heuristic. Some special heuristic methods also were developed by many of the researchers to solve TSP. Reference [7] solved TSP with precedence constraints using a heuristic method. An evolutionary algorithm, called the heterogeneous selection evolutionary algorithm (HeSEA) proposed by [5] for solving large TSP. Reference [18] proposed a new evolution strategy based on clustering and local search scheme for large-scale TSP. The problem is divided into several sub problems, the approximate optimal

Satish Kumar is working as Assistant Professor in the Department of Mechanical Engineering at Vaish College of Engineering, Rohtak Haryana India (e-mail: satish1212@gmail.com).

Arun Kumar Gupta is working as Assistant Professor in the Department of Mechanical Engineering at Vaish College of Engineering, Rohtak Haryana India (phone: 01262-274069; e-mail: arun_gupta@ engineer.com).

Pankaj Chandna is serving as a Professor in the Department of Production & Industrial Engineering at the National Institute of Technology Kurukshetra Haryana, India (e-mail: pchandna08@gmai.com).

tour for each sub problem is searched by a local search technique and the sub tours are properly connected. Reference [6] developed an ant colony optimization (ACO) algorithm for the optimization of hole-making operations. Reference [17] used economize machining process to optimize feed rate, spindle speed, depth of cut, machining time.

Continuous improvements on the performance of different machining processes by minimization of non-productive time and synchronizing of tool path movements have been reported by numerous researchers in the last two decade. Reference [4] studied the optimization of toolpath during rapid movements (non-productive machining) for drilling, pocketing and face milling operations and the machining time during chip generation was also minimized by optimizing the cutting parameters and tool sequences. Reference [2] solved TSP by Combined global search feature of GA and local search feature of SA, hybrid approach using GA and SA produced about 1.5% better minimum path solutions than standard GA and 47% better minimum path solutions than standard SA. Reference [1] proposed a hybrid GA to optimize non-productive movement for 2.5 D milling for different job sizes. Reference [15] proposed a Genetic algorithm for the optimization of process parameters of CNC drilling operation. Reference [14] used hybrid genetic algorithm for hole making operation. Reference [13] considered airtime as a factor in controlling network traffic flow to and from client devices via a wireless network interface.

In the present work, an attempt has been made to improve the efficiency of optimization techniques by hybridizing of GA for minimization of non-productive time. For evaluating the performance of these HGAs three different complexity jobs named as hard, medium and easy problem has been considered. This separation of jobs is based upon number of retraction points or number of nodes. The results are also compared with simple GA using statistical technique relative percentage deviation (RPD) index.

II. Problem Formulation

In a milling operation, when numerous regions have to machine with a small size tool, the tool has to retract and reposition several times. The problem is formulated as:

Let P be the entry/exit point of one region of tool path. K_{ij} be the Non looping constraint with value 1 or 0. If tool is moving from point '*i*' to' *j*' than $K_{ij} = 1$ otherwise $K_{ij} = 0$.

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} P_{ij} K_{ij} \tag{1}$$

$$\sum_{i=1}^{n} K_{ij} = 1 \tag{2}$$

$$\sum_{i=1}^{n} K_{ji} = 1 \tag{3}$$

The distance between i_{th} and j_{th} nodes is P_{ij}. The equation (1) shows total length of path traveled by the tool. Therefore, this equation represents the fitness function of the non productive machining time problem.

A. Cost Mechanism of Machining

The total machining cost comprises of the tooling cost and the machining cost as shown in (7). The time consumed during machining (T) includes the time taken during rapid movement of the tool, setup time (t_{setup}) and time spent for actual machining. Besides this in case of multi tooling job tool change time (t_{change}) also include in total machining time (T). Total machining cost can be calculated as [14] shown in (4)-(6).

$$\mathrm{T} = \left[\frac{l_{machining}}{\mathrm{feed}_{machining}} + \frac{l_{non-productive}}{\mathrm{feed}_{rapid}} + t_{change} + t_{setup} \right] \tag{4}$$

$$\mathrm{C}_{\mathrm{machine}} = \frac{\left(\mathrm{T} + t_{change}\right) \times h}{60} \tag{5}$$

$$C_{tool} = \frac{l_{machining}}{feed_{machining}} \left(\frac{x}{t_{life}} \right) \tag{6}$$

$$C_{total} = C_{tool} + C_{machine} \tag{7}$$

The 'x' and 'h' represents the cost of tool and machining cost respectively. The present study emphasizes on minimization of machining time 'T' by minimizing the non-productive movements of the tool and all other factors are considered as constant.

III. Proposed Algorithm

A. Simple Genetic Algorithm

A Genetic algorithm is the optimization technique which can be applied to various problems, including those that are NP-hard [3]. The algorithm does not ensured an optimal solution; however it usually provides good approximations in a reasonable amount of time as compared to exact algorithms. It must be initialized with a starting population. Generating of initial population may be varied: feasible only, randomized, using some heuristics etc.

The non-productive time has been minimized by optimizing the non-productive movements during 2.5D milling operation using simple genetic algorithm (GA). The algorithm works on probabilistic selection as a basis 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 constrained mentioned. Best solution is selected from the population as evaluated by fitness function. This best solution is termed as elite solution. The new population is again passed from the same process and the process is repeated to calculate best solution. The process remains continue till the stopping limit has not been achieved. Following parameters are considered for the running of SGA.

TABLE I
PARAMETERS FOR GENETIC ALGORITHM

Parameter	Value
Population size	20,40,60,80 and 100
Crossover function	PMX with fraction 0.8
Mutation function	RX with fraction 0.15
Elite count	2
Stopping criteria	600 sec

B. Proposed Hybrid Genetic Algorithm (HGA)

As the toolpath optimization problem is NP hard and there is a large search space of possible toolpath depending upon the number of nodes for finding optimal solutions. In genetic algorithm the randomly generated initial solutions probably provide relatively weaker results than the special heuristic based initial solutions with same stopping limit of specified time. Therefore, to improve the efficiency of SGA an initial solution has been developed using special heuristics which is inspired by [11] and termed as hybrid genetic algorithm (HGA). However, the HGA is similar to simple genetic algorithm with only difference of generation of one initial solution using proposed special heuristics as shown in Fig. 1.

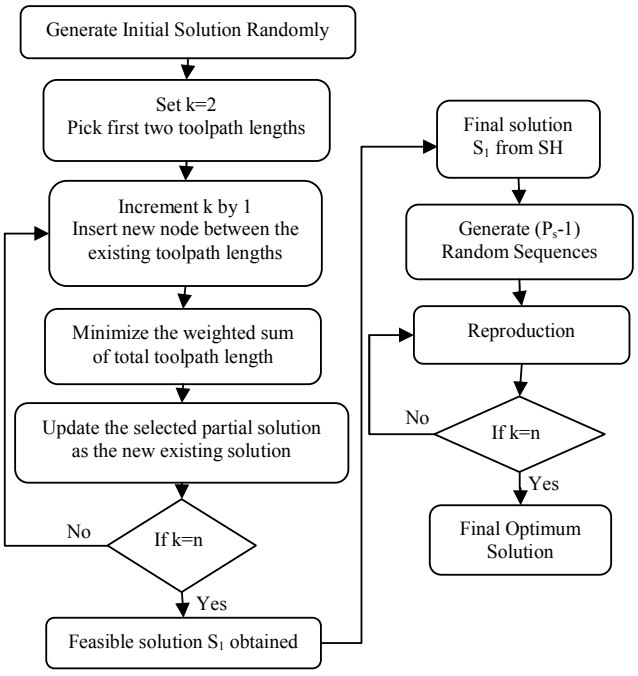


Fig. 1 Hybrid algorithm using SH and GA

The special heuristics is designed so that it might give a good quality initial solution with low processing efforts. The initial sequence of the nodes has been selected randomly and the first two nodes have been selected from the initial sequence with the value of count k=2. Increase k by 1 and Generate candidate sequence by introducing the new node in the residual sequence into each slot of the existing solution. Amongst these nodes, select the better one with the least partial minimization of the toolpath length. Update the selected partial solution as the new existing solution and reposition them in order to minimize the weighted sum of total toolpath length as if there are only two lengths. Set the better as the existing solution. The iterations get stopped for $k = n$ as shown in Fig. 1, where n is the total number of nodes. Further the sequence obtained is fed to the genetic algorithm. This initial sequence S_1 is combined with the randomly generated population of size (P_s-1) and P_s solutions are further be processed in different stages of genetic algorithm and parameters for the HGA remains same as explained in Table I.

C. Hybrid Simulated Annealing Genetic Algorithm (HSAGA)

It has been commonly accepted that finding optimality to NP hard problems is not a viable option since large amount of computational time is needed for judgment of such solutions. As discussed earlier, a good initial solution can be obtained by a heuristic in a reasonable computational time. In this part of work, initial solution is generated using simulated annealing (SA) and the solution is inserted with the randomly generated initial population for GA. However, both the techniques are applied for optimization of different kind of problems, since the chances of entrapping in local solution of GA are less as compared to SA [16]. Therefore, such type of hybridization of GA with simulated annealing might be faster and more efficient.

The SA algorithm is an iterative search procedure based on a neighborhood structure. In this algorithm firstly for simulated annealing, create initial solution and set initial temperature as shown in Fig. 2. A neighboring solution has been generated randomly and the cost of the new solution has been compared with the current solution. If the cost decreases, the current solution is replaced by the generated neighborhood solution. Otherwise, the current solution is replaced with the new neighborhood solution with some probability, it is generated using probability Boltzmann function specified in (9) and the same steps are repeated. After the new solution is accepted, inner loop is checked. If the inner loop criterion is met, the value of temperature is decreased using a predefined cooling schedule. Otherwise, a new neighborhood solution is regenerated and the same steps are repeated.

The neighbor-hood move rule and then the cost between neighborhood solution & current solution can be calculated with (8) and (9).

$$\Delta C = C_o - C_{new} \tag{8}$$

where, ΔC represents the change of amount between costs of the two solutions. C_o and C_{new} represent current solution and neighborhood solution, respectively.

$$Q = e^{-\Delta C / t} \tag{9}$$

The 'Q' represents probability of acceptability and 't' is the temperature considered. The initial seed solution of GA has been generated with the help of SA and the hybridization is termed as HSAGA. The searching is repeated until the

termination criteria is set as 300 sec. are met and then this solution is used as the starting solution for genetic algorithm. The parameters considered for HSAGA are tabulated in Table II.

TABLE II
PARAMETER FOR SIMULATING ANNEALING

Parameter	Value
Initial temperature	200
Annealing function	Modified N.C.T.P
Temperature function	Cooling rate 0.95
Acceptance function	Boltzmann probability function
Renewal interval	100
Stopping criteria	600 sec

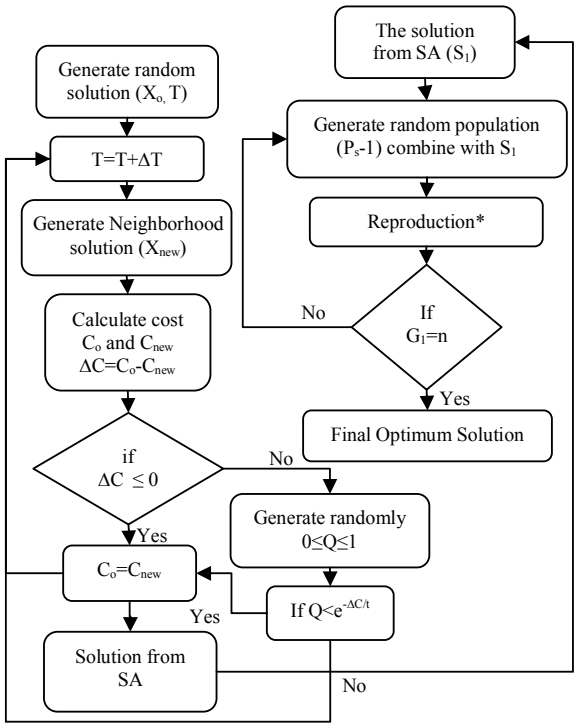


Fig. 2 Hybrid Simulated Annealing Genetic Algorithm (HSAGA)

IV. RESULTS AND DISCUSSIONS

In this study, three type of problems (Easy, medium and hard) have been considered for tool path optimization. The codes for simple genetic algorithm (SGA), hybrid genetic algorithm (HGA) and hybrid simulated annealing genetic algorithm (HSAGA) have been developed in MATLAB. These codes have been executed on a personal computer with a 2.30 GHz Intel core i3-235OM Processor with 2 GB of RAM. The algorithms are applied on three different sizes of problem. The results obtained by proposed algorithms are compared with SGA for minimization of non-productive time during machining for different sizes with same stopping limit of time of 600 seconds. In order to improve the performances of these genetic algorithms the non-productive time has been optimized at varying population sizes.

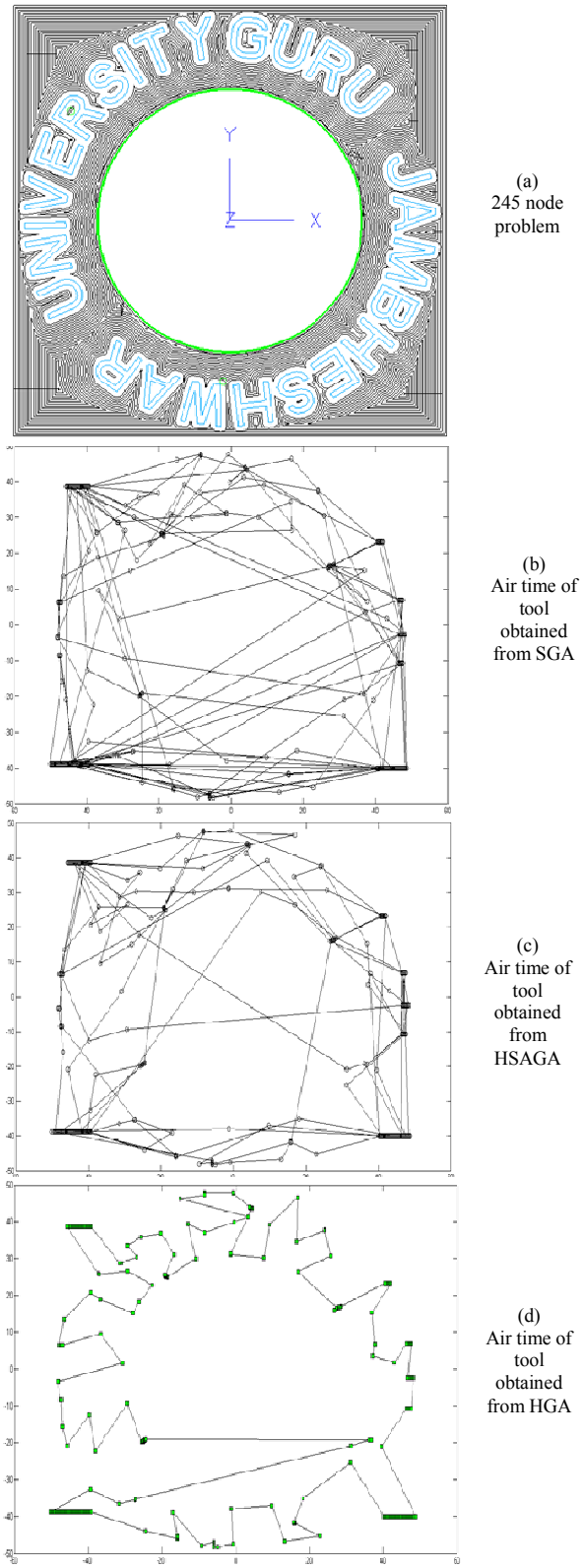
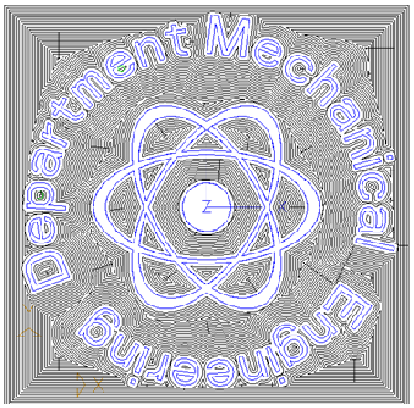
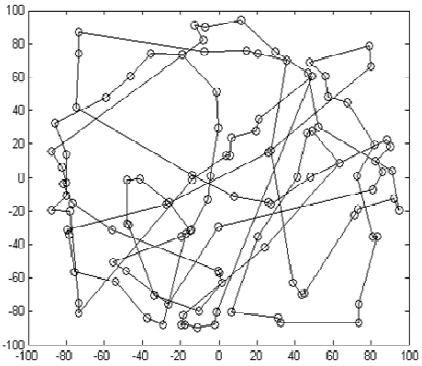


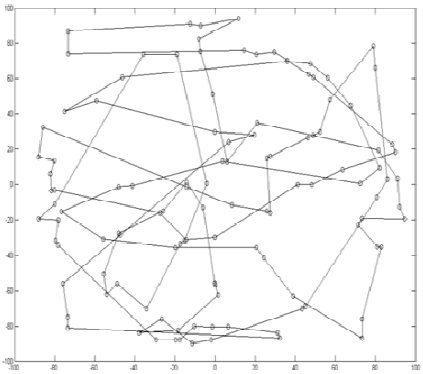
Fig. 3 Solution of Hard Problem using Different Heuristics



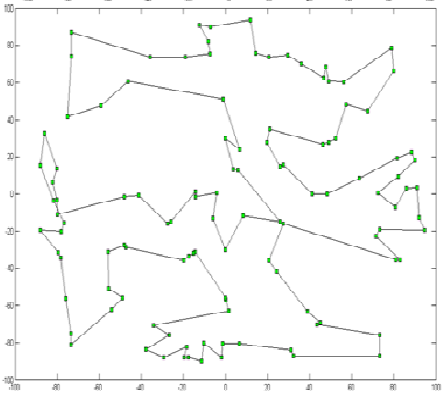
(a) 150 node problem



(b) Air time of tool obtained from SGA



(c) Air time of tool obtained from HSAGA



(d) Air time of tool obtained from HGA

Fig. 4 Solution of Medium Problem using Different Heuristics

The three types of problems for Milling of a post sign have been considered on the basis of number of retraction points. These problems with 245 nodes and 150 nodes are illustrated in Figs. 3 and 4 are termed as easy, medium, and hard problem.

The contour parallel toolpath has been applied on the three problems with different complexity levels. The tool is being run with 560 MMPM rapid speed. By increasing the tool diameter for 245 retraction points hard problems the number of nodes have been reduced to 79 and the problem is termed as easy problem. However, problem with 150 retraction point is relatively medium problem.

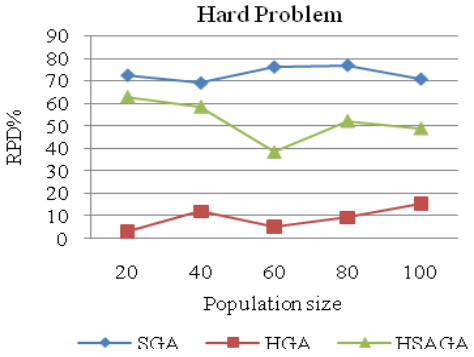


Fig. 5 RPD index for Hard Problem

The results of HGA, HSAGA and SGA are being compared with the help of relative percentage deviation (*RPD%*) by relation defined as:

$$RPD\% = \frac{Method_{SOL} - Best_{SOL}}{Best_{SOL}} \quad (10)$$

$Best_{Sol}$ 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, HSAGA and SGA on 245 nodes, 150 nodes and 79 nodes have been shown in Figs. 5-7.

The results are obtained for easy, medium and hard types of problems using the three different heuristics with different population sizes and stopping limit of 600 seconds. It has been found that HGA with population size 20 provides minimum non-productive tool path length 2745 mm in 2926 generations for hard problem. However, SGA and HSAGA give comparatively higher tool path length for each population size and higher RPD% as shown in Fig. 5. Similarly for medium problem as shown in Fig. 6, HGA with population size 60 provides low RPD% and minimum non cutting tool path length 2665.67 mm in 5453 generations. The SGA and HSAGA also give comparatively higher RPD% and greater tool path length for easy problem with all population sizes as shown in Fig. 7. Similarly after comparing all results for different sizes i.e. 40, 60, 80 and 100, the Proposed algorithm i.e. Hybrid Genetic Algorithm (HGA) shows superiority over SGA by 40% and HSAGA is also performed 20% better than SGA for same stopping criteria.

As the problem gets harder, this difference enlarges. The fitness values of SGA solution are shown in Fig. 3 (b). The

convergence rate of the fitness value decreases with the number of generations increases. However, HGA and HSAGA provide only the approximate solution.

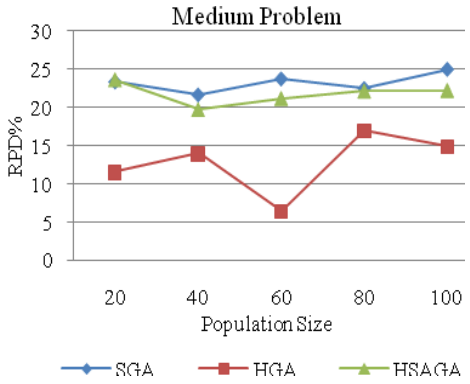


Fig. 6 RPD index for Medium Problem

In HSAGA solution, the result from SA solution after 300 seconds is taken as the input to the SGA. This improves the best local solution as shown in Fig. 3 (c). In HGA, initial solutions are produced with the help of heuristic and the best solution obtained in this algorithm is shown in Fig. 3 (d).

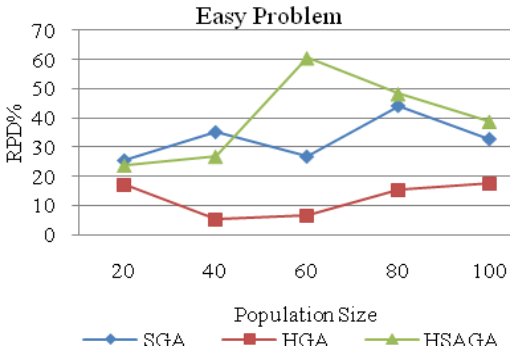


Fig. 7 RPD index for Easy Problem

The performance of proposed HGA and HSAGA gives better results than SGA for all the three problems with RPD of 3.00% and 6.43%, 5.20 % and 38.51%, 19.74% and 23.70% respectively. It has also been found that, performance of HGA and HSAGA varies with increase in population size. Fig. 7 clearly illustrate that the value of % RPD of HGA and HSAGA is found better as compared to SGA irrespective of all population sizes and all types of problems considered. For the population size of 60, HGA and HSAGA showed best result for hard and medium problem.

V. Conclusions

In the present work, proposed algorithm i.e. Hybrid Genetic Algorithm (HGA) shows superiority over SGA by 40% under the application of same stopping criteria. As the population size increases SGA requires more time to perform the iterations; however, the proposed HGA has been found more effective as it performs better with lesser population size. For the population size 60, the HGA provides optimal results for any type of problem considered. HSAGA is also performed 20% better than SGA for all the three types of problems; however it stands next to HGA. Therefore, it can be concluded that the hybridization of genetic algorithm effectively reduces the computational effort and the suitable population size also affects the performance of GA.

References

[1] A. K. Gupta, P. Chandna and P. Tandon "Hybrid genetic algorithm for minimizing non productive machining time" *International Journal of Engineering, Science and Technology*" vol. 3, no. 1, pp. 183-190, 2011.
[2] C. Oysu, Z. Bingul, "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, 2009.
[3] D. E. Goldberg, "Genetic algorithm in search, optimization & machining" *Pearsons Publication*, 2008.
[4] F. Kolahan and M. Liang, "Optimization of hole-making operations: a tabu- search approach", *International Journal of Machine Tools & Manufacture*, vol. 40, no. 12, , pp. 1735–1753, 2000.
[5] H. K. Tsai, J. M. Yang, Y.F. Tsai and C. Y. Kao, "An Evolutionary Algorithm for Large Traveling Salesman Problems", *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 4, pp. 1718-1729, 2004.
[6] H. Ghaiebi and M. Solimanpur, "An ant algorithm for Optimization of hole-making operations". *Computers & Industrial Engineering*, vol. 52, pp. 308–319, 2007.
[7] K. Castelino, R. D'Souza and P. K. Wright. "Tool-path Optimization for Minimizing Airtime during Machining", *Journal of Manufacturing Systems*, vol. 22, no. 3, pp. 173-180, 2003.
[8] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents" *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, vol. 26, no. 1, pp.1-13, 1996.
[9] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 308 – 313, 1997.
[10] N. L. J. Ulder, E. H. L. Aarts, H.J. Bandelt, P. J. M. V. Laarhoven, and E. Pesch, "Genetic local search algorithms for the traveling salesman problem", *Lecture Notes in Computer Science*, vol. 496, pp.109-111.1991.
[11] Nawaz, Muhammad Enscore Jr, E Emory Ham and Inyong, "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, 1983.
[12] P. Jog, J. Y. Suh, " Parallel genetic algorithms applied to the Traveling Salesman Problem", *SIAM Journal of Optimization*, vol. 1, pp. 515–529, 1991.
[13] P. Wu, Sreekanth Reddy, Jianlin Zeng and Changming Liu, "Airtime based packet scheduling for wireless networks", *United State Patent pat.*, no. 8,483,194, 2014.
[14] R. D'Souza, P. Wright and C. Skquin, "Automated Micro planning for 2.5-D Pocket Machining", *Journal of Manufacturing Systems*, vol. 20, no. 4, pp. 288–296. 2001.
[15] R. Phipon and B.B. Pradhan"Process parameters optimisation of CNC drilling operation using genetic algorithm," *International Journal of Machining and Machinability of Materials*, vol. 12, no. 1, pp. 54-65, 2012.
[16] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by Simulated Annealing", *Science is currently published by American Association for the Advancement of Science*, vol. 220, no. 4598, pp. 671 – 680, 1983.
[17] W.A. Khan, D.R. Hayhurst and C. Cannings, "Determination of optimal path under approach and exit constraints". *European Journal of Operational Research*, vol. 117, no. 2, pp. 310-325, 1999.
[18] Y. Wang & J. Qin, "A memetic-clustering-based evolution strategy for traveling salesman problems", *Proceedings of the 2nd international conference on Rough sets and knowledge technology,* 2007.