Optimization of Milling Operation using Genetic and PSO Algorithm

U. Deepak

Abstract--- *Metal cutting is one of the important and widely used manufacturing processes in engineering industries. Optimizing the machining parameters has become an essential one in order to be competitive and to meet customer demands quickly. For this purpose several optimization techniques are used. Among those techniques Particle Swarm Optimization and Genetic Algorithm is used in this paper because of its better ability. A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of Evolutionary Algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Such methods are commonly known as metaheuristics as they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. These techniques are used to optimize the machining parameters like depth of cut, feed rate and cutting speed. This will help in better optimization of milling operation. The developed techniques are evaluated with a case study*

.

Keywords--- *Particle Swarm Optimization, Genetic Algorithm, Optimization, Profit Maximization.*

I. Introduction

MILLING is the most common form of machining, a material removal process, which can create a variety of features on a part by cutting away the unwanted material. The milling process requires a milling machine, workpiece, fixture, and cutter. The workpiece is a piece of pre-shaped material that is secured to the fixture, which itself is attached to a platform inside the milling machine. The cutter is a cutting tool with sharp teeth that is also secured in the milling machine and rotates at high speeds. By feeding the workpiece into the rotating cutter, material is cut away from this workpiece in the form of small chips to create the desired shape [1].

Increasing productivity, decreasing costs, and maintaining high product quality at the same time are the main challenges manufacturers face today. The proper selection of machining parameters is an important step towards meeting these goals and thus gaining a competitive advantage in the market [4]. Many researchers have studied the effects of optimal selection of machining parameters of end milling [5]. It can be formulated and solved as a multiple objective optimization problem [6]. In practice, efficient operation of milling operation requires the simultaneous consideration of multiple objectives, including maximum tool-life, desired roughness of the machined surface, target operation productivity, metal removal rate, etc [7]. In some instances, parameter settings that are optimal for one defined objective function may not be particularly suited for another objective function. Traditional optimization methods are difficult and the only way is to reduce the set of objectives in to a single objective and handle it accordingly. Therefore evolutionary algorithms such as genetic algorithms (GA) and particle swarm optimization (PSO) are more convenient and usually utilized in multiobjective optimization problems. These methods are summarized by [8]. The PSO is an efficient alternative over other stochastic and population-based search algorithms, especially when dealing with multi-objective optimization problems. It is relatively easy to implement and has fewer parameters to adjust compared to genetic algorithms. As mentioned above, neural networks are used to model complex relationships in the process, and an integrated system of neural networks and particle swarm optimizer is utilized in solving multi-objective problems observed in milling operations.

II. Mathematical Model

The various parameters like cutting speed, feed rate and depth of cut. These are the essential machining parameters are used. Essentially the Depth of cut, feed rate, and Cutting speed have the greatest effect for the success of the machinining operation. Depth of cut is usually based on the operation sequence and work piece [2]. The workpiece with the required depth in one pass to keep the machining time, cost and low when possible. The problem of determining the minimization of the machining parameter is to reduce the cutting speed and feed rate combination.

A. Profit Rate

The main aim of this work is to maximize the profit rate criterion with the respect to the various operations like face milling, corner milling, pocket milling, slot milling. The formula used for calculating the total profit rate is given by M. Tolouei-Red et al. [12]

$$P_r = \frac{S_p - C_u}{T_u} \qquad (1)$$

U. Deepak, Kongu Engineering College, Erode, Tamilnadu, India, E-mail: kdeepak.mts@gmail.com

The unit cost can be represented by [3]

$$C_u = C_{mat} + (C_l + C_o)t_s + \sum_{i-1}^{m}(C_l + C_o)K_{1i}V_i^{-1}f_i^{-1} + \sum_{i=1}^{m} C_{ti}K_{3i}V_i^{\left(\frac{1}{n}\right)-1} f_i^{\left[\frac{(w+g)}{n}\right]-1} + \sum_{i=1}^{m}(C_l + C_o) \qquad (2)$$

$$T_u = ts + \sum_{i-1}^{m} K_{1i}V_i^{-1}f_i^{-1} + \sum_{i=`}^{m} t_{tci} \qquad (3)$$

Where

P_r	denotes the profit rate
S_p	denotes the selling price
C_u	notes the unit cost
T_u	denotes the unit time

B. Constraints

The range of cutting speed and feed rate can be limited by the following constraints [13]:

- Maximum machine power
- Surface finish Requirement
- Maximum cutting force permitted by the rigidity of the tool
- Available feed rate and spindle speed on the machine tool

C. Power

Maximum machine power is used based on the machining parameters .The machining power should be less than the calculated motor power. The power constraint is given by

$$C_5 = Vf^{0.8} \leq 1 \qquad (4)$$

Where

$$C_5 = \frac{0.78K_pW_{za_{rad}a}}{60\pi deP_m}Vf^{0.8} \qquad (5)$$

D. Surface Finish

Similarly the surface finish operation should be less than the attainable surface finish. The surface finish is used for the end milling operation is given by

$$C_6 f \leq 1 \qquad (6)$$

where

$$C_6 = \frac{318[\tan(l_a) + \cot(c_a)]^{-1}}{R_{a(at)}} \qquad (7)$$

and for the end milling operation

$$C_7 f^2 \leq 1 \qquad (8)$$

where

$$C_7 = \frac{318(4d)^{-1}}{R_{a(at)}} \qquad (9)$$

E. Cutting Force

Similarly cutting force f_c should be less than the permitted cutting force $F_c(per)$ the permitted cutting force is considered for the each tool based on the maximum limit. The equation is

$$C_8F_c \leq 1 \qquad (10)$$

where

$$C_8 = \frac{1}{F_c(per)} \qquad (11)$$

where

Fc (per) denotes the permitted cutting force

F. Speed Limits

There are various speed limits with the respect to the various operations like

- Face milling : 60-120 m/min
- Corner milling: 40-70 m/min
- Pocket milling: 40-70 m/min
- Slot milling 1: 30-50m/min
- Slot milling 2:30-50 m/min

G. Feed Rate Limits

There are various feed rate limits with the respect to the various operations like

- Face milling :0.05-0.4 mm/tooth
- Corner milling:0.05-0.5 mm/tooth
- Pocket milling: 0.05-0.5 mm/tooth
- Slot milling1 :0.05-0.5 mm/tooth
- Slot milling 2:0.05-0.5mm/tooth

III. Methodology

There are various traditional approaches used for solving the optimization techniques for solving the machining problem. The mainly used the nontraditional optimization techniques like genetic algorithm, genetic algorithm and particle swarm optimization .These are the essential approaches used for calculating the optimal results based on the cutting speed, feed rate, unit cost and unit time. The results are determined using these approaches with help of the software package like c program.

IV. Case Study

The component as show in fig 1 is to produce using CNC milling machine. The give data is constant.

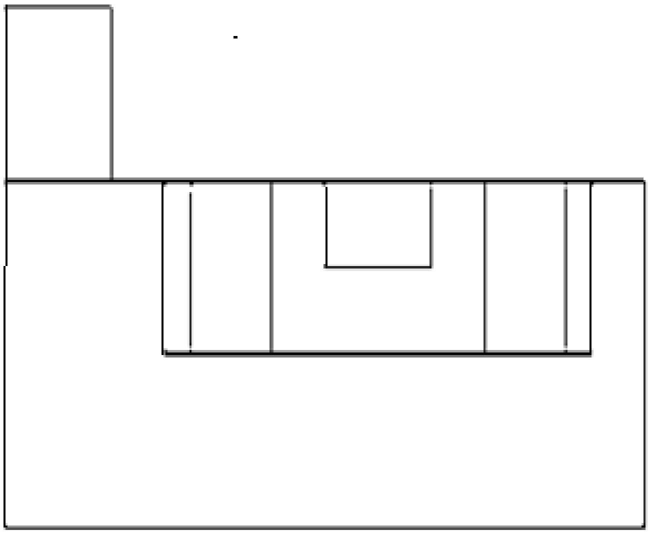
S_p= $25

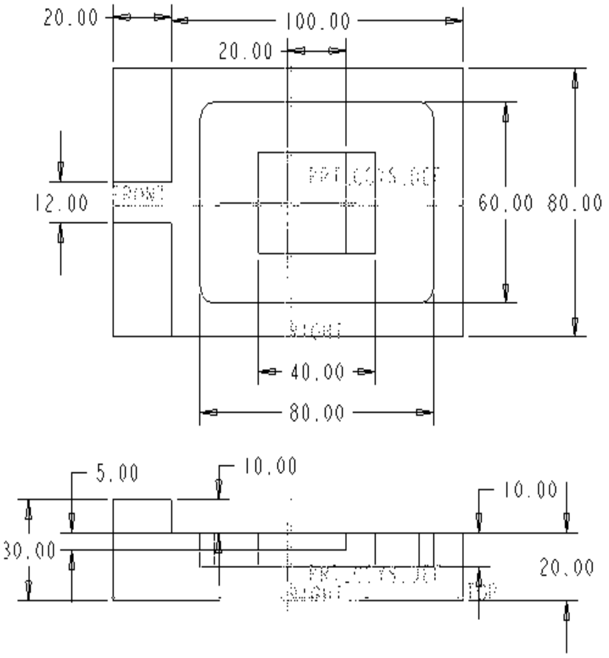
C_{mat}= \$0.50
C_o=\$1.45 per min
$C_{1=}$\$ 0.45 per min
T_s= 2 min
T_{ct}= 0.5 min
C= 33.98 for HSS tools
w= 0.28
C= 100.05 for carbide tool
K_p= 2.24
W= 1.1
n= 0.15 for HSS tool
n=0.3 for carbide tool
g= 0.14
Machine tool data: Type: Vertical CNC milling machine Pm = 8.5 kW and e= 95%
Material data: Quality: 10 l50 leaded steel. Hardness =225 BHN

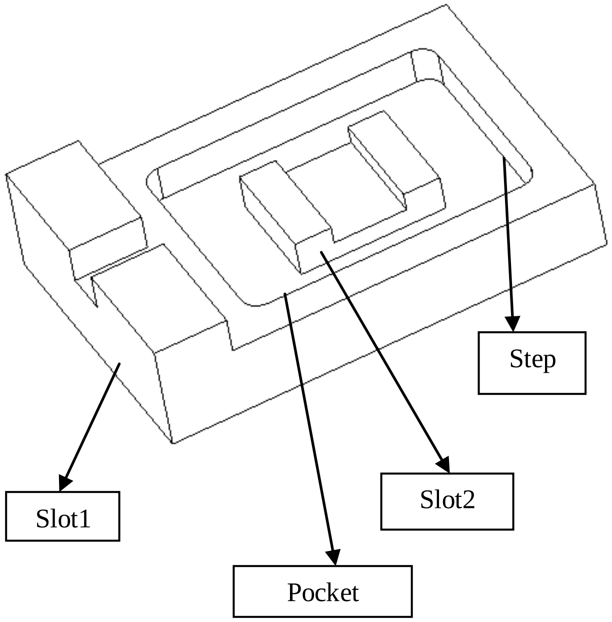
Where

S_p	Sale price of the product(\$)
C_{mat}	Machine cost, cost of raw material per part, cost of a cutting tool (\$)
C_o, C_1	Labour cost, overhead cost (\$/min)
T_s ,T_{ct}	Tool life, unit time
C	Constant in cutting speed equation
w	Exponent of chip cross sectional area
K_p	Power constant depending on the workpiece material
W	Tool wear factor
n	Tool life exponent
g	Slenderness ratio exponent of slenderness ratio

Fig 1







V. Optimization Procedure

To determine the optimum parameters, the following optimization methods are used:

1. Genetic Algorithm
2. Particle Swarm Optimization

a. Genetic Algorithm

Three fundamental operations are used in the genetic algorithm are used to be described as follows they are

- Reproduction

- Crossover
- Mutation

b. Method used in Reproduction

- Rank order method is used as a reproduction operator. In these method ideas, rank is given according to the fitness value of each chromosome.
- The probability values for selection is given according to the expected value

$$(i,t) = min + (max - min)(Rank(I,t) - 1)/N - 1$$

c. Crossover

In reproduction, good strings in a population are probabilistically assigned a large number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In crossover; a new strings are created by exchanging the information among the strings of the mating pool.

Steps in crossover

- Select the random number between 1 and 19 and perform crossover operation after that point.
- For example, random number 7 is generated. This means cross over is performed between first and second chromosome after 7th bit
  1.0000111111-0001000100
  2.1101001000-0000111000

d. Mutation

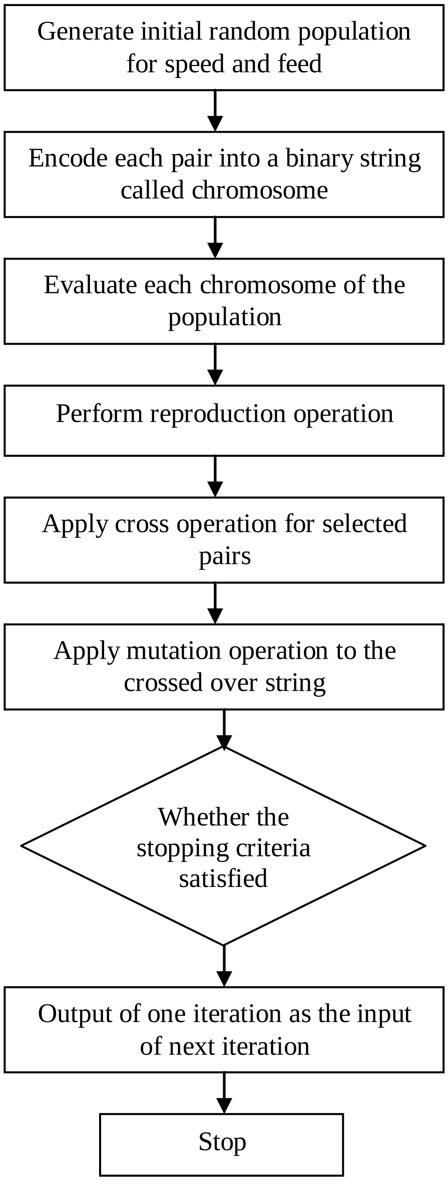
The Mutation operator changes 1or 0 and vice versa with a small mutation probability pm. Flipping the coin with the probability pm performs the bit-wise mutation. If at any outcome is true the bit is altered otherwise the bit is unchanged.

Before mutation: 1101001000-0000111000
After Mutation: 1100001000-0000111000

Algorithm for the Genetic Algorithm

Step 1	Choose a coding to represent problem parameters,a selection operator, a crossover operator and a mutation operator. Choose population size n, crossover probability Pc. Initially a random population of string of size 1.choose a maximum allowable number tmax set t=0
Step 2	Evaluate each string in the population
Step 3	if t> tmax or other termination criteria is satisfied.
Step 4	Perform reproduction on the population
Step 5	Perform crossover on random pairs of strings Step 6: perform bit-wise mutation
Step 6	Perform bit-wise mutation
Step7	Evaluate string in the new population set t=t+1 and goto step3 End.

Flowchart for the Genetic Algorithm:


Generate initial random population for speed and feed
Encode each pair into a binary string called chromosome
Evaluate each chromosome of the population
Perform reproduction operation
Apply cross operation for selected pairs
Apply mutation operation to the crossed over string
Whether the stopping criteria satisfied
Output of one iteration as the input of next iteration
Stop

Output of C program using genetic Algorithm

```
Cross over f= 0.365000   cross over v=54.500000
Crossover f m=1    cross v=55
Cross over f= 0.275000 Cross over v=47.500000
Crossover f m=1 cros v=46
Cross overf= 0.410000 Cros v=58.000000
Cross overf m=1 Cross v=59
Crossover f=0.140000 cross v= 37.000000
Cross over f m=1 cross v=36
Sum1= 19.720317   sum 2=0.000004 sum 3= 10.379114 cu=
43.020321 tu= 17.379114
The maxf=1.000000 max v=60.00000 minf=1.00000000 min
v= 32.0000000
Cross over f = 1.000000      Cross v=43.200000
Cross over f m=0 cross v=42
Cross over f=1.000000 cross v= 57.200000000
Cross over f m=0, cross over v= 56
```

```
Cros, s over f= 1.000000 cross v=48.8000000
Cross over f m=0 cross v=49
Cross over f = 1.000000      cross v=34.800000
Cross over f m=0 cross over v=35
Cross over f=1.000000 cross v=51.6000000
Cross over f m=0, cross v=50
Cross over f = 1.000000      cross v=46.0000000
Cross over f m=0, cross v=47
Cross over f = 1.000000      v= 54.4000000
Cross over f m=0, cross v=55
Cross over f = 1.000000      cross v=37.6000000
Cross over f m=0, cross v= 36
```

Particle Swarm Optimization

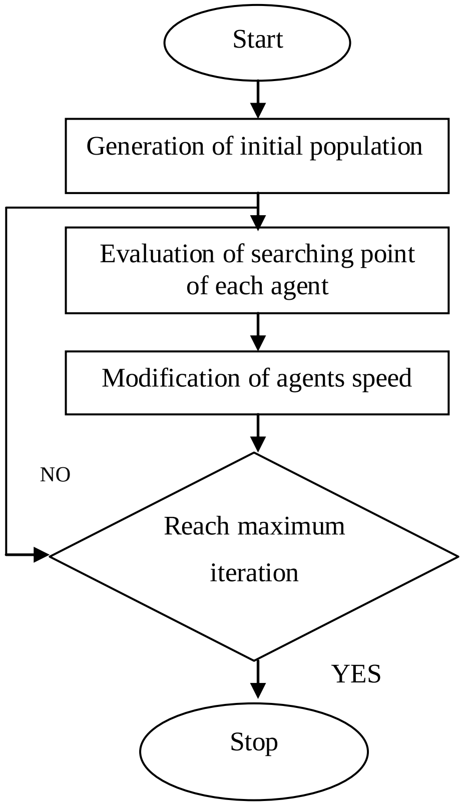
Particle Swarm Optimization (PSO) is a relatively new technique, for optimization of continuous non-linear functions [11]. It was first presented in 1995. Jim Kennedy discovered the method through simulation of a simplified social model, the graceful but unpredictable choreography of a bird swarm [12]. PSO is a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, so is computationally in expensive in terms of both memory requirements and speed. PSO has been recognized as an evolutionary computation technique [13] and has features of both genetic algorithms (GA) and evolution strategies (ES). Other evolutionary computation (EC) techniques such as genetic algorithm (GA) also utilize some searching points in the solution space. It is similar to a GA in that the system is initialized with a population of random solutions. While GA can handle combinatorial optimization problems, PSO can handle continuous optimization problems. However, unlike a GA each population individual is also assigned a randomized velocity, in effect, flying them through the solution hyperspace. PSO has been expanded to handle also the combinatorial optimization problems and both discrete and continuous variables as well. Unlike other EC techniques, PSO can be realized with only small program. Namely PSO can handle mixed-integer nonlinear optimization problems with only small program. The feature of PSO is one of the advantages compared with other optimization techniques. Natural creatures sometimes behave as a swarm. One of the main goals of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer.

Basic of particle swarm optimization

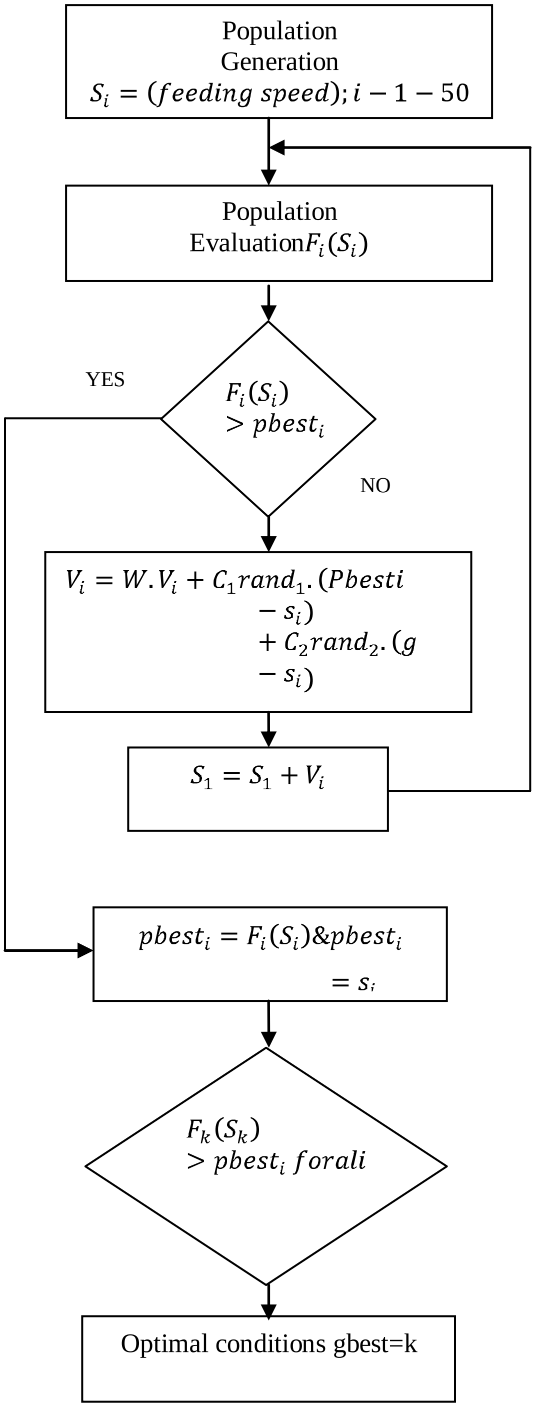
According to the background of PSO and simulation of swarm of bird, researchers [12] developed a PSO concept. Namely, PSO is basically developed through simulation of bird flocking in two- dimension space. The position of each agent is represented by XY axis position and also the velocity is expressed by vx (the velocity of X axis) and vy (the velocity of Y axis). Modification of the particle position is realized by the position and velocity information. Bird flocking optimizes a certain objective function. Each agent (particle) knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among (pbests). This information is analogy of knowledge of how the other agents around them have performed. Each agent tries to modify its position using the following information: - the current positions (x, y), - the current velocities (vx, vy), - the distance between the current position and (pbest) - the distance between the current position and (gbest). This modification can be represented by the concept of velocity. The general flow chart of PSO can be described as follows:

Step 1	Generation of initial condition of each agent. Initial searching points (S_i^0) and velocities (v_i^0) of each agent are usually generated randomly within the allowable range.
Step 2	Evaluation of searching point of each agent. The objective function value is calculated for each agent. If the value is better than the current pbest of the agent, the pbest value is replaced by the current value.
Step 3	Modification of each searching point.
Step 4	Checking the exit condition. Otherwise, go to step 2.

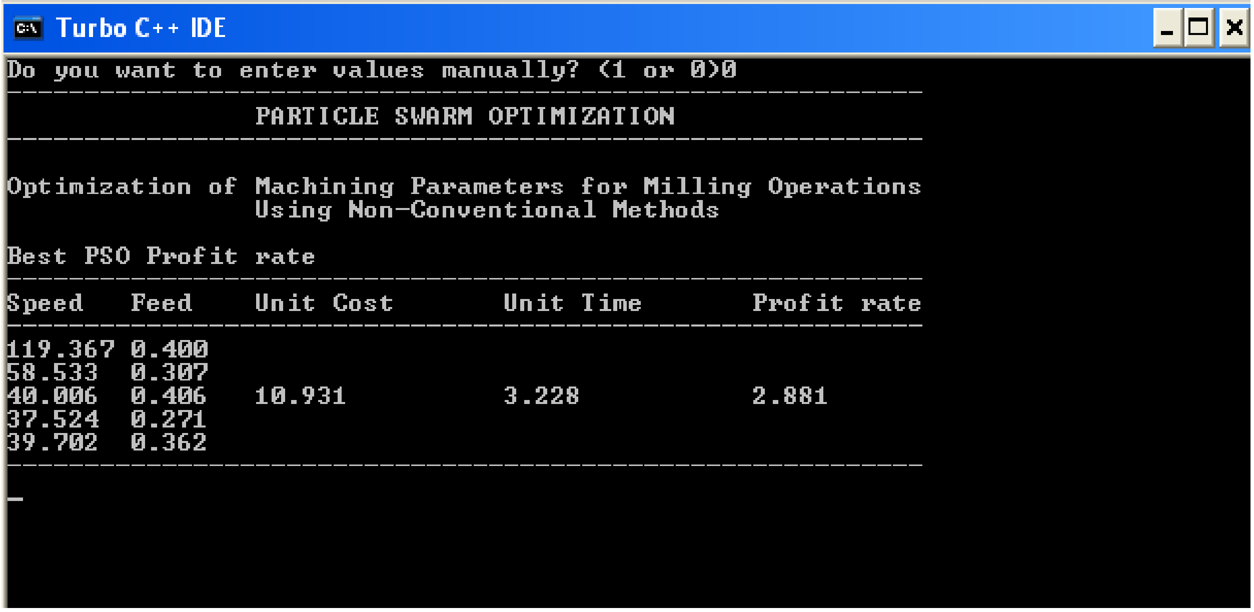
The General Flow Chart of PSO Strategy


Start
Generation of initial population
Evaluation of searching point of each agent
Modification of agents speed
NO
Reach maximum iteration
YES
Stop

PSO Algorithms for Optimization


Population Generation $S_i = (feeding\ speed); i - 1 - 50$
Population Evaluation$F_i(S_i)$
YES
$F_i(S_i) > pbest_i$
NO
$V_i = W.V_i + C_1 rand_1.(Pbesti - s_i) + C_2 rand_2.(g - s_i)$
$S_1 = S_1 + V_i$
$pbest_i = F_i(S_i)\&pbest_i = s_i$
$F_k(S_k) > pbest_i\ forali$
Optimal conditions gbest=k

Output of PSO



```
Do you want to enter values manually? (1 or 0)0
---------------------------------------------------------------
                PARTICLE SWARM OPTIMIZATION
---------------------------------------------------------------

Optimization of Machining Parameters for Milling Operations
                Using Non-Conventional Methods

Best PSO Profit rate
---------------------------------------------------------------
Speed   Feed    Unit Cost        Unit Time        Profit rate
---------------------------------------------------------------
119.367 0.400
58.533  0.307
40.006  0.406   10.931           3.228            2.881
37.524  0.271
39.702  0.362
---------------------------------------------------------------
_
```

VI. Results and Conclusion

Thus the C program using the optimal results by the genetic algorithm is executed. This work also executes the C program using Particle swarm optimization to determine and analyses the optimal solution. This paper helps in determining the better optimization of milling operation parameters with the help of Genetic Algorithm and PSO algorithm

References

[1] M. Tolouei-Rad and I.M. Bindhendi, on the Optimization of Machining Parameters For Milling Operations, Int.J.Mach Tools Manufact, Vol.37, No.1, Pp. 1-16, 1997

[2] N. Baskar, P. Asokan, R. Saravanan and G. Prabhaharan, Optimization of Machining Parameters for Milling Operation Using Non-Conventional Method, Int J Adv Manuf technol (2005)

[3] N. Baskar, P. Asokan, R. Saravanan and G. Prabhaharan, Selection of Optimal Machining Parameters for Multi-Tool Milling Operation using a Memetic Algorithm, Journal of material processing Technology 174(2006).

[4] U. Zuperl, F. Cus, M. Milfelner, Fuzzy control strategy for an adaptive force control in end-milling, Journal of Materials Processing Technology 164 (2005) 1472-1478.

[5] J. Kopac, Influence of high speed cutting on the structure of machined high speed steel material, The Eleventh Scientific Conference on Contemporary Achievements in Mechanics, Manufacturing and Materials Science CAM3S'2005, Gliwice-Zakopane, 2005, 40-44.

[6] F. Cus, U. Zuperl, E. Kiker, M. Milfelner, Adaptive controller design for feedrate maximization of machining process, Journal of Achievements in Materials and Manufacturing Engineering 17 (2006) 237-240.

[7] J. Balic, A new NC machine tool controller for step-by-step milling, Internation al Journal of Advanced Manufacturing Technology 18 (2001) 399-403.

[8] J. Balic, Optimization of cutting process by GA approach, Robotics and Computer Integrated Manufacturing 19 (2003) 113-121.

[9] Tolouei-Red et.al (1997) on the optimization of machining parameters for milling operation. Int J Mach Tool Manuf 37(1)-16

[10] Ermer DS (1971), " Optimization of the Constrained machining economics problem by geometric programming" , Trans ASME J Eng Ind 93:1067–1072

[11] Y. Shi, R. Eberhart, Parameter selection in particle swarm optimization, In Evolutionary Programming VII: Proc. EP98, New York: Springer-Verlag, 1998, 591-600.

[12] Y.Y. Peng, A Discrete Particle Swarm algorithm for optimal polygonal approximation of digital curves, Journal of Visual Communication and Image Representation 15 (2004) 241-260.

[13] E. Ozcan, C. Mohan, Analysis of a simple Particle Swarm Optimization system, Intelligent Engineering Systems Through Artificial Neural Networks 8 (1998) 253-258.