



Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms

R. Venkata Rao^{a,*}, P.J. Pawar^b

^a Department of Mechanical Engineering, S.V. National Institute of Technology, Ichchanath, Surat, Gujarat – 395 007, India

^b Department of Production Engineering, K.K. Wagh Institute of Engineering Education and Research, Nashik – 422 003, Maharashtra, India

ARTICLE INFO

Article history:

Received 7 February 2009

Received in revised form 9 May 2009

Accepted 2 August 2009

Available online 8 August 2009

Keywords:

Multi-pass milling

Process parameters optimization

Artificial bee colony

Particle swarm optimization

Simulated annealing

ABSTRACT

The effective optimization of machining process parameters affects dramatically the cost and production time of machined components as well as the quality of the final products. This paper presents optimization aspects of a multi-pass milling operation. The objective considered is minimization of production time (i.e. maximization of production rate) subjected to various constraints of arbor strength, arbor deflection, and cutting power. Various cutting strategies are considered to determine the optimal process parameters like the number of passes, depth of cut for each pass, cutting speed, and feed. The upper and lower bounds of the process parameters are also considered in the study. The optimization is carried out using three non-traditional optimization algorithms namely, artificial bee colony (ABC), particle swarm optimization (PSO), and simulated annealing (SA). An application example is presented and solved to illustrate the effectiveness of the presented algorithms. The results of the presented algorithms are compared with the previously published results obtained by using other optimization techniques.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

In today's manufacturing environment many large industries have attempted to introduce the flexible manufacturing system (FMS) as a strategy to adapt to the ever-changing competitive market requirement. The flexible manufacturing system involves highly automated and computer controlled machines. Due to high capital and machining costs, there is an economic need to operate these machines as efficiently as possible in order to obtain the required pay back. The success of the machining operation depends on the selection of machining process parameters. Proper selection of process parameters play a significant role to ensure quality of product, to reduce the machining cost, to increase productivity in computer controlled machining processes and to assist in computer aided process planning. However, determination of optimum process parameters of any machining process is usually a difficult task where the following aspects are required: knowledge of manufacturing process, empirical equations to develop realistic constraints, specification of machine tool capabilities, development of effective optimization criteria, and knowledge of mathematical and numerical optimization techniques.

A human process planner selects machining process parameters using his own experience or from the handbooks. But these parameters do not give optimal result. Various optimization strategies and algorithms ranging from elementary numerical search methods to more systematic approaches employing non-traditional techniques for optimization of process parameters in case of single pass milling operation had been reported in the literature. However, as multi-pass operations are often preferred to single pass operations for economic reasons, recent efforts have been directed towards determination of optimal machining conditions for multi-pass operations.

Traditionally, mathematical programming techniques like linear programming, method of feasible direction, dynamic programming and geometric programming had been used to solve optimization problems in milling. However, these traditional methods of optimization do not fare well over a broad spectrum of problem domains. Moreover, traditional techniques may not be robust. Numerous constraints and multiple passes make machining optimization problems complicated and hence these techniques are not ideal for solving such problems as they tend to obtain a local optimal solution.

Considering the drawbacks of traditional optimization techniques, attempts are being made to optimize the machining problem using evolutionary optimization techniques. Evolutionary computation consists of a variety of methods including optimization paradigms that are based on evolution mechanisms such as

* Corresponding author. Tel.: +91 261 2201661; fax: +91 261 2201571.

E-mail address: ravipudirao@gmail.com (R. Venkata Rao).

biological genetics and natural selection. These methods use the fitness information instead of the functional derivatives making them more robust and effective. Most commonly used non-traditional optimization technique is genetic algorithm (GA). However, GA provides a near optimal solution for a complex problem having large number of variables and constraints as in the case of multi-pass milling process optimization. This is mainly due to difficulty in determination of optimum controlling parameters such as population size, crossover rate and mutation rate. Therefore, the efforts are continuing to use more recent optimization algorithms, which are more powerful, robust and able to provide accurate solution. Artificial bee colony (ABC) algorithm developed by Karaboga [1] and Karaboga and Basturk [2,3] is one of the most recent algorithms and no effort has been yet made for optimization of process parameters of any of the machining processes by using this algorithm. Hence, in this paper an attempt is made to apply the artificial bee colony algorithm (ABC). For comparison purpose, other non-traditional methods of optimization such as particle swarm optimization (PSO) and simulated annealing (SA) algorithms have also been tried for optimization of process parameters of multi-pass milling operation. Particle swarm optimization has been successfully applied to some manufacturing processes such as pulsed laser micromachining, electro-chemical machining, friction welding, boring, milling, etc. by various researchers [4–7,19].

2. Review of literature on multi-pass milling process optimization

Milling is the machining process in which the metal is removed by a rotating multiple tooth cutter. Fig. 1 shows the milling operation. As the cutter rotates, each tooth removes a small amount of material from the advancing work for each spindle revolution. The relative motion between cutter and the work piece can be in any direction and hence surfaces having any orientation can be machined in milling. Milling operation can be performed in a single pass or in multiple passes. Multi-pass operations are often preferred to single pass operations for economic reasons and are generally used to machine stocks that cannot be removed in a single pass. Various investigators have presented optimization techniques, both traditional and non-traditional, for optimization of multi-pass milling operation.

Shin and Joo [8] used the dynamic programming optimization method for milling process parameter optimization. However, for

the optimization problem involving large amount of independent parameters with a wide range of values such as the cutting parameters in milling operation, the use of dynamic programming is limited. Wang [9] used a neural network based approach to optimize milling process parameters. However, optimization by using neural networks may often ends in local minima or fails to converge on a result.

Tolouei-Rad and Bidhendi [10] used the method of feasible direction and considered maximization of profit rate as an objective function in milling operation. The feasible solution denotes the local minimum of the problem. However, this local minimum need not be the global one unless the problem is convex programming problem. Optimization model developed in their work was non-convex.

Sonmez et al. [11] studied multi-pass milling operation based on the maximum production rate criterion and used an algorithm adopted from the study of Agapiou [12] which was presented for the multi-pass turning operations. Although the results showed significant improvement over handbook recommendations, the optimization techniques used in their work (dynamic programming and using geometric programming) either tend to result in local minima or take a long time to converge on a reasonable result. Shunmugam et al. [13] used genetic algorithm (GA) for milling process parameter optimization with total production cost as the objective function. Although GA has advantages over the traditional techniques, the successful application of GA depends on the population size or the diversity of individual solutions in the search space. If GA cannot hold its diversity well before the global optimum is reached, it may prematurely converge to a local optimum. Although maintaining diversity is the predominant concern of GA, it also reduces the performance of GA [3,14].

Lui and Wang [15] modified the genetic algorithm by defining and changing the operating domain and used for optimization of milling parameters. The results and the convergence speed of their approach were better than that of genetic algorithm. Wang et al. [16] presented a new hybrid approach, named genetic simulated annealing (GSA) and parallel genetic simulated annealing (PGSA), based on genetic algorithm and simulated annealing to find optimal machining parameters in milling operations. They pointed out that the results obtained were found to be better than those of genetic algorithm and geometric programming. Baskar et al. [17] considered a specific case in milling operation and solved the same by using three different non-traditional optimization techniques comprising a genetic algorithm, local hill climbing and memetic algorithm.

Onwubolu [18] presented a new optimization technique based on tribes for determination of the cutting parameters in multi-pass milling operations. Although the results obtained in his work using tribes showed significant improvement over other traditional and non-traditional algorithms, but the results are not valid as some of the constraints in the solution obtained are violated. This is explained in the present work in Section 5. Yildiz [19] developed a new hybrid optimization approach by hybridizing the immune algorithm with hill climbing local search algorithm to maximize the total profit rate in milling operations. Zarei et al. [20] presented a harmony search (HS) algorithm to determine the optimum cutting parameters for multi-pass face-milling.

The present study is mainly focused on optimization of process parameters of multi-pass milling operations considering minimization of total production time as the objective function (i.e. maximization of production rate) with constraints of arbor strength, arbor deflection and cutting power. Feed per tooth, speed and depth of cut are considered as process parameters. The upper and lower bounds of the process parameters are also included in the study.

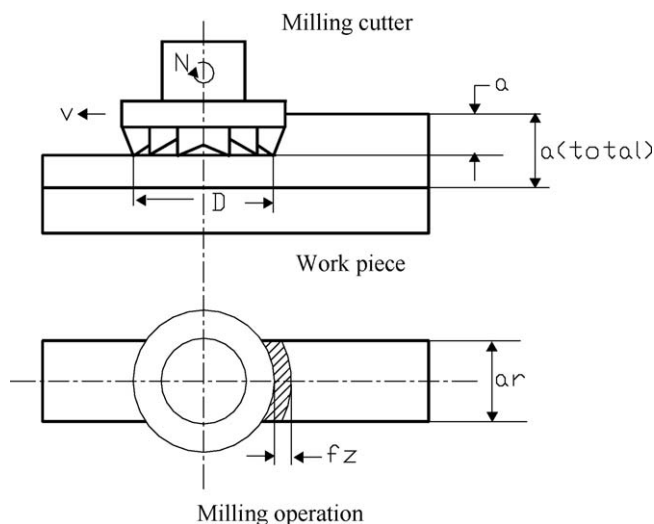


Fig. 1. Milling operation.

The next section presents an optimization model of multi-pass milling process.

3. Optimization model of multi-pass milling process

The optimization model of milling process formulated in the present work is based on the analysis given by Sonmez et al. [11]. The decision variables (i.e. process parameters) considered for this model are feed per tooth (f_z), cutting speed (V) and depth of cut (a). The objective function and the constraints are formulated as discussed in the following sections.

3.1. Objective function

For a milling operation the total production time (T_{pr}) is composed of the following items:

- (a) Machine preparation time (T_p) as given by Eq. (1)

$$T_p = \frac{T_s}{N_b} \quad (1)$$

where T_s = set-up time and N_b = total number of components in a produced batch.

- (b) Loading–unloading time (T_L).

- (c) Process adjusting and quick return time (T_a).

- (d) Machining time (T_m).

- (e) Tool changing time per component (T_c), which is given by Eq. (2).

$$T_c = \frac{T_d T_m}{T} \quad (2)$$

where T_d = time for changing a dull cutting edge or tool and T = tool life.

For a single pass milling operation the total production time (T_{pr}) is the sum of the above time elements and can be written as

$$T_{pr} = T_p + T_L + T_a + T_m + T_c \quad (3)$$

or

$$T_{pr} = \frac{T_s}{N_b} + T_L + T_a + T_m + T_d \left(\frac{T_m}{T} \right) \quad (4)$$

For a multi-pass operation, Eq. (4) becomes

$$T_{pr} = \frac{T_s}{N_b} + T_L + \sum_{i=1}^{N_p} \left(T_{ai} + T_{mi} + T_d \frac{T_{mi}}{T} \right) \quad (5)$$

where N_p = total number of passes and subscript i denotes i th pass.

For a particular milling operation, the machining time is given as

$$T_m = \frac{L}{f} \quad (6)$$

where L = length of cut, f = feed rate = $f_z Z N$, f_z = feed per tooth, Z = number of teeth on milling cutter, and N = spindle speed in rpm.

N is given by equation:

$$N = \frac{1000 \times V}{\pi D} \quad (7)$$

where D = cutter diameter, V = cutting speed.

Tool life can be determined by using the formula given by Eq. (8).

$$T = \frac{C_v^{1/m} D^{b_v/m} \times (B_m B_h B_p B_t)^{1/m}}{V^{1/m} a^{a_v/m} f_z^{u_v/m} a_r^{r_v/m} Z^{n_v/m} \lambda_s^{q_v/m}} \quad (8)$$

where a = depth of cut, a_r = width of the cut, B_m , B_h , B_p , B_t = correction coefficients, m , e_v , u_v , r_v , n_v , q_v , b_v = exponents, C_v = process constant, λ_s = cutting inclination angle.

On substituting Eqs. (6)–(8) in Eq. (5), the objective function for multi-pass milling operation is expressed as given by Eq. (9).

$$T_{pr} = \frac{T_s}{N_b} + T_L + N_p T_a + \sum_{i=1}^{N_p} \frac{\pi D L}{f_{zi} Z \times 1000 \times V_i} + \frac{T_d \pi L V_i^{(1/m-1)} a_i^{e_v/m} f_{zi}^{u_v/m-1} a_{ri}^{r_v/m} Z^{(n_v/m-1)} \lambda_s^{q_v/m}}{1000 \times C_v^{1/m} D^{(b_v/m-1)} \times (B_m B_h B_p B_t)^{1/m}} \quad (9)$$

3.2. Constraints

Following three constraints are considered in this optimization model.

3.2.1. Arbor strength

The arbor is subjected to torsion from the action of resistance to cutting. Therefore, the selected values of process parameters should ensure that the arbor is safe from strength point of view.

$$F_s - F_c \geq 0 \quad (10)$$

where

$$\text{Mean peripheral cutting force} = F_c = C_{zp} a_r Z D^{b_z} a^{e_z} f_z^{u_z} \quad (11)$$

where C_{zp} = process constants, and b_z , e_z , and u_z are exponents.

Permissible force for arbor strength (kg) = F_s

$$= \frac{0.1 k_b d_a^3}{0.08 L_a + 0.65 \sqrt{(0.25 L_a)^2 + (0.5 \alpha D)^2}} \quad (12)$$

where k_b = permissible bending strength of arbor, d_a = arbor diameter, L_a = arbor length between supports, $\alpha = k_b / (1.3 k_t)$, and k_t = permissible torsional strength of arbor.

3.2.2. Arbor deflection

The selected values of process parameters should be checked for arbor deflection as given by Eq. (13).

$$F_d - F_c \geq 0 \quad (13)$$

where

$$\text{Permissible force for arbor deflection (kg)} = F_d = \frac{4 E e d_a^4}{L_a^3} \quad (14)$$

where E = modulus of elasticity of arbor material and e = permissible value of arbor deflection.

For roughing operation $e = 0.2$ mm and for finishing operation $e = 0.05$ mm.

3.2.3. Power

Power required for the cutting operation should not exceed the effective power transmitted to cutting point by the machine tool. This is ensured by Eq. (15).

$$P_c - \frac{F_c V}{6120} \geq 0 \quad (15)$$

where P_c = cutting power (kW) = $P_m \eta$, P_m = nominal motor power, and η = overall efficiency.

3.3. Process parameters

The three process parameters and their bounds considered in this work are given in the following sections.

3.3.1. Feed per tooth

The optimum feed must be in the range determined by maximum and minimum values of the feed rates of the machine.

$$f_{z_{min}} \leq f_z \leq f_{z_{max}} \quad (16)$$

where

$$f_{z_{min}} = \frac{f_{min}}{zN_{max}} \quad (17)$$

$$f_{z_{max}} = \frac{f_{max}}{zN_{min}} \quad (18)$$

where f_{max} = maximum spindle feed rate (mm/min), f_{min} = minimum spindle feed rate (mm/min), N_{max} = maximum spindle speed, and N_{min} = minimum spindle speed.

3.3.2. Cutting speed

The optimum cutting speed must be in the range determined by maximum spindle speed (N_{max}) and minimum spindle speed (N_{min}) of the machine.

$$V_{min} \leq V \leq V_{max} \quad (19)$$

where

$$V_{max} = \frac{\pi DN_{max}}{1000} \quad (20)$$

$$V_{min} = \frac{\pi DN_{min}}{1000} \quad (21)$$

3.3.3. Depth of cut

For a milling operation the upper and lower bounds for depth of cut are as specified by Eq. (22).

$$a_{min} \leq a \leq a_{max} \text{ (mm)} \quad (22)$$

where a_{min} is the minimum depth of cut and a_{max} is the maximum depth of cut.

The above optimization model with the given process parameters, objective function and the constraints is considered in the present work for multi-pass milling process using non-traditional optimization algorithms such as ABC, PSO and SA. These algorithms are explained briefly in the following sections.

4. Presented non-traditional optimization algorithms

Three non-traditional optimization algorithms are considered in the present work for multi-pass milling process parameter optimization and are described in the following sections.

4.1. Artificial bee colony algorithm

A branch of nature inspired algorithms, called swarm intelligence, is focused on insect behavior in order to develop some meta-heuristics which can mimic insect's problem solution abilities. Interaction between insects contributes to the collective intelligence of the social insect colonies. These communication systems between insects have been adapted to scientific problems for optimization. The foraging behavior, learning, memorizing and information sharing characteristics of honeybees have recently been one of the most interesting research areas in swarm intelligence. Artificial bee colony (ABC) algorithm is developed to model the intelligent behaviors of honeybee swarms [1–3]. The honeybee swarms consists of two essential components (i.e. food sources and foragers) and defines two leading modes of the behavior (i.e. recruitment to a nectar source and abandonment of a source).

4.1.1. Food sources

The value of a food source depends on different parameters such as its proximity to the nest, richness of energy and ease of

extracting this energy. For the simplicity, the “profitability” of a food source can be represented with a single quantity.

4.1.2. Foragers

Foragers can be unemployed, employed or experienced.

4.1.2.1. Unemployed foragers. If it is assumed that a bee has no knowledge about the food sources in the search field, bee initializes its search as an unemployed forager. There are two possibilities for an unemployed forager:

- **Scout bee:** If the bee starts searching spontaneously without any knowledge, it will be a scout bee. The percentage of scout bees varies from 5% to 30% according to the information into the nest. The mean number of scouts averaged over conditions is about 10%.
- **Recruit:** If the unemployed forager attends to a waggle dance done by some other bee, the bee will start searching by using the knowledge from waggle dance.

4.1.2.2. Employed foragers. When the recruit bee finds and exploits the food source, it will raise to be an employed forager who memorizes the location of the food source. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive. There are three possible options related to residual amount of nectar for the foraging bee. If the nectar amount decreased to a low level or exhausted, foraging bee abandons the food source and become an unemployed bee. If there are still sufficient amount of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates or it can go to the dance area to perform waggle dance for informing the nest mates about the same food source. The probability values for these options highly related to the quality of the food source.

4.1.2.3. Experienced foragers. These types of foragers use their historical memories for the location and quality of food sources. This type of forages can be an inspector, which controls the recent status of food source already discovered. It can also be a reactivated forager by using the information from waggle dance. It tries to explore the same food source discovered by self if there are some other bees confirm the quality of same food source. It can also be scout bee to search new patches if the whole food source is exhausted. It can also be a recruit bee, which is searching a new food source declared in dancing area by another employed bee.

Communication among bees related to the quality of food sources occurs in the dancing area. The related dance is called waggle dance. Since information about all the current rich sources is available to an onlooker on the dance floor, she probably could watch numerous dances and choose to employ herself at the most profitable source. There is a greater probability of onlookers choosing more profitable sources since more information is circulating about the more profitable sources. Employed foragers share their information with a probability, which is proportional to the profitability of the food source, and the sharing of this information through waggle dancing is longer in duration. Hence, the recruitment is proportional to profitability of a food source. The flowchart of ABC algorithm is shown in Fig. 2 [21].

The steps of ABC algorithm are explained in Section 5.

4.2. Particle swarm optimization algorithm

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart [22]. It exhibits common evolutionary computation attributes including initialization with a population of random solutions and searching for optima by updating generations. Potential solutions, called

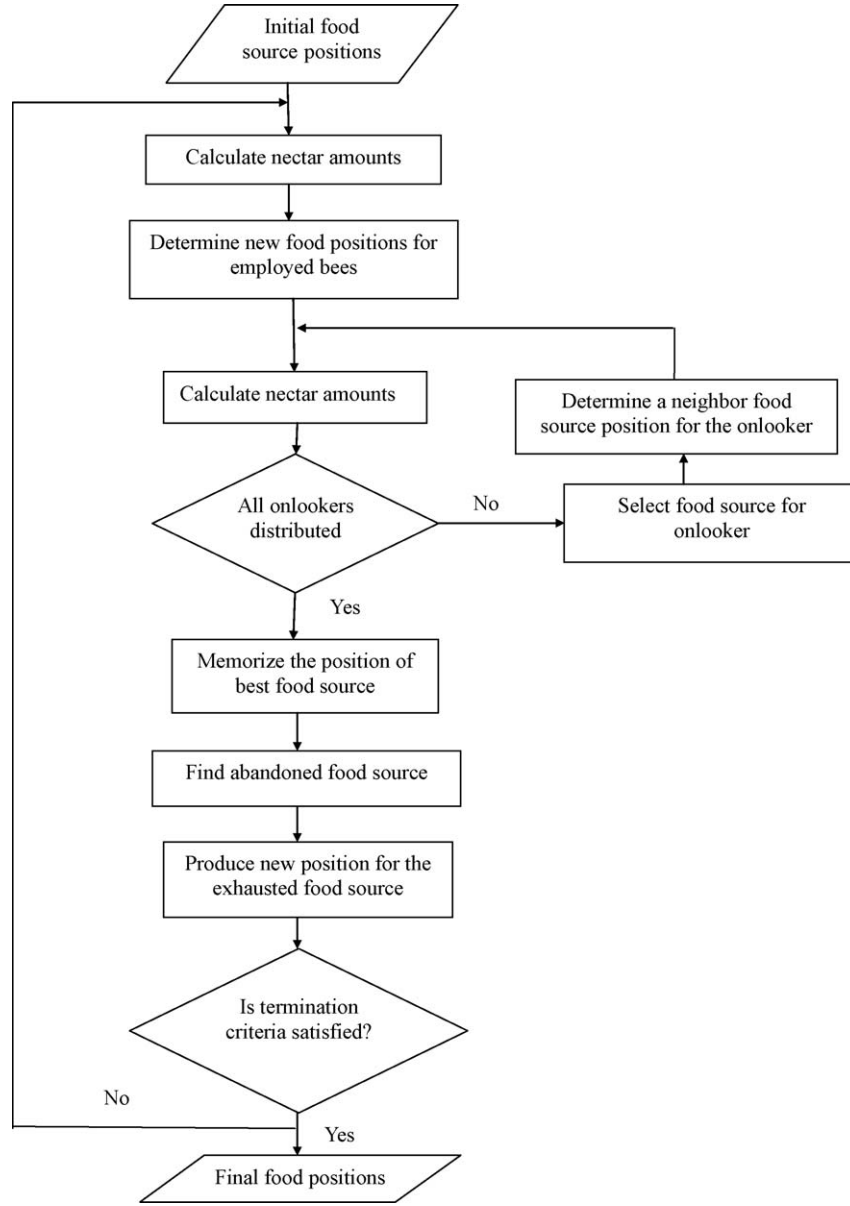


Fig. 2. Flowchart of the ABC algorithm [20].

particles, are then “flown” through the problem space by following the current optimum particles. The updates of the particles are accomplished as per the following equations:

$$V_{i+1} = wV_i + c_1r_1(pBest_i - X_i) + c_2r_2(gBest_i - X_i) \quad (23)$$

$$X_{i+1} = X_i + V_{i+1} \quad (24)$$

Eq. (23) calculates a new velocity (V_{i+1}) for each particle (potential solution) based on its previous velocity, the best location it has achieved ($pBest$) so far, and the global best location ($gBest$), the population has achieved. Eq. (24) updates individual particle's position (X_i) in solution hyperspace. The two random numbers r_1 and r_2 in Eq. (23) are independently generated in the range [0,1]. The acceleration constants c_1 and c_2 in Eq. (23) represent the weighting of the stochastic acceleration terms that pull each particle towards $pBest$ and $gBest$ positions. c_1 represents the confidence the particle has in itself (cognitive parameter) and c_2 represents the confidence the particle has in swarm (social parameter). Thus, adjustment of these constants changes the amount of tension in the system. The inertia weight w plays an

important role in the PSO convergence behavior since it is employed to control the exploration abilities of the swarm. To achieve the dimensional consistency of Eqs. (23) and (24), the dimension of the term cr in Eq. (23) could be taken as $(\text{time})^{-2}$. This way, the second and the third terms in Eq. (23) assume the dimension of acceleration. To get the correct dimension of velocity, as required by the left hand side, one needs to multiply them by Δt , the time step, which becomes unity in the present case, denoting changes from iteration i to $i+1$. Similarly, the second term in Eq. (24) assumes the correct dimension when taken as $V_{i+1} \Delta t$. However, the present form results through the implicit assumption that Δt equals 1 [23,24].

4.3. Simulated annealing algorithm

Simulated annealing is a probabilistic hill climbing soft computing algorithm. The methodology of simulated annealing algorithm is described below.

If i is the current configuration with cost $C(i)$ then using the Metropolis algorithm [25], we can say that the probability of

accepting j as next configuration depends on the difference in the function value at these two points or on $\Delta C = C(j) - C(i)$ and is calculated using the Boltzman probability distribution:

$$Pr\{new = j | current = i\} = \begin{cases} 1 & \text{if } \Delta C \leq 0 \\ e^{-\Delta C/T} & \text{otherwise} \end{cases} \quad (25)$$

The nest section provides an application example to demonstrate and validate the application of the presented ABC, PSO and SA algorithms.

5. Application example

Now an application example is considered to demonstrate and validate the presented ABC, PSO and SA algorithms for the optimization of process parameters of the multi-pass milling operation. The example is based on the model developed by Sonmez et al. [11]. Specifications of the required parameters and values of the constants considered by Sonmez et al. [11] and used in the present work are as follows:

Type of machining: plain milling.

Motor power (P_m) = 5.5 kW, efficiency, $\eta = 0.7$.

Arbor diameter, $d_a = 27$ mm, arbor length between supports, $L_a = 210$ mm.

Permissible bending stress of arbor, k_b : 140 MPa.

Permissible torsional stress of arbor, k_t : 120 MPa.

Modulus of elasticity of arbor material, $E = 200$ GPa.

Spindle speed range: (31.5–2000) rpm, feed rate range: (14–900) mm/min.

Tool material: HSS, tool diameter, $D = 63$ mm, number of teeth, $z = 8$.

Material: structural carbon steel (C # 0.6%).

Tensile strength: 750 MPa, Brinell hardness number = 150.

Length of cut, $L_a = 160$ mm, width of cut, $a_r = 50$ mm, depth of cut, $a = 5$ mm.

Loading and unloading time of one work piece, $T_L = 1.5$ min.

Set-up time of fixtures and machine tool, $T_s = 10$ min.

Tool change time, $T_c = 5$ min.

Process adjusting and quick return time, $T_a = 0.1$ (min/part).

Lot size (number of parts in the batch), $N_b = 100$.

Cutting inclination = 30° .

Constants: $B_m = 1$, $B_k = 1$, $B_p = 0.8$, $B_t = 0.8$, $m = 0.33$, $e_v = 0.3$, $u_v = 0.4$, $r_v = 0.1$, $n_v = 0.1$, $q_v = 0$, $C_v = 35.4$, $b_v = 0.45$, $C_{zp} = 68.2$, $b_z = -0.86$, $e_z = 0.86$, and $u_z = 0.72$.

$$f_{z_{min}} = \frac{f_{min}}{zN_{max}} = \frac{14}{8 \times 2000} = 0.000875 \text{ (mm/tooth)}$$

$$f_{z_{max}} = \frac{f_{max}}{zN_{min}} = \frac{900}{8 \times 31.5} = 3.571 \text{ (mm/tooth)}$$

Thus,

$$0.000875 \leq f_z \leq 3.571 \quad (26)$$

$$V_{max} = \frac{\pi D N_{max}}{1000} = \frac{\pi \times 63 \times 2000}{1000} = 395.84 \text{ (m/min)}$$

$$V_{min} = \frac{\pi D N_{min}}{1000} = \frac{\pi \times 63 \times 31.5}{1000} = 6.234 \text{ (m/min)}$$

Thus,

$$6.234 \leq V \leq 395.84 \text{ (m/min)} \quad (27)$$

For a milling operation the upper and lower bounds for depth of cut are as specified by Eq. (28).

$$0.5 \leq a \leq 4 \text{ (mm)} \quad (28)$$

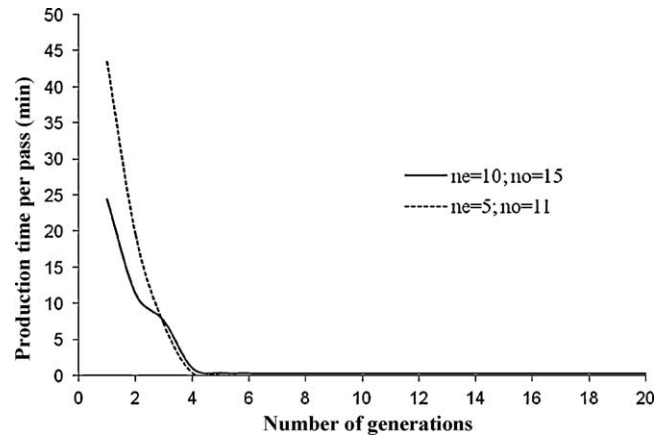


Fig. 3. Effect of number of employed bees (n_e) on convergence rate for rough milling.

5.1. Optimization using artificial bee colony algorithm

Following steps are used for optimization of multi-pass milling operation using presented artificial bee colony algorithm.

5.1.1. Step 1: parameter selection

As discussed in the description of ABC algorithm, food source represents a possible solution to the problem of minimization of production time in the present work. The number of initial solutions (i.e. the number of food sources) considered in this work is five. The value of each food source depends on the fitness value of the objective function given by Eq. (9).

For every food source there is only one employed bee (employed forager). In other words, the number of employed bees is equal to number of food sources. Hence, in the present work number of employed bees is considered to be five. It can be observed from Fig. 3 that even though the number of employed bees increases from 5 to 10, the convergence rate of the algorithm slightly increases, however it does not affect the final solution. Moreover, computational efforts increase as number of employed bees increases. From this point of view the selection of the number of employed bee (equal to five) is appropriate. The unemployed forager can be scout or an onlooker bee. The number of onlooker bees must be greater than the number of employed bees. The effect of number of onlooker bees on the convergence of solution is shown in Fig. 4. It is observed that as the number of onlooker bees and hence the population size increases, the algorithm performs better in terms of convergence rate. However, after a sufficient number of onlooker bees, any increment in the value does not

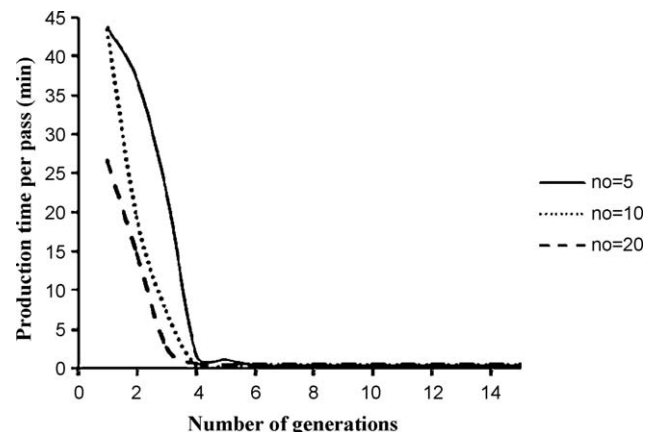


Fig. 4. Effect of number of onlooker bees (n_o) on convergence rate for rough milling.

improve the performance of the algorithm. For the problem considered in this work, number of onlooker bees is considered to be 11, which can provide an acceptable convergence speed for search. The colony size is the sum of number of employed bees and number of onlooker bees. Hence the colony size is 16. Number of scout bees is usually 5–30% of the colony size. In the present work, the number of scout bee is taken as 5% of the colony size i.e. one. The parameters of optimization thus selected in this work are summarized as

- Number of employed bees = 5.
- Number of onlookers bees = 11.
- Number of scout bees = 1.
- Maximum number of iterations = 150.

It may be mentioned here that the algorithm is tried for parameters of optimization values of number of employed bees = 10 and colony size = 25. However, the results are not better than the results obtained using number of employed bees = 5 and colony size = 16.

5.1.2. Step 2: calculate the nectar amount of each food source

The employed bees are moved to the food sources and the nectar amount of these food sources is evaluated based on their fitness value as defined by the objective function given by Eq. (9) subjected to constraints given by Eqs. (10), (13) and (15).

5.1.3. Step 3: determine the probabilities by using the nectar amount

If the nectar amount of a food source θ_i is f_i , then the probability (P_i) of choosing this food source by an onlooker bee is expressed as

$$P_i = \frac{\left[\sum_{i=1}^S (1/f_i) \right]^{-1}}{f_i} \quad (29)$$

where S is the number of food sources.

5.1.4. Step 4: calculate the number of onlookers bees, which will be sent to food sources

Based on the probabilities calculated in step 3, the number (N) of onlookers bees sent to food source θ_i is calculated as

$$N = P_i m \quad (30)$$

where m is the total number of onlooker bees.

5.1.5. Step 5: calculate the fitness value of each onlooker bee

After watching the dances of employed bees, an onlooker bee goes to the region of food source θ_i by the probability given by Eq. (29). The position of the selected neighbor food source is calculated as the shown in Eq. (31).

$$\theta_i(c+1) = \theta_i(c) \pm \phi_i(c) \quad (31)$$

where c is number of generation. $\phi_i(c)$ is a randomly produced step to find a food source with a more nectar around θ_i . $\phi_i(c)$ is calculated by taking the difference of the same parts of $\theta_i(c)$ and $\theta_k(c)$ (k is a randomly produced index) food positions. If the nectar amount $F_i(c+1)$ at $\theta_i(c+1)$ is higher than at $\theta_i(c)$, then the bees go to the hive and share information with others and the position $\theta_i(c)$ of the food source is changed to $\theta_i(c+1)$ otherwise $\theta_i(c)$ is kept as it is. If the position θ_i of the food source i cannot be improved through the predetermined number of trials, then that food source θ_i is abandoned by its employed bee and then the bee becomes a scout. The scout starts searching new food source, and after finding the new source, the new position is accepted as θ_i .

5.1.6. Step 6: evaluate the best solution

Position of the best onlooker bee is identified for each food source. The global best of the honeybee swarm in each generation is obtained and it may replace the global best at previous generation if it has better fitness value.

5.1.7. Step 7: update the scout bee

The worst employed bees, as many as the number of scout bees in the population, are respectively compared with the scout solutions. If the scout solution is better than employed solution, employed solution is replaced with scout solution. Else employed solution is transferred to the next generation without any change.

In the present work, various feasible cutting strategies are adopted to determine the optimum number of passes required and depth of cut for each pass. The results of optimization for these strategies using ABC algorithm are shown in Table 1. It is observed from Table 1 that among various cutting strategies for multi-pass milling with total depth of cut of 5 mm, the strategy 2 with three rough cuts each of 1.5 mm and a finish cut of 0.5 mm is optimum as indicated by minimum production time of 3.24 min. The value of feed per tooth for finishing cut provided by strategy 2 is also much less than that provided by other strategies. In milling operation as the feed per tooth decreases, surface finish increases. Hence

Table 1

Results of optimization using ABC for various cutting strategies.

S. no.	Cutting strategy	f_z (mm/tooth)	V (m/min)	T_2 (per pass; min)	T_2 (min)	T_1 (min)	T_{pr} ($T_1 + T_2$; min)
1	$a_{rough} = 2$	0.231	48.117	0.475	1.378	1.9	3.278
	$a_{rough} = 2$	0.231	48.117	0.475			
	$a_{finish} = 1$	0.189	74.090	0.428			
2	$a_{rough} = 1.5$	0.337	46.982	0.343	1.240	2.0	3.240
	$a_{rough} = 1.5$	0.337	46.982	0.343			
	$a_{rough} = 1.5$	0.337	46.982	0.343			
	$a_{finish} = 0.5$	0.432	64.41	0.211			
3	$a_{rough} = 2$	0.231	48.117	0.475	1.355	2.0	3.355
	$a_{rough} = 1$	0.552	47.519	0.226			
	$a_{rough} = 1$	0.552	47.519	0.226			
	$a_{finish} = 1$	0.189	74.090	0.428			
4	$a_{rough} = 1$	0.552	47.519	0.226	1.332	2.1	3.432
	$a_{rough} = 1$	0.552	47.519	0.226			
	$a_{rough} = 1$	0.552	47.519	0.226			
	$a_{rough} = 1$	0.552	47.519	0.226			
	$a_{finish} = 1$	0.189	74.090	0.428			

a_{rough} = depth of cut for rough pass (mm); a_{finish} = depth of cut for finish pass (mm); $T_1 = T_s/N_b + T_L + N_p T_a$; $T_2 = \sum_{i=1}^{N_p} \frac{\pi D L}{T_{az} \times 1000 \times V_i} + \frac{T_d \pi L V_i^{1/m-1} a_{r_i}^{6/m} f_z^{(u_r/m-1)} a_{f_i}^{(u_f/m-1)} z^{(n_r/m-1)} \lambda_s^{q_r/m}}{1000 C_v^{1/m} D^{(b_r/m-1)} (B_m B_h B_p B_r)^{1/m}}$.

strategy 2 also ensures better surface finish as compare to other strategies. The optimum process parameter values obtained by ABC algorithm for a given multi-pass milling operation are as follows:

- Number of passes required = 4.
- Number of rough cutting passes = 3 with depth of cut in each roughing pass = 1.5 mm.
- Number of finish pass = 1 with depth of cut in finishing pass = 0.5 mm.
- Feed per tooth for roughing pass = 0.337 mm/tooth.
- Cutting speed for roughing pass = 46.982 m/min.
- Feed per tooth for finishing pass = 0.432 mm/tooth.
- Cutting speed for finishing pass = 64.41 m/min.
- Total production time (T_{pr}) = 3.240 min.

Optimality of the above mentioned solution could be confirmed from Figs. 5 to 10. Fig. 5 shows variation of production time and various constraints with feed per tooth for rough milling operation. As feed per tooth increases, production time reduces; hence higher value of feed rate is desired. However, as shown in Fig. 5, the strength constraint is violated after feed per tooth attains a value of 0.337 mm. This confirms the optimum value of feed per tooth selected using ABC algorithm for rough milling operation. Fig. 6 shows variation of production time and constraints with cutting speed for rough milling operation. Since the deflection constraint is having a constant positive value in this figure, Fig. 7 is plotted neglecting deflection constraint to indicate more clearly the variation of production time and other two constraints with cutting speed. As shown in Fig. 7, production time initially decreases with increase in cutting speed, till cutting speed attains a

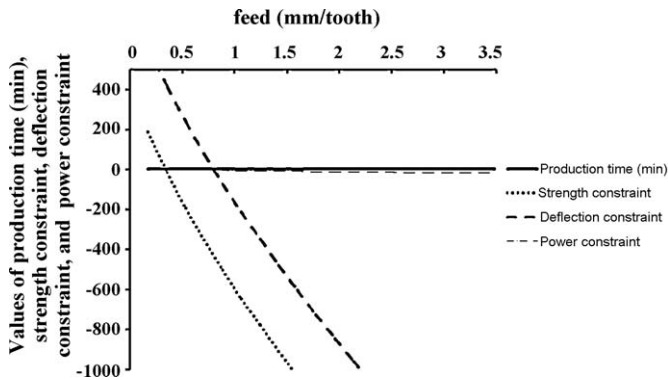


Fig. 5. Variation of production time and constraints with feed per tooth for rough milling.

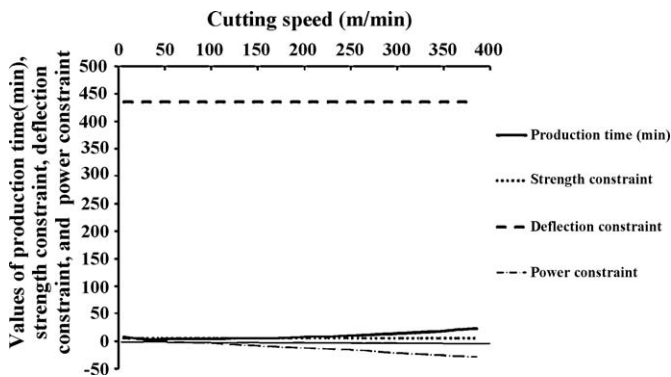


Fig. 6. Variation of production time and constraints with cutting speed for rough milling.

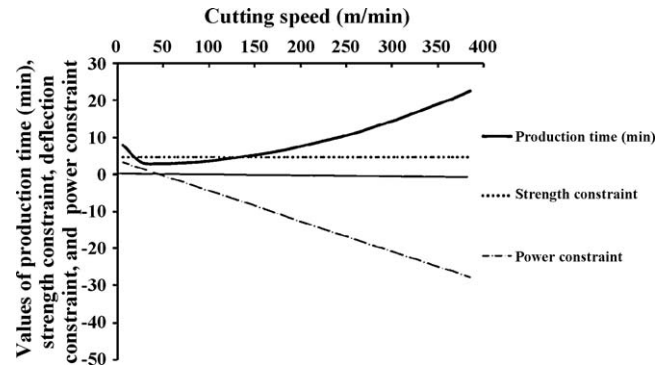


Fig. 7. Variation of production time and constraints with cutting speed for rough milling operation neglecting deflection constraint.

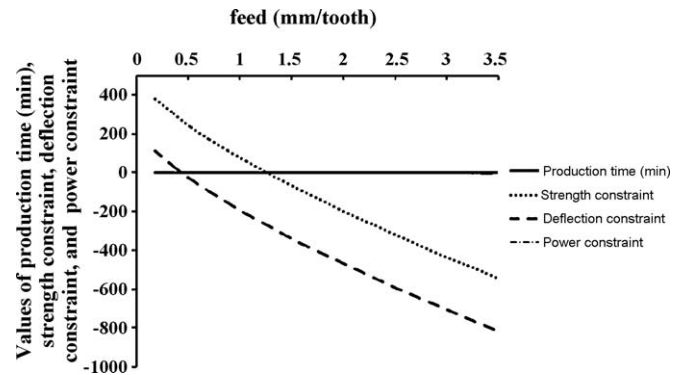


Fig. 8. Variation of production time and constraints with feed per tooth for finish milling.

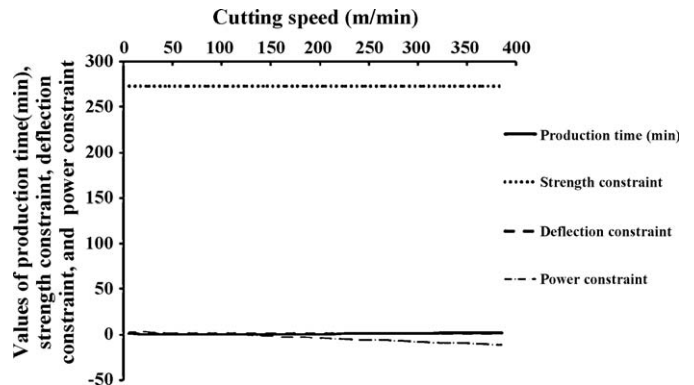


Fig. 9. Variation of production time and constraints with cutting speed for finish milling.

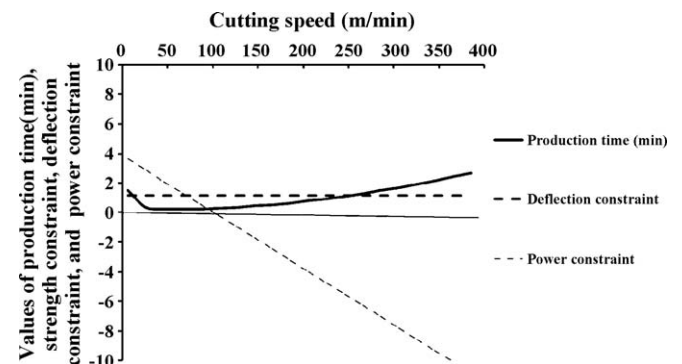


Fig. 10. Variation of production time and constraints with cutting speed for finish milling operation neglecting arbor strength constraint.

value of 47 m/min, after which the production time increases with increase in cutting speed. Thus, a cutting speed of 47 m/min is considered as optimum. Production time is minimum at this cutting speed value without violating any constraint.

Fig. 8 shows variation of production time and constraints with feed per tooth for finish milling operation. Although higher value of feed per tooth is desired to achieve minimum value of production time, the deflection constraint is violated for any value of feed per tooth higher than 0.432 mm. This confirms the optimum value of feed per tooth selected using ABC algorithm for finish milling operation. Fig. 9 shows variation of production time and constraints with cutting speed for finish milling operation. Since the strength constraint is having a constant positive value in this figure, Fig. 10 is plotted neglecting the strength constraint to indicate more clearly the variation of production time and other two constraints with cutting speed. As shown in Fig. 10, production time initially decreases with increase in cutting speed, till cutting speed attains a value of 64.41 m/min, after which the production time increases with increase in cutting speed. Thus for finish milling operation, a cutting speed of 64.41 m/min is considered as optimum and the production time is minimum at this cutting speed value without violating any constraint.

5.2. Optimization using particle swarm optimization algorithm

The optimum selection of operating parameters of the algorithm like acceleration constants c_1 and c_2 as well as inertia coefficient w is very essential for convergence of the algorithm. To ensure the convergence of PSO algorithm, the condition specified by Eq. (32) must be satisfied [26].

$$\max(|\lambda_1|, |\lambda_2|) < 1 \quad (32)$$

where λ_1 and λ_2 are the eigen values given by Eqs. (33) and (34).

$$\lambda_1 = \frac{(1 + w - \phi_1 - \phi_2 + \gamma)}{2} \quad (33)$$

$$\lambda_2 = \frac{(1 + w - \phi_1 - \phi_2 - \gamma)}{2} \quad (34)$$

and

$$\gamma = [(1 + w - \phi_1 - \phi_2)^2 - 4w]^{1/2} \quad \phi_1 = r_1 c_1 \text{ and } \phi_2 = r_2 c_2 \quad (35)$$

Considering the feasible range for the value of $\phi_1 + \phi_2$ as 0–4 and that for w as 0–1, it can be observed that for convergent

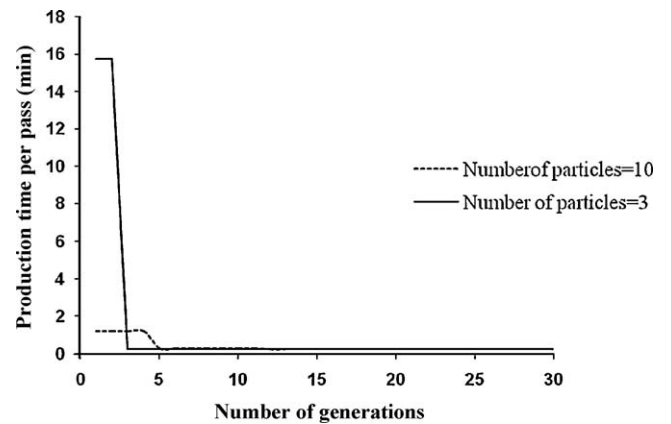


Fig. 11. Effect of number of particles in swarm on convergence rate and accuracy of solution for finish milling operation.

trajectories the relation given by Eq. (36) must be satisfied.

$$w > 0.5(\phi_1 + \phi_2) - 1 \quad (36)$$

In the present study the values of w , c_1 and c_2 are 0.65, 1.65, and 1.75 respectively. Considering the extreme possibility of random number as $r_1 = 0.95$ and $r_2 = 0.95$, the right hand term in Eq. (36) is $0.5 \times (0.95 \times 1.65 + 0.95 \times 1.75) - 1 = 0.615$, which is less than 0.65 thus satisfies Eq. (36). Hence, the values of w , c_1 and c_2 selected in the present work are appropriate for convergence of the algorithm.

Now, the presented PSO algorithm is applied to solve the above optimization problem. The following parameters of optimization algorithm are selected after various trials:

- Maximum number of iterations = 40.
- Inertia weight factor (w) = 0.65.
- Acceleration coefficients: $c_1 = 1.65$ and $c_2 = 1.75$.
- Number of particles in swarm = 3.

It can be observed from Fig. 11, the convergence rate of algorithm slightly increases when the number of particles in swarm increased from 3 to 10. However with this higher population size the accuracy of the solution is unaffected and also more computational efforts are required. Hence the selection of number of particles in swarm for the present example is appropriate. The results of optimization using PSO algorithm for various strategies, indicated in Table 1, are shown in Table 2. The results observed from Table 2 also confirm the results obtained by

Table 2
Results of optimization using PSO algorithm for various cutting strategies.

S. no.	Cutting strategy	f_z (mm/tooth)	V (m/min)	T_2 (per pass; min)	T_2 (min)	T_1 (min)	T_{pr} ($T_1 + T_2$; min)
1	$a_{rough} = 2$	0.240	46.53	0.464	1.356	1.9	3.256
	$a_{rough} = 2$	0.240	46.53	0.464			
	$a_{finish} = 1$	0.19	70.84	0.428			
2	$a_{rough} = 1.5$	0.34	46.61	0.343	1.240	2.0	3.240
	$a_{rough} = 1.5$	0.34	46.61	0.343			
	$a_{rough} = 1.5$	0.34	46.61	0.343			
	$a_{finish} = 0.5$	0.434	63.58	0.211			
3	$a_{rough} = 2$	0.240	46.53	0.464	1.342	2.0	3.342
	$a_{rough} = 1$	0.553	46.42	0.225			
	$a_{rough} = 1$	0.553	46.42	0.225			
	$a_{finish} = 1$	0.19	70.84	0.428			
4	$a_{rough} = 1$	0.553	46.42	0.225	1.328	2.1	3.428
	$a_{rough} = 1$	0.553	46.42	0.225			
	$a_{rough} = 1$	0.553	46.42	0.225			
	$a_{rough} = 1$	0.553	46.42	0.225			
	$a_{finish} = 1$	0.19	70.84	0.428			

using ABC algorithm that the optimum cutting strategy is strategy 2. The optimum process parameter values obtained by PSO algorithm for the given multi-pass milling operation are as follows:

- Number of passes required = 4.
- Number of rough cutting passes = 3 with depth of cut in each roughing pass = 1.5 mm.
- Number of finish pass = 1 with depth of cut in finishing pass = 0.5 mm.
- Feed per tooth for roughing pass = 0.34 mm/tooth.
- Cutting speed for roughing pass = 46.61 m/min.
- Feed per tooth for finishing pass = 0.434 mm/tooth.
- Cutting speed for finishing pass = 63.58 m/min.
- Total production time (T_{pr}) = 3.240 min.

5.3. Optimization using simulated annealing algorithm

Using the simulated annealing technique, the objective function to minimize the total production time is written as

$$\min Z = -Z_1 - (P_1 Z_2) - (P_2 Z_3) - (P_3 Z_4) \quad (37)$$

where Z = combined objective function, Z_1 = objective function given by Eq. (9), Z_2, Z_3, Z_4 are the constraints given by Eqs. (10), (13) and (15) respectively; P_1, P_2, P_3 are the penalties assigned for violation of constraints Z_2, Z_3, Z_4 respectively. In present case, $P_1 = 0.05, P_2 = 0.04$ and $P_3 = 2.5$ if a particular constraint is violated, else penalty = 0.

The initial temperature is obtained by calculating the average of the function values at a boundary points.

$$\text{Initial temperature}(T_0) = \frac{\sum Z_{Nb}}{n} \quad (38)$$

where Z_{Nb} = value of objective function at each boundary point and n = number of boundary points. The initial temperature is calculated as 200 and the decrement factor is considered as 0.1. At any current point $X(t)$, the new value of the parameters for the successive iterations is calculated using the formula:

$$X(t+1) = X(t) + \sigma \sum_{i=1}^N R_i - \frac{N}{2} \quad (39)$$

where $\sigma = (X_{max} - X_{min})/6$, R = random number, and N = number of random numbers used. In the present work, 6 random numbers are used. While starting the process, the initial values for the parameters are taken as the average of the respective parameter limits. The algorithm is terminated when a sufficiently small temperature is obtained or a small enough change in function value is found. Application of SA algorithm to solve the above optimization problem leads to the following optimum solution:

- Number of passes required = 4.
- Number of rough cutting passes = 3 with depth of cut in each roughing pass = 1.5 mm.
- Number of finish pass = 1 with depth of cut in finishing pass = 0.5 mm.
- Feed per tooth for roughing pass = 0.336 mm/tooth.
- Cutting speed for roughing pass = 44.633 m/min.
- Feed per tooth for finishing pass = 0.429 mm/tooth.
- Cutting speed for finishing pass = 57.23 m/min.
- Total production time (T_{pr}) = 3.263 min.

Table 3 shows the comparative performance of various presented algorithms i.e. ABC, PSO and SA along with those reported in literature by using various other optimization techniques. It is observed that for the optimum solution obtained by using optimization methods like GP, GA, PGSA, and Tribes violate most of the constraints (indicated by negative values in Table 3). This is due to the fact that previous researchers [11,16,18] considered the unit of cutting force specified by Eq. (11) in Newtons whereas it must be in kg. The unit of cutting force in kg as considered in the present work is supported by the following two facts:

- By considering the unit of cutting force in Newtons, the optimum solutions obtained by using GP, GA, PGSA and Tribes [11,16,18] are not at all influenced by the any of the constraints. This is indicated by a very large difference between permissible value of the constraining parameter and its corresponding value provided by the optimum solution. This clearly indicates that the consideration of unit of cutting force in Newtons by the previous researchers is wrong.

Table 3
Results of optimization by using various optimization algorithms.

Method	Cutting strategy	f_z (mm/tooth)	V (m/min)	SC	DC	PC	T_2 (min)	T_{pr} ($T_1 + T_2$; min)
GP [11]	$a_{rough} = 3$	0.338	26.40	-405	24.92	-0.08	0.813	2.614
	$a_{finish} = 2$	0.570	25.16	-430	-702	0		
GA [15]	$a_{rough} = 3$	0.366	24.69	-459	-28.81	-0.04	0.8102	2.61
	$a_{finish} = 2$	0.5667	25.16	-427	-698	0		
PGSA [15]	$a_{rough} = 3$	0.3693	24.25	-465	-35	0.2	0.8	2.60
	$a_{finish} = 2$	0.5886	24.58	-452	-74	0		
Tribes [17]	$a_{rough} = 3$	0.587	36.27	-8.50	-420	-4.18	0.512	2.212
	$a_{finish} = 2$	0.902	30.16	-797	-1069	-2.57		
ABC	$a_{rough} = 1.5$	0.337	46.982	4.708	435.02	0.0047	1.240	3.240
	$a_{rough} = 1.5$	0.337	46.982	4.708	435.02	0.0047		
	$a_{rough} = 1.5$	0.337	46.982	4.708	435.02	0.0047		
	$a_{finish} = 0.5$	0.432	64.41	271.97	1.131	1.400		
PSO	$a_{rough} = 1.5$	0.34	46.61	1.5	431.9	0.01	1.240	3.240
	$a_{rough} = 1.5$	0.34	46.61	1.5	431.9	0.01		
	$a_{rough} = 1.5$	0.34	46.61	1.5	431.9	0.01		
	$a_{finish} = 0.5$	0.434	63.58	271.9	0.35	1.422		
SA	$a_{rough} = 1.5$	0.336	44.633	5.779	436.09	0.204	1.263	3.263
	$a_{rough} = 1.5$	0.336	44.633	5.779	436.09	0.204		
	$a_{rough} = 1.5$	0.336	44.633	5.779	436.09	0.204		
	$a_{finish} = 0.5$	0.429	57.23	273.91	2.296	1.683		

SC: arbor strength constraint; DC: arbor deflection constraint; PC: power constraint.

- The values of feed per tooth obtained by using GP, GA, PGSA are much higher (>0.55 mm) than practically achievable value of feed per tooth (<0.45 mm) in case of plain milling operation.

Due to the above mentioned reasons, the optimum solutions obtained by previous researchers [11,16,18] are not valid. For the optimum solution obtained by using ABC, PSO and SA algorithms, the difference between permissible value of the constraining parameter and its corresponding value provided by optimum solution is very less, which proves the validity and accuracy of the solution. It also reveals the fact that the unit of cutting force considered (in kg) in this work is appropriate. Also, the value of feed per tooth provided by the optimum solution using ABC, PSO and SA is more appropriate from practical point of view. Thus, although the optimum solutions obtained by using methodologies like GP, GA, PGSA and Tribes seems to be better than that obtained by using, ABC, PSO and SA algorithms, the optimum solution obtained by using the presented algorithms in this work i.e. ABC, PSO and SA is only valid and appropriate.

The convergence rate of ABC and PSO algorithms is also very high and the algorithms require only 30–40 iterations for convergence to the optimal solution, whereas SA algorithm requires about 100 iterations for convergence. The convergence of ABC, PSO and SA algorithms for rough milling and finish milling is shown in Figs. 12 and 13. The results also show that the accuracy of solution obtained by using ABC and PSO algorithms is equally good, whereas the results obtained by SA are slightly inferior to the results obtained by using ABC and PSO algorithms. Artificial bee colony algorithm combines both, the stochastic selection scheme carried out by onlooker bees, and greedy selection scheme used by onlookers and employed bees to update the source position. Also

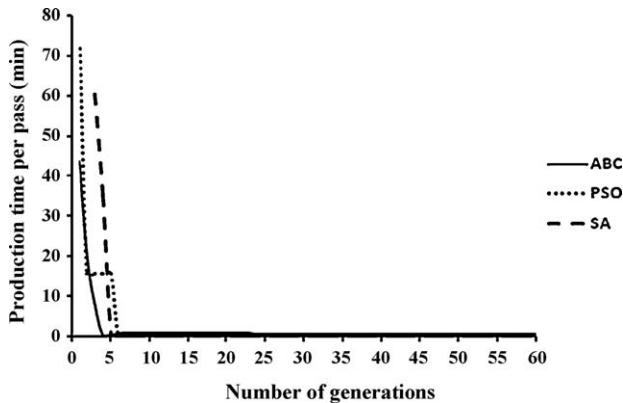


Fig. 12. Convergence of ABC, PSO and SA algorithms for rough milling operation.

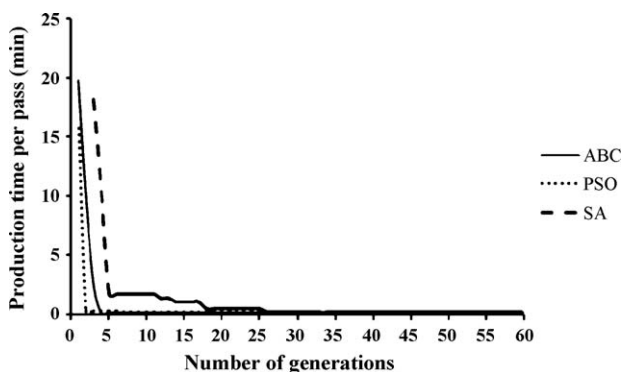


Fig. 13. Convergence of ABC, PSO, and SA algorithms for finish milling operation.

the neighbor source production mechanism in ABC algorithm is similar to the mutation process, which is self-adapting. It is also observed that few trials are required to predict the best and worst operating parameters for PSO algorithm.

6. Conclusions

Multi-pass operations are often preferred to single pass operations for economic reasons. Determination of the optimal process parameters such as the number of passes, depth of cut for each pass, speed, and feed is considered as a crucial stage of multi-pass machining as in the case of all chip removal processes and especially in process planning. The effective optimization of the process parameters affects dramatically the cost and production time of machined components as well as the quality of the final products. In the present work, optimization aspects of multi-pass milling operation are considered. The objective considered is minimization of production time (i.e. maximization of production rate) subjected to the constraints of arbor strength, arbor deflection and cutting power. Process parameters considered are feed per tooth, cutting speed, and depth of cut with their upper and lower bound values.

The performance of three non-traditional optimization algorithms such as artificial bee colony (ABC), particle swarm optimization (PSO) and simulated annealing (SA) is studied in terms of convergence rate and accuracy of the solution. These algorithms are applied to obtain the optimum process parameter values for various selected cutting strategies. The optimum strategy is selected based on the minimum production time. The convergence rate of ABC and PSO algorithms is very high and these algorithms require only little iteration for convergence to the optimal solution, whereas SA algorithm requires comparatively more iterations for convergence. The accuracy of solution obtained by using ABC and PSO algorithms is better as compared to the results obtained by using SA algorithm. This paper invalidates the attempts made by previous researchers to optimize the process parameters of milling operation for the same example considered in this work. The presented ABC, PSO, and SA algorithms can be easily modified to suit optimization of process parameters of other machining processes such as grinding, turning, drilling, etc.

References

- [1] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.
- [2] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* 39 (2003) 459–471.
- [3] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing* 8 (2008) 687–697.
- [4] J. Ciurana, G. Arias, T. Ozel, Neural network modeling and particle swarm optimization (PSO) of process parameters in pulsed laser micromachining of hardened AISI H13 Steel, *Materials and Manufacturing Processes* 24 (3) (2009) 358–368.
- [5] R.V. Rao, P.J. Pawar, R. Shankar, Multi-objective optimization of electro-chemical machining process parameters using a particle swarm optimization algorithm, *Journal of Engineering Manufacture* 122 (2008) 949–958.
- [6] P. Sathiyar, S. Aravindan, A. Narool Haq, K. Paneerselvam, Optimization of friction welding parameters using evolutionary computational techniques, *Journal of Material Processing Technology* 209 (5) (2009) 2576–2584.
- [7] J. Zhai, Z. Duan, Y. Li, J. Deng, D. Yu, PSO based neural network optimization and its utilization in a boring machine, *Journal of Material Processing Technology* 178 (1–3) (2006) 19–23.
- [8] Y.C. Shin, Y.S. Joo, Optimization of machining conditions with practical constraints, *International Journal of Production Research* 30 (1992) 2907–2919.
- [9] J. Wang, Multiple objective optimization of machining operations based on neural networks, *International Journal of Advanced Manufacturing Technology* 8 (1993) 235–243.
- [10] M. Tolouei-Rad, I.M. Bidhendi, On the optimization of machining parameters for milling operations, *International Journal of Machine Tools & Manufacture* 37 (1997) 1–16.

- [11] A.I. Sonmez, A. Baykasoglu, T. Dereli, I.H. Filiz, Dynamic optimization of multi-pass milling operations via geometric programming, *International Journal of Machine Tools & Manufacture* 39 (1999) 297–332.
- [12] J.S. Agapiou, The optimization of machining operations based on a combined criterion—Part 2. Multi-pass operations, *Journal of Engineering for Industry* 114 (1992) 508–513.
- [13] M.S. Shunmugam, S.V.B. Reddy, A.A. Narendran, Selection of optimal conditions in multi-pass face-milling using a genetic algorithm, *International Journal of Machine Tools & Manufacture* 40 (2000) 401–414.
- [14] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary based optimization algorithms, *Advanced Engineering Informatics* 19 (2005) 43–53.
- [15] Y.M. Lui, C.J. Wang, A modified genetic algorithm based optimization of milling parameters, *International Journal of Advanced Manufacturing Technology* 15 (1999) 796–809.
- [16] Z.G. Wang, M. Rahman, Y.S. Wong, J. Sun, Optimization of multi-pass milling using parallel genetic algorithm and parallel genetic simulated annealing, *International Journal of Machine Tools & Manufacture* 45 (2005) 1726–1734.
- [17] N. Baskar, P. Asokan, R. Saravanan, G. Prabhakaran, Selection of optimal machining parameters for multi-tool milling operations using a memetic algorithm, *Journal of Material Processing Technology* 174 (2006) 239–249.
- [18] G.C. Onwubolu, Performance-based optimization of multi-pass face milling operations using tribes, *International Journal of Machine Tools & Manufacture* 46 (2006) 717–727.
- [19] A.R. Yildiz, A novel hybrid immune algorithm for optimization of machining parameters in milling operations, *Robotics and Computer-Integrated Manufacturing* 25 (2) (2009) 261–270.
- [20] O. Zarei, M. Fesanghary, B. Farshi, R.J. Saffar, M.R. Razfar, Optimization of multi-pass face-milling via harmony search algorithm, *Journal of Materials Processing Technology* 209 (2009) 2386–2392.
- [21] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute* 346 (2009) 328–348.
- [22] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [23] N. Chakraborti, R. Jayakanth, S. Das, E.D. Calisir, S. Erkoc, Evolutionary and genetic algorithms applied to Li⁺–C system: calculations using differential evolution and particle swarm algorithm, *Journal of Phase Equilibria and Diffusion* 28 (2) (2007) 140–149.
- [24] N. Chakraborti, S. Das, R. Jayakanth, R. Pekoz, S. Erkoc, Genetic algorithms applied to Li⁺ ions contained in carbon nanotubes: an investigation using particle swarm optimization and differential evolution along with molecular dynamics, *Materials and Manufacturing Processes* 22 (5–6) (2007) 562–569.
- [25] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, *Journal of Chemical Physics* 21 (1953) 1087–1092.
- [26] F. Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, *Information Sciences* 176 (2006) 937–971.