Grinding process parameter optimization using non-traditional optimization algorithms

R V Rao* and **P J Pawar**
Department of Mechanical Engineering, SV National Institute of Technology, Gujarat, India

The manuscript was received on 16 August 2009 and was accepted after revision for publication on 20 October 2009.

DOI: 10.1243/09544054JEM1782

Abstract: Selection of machining parameters in any machining process significantly affects the production rate, quality, and cost of a component. This paper presents the multi-objective optimization of process parameters of a grinding process using various non-traditional optimization techniques such as artificial bee colony, harmony search, and simulated annealing algorithms. The objectives considered in the present work are production cost, production rate, and surface finish subjected to the constraints of thermal damage, wheel wear, and machine tool stiffness. The process variables considered for optimization are wheel speed, workpiece speed, depth of dressing, and lead of dressing. The results of the algorithms presented are compared with the previously published results obtained by using other optimization techniques.

Keywords: multi-objective optimization, artificial bee colony, harmony search algorithm, simulated annealing, grinding

1 INTRODUCTION

Grinding is a key process in automobile industries where a variety of components in large numbers are required to be ground to very close tolerance with a high surface finish. The success of the grinding process in terms of cost and quality depends on proper selection of various operating conditions in the grinding process such as wheel speed, workpiece speed, depth of dressing and lead of dressing, area of contact, grinding fluid, and so on. A significant improvement in the process efficiency may be obtained by optimization of these process parameters that identifies and determines the regions of critical process control factors leading to desired outputs with acceptable variations ensuring a lowest cost of manufacturing.

Previous work on the optimization of grinding parameters has concentrated on possible approaches for optimizing constraints during grinding. Amitay [**1**] reported the technique of optimizing both grinding and dressing conditions for the maximum workpiece removal rate subjected to constraints on workpiece burn and surface finish in an adaptive control system. Wen *et al.* [**2**] applied the successive quadratic programming (QP) approach using a multi-objective function model with a weighted approach for optimization of surface-grinding process parameters. However, using this approach the convergence to an optimal solution depends on the chosen initial solution. Also, the algorithm tends to become stuck to the local optimal solution. Rowe *et al.* [**3**] provided an extensive review of various approaches based on the application of artificial intelligence to the grinding process. A genetic algorithm (GA)-based optimization procedure has been developed by Saravanan *et al.* [**4**] to optimize the grinding conditions. However, the GA has its own limitations such as risk of replacement of a good parent string with the deteriorated child, less convergence speed, and difficulty in selecting the controlling parameters such as population size, crossover rate, and mutation rate. Also, the results of GA presented by the authors are erroneous. Dhavalikar *et al.* [**5**] applied combined Taguchi and dual response methodology to determine the robust condition for minimization of out-of-roundness error of a workpiece for centreless grinding operation. Optimization was then carried out by using the Monte Carlo simulation procedure. Mitra and Gopinath [**6**] used non-dominated sorting GAs for the multi-objective optimization of industrial grinding process. Krishna

*Corresponding author: Department of Mechanical Engineering, SV National Institute of Technology, Ichchanath, Surat, Gujarat 395 007, India.
email: ravipudirao@gmail.com*

[7] applied a differential evolution (DE) algorithm for the optimization of the process parameters of the grinding operation. However, the solutions obtained using the differential algorithm are erroneous for rough grinding operations whereas, for the finish grinding operation the optimum values suggested by the author lie outside their respective bounds and hence, the solution is not valid.

Although few non-traditional methods are applied to the optimization of process parameters of the grinding operation, the efforts must be continued to use more recent optimization algorithms, which are more powerful, robust, and able to provide accurate solutions. The artificial bee colony (ABC) algorithm developed by Karaboga [**8**] and Karaboga and Basturk [**9**, **10**] is one of the most recent algorithms and no effort has been made yet for the 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 ABC. For purposes of comparison, other non-traditional methods of optimization such as the harmony search (HS) and simulated annealing (SA) algorithms have also been tried for optimization of the process parameters of the grinding operation.

Section 2 provides the details of the optimization model of the grinding process used in the present work.

2 OPTIMIZATION MODEL OF THE GRINDING PROCESS

The optimization model for the grinding process formulated in the present work is based on the analysis given by Wen *et al.* [**2**]**.** The four decision variables considered for this model are: wheel speed 'V_s' (m/min), workpiece speed 'V_w' (m/min), depth of dressing '*doc*' (mm), and lead of dressing '*L*' (mm/rev).

2.1 Objectives

The three objectives considered in this work are:

(a) minimization of production cost 'C_T '($/pc);
(b) maximization of the production rate in terms of workpiece removal parameter 'WRP' (mm^3/min N).
(c) minimization of surface roughness 'R_a' (μm).

However, bearing in mind the specific requirements of the finish grinding and rough grinding operations, these three objectives are divided into two groups as follows:

1. For the rough grinding operation the following two objective functions are considered with the condition that the surface roughness value should not exceed 1.8 μm:

(a) minimization of production cost (C_T) in $/piece;
(b) maximization of the production rate in terms of workpiece removal parameter 'WRP' (mm^3/min N).

2. For the finish grinding operation the following two objective functions are considered with the condition that the workpiece removal parameter should not be less than 20 mm^3/min N:

(a) minimization of production cost 'C_T '($/pc);
(b) minimization of surface roughness 'R_a' (μm).

These three objective functions 'C_T', 'WRP', 'R_a' can be expressed in terms of the process variables as given by equations (1), (2), and (3) respectively [**2**].

$$C_T = \frac{M_c}{60p}\left(\frac{L_w + L_e}{V_w 1000}\right)\left(\frac{b_w + b_e}{f_b}\right)\left(\frac{a_w}{a_p} + S_p + \frac{a_w b_w L_w}{\pi D_e b_s a_p G}\right) + \frac{M_c}{60p}\left(\frac{S_d}{V_r} + t_1\right) + \frac{M_c t_{ch}}{60N_t} + \frac{M_c \pi b_s D_e}{60pN_d L V_s 1000} + C_s\left(\frac{a_w b_w L_w}{pG} + \frac{\pi(doc) b_s D_e}{pN_d}\right) + \frac{C_d}{pN_{td}} \quad (1)$$

where M_c is the cost per hour labour and administration, L_w is the length of the workpiece, L_e is the empty length of grinding, b_w is the width of the workpiece, b_e is the empty width of the grinding, f_b is the cross feed rate, a_w is the total thickness of cut, a_p is the down feed of the grinding, S_p is the number of the spark out grinding, D_e is the diameter of the wheel, b_s is the width of the wheel, G is the grinding ratio, S_d is the distance of the wheel idling, p is the number of workpieces loaded on the table, V_r is the speed of wheel idling, t_1 is the time of loading and unloading workpieces, t_{ch} is the time of adjusting the machine tool, N_t is the batch size of the workpieces, N_d is the total number of workpieces to be ground between two dressings, N_{td} is the total number of workpieces to be ground during the life of the dressing, and C_d is cost of dressing.

$$\mathrm{WRP} = 94.4\frac{(1 + (2doc/3L))L^{11/19}(V_w/V_s)^{3/19}V_s}{D_e^{43/304} VOL^{0.47} d_g^{5/38} R_c^{27/19}} \quad (2)$$

where VOL is the wheel bond percentage, d_g is the grind size, R_c is the workpiece hardness.

$$R_a = 0.4587T_{ave}^{0.30} \quad \text{for} \quad 0 < T_{ave} < 0.254$$
$$\text{else, } R_a = 0.78667T_{ave}^{0.72} \quad \text{for} \quad 0.254 < T_{ave} < 2.54 \quad (3)$$

where

$$T_{ave} = 12.5\times 10^3 \frac{d_g^{16/27} a_p^{19/27}}{D_e^{8/27}}\left(1 + \frac{doc}{L}\right)L^{16/27}\left(\frac{V_w}{V_s}\right)^{16/27} \quad (4)$$

2.2 Constraints

Three constraints are considered in this optimization model [**2**].

2.1.1 Thermal damage constraint

The grinding process requires very high energy per unit volume of material removed. Whatever the energy that is concentrated within the grinding zone, it is converted into heat. The high thermal energy causes damage to the workpiece, and it leads to the reduced production rate. The specific energy U is calculated by equation (5)

$$U = 13.8 + \frac{9.64\times10^{-4}V_s}{a_p V_w} + \left(6.9\times10^{-3}\frac{2102.4V_w}{D_e V_s}\right) \times\left(A_0 + \frac{K_u V_s L_w a_w}{V_w D_e^{1/2} a_p^{1/2}}\right)\frac{V_s D_e^{1/2}}{V_w a_p^{1/2}} \quad (5)$$

where K_u is the wear constant.

The critical specific energy U^* at which burning starts is expressed in terms of the operating parameters as

$$U^* = 6.2 + 1.76\left(\frac{D_e^{1/4}}{a_p^{3/4} V_w^{1/2}}\right) \quad (6)$$

The thermal damage constraint is then specified as

$$U^* - U \geqslant 0 \quad (7)$$

2.1.2 Wheel wear parameter constraint

The wheel wear parameter WWP (mm^3/min N) is related directly to the grinding conditions. For single-point diamond dressing, it is given by equation (8).

$$WWP = \left(\frac{k_p a_p d_g^{5/38} R_c^{27/29}}{D_c^{1.2/VOL-43/304} VOL^{0.38}}\right) \times\frac{[1+(doc/L)]L^{27/19}(V_s/V_w)^{3/19}V_w}{[1+(2doc/3L)]} \quad (8)$$

From equations (2) and (8) the wheel wear constraint is obtained as

$$\frac{\mathrm{WRP}}{\mathrm{WWP}} - G \geqslant 0 \quad (9)$$

2.1.3 Machine tool stiffness constraint

Chatter results in poorer surface quality and lowers the machining production rate. Chatter avoidance is therefore a significant constraint in the selection of machining parameters. The relationship between grinding stiffness K_c (N/mm), wheel wear stiffness Ks (N/mm), and operating parameters during grinding is given below

$$K_c = \frac{1000 V_w f_b}{\mathrm{WRP}} \quad (10)$$

$$K_s = \frac{1000 V_s f_b}{\mathrm{WWP}} \quad (11)$$

To avoid chatter during machining, the constraint given by equation (12) has to be fulfilled

$$\mathrm{MSC} - \frac{|R_{em}|}{K_m} \geqslant 0 \quad (12)$$

where MSC is machine stiffness constraint and

$$\mathrm{MSC} = \frac{1}{2K_c}\left(1 + \frac{V_w}{V_s G}\right) + \frac{1}{K_s} \quad (13)$$

where R_{em} is the dynamic machine characteristics, K_m is the static machine stiffness.

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

3 PRESENTED NON-TRADITIONAL OPTIMIZATION ALGORITHMS

Three non-traditional optimization algorithms are considered in the present work for grinding parameter optimization and are described below.

3.1 Artificial bee colony algorithm

A branch of nature-inspired algorithms, called swarm intelligence, is focused on insect behaviour in order to develop some meta-heuristics which can mimic an 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 behaviour, learning, memorizing, and information-sharing characteristics of honey bees have recently been some of the most interesting research areas in swarm intelligence. The ABC algorithm has been developed to model the intelligent behaviours of honey bee swarms [**8–10**]. The honey bee swarms consist of two essential components (i.e. food sources and foragers) and define two leading modes of behaviour (i.e. recruitment to a nectar source and abandonment of a source). A flow chart for the ABC algorithm is shown in Fig. 1 [**11**]. Various steps used

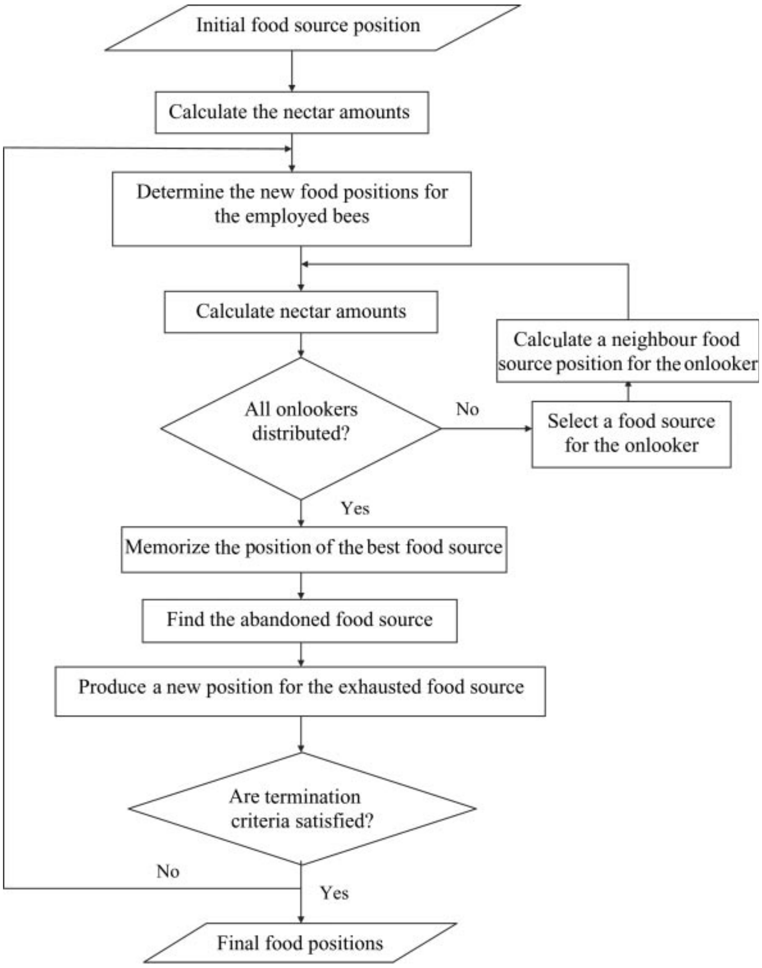

Fig. 1 Artificial bee colony algorithm [**11**]

for the optimization of the grinding operation using ABC algorithm are discussed below.

Step 1: Parameter selection

As discussed in the description of the ABC algorithm, food source represents a possible solution to the problem of minimization of production time in the present work. Five initial solutions (i.e. the number of food sources) were considered in this work. The value of each food source depends on the fitness value of the objective function given by equation (22).

For every food source there is only one employed bee (employed forager). In other words, the number of employed bees is equal to the number of food sources. Hence, in the present work the number of employed bees is considered to be five. The unemployed forager can be a scout or an onlooker bee. The number of onlooker bees must be greater than the number of employed bees. 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 value number of onlooker bees, any increment in the value does not improve the performance of the algorithm. For the problem considered in this work, the 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 the number of employed bees and the number of onlooker bees. Hence, the colony size is 16. The number of scout bees is usually 5–30 per cent of the colony size. In the present work, the number of scout bees is taken to be 5 per cent of the colony size, i.e. one. The parameters of optimization thus selected in this work are summarized below:

(a) number of employed bees = 5;
(b) number of onlookers bees = 11;
(c) number of scout bees = 1;
(d) maximum number of iterations = 150.

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 equation (22) subject to constraints given by equations (7), (9), and (12).

Step 3: Determine the probabilities by using the nectar amount

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

$$P_i = \frac{\left[\sum_{i=1}^{S} (1/f_i)\right]^{-1}}{f_i} \quad (14)$$

where S is the number of food sources.

Step 4: Calculate the number of onlooker bees that will be sent to food sources

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

$$N = P_i{}^{*}m \quad (15)$$

where m is the total number of onlooker bees.

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 equation (14). The position of the selected neighbour food source is calculated as shown in equation (16).

$$\theta_i(c+1) = \theta_i(c) \pm \phi_i(c) \quad (16)$$

where c is the generation number, $\phi_i\,(c)$ is a randomly produced step to find a food source with more available nectar 'θ_i'. The value of $\phi_i\,(c)$ 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 the bee becomes a scout. The scout starts searching for a new food source, and after finding the new source, the new position is accepted as 'θ_i'.

Step 6: Evaluate the best solution

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

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 solution. If the scout solution is better than the employed solution, then the employed solution is replaced with the scout solution. Otherwise, the employed solution is transferred to the next generation without any change.

3.2 Harmony search algorithm

The HS algorithm, which is a meta-heuristic optimization algorithm, has been recently developed by Geem *et al.* [**12**]. This algorithm is conceptualized from the musical process of searching for a perfect state of harmony, such as jazz improvement. The jazz improvisation seeks the best state (fantastic harmony) determined by an estimation performed by a set of pitches played by each instrument.

The various steps used for the optimization of the grinding operation using the harmony search algorithm are discussed below.

Step 1: Determine algorithm parameters

In this step, the optimization problem is specified in terms of objective functions, constraints, and decision variables along with their upper and lower-bound values. The HS algorithm parameters are the harmony memory size (HMS), or the number of solution vectors in the harmony memory; the harmony memory considering rate (HMCR); the pitch adjusting rate (PAR); the number of decision variables (N), and the number of improvisations (NI), or stopping criteria. For the harmony search algorithm a value of HMCR is considered between 0.7–0.95 and a PAR is considered between 0.05–0.7. For the present example the following values of the algorithm parameters are selected after various trials:

(a) HMS = 5;
(b) HMCR = 0.9;
(c) PAR = 0.4;
(d) NI = 150.

Step 2: Improvise a new harmony

A new harmony vector is generated based on three rules: (a) memory consideration, (b) pitch adjustment, and (c) random selection. Generating a new harmony is called 'improvisation'. In the memory consideration stage, the value of the first decision variable (x_1) for the new vector is chosen from any of the values in the specified HM range (x_1–x_{HMS}). Values of the other decision variables are chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, while (1–HMCR) is the rate of randomly selecting one value from the possible range of values, as shown in the following equation:

$$\begin{aligned}&\text{If} \quad \mathrm{HMCR} > \mathrm{rand}(), x_i^{'} \in \{x_i^1, x_i^2, \ldots\ldots x_i^{\mathrm{HMS}}\}\\ &\text{otherwise} \quad x_i^{'} \in X_i\end{aligned} \quad (17)$$

where, rand () is a random number between 0 and 1. Every component obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment as follows:

$$\begin{aligned} &\text{If} \quad \text{PAR} > \text{rand}(), x_i' = x_i^{\cdot} \\ &\text{then} \quad x_i' = x_i' \pm rand() \times bw \end{aligned} \tag{18}$$

where *bw* is an arbitrary distance bandwidth. In this step, harmony memory consideration and pitch adjustment are applied to each variable of the new harmony vector one by one.

Step 3: Update harmony memory

If the new harmony vector has better fitness function than the worst harmony in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM. The harmony search algorithm is shown in Fig. 2.

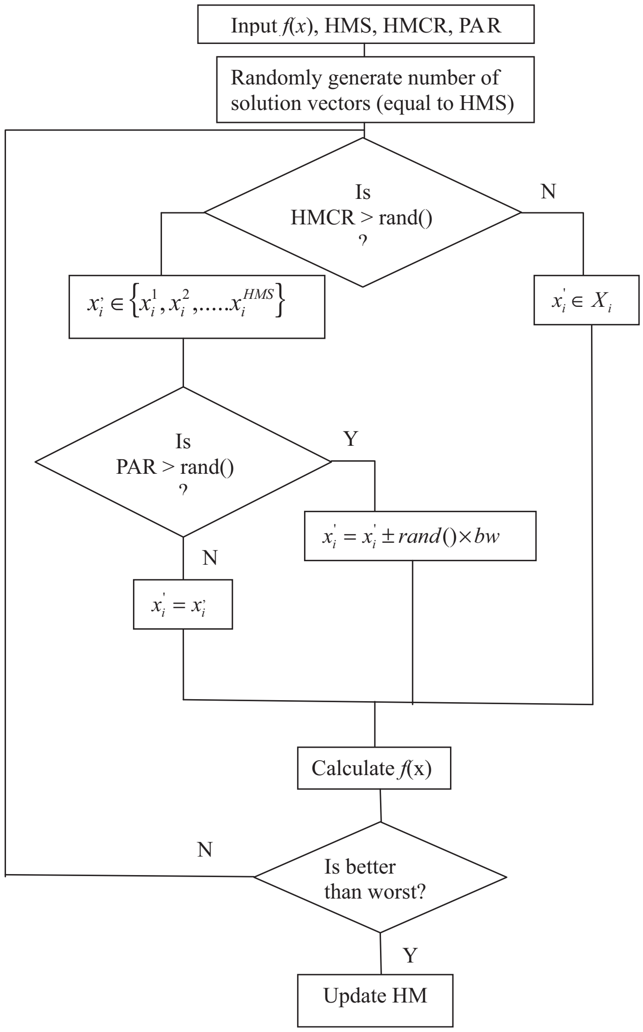

Fig. 2 Harmony search algorithm [**12**]

3.3 Simulated annealing algorithm

Simulated annealing (SA) is a probabilistic hill-climbing algorithm in which if '*i*' is the current configuration with cost $C(i)$, then using the Metropolis algorithm [**13**], the probability of accepting '*j*' as the 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

$$\Pr\{new = j | current = i\} = \left\{ \begin{array}{cc} 1 & if \quad \Delta C \leqslant 0 \\ e^{-\Delta C/T} & otherwise \end{array} \right\} \tag{19}$$

For the SA technique, the parameters of optimization are the initial temperature, decrement factor, and number of improvisations. The initial temperature (T_0) can be obtained by calculating the average of the function values at a boundary point

$$T_0 = \Sigma Z_{Nb}/n \tag{20}$$

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

$$x(t+1) = x(t) + \sigma \times \sum_{i=1}^{N} R_i - \frac{N}{2} \tag{21}$$

where $\sigma = (x_{\max} - x_{\min})/6$, R is the random number; and N is the number of random numbers used. In the present work, six 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.

The software for the optimal selection of process parameters in the grinding process using ABC, HS, and SA is written in C++ language and has been implemented on a Pentium-IV system.

Section 4 provides two examples to demonstrate and validate the application of the presented ABC, HS, and SA algorithms.

4 APPLICATION EXAMPLES

To demonstrate and validate the proposed algorithms, two examples are considered for the optimization of grinding process parameters.

4.1 Example 1

This example presents the multi-objective optimization of the rough grinding process. The combined objective function (to be minimized) formulated for the rough grinding operation (Z_R) is given in equation (22)

$$\mathrm{Min}Z_R = W_1{}^*(C_T/C_T{}^*) - W_2{}^*(\mathrm{WRP}/\mathrm{WRP}^*) \quad (22)$$

where W_1 and W_2 are the weighting factors with value 0.5 each. This is subject to the constraints specified by equations (7), (9), and (12).

Parameter bounds for the four process variables are as follows

$1000 \leqslant V_s \leqslant 2023$ m/min
$10 \leqslant V_w \leqslant 22.70$ m/min
$0.01 \leqslant doc \leqslant 0.1370$ mm
$0.01 \leqslant L \leqslant 0.1370$ mm/rev

Values of the constants and parameters considered in the present work are as given in Table 1. The optimum process parameter values obtained by using the ABC algorithm are given below:

(a) wheel speed (V_s) = 2023 (m/min);
(b) workpiece speed (V_w) = 10.973 (m/min);
(c) depth of dressing (doc) = 0.097 (mm);
(d) lead of dressing (L) = 0.137 (mm/rev);
(e) total production cost (C_T) = 7.942 \$/piece;
(f) workpiece removal parameter (WRP) = 25.00 mm^3/min-N;
(g) surface roughness (R_a) = 1.80 μ.

Optimality of the above-mentioned solution could be confirmed from Figs 3 to 6. Figure 3 shows variation of the wheel wear parameter constraint, surface roughness constraint, and combined objective function with wheel speed. Since the thermal damage constraint and machine tool stiffness constraint have almost constant positive values for all values of wheel speed, Fig. 3 is plotted neglecting thermal damage constraint and machine tool stiffness constraint to indicate more clearly the variation of the other two constraints with wheel speed. As shown in Fig. 3, the combined objective function value reduces as wheel speed increases. This is because with increase in wheel speed, the workpiece removal parameter increases without affecting cost. Moreover, the surface roughness constraint is satisfied only at a wheel speed of 2023 m/min. Hence, the optimum value of wheel speed selected at its upper bound value of 2023 m/min is appropriate. If the wheel speed is increased, the size of the chips removed by a single abrasive grain is reduced, which in turn reduces the wear of the wheel. Thus, from the point of view of wear also, it is better to operate at a higher wheel speed.

Figure 4 shows the variation of thermal damage constraint, wheel wear parameter constraint, surface roughness constraint, and combined objective function with workpiece speed. Figure 4 is plotted neglecting the machine tool stiffness constraint as it has almost constant positive values for all values of workpiece speed. As shown in Fig. 4, the combined objective function value reduces (as the workpiece removal parameter increases and cost reduces) as the workpiece speed increases. Thus, the higher value of workpiece speed is desirable. However, at any value of workpiece speed higher than 10.973 m/min, the surface roughness constraint is violated. This is due to the fact that, if the workpiece speed is high, the wheel wear will be high.

Figure 5 shows the variation of the wheel wear parameter constraint, surface roughness constraint, and combined objective function with depth of dressing. Since the thermal damage constraint and machine tool stiffness constraint have almost constant positive values for all values of wheel speed, Fig. 5 is plotted neglecting the thermal damage constraint and the machine tool stiffness constraint. As

Table 1 Values of the constants and parameters used in grinding process parameter optimization

Notation	Description	Unit	Value
a_p	Down feed of grinding	mm/pass	0.0505
a_w	Total thickness of cut	mm	0.1
b_e	Empty width of grinding	mm	25
b_s	Width of wheel	mm	25
b_w	Width of workpiece	mm	60
C_s	Cost of wheel per mm^3	\$	0.003
C_d	Cost of dressing	\$	25
d_g	Grind size	mm	0.3
D_e	Diameter of wheel	mm	355
f_b	Cross feed rate	mm/pass	2
G	Grinding ratio		60
K_a	Constant dependent on coolant and grain type		0.0869
K_m	Static machine stiffness	N/mm	100 000
K_u	Wear constant	mm^{-1}	3.937×10^{-7}
L_e	Empty length of grinding	mm	150
L_w	Length of workpiece	mm	300
M_c	Cost per hour of labour and administration	\$/hr	30
N_d	Total number of workpieces to be ground between two dressings		20
N_t	Batch size of the workpieces		12
N_{td}	Total number of workpieces to be ground during the life of the dresser		2000
p	Number of workpieces loaded on the table		1
R_c	Workpiece hardness	HRC	58
R_{em}	Dynamic machine characteristics		1
S_d	Distance of wheel idling	mm	100
S_p	Number of spark out grinding		2
t_1	Time of loading and unloading workpieces	min	5
t_{ch}	Time of adjusting machine tool	min	30
V_r	Speed of wheel idling	mm/min	254
VOL	Wheel bond percentage		6.99

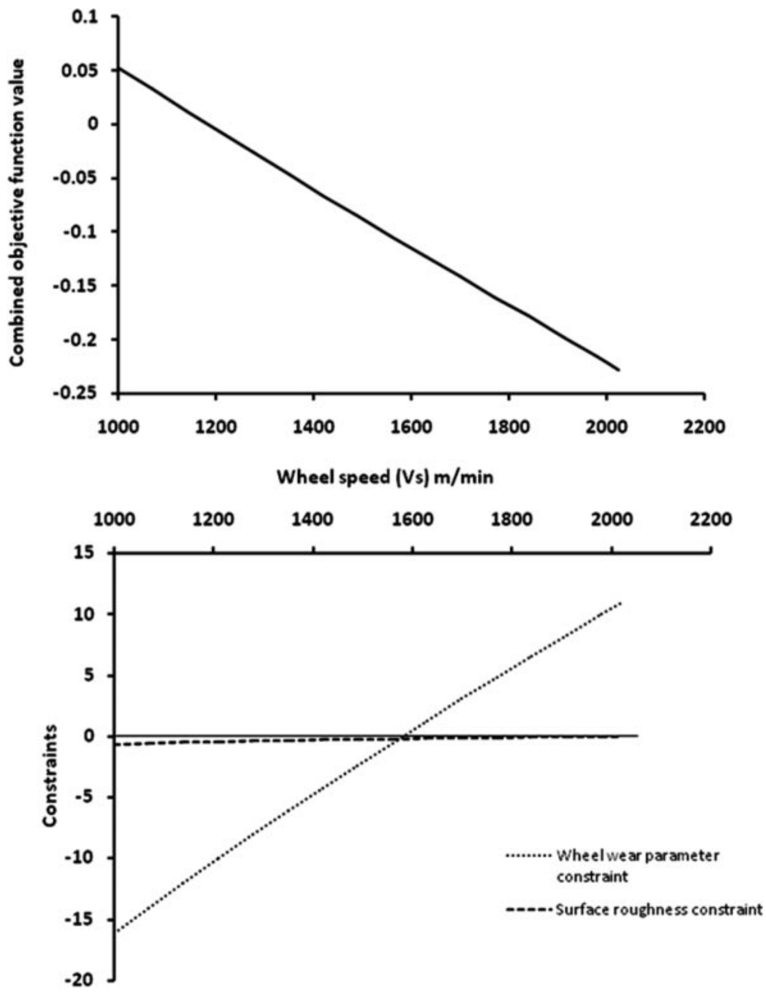

Fig. 3 Variation of wheel wear parameter constraint, surface roughness constraint, and combined objective function with wheel speed (V_s)

shown in Fig. 5, the combined objective function value decreases with the increase in depth of dressing. Thus, the higher value of depth of dressing is desirable. However, for any value of depth of dressing higher than 0.097 mm, the surface roughness constraint is violated. This confirms the optimum value depth of dressing selected using the particle swarm optimization algorithm for the rough grinding operation. Figure 6 shows variation of wheel wear parameter constraint, surface roughness constraint, and the combined objective function with lead of dressing. Since the thermal damage constraint and machine tool stiffness constraint have almost constant positive values for all values of wheel speed, Fig. 6 is plotted neglecting the thermal damage constraint and machine tool stiffness constraint. As shown in Fig. 6, the combined objective function value initially increases up to a certain value and thereafter decreases with the increase in the lead of dressing. Thus, the minimum value of the combined objective function occurred at both lower bound and upper bound values of lead of dressing. However, the upper bound value of lead of dressing should be selected, as at lower bound value of lead of dressing, the surface roughness constraint is violated.

Table 2 shows the optimum process parameter data for the above example using ABC, HS, and SA algorithms along with the previously published results obtained by using other methods. As shown in Table 2, although the result of optimization using the DE algorithm [**7**] seems to be better than that using ABC, HS, and SA algorithms, it is erroneous and the corrected result is not valid as the surface roughness value (1.87 μm) exceeds the permissible value (1.80 μm) for the given parameter combination. Table 3 shows the improvement in combined objective function for rough grinding using various algorithms rather than quadratic programming [**2**]. Figure 7 shows the convergence of ABC, HS, and SA algorithms for rough grinding operations. As shown in Fig. 7, the convergence rate of the ABC algorithm is better than those of the HS and SA algorithms.

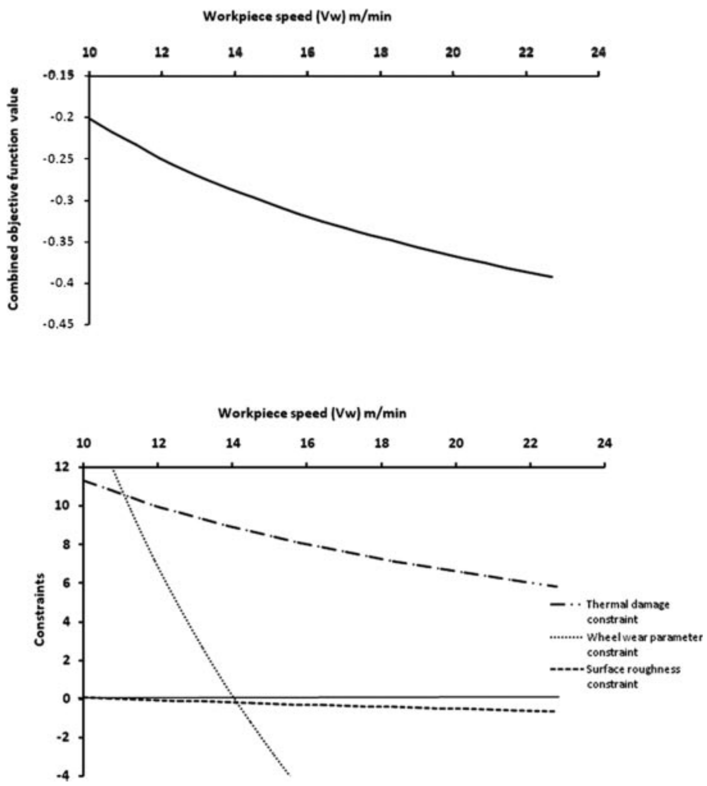

Fig. 4 Variation of thermal damage constraint, wheel wear parameter constraint, surface roughness constraint, and combined objective function with workpiece speed (V_w)

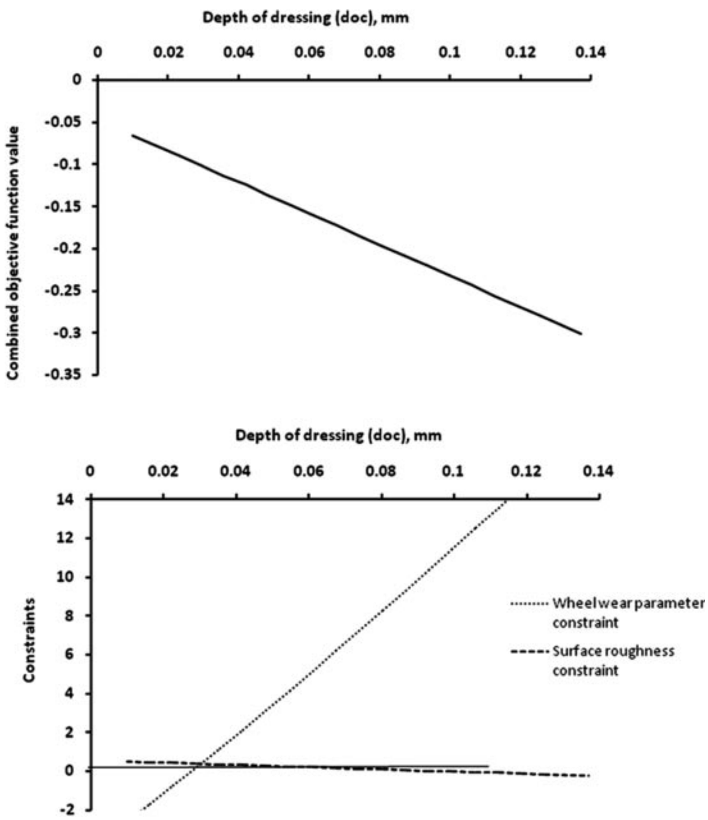

Fig. 5 Variation of wheel wear parameter constraint, surface roughness constraint, and combined objective function with depth of dressing (*doc*)

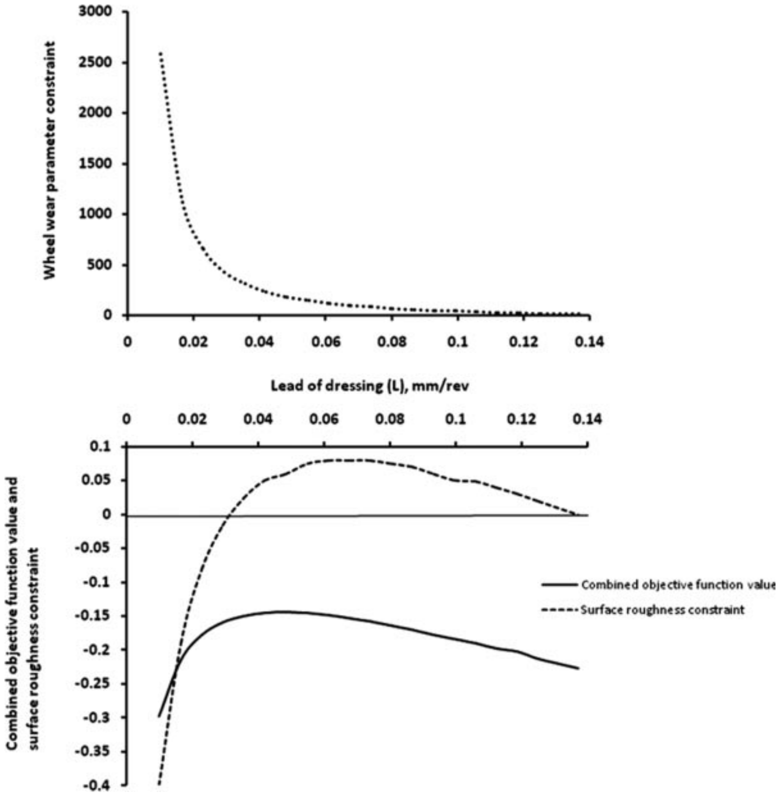

Fig. 6 Variation of wheel wear parameter constraint, surface roughness constraint, and combined objective function with lead of dressing (L)

Table 2 Results of optimization for rough grinding operation

Method	Authors	V_s	V_w	*doc*	L	C_T	WRP	R_a	Z_R
QP	Wen *et al.* [**2**]	2000	19.96	0.055	0.044	6.2	17.47	1.74	−0.127
GA	Saravanan *et al.* [**4**]	1998	11.30	0.101	0.065	7.1	21.68	1.79	−0.187
DE	Krishna [**7**]	2023	10.00	0.130	0.109	7.9	26.57	1.80*	−0.249
DE		2023	10.00	0.130	0.109	7.9	26.57	1.87†	−0.249
SA		2023	11.48	0.089	0.137	7.755	24.45	1.789	−0.223
HS		2019.35	12.455	0.079	0.136	7.455	23.89	1.796	−0.225
ABC		2023	10.973	0.097	0.137	7.942	25.00	1.80	−0.226

* Values wrongly calculated by Krishna [**7**],
† Corrected values

Table 3 Improvement in combined objective function in rough grinding operation using various algorithms over that using quadratic programming (QP)

Method	Authors	COF	% improvement (over QP)
QP	Wen *et al.* [**2**]	−0.127	—
GA	Saravanan *et al.* [**4**]	−0.187	47.24
SA		−0.223	75.59
HS		−0.225	77.16
ABC		−0.226	78.00

4.2 Example 2

This example presents the multi-objective optimization of the finish grinding process. The combined objective function formulated for the finish grinding operation (Z_F) is given in equation (23)

$$\text{Min.}Z_F = W_1{*}(C_T/C_T{*}) + W_3{*}(R_a/R_a{*}) \tag{23}$$

where W_1 and W_3 are the weighting factors with values 0.3 and 0.7 respectively, subjected to the constraints specified by equations (7), (9), and (12). The

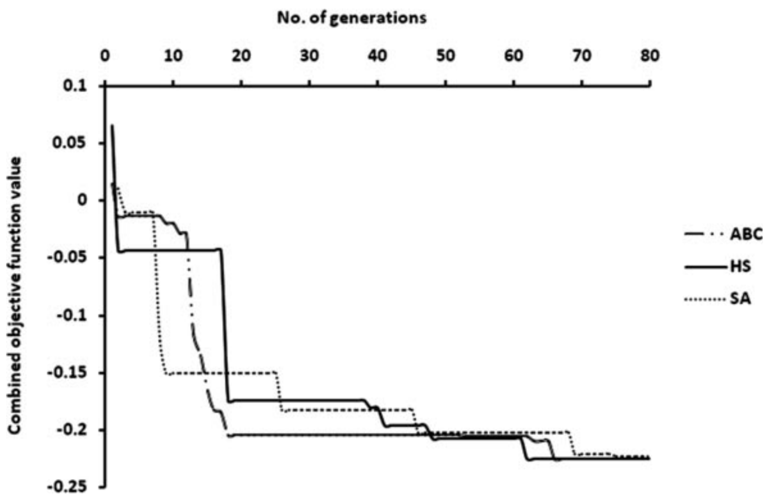

Fig. 7 Convergence of ABC, HS, and SA algorithms for rough grinding

Table 4 Results of optimization for finish grinding operation

Method	Author	V_s	V_w	*doc*	*L*	C_T	WRP	R_a	Z_F
QP	Wen *et al.* [**2**]	2000	19.99	0.052	0.091	7.7	20.00	0.83	0.554
GA	Saravanan *et al.* [**4**]	1986	21.40	0.024	0.136	6.6*	20.08	0.83	0.521**a**
GA		1986	21.40	0.024	0.136	7.36†	20.08	0.83	0.542**b**
DE	Krishna [**7**]	2170	17.49	0.008	0.137	7.48	20.33	0.65	0.497
SA		2023	22.7	0.01	0.137	7.11	20.01	0.79	0.520
HS		2023	22.7	0.01	0.137	7.11	20.01	0.79	0.520
ABC		2023	22.7	0.01	0.137	7.11	20.01	0.79	0.520

* Values wrongly calculated by Saravanan *et al.* [**4**],
† Corrected values

Table 5 Improvement in the combined objective function in rough grinding operation using various algorithms over that using quadratic programming

Method	Author	COF	% improvement (over QP)
QP	Wen *et al.* [**2**]	0.554	—
GA	Saravanan *et al.* [**4**]	0.542	2.30
ABC, HS and SA		0.520	6.54

parameter bounds for the four process variables are the same as those given in example 1.

Table 4 shows the optimum process parameter data for the above example, along with the previously published results using other methods. As shown in Table 4, although the result of optimization using the DE algorithm **[7]** seems to be better than those of the SA, HS, and ABC algorithms, this is not valid as the values of some process parameters such as wheel speed (V_s) and depth of dressing (*doc*) lie outside their respective bounds ($V_s = 2170 > 2023$ and $doc = 0.008 < 0.01$). The result obtained by using GE **[4]** is erroneous. Table 5 shows the improvement in the combined objective function for finish grinding using various algorithms over that of quadratic programming **[2]**.

5 CONCLUSIONS

In the present work, multi-objective optimization aspects of rough grinding as well as finish grinding process parameters are considered using three non-traditional algorithms; ABC, HS, and SA. The three objectives considered are: minimization of production cost, maximization of production rate, and maximization of surface finish, i.e. minimization of roughness value subjected to the constraints of thermal damage, wheel wear parameter, and machine tool stiffness.

The performance of three non-traditional optimization algorithms such as ABC, HS, and SA is studied in terms of convergence rate and accuracy of the solution. It can be seen from Table 3 that for the rough grinding operation, the ABC algorithm outperformed all other algorithms, i.e. quadratic programming, GA, HS, and SA, showing significant improvement of 78 per cent over quadratic programming. This improvement is due mainly to the fact that the ABC algorithm combines both the stochastic selection scheme carried out by onlooker bees, and the greedy selection scheme used by onlookers and employed bees to update the source position. Also, the neighbour source production mechanism in ABC is similar to the mutation process, which is self-adapting. The random selection process carried out by the scout bees maintains diversity in the solution. Table 5 shows that the accuracy of solution for process parameter optimization of the finish grinding operation obtained using the ABC, HS, and SA algorithms is equally good. The convergence rate of the ABC algorithm is better than the HS and SA algorithms. The presented ABC, HS, and SA algorithms can be easily modified to suit optimization of process parameters of other machining processes such as milling, turning, drilling, and so on.

REFERENCES

1 **Amitay, G.** Adaptive control optimization of grinding. *J. Engng Ind.*, 1981, **103**(1), 103–108.

2 **Wen, X. M., Tay, A. A. O.,** and **Nee, A. Y. C.** Microcomputer based optimization of the surface grinding process. *J. Mater. Process. Technol.*, 1992, **29**, 75–90.

3 **Rowe, W. B., Yan, L., Inasaki, I.,** and **Malkin, S.** Application of artificial intelligence in grinding. *CIRP ann.*, 1994, **43**(2), 521–531.

4 **Saravanan, R., Asokan, P.,** and **Sachidanandam, M.** A multi-objective genetic algorithm approach for optimization of surface grinding operations. *Int. J. Mach. Tools Mf.*, 2002, **42**, 1327–1334.

5 **Dhavalikar, M. N., Kulkarni, M. S.,** and **Mariappan, V.** Combined Taguchi and dual response method for optimization of a centerless grinding operation. *J. Mater. Process. Technol.*, 2003, **132**, 90–94.

6 **Mitra, K.** and **Gopinath, R.** Multiobjective optimization of an industrial grinding operation using elitist non-dominated sorting genetic algorithm. *Chem. Engng Sci.*, 2004, **59**, 385–396.

7 **Krishna, A. G.** Optimization of surface grinding operations using a differential evolution approach. *J. Mater. Process. Technol.*, 2007, **183**, 202–209.

8 **Karaboga, D.** An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Computer Engineering Department, Erciyes University, Turkey, 2005.

9 **Karaboga, D.** and **Basturk, B.** A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Global Optim.*, 2003, **39**, 459–471.

10 **Karaboga, D.** and **Basturk, B.** On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.*, 2008, **8**, 687–697.

11 **Karaboga, N.** A new design method based on artificial bee colony algorithm for digital IIR filters. *J. Franklin Inst.*, 2009, **346**, 328–348.

12 **Geem, Z. W., Kim, J. H.,** and **Loganathan, G. V.** A new heuristic optimization algorithm: harmony search. *Simulation*, 2001, **76**(2), 60–68.

13 **Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A.,** and **Teller, E.** Equation of state calculations by fast computing machines. *J. Chem. Phys*, 1953, **21**, 1087–1092.