Neural Comput & Applic (2018) 29:1321–1336
https://doi.org/10.1007/s00521-016-2644-6

ORIGINAL ARTICLE

Optimization of production time in the multi-pass milling process via a Robust Grey Wolf Optimizer

Soheyl Khalilpourazari[1] · Saman Khalilpourazary[2]

Received: 12 August 2016 / Accepted: 21 October 2016 / Published online: 3 November 2016

Abstract In order to optimize multi-pass milling process, selection of optimal values for the parameters of the process is of great importance. The mathematical model for optimization of multi-pass milling process is a multi-constrained nonlinear programing formulation which is hard to be solved. Therefore, a novel robust meta-heuristic algorithm named Robust Grey Wolf Optimizer (RGWO) is proposed. In order to develop a RGWO, a robust design methodology named Taguchi method is utilized to tune the parameters of the algorithm. Therefore, in contradiction to previous researches, there is no need to design costly experiments to obtain the optimal values of the parameters of the GWO. In addition, an efficient constraint handling approach is implemented to handle complex constraints of the problem. A real-world problem is adopted to show the effectiveness and efficiency of the proposed RGWO in optimizing the milling process within different strategies. The results indicated that the RGWO outperforms the other solution methods in the literature as well as two novel meta-heuristic algorithms by obtaining better and feasible solutions for all cutting strategies.

Keywords Multi-pass milling · Robust design · Grey Wolf Optimizer · Taguchi method · Constraint handling

✉ Saman Khalilpourazary
s.khalilpour@mee.uut.ac.ir

Soheyl Khalilpourazari
std_khalilpourazari@khu.ac.ir

[1] Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran

[2] Department of Mechanical Engineering, Urmia University of Technology, Urmia, Iran

1 Introduction

Solving real-world optimization problems is one of the most challenging tasks. Most of the real-world optimization problems are complex; therefore, many researchers paid attention to propose new optimization methods. In recent years, different meta-heuristic algorithms are developed inspiring from nature, mathematics, physics or even animal behaviour to solve complex real-world problems, efficiently. Mirjalili et al. [1] proposed a novel meta-heuristic algorithm named multi-verse optimizer (MVO) which mimics the physic laws in universe. It uses the definition of black holes and white holes to perform optimization. Mirjalili et al. [1] investigated the performance of the MVO against different state-of-the-art algorithms and showed that the MVO is able to provide very competitive results even in challenging composite benchmark functions. In a different research, Saremi et al. [2] presented a chaos version of the biogeography-based optimization (BBO). They showed that their modification significantly improved the performance of the algorithm. Salimi [3] presented a new meta-heuristic algorithm named stochastic fractal search (SFS). The SFS is based on strong mathematical concepts such as Levy flight and Gaussian walks. He showed that the SFS is very competitive with other algorithms and outperforms them in majority of the benchmark functions. Since most of the optimization problems deal with different types of decision variables such as binary variables, many researchers paid attention to modify existing algorithms to solve various complex problems. For example, Mirjalili et al. [4] proposed a binary bat algorithm to solve complex binary optimization problems. For more information, see [5].

One of the most recently developed algorithms is Grey Wolf Optimizer (GWO). The GWO first proposed by

Mirjalili et al. [6]. GWO mimics the hunting behaviour of the grey wolves in nature. GWO is designed very intelligently, which enables it to make an appropriate trade-off between exploration and exploitation abilities of the algorithm. Mirjalili et al. [6] compared the performance of the GWO against state-of-the-art meta-heuristic algorithms and showed that the GWO is able to outperform other algorithms in most of the benchmark functions. Since the GWO performed very efficient in solving complex benchmark functions, many researches started to implement it to solve complex optimization problems in different fields of study. Jayakumar et al. [7] used GWO for optimization of combined heat and power dispatch with cogeneration systems. They showed that the GWO is able to solve the problem, efficiently. Song et al. [8] implemented GWO for parameter estimation in surface waves and demonstrated the superiority of the GWO. Pradhan et al. [9] showed that GWO performs very effective in solving economic load dispatch problem. By solving different problems, Medjahed er al. [10] showed that the GWO is able to solve the hyperspectral band selection problem efficiently. In addition to above researches, which aimed to develop new techniques to solve complex problems or to show the efficiency of the novel algorithms, some researchers aimed to optimize the performance of the existing algorithms. One of the most commonly used approaches to optimizing the efficiency of the meta-heuristic algorithms is to calibrate the values of input parameters of the algorithms using a robust design methodology. One of the most widely used robust design methods is Taguchi method [11, 12]. Najafi et al. [13] used Taguchi method to tune the parameters of the genetic algorithm (GA) to solve a complex resource investment problem. Sadeghi et al. [14] utilized Taguchi method to calibrate the parameters of the GA to solve a constrained nonlinear programming model of the vendor-managed inventory (VMI) problem. To show the efficiency of the Taguchi method, Sadeghi et al. [15] used Taguchi method to design a robust GA to solve a NP-hard hybrid problem.

In this research, the aim is to optimize the milling process which is one of the most widely used machining processes. In milling process, to enhance the final quality of machined products, many researches focused on the optimization of the total production cost, total production time, final surface roughness and material removal rate. In this category, the input parameters of the machining process must be selected attentively to achieve the favourable outcomes. Instead of single pass, multi-pass machining has been utilized in turning, boring and face milling processes to decrease the power consumption, machining forces, chatter phenomena and depreciation rate of the machine tools. Many experimental and analytical methods have been developed to optimize multi-pass milling parameters considering operational constraints such as permissible cutting force, power and tool life. The main parameters of the problem include the number of passes, depth of cut, cutting speed and feed rate. Therefore, there are different variables and operational constraints that must be considered in multi-pass milling process which results in a constrained nonlinear programing (NLP) model. Due to nonlinearity and complexity of the problem, traditional optimization algorithms usually obtain local optimal solutions. Conversely, meta-heuristic algorithms based on their exploration ability can obtain better solutions comparing to the traditional optimization techniques. In recent decades, scholars utilized meta-heuristic algorithms to optimize multi-pass milling process.

To specify the number of passes and optimum values of the input parameters in multi-pass milling process, Sonmez et al. [16] developed a dynamic programming (DP) and utilized geometric programming (GP) to determine the optimal cutting variables. The results demonstrated that different depths of cut are better than equal depths for multi-pass milling process. Wang et al. [17] proposed a combinational optimization method using GA and simulated annealing (SA). The archived values for the cutting speed, feed rate and machining time have been compared with the DP and conventional parallel GA (PGA). The results showed the high capability of the presented method to find proper values for milling parameters. Onwubolu [18] utilized various depths of cut to minimize production time with a new optimization method called Tribes.

To find the maximum total profit rate in the process, Baskar et al. [19] introduced a new algorithm using the genetic algorithm and hill climbing algorithm for optimization of the milling parameters. Yildiz [20] proposed a combinational optimization method to find maximum value of the total profit rate in milling processes and compared the results with genetic algorithm in the literature. Gao et al. [21] used cellular particle swarm optimization (CPSO) technique to optimize multi-pass milling process. The comparison showed that their approach was better than other optimization algorithms in obtaining more proper values for milling parameters. Other algorithms have been introduced by Rao and Pawar [22] and Pawar and Rao [23] named the artificial bee colony (ABC) and teaching–learning-based optimization algorithm (TLBO) to optimize machining processes, respectively. Yang et al. [24] used a plain method to predict the optimum values of the multi-pass milling parameters. Recently, Mellal and Williams [25] suggested a cuckoo search optimization algorithm approach to minimizing the total production time in the milling process. The results compared to the previous researches in the literature and showed that cuckoo algorithm is an appropriate optimization method to specify minimum total production time considering process constraints and variables. In different research, Mellal and Williams [26] used a cuckoo optimization algorithm to optimize machining process parameters. They showed that the CS is able to provide very competitive results. In more recent research, Mellal and Williams [27] utilized a cuckoo optimization algorithm to optimize production costs in multi-pass milling process. Beside the wide application of meta-heuristic algorithms, many researchers aimed to optimize machining, turning and grinding processes using hybrid methods which use neural networks and meta-heuristic algorithms, simultaneously. For more information, see [28–32].

Most of the researches in the literature proposed a meta-heuristic algorithm to optimize the parameters of the milling process and find better solutions. Since the proposed mathematical formulation is a nonlinear programing formulation (NLP) and hard to be solved [33], a meta-heuristic algorithm is needed to solve the problem which can make a proper trade-off between exploration and exploitation phrases of the algorithm to avoid trapping in local optima and fast convergence, simultaneously.

In this paper, a robust nature inspired algorithm named Robust Grey Wolf Optimizer (RGWO) is developed to optimize multi-pass milling process parameters. Since the values of the main parameters of meta-heuristic algorithms significantly affect the performance, therefore Taguchi method is utilized to obtain the optimal values of the main parameters of the algorithm to increase its efficiency, while in previous researches in the literature, the values of the main parameters of the developed algorithms are determined by trial and error. In addition, an effective constraint handling technique is used to handle complex operational constraints of the problem. Efficiency and effectiveness of the proposed RGWO is investigated by solving a real-world problem proposed by Sonmez et al. [16]. The results are compared to the previous researches and demonstrated that the RGWO can find significantly better solutions comparing to other solution methodologies in the literature.

2 Mathematical model

In this research, the mathematical formulation presented by Sonmez et al. [16] is implemented to optimize the multi-pass milling process parameters, where the main parameters of the milling process are considered as decision variables of the problem include: feed per tooth and cutting speed. The depth of cut (a) as one of the main parameters of the process is considered to be known. Using different strategies, the value of depth of cut is given in the problem for each pass. The goal of this research is to minimize total production time in multi-pass milling process. The notations are presented in Table 1.

2.1 Objective function

The total production time in the multi-pass milling process is the sum of the required times for machining the workpiece with several single passes as presented in Eq. (1).

$$T_{\mathrm{pr}}(f_{z_i}, V_i, a_i) = T_\omega + T_\varphi = \left\{T_{\mathrm{L}} + N_{\mathrm{p}} T_{\mathrm{a}} + \frac{T_{\mathrm{S}}}{N_{\mathrm{b}}}\right\} + \left\{\sum_{i=1}^{N_{\mathrm{p}}} \frac{\pi DL}{1000 Z V_i f_{z_i}} + \frac{\pi L V_i^{\left(\frac{1}{m}-1\right)} a_i^{\left(\frac{e_{\mathrm{v}}}{m}\right)} a_r^{\left(\frac{r_{\mathrm{v}}}{m}\right)} \lambda_{\mathrm{s}}^{\left(\frac{q_{\mathrm{v}}}{m}\right)} Z^{\left(\frac{n_{\mathrm{v}}}{m}-1\right)} f_{z_i}^{\left(\frac{u_{\mathrm{v}}}{m}-1\right)} T_{\mathrm{d}}}{1000 \times D^{\left(\frac{b_{\mathrm{v}}}{m}-1\right)} \left(B_{\mathrm{m}} B_{\mathrm{h}} B_{\mathrm{p}} B_{\mathrm{t}}\right)^{\left(\frac{1}{m}\right)} C_{\mathrm{v}}^{\left(\frac{1}{m}\right)}}\right\} \tag{1}$$

2.2 Constraints

Multi-pass milling process has some experimental and operational constraints which depend on the material and geometry of the workpiece. These operational constraints significantly reduce the feasible solution space of the problem and make the problem hard to be solved.

2.2.1 *Arbour stability*

The optimal values of the decision variables should satisfy the arbour stability constraint as presented in Eq. (2).

$$C_{zp} Z a_r a_i^{e_z} D^{b_z} f_{z_i}^{u_z} - \frac{0.1 k_{\mathrm{b}} d_{\mathrm{a}}^3}{(0.08 L_{\mathrm{a}}) + \left\{0.65 \sqrt{(0.25 L_{\mathrm{a}})^2 + \left(\frac{0.38461 k_{\mathrm{b}} D}{k_{\mathrm{t}}}\right)^2}\right\}} \leq 0 \tag{2}$$

where the first phrase of Eq. (2) shows the mean peripheral cutting force and the second phrase presents permissible force.

2.2.2 *Arbour deflection*

The arbour deflection constraint should be satisfied by the optimal values of the decision variables. Equation (3) presents the arbour deflection constraint.

$$C_{zp} Z a_r a_i^{e_z} D^{b_z} f_{z_i}^{u_z} - \left(4 E d_{\mathrm{a}}^4\right) \frac{e}{L_{\mathrm{a}}^3} \leq 0 \tag{3}$$

where the second phrase of Eq. (3) shows the permissible force with regard to the arbour deflection.

2.2.3 *Power constraint*

The following constraint ensures that the power required for the cutting operation is less or equal to the effective transmitted power to the cutting point which is given by Eq. (4).

Table 1 Parameters and decision variables

Parameter	Definition	Parameter	Definition
a_i	Depth of cut in each pass (mm)	N_{b}	Total number of components in the batch
a_r	Milling width (mm)	N_{p}	Total number of passes
$B_{\mathrm{m}}, B_{\mathrm{h}}, B_{\mathrm{p}}, B_{\mathrm{t}}$	Constant coefficients	P_{c}	Cutting power (W)
C_{zp}	Constant of the cutting force equation	P_{m}	Nominal motor power (W)
C_{v}	A constant	λ_{s}	Cutting inclination angle (°)
D	Arbour deflection values (mm)	V_i	Cutting speed (m/min)
d_{a}	Arbour diameter (mm)	T_{a}	Process set-up time (min/part)
E	Arbour material modulus of elasticity (kg/mm^2)	T_{d}	Tool changing time (min)
e	Values of arbour deflection (mm)	T_{L}	Loading and unloading time (min)
e_v, e_z, u_v, u_z	Empirical exponents	T_{pr}	Total production time (min)
L_{a}	Arbour length (mm)	T_{s}	Machine set-up time for a new batch (min)
f_{z_i}	Feed per tooth (mm/tooth)	Z	Number of teeth on the cutter
k_{b}	Bending stress (arbour material) (kg/mm^2)	η	Efficiency of the machine tool
k_{t}	Torsional stress of the arbour material (kg/mm^2)	$T_\omega = \frac{T_{\mathrm{S}}}{N_{\mathrm{b}}} + T_{\mathrm{L}} + N_{\mathrm{p}} T_{\mathrm{a}}$	
L	Length of cut (mm)	$T_\varphi = \sum_{i=1}^{N_{\mathrm{p}}} \left(\frac{\pi DL}{1000 Z V_i f_{z_i}} + \frac{\pi L V_i^{\left(\frac{1}{m}-1\right)} a_i^{\left(\frac{e_{\mathrm{v}}}{m}\right)} a_r^{\left(\frac{r_{\mathrm{v}}}{m}\right)} \lambda_{\mathrm{s}}^{\left(\frac{q_{\mathrm{v}}}{m}\right)} Z^{\left(\frac{n_{\mathrm{v}}}{m}-1\right)} f_{z_i}^{\left(\frac{u_{\mathrm{v}}}{m}-1\right)} T_{\mathrm{d}}}{1000 \times D^{\left(\frac{b_{\mathrm{v}}}{m}-1\right)} \left(B_{\mathrm{m}} B_{\mathrm{h}} B_{\mathrm{p}} B_{\mathrm{t}}\right)^{\left(\frac{1}{m}\right)} C_{\mathrm{v}}^{\left(\frac{1}{m}\right)}} \right)$	
$r_v, r_z, n_v, n_z, q_v, b_z, b_v, m$	Empirical exponents		

$$\frac{C_{zp} Z a_r a_i^{e_z} D^{b_z} f_{z_i}^{u_z} V_i}{6120} \le P_{\mathrm{m}} \eta \tag{4}$$

The last constraints present the lower and upper bound constraints of the decision variables.

$$f_{z_{\min}} \le f_{z_i} \le f_{z_{\max}} \tag{5}$$

$$V_{\min} \le V_{\mathrm{i}} \le V_{\max} \tag{6}$$

3 Solution methodology

The proposed mathematical model in the previous section is a constrained nonlinear programming (CNLP) model, which is hard to be solved using exact methods [33]. In last decade, many meta-heuristic algorithms have been proposed and commonly used to optimize complex NLP problems. Most of them are inspired from the nature such as water cycle algorithm [34–36], or animal's behaviour such as particle swarm optimization [37, 38], or other physical phenomena such as gravitational search algorithm [39]. In order to solve the above model, a new meta-heuristic algorithm named GWO is utilized [6]. As mentioned earlier, the GWO is widely used to solve engineering problems, while in this research GWO is used to optimize machining process parameters for the first time. GWO uses intelligent behaviour of the grey wolves in hunting and attacking prey in nature which helps the algorithm to perform well in solving complex problems. In this paper, a robust GWO is developed using Taguchi method.

3.1 Grey Wolf Optimizer

The GWO is inspired from grey wolves and their behaviour in nature. Grey wolves are one of the most famous predators in nature. They usually live in pack with five to twelve members. One of the most interesting behaviours of grey wolves is their social dominant hierarchy [6]. The hierarchy of the wolves and their responsibilities are presented in Fig. 1.

The alpha wolf is known as the leader of the pack, and his/her responsibility is to make decisions about hunting and other activities in the pack such as find a place for sleep. The beta wolf helps the alpha wolf in making decisions. The omega wolves are the lowest in ranking in the pack. Omegas are responsible to submit information to alpha, beta and delta wolves. All other wolves are called delta. The delta wolves should respect to alpha and beta, and they dominate omega wolves. There are three steps in hunting process of grey wolves as described in following [6].

- Tracking the prey.
- Encircling the prey.
- Attacking to the prey.

GWO is a mathematical representation of the hunting method of the grey wolves. GWO uses the hunting process of the grey wolves to solve complex optimization problems. For this purpose, the optimal solution of the problem is considered as prey. In each iteration of the GWO, first three best solutions are considered as alpha, beta and delta,

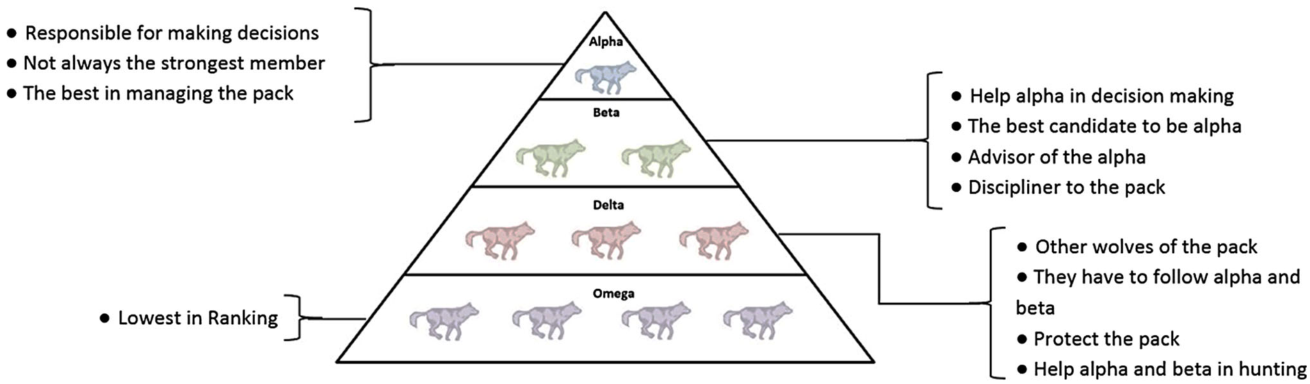

Fig. 1 Grey wolves' hierarchy and their responsibilities

respectively. The rest of the solutions are considered as omega. Hunting is leaded by alpha, beta and delta wolves, and the omegas follow these dominant wolves.

3.1.1 Encircling prey

The first step in hunting process is encircling prey. The mathematical formulation to mimic the encircling process is given in the following.

$$Di\vec{s} = \left|\vec{C}\cdot\vec{X}_{\mathrm{p}}(t) - \vec{X}(t)\right| \tag{7}$$

$$\bar{X}(t+1) = \vec{X}_{\mathrm{p}}(t) - \vec{A}\cdot\vec{D} \tag{8}$$

where $\vec{C}$ and $\vec{A}$ are coefficient vectors, t is the current iteration, while $\vec{X}_{\mathrm{p}}$ and $\vec{X}$ are the position of prey and a randomly chosen grey wolf, respectively [6]. The coefficients $\vec{C}$ and $\vec{A}$ are determined using following formulas.

$$\vec{A} = 2\vec{a}\cdot\vec{r}_1 - \vec{a} \tag{9}$$

$$\vec{C} = 2\cdot\vec{r}_2 \tag{10}$$

where the $\vec{a}$ vector decreases over the iterations of the GWO from two to zero and $\vec{r}_1$, $\vec{r}_2$ are randomly generated parameters between 0 and 1. Equations (9) and (10) allow the grey wolves to change their positions around the prey. This allows the GWO to search the n-dimensional solution space of the problem more efficiently.

3.1.2 Hunting

As mentioned above, the alpha, beta and delta wolves guide hunting process and rest of the wolves in the pack follow them. Mathematically speaking, the wolves update their position using the following equations.

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|,\ \ \vec{D}_\beta = \left|\vec{C}_2\cdot\vec{X}_\beta - \vec{X}\right|,\\ \vec{D}_\delta = \left|\vec{C}_3\cdot\vec{X}_\delta - \vec{X}\right| \tag{11}$$

$$\bar{X}_1 = \vec{X}_\alpha - \vec{A}_1\cdot\vec{D}_\alpha,\ \ \bar{X}_2 = \vec{X}_\beta - \vec{A}_2\cdot\vec{D}_\beta,\\ \bar{X}_3 = \vec{X}_\delta - \vec{A}_3\cdot\vec{D}_\delta \tag{12}$$

$$\bar{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{13}$$

3.1.3 Attacking prey

Decreasing the value of $\vec{a}$ over the iterations from 2 to 0 decreases the value of $\vec{A}$ over the iterations as well. As the value of $\vec{a}$ decreases, the grey wolves are more likely to attack towards the prey and get closer and closer to the prey over the last iterations of the GWO. In other words, the search radius of the grey wolves decreases over the course of iterations. This makes a proper trade-off in exploration and exploitation by focusing on exploration at first iterations and exploitation in last iterations.

Actually, the proposed basic GWO is to solve unconstrained problems and no version is developed to solve constrained problems. Therefore, we modified the GWO to obtain the optimal values of the decision variables of the constrained NLP model. The pseudo-code of the developed GWO is given in Table 2.

3.1.4 Constraint handling

During constrained optimization, both inequality and equality constraints should be satisfied. In this process, solutions are divided into two solutions, feasible and infeasible solutions [40]. As classified by Coello [41], five methods have been used to handle constraints during constrained optimization process: penalty functions, repair algorithms, hybrid methods, special operators and separation of objective functions and constraints. The penalty functions are the simplest technique to handle constraints which penalize the infeasible solutions. Consider a NLP problem as follows:

Table 2 Pseudo-code of the GWO

```
set the parameters of GWO α, A and C as obtained from Taguchi method
for i=1:number of search agents
    Create a randomly generated solution
    Calculate the objective function value of each search agent
    Use static penalty function to handle the constraints
end for
set the best solution as alpha
set the second best solution as beta
set the third best solution as delta
i=1;
While    i < max it
         for i=1:number of search agents
         update the position of each wolf
         end for
Iteratively decrease the value of α from 2 to 0
decrease A and C parameters
Calculate objective function value of each grey wolf
Update X⃗_α, X⃗_β and X⃗_δ
i=i+1;
end while
```

$$\begin{aligned} &\text{Min } z = f(\vec{x}) \\ &\text{s.t.} \\ &g_i(x) \leq 0 \\ &h_j(x) = 0 \end{aligned} \tag{14}$$

The penalty functions convert a constrained optimization model to an unconstrained one. The general formulation of the penalty functions can be presented as follows [41]:

$$\beta(\vec{x}) = f(\vec{x}) \pm \left[\sum_{i=1}^{n} r_i \times \max[0,\ g_i(x)]^{\beta} + \sum_{j=1}^{p} c_j \times |h_j(x)|\right] \tag{15}$$

where $\beta(\vec{x})$ shows the modified objective function and r_i, c_j are positive penalty factors [41]. There are different penalty function techniques which aim to handle the constraints in different manners such as dynamic penalty, static penalty, adaptive co-evolutionary penalty, annealing penalty and death penalty functions [42].

Most of the researchers in the literature used death penalty method to handle constraints in solving parameter optimization problems. Death penalty technique assigns a big positive number in minimization problems to objective function value of any infeasible solution. In other words, this technique does not use any information of infeasible solutions. This point makes the death penalty a weak technique to handle complex constraints in an optimization problem, that is why the algorithm needs more time to find the feasible solution space. Besides, the information of the infeasible solutions can help the algorithm to find the feasible solution space much faster. For this purpose, in this research static penalty technique is used to handle complex constraints of the proposed mathematical model. This method uses the information of the infeasible solutions (violation of each constraint) to reach to the feasible solution space. In the constraint handling approach proposed by Homaifar et al. [43], the user determines different levels of violation, and penalty constants for all constraints in a way that the penalty coefficient increases as the algorithm reaches higher violations [44–46].

$$\text{fitness}\ (\vec{x}) = \text{objective function}\ (\vec{x}) + \sum_{i=1}^{n} R_{k,i} \times \max[0, g_i(\vec{x})]^2 \tag{16}$$

where $R_{k,i}$ are user defined positive penalty coefficients. This methodology is utilized in this paper to handle complex operational constraints of the problem.

4 Tuning the parameters of the GWO

In all of the meta-heuristic algorithms, the values of input parameters of the algorithms play a prominent role in the performance of the algorithms. As mentioned above, GWO, as a meta-heuristic algorithm, has three main input parameters which effect the performance of the algorithm. The first one is the number of grey wolves (number of search agents). Consider a milling process where the aim is to minimize the total production time considering four passes, three rough millings at first and one finish milling at the end. Figure 2 presents the convergence curve of the GWO with different numbers of wolves in solving the problem.

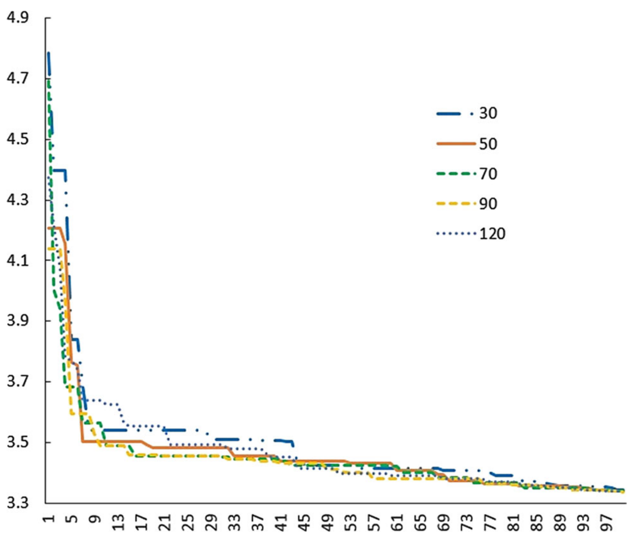

Fig. 2 Effect of change in the number of search agents on the performance of the GWO

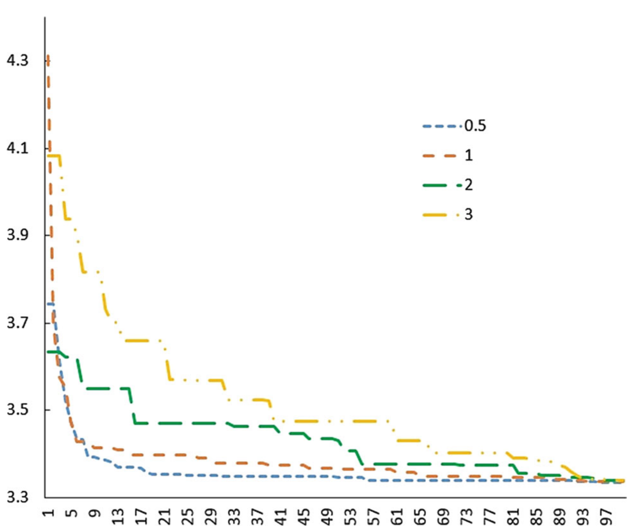

Fig. 3 Effect of change in the convergence constant on the performance of the GWO

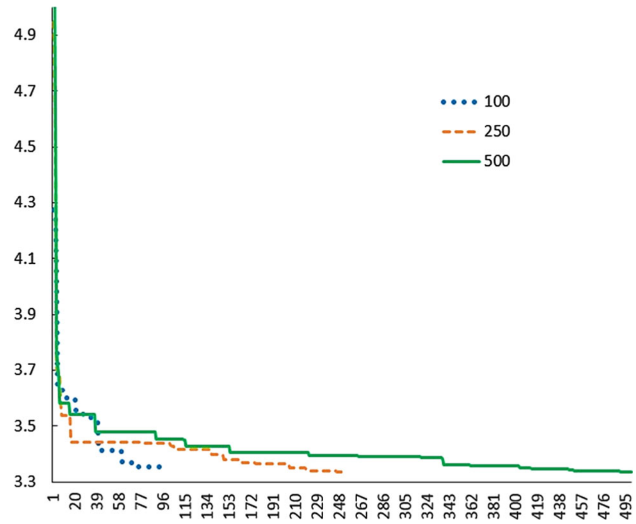

Fig. 4 Effect of change in the maximum number of iterations on the performance of the GWO

From Fig. 2, it is obvious that the efficiency of the GWO is significantly influenced by the number of grey wolves. The second parameter, which changes the performance of the GWO, is the convergence constant. The convergence constant is the most important parameter in the GWO, since it determines the search radius of the grey wolves and makes an appropriate trade-off between exploration and exploitation in GWO. To show the effect of change in this parameter, Fig. 3 is presented.

The third parameter (MaxIt) determines the maximum number of iterations. This parameter not only affects the performance of the GWO, but also effects the performance of all the meta-heuristic algorithms. Figure 4 depicts the effect of this parameter on the performance of the GWO.

As mentioned above, the performance of the GWO is directly influenced by the main parameters of the GWO. Therefore, determining the best values of the main parameters of the GWO results in better solutions. Different methodologies are proposed by the researchers to obtain the optimal level of the parameters of a meta-heuristic algorithm such as Taguchi method and response surface methodology. In this research, the Taguchi method approach is utilized to develop a robust GWO.

Factorial designs are known as design of experiments (DOE) proposed by Fisher and aim to determine the effect of several factors on a response [11, 12]. In this research, the objective function value is considered as the response and the factors are the main parameters of the GWO. Therefore, the effect of change in main parameters of the GWO on objective function value is investigated to determine the optimal values of the main parameters of the algorithm. In the Taguchi method, the main factors are

Table 3 Different levels of the parameters

Algorithm parameters	Parameters range	Level			
		Low (1)	Medium (2)	Medium (3)	High (4)
Initial value of α	0.5–3	0.5	1	2	3
Number of search agents (NSA)	30–90	30	50	70	90
Maximum number of iterations (MaxIt)	550–1150	550	750	950	1150

Table 4 Input parameters of the numerical example

r_v	0.1	C_v	35.4	e (finishing)	0.05 mm
a_r	50 mm	b_z	−0.86	e_z	0.86
T_s	10 min	T_a	0.1 (min/part)	e (roughing)	0.2 mm
N_b	100	T_L	1.5 min	η	0.7
B_h	1	B_t	0.8	u_v	0.4
B_p	0.8	e_v	0.3	B_m	1
L	160 mm	T_d	5 min	C_{zp}	68.2
Z	8	λ_s	30°	D	63 mm
P_m	5500 W	k_t	120 MPa = 12.23 (kg/mm^2)	b_v	0.45
E	200 Gpa = 20,387 (kg/mm^2)	k_b	140 MPa = 14.27 (kg/mm^2)	L_a	210 mm
d_a	27 mm	m	0.33	u_z	0.72
n_v	0.1	a_i	[0.5, 4] mm	q_v	0
f_{z_i}	[0.000875, 3.571] mm/tooth	V_i	[6.234, 395.84] m/min		

classified into two groups: controllable factors (S) and uncontrollable factors or noise factors (N). Taguchi method aims to determine control factors by minimizing the effect of noise factors to reduce the variation around response and to obtain a robust design [13–15].

4.1 Taguchi method

The main parameters of the GWO are tuned using Taguchi method to obtain better solutions and as fast as possible without trapping in local minima.

4.2 Grey Wolf Optimizer parameter tuning

Four levels of possible values are considered for each parameter of the GWO. Table 3 presents the levels of the parameters of the GWO.

The L^9 orthogonal array of the Taguchi method is used to find the optimal level of the parameters of the GWO. GWO is utilized to solve a milling process where the aim is to minimize the total production time considering four passes including three rough millings and one finish milling at the end to obtain the optimal values of the GWO parameters. For this purpose, a real case provided by Sonmez et al. [16] is utilized to develop RGWO. The values of required parameters of the case are presented in Table 4.

The problem is solved five times using GWO. Each repetition is shown by y_i. Table 5 presents the objective function values in five runs.

Figure 5 depicts the performance of the GWO in fourth, eighth, twelfth, thirteenth and fifteenth orders.

The first column of Fig. 5 presents the best solution over the course of iterations. The second column of Fig. 5 shows the average objective function value of grey wolves which are in the feasible solution space of problem. The third column of Fig. 5 presents the average penalty values of the wolves that are out of the feasible solution space of the problem. The last column of Fig. 5 presents the convergence constant value over the iterations.

To perform the Taguchi method, first the obtained objective functions values are normalized using linear norm. The linear normalization assigns one to the best solution (minimum objective function value) and positive values between 0 and 1 to other solutions based on their deviation from the best solution. Therefore, the solution with larger normalized value is preferred. Since the aim is to find solutions with larger normalized value, highest rate in S/N ratio is suitable, see [11, 12]. The S/N ratio for this type objective function is calculated as follows:

$$S/N = -10\log\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{y_i^2}\right) \quad (17)$$

where n is the replications which in our example is five and y_i is the normalized objective function value of the

Table 5 Experimental results of the GWO

Run order	α	NSA	It	y_1	y_2	y_3	y_4	y_5
1	1	1	1	3.3344	3.3359	3.3352	3.3351	3.3348
2	1	2	2	3.3343	3.3347	3.3344	3.3348	3.3344
3	1	3	3	3.3344	3.3344	3.3344	3.3341	3.3344
4	1	4	4	3.3342	3.3341	3.3341	3.3344	3.3343
5	2	1	2	3.3353	3.3352	3.335	3.3345	3.3345
6	2	2	1	3.3348	3.3359	3.3345	3.3348	3.3351
7	2	3	4	3.3344	3.3344	3.3345	3.3348	3.3342
8	2	4	3	3.3348	3.3354	3.3345	3.3342	3.3342
9	3	1	3	3.3347	3.3357	3.3347	3.3348	3.3369
10	3	2	4	3.3348	3.3362	3.3353	3.3352	3.3349
11	3	3	1	3.3349	3.3352	3.3353	3.3351	3.3353
12	3	4	2	3.3353	3.3355	3.3353	3.3354	3.3353
13	4	1	4	3.3356	3.3351	3.3351	3.3361	3.3349
14	4	2	3	3.3349	3.3358	3.335	3.336	3.3352
15	4	3	2	3.3352	3.3349	3.3351	3.3356	3.3352
16	4	4	1	3.335	3.3358	3.3361	3.3366	3.3348

different combinations of the parameters levels. Table 6 presents the results.

Figure 6 presents the results of the Taguchi method implementation. As in Fig. 6, the highest value of the *S/N* ratio is the best level of the parameter.

From the results, the optimal values of the parameters of the GWO algorithm can be determined. The optimal value of the α parameter (convergence constant) is 0.5. The optimal number of grey wolves in the initial population is 70, and the optimal number of iterations to solve the problem is 1150.

5 Results and discussion

The real case provided by Sonmez et al. [16] is utilized to evaluate the efficiency of the developed RGWO in optimizing the multi-pass milling process. Three main strategies are adopted for given values of a_i as reported in the literature. The first two strategies include three rough and one finish passes. The third strategy contains one rough and one finish passes. The value of input parameters of the case is as presented earlier in Table 4. Table 7 presents the three cutting strategies and the value of the parameter a in each pass.

To optimize the parameters of the multi-pass milling process using the RGWO, the initial pack of wolves is created as shown in Table 8 [44–46], where *rand* is a random number between 0 and 1.

Tables 9, 10 and 11 present the computational results of optimizing the multi-pass milling process using the developed RGWO within different strategies. In addition, the results of the RGWO are compared with other meta-heuristic algorithms in the literature such as particle swarm optimization, simulated annealing and artificial bee colony algorithm. To validate the results of the RGWO, the performance of the RGWO is compared to two novel meta-heuristic algorithms named dragonfly algorithm and multi-verse optimizer within the same number of function evaluations, as well [1, 47].

From Table 9, the proposed RGWO obtains better solutions which has significantly less production time comparing to other solution methodologies in the literature. In addition, the two novel meta-heuristic algorithms obtained two solutions which are significantly worst than the solution of the RGWO. A schematic view of the comparison of RGWO and previous researches is presented in Fig. 7.

The computational results of the second strategy are given in Table 10. The second strategy contains three rough and one finish passes.

From Table 10, it is clear that the RGWO obtained better solutions with less total process time comparing to other methods in the literature. In addition, the solution obtained by the RGWO satisfies all the constraints of the problem. Figure 8 depicts the comparison of the results of the RGWO and other solution methods in the literature in a graphical manner.

Table 11 presents the computation results of the third strategy. In contradiction with the first two strategies, the third strategy includes one rough and one finish passes.

One of the important points in the results of the third strategy is that in most of the previous researches, the developed meta-heuristic algorithms could not obtain a solution which satisfy all constraints; in other words, their solutions are not feasible. Conversely, the developed

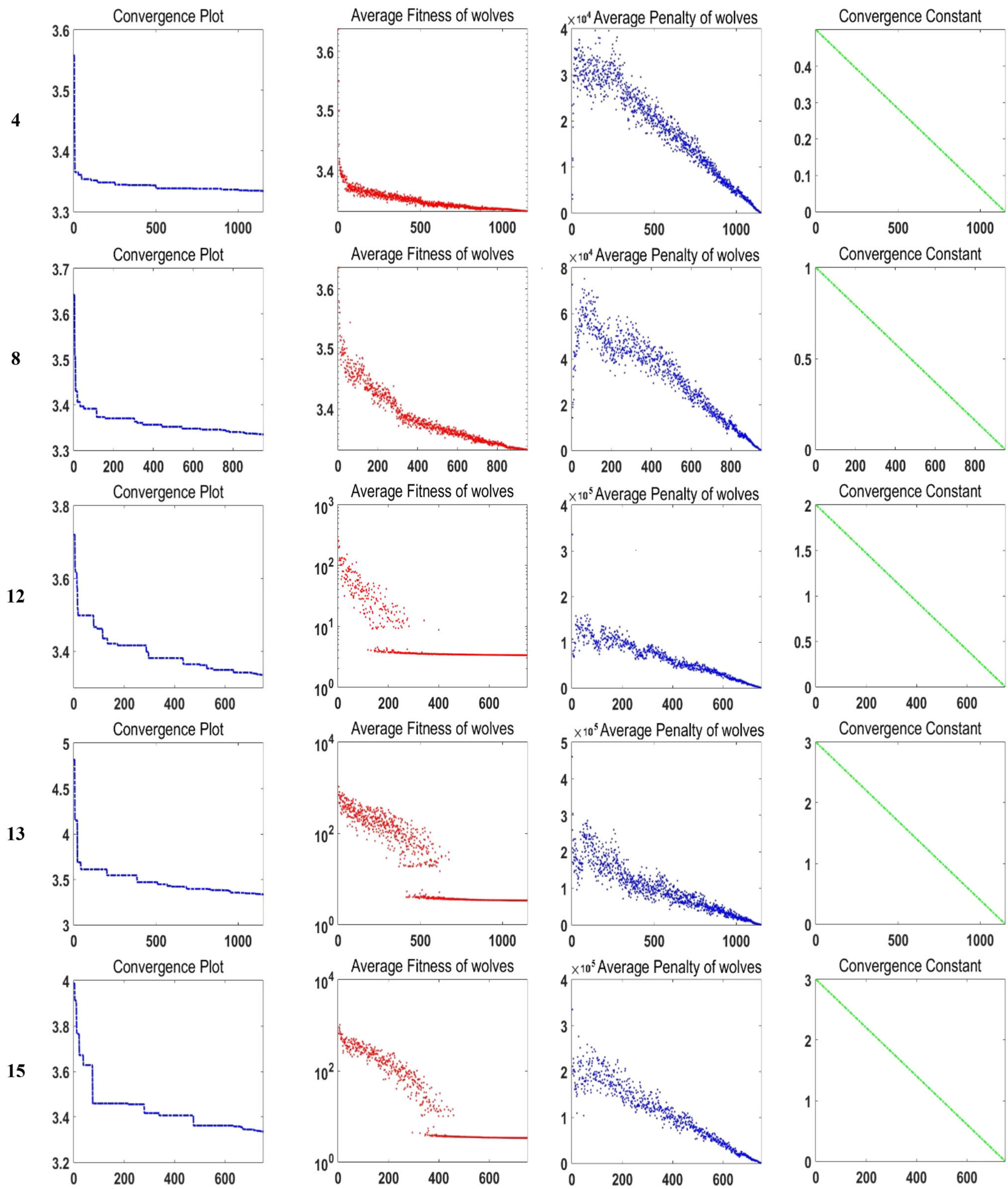

Fig. 5 Performance of the GWO in fourth, eighth, twelfth, thirteenth and fifteenth orders

RGWO in this research determines better feasible solutions which satisfy all constraint. This shows the efficiency and applicability of the proposed solution methodology in optimizing multi-pass milling process. In the GWO, the omega wolves follow the three leading wolves (alpha, beta and delta wolves) and update their position in each iteration

Table 6 Normalized values of the experimental results of GWO

Run order	α	NSA	It	y_1	y_2	y_3	y_4	y_5	Mean	SNRA
1	1	1	1	0.99996	0.99946	0.99968	0.99972	0.99983	0.99973	−0.00234
2	1	2	2	0.99997	0.99984	0.99992	0.99981	0.99993	0.9999	−0.0009
3	1	3	3	0.99994	0.99993	0.99991	1	0.99994	0.99995	−0.00047
4	1	4	4	1	1	1	0.99994	0.99998	0.99998	−0.00014
5	2	1	2	0.99968	0.99967	0.99974	0.99989	0.99991	0.99978	−0.00191
6	2	2	1	0.99983	0.99947	0.99989	0.99981	0.99972	0.99974	−0.00222
7	2	3	4	0.99994	0.99992	0.99987	0.99979	1	0.9999	−0.00083
8	2	4	3	0.99983	0.99961	0.99987	0.99997	0.99999	0.99986	−0.00125
9	3	1	3	0.99986	0.99954	0.99982	0.99979	0.99919	0.99964	−0.00309
10	3	2	4	0.99983	0.99938	0.99963	0.99967	0.99978	0.99966	−0.00295
11	3	3	1	0.99978	0.9996	0.99965	0.99972	0.99967	0.9997	−0.00261
12	3	4	2	0.99968	0.99958	0.99965	0.99962	0.99968	0.99964	−0.0031
13	4	1	4	0.99959	0.99971	0.99969	0.99942	0.99979	0.99964	−0.00311
14	4	2	3	0.99978	0.99951	0.99972	0.99944	0.99969	0.99963	−0.0032
15	4	3	2	0.9997	0.99977	0.99971	0.99957	0.99971	0.99969	−0.00267
16	4	4	1	0.99976	0.9995	0.9994	0.99927	0.99981	0.99955	−0.00392

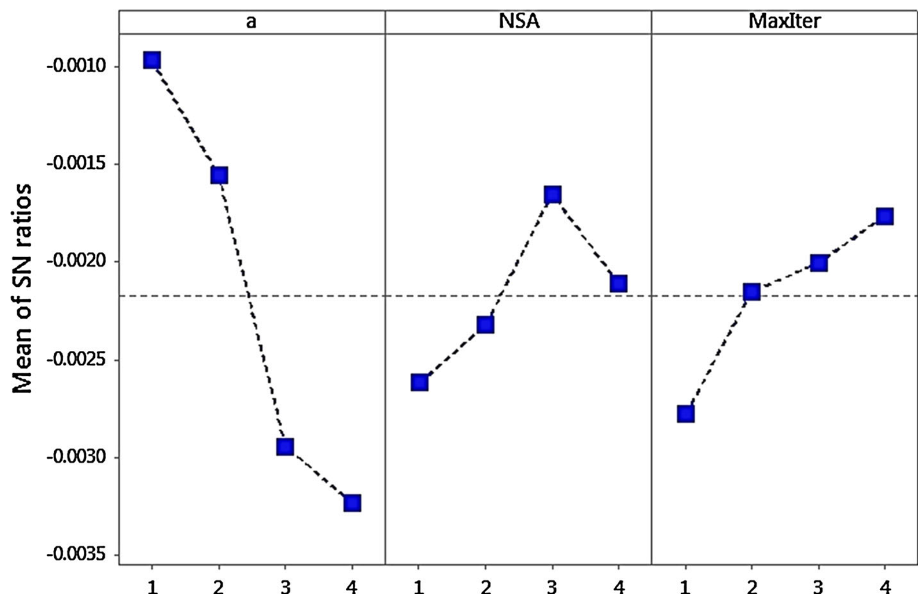

Fig. 6 Results of the Taguchi method

Table 7 Three cutting strategies

Strategy	# of passes	a_{rough1}	a_{rough2}	a_{rough3}	$a_{finish1}$
1	4	1.5	1.5	1.5	0.5
2	4	2	1	1	1
3	2	3	–	–	2

Table 8 Generating the initial population in GWO

```
Enter the lower bound of the decision variables.
Enter the upper bound of the decision variables.
 for i= 1:number of wolves
        for j= 1:number of decision variables
          Ini pop(i,j)=lb(j)+(ub(j)-lb(j))*rand
        end for
end for
```

with respect to the position of these three leading wolves. In other words, they share the best solution in the algorithm. This approach helps the search agents (omega wolves) to find the feasible solution space faster and the best solution as well [44–46]. Figure 9 shows the comparison of the results of the RGWO in third strategy and other solution methodologies in the literature, where points in the red circle are infeasible solutions obtained in the previous researches.

Table 9 Computational results of the first strategy

Researchers	Method	f_z (mm/tooth)	V (m/min)	T_ω(min)	T_φ(min)	T_{pr}(min)	Constraint violation		
							1	2	3
Yang et al. [48]	ICA	(0.342, 0.342, 0.342, 0.436)	(46.554, 46.554, 46.554, 64.164)	2.0	1.239	3.239	No	No	No
Gao et al. [21]	PSO	(0.341, 0.341, 0.341, 0.435)	(54.027, 51.713, 49.041, 65.054)	2.0	1.234	3.234	No	No	No
Gao et al. [21]	CPSO	(0.341, 0.341, 0.341, 0.435)	(50.864, 50.864, 50.864, 64.191)	2.0	1.233	3.2330	No	Yes	No
Rao and Pawar [22]	SA	(0.336, 0.336, 0.336, 0.429)	(44.633, 44.633, 44.633, 57.23)	2.0	1.263	3.263	No	No	No
Rao and Pawar [22]	PSO	(0.34, 0.34, 0.34, 0.434)	(46.61, 46.61, 46.61, 63.58)	2.0	1.240	3.240	No	No	No
Rao and Pawar [22]	ABC	(0.337, 0.337, 0.337, 0.432)	(46.982, 46.982, 46.982, 64.41)	2.0	1.240	3.240	No	No	No
Pawar and Rao [23]	TLBO	(0.341, 0.341, 0.341, 0.434)	(46.641, 46.641, 46.641, 66.8576)	2.0	1.237	3.237	No	No	No
Huang et al. [49]	CS	(0.3413, 0.3413, 0.3414, 0.4349)	(50.471, 50.7762, 50.8559, 64.6164)	2.0	1.233	3.233	No	No	No
Huang et al. [49]	TLCS	(0.3414, 0.3414, 0.3414, 0.4349)	(50.8643, 50.8643, 50.8643, 64.1905)	2.0	1.2323	3.2323	Yes	No	No
Mellal and Williams [25]	COA	(0.3413, 0.3413, 0.3413, 0.4349)	(51.0425, 50.8967, 50.6023, 63.4896)	2.0	1.2325	3.2325	No	No	No
This research	DA	(0.33047, 0.34078, 0.3413, 0.42189)	(62.7755, 60.9232, 67.0884, 65.5291)	2.0	1.29905	3.29905	No	No	No
This research	MVO	(0.34135, 0.34126, 0.34128, 0.4348)	(50.376, 49.861, 50.927, 65.721)	2.0	1.23220	3.23220	No	No	No
This research	RGWO	(0.34134, 0.34133, 0.34134, 0.43478)	(50.2151, 51.4133, 51.0923, 63.5332)	2.0	1.2320	3.2319	No	No	No

Table 10 Computational results of the second strategy

Researchers	Method	f_z (mm/tooth)	V (m/min)	T_ω(min)	T_φ(min)	T_{pr}(min)	Constraint violation		
							1	2	3
Yang et al. [48]	ICA	(0.242, 0.555, 0.555, 0.190)	(46.554, 46.554, 46.554, 72.547)	2.0	1.342	3.342	No	No	No
Rao and Pawar [22]	PSO	(0.240, 0.553, 0.553, 0.19)	(46.53, 46.42, 46.42, 70.84)	2.0	1.342	3.342	No	No	No
Rao and Pawar [22]	ABC	(0.231, 0.552, 0.552, 0.189)	(48.117, 47.519, 47.519, 74.090)	2.0	1.355	3.355	No	No	No
Gao et al. [21]	PSO	(0.242, 0.554, 0.553, 0.190)	(52.282, 48.321, 54.230, 66.788)	2.0	1.343	3.343	No	No	No
Gao et al. [21]	CPSO	(0.242, 0.554, 0.554, 0.190)	(53.533, 47.327, 47.327, 72.608)	2.0	1.335	3.335	No	No	No
Mellal and Williams [25]	COA	(0.2420, 0.5540, 0.5540, 0.1900)	(53.5553, 47.2775, 47.1194, 72.6187)	2.0	1.3348	3.3348	No	No	No
This research	DA	(0.23262, 0.5393, 0.54743, 0.1900)	(100, 51.5592, 48.1683, 74.5482)	2.0	1.58560	3.58560	No	No	No
This research	MVO	(0.24205, 0.5538, 0.5540, 0.1899)	(51.5340, 49.47629, 46.0717, 72.6108)	2.0	1.33543	3.33543	No	No	No
This research	RGWO	(0.24208, 0.55401, 0.55403, 0.19002)	(54.0959, 47.3047, 47.4863, 72.6027)	2.0	1.3341	3.3341	No	No	No

Table 11 Computational results of the third strategy

Researchers	Method	f_z (mm/tooth)	V (m/min)	T_ω (min)	T_φ (min)	T_{pr} (min)	Constraint violation		
							1	2	3
Sonmez et al. [16]	GP	(0.338, 0.570)	(26.40, 25.16)	1.801	0.813	2.614	Yes	Yes	Yes
Wang et al. [17]	GA	(0.366, 0.5667)	(24.69, 25.16)	1.7998	0.8102	2.61	Yes	Yes	Yes
Wang et al. [17]	PGSA	(0.3693, 0.5886)	(24.25, 24.58)	1.80	0.80	2.6	Yes	Yes	No
Onwubolu [18]	Tribes	(0.587, 0.902)	(36.27, 30.16)	1.7005	0.512	2.2125	Yes	Yes	Yes
Mellal and Williams [25]	COA	(0.14915, 0.08303)	(57.63565, 82.10743)	1.80	1.55480	3.35480	No	No	No
This research	DA	(0.14915, 0.08303)	(63.1735, 77.4000)	1.80	1.5633	3.36328	No	No	No
This research	MVO	(0.149085, 0.08302)	(56.1955, 81.8732)	1.80	1.5546	3.3546	No	No	No
This research	RGWO	(0.149158, 0.08303)	(57.3707, 82.40126)	1.80	1.5540	3.35395	No	No	No

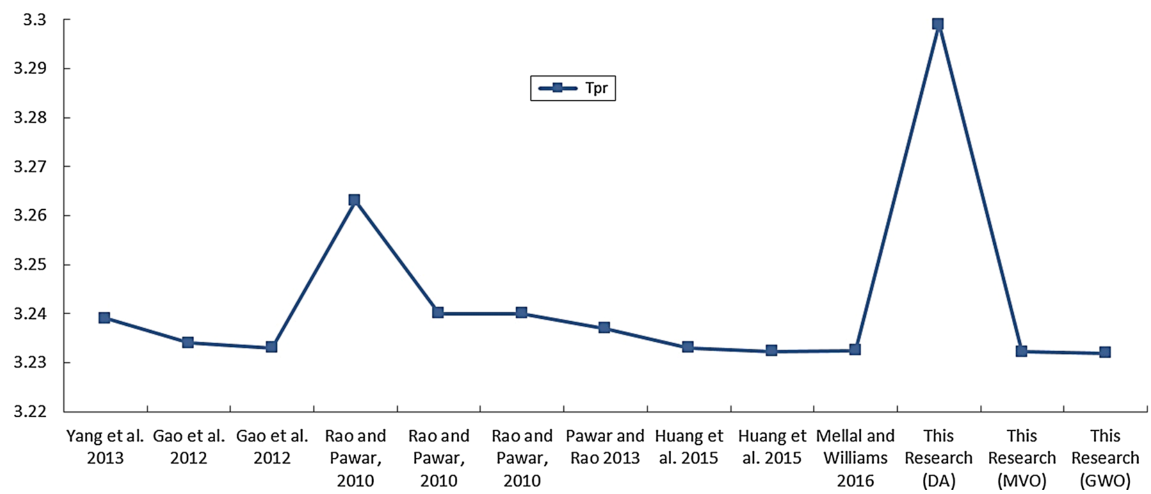

Fig. 7 Comparison of RGWO and other researches in the first strategy

As is shown in the above comparison, the performance of the RGWO is significantly better than other algorithms in the literature as well as other novel meta-heuristic algorithms, but why? To perform well, each meta-heuristic algorithm has to explore and exploit the solution space of the problem, efficiently. GWO performs very well in making an appropriate trade-off between exploitation and exploration using the convergence constant. As mentioned earlier, the value of the convergence constant decreases over the course of iterations of the GWO. The convergence constant is similar to search radius of the grey wolves. Therefore, in the first iterations, when the value of the convergence constant is high, the GWO focuses on exploring of the solution space and finding better solutions. Then, the GWO starts to decrease the convergence constant, and this process decreases the search radius of the grey wolves. Thus, GWO aims to exploit the solution space more efficiently as the iterations go on. By making this appropriate trade-off, GWO is able to explore and exploit the solution space effectively and this helps the GWO to obtain better solutions. The second reason is that the GWO can reach the feasible solution space of the problem very fast. In the first iterations, most of the randomly generated solutions are infeasible, and the static penalty approach calculates the violation level of each infeasible solution and chooses the best infeasible solution as alpha. Therefore, the omega wolves move towards a solution which is the best infeasible solution with least violation level. Thus, all the wolves in the pack move very fast towards the feasible solution space, and when alpha becomes feasible, all other wolves start to follow him instantly towards the feasible solution space.

As is known, the values of the main parameters of the meta-heuristic algorithms significantly affect their performance. In this research, the main parameters of the GWO were tuned using a robust design approach within five repetitions to obtain the best possible solution. One of the most critical parameters of the GWO is convergence constant. The optimal value of this parameter is also obtained using Taguchi method. As mentioned earlier, this parameter makes the appropriate trade-off between exploration and exploitation abilities of the GWO. Thus, by obtaining the optimal value of this parameter, the GWO reached its maximum efficiency in solving the

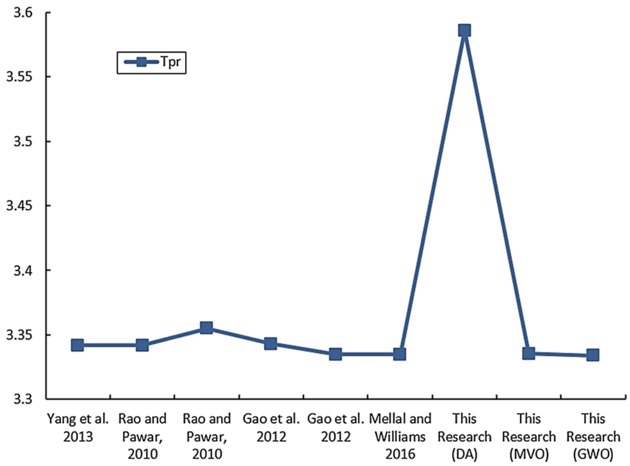

Fig. 8 Comparison of RGWO and other researches in second strategy

problem, since it makes the best possible trade-off between exploration and exploitation. All the above-mentioned three reasons make the GWO a powerful algorithm in solving the problem.

As is clear, the other algorithms in the literature lack from an efficient exploration and exploitation abilities. The best values of the main parameters of those algorithms are obtained by trial and error which does not guarantee the optimality of the values of the main parameters. In addition, they may lack from an efficient constraint handling procedure.

Note that in all the simulations, the number of function evaluations (NFE) of the RGWO was 80,500, while the computation time was <10 s. Although the NFE of the RGWO is high, it can solve the problem accurately and very fast.

6 Conclusion

This paper aims to minimize total production time in a multi-pass milling process. Since the mathematical model of the multi-pass milling process is nonlinear programming, traditional techniques cannot solve the problem efficiently. Therefore, in this research a novel RGWO developed to solve the optimization problem. A robust design methodology named Taguchi method was implemented to tune the parameters of the GWO. Thus, in contradiction to previous researches, costly experiments are not required to calibrate the values of input parameters of the algorithm. The optimal values of the main parameters of the GWO guarantee the best appropriate trade-off between exploration and exploitation abilities of the algorithm. Also, an efficient constraint handling technique was utilized to handle complex operational constraints of the problem. By solving an experimental example, the effectiveness and applicability of the proposed RGWO was investigated. The performance of the RGWO was compared to the other solution methods in the literature as well as two novel meta-heuristic algorithms named dragonfly

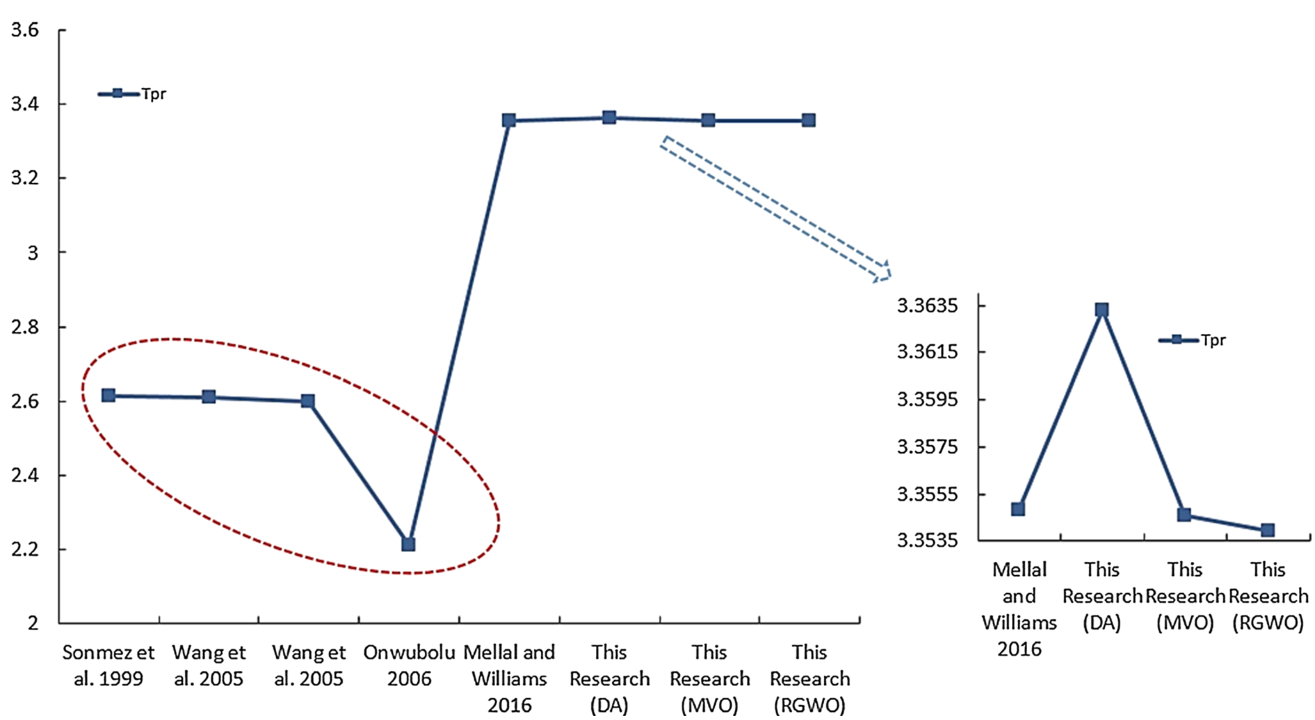

Fig. 9 Comparison of RGWO and other researches in third strategy

algorithm and multi-verse optimizer. The results indicated that the RGWO outperforms the other solution methods in the literature as well as novel meta-heuristic algorithms in minimizing total production time. Using the robust design methodology and efficient constraint handling approach, the RGWO was able to obtain the best feasible solution for different cutting strategies.

For future research, it is very interesting to consider some of the main parameters of the mathematical model under uncertainty and develop more realistic models. Considering different other operational constraints of the multi-pass milling process as well as multiple objectives will absolutely help the decision-maker to choose the best possible solution. Developing other efficient solution methodologies to solve the problem would be worthwhile. Using other constraint handling techniques to solve the problem is another extension of this research.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. doi:10.1007/s00521-015-1870-7
2. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097. doi:10.1007/s00521-014-1597-x
3. Salimi H (2015) Stochastic fractal search: a powerful metaheuristic algorithm. Knowl Based Syst 75:1–18. doi:10.1016/j.knosys.2014.07.025
4. Mirjalili S, Mirjalili SM, Yang XS (2014) Binary bat algorithm. Neural Comput Appl 25(3–4):663–681. doi:10.1007/s00521-013-1525-5
5. Mirjalili S, Wang GG, Coelho LDS (2014) Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. Neural Comput Appl 25(6):1423–1435. doi:10.1007/s00521-014-1629-6
6. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey Wolf Optimizer. Adv Eng Soft 69:46–61. doi:10.1016/j.advengsoft.2013.12.007
7. Jayakumar N, Subramanian S, Ganesan S, Elanchezhian EB (2016) Grey wolf optimization for combined heat and power dispatch with cogeneration systems. Int J Elect Power Energy Sys 74:252–264. doi:10.1016/j.ijepes.2015.07.031
8. Song X, Tang L, Zhao S, Zhang X, Li L, Huang J, Cai W (2015) Grey Wolf Optimizer for parameter estimation in surface waves. Soil Dyn Earthq Eng 75:147–157. doi:10.1016/j.soildyn.2015.04.004
9. Pradhan M, Roy PK, Pal T (2016) Grey wolf optimization applied to economic load dispatch problems. Int J Elect Power Energy Sys 83:325–334. doi:10.1016/j.ijepes.2016.04.034
10. Medjahed SA, Saadi TA, Benyettou A, Ouali M (2016) Gray Wolf Optimizer for hyperspectral band selection. Appl Soft Comput 40:178–186. doi:10.1016/j.asoc.2015.09.045
11. Roy R (1990) A primer on the Taguchi method. Society of Manufacturing Engineers, New York
12. Taguchi G, Chowdhury S, Wu Y (2005) Taguchi's quality engineering handbook. Wiley, Hoboken
13. Najafi AA, Niaki STA, Shahsavar M (2009) A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations. Comput Oper Res 36(11):2994–3001. doi:10.1016/j.cor.2009.01.016
14. Sadeghi J, Sadeghi A, Saidi-Mehrabad M (2011) A parameter-tuned genetic algorithm for vendor managed inventory model for a case single-vendor single-retailer with multi-product and multi-constraint. J Optim Ind Eng 4(9):57–67
15. Sadeghi J, Sadeghi S, Niaki STA (2014) A hybrid vendor managed inventory and redundancy allocation optimization problem in supply chain management: an NSGA-II with tuned parameters. Comput Oper Res 41:53–64. doi:10.1016/j.cor.2013.07.024
16. Sonmez AI, Baykasoglu A, Dereli T, Filiz IH (1999) Dynamic optimization of multi-pass milling operations via geometric programming. Int J Mach Tools Manuf 39:297–320. doi:10.1016/S0890-6955(98)00027-3
17. Wang ZG, Rahman M, Wong YS, Sun J (2005) Optimization of multi-pass milling using parallel genetic algorithm and parallel genetic simulated annealing. Int J Mach Tools Manuf 45:1726–1734. doi:10.1016/j.ijmachtools.2005.03.009
18. Onwubolu GC (2006) Performance-based optimization of multi-pass face milling operations using Tribes. Int J Mach Tools Manuf 46:717–727. doi:10.1016/j.ijmachtools.2005.07.041
19. Baskar N, Asokan P, Saravanan R, Prabhaharan G (2006) Selection of optimal machining parameters for multi-tool milling operations using a memetic algorithm. J Mater Process Technol 174:239–249. doi:10.1016/j.jmatprotec.2005.09.032
20. Yıldız AR (2009) A novel hybrid immune algorithm for global optimization in design and manufacturing. Robot Comput Integr Manuf 25:261–270. doi:10.1016/j.rcim.2007.08.002
21. Gao L, Huang J, Li X (2012) An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process. Appl Soft Comput 12:3490–3499. doi:10.1016/j.asoc.2012.06.007
22. Rao RV, Pawar PJ (2010) Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms. Appl Soft Comput 10:445–456. doi:10.1016/j.asoc.2009.08.007
23. Pawar PJ, Rao RV (2013) Parameter optimization of machining processes using teaching–learning-based optimization algorithm. Int J Adv Manuf Technol 67:995–1006. doi:10.1007/s00170-012-4524-2
24. Yang WA, Guo Y, Liao WH (2011) Optimization of multi-pass face milling using a fuzzy particle swarm optimization algorithm. Int J Adv Manuf Technol 54:45–57. doi:10.1007/s00170-010-2927-5
25. Mellal MA, Williams EJ (2016) Total production time minimization of a multi-pass milling process via cuckoo optimization algorithm. Int J Adv Manuf Technol 1:1–8. doi:10.1007/s00170-016-8498-3
26. Mellal MA, Williams EJ (2014) Parameter optimization of advanced machining processes using cuckoo optimization algorithm and hoopoe heuristic. J Intell Manuf. doi:10.1007/s10845-014-0925-4
27. Mellal MA, Williams EJ (2015) Cuckoo optimization algorithm for unit production cost in multi-pass turning operations. Int J Adv Manuf Technol 76(1–4):647–656. doi:10.1007/s00170-014-6309-2
28. Pal SK, Chakraborty D (2005) Surface roughness prediction in turning using artificial neural network. Neural Comput Appl 14(4):319–324. doi:10.1007/s00521-005-0468-x
29. Venkatesan D, Kannan K, Saravanan R (2009) A genetic algorithm-based artificial neural network model for the optimization of machining processes. Neural Comput Appl 18(2):135–140. doi:10.1007/s00521-007-0166-y
30. Prabhu S, Uma M, Vinayagam BK (2015) Surface roughness prediction using Taguchi-fuzzy logic-neural network analysis for

CNT nanofluids based grinding process. Neural Comput Appl 26(1):41–55. doi:10.1007/s00521-014-1696-8
31. Arriandiaga A, Portillo E, Sánchez JA, Cabanes I, Pombo I (2016) A new approach for dynamic modelling of energy consumption in the grinding process using recurrent neural networks. Neural Comput Appl 27(6):1577–1592. doi:10.1007/s00521-015-1957-1
32. Arriandiaga A, Portillo E, Sánchez JA, Cabanes I, Zubizarreta A (2016) Recurrent ANN-based modelling of the dynamic evolution of the surface roughness in grinding. Neural Comput Appl 1–15. doi:10.1007/s00521-016-2568-1
33. Gen M (1997) Genetic algorithm and engineering design. Wiley, New York
34. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166. doi:10.1016/j.compstruc.2012.07.010
35. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle algorithm for solving multi-objective optimization problems. Soft Comput 19(9):2587–2603. doi:10.1007/s00500-014-1424-4
36. Sadollah A, Eskandar H, Kim JH (2015) Water cycle algorithm for solving constrained multi-objective optimization problems. Appl Soft Comput 27:279–298. doi:10.1016/j.asoc.2014.10.042
37. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. pp 1942–1948. doi:10.1109/ICNN.1995.488968
38. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: The 1998 IEEE international conference on evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence. pp 69–73. doi:10.1109/ICEC.1998.699146
39. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. doi:10.1016/j.ins.2009.03.004
40. Khalilpourazari S, Pasandideh SHR (2016) Multi-item EOQ model with nonlinear unit holding cost and partial backordering: moth-flame optimization algorithm. J Ind Prod Eng 1:1–10. doi:10.1080/21681015.2016.1192068
41. Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191:1245–1287. doi:10.1016/S0045-7825(01)00323-1
42. Khalilpourazari S, Pasandide SHR, Niaki STA (2016) Optimization of multi-product economic production quantity model with partial backordering and physical constraints: SQP, SFS, SA, and WCA. Appl Soft Comput. doi:10.1016/j.asoc.2016.08.054
43. Homaifar A, Lai SHY, Qi X (1994) Constrained optimization via genetic algorithms. Simulation 62:242–254. doi:10.1177/003754979406200405
44. Khalilpourazari S, Khalilpourazary S (2016) A lexicographic weighted Tchebycheff approach for multi-constrained multi-objective optimization of the surface grinding process. Eng Optim. doi:10.1080/0305215X.2016.1214437
45. Khalilpourazari S, Pasandideh SHR (2016) Bi-objective optimization of multi-product EPQ model with backorders, rework process and random defective rate. In: 12th international conference on industrial engineering. pp 36–40. doi:10.1109/INDUSENG.2016.7519346
46. Khalilpourazari S, Mohammadi M (2016) Optimization of closed-loop Supply chain network design: a water cycle algorithm approach. In: 12th international conference on industrial engineering. pp 41–45. doi:10.1109/INDUSENG.2016.7519347
47. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073. doi:10.1007/s00521-015-1920-1
48. Yang Y, Li X, Gao L (2013) Parameters optimization of a multipass milling process based on imperialist competitive algorithm. In: IEEE 17th international conference on computer supported cooperative work in design (CSCWD), pp 406–410. doi:10.1109/CSCWD.2013.6580997
49. Huang J, Gao L, Li X (2015) An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes. Appl Soft Comput 36:349–356. doi:10.1016/j.asoc.2015.07.031