FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 11, N° 1, 2013, pp. 29 - 44

COMPARISON OF META-HEURISTIC ALGORITHMS FOR SOLVING MACHINING OPTIMIZATION PROBLEMS

UDC 519.863; 621.7.01

Miloš Madić, Danijel Marković, Miroslav Radovanović

Universityof Niš, Faculty of Mechanical Engineering, Serbia

Abstract. *Since optimization of the machining parameters not only increases machining efficiency and economics, but it also enhances the end product quality, this topic is still the subject of many studies. The selection of the optimal machining parameters is often performed in a two-stage approach, i.e. mathematical modeling of machining performance and optimization using an optimization method. Among the traditional optimization methods, in recent years, the modern meta-heuristic algorithms are being increasingly applied to solving machining optimization problems. Their ability to deal with complex, multi-dimensional and ill-behaved optimization problems has made them choice number one by most researchers and practitioners. In the present study, an attempt is made to compare the optimization results of different meta-heuristic algorithms applied to solving machining optimization problems. Four meta-heuristic algorithms are taken into consideration, namely, real coded genetic algorithm (RCGA), simulated annealing (SA), improved harmony search algorithm (IHSA) and cuckoo search algorithm (CSA). These meta-heuristic algorithms are applied to searching for optimal combinations of different machining parameters for five case studies taken from the literature. The optimization results obtained by applying RCGA, SA, IHSA and CSA for parametric optimization of these machining processes are compared with those derived by the past researchers.*

Key Words: *Machining, Optimization, Meta-heuristic Algorithms*

1. INTRODUCTION

In today's manufacturing world, it is a vital task to define optimal machining parameters for achieving machining cost and efficiency [1]. Modeling of machining processes aimed at better understanding, optimization and process control is very important in manufacturing practice. This is usually achieved by integrating empirical models based on the regression

Received January 10, 2013
[*] **Acknowledgments**. The paper is a part of the research done within the project TR35034. The authors would like to thank to the Ministry of Education and Science, Republic of Serbia.
Corresponding author: Miloš Madić
University of Niš, Faculty of Mechanical Engineering, Str. Aleksandra Medvedeva 14, 18000 Niš, Serbia
E-mail: madic@masfak.ni.ac.rs

analysis or artificial neural networks with an optimization method. The application of the Taguchi method without formulating any kind of model is also an attractive alternative, particularly in the case of multi-objective optimization problems [2].

Identification of the optimal machining parameters is very important for reduction of machining costs, product quality improvement and increased productivity and profit. Therefore, the machining processes optimization is one of the most investigated issues. Thus, many researchers have tried various conventional methods for solving machining optimization problems including design of experiments, graphical methods, analytic methods and mathematical programming methods such as linear and nonlinear programming, dynamic programming, geometric programming, goal programming, integer programming, stochastic programming, etc. [3]. Despite the fact that the machining optimization problems have been extensively investigated, the complexity of machining economics problems has led to the requirement for increasingly effective optimization algorithms [4].

The convergence speed of the meta-heuristic algorithms to the global (or nearly global) optimal results is better than that of traditional techniques. Therefore, the meta-heuristic algorithms have been increasingly used to further improve the solution of machining optimization problems with complex nature in many applications [1]. It is reported that the meta-heuristic algorithms have been applied in machining because of their ability to deal with highly complex, non-linear, and multi-dimensional machining optimization problems [5]. In the current trend of optimizing machining process parameters, various evolutionary or meta-heuristic algorithms have been used [6].

The most recent research of the meta-heuristic algorithms for machining process parameters optimization have been demonstrated by Zain et al. [7], Rao et al. [8], Samanta and Chakraborty [9], Madić et al. [10], Yildiz [4, 1, 11], Ciurana et al. [12], Pansare and Kavade [13], Liu et al. [Liu]. A comprehensive review paper regarding machining parameters optimization by means of the meta-heuristic algorithms is presented by Yusup et al. [6]. As has been reported in the literature, three types of meta-heuristic-based search algorithms viz. genetic algorithm (GA), simulated annealing (SA) and particle swarm optimization (PSO) have been mostly applied in the domain of the machining parameters optimization. However, in recent years there is an increasing trend in the application of other meta-heuristic algorithms such as ant colony optimization (ACO), artificial bee colony (ABC), improved harmony search algorithm (IHSA), and cuckoo search algorithm (CSA) for solving machining optimization problems.

This paper aims at comparing the performance of the GA, SA, IHSA and CSA when applied to different machining optimization problems. After a brief discussion of the basic methodology, characteristics and principles of the meta-heuristic algorithms in general, five machining optimization case studies are considered. The obtained optimization results of the GA, SA, IHSA and CSA are compared and discussed. The optimization results are also compared with those derived by the earlier researchers.

2. META-HEURISTIC ALGORITHMS

The algorithms used for solving optimization problems can be very diverse, from conventional algorithms to modern meta-heuristic algorithms [14]. The optimization algorithms developed so far can be broadly classified into deterministic and stochastic. The main difference between deterministic and stochastic algorithms is that in stochastic methods, the points that do not strictly improve the objective function can also be created and take part in the search process [15]. Most conventional or classic algorithms are deterministic. Some deterministic optimization algorithms use gradient information; they are called gradient-based algorithms (such as the Newton-Raphson algorithm). These algorithms use the function values and their derivatives and find a greater use in solving smooth unimodal problems. Gradient-free/derivative-free algorithms do not use any derivative, but only the function values. Hooke-Jeeves pattern search and Nelder-Mead downhill simplex are examples of gradient-free algorithms [14]. Within stochastic algorithms, there are heuristic and meta-heuristic algorithms. Loosely speaking, heuristic means "to find" or "to discover by trial and error". Quality solutions to a tough optimization problem can be found in a reasonable amount of time, but there is no guarantee that optimal solutions are reached [14]. While the heuristic algorithms resemble trial and error mechanisms, and depend on computational capacity, the meta-heuristic algorithms tend to learn as they run, and tend to be more intelligent and adaptive [16].

The term meta-heuristics was introduced by Glover [17] and represents a class of promising algorithms for solving hard optimization problems. The meta-heuristic algorithms are aiming at efficient and comprehensive exploration of the search space, using the governing mechanisms which imitate certain strategies taken from nature, social behavior, physical laws, etc., in order to find near optimal solutions. Some popular global optimization algorithms include: GA, SA, PSO, ACO, IHSA, CSA, artificial bee colony (ABC), taboo search (TS), artificial immune system (AIS), teaching-learning-based optimization (TLBO) algorithm, gravitational search algorithm (GSA), shuffled frog leaping (SFL), scatter search (SS), firefly algorithm (FA), etc. Besides these well known algorithms, the investigations on the meta-heuristic algorithms are still being done and new algorithms are being developed continually. The past 20 years have witnessed the development of numerous meta-heuristic algorithms in various communities that are at the intersection of several fields, including artificial intelligence, computational intelligence and soft computing [18].

Optimization based on using meta-heuristic algorithms starts with an initial set of independent variables and then evolves to obtain the global minimum/maximum of the objective (fitness) function. The objective function is a mathematical model (function) that assigns a value to each solution in the search space. Starting from an initial solution built by some heuristic, meta-heuristics improve it iteratively until a stopping criterion is met. The stopping criterion can be elapsed time, number of iterations, etc. The operation of meta-heuristic algorithms works in such a way to determine the final solution, only some existing solutions are actually visited. The search is conducted under a process that is specific to each meta-heuristic algorithm, but it is a way that attempts to intelligently find good solutions. However, there is no guarantee that the solution returned by a meta-heuristic is the best [19]. A universal step by step optimization procedure for any type of meta-heuristic algorithm can be described as follows [20]:

- Initializing the population in the search domain by seeding the population with random values.
- Evaluating the fitness for each individual of the population.
- Generating a new population by reproducing selected individuals through evolutionary operations, such as crossover, mutation, and so on.
- Looping to step 2 until stopping criteria are satisfied.

All the meta-heuristic algorithms use certain trade-off of local search and global exploration. A variety of solutions is often realized via randomization which provides a good way of moving away from the local search to that on the global scale [14].

The two main concepts of any meta-heuristic algorithms are: intensification and diversification, or exploitation and exploration [21]. Diversification means to generate diverse solutions so as to explore the search space on the global scale, while intensification means to focus on the search in a local region by exploiting the information that a current good solution is found in this region. This is in combination with the selection of the best solutions [14].

The fine balance between these two components is very important to the overall efficiency and performance of an algorithm. Too little exploration and too much exploitation could cause the system to be trapped in local optima, which makes it very difficult or even impossible to find the global optimum. On the other hand, in the case of too much exploration but too little exploitation, it may be difficult for the system to converge and thus it slows down the overall search performance [14]. The main difference between developed meta-heuristic algorithms is in the means by which they try to achieve this balance.

Meta-heuristic algorithms can be divided into two categories: single-solution meta-heuristic algorithms where a single solution (and search trajectory) is considered at a time and population meta-heuristic algorithms where a multiplicity of solutions evolve concurrently. Within each category, it is also possible to distinguish between primarily constructive meta-heuristic algorithms, where a solution is built from scratch (through the introduction of new elements at each iteration) and improvement meta-heuristic algorithms which iteratively modify a solution [22].

The main idea behind designing the meta-heuristic algorithms is to tackle complex optimization problems where other optimization methods have failed to be effective. These methods are now recognized as some of the most practical approaches for solving many real world problems [18]. There are several advantages of using meta-heuristic algorithms for optimization, namely:

- Broad applicability: they can be applied to any problems that can be formulated as function optimization problems.
- Hybridization: they can be combined with more traditional optimization techniques.
- Ease of implementation: typically easier to understand and implement.
- Efficiency and flexibility: they can solve problems larger problems faster. Moreover, they are simple to design and implement, and are very flexible [18].
- The use of meta-heuristics can be justified due to: (*i*) complexity of the internal problem that prevents the application of exact techniques and (*ii*) a very large quantity of possible solutions that prevent the use of exhaustive algorithms [23]. [23].

However, there are some disadvantages of the meta-heuristic algorithms that should be here noted:

- In general, the optimization performance is highly dependent on fine parameter tuning.
- They do not have "sound" mathematical foundation, when compared to more traditional techniques [24].
- They cannot prove optimality.
- They cannot probably reduce the search space.
- Repeatability of optimization results obtained with the same initial condition settings is not guaranteed.

3. Application of Meta-Heuristic Algorithms for Solving Machining Optimization Problems

In this section, the performance of the selected meta-heuristic algorithms for the optimization of real engineering optimization problems will be compared. Reviewing the literature one can see that meta-heuristic algorithms have been successfully applied to a wide range of optimization problems. For illustrative purposes, this section will thus focus on a particular class of problems, namely machining optimization problems, as they exhibit an impressive record of successful implementations. This paper aims at comparing the optimization results of the real-coded GA (RCGA), SA, IHSA and CSA for different machining optimization problems. A brief description of these algorithms is provided in the following subsections.

3.1. Real coded genetic algorithms

Genetic algorithms, developed by Holland [25], are artificial genetic systems based on the process of natural selection. They are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover. The evolution of population is performed through a specific number of generations where the next generation gives a better solution than the previous one. In RCGA, the solution is directly represented as a vector of real parameter decision variables; thus, the representation of the solutions is very close to the natural formulation of many problems. The use of real-parameters makes it possible to use large domains for variables. The RCGA have been used to solve engineering problems that are complex and difficult to solve by conventional optimization methods. Implementation of the RCGA requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators, initialization, termination and evaluation function.

3.2. Simulated annealing

Initially presented by Kirkpatrick et al. [26], SA is a random search technique for global optimization problems able to escape local optima. The concept of simulated annealing is taken from nature and it mimics the metals recrystallization in the process of annealing. Annealing is the slow cooling of metal that produces good low energy state crystallization, whereas fast cooling produces poor crystallization. SA uses single point search method. It is a memoryless search algorithm in the sense that no information is saved from previous searches [23]. SA algorithm starts with an random initial design vector (solution) X_i and high temperature T. A second design point is created at random in the vicinity of the initial point and the difference in the function values ΔE at these two points is calculated as [27]:

$$\Delta E = \Delta f = f_{i+1} - f_i \equiv f(X_{i+1}) - f(X_i) \tag{1}$$

If the new solution's objective function value is smaller, the new solution is automatically accepted and becomes the current solution from which the search will continue. Otherwise the point is accepted with a probability $e^{(-\Delta E/kT)}$ where k is the Boltzmann's constant. This completes one iteration of the SA. Due to the probabilistic acceptance of a nonimproving solution, SA can escape from local optima. At a certain temperature T, a predetermined number of new points are tested. The algorithm is terminated when current value of temperature is small enough or when changes in function values (Δf) are sufficiently small.

3.3. Improved harmony search algorithm

Harmony search algorithm (HSA), developed by Geem [28] has been successfully applied to various benchmark and real world problems. It is a meta-heuristic optimization algorithm conceptualized by using the musical process of searching for a perfect state of harmony. Musical performances seek to find pleasing harmony (a perfect state) as determined by an aesthetic standard, just as the optimization process seeks to find a global solution (a perfect state) as determined by an objective function. Optimizations procedure of the HSA includes five steps [29]. The algorithm requires several parameters [26], including harmony memory (*HM*), number of improvisations (*NI*), harmony memory considering rate (*HMCR*), pitch adjusting rate (*PAR*), bandwidth vector (bw). Mahdavi [29] suggested an improvement of the traditional HSA with the key difference in the way of adjusting PAR and bw. Namely, to improve the performance of the HSA and eliminate the drawbacks that originate from fixed values of *PAR* and *bw*, the improvement of the traditional HSA uses variables *PAR* and *bw* in the improvisation step. In this paper, the improved harmony search algorithm (IHSA) is used.

3.4. Cuckoo search algorithm

Cuckoo search algorithm (CSA) is a novel population based stochastic global search meta-heuristic algorithm developed by Yang and Deb [30]. CSA is inspired by natural mechanisms and mimics, the breeding behavior of some cuckoo species that lay their eggs in the nests of host birds. Each egg represents a solution, and a cuckoo egg represents a new solution. The goal is to use new and potentially improved solutions (cuckoos) to replace worse solutions in the nests. CSA can be briefly described using the following three idealized rules [30]:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high quality of eggs (solutions) will carry over to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with a probability Pa [0, 1].

3.5. Parameter settings of the RCGA, SA, IHSA and CSA

Due to the unique functionality of each type of the meta-heuristic algorithm, the comparison of meta-heuristic algorithms is in many ways more difficult than other algorithmic comparisons [31, 32]. As it is known, each type of the meta-heuristic algorithm has a number of parameters that must be set before the algorithm execution. These parameters are vital components of an algorithm and can be changed to alter the performance of the algorithm. Although some general guidelines about selection of these parameters exist in relevant literature, it has been widely reported that the optimal setting is strongly related to the design problem under consideration. In that sense, it is decided to use the most common parameter settings for each of the meta-heuristic algorithms. The list of the parameter settings used for each meta-heuristic algorithm applied is given in Table 1.

As meta-heuristic algorithms have stochastic nature, each run will usually produce different results. Hence, to reduce randomness, each optimization problems has been attempted 20 times with different seeds and the best solution for each algorithm is recorded.

Table 1 Meta-heuristics and their parameter settings used

RCGA	Population size = 20 Number of generations = 100 Selection function = Stochastic uniform Reproduction: Elite count: 2; Crossover rate: 0.8 Mutation: constraint dependent default Crossover function: Scattered Migration: Migration direction: forward; Migration interval: 20; Migration fraction: 0.2.
SA	Annealing function: Fast annealing Reannealing interval: 100 Temperature update function: Exponential temperature update Initial temperature: 100 Acceptance probability function: Simulated annealing acceptance Start point: set such that all variables take lower bound values
IHSA	Harmony memory size: 10 Harmony memory consideration rate: 0.95 Minimum pitch adjusting rate: 0.1 Maximum pitch adjusting rate: 0.85 Minimum bandwidth: 0.001 Maximum bandwidth: 0.8 Number of improvisations: 50000
CSA	Number of nests: 20 Discovery rate : 0.25

3.6. Machining optimization problem formulation

In machining optimization problems the aim is usually to minimize/maximize an objective function, often representing a machining performance, under some machining parameter constraints. The problem may be expressed as follows:

$$\begin{aligned} &\text{Minimize(or maximize)}\, f(x) \\ &\text{subject to: } g_i(X) \le 0,\ i = 1...m \\ &\qquad X_j^l \le X_j \le X_j^u,\ j = 1...n \end{aligned} \tag{2}$$

where, $\boldsymbol{X}$ is the vector of machining parameters; $f(x)$ is the objective function to be minimized (maximized); $g_i(X)$ is the *i-th* functional constrain; and X_j^l and X_j^u are lower and upper bounds of *j-th* machining parameter X_j.

4. CASE STUDIES

The search for papers dealing with machining optimization problems is restricted to those based on empirical models developed using the regression analysis. The reason for this is that, since regression models are explicitly expressed, the obtained optimization results can readily be checked and compared.

4.1. Case study I

Pansare and Kavade [13] present an experimental work to investigate the effects of the cutting parameters (feed rate, cutting speed and depth of cut) on surface roughness in turning oil hardened non-shrinkable steel. Based on the experimental results from the Taguchi's L_9 orthogonal array, the authors have developed the relationship between the cutting parameters and the surface roughness using multiple linear regression. The regression equation developed by Pansare and Kavade [13] is as follows:

$$R_a = 8.11 - 0.0217 \cdot A - 25.9 \cdot B - 6.37 \cdot C + 0.0563 \cdot AB + 0.0153 \cdot AC + 19.4 \cdot BC \quad (3)$$

where A is the cutting speed (m/min), B is the feed rate (mm/rev), C is the depth of cut (mm), and R_a is the surface roughness (μm).

In an attempt to obtain optimum turning parameters for minimum surface roughness value, the authors apply ACO. The turning optimization problem is formulated as follows:

$$\begin{aligned} &\text{Minimize } R_a = f(A,B,C) \\ &\text{subject to: } 150 \le \text{A} \le 250 \text{ (m/min)};\ 0.1 \le B \le 0.2\left(\text{mm/rev}\right);\ 0.5 \le \text{C} \le 1.5 \text{ (mm)} \end{aligned} \quad (4)$$

The optimum turning parameter values and surface roughness value obtained by Pansare and Kavade [13] are given in Table 1. The optimization problem formulated in Eq. 4 is attempted using the RCGA, SA, IHSA and CSA with the parameter settings as discussed before. The best obtained optimization results are summarized in Table 2.

Table 2 Optimization results for case study I

Algorithm	Cutting speed (m/min)	Feed rate (mm/rev)	Depth of cut (mm)	Surface roughness (μm)
ACO*	150	0.1032	1.4677	0.0114
RCGA	150	0.1	1.5	–0.093
SA	150.011	0.1	1.5	–0.092
IHSA	150	0.1	1.5	–0.093
CSA	153.3242	0.1015	1.4391	0.0727

*Results reported by Pansare and Kavade [13]

Regarding the optimization performance of the selected meta-heuristic algorithms in terms of the solution quality obtained and the computational time, the following observations are made: (*i*) IHSA is most consistent, (*ii*) RCGA yields scatter results, (*iii*) SA has the slowest convergence, (*iv*) although better than ACO, CSA performs poorer that the other meta-heuristic algorithms, (*v*) although statistically is adequate, in some cases, mathematical model may not have adequate physical meaning and should be used with care. In such situations, an experimental verification of the obtained optimization results should be carried out.

4.2. Case study II

Shivakoti et al. [33] have carried out turning experiments on mild steel workpiece by considering the spindle speed and cutting speed at different feed rate values. The ultimate goal is to find the optimum parameters values for turning operations for maximizing the material removal rate (MRR). Based on the experimental results, the authors have developed explicit mathematical model based on the regression analysis. The regression equation developed for calculation of MRR is as follows:

$$MRR = 1.42 - 1.83 \cdot A - 0.9 \cdot B + 10 \cdot C + 103 \cdot AB - 112 \cdot AC + 0.000014BC \qquad (5)$$

where A is the feed rate (mm/rev), B is the spindle speed (rpm), and C is the cutting speed (m/min).

To identify optimal cutting parameters, Eq. 4 is selected as objective function and solved using GA by formulating the optimization problem as:

$$\begin{aligned}&\text{Maximize MRR } = f(A,B,C)\\&\text{subject to: } 0.62 \le \text{A} \le 0.98 \text{ (mm/rev)};\ 40 \le B \le 1000 \text{ (rpm)};\ 3.5 \le \text{C} \le 95.5 \text{ (m/min)}\end{aligned} \qquad (6)$$

The obtained optimization results by Shivakoti et al. [33] and the optimization results of the RGCA, SA, IHSA and CSA are compared in Table 3.

Table 3 Optimization results for case study II

Algorithm	Feed rate (mm/rev)	Spindle speed (rpm)	Cutting speed (m/min)	MRR (mm^3/s)
GA*	0.582	891.520	25.580	51247.549
RCGA	0.98	999.926	3.936	99591
SA	0.98	1000	3.5	99691
IHSA	0.98	1000	3.5	99691
CSA	0.9717	949.3741	5.9017	93581

*Results reported by Shivakoti et al. [33]

Regarding the optimization performance for the second case study, the following observations are made: (*i*) again, IHSA produces most consistent solutions, (*ii*) scattered nature of the RCGA solutions is evident, (*iii*) Although SA converges too slowly in comparison to other algorithms, it yields the best solution.

4.2. Case study III

Mohruni [34] has conducted a machining experiment to measure the surface roughness value in the end milling operation. Based on the data for the real machining experiments, Zain et al. [7] have developed regression models for each cutting tool to predict surface roughness. The obtained equations are obtained as:

$$R_{uncoated} = 0.451 - 0.00267 \cdot A + 5.671 \cdot B + 0.0046 \cdot C \tag{7a}$$

$$R_{TiA1N} = 0.292 - 0.000855 \cdot A + 5.383 \cdot B - 0.00553 \cdot C \tag{7b}$$

$$R_{SNTR} = 0.237 - 0.00175 \cdot A + 8.693 \cdot B - 0.00159 \cdot C \tag{7c}$$

where: where A is the cutting speed (m/min), B is the feed rate (mm/tooth), and C is the radial rake angle (°).

Zain et al. [7] propose Eq. 7c as the fitness function in the GA optimization module. The minimization of the fitness function value of Eq. 7c is subjected to the boundaries (limitations) of cutting condition values. The optimization problem is formulated as:

$$\begin{aligned}&\text{Minimize } R_{SNTR} = f(A,B,C)\\&\text{subject to: } 124.53 \le \text{A} \le 167.03 \text{ (m/min)};\ 0.025 \le B \le 0.083 (\text{mm/tooth});\ 6.2 \le \text{C} \le 14.8\ (^\circ)\end{aligned} \tag{8}$$

To determine best optimal milling parameter values, Zain et al. [7] have tried several combinations of GA parameter rates. The obtained optimization results are given in Table 4. To investigate the efficiency of the PSO algorithm for machining problem solving, the optimization problem formulated in Eq. 8 is attempted by Pare et al. [35]. Although the authors report superior results than those obtained by Zain et al. [7], the optimization results are not valid since the constraint for the feed rate is not satisfied.

Table 4 Optimization results for case study III

Algorithm	Cutting speed (m/min)	Feed rate (mm/tooth)	Radial rake angle (°)	Surface Roughness (μm)
GA*	167.029	0.025	14.769	0.138* 0.13854154***
PSO**	124.53	0.0025	6.2	0.0186
-	167.03	0.025	14.8	0.13849225

* Results reported by Zain et al. [7]
** Results reported by Pare et al. [35]
***Results obtained after substituting the obtained parameter values in Eq. 6c

Regarding this case study, it should be noted that application of any classical mathematical or meta-heuristic optimization technique is redundant. Since the mathematical model is expressed as a linear combination of independent parameters with no interaction terms, and since there are no other additional non-linear constraints, the optimization solution of the above machining optimization problems is obvious. For this case study, from Eq. 7c it is seen that surface roughness increases with an increase in feed rate, and decreases with an increase in cutting speed and radial rake angle. Therefore, considering machining constraints, the optimal machining parameters values can be easily determined, i.e. they represent the boundary points in the experimental space covered. The best optimal machining parameter values and corresponding surface roughness value are given in Table 4.

4.2. Case study IV

Saravanakumar et al. [36] have conducted turning experiments on Inconel 718 in order to investigate the influence of machining process parameters such as the cutting speed, feed rate, and depth of cut on the output parameters such as MRR and surface roughness. Using the experimental data, the authors have developed regression equations for material removal rate (MRR) and surface roughness R_a in the following form:

$$MRR = 19158 - 298 \cdot A - 112136 \cdot B + 91493 \cdot C + 1749 \cdot AB + 1417 \cdot AC + \\ + 537343 \cdot BC - 7880 \cdot ABC \tag{9}$$

$$R_a = 23.6 - 0.331 \cdot A - 110 \cdot B - 88 \cdot C + 1.66 \cdot AB + 1.29 \cdot AC + 463 \cdot BC - 6.93 \cdot ABC \tag{10}$$

where A is the cutting speed (m/min), B is the feed rate (mm/rev), and C is the depth of cut (mm).

To identify optimal cutting parameters, Eq. 9 is to be maximized whereas Eq. 10 is to be minimized considering the following machining parameter constraints:

$$60 \le \mathrm{A} \le 80 \text{ (m/min)};\ 0.15 \le B \le 0.25 \left(\text{mm/rev}\right);\ 0.1 \le C \le 0.25 \text{ (mm)} \tag{11}$$

The optimization problem is then selected as an objective function and solved using GA. The optimization results obtained by Saravanakumar et al. [36] as well using RCGA, SA, IHSA and CSA are given in Table 5.

Table 5 Optimization results for case study IV

Algorithm	Cutting speed (m/min)	Feed rate (mm/rev)	Depth of cut (mm)	Surface Roughness (μm)	MRR (mm^3/min)
GA*	79.99	0.25	0.1	-	2122.23
	79.9	0.15	0.1	0.69	-
RCGA	79.998	0.25	0.1	-	2123. 89
	79.999	0.15	0.1	0.6896	-
SA	80	0.25	0.1	-	2124.275
	79.943	0.15	0.1	0.689956	
IHSA	80	0.25	0.1	-	2124.275
	80	0.15	0.1	0.689	-
CSA	79.9374	0.2493	0.1397	-	2054.9
	78.2857	0.1492	0.1086	0.7871	-

* Results reported by Saravanakumar et al. [36]

Regarding the optimization performance for this case study, the following observations are made: (*i*) SA yields the best solution, (*ii*) For maximization of MRR, the best and the most consistent results are obtained by SA and IHSA, (*iii*) CSA performs poorly with very diversified solutions, (*iv*) for surface roughness minimization best optimization solution as achieved with IHSA.

4.2. Case study V

Sharma et al. [37] have conducted turning experiments on aluminum 6061 alloy and metal matrix composites of aluminum. An attempt has been made to establish mathematical relationships between the cutting parameters and surface roughness based on the regression analysis. Mathematical relationships between the responses and the machining parameters for Al/SiC (5%) and Al/SiC(10%) composite material are obtained as follows:

$$\begin{aligned} R_{a|Al|SiC(5\%)} &= -18.7 - 0.00122 \cdot A + 443 \cdot B + 10.4 \cdot C + 0.000001 \cdot A^2 - \\ &-2541 \cdot B^2 - 4.71 \cdot C^2 - 0.0015 \cdot AB - 0.00229 \cdot AC + 40.7 \cdot BC \end{aligned} \tag{12a}$$

$$\begin{aligned} R_{a|Al|SiC(10\%)} &= -3.61 + 0.00252 \cdot A - 14.2 \cdot B + 13.2 \cdot C + 0.000002 \cdot A^2 + \\ &+640 \cdot B^2 - 9.67 \cdot C^2 - 0.0407 \cdot AB - 0.00212 \cdot AC + 21.2 \cdot BC \end{aligned} \tag{12b}$$

where A is the cutting speed (m/min), B is the feed rate (mm/rev), and C is the depth of cut (mm).

In an attempt to obtain optimum turning parameters for minimum surface roughness value, Sharma et al. [37] apply PSO considering the following constraints for the machining parameters:

$$228 \le A \le 740 \text{ (mm/min)}; \; 0.05 \le B \le 0.1 (\text{mm/rev}); \; 0.4 \le C \le 1 \text{ (mm)} \tag{13}$$

The optimization results obtained by Sharma et al. [37] as well using RCGA, SA, IHSA and CSA are given in Table 6.

Table 6 Optimization results for case study V

Algorithm	Cutting speed (mm/min)	Feed rate (mm/rev)	Depth of cut (mm)	R_a (μm) Al/SiC (5%)	R_a (μm) Al/SiC (10%)
PSO*	233	0.05	0.4	1.2883* 0.874501**	–
	228	0.05	0.4	–	1.558* 1.84469***
RCGA	726.542	0.05	0.4	0.23937	–
	228	0.05	0.4	–	1.84469
SA	724.214	0.05	0.4	0.2411	–
	228	0.05	0.4	–	1.84469
IHSA	740	0.05	0.4	0.22936	–
	228	0.05	0.4	–	1.84469
CSA	736.2804	0.0534	0.4026	0.9216	–
	370.7913	0.0539	0.4880	–	2.9578

*Results reported by Sharma et al. [37]
** Results obtained after substituting the obtained parameter values in Eq. 11a
*** Results obtained after substituting the obtained parameter values in Eq. 11b

Regarding the optimization performance for this case study, the following observations are made: (*i*) IHSA yields the best optimization results followed by RCGA, (*ii*) For the first objective functions, SA and CSA perform poorly with very diversified solutions, (*iii*) When optimizing second objective function, all algorithms except CSA have proved efficient and found best solution.

5. CONCLUSION

In this paper, an attempt is made to compare the optimization solutions obtained by the meta-heuristic algorithms applied to solving machining optimization problems. Four meta-heuristic algorithms are taken into consideration, namely, RCGA, SA, IHSA and CSA and applied to solving five machining case studies. On the basis of the obtained results the following conclusions can be made:

In the case of machining optimization problems IHSA has proved to be the most efficient meta-heuristic algorithm in terms of computational time and solution quality. The IHSA parameter settings as proposed in this study enable IHSA efficient and fast exploration of the search space without getting trapped in local minima.

- Although SA has the slowest convergence, it can be used efficiently for solving machining optimization problems. In some cases, it is found to be superior to the other meta-heuristic algorithms.
- CSA performs much poorer that the other meta-heuristic algorithms indicating a need for fine tuning of the parameters for solving machining optimization problems.
- Not intending to be suspicious about the search space exploration capability and efficiency of other meta-heuristic algorithm, the optimization results indicate that more effort is needed for efficient implementation of RCGA and CSA to the solving machining optimization problems.
- An approach presented in [32] based on Taguchi's experimental design has proved efficient for fine tuning of IHSA parameters as used in this study. Therefore, in the authors' opinion, it represents an appropriate way for fine tuning of the RCGA and CSA algorithm parameters so as to improve its performance for solving machining optimization problems.
- The analysis of the obtained optimization results indicate that, regardless of the meta-heuristic algorithm applied, there is a need to check the boundary points in the hyperspace of the machining parameters in order to check the existence of optimal solution.
- An analysis of the mathematical model and optimization problem formulation is to be done prior to the application of a meta-heuristic algorithm, since in some situations it may happen that the optimal conditions can be determined in a much easier way.
- Finally, it is necessary to emphasize the need for validation of the optimization solutions obtained using meta-heuristic algorithms.

REFERENCES

1. Yıldız, A.R., 2009, *A novel particle swarm optimization approach for product design and manufacturing*, The Internation Journal of Advanced Manufacturing Technology 40(5). pp. 617–628.
2. Marinković, V., Madić, M., 2011, *Optimization of surface roughness in turning alloy steel by using Taguchi method*, Scientific research and Essay, 6 (16), pp. 3474-3484.
3. Madić, M., Radovanović, M., 2010, *Savremene metode optimizacije obradnih procesa*, IMK-14 Istraživanje i razvoj, 37(4), pp. 19-24.
4. Yıldız, A.R., 2009, An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry, Journal of Materials Processing echnology 209(6), p. 2773–2780.
5. Mukherjee, I., Ray, P.K., 2006, *A review of optimization techniques in metal cutting processes*, Computers & Industrial Engineering 50(4), pp. 15–34.
6. Yusup, N., Zain, A.M., Hashim, S.Z.M., 2012, Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007-2011), Expert System with Applications, 39(10), pp. 9909-9927.
7. Zain, A.M., Haron, H., Sharif, S., 2010, Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process, Expert Systems with Applications 37(6), pp. 4650–4659.
8. Rao, R.V., Pawar, P.J., Davim, J.P., 2010, *Optimisation of process parameters of mechanical type advanced machining process using a simulated annealing algorithm*, International Journal of Materials and Product Technology, 37 (1-2), pp. 83-101.
9. Samanta, S., Chakraborty, S., 2011, *Parametric optimization of some non-traditional machining processes using artificial bee colony algorithm,* Engineering Applications of Artificial Intelligence 24(6), pp. 946–957.
10. Madić, M, Marković, D, Radovanović, M, 2012, *Optimization of surface roughness when turning polyamide using ANN-IHSA approach*, International Journal of Engineering & Technology, 1(4), pp. 432-443.
11. Yildiz, A.R., 2013, *Cuckoo search algorithm for the selection of optimal machining parameters in milling operations*, The International Journal of Advanced Manufacturing Technology, 64(1-4), pp. 55-61.
12. Ciurana, J., Arias, G., Özel, T., 2009, Neural network modelling and particle swarm optimization (PSO) of process parameters in pulsed laser micromachining of hardened AISI H13 steel, Materials and Manufacturing Processes, 24(3), pp. 358–368.
13. Pansare, V.B., Kavade, M.V., 2012, *Optimization of cutting parameters in multipass turning operation using ant colony algorithm,* International Journal of Engineering Science & Advanced Technology, 2(4), pp. 955 – 960.
14. Yang, X.S., 2011, *Review of Metaheuristics and Generalized Evolutionary Walk Algorithm*, International Journal of Bio-Inspired Computation, 3(2), pp. 77-84.
15. Al-Harkan, I.M, Trafalis, T.B., 2002, A hybrid scatter genetic tabu approach for continuous global optimization, 11-31: In Combinatorial and Global Optimization, P.M. Pardalos, A. Migdalas, R. Burkard, 2002, World Scientific Publishing Co.
16. Manda, K,. Satapathy, S.C., Poornasatyanarayana, B., 2012, *Population based meta-heuristic techniques for solving optimization problems: A selective survey*, International Journal of Emerging Technology and Advanced Engineering, 2(11), pp. 206-211.
17. Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, Computers & Operations Research, 13(5), pp. 533–549.
18. Gholizadeh, S., Barati, H., 2012, *Comprative study of three metaheuristics for optimum design of trusses*, International Journal of Optimization in Civil Engineering, 3, pp. 423-441.
19. Gomes de Freitas, F.G., Brito Maia, C.L., Lima de Campos, G.A., Teixeira de Souza, J., 2010, *Optimization in Software Testing Using Metaheuristics*, Revista de Sistemas de Informação da FSMA, 5, pp. 3-13.
20. Khajehzadeh, M., Raihan Taha, M., El-Shafie, A., Eslami, M., 2011, *A Survey on Meta-Heuristic Global Optimization Algorithms*, Research Journal of Applied Sciences, Engineering and Technology 3(6), pp. 569-578.
21. Blum, C., Roli, A., 2003, Metaheuristics in combinatorial optimisation: Overview and conceptual comparision, ACM Computing Surveys, 35(3) 268-308.

22. Gendreau, M., Potvin, J.Y., 2005, *Metaheuristics in Combinatorial Optimization*, Annals of Operations Research, 140(1), pp. 189-213.
23. Talbi, E., 2009, Metaheuristics: From Design to Implementation, Wiley Publishing.
24. Viana, A., Sousa, J.P., Matos, M.M, 2005, Constraint oriented neighborhoods – a new search strategy in metaheuristics, In: Metaheuristics: Progress as real problem solvers, Edited by: T. Ibakari, K. Nonobe, M. Yagiura, Springer.
25. Holland, J., 1975*, Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Anbor.
26. Kirkpatrick, S., Gelatt, C., Vecchi, M., 1983, *Optimization by Simulated Annealing*, Science, 220(4598), pp. 671-680.
27. Rao, S.S., 1996, Engineering Optimization: Theory and Practice, John Wiley & Sons.
28. Geem, Z.W., Kim, J.H., Loganathan, G.V., 2001, *A new heuristic optimization algorithm: harmony search*, Simulation, 76(2), pp. 60–68.
29. Mahdavi, M., Fesanghary, M., Damangir, E., 2007, *An improved harmony search algorithm for solving optimization problems*, Applied mathematics and computation, 188(2), pp. 1567–1579.
30. Yang, X.S., Deb, S., 2010, *Engineering optimisation by cuckoo search*, International Journal of Mathematical Modelling and Numerical Optimisation 1(4), pp. 330 – 343.
31. Silberholz, J., Golden, B., 2010, *Comparison of metaheuristics*. In: Gendreau, M., Potvin, J.-V. (eds) Handbook of Metaheuristics, 2nd edn, Springer, Heidelberg.
32. Marković, D., Madić, M., Petrović, G., 2012, Assessing the performance of improved harmony search algorithm (IHSA) for the optimization of unconstrained functions using Taguchi experimental design, Scientific research and essays, 7(12), pp. 1312-1318.
33. Shivakoti, I., Diyaley, S., Kibria, G., Pradhan, B.B., 2012, *Analysis of Material Removal Rate using Genetic Algorithm Approach*, International Journal of Scientific & Engineering Research,3(5), pp. 1-6.
34. Mohruni, A.S., 2008, Performance evaluation of uncoated and coated carbide tools when end milling of titanium alloy using response surface methodology. Thesis for Doctor of Philosophy, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.
35. Pare, V., Agnihotri G., Krishna, C.M., 2011, *Optimization of Cutting Conditions in End Milling Process with the Approach of Particle Swarm Optimization*, International Journal of Mechanical and Industrial Engineering, 1(2), pp. 21-25.
36. Saravanakumar, K., Shaik Dawood, A.K., Pratheesh Kumar, M.R., *Optimization of CNC Turning Process Parameters on INCONEL 718 Using Genetic Algorithm*, Engineering Science and Technology: An International Journal, 2(4), pp. 532-537.
37. Sharma, A.V.N.L., Sandeep Kumar, P., Gopichand, A., Mohan Rao, R., 2012, *Optimal machining conditions for turning of Al/SiC MMC using PSO and Regression analysis*, International Journal of Engineering Research and Applications, 2(6), pp. 497-500.

POREĐENJE META-HEURISTIČKIH ALGORITAMA ZA REŠAVANJE PROBLEMA OPTIMIZACIJE PARAMETRA OBRADE

Optimizacija parametara obrade ne utiče samo da efikasnost i ekonomičnost obrade već i na finalni kvalitet proizvoda, pa samim tim ova tema je još uvek predmet izučavanja mnogih studija. Izbor optimalnih parametara obrade često se obavlja u dve faze, odnosno matematičkim modeliranjem performansi obrade i optimizacijom pomoću optimizacionih metoda. Njihova mogućnost da rešavaju složene i višediemzionalne optimizacione probleme učinila je da postanu veoma popularan izbor od strane većine istraživača. U ovom radu, autori su uporedili rezultate optimizacije različitih meta-heurističkih algoritama koji su primenjeni na rešavanje optimizacionih problema obrade. Razmatrana su četiri meta-heuristička algoritma i to: realno kodirani genetski algoritam, simulirano kaljenje, poboljšani algoritam harmonijskog pretraživanja i algoritam kukavice. Pomoću ovih meta-heurističkih algoritmama su tražene optimalne kombinacije različitih parametra obrade za pet studija slučaja uzetih iz literature. Rezultati optimizacije, dobijeni pomoću prethodno navedenih pet meta-heuristička algoritma za parametrsku optimizaciju procesa obrade, su upoređeni sa rezultatima poslednjih istraživanja.

Ključne reči: *mašinska obrada, optimizacija, meta-heuristički algoritmi*