Pergamon

International Journal of Machine Tools & Manufacture •• (2002) ••–••

INTERNATIONAL JOURNAL OF
MACHINE TOOLS & MANUFACTURE
DESIGN, RESEARCH AND APPLICATION

NC end milling optimization using evolutionary computation

V. Tandon [a], H. El-Mounayri [a,*], H. Kishawy [1,b]

[a] *Mechanical Engineering Department at Purdue School of Engineering, 723 West Michigan Street, SL 260, Indianapolis, IN 46202-5132, USA*

[b] *Mechanical Engineering Department, University of New Brunswick, Fredericton, NB, Canada E3B 5A3*

Received 23 October 2000; received in revised form 12 October 2001; accepted 16 October 2001

Abstract

Typically, NC programmers generate tool paths for end milling using a computer-aided process planner but manually schedule “conservative” cutting conditions. In this paper, a new evolutionary computation technique, particle swarm optimization (PSO), is proposed and implemented to efficiently and robustly optimize multiple machining parameters simultaneously for the case of milling. An artificial neural networks (ANN) predictive model for critical process parameters is used to predict the cutting forces which in turn are used by the PSO developed algorithm to optimize the cutting conditions subject to a comprehensive set of constraints. Next, the algorithm is used to optimize both feed and speed for a typical case found in industry, namely, pocket-milling. Machining time reductions of up to 35% are observed. In addition, the new technique is found to be efficient and robust. © 2001 Published by Elsevier Science Ltd.

Keywords: End milling; Optimization; Particle swarm optimizer

1. Literature survey

NC programs generated today, experience a large variation in cutting forces due to non-uniformity in metal removal along the cutter path. This may be due to a variety of factors, surface nature (curvature), tool inclination, cornering etc. In order to increase productivity, process parameters should be assigned according to the NC tool path in addition to the conditions of the part, tools, setup, and the machine. The idea is to change these variables according to the current in-process part geometry and tool path so that the cutting force is in control [1].

The current development uses experimental force data acquired by running exhaustive sample cases in 2.5 and 3-D. Such sample cases have been used by various researchers for evaluating optimization efficacy [2–4]. Tests are conducted at constant feed rates specified by the NC programmer to acquire the force variation data. Next, the force vs time data is correlated to position and the information used to insert optimized cutting parameters at the required positions (NC blocks). The modified NC program is then executed and a fresh force scenario acquired. Comparison of the two acquisitions demonstrates the efficiency of our development. Takata et al. [5] and Park et al. [6] reported the use of similar techniques for optimization implementations.

Most researchers have concentrated their efforts on modifying the feed rate alone, though some groups have tried to work with other parameters as well. All the reported efforts in the area have tried to re-schedule the feed rate per NC block instruction [1,2,7–16]. Essentially, each input NC block (either in NC or CLSF stage) is read, analyzed and then outputted with a modified feed rate code. It is the assumptions and the methods of analysis that differentiate various studies. Further, most of these studies do not address themselves to the specifics of the CNC end milling process and instead concentrate on a generic operation.

Most works on the development of NC code optimization developments involve the use of very simplistic forms of force prediction algorithms. Also, the literature shows use of volume of removed material as feedback or the machine tool horsepower as the constraint to regulate the feed rate.

* Corresponding author. Tel.: +1-317-278-3320; fax: +1-317-274-9744.

E-mail addresses: vtandon@iupui.edu (V. Tandon), helmoun@engr.iupui.edu (H. El-Mounayri), kishawy@unb.ca (H. Kishawy).

[1] Tel.: +1-506-458-7767; fax: +1-506-453-5025.

PII: S0890-6955(01)00151-1

The cutting force, as a single parameter for describing the net effect of all input variables, is found to be an optimal quantity for use as a feedback from the simulated process for feed rate optimization [8]. Simple specific energy models of metal cutting have been presented in textbooks for quite some time [17,18]. These are sometimes employed along with correction coefficients, by a number of researchers such as Wang [1], Fussell et al. [11], Lazaro et al. [19], etc.

Other more accurate models (e.g. Yellowey [20], Kline and DeVor [21], and Fussell et al. [22]) use the simple steady state milling force model by Yellowey, also known as the average cutting coefficient model implemented by Altintas et al. [23,24].

Bailey et al. [25] studied these different approaches and found the MRR approach to be adequate for estimating required spindle torque and power, but not when constraints such as chip load, maximum cutting force, deflections and chatter are considered. They also note the widespread use of static mechanistic models since, due to their closed form, they can be inverted to directly solve for the feed, which is the primary variable of concern.

The independent variables for optimal cutting parameters have been identified as the following:

- Tool diameter and length
- Number of passes
- Depth of cut (radial & axial) for each pass [26]
- Spindle speed and
- Feed (per tooth, per revolution or per unit time) [27].

Most studies state one of three objectives:

1. Minimum manufacturing cost [2].
2. Maximum production rate [26].
3. A variant of maximum productivity [5].

It has also been realized that a combination of the minimum production cost and minimum production time [28–31] is the most effective objective since neglecting either requirement alone does not do justice to the problem at hand. Agapiou [29–31] has investigated this concept extensively.

There are a variety of constraints (and various forms) that have been considered applicable by many researchers for different machining situations [32–34]. However, a comprehensive list of constraints reported in the literature is presented here:

1. Available feed and speeds (machine tool related), power, arbor rigidity, and arbor deflection [29].
2. Maximum available machine power and maximum permitted cutting edge load for roughing, and allowed maximum tool deflection for finishing [35].
3. Tool normal and tangential deflection limits [10].
4. Machine tool limiting power, spindle torque, maximum feed force, spindle speed boundaries, and feed per tooth boundaries [26].
5. Avoid excess cutting force and chatter vibration, to maintain the required machining accuracy [19].
6. The maximum cutting power available, the surface roughness required, the maximum cutting force permitted by the rigidity of the machine tool and the accuracy required, and the maximum feed rate and rotational speed available on the machine tool [28].
7. Constant cutting force, constant machining error, and the maintaining of moderate changes in cutting states [19].
8. Shank breakage and tooth breakage (limiting force and chip thickness respectively) [17].

Imani and Elbestawi [10], for example, recommend ignoring stability and maximum cutting force constraints for semi-finishing of free cutting steels.

Cutting force, once again, is found to be one of the most important process parameters used as a constraint in the cutting operation, as it relates to a large number of abnormal occurrences such as tool breakage and excess tool wear as well as basic data for estimation of chatter vibration and machining error [19].

Availability of quantitatively reliable machining performance equations relating the tool-life, forces, torque, power, surface finish, etc. to the cutting or process variables is critical to the development of an optimization algorithm [26]. It is not uncommon for researchers to forego this requirement and in some cases even assume a direct linear relation between the feed (independent variable) and the force (constraint) as well as the machining error. Such a study is presented by Takata et al. in [19]. While they report improvement in machining performance, it is easy to note that such improvement obtained is nothing more than a small portion of the possible gain. Further, in certain situations it may even result in wrong inferences leading to catastrophic tool failure conditions.

A similar approach is also reported by Weinert et al. [28]; their method of feed rate adaptation is based on the cross sectional area of the cut. While two correction coefficients have been included to account for influence of symmetry of engagement and the direction/inclination of the cut, essentially a linear relationship is assumed between the feed rate and the modified cross section parameter or MRR. However, an important addition to the literature is made by realizing that it is necessary to take into account the dynamic capabilities of the machine. The new feed-rate value must be added to the NC-file at the correct position. This is done to ensure that the calculated optimal feed rate is reached before the volume to be cut exceeds a given value. An applicable length of deceleration is calculated and included along with the cutter diameter as a safety parameter. This work is unique in considering control related issues with feed rescheduling.

2. Force model

A feed forward neural network with one to two hidden layers and sigmoid activation functions is designed to predict the maximum force, mean force and other relevant conditions (the details are given in [36]).

Back propagation (BP) is the training method of choice. Standard BP is a gradient descent algorithm and the name BP refers to the manner in which the gradient is computed for non-linear multi-layer ANN. One of the main attractions of using BP networks is that they tend to give reasonable results when presented with inputs that they have never seen. Typically, a new input will lead to an output similar to the correct output for input vectors used in the training, that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs, which is generally impossible in situations such as ours. Also, with this method the order in which the patterns are presented to the network do not influence the training. This is also because adaptation is done only at the end of each epoch.

A variation of the standard BP technique of adjusting the network weights is the Levenberg–Marquardt technique [10]. Similar to quasi-Newton methods, this algorithm approaches second-order training speed without much added computational expense.

A feed forward ALM network is designed and studied for effectiveness in learning the non-linear map between the input machining parameters and the output conditions. The ANN designed for this application is presented in Fig. 1.

3. Optimization of end milling

3.1. Objective function

In the case of end-milling a variety of objective functions have been proposed to date [35,10,17,19,26–28,32–34]. The more important ones found in the literature being:

(1) The machining time, t_m

$$t_m = \frac{\pi \cdot L \cdot D}{1000 \cdot v \cdot Z \cdot S_z}\left(\frac{W}{b}\right) \quad (1)$$

where L=length of workpiece [mm], D=cutter diameter [mm], W=width of workpiece [mm], v=cutting speed [m/min], Z=number of teeth/flutes, S_z=feed per tooth [mm/tooth], and b=width of cut /radial depth of cut [mm].

(2) Tool life (for ith operation), T_i

$$T_i = \frac{C_v^{\alpha} \cdot D^{\omega}}{v_i^{\alpha} \cdot S_{Z_i}^{\beta} \cdot d_i^{\gamma} \cdot b_i^{\delta} \cdot Z^{\lambda}} \quad (2)$$

where d=axial depth of cut [mm].

Here, α, β, γ, δ, ω, λ are exponents in the tool life equation and are experimentally determined. Also, C_v is the constant of proportionality. Note that these variables are unique for different tool–workpiece combinations.

(3) Production cost, C, for plain milling (combination of 1 & 2),

$$C = A_1 + A_2 \cdot v^{-1} \cdot S_z^{-1} \cdot b^{-1} + A_3 \cdot v^{\alpha-1} \cdot S_z^{\beta-1} \cdot d^{\gamma} \cdot b^{\delta-1} \quad (3)$$

where

$$A_1 = M \cdot t_1$$

$$A_2 = \frac{\pi \cdot L \cdot D \cdot M \cdot W}{1000 \cdot Z}$$

$$A_3 = [M \cdot t_c + C_t]\frac{\pi \cdot L \cdot D \cdot W \cdot Z^{\lambda}}{1000 \cdot Z \cdot C_v^{\alpha} \cdot D^{\omega}}$$

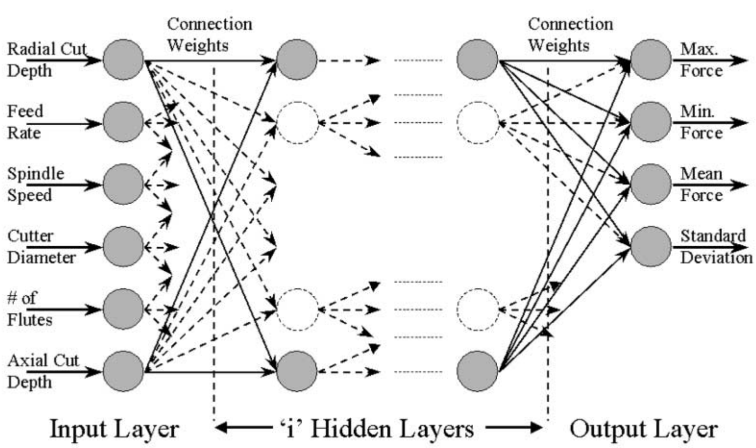


Fig. 1. Predictive force model network topology.

In another form, it may be cast as the production time for plain milling, t_p,

$$t_p=A_4+A_5\cdot v^{-1}\cdot S_z^{-1}\cdot b^{-1}+A_6\cdot v^{\alpha-1}\cdot S_z^{\beta-1}\cdot d^{\gamma}\cdot b^{\delta-1} \quad (4)$$

where

$$A_4=t_1$$

$$A_5=\frac{\pi\cdot L\cdot D\cdot W}{1000\cdot Z}$$

$$A_6=\frac{\pi\cdot L\cdot D\cdot W\cdot Z^{\lambda}\cdot t_c}{1000\cdot Z\cdot C_v^{\alpha}\cdot D^{\omega}}$$

Eq. (3) is used in this work to define the objective function; this will be illustrated in Section 4.4.

3.2. Constraints

Apart from the objective functions, there exist a number of constraints that must be satisfied for a meaningful optimization of the machining process. While some are obvious from the machine tool capabilities, others are derived from product requirements such as surface finish, force-bearing capacity of the tool etc. Jha et al. [37], in a study on milling cutter design, present the most comprehensive list of constraints. These are presented in Table 1.

From the list of constraints in Table 1, we find the following. The first two (dual) constraints are related to machine capabilities and have to be considered for a meaningful optimization exercise. Constraints 3 through 8 are related to the geometrical aspects of the manufacturing process and are as such not in the scope of our current exercise. For example, if the optimizer changes the depth of cut, it will be necessary to regenerate the tool path. The latter requires a feedback integration of the optimizer to the CAM application. It is however, necessary to note that the proposed optimizing algorithm can be readily extended to encompass multi-dimensional solution space. Constraints 9 through 13 are essentially force related and can be condensed to generate a single constraint on the maximum cutting force permissible. Note that since all five constraints are inequalities placing a limit on the cutting force a simple “minimum of” selection is enough to condense the constraints. Practically, during rough milling, horsepower limitation may be the active constraint, while during finish milling, surface finish may be the active constraint. The non-negativity constraint as such is redundant since parameters of concern are in any case limited to a subset of positive real space.

Thus, we arrive at the following constraints on the optimization problem, where v is the cutting velocity and S_z is feed per tooth

$$v_{\min}\le v\le v_{\max} \quad (5)$$

$$S_{z\,\min}\le S_z\le S_{z\,\max} \quad (6)$$

$$F(v,S_z)\le F_{\lim} \quad (7)$$

Here the first two constraints are machine tool related, specifically, the available range of spindle speed and feed rates. The third constraint is force related. As recognized earlier in the literature review, the cutting force, as a single parameter, is an optimal quantity for describing the net effect of all input variables, for use as a feedback to optimization. More specifically, $F_{\lim}$ epitomizes a variety of force related constraints. Here, it is important to note the complexity introduced in the optimization problem by the force function. While feed and speed define the two dimensions, force is the third dimension. This third dimension, however, is a complex function of the first two. Thus, the feasible region of the force constraint is also affected by the solution coordinates in the feed–speed space. This is illustrated in Fig. 2, which also demonstrates the complexity of the search for an optimum solution.

4. Particle swarm optimization

Evolutionary computation (EC) comprises a variety of methods including optimization paradigms that are based on evolution mechanisms such as biological genetics and natural selection.

While EC provides many characteristics that make it the method of choice in our problem situation, the most important reasons are firstly, these paradigms use direct “fitness” information instead of functional derivatives or other related knowledge. This fits in perfectly with our development since we do not have an explicit functional representation of the process model and hence the derivative etc. is also not known. Secondly, they use probabilistic, rather than deterministic, transition rules. This overcomes the problem of getting stuck in local optima prevalent with deterministic transition rules. Also, since we start with a diverse set of points, many optima can be explored efficiently, lowering the probability of getting stuck.

Particle Swarm Optimization (PSO) is a relatively new technique, first presented in 1995 [38], for optimization of continuous non-linear functions [39,40]. Jim Kennedy discovered the method through simulation of a simplified social model, the graceful but unpredictable choreography of a bird flock [41].

PSO is a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, so is computationally inexpensive in terms of both memory requirements and speed. These characteristics are of immense value to the application situation at hand.

PSO has been recognized as an evolutionary compu-

Table 1
Constraints and their expressions in terms of common variables

	Constraint	Expression in variables
1	Feed rate constraint, $f_{\max}$ & $f_{\min}$	$f_{\max} \le \frac{1000 \cdot Z}{\pi \cdot D} v \cdot S_z \le f_{\max}$
2	Spindle speed constraint	$N_{\max} \le \frac{1000}{\pi \cdot D} \cdot v \le N_{\max}$
3	Depth of cut constraint	$b_{\min} \le b \le b_{\max}$
4	At least two teeth in contact constraint	$\frac{Z \cdot d^{0.5}}{\pi \cdot D^{0.5}} \le 1$
5	Diameter of cutter	$D_{\min} \le D \le D_{\max}$
6	Angular pitch (α) constraint	$\frac{\pi}{Z \cdot \alpha} \le 1$
7	Tooth height constraint	$\frac{h \cdot Z}{1.2 \cdot D} \le 1$
8	Horsepower constraint	$\frac{Z \cdot D^{-1}}{\pi \cdot V_s} \cdot v \cdot S_z \cdot d \cdot b \le P_{\text{allow}}$
9	Maximum loading on feeding mechanism constraint	$\frac{Z \cdot D^{-1}}{\pi \cdot V_s} \cdot S_z \cdot d \cdot b \le F_t$
10	Surface roughness constraint	$\frac{1}{8 \cdot R_a} \cdot S_z^2 \le h_{\max}$
11	Bending stress constraint	$\frac{12F_a}{\pi \sigma_b w} Z \cdot h \cdot D^{-1} + \frac{6F_a}{\pi \sigma_b w} \cdot d \cdot D^{-1} \le 1$
12	Fatigue constraint	$\frac{F_a \cdot Z}{2\pi w} \cdot \left(\frac{1}{S_e} + \frac{1}{S_u}\right) \cdot Z \cdot D^{-1} \le 1$
13	Non-negativity constraint	$D \ge 0$; $N \ge 0$; $d \ge 0$; $Z \ge 0$; $f \ge 0$; $h \ge 0$; $\alpha \ge 0$

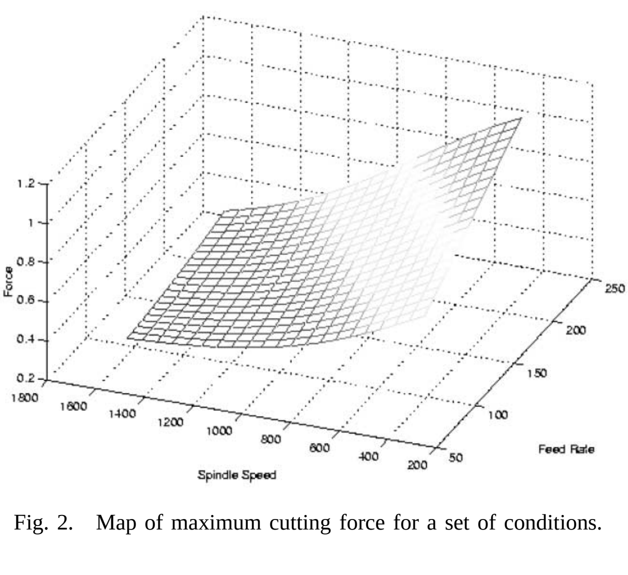


Fig. 2. Map of maximum cutting force for a set of conditions.

tation technique [42] and has features of both genetic algorithms (GA) and evolution strategies (ES). It is similar to a GA in that the system is initialized with a population of random solutions [43]. However, unlike a GA each population individual is also assigned a randomized velocity, in effect, flying them through the solution hyperspace. As is obvious, it is possible to simultaneously search for an optimum solution in multiple dimensions.

Also, since each particle keeps track of its coordinates in hyperspace which are associated with the best fitness it has achieved so far, as well as the overall best value obtained by any member of the population, PSO may be considered as having characteristics of ES.

4.1. *Fitness function*

An important problem in the implementation of EC techniques is the construction of a fitness function adequately epitomizing the nature of the problem. Michalewicz [44,45] puts it as "…the evaluation function serves as the only link between the problem and the algorithm".

Here we are faced with a non-linear objective function along with a set of inequality constraints that may also be highly non-linear (ANN maps). The presence of constraints in non-linear programming creates additional problems for finding the minimum. Some of the more important ones being [46]:

1. The constraints define an admissible region, which

must be non-empty for a meaningful optimization problem to exist.

2. A constrained minimization problem may have local minima even if the corresponding unconstrained problem does not have a local minimum.
3. None of the local minima may correspond to the global minimum of the corresponding unconstrained problem.

4.2. Exterior penalty function method

The penalty approach belongs to a class of indirect methods for solving a constrained non-linear programming problem via a sequence of one or more unconstrained minimization problems. This is based on a transformation of the above general problem with constraints to an unconstrained problem having the following general energy function, E (pseudo-cost function)

$$E(\vec{x},\kappa)=f(\vec{x})+P(\vec{g}(\vec{x})),\kappa) \tag{8}$$

where κ is a controlling parameter, P is a real-valued non-negative function called the penalty function, and f is the non linear objective function. The basic idea in the so-called exterior penalty function is to eliminate some or all of the constraints and to add to the objective function penalty terms, which prescribe a high cost to infeasible points [46].

This leads us to the choice of the penalty function in our case as follows:

$$P_{ig}(g_i(\mathbf{x}))=[\min\{0,g_i(\mathbf{x})\}]^2 \tag{9}$$

Here $g_i(\mathbf{x})$ are the inequality constraints to the optimization problem. Note that in the feasible space of solutions, the contribution from the penalty function is zero. This is because of incorporating the constant 0 (zero) in the argument of the "minimum of" function.

4.3. Application of PSO

Applying the PSO method consists of the following steps [40]:

1. Initialize an array of particles with random positions and velocities in 2 dimensions, feed rate and spindle speed. This constitutes Generation 0.
2. Evaluate the desired fitness function in the 2 variables.
3. Compare evaluation with particle's (personal) previous best value PBEST[i],if current value< PBEST[i], (i.e. it has achieved a new personal best) then PBEST[i]=current value and PBESTx[i][d]=current position in 2-dimensional hyperspace.
4. Compare evaluation with group's overall previous best (PBEST[GBEST]), If current value< PBEST[GBEST] then GBEST=particle′s array index (i.e. found a new best fitness for the population as a whole)
5. Compute particles' new velocity by using the following formula,

$$\begin{aligned}V[i][d]&=w\cdot V[i][d]\\&+\mathrm{ACC\_CONST}\mathrm{rand}(\ )\cdot(\mathrm{PBEST}x[i][d]\\&-\mathrm{Present}x[i][d])\\&+\mathrm{ACC\_CONST}\mathrm{rand}(\ )\cdot(\mathrm{PBEST}x[\mathrm{GBEST}][d]\\&-\mathrm{Present}x[i][d])\end{aligned} \tag{10}$$

6. Update particle's position by moving to Present$x$[i][d]+V[i][d],
7. Loop to step 2 until a criterion is met.

Note that in step 5, the particle retains a portion of the velocity that brought it to the current position. This is achieved by an inertia factor w. Recent studies [47,48] indicate the use of an adaptive inertia weight/constriction factor to insure convergence.

Further, the particle is given: (a) a velocity component toward the region where it achieved its personal best fitness and (b) another velocity component toward the location where the best fitness was achieved by any particle in the population as a whole.

These are proportional to the particle's current distance from the latter two, thus pushing it toward more lucrative feasible spaces to conduct the search.

The procedure is also illustrated in Fig. 3 with a flowchart. Here i refers to the particular particles index in the array.

4.4. Results

Fig. 4 depicts a typical particle swarm movement toward the optimum solution. Generation 0 shows the

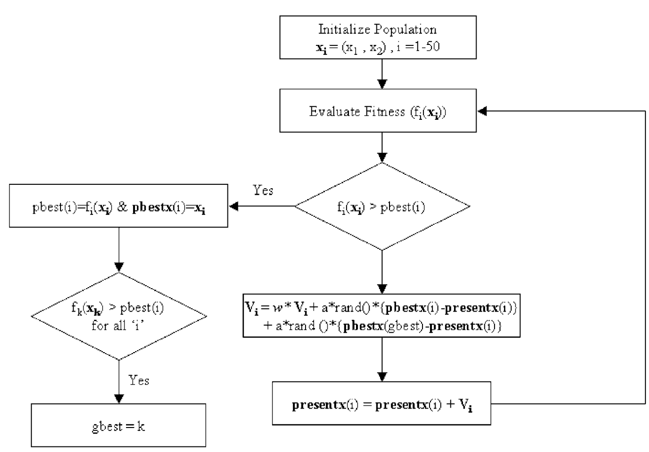
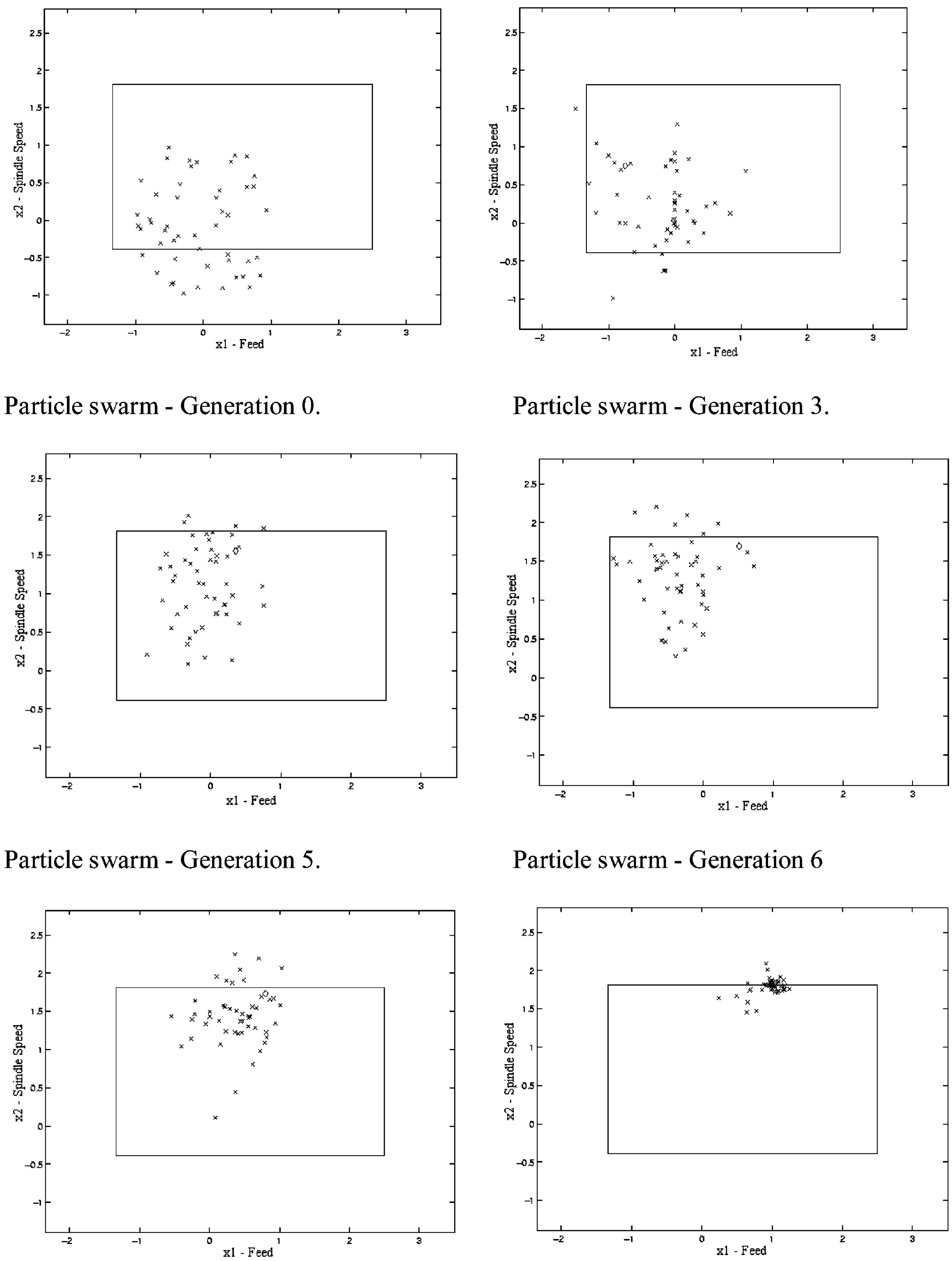


Fig. 3. PSO flowchart.



Particle swarm - Generation 0. Particle swarm - Generation 3.

Particle swarm - Generation 5. Particle swarm - Generation 6

Particle swarm - Generation 8. Particle swarm - Generation 15.

Fig. 4. PSO simulation.

random initialization of the particle's coordinates in the solution space. In subsequent generations, the swarm is tracked (×). Also, the best achieved by any population member thus far, is shown (◇). Also, the feasible space is graphed by the rectangle. An acceptable solution has to be found within this two-dimensional space. Note that the third constraint on force is also active and as such is not part of these illustrations.

To establish repeatability and robustness of the algorithm, the following machining conditions were presented for optimization: radial depth of cut=0.75 D, axial depth of cut=0.25 in., cutter diameter=0.25 in., number of flutes=2, rake angle=14°, primary clearance angle=16°.

Thus, the problem consists of minimizing the cost function under given constraints.

$$150 \leq N \leq 1500 \tag{11}$$

$$10 \leq f \leq 250 \tag{12}$$

$$F(f,N) \leq 300N \tag{13}$$

for the cost function, C, from Eq. (3),

$$C(\vec{x}) = \frac{5\times10^{-5}}{x_1} + 8.186\times10^{-20} x_2 x_1 \tag{14}$$

where

$$x_1 = \text{feed rate}\left(=\frac{1000\cdot Z}{\pi\cdot D}v\cdot S_z\right)$$

$$x_2 = \text{spindle speed}\left(=\frac{1000}{\pi\cdot D}\cdot v\right)$$

This problem is solved using the PSO algorithm implemented, and results are tabulated in Table 2. The run number corresponds to each time the program is run to find the optimum machining parameters. The best cost obtained, the corresponding feed, speed along with the force (constraint) are tabulated along with the number of generations it took to reach that optimum cost.

While the repeatability of finding the solution is established, we also find that the number of iterations/generations required to reach a reasonable solution is never greater than 30. Hence, the terminating criterion is set on a maximum number of iterations=35.

A sample of the evolution of the particle swarm is presented in Fig. 5. This optimization method affords a higher order (>2) of convergence, unlike traditional Newton/quasi-Newton methods of optimization. Further, various researchers empirically proved that the method is always successful in finding the global optimum.

Table 2
Repeatability of results over a number of runs

Run	Cost	Spindle speed	Feed	Force	Effective number of generations
1	4.094	1498	122.27	0.3	12
2	4.1	1495	121.93	0.3	15
3	4.099	1498	121.95	0.2997	18
4	4.135	1497	120.9	0.2985	26
5	4.086	1500	122.39	0.3	12
6	4.097	1498	122.03	0.2998	21
7	4.088	1499	122.29	0.3	6
8	4.097	1498	122.03	0.2997	11
9	4.095	1499	122.1	0.2997	23
10	4.09	1500	122.24	0.2999	20

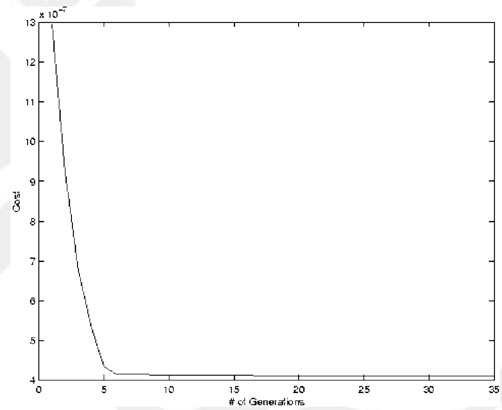


Fig. 5. Cost vs generations for run #7.

It is also worth noting that the solution obtained is for a specific cost function. This cost function may take a different form for different situations. It is easy to note the dependence of the nature of the cost function on M, C_t, α, β, Z and a number of other parameters. It is however, not of concern to consider dependence on the length and width of cut, since the optimization is being performed per unit length.

5. Test case

We shall now demonstrate the application of our research to a very common machining situation namely, pocket milling. We need to machine the workpiece so as to create the rectangular pocket on the top face seen in Fig. 6.

Four different immersion levels are found in the tool path. The geometric conditions are analyzed using the predictive model developed and the results are presented in Table 3.

Note that our machine tool has a small 0.2–0.4 second delay between different segments of the cutter path. From Table 3, a corresponding graph can be generated taking into account the above delay. This is presented in Fig. 7(a), which shows the simulated cutting forces using non-optimum cutting conditions. The above machining operation is conducted and the actual cutting force data acquired using the setup described above. This is also presented, for comparison, in Fig. 7(b). The differences in the two figures are attributed to the increase of immersion in cornering (neglected in simulation calculations) as well as the wear condition of the tool (which leads to an increase in cutting force). On the other hand, the PSO algorithm is used to optimize the cutting conditions and the resulting cutting forces are shown in Fig. 8(a) (simulated values) and Fig. 8(b) (measured values). The machining time is reduced by

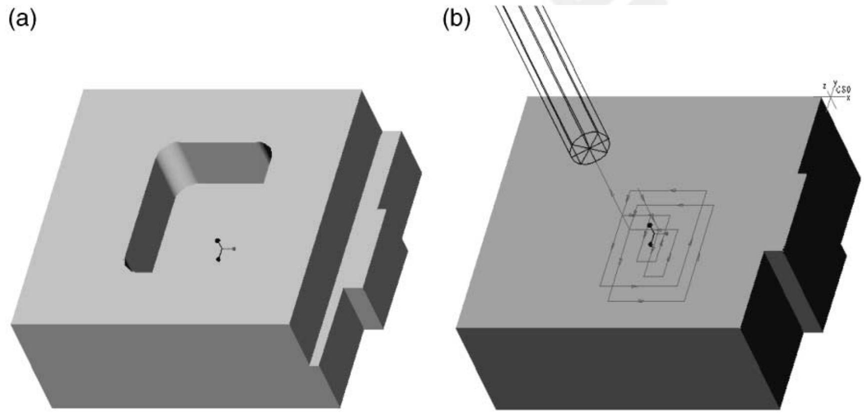


Fig. 6. Design model and tool path on workpiece.

Table 3
Various immersions and associated machining parameters (non-optimal)

Immersion	Feed/speed conditions	Max. force (N)
1	100/800	481
3/4	100/800	401
1/2	100/800	347
1/4	100/800	195.7

35% as a result of optimizing the feed and speed. Table 4 shows a comparison between the optimal cutting conditions and the non-optimal ones.

6. Conclusions and future research

This work has presented a new approach to optimizing the cutting conditions in end milling (feed and speed) subject to a near to comprehensive set of constraints. The original set of seventeen constraints was reduced to an equivalent set (of only three equations). Next, a production cost objective function was used to define the parameter to optimize (in this case, minimize). An algorithm for PSO was then developed and used to robustly and efficiently find the optimum cutting conditions. Both feed and speed were considered during optimization. The new technique has several advantages and benefits and is suitable for use with ANN based models where no explicit relation between inputs and outputs is available. This work opens the door for a new class of optimization techniques (i.e. EC based) in the area of machining.

The current implementation and testing of the new technique was limited in terms of the process, material, and number of process parameters and inputs considered. For example, the depth of cut was not one of the input parameters considered during the optimization. Optimizing the depth of cut requires the regeneration of the tool

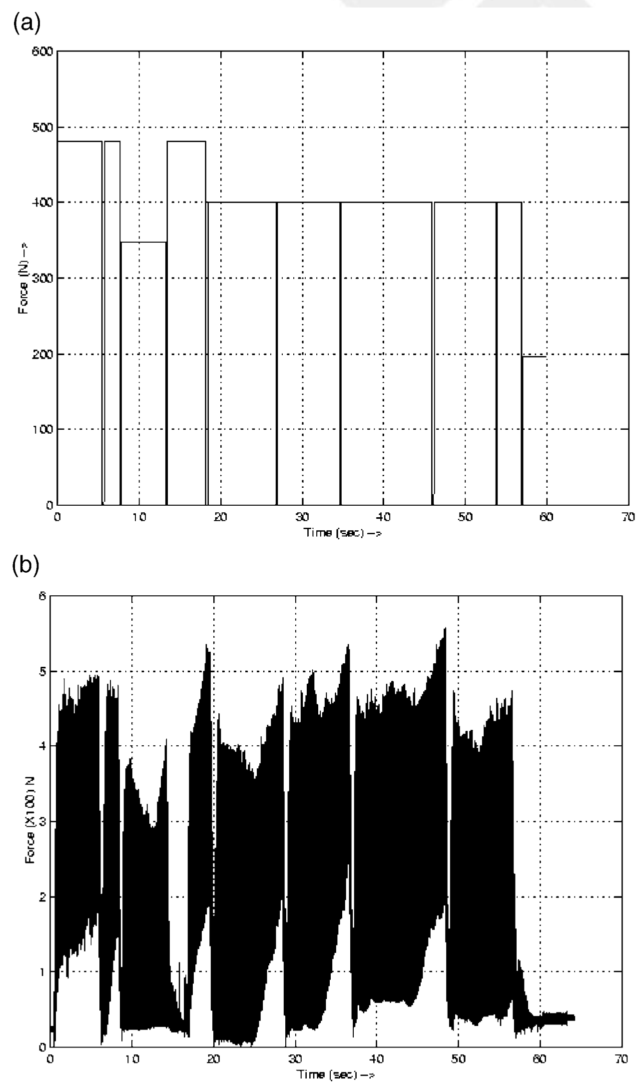


Fig. 7. Simulated vs predicted forces before optimization. (a) Simulated force variation; (b) experimentally acquired force variation.

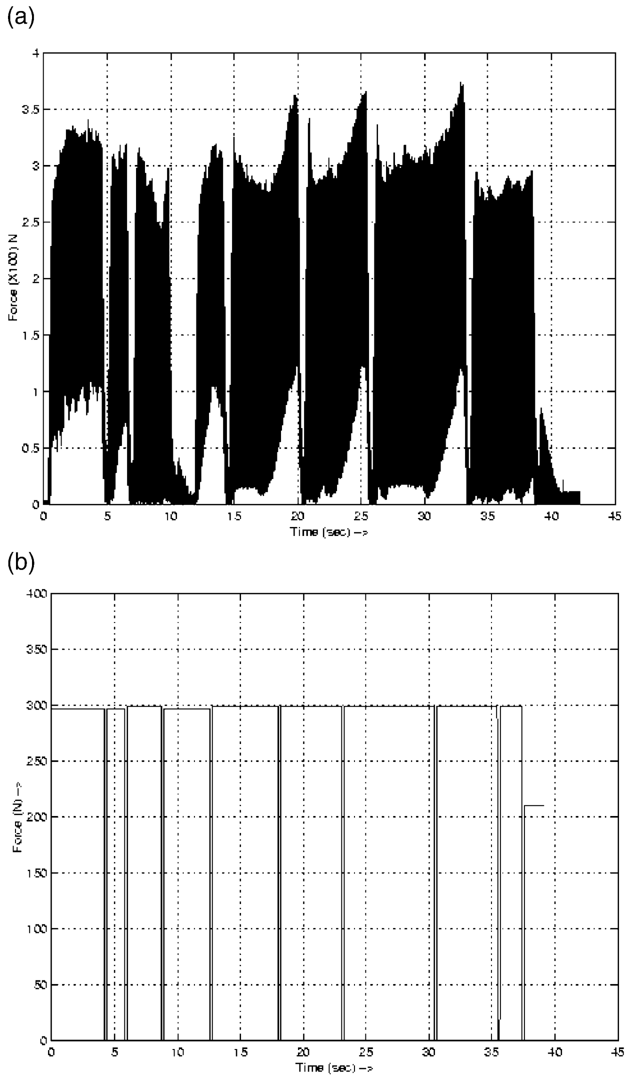


Fig. 8. Simulated vs predicted forces after optimization. (a) Measured forces; (b) predicted forces.

Table 4
Various immersions and associated machining parameters (comparison of non-optimal with optimal ones)

Immersion	Feed/speed conditions	Max. force (N)	Optimized feed/speed	Max. force
1	100/800	481	131.3/1499	296
3/4	100/800	401	157.8/1499	299
1/2	100/800	347	199.95/1487	299
1/4	100/800	195.7	199.95/1353	210

path and thus an integration of the optimizer to the CAM application. These additional requirements as well as the expansion and extension of the models and testing (to other tool shapes, material, etc.) shall be the subjects of future work.

References

[1] W.P. Wang, Solid modeling for optimizing metal removal of three-dimensional NC end milling, Journal of Manufacturing Systems 7 (1) (1988) 57–65.
[2] K.D. Bouzakis, K. Efstathiou, R. Paraskevopoulou, NC-code preparation with optimum cutting conditions in 3-axis milling, CIRP Annals 41 (1) (1992) 513–516.
[3] T. Bailey, Y. Ruget, A.D. Spence, M.A. Elbestawi, Open architecture controller for Die and Mold machining, Proceedings of the American Control Conference 1 (1995) 194–199.
[4] A.D. Spence, Y. Altintas, CAD assisted adaptive control for milling, ASME Journal of Dynamic Systems, Measurement and Control 113 (1991) 444–450.
[5] S. Takata, Generation of a machining scenario and its applications to intelligent machining operations, CIRP Annals 42 (1) (1993) 531–534.
[6] S.J. Park, Y.T. Lee, C.W. Yang, M. Yang, Determining the cutting conditions for sculptured surface machining, International Journal of Advanced Manufacturing Technology 8 (1993) 61–70.
[7] W.P. Wang, Application of solid modeling to automate machining parameters for complex parts, Manufacturing Systems 17 (1) (1988) 57–63.
[8] Z. Yazar, K.F. Koch, T. Merrick, T. Altan, Feed rate optimization based on cutting force calculations in 3-axis milling of dies and molds with sculptured surfaces, International Journal of Machine Tools & Manufacture 34 (3) (1994) 365–377.
[9] K. Yamazaki, N. Kojima, C. Sakamoto, Real-time model reference adaptive control of 3-D sculptured surface machining, CIRP Annals Manufacturing Technology 41st General Assembly of CIRP 40 (1) (1991) 479–482.
[10] B.M. Imani, M.A. Elbestawi, Model-based feed scheduling in die semi-finishing operation.
[11] B.K. Fussell, C. Ersoy, R.B. Jerard, Computer generated CNC machining feedrates, in: Proc. of the 1992 Japan–USA Symposium on Flexible Automation ASME, vol. 1, no. 1, 1992, pp. 377–384.
[12] K. Weinert, A. Enselmann, J. Friedhoff, Milling simulation for process optimization in the field of die and mould manufacturing, CIRP Annals 46 (1) (1997) 325–328.
[13] E.M. Lim, C.H. Menq, Integrated planning for precision machining of complex surfaces. Part 1: cutting-path and feed rate optimization, International Journal of Machine Tools and Manufacture 37 (1) (1997) 61–75.
[14] E.M. Lim, C.H. Menq, Integrated planning for precision machining of complex surfaces Part 2: application to the machining of turbine blade die, International Journal of Machine Tools and Manufacture 37 (1) (1997) 77–91.
[15] M. Tolouei-Rad, I.M. Bidhendi, On the optimization of machining parameters for milling operations, International Journal of Machine Tools and Manufacture 37 (1) (1997) 1–16.
[16] S. Takata, M.D. Tsai, Model based NC programming for end milling operations, Manufacturing Science and Engineering PED-v68-2 (1994) 809–818.
[17] E.J.A. Armarego, R.H. Brown, The Machining of Metals, Prentice Hall, 1969.
[18] E.M. Trent, Metal Cutting, Butterworths & Co, 1984, ISBN 0-408-10856-8.
[19] S.A. Lazaro, J. Zhang, L.A. Kendall, Knowledge-based approach for improvement of CNC part programs, Journal of Manufacturing Systems 13 (1) (1994) 20–30.
[20] I. Yellowey, E.A. Gunn, The optimal subdivision of cut in multi-pass machining operations, International Journal of Production Research 27 (9) (1989) 1573–1588.
[21] W.A. Kline, R.E. DeVor, J.R. Lindberg, The prediction of cutting forces in end milling with application to cornering cuts, Inter-

national Journal of Machine Tool Design Research 22 (1) (1982) 7–22.
[22] B.K. Fussell, K. Srinivasan, Adaptive control of force in end milling operations — an evaluation of available algorithms, Journal of Manufacturing Systems — SME 10 (1) (1991) 8–20.
[23] G. Yucesan, Y. Altintas, Prediction of ball end milling forces, Journal of Engineering for Industry 118 (1996) 95–103.
[24] Y. Altintas, A. Spence, End milling force algorithms for CAD systems, CIRP Annals 40 (1) (1991) 31–34.
[25] T. Bailey, Y. Ruget, A.D. Spence, M.A. Elbestawi, Open architecture controller for die and mold machining, Proceedings of the American Control Conference 1 (1995) 194–199.
[26] E.J.A. Armarego, A.J.R. Smith, J. Wang, Computer-aided constrained optimization analyses and strategies for multipass helical tooth milling operations, CIRP Annals 43 (1) (1994) 437–442.
[27] A.I. Sonmez, A. Baykasoglu, D. Turkay, I.H. Filiz, Dynamic optimization of multipass milling operations via geometric programming, International Journal of Machine Tools & Manufacture 39 (2) (1999) 297–320 (Feb.).
[28] P.G. Petropoulos, Optimal selection of machining rate variables by geometric programming, International Journal of Production Research 11 (4) (1973) 305–314.
[29] J.S. Agapiou, The optimization of single or multi-pass machining operation based on a combined criterion, Advances in Manufacturing Systems Engineering, ASME, (1989) 103–115.
[30] J.S. Agapiou, The optimization of machining operations based on a combined criterion, Part 1: The use of combined objectives in single pass operations, Journal of Engineering for Industry, Transactions of ASME 114 (1992) 500–507.
[31] J.S. Agapiou, The optimization of machining operations based on a combined criterion, Part 2: Multi pass operations, Journal of Engineering for Industry, Transactions of ASME 114 (1992) 508–513.
[32] E.J.A. Armarego, A.J.R. Smith, J. Wang, Constrained optimization strategies and CAM software for single-pass peripheral milling, International Journal of Production Research 31 (9) (1993) 2139–2160.
[33] T.C. Chang, R.A. Wysk, R.P. Davis, B. Choi, Milling parameter optimization through a discrete variable transformation, International Journal of Production Research 20 (4) (1982) 507–516.
[34] B.K. Lambert, A.G. Walvekar, Optimization of multi-pass machining operations, International Journal of Production Research 16 (4) (1978) 259–265.
[35] C. Machover, The CAD/CAM Handbook, McGrawHill, 1996 (ISBN 0-07-039375-3).
[36] V. Tandon, H. El-Mounayri, A novel artificial neural networks force model for end milling, International Journal of Advanced Manufacturing Technology (IJAMT), submitted.
[37] N.K. Jha, K. Hornik, Integrated computer-aided optimal design and finite element analysis of a plain milling cutter, Applied Mathematical Modeling 19 (1994) 343–353.
[38] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43.
[39] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE Int. Conference on Neural Networks, Perth, Australia, vol. 4, IEEE Service Center, Piscataway, NJ, pp. 1942–1948.
[40] R.C. Eberhart, P. Simpson, R. Dobbins, Computational Intelligence PC Tools, AP Professional, 1996 (ISBN 0-12-228630-8).
[41] J. Kennedy, The behavior of particles, in: Proceedings of the 7th ICEC, 1998, pp. 581–589.
[42] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, in: Proceedings of the 7th ICEC, 1998, pp. 601–610.
[43] R.C. Eberhart, Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in: Proceedings of the 7th ICEC, 1998, pp. 611–616.
[44] Z. Michalewicz, A survey of constraint handling techniques in evolutionary computation methods, in: Proceedings of the 4th Annual Conference on Evolutionary Programming, San Diego, CA, 1995, pp. 135–155.
[45] M. Schoenauer, Z. Michalewicz, Sphere operators and their applicability for constrained parameter optimization problems, in: Proceedings of the 7th Annual Conference on Evolutionary Programming, San Diego, CA, 1998, pp. 241–250.
[46] A. Cichocki, R. Unbehauen, Neural Networks for Optimization and Signal Processing, John Wiley & Sons, 1993 (ISBN 0 471 93010 5).
[47] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Proceedings ICEC, Washington, DC, 1999, pp. 1951–1957.
[48] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in: Proceedings of the 7th ICEC, 1998, pp. 591–600.