Strojniški vestnik - Journal of Mechanical Engineering 53(2007)6, 354-368
UDK - UDC 621.914:004.89
Izvirni znanstveni članek - Original scientific paper (1.01)

Optimiranje značilnih parametrov frezanja z uporabo razvojne tehnike optimizacije jate delcev

Optimization of the Characteristic Parameters in Milling Using the PSO Evolution Technique

Uroš Župerl[1] - Franci Čuš[1] - Valentina Gecevska[2]
([1]Fakulteta za strojništvo, Maribor; Fakulteta za strojništvo v Skopju, Makedonija)

Izbira rezalnih parametrov je najpomembnejši korak pri postopku načrtovanja proizvodnje, zato izdelamo novo tehniko razvojnega računanja za optimiranje procesa odrezovanja. V prispevku je uporabljena tehnika, ki oponaša dinamiko delcev v velikih skupinah (optimizacija jate delcev - OJD), za učinkovito in simultano optimiranje postopkov frezanja. V omenjenih postopkih smo soočeni s problemom več ciljnih dejavnikov. Najprej uporabimo umetno nevronsko mrežo (UNM) za napovedovanje rezalnih sil, nato z algoritmom OJD pridobimo optimalno rezalno hitrost in podajanja. Cilj optimizacije je, ob upoštevanju omejitev, določiti ekstrem ciljne funkcije (napovedna površina največjih sil). Med optimizacijo se delci inteligentno "gibljejo" v prostoru rešitev in "iščejo" optimalne rezalne pogoje po strategiji algoritma OJD. Rezultati pokažejo, da je integriran sistem nevronskih mrež in spoznavanja jate učinkovita metoda pri reševanju večciljnih optimizacijskih problemov. Njena velika učinkovitost na širokem območju rezalnih parametrov potrjuje, da sistem lahko praktično uporabimo v proizvodnji. Rezultati simulacij nakazujejo, da predlagan algoritem v primerjavi z rodovnimi algoritmi (GA) in simuliranim (SA) žarjenjem (popuščanjem) lahko poveča natančnost rešitve in konvergenco postopka. Nova tehnika razvojnega računanja ima nekoliko prednosti ter koristi in je primerna za uporabo v kombinaciji z modeli na osnovi umetnih nevronskih vezij, pri katerih niso na voljo izrecne relacije med vstopnimi in izstopnimi veličinami. Raziskava odpira vrata na področju obdelave z odrezovanjem za nov razred optimizacijskih tehnik, ki slonijo na razvojnem računanju.

(Ključne besede: odrezovanje, končno frezanje, rezalni parametri, nevronske mreže)

The selection of machining parameters is an important step in process planning; therefore, a new evolutionary computation technique has been developed to optimize the machining process. In this paper, Particle Swarm Optimization (PSO) is used to efficiently optimize the machining parameters simultaneously in milling processes where multiple conflicting objectives are present. First, an artificial neural network (ANN) predictive model is used to predict the cutting forces during machining and then the PSO algorithm is used to obtain the optimum cutting speeds and feed rates. The goal of the optimization is to determine the objective function maximum (the predicted cutting-force surface) by considering the cutting constraints. During optimization the particles 'fly' intelligently in the solution space and search for optimal cutting conditions according to the strategies of the PSO algorithm. The results showed that an integrated system of neural networks and swarm intelligence is an effective method for solving multi-objective optimization problems. The high accuracy of the results within a wide range of machining parameters indicates that the system can be practically applied in industry. The simulation results show that compared with genetic algorithms (GAs) and simulated annealing (SA) the proposed algorithm can improve the quality of the solution while speeding up the convergence process. The new computational technique has several advantages and benefits and is suitable for use when combined with ANN-based models where no explicit relation between the inputs and the outputs is available. This research opens the door for a new class of optimization techniques that are based on evolution computation in the area of machining.

(Keywords: cutting, end-milling, cutting parameters, neural networks)

0 UVOD

Povečevanje storilnosti, zmanjševanje stroškov in sočasno ohranjanje kakovosti izdelka so glavni izzivi, s katerimi se srečujejo proizvajalci. Pravilna izbira rezalnih parametrov je pomemben korak k zadoščanju teh ciljev in ohranjanju konkurenčne prednosti na trgu [1]. Mnogo raziskovalcev se je ukvarjalo s proučevanjem učinkov optimalne izbire rezalnih razmer pri frezanju [2]. Problem optimiranja frezanja se lahko oblikuje in reši kot večciljni optimizacijski problem [3]. V praksi učinkovita izbira frezalnih parametrov zahteva sočasno upoštevanje več ciljnih dejavnikov, to so: največja obstojnost orodja, zahtevana hrapavost obdelane površine, ciljna storilnost opravil, stopnja odvzemanja materiala itn. [4]. V nekaterih primerih nastavitve parametrov, ki so optimalne za eno definirano ciljno funkcijo, morda ne bodo posebej primerne za drugo ciljno funkcijo. Reševanje večciljnih problemov z običajnimi optimizacijskimi metodami je težko. Edini način rešitve je zmanjšati niz ciljnih dejavnikov v en nadomestni ciljni dejavnik in ga potemtakem obdelati. Zatorej so razvojni algoritmi, kakršni so genetski algoritmi (GA) in ORD, bolj primerni in pogosteje uporabljeni pri reševanju večciljnih optimizacijskih problemov. Te metode so podane v delu [5]. ORD je učinkovita alternativa drugim naključnim in na populacijah temelječim iskalnim algoritmom, še posebej ko imamo opravka z večciljnimi optimizacijskimi problemi. Njena izvedba je razmeroma preprosta, potrebno je manj nastavljanja parametrov v primerjavi z genetskimi algoritmi.

V naši raziskavi so nevronske mreže uporabljene za modeliranje zapletenih razmerij v procesu. Integriran sistem nevronskih mrež in optimizatorja roja delcev je nato uporabljen pri reševanju večciljnega problema pri opravilih frezanja (sl. 1).

1 OPTIMIZACIJA PSO

Optimizacija, ki temelji na dinamiki jate delcev (OJD), je razmeroma nova tehnika za optimiranje nelinearnih funkcij [6]. Prvič je bila predstavljena leta 1995 [7]. Jim Kennedy jo je odkril med simuliranjem poenostavljenega socialnega modela, ki oponaša skladno in nepredvidljivo gibanje jate ptic [8]. Reynolds z izdelanim modelom jate s preprostimi pravili vzbuja njeno zapleteno gibanje [9]. Takšne raziskave poimenujemo "spoznavanje jate".

0 INTRODUCTION

Increasing productivity, decreasing costs, and maintaining high product quality at the same time are the main challenges faced by manufacturers today. The proper selection of machining parameters is an important step towards meeting these goals and thus gaining a competitive advantage in the market [1]. Many researchers have studied the effects of the optimal selection of machining parameters on end milling [2]. This problem can be formulated and solved as a multiple objective optimization problem [3]. In practice, the efficient selection of milling parameters requires the simultaneous consideration of multiple objectives, including maximum tool-life, desired roughness of the machined surface, target operation productivity, metal removal rate, etc [4]. In some instances, parameter settings that are optimal for one defined objective function may not be particularly suited to another objective function. Solving multi-objective problems with traditional optimization methods is difficult and the only way is to reduce the set of objectives into a single objective and handle it accordingly. Therefore evolutionary algorithms such as genetic algorithms (GAs) and particle swarm optimization (PSO) are more convenient and usually utilized in multi-objective optimization problems. These methods are summarized in [5]. The PSO is an efficient alternative over other stochastic and population-based search algorithms, especially when dealing with multi-objective optimization problems. It is relatively easy to implement and has fewer parameters to adjust compared to genetic algorithms.

In our research neural networks are used to model complex relationships in the process, and an integrated system of neural networks and a particle swarm optimizer are utilized in solving multi-objective problems observed in milling operations (Fig. 1).

1 PSO OPTIMIZATION

Particle swarm optimization (PSO) is a relatively new technique for the optimization of continuous non-linear functions [6]. It was first presented in 1995 [7]. Jim Kennedy discovered the method through the simulation of a simplified model, i.e., the graceful but unpredictable movement of a bird swarm [8]. Reynolds developed a swarm model with simple rules and generated complicated swarm behavior [9]. These researches are called "Swarm Intelligence"

OJD je zelo preprosta zasnova, vzorce gibanja ponazorimo le z nekaj vrsticami računalniškega zapisa. Metoda uporablja le osnovne matematične operatorje, zato je računsko nezahtevna glede na hitrost in zasedenost spomina. OJD je bila prepoznana kot tehnika razvojnega računanja [10]. Ima lastnosti tako genetskih algoritmov (GA) kakor tudi razvojnih strategij (RS). Preostale tehnike razvojnega računanja (RR), npr.: genetski algoritmi prav tako uporabljajo več iskalnih točk v prostoru rešitev. Z GA ima to podobnost, da se sistem pri obeh začne s populacijo naključnih rešitev.

Medtem ko z GA lahko rešujemo združevalne optimizacijske probleme, lahko s OJD rešujemo neprekinjene optimizacijske probleme. Vendar se v nasprotju z GA vsakemu osebku v populaciji priredi naključna hitrost, s katero leti skozi evklidski prostor rešitev. OJD so nadgradili tudi za reševanje združevalnih optimizacijskih problemov. Očitno je, da je mogoče sočasno iskanje optimuma v več razsežnostih. OJD lahko izvedemo z majhnim programom v nasprotju z drugimi tehnikami RR. Živa bitja se včasih združujejo in gibajo v rojih, jatah. Eden izmed glavnih ciljev raziskovalcev umetnega življenja je ugotoviti, kako se živa bitja obnašajo v rojih in kako modelirati njihovo obnašanje na računalniku.

OJD ima dve preprosti zasnovi. Z nekaj preprostimi pravili lahko modeliramo obnašanje jate delcev. Čeprav so pravila obnašanja posameznega delca v jati preprosta, je lahko dinamika celotne jate zelo zapletena. Gibanje delca v jati ponazorimo s preprostimi vektorji. Ta značilnost je prva osnova OJD.

Po Boydovi raziskavi [11] ljudje pri odločanju upoštevajo dva pomembna tipa informacij: lastne izkušnje in izkušnje drugih ljudi. Prvi tip so lastne izkušnje, iz katerih vedo, katere odločitve so bile v preteklosti uspešne. Vsaka oseba se torej odloča na podlagi lastnih izkušenj in izkušenj drugih ljudi. Ta značilnost je druga osnova tehnike OJD.

Uporabe OJD najdemo v: učnih algoritmih nevronskih mrež [12], izdelavi pravil v mehkih nevronskih mrežah [13], optimiranju računalniško vodenega frezanja [14], nadzoru električne moči in napetosti [15]. Uporab OJD na drugih področjih je malo, vendar pričakujemo njihov porast. Večina člankov obravnava samo metodo, njene spremembe in primerja njene zmogljivosti s preostalimi metodami RR ([14] in [15]).

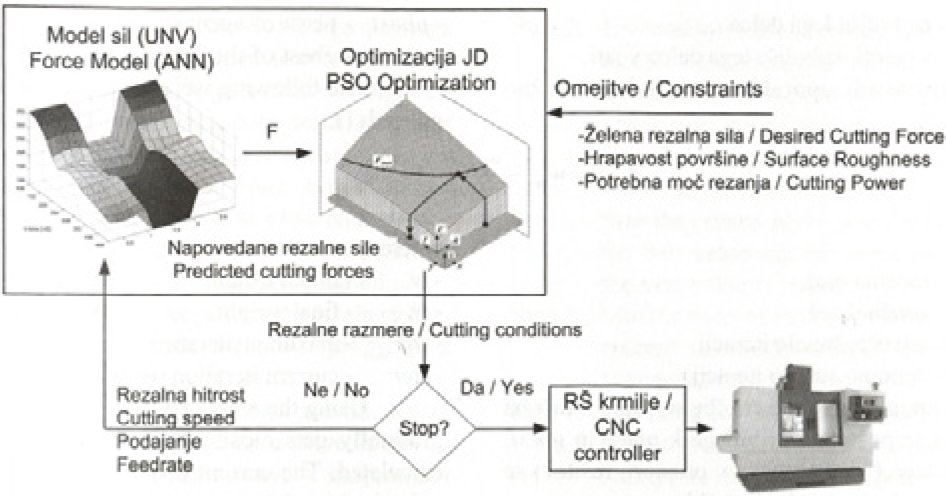
PSO is a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, so is computationally inexpensive in terms of both memory requirements and speed. PSO has been recognized as an evolutionary computation technique [10] and has features of both genetic algorithms (GAs) and evolution strategies (ESs). Other evolutionary computation (EC) techniques such as genetic algorithms also utilize some searching points in the solution space. It is similar to a GA in that the system is initialized with a population of random solutions.

While GAs can handle combinatorial optimization problems, PSO can handle continuous optimization problems. However, unlike a GA each population individual is also assigned a randomized velocity, in effect, flying them through the solution hyperspace. PSO has been expanded to also handle combinatorial optimization problems. As is obvious, it is possible to simultaneously search for an optimum solution in multiple dimensions. Unlike other EC techniques, PSO can be realized with only a small program. Natural creatures sometimes behave as a swarm. One of the main goals of artificial life researches is to examine how natural creatures behave as a swarm and reconfigure the swarm models inside a computer.

PSO has two simple concepts. The swarm behaviour can be modelled with a few simple rules. Even if the behaviour rules of each individual (particle) are simple, the behaviour of the swarm can be very complex. The behaviour of each agent inside the swarm can be modelled with simple vectors. This characteristic is the basic concept of PSO.

According to Boyd's examination [11], people utilize two important kinds of information in the decision process. The first one is their own experience; they have tried the choices and know which state has been better so far, and they know how good it was. Therefore, each person makes his or her decision using his or her own experiences and other peoples' experiences. This characteristic is another basic concept of PSO.

The applications of PSO are as follows: neural network learning algorithms [12], rule extraction in fuzzy neural networks [13], computer-controlled milling optimization [14], as well as power and voltage control [15]. The application of PSO in other fields is at the early stage, and more applications can be expected. Most papers are related to the method itself, and its modification and comparison with other EC methods ([14] and [15]).



Sl. 1. *Optimizacijska shema, ki temelji na metodi OJD in nevronski mreži*
Fig. 1. PSO-based neural network optimization scheme

2 OSNOVE OPTIMIZACIJE JD

OJD je izdelana na osnovi simulacij gibanja jate ptic v dvorazsežnem prostoru. Lega vsakega delca je določena s koordinato *XY*. Hitrost delca je izražena z *vx* (hitrost v smeri osi *X*) in *vy* (hitrost v smeri osi *Y*). Spremembe lege delca je izvedena na podlagi informacije o legi in hitrosti. Jata ptic optimira določeno ciljno funkcijo. Vsak delec pozna svojo najboljšo vrednost do zdaj (*pbest*) in svojo *XY* lego. V analogiji predstavlja ta informacija osebne izkušnje posameznega delca.

Nadalje, vsak delec pozna najboljšo vrednost v skupini (*gbest* med *pbest*). Ta informacija pomeni uspešnost preostalih delcev. Vsak delec skuša spremeniti svojo lego na temelju naslednjih informacij: trenutne lege (*x, y*), trenutne hitrosti (*vx, vy*), razdalje med trenutno lego in mestom *pbest*, razdalje med trenutno lego in mestom *gbest*. To modifikacijo lahko ponazorimo z zasnovo hitrosti.

Hitrost delca se lahko spreminja po naslednji enačbi:

2 BASICS OF PSO OPTIMIZATION

PSO is developed through the simulation of bird flocking in two-dimensional space. The position of each agent is represented by the *XY* axis position and the velocity is expressed by *vx* (the velocity in the *X* axis) and *vy* (the velocity in the *Y* axis). The modification of the agent position is realized using the position and velocity information. Bird flocking optimizes a certain objective function. Each agent knows its best value so far (*pbest*) and its *XY* position. This information is an analogy with the personal experiences of each agent.

Furthermore, each agent knows the best value so far in the group (*gbest*) among the best values (*pbests*). This information is an analogy of the knowledge of how the other agents around them have performed. Each agent tries to modify its position using the following information: the current positions (*x, y*), the current velocities (*vx, vy*), the distance between the current position and (*pbest*), and the distance between the current position and (*gbest*). This modification can be represented by the concept of velocity.

The velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot rand_1 \cdot (pbest_i - s_i^k) + c_2 rand_2 \cdot (gbest - s_i^k) \qquad (1),$$

kjer so:
- v_i^k - hitrost delca *i* v iteraciji *k*,
- w - utežna funkcija,
- c_j - utežni faktor,
- *rand* - naključno število med 0 in 1,
- s_i^k - trenutna lega delca *i* v iteraciji *k*,

where:
- v_i^k - velocity of agent *i* at iteration *k*,
- w - weighting function,
- c_j - weighting factor,
- *rand* - random number between 0 and 1,
- s_i^k - current position of agent *i* at iteration *k*,

- *$pbest_i$* - najboljša lega delca *i*,
- *gbest* - v celoti najboljša lega delca v jati.

Po navadi uporabimo naslednjo utežno funkcijo (1):

- *$pbest_i$* - pbest of agent *i*,
- *gbest* - gbest of the group.

The following weighting function is usually utilized (1):

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter \quad (2),$$

kjer so:

- w_{max} - začetna utež,
- w_{min} - končna utež,
- $iter_{max}$ - največje število iteracij,
- $iter$ - trenutno število iteracij.

Z uporabo zgornje enačbe se lahko izračuna hitrost, ki se postopno približuje k *pbest* in *gbest*. Trenutna lega (iskalna točka v prostoru rešitev) se lahko popravi po naslednji enačbi:

where:

- w_{max} - initial weight,
- w_{min} - final weight,
- $iter_{max}$ - maximum iteration number,
- $iter$ - current iteration number.

Using the above equation, a velocity, which gradually gets close to *pbest* and *gbest*, can be calculated. The current position (searching for the point in the solution space) can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \quad (3).$$

Na sliki 2 je prikazana zasnova spreminjanja iskalne točke z algoritmom OJD. Na sliki 3 je prikazana zasnova iskanja z delci v prostoru rešitev. Vsak delec spreminja svojo trenutno lego po postopku seštevanja vektorjev, ki so podani na sliki 2.

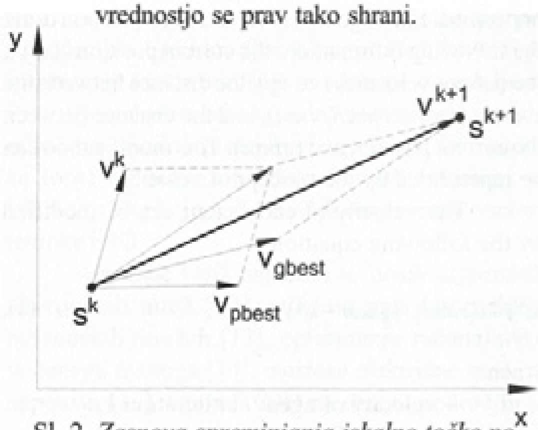
Splošen diagram poteka metode OJD je podan z naslednjimi koraki:

Korak 1: Ustvarjanje začetnih pogojev vsakega posameznega delca. Začetne iskalne točke (s_i^0) in hitrosti (v_i^0) vsakega posameznega delca se določijo naključno znotraj dovoljenih mej. Trenutna iskalna točka vsakega delca se shrani v *pbest*. Najboljša vrednost *pbest* se shrani v *gbest*. Številka delca z najboljšo vrednostjo se prav tako shrani.

Figure 2 shows the concept of modifying a searching point using the PSO algorithm. Figure 3 shows a searching concept with agents in a solution space. Each agent changes its current position using the integration of vectors, as shown in Figure 2.

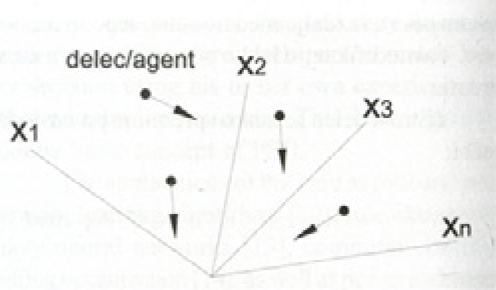
The general flow chart of the PSO method can be described as follows:

Step 1: Generation of the initial condition of each agent. The initial searching points (s_i^0) and velocities (v_i^0) of each agent are generated randomly within the allowable range. The current searching point is set to *pbest* for each agent. The best-evaluated value of *pbest* is set to *gbest* and the agent number with the best value is stored.



Sl. 2. *Zasnova spreminjanja iskalne točke po algoritmu OJD*

Fig. 2. *Concept of modifying a searching point according to a PSO algorithm*



Sl. 3. *Z delci ponazorjeno načelo iskanja v prostoru rešitev*

Fig. 3. *Concept of searching with agents in a solution space*

Korak 2: Ocenjevanje uspešnosti iskanja vsakega posameznega delca. Izračun ciljne vrednosti za vsak delec. Če je vrednost boljša od trenutne vrednosti *pbest* delca, se *pbest* nadomesti z novo vrednostjo. Če je najboljša vrednost *pbest* boljša kakor trenutna *gbest*, potem se *gbest* nadomesti z najboljšo vrednostjo. Števiika delca se skupaj z najboljšo vrednostjo shrani.

Korak 3: Sprememba vsake posamezne iskalne točke. Trenutna iskalna točka se spreminja z uporabo enačb (1), (2) in (3).

Korak 4: Preverjanje ustavitvenega pravila. Ko je doseženo vnaprej določeno največje število iteracij, se algoritem ustavi. V nasprotnem primeru se preide na korak 2.

Na sliki 4 je prikazan splošen algoritem strategije OJD.

Lastnosti postopka OJD lahko povzamemo v naslednjih točkah:

1. Iz enačb (1), (2) in (3) je razvidno, da lahko z OJD rešujemo zvezne optimizacijske probleme.
2. PSO uporablja več iskalnih točk, tako kakor genetski algoritem (GA). Iskalne točke se z

Step 2: Evaluation of the searching point of each agent. The objective function value is calculated for each agent. If the value is better than the current *pbest* of the agent, the *pbest* value is replaced by the current value. If the best value of *pbest* is better than the current *gbest*, *gbest* is replaced by the best value and the agent number with the best value is stored.

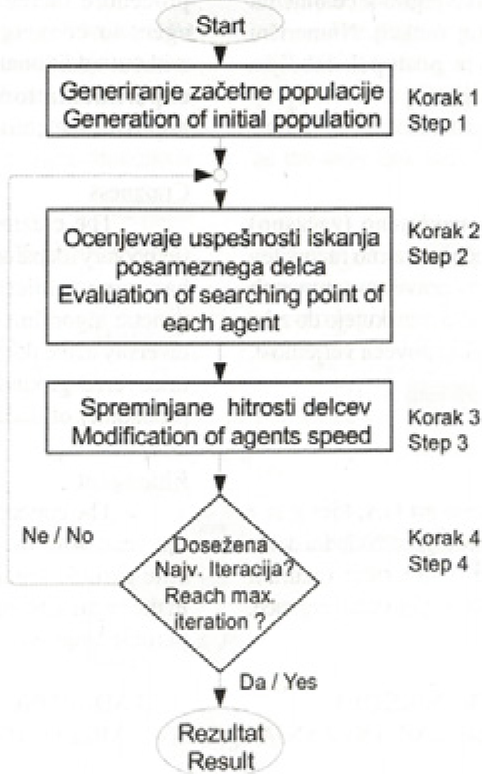
Step 3: Modification of each searching point. The current searching point of each agent is changed using (1), (2) and (3).

Step 4: Checking the exit condition. The current iteration number reaches the predetermined maximum iteration number, then exits. Otherwise, go to step 2.

Fig. 4 shows the general flow chart of the PSO strategy.

The features of the PSO procedure can be summarized as follows:

1. As shown in (1), (2) and (3), the PSO can handle continuous optimization problems.
2. The PSO utilizes several searching points, like a genetic algorithm (GA), and the searching points



Sl. 4. *Splošen algoritem OJD*

Fig. 4. *A general PSO algorithm*

uporabo *pbest* in *gbest* počasi približujejo optimalni točki.
3. Zgornja zasnova je razložena le na oseh *XY* (dvorazsežni prostor). Vendar lahko metodo z lahkoto uporabimo na *n*-razsežnem prostoru.

Z namenom izboljšati stopnjo konvergence algoritma OJD so raziskovalci ([8] in [9]) predlagali spremembe sedanje OJD. Te spremembe se nanašajo na uporabo najboljše lege, največjega skoka hitrosti, vztrajnosti, izletavanja, izbranega delca in izbrane hitrosti.

Največja hitrost

Na osnovi numeričnih preizkusov izberemo začetno vrednost $v^0_{max}=100$, nato to vrednost zmanjšamo za delež ψ. Numerični preizkusi dajejo slutiti, da ta postopek izboljša stopnjo konvergence algoritma:

Najboljša lega

Pomeni, da do zdaj najboljša lega v jati nadomesti najboljšo lego jate. S postopkom se poveča pritisk na delec, da konvergira k celotnemu optimumu, brez preračunavanj funkcij. Numerični preizkusi dajejo slutiti, da ta postopek izboljša stopnjo konvergence algoritma.

Izletavanje

Ta dejavnik oponaša naključno (začasno) izletavanje ptic iz jate, ki povečuje smerno razpršitev v jati in ima nekaj podobnosti z opravilom sprememb v genetskem algoritmu. Ptice tako preiskujejo do zdaj neznano področje, kar na splošno poveča verjetnost, da se najde optimum.

Izbrani delec

Ta zasnova je izposojena pri GA, kjer gen z najboljšo prilagoditvijo nikoli ne izgine. Skladni delec nadomesti najslabšega v jati. Numerični rezultati kažejo, da skladni delec izboljša stopnjo konvergence.

3 PRILAGODITEV TEHNIKE OJD OPTIMIZACIJSKEMU PROBLEMU FREZANJA

Z namenom iskanja optimalnih rezalnih parametrov integriramo nevronski model rezalnih

gradually get close to the optimal point using their pbests and the gbest.
3. The above concept is explained using only the XY-axes (two-dimensional space). However, the method can be easily applied to an n-dimensional problem.

With the objective to improve the rate of convergence of the PSO algorithm, researchers ([8] and [9]) proposed some modifications to the existing PSO. These modifications relate to the use of the best ever position, the maximum velocity, the inertia, the craziness, the elite particle and the elite velocity.

Maximum velocity

Based on a numerical experimentation, we select a starting value $v^0_{max}=100$ and then decrease this value by the fraction ψ. The numerical experimentation suggests that this approach improves the convergence rate of the algorithm:

$$v^{k+1}_{max}=\psi\cdot v^{k}_{max};\quad 0\le\psi\le1 \tag{4}$$

Best ever position

This means that the best ever position in the swarm replaces the best position of the swarm. This procedure increases the pressure exerted on the agent to converge towards the global optimum without additional function evaluations. Numerical experimentation suggests that this approach improves the convergence rate of the algorithm.

Craziness

The craziness operator mimics the random (temporary) departure of birds from the flock. Craziness has some similarity to the mutation operator in a genetic algorithm, since it increases the directional diversity in the flock. "Crazy" birds explore previously uncovered ground, which in general increases the probability of finding the optimum.

Elite agent

The concept is borrowed from the GA where the gene with the best fitness never vanishes. The elite particle replaces the worst positioned particle in the swarm. Numerical results indicate that the elite particle improves convergence rates.

3 ADAPTING THE PSO TECHNIQUE TO A MILLING OPTIMIZATION PROBLEM

In order to search for optimal process parameters, the neural network model of cutting force sil z optimizatorjem OJD. Zgradba sistema je prikazana na sliki 1. Skupek nevronskih modelov je razvrščen v splošni nevronski model, njegovi izhodi so posredovani večciljnemu optimizatorju jate delcev, kjer so definirane ciljne funkcije in omejitve. Algoritem OJD je začet z naključno ustvarjenimi delci, ki so kandidati za optimalno rešitev. Model nevronske mreže napove rezalne sile za vsak delec. Napovedane sile se uporabijo pri izračunu ciljne funkcije, ki jo želi OJD najbolj zvečati.

Postopek optimiranja poteka v dveh fazah. V prvi fazi nevronski model rezalnih sil ustvari v odvisnosti od priporočenih rezalnih pogojev 3-D površino rezalne sile, ki predstavlja prostor mogočih rešitev za algoritem OJD. Ta površina je omejena z ravninami, ki ponazarjajo omejitve postopka odrezavanja. Med postopkom optimiranja frezanja upoštevamo 7 omejitev, ki izvirajo iz tehnoloških določil. Podane so v preglednici 1. Soočimo se z nelinearno ciljno funkcijo skupaj z nizom omejitvenih neenačb, ki so prav tako lahko močno nelinearne. Prisotnost nelinearnosti povzroča dodatne težave pri iskanju optimuma.

Največji problem pri uvajanju tehnike OJD je izpeljava ciljne funkcije, ki ponazarja naravo optimizacijskega problema. Ciljna funkcija se uporabi kot edina povezava med optimizacijskim problemom in algoritmom OJD. Za ciljno funkcijo je

was integrated with a particle swarm optimizer. The architecture of the system is shown in Figure 1. Multiple neural network models are grouped together under the general neural network model, and its output is fed into the multi-objective particle swarm optimizer, where the objective functions and constraints are defined. The PSO algorithm is initiated with randomly generated particles that are optimum solution candidates. The neural network model predicts the cutting forces for each of the particles. The predicted forces are used in the calculation of an objective function in which the PSO tries to maximize.

The optimization process executes in two phases. In first phase, the neural prediction model, on the basis of the recommended cutting conditions, generates the 3D surface of the cutting forces, which represent the feasible solution space for the PSO algorithm. The cutting force surface is limited with planes that represent the constraints of the cutting process. Seven constraints, which arise from technological specifications, are considered during the optimization process. Those constraints are listed in Table 1. Here we are faced with a non-linear objective function along with a set of inequality constraints that may also be highly non-linear. The presence of non-linearities creates additional problems for finding the minimum.

The biggest problem in the implementation of the PSO technique is the construction of a fitness (objective) function that adequately epitomizes the nature of the problem. The objective function serves as the only link between the optimization problem

Preglednica 1. *Uporabljene omejitve z neenačbami*
Table 1. *Used constraints and their expressions*

Constraints	Izraz / Expression	Spremenljivke / Variables
podajanje feedrate	$f_{\min} \le \frac{1000 \cdot z}{\pi \cdot D} v_c \cdot f_z \le f_{\max}$	z – število zob / number of teeth, f_z – podajanje na zob / feeding per tooth, D – premer frezala / diameter of cutter
vrtilna frekvenca spindle speed	$n_{\min} \le \frac{1000}{\pi \cdot D} v_c \le n_{\max}$	v_c – rezalna hitrost / cutting speed
prečna globina rezanja radial depth of cut	$R_D \le ae_{\max}$	ae_{max} – najv. radialna globina reza / max. radial depth of cutting
vzdolžna globina rezanja axial depth of cut	$A_D \le ap_{\max}$	ap_{max} – najv. aksialna globina rezanja / max. axial depth of cutting
moč rezanja power of cutting	$\frac{MRR \cdot Kc}{60} \le P_{dov}$	MRR – stopnja odrezavanja materiala / metal removal rate, Kc – specifična rezalna sila / specific cutting force
rezalna sila cutting force	$F(f,n) \le F_{ref}$	F_{ref} – želena rezalna sila / desired cutting force
hrapavost površine surface roughness	$R_a \le R_{a\,ref}$	$R_{a\,ref}$ - želena hrapavost površine / desired surface roughness

izbrana z UNV ustvarjena površina največje rezalne sile.

V drugi fazi algoritem OJD ustvari jato delcev na površini rezalne sile. Jata delcev potuje po površini in išče skrajnost rezalne sile. Koordinate tistega delca, ki je našel največjo rezalno silo, predstavljajo optimalne rezalne pogoje (v, n). Naloga algoritma OJD je s prilagajanjem podajanja in vrtljajev ohranjati največjo rezalno silo na ravni referenčne vrednosti. Na sliki 5 je prikazan diagram postopka OJD za optimiranje postopka frezanja.

Potek optimizacije je podan z naslednjimi koraki:

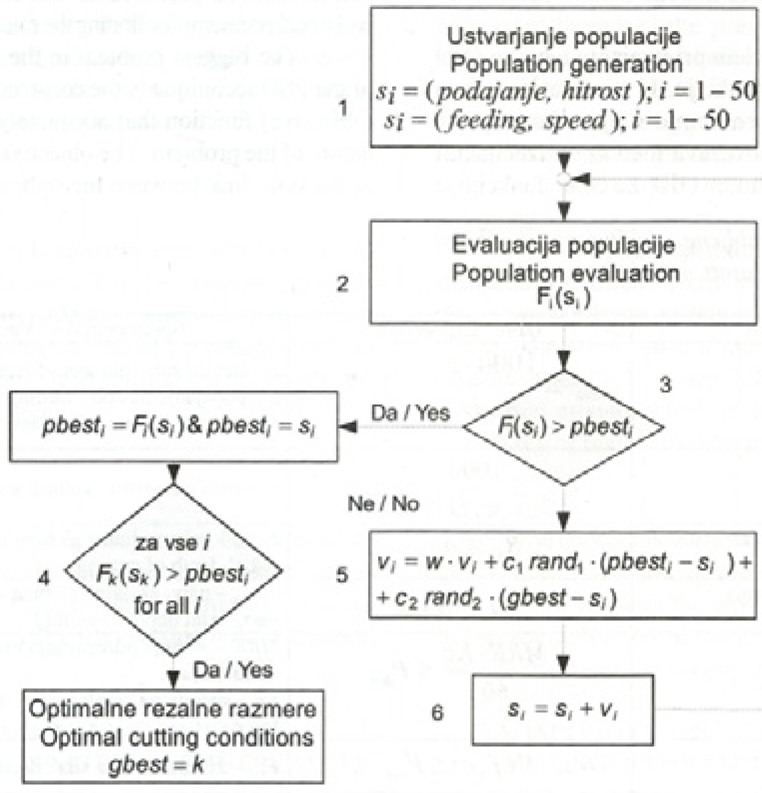
1. Ustvarjanje in začenjanje matrike 50 delcev z naključnimi legami in hitrostmi. Vektor hitrosti ima dve razsežnosti. Prva je podajanje, druga je vrtilna frekvenca. Tako je zastavljena generacija 0.
2. Ovrednotenje optimizacijske funkcije (površina rezalnih sil) za vsak delec.
3. Izračun vrednosti sil za novo lego vsakega delca. Če delec doseže ugodnejšo lego, njegova trenutna vrednost nadomesti vrednost *pbest*.

and the PSO algorithm. For the objective function a surface of maximum cutting forces is selected, generated by the ANN.

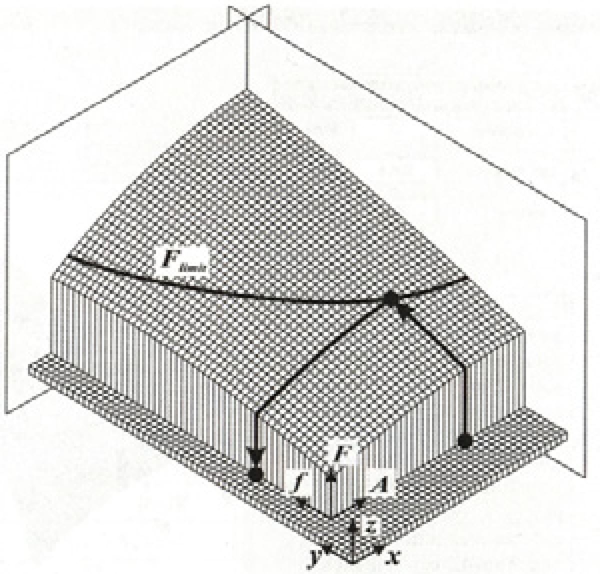
The PSO algorithm generates a swarm of particles on the cutting-force surface during the second phase. The swarm of particles flies over the cutting-force surface and searches for the maximal cutting force. The coordinates of a particle that has found the maximal (but still allowable) cutting force represent the optimal cutting conditions. Figure 5 shows the PSO flowchart for the optimization of the milling process.

The optimization process is described by the following steps:

1. Generation and initialization of an array of 50 particles with random positions and velocities. The velocity vector has two dimensions: feed rate and spindle speed. This constitutes Generation 0.
2. The evaluation of the objective (cutting-force surface) function for each particle.
3. The cutting-force values are calculated for new positions of each particle. If a better position is achieved by particle, the *pbest* value is replaced by the current value.



Sl. 5. *Algoritem OJD za optimiranje rezalnih razmer*
Fig. 5. *PSO algorithm for optimizing the cutting conditions*



Sl. 6. *Postopek iskanja optimalnega podajanja*
Fig. 6. *Optimal feeding searching procedure*

4. Preverjanje, ali je delec našel največjo rezalno silo v populaciji. Če je nova *gbest* vrednost boljša od prejšnje se shrani v spremenljivko *gbest*. Rezultat optimizacije je vektor *gbest* (podajanje, vrtilna hitrost)
5. Izračun nove hitrosti delca.
6. Popravek lege delca s pomikom k največji rezalni sili.
7. Ponavljanje korakov 1 in 2, dokler ne dosežemo vnaprej določene iteracije.

Na sliki 6 je prikazano poenostavljeno načelo optimiranja rezalnih razmer pri frezanju z uporabo metode OJD. V tem primeru jata delcev potuje po površini sile in išče optimalno podajanje pri nespremenljivem prerezu odrezka A. Optimalno podajanje je v presečišču naslednjih treh ploskev: površine rezalne sile, ploskve z nespremenljivim prerezom odrezka (navpična ravnina) in ravnine primerjalne sile. Koordinata delca v jati, ki se najbolj približa presečišču ravnin, pomeni optimalno vrednost podajanja.

4. The determination if the particle has found the maximal force in the population. If the new gbest value is better than previous gbest value, the gbest value is replaced by the current *gbest* value and stored. The result of the optimization is the vector *gbest* (feedrate, spindle speed).
5. Computation of the particles' new velocity
6. Update particle's position by moving towards the maximal cutting force.
7. Steps 1 and 2 are repeated until the iteration number reaches a predetermined iteration

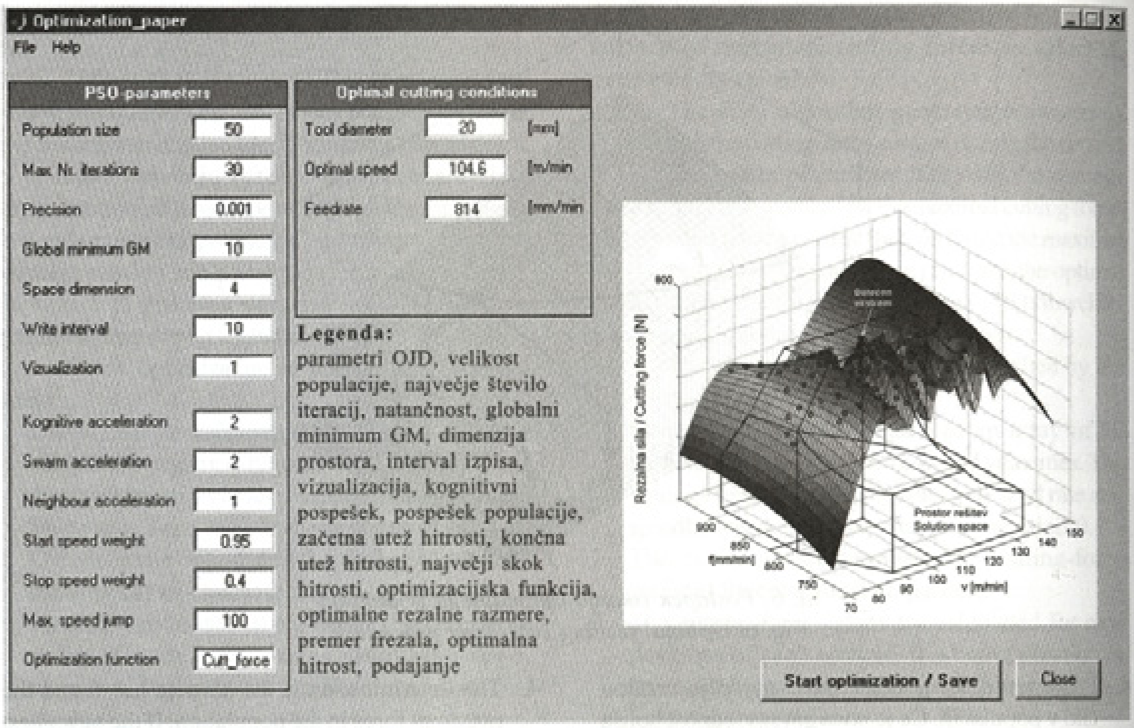
Figure 6 shows the simplified principle of the optimization of the cutting conditions using the PSO. In this case the swarm flies over the force surface and searches for optimal feeding at a constant cheap cross-section A. The optimal feed rate is located at the cross-section of the following three planes: the cutting force surface, the plane with the constant cheap cross-section (vertical plane) and the desired cutting force plane. The coordinate of the particle that is the nearest to the mentioned cross-section represents the optimal feed rate.

4 RAČUNALNIŠKI PROGRAM ZA OPTIMZACIJO OJD

Programsko opremo OJD sestavlja zbirka Matlabovih m- datotek. Programska oprema se lahko uporabi za optimizacijo poljubnega nelinearnega sistema. Zahtevane vrednosti se lahko vnesejo v programskem oknu, ki je prikazano na sliki 7. V levem

4 COMPUTER SOFTWARE FOR PSO OPTIMIZATION

A collection of Matlab's m-files forms the PSO software for the optimization. This software can be used for the optimization of an arbitrary non-linear system. The required input values can be inserted into the software window shown in Figure 7. On the



Sl. 7. *Programsko okno optimizacije OJD*

Fig. 7. *Software window for PSO optimization*

delu okna se nastavijo parametri, ki so potrebni za delovanje algoritma OJD. Rezultat optimizacije (optimalni rezalni parametri) se prikažejo na sredini okna. Postopek optimizacije grafično spremljamo na grafu.

5 TESTNI PRIMER OPTIMIZACIJE PSO REZALNIH POGOJEV

Na naslednjem testnem primeru je prikazana ponovljivost in robustnost algoritma OJD. Da preverimo stabilnost in robustnost predlagane optimizacijske strategije, sistem najprej analiziramo s simulacijami, nato sistem preverimo na RK frezalnem stroju (tip HELLER BEA1) za Ck 45 in 16MnCrSi5 XM jeklene obdelovance [16]. Za preizkuse uporabimo krogelno končno frezalo (R220-20B20-040) premera 20 mm z dvema rezalnima robovoma in kotom vijačnice 10°. Uporabljeni so naslednji rezalni parametri in omejitve: Širina rezanja R_D=3 mm, globina rezanja A_D=5 mm, rezalna hitrost v_c=80 m/min, $n \leq 2000$ min^{-1}, $10 \leq f \leq 900$ mm/min, $F(f, n) \leq Fref = 600$ N. Ciljna funkcija je določena z nevronskim modelom rezalnih sil (simulator rezalnih sil). Cilj primera je do skrajnosti zvečati ciljno funkcijo ob upoštevanju danih omejitev [17]. Ta problem je

left-hand side of the window, the parameters required for executing the PSO algorithm can be set. The result of the optimization (the optimal cutting parameters) is shown in the middle of the window. The process of optimization is monitored on a graph.

5 PSO OPTIMIZATION OF TEST-CASE CUTTING CONDITIONS

The repeatability and robustness of the PSO algorithm is demonstrated with the following test case. To examine the stability and robustness of the proposed optimization strategy, the system is first analyzed by simulations; then the system is verified by experiments on a CNC milling machine (type HELLER BEA1) for Ck 45 and 16MnCrSi5 XM steel workpieces [16]. The ball-end milling cutter (R220-20B20-040) with two cutting edges, of 20 mm diameter and 10° helix angle, was selected for the experiments. The following cutting parameters and constraints are used: milling width R_D=3 mm, milling depth A_D=5 mm, cutting speed v_c=80 m/min, $n \leq 2000$ min^{-1}, $10 \leq f \leq 900$ mm/min, and $F(f, n) \leq Fref = 600$ N. The objective function is determined by the neural cutting-force model (the cutting-force simulator). The goal of this case is to maximize the objective function

Preglednica 2. *Ponovljivost rezultatov*
Table 2. *Repeatability of results*

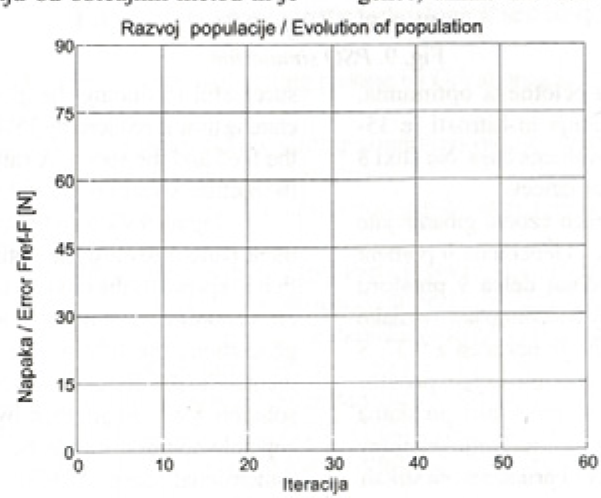
Test/Run	n min^{-1}	f mm/min	F N	Št. iteracij Nr. of iterations
1	1998	808,2	598	22
2	1995	810,1	600	25
3	1997	811,2	600	28
4	1997	819,7	598	32
5	2000	819,1	598	22
6	1999	819,2	598	31
7	1999	808,0	597	26
8	1998	808,8	598	21
9	1998	808,9	598	32
10	2000	808,1	597	30

rešen z uporabo algoritma OJD. V algoritmu OJD uporabimo 50 delcev. Postopek iskanja se izvaja, dokler napaka gradienta ni manjša od izbrane vrednosti. Matlab® simulira naučeno nevronsko mrežo pri napovedovanju rezalnih sil za dane rezalne razmere, te vrednosti se nato uporabijo pri izračunu ciljne funkcije, ki jo algoritem OJD skuša najbolj povečati. Rezultati so prikazani v preglednici 2. Številka testa ustreza vsakemu poskusu programa, da poišče optimalne rezalne parametre. V preglednici 2 so prikazane optimalne rezalne razmere skupaj s številom iteracij, ki so potrebne za dosego omenjenega optimuma.

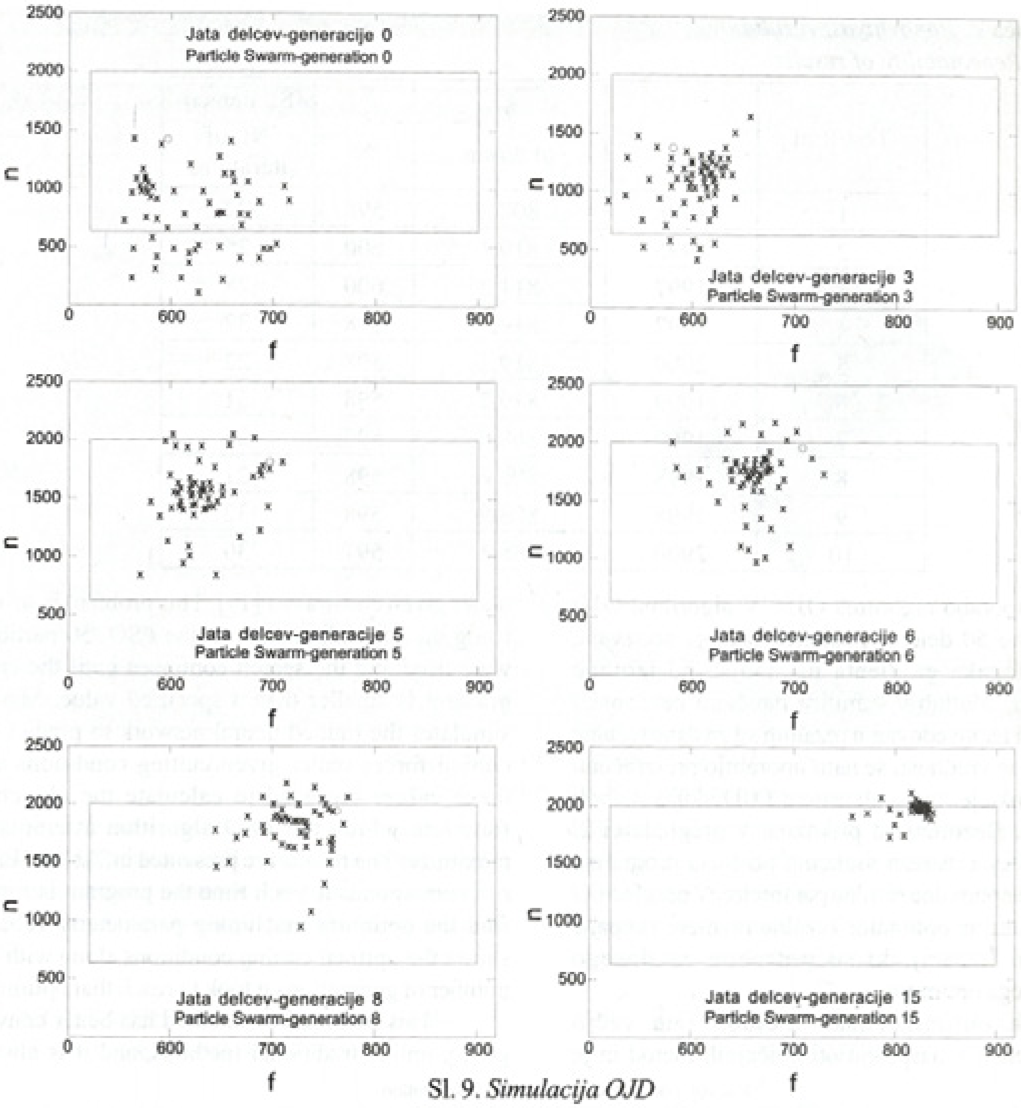
Ta optimizacijska metoda ima večjo konvergenco v nasprotju od običajnih metod in je

under given constraints [17]. This problem is solved using the PSO algorithm. In the PSO, 50 particles were used and the search continues until the error gradient is smaller than a specified value. Matlab simulates the trained neural network to predict the cutting forces under given cutting conditions and these values are used to calculate the objective function, which the PSO algorithm attempts to maximize. The results are presented in Table 2. Each run corresponds to each time the program is run to find the optimum machining parameters. Table 2 shows the optimal cutting conditions along with the number of generations it took to reach that optimum.

This optimization method has better convergence, unlike traditional methods, and it is always



Sl. 8. *Zmanjševanje napake optimacije med razvojem jate*
Fig. 8. *Decrease of optimization error during swarm evolution*



Sl. 9. *Simulacija OJD*

Fig. 9. *PSO simulation*

vedno uspešna pri iskanju celotnega optimuma. Rezultat optimiranja podajanja in hitrosti je 35-odstotno zmanjšanje obdelovalnega časa. Na sliki 8 je prikazan primer razvoja jate delcev.

Slika 9 prikazuje tipičen vzorec gibanja jate delcev proti optimalni rešitvi. Generacija 0 pomeni naključno začenjanje koordinat delca v prostoru rešitev. V nadaljnjih generacijah sledimo jati z oznako "x". Najboljši član populacije je označen z "O". S pravokotnikom je grafično ponazorjen prostor rešitev. Sprejemljivo rešitev mora biti poiskana znotraj tega dvorazsežnega prostora. Tretjo omejitev - silo tudi upoštevamo, čeprav ni prikazana na slikah. S simulacijami prikažemo robustnost in učinkovitost algoritma.

successful in finding the global optimum. The machining time is reduced by 35% as a result of optimizing the feed and the speed. A sample of the evolution of the particle swarm is presented in Fig. 8.

Figure 9 shows a typical particle swarm movement pattern toward the optimum solution. Generation 0 represents the random initialization of the particle's coordinates in the solution space. In subsequent generations, the swarm is tracked with "x". The best member in the population is represented by "O". The solution space is graphed by the rectangle. An acceptable solution has to be found within this two-dimensional space. A third constraint, acting on the force is also active, and as such is not part of these illustrations. Using simulations the robustness and the efficiency of the algorithm are demonstrated.

6 POVZETEK IN NADALJNJE RAZISKAVE

V raziskavi je prikazan postopek večciljnega optimiranja postopka frezanja z uporabo nevronskega modeliranja in optimizacije, ki temelji na zakonitostih gibanja majhnih delcev v velikih jatah. Za napovedovanje rezalnih sil je uporabljen model rezalnih sil, za določitev optimalne rezalne hitrosti in podajanja uporabimo algoritem OJD. Med optimizacijo smo up...bili zbirko sedmih omejitev. Nato z nevronskim ... lelom rezalnih sil napovemo ciljno funkcijo. Nad... je uporabljen algoritem OJD, s katerim optimiram... o podajanje in rezalno hitrost za značilen primer ... industrije. Eksperimentalni rezultati pokažejo, ... se MRR izboljša za 28%. Opažena je tudi ... odstotno zmanjšanje časa obdelave. Prispeve... ipravlja teren za nov razred RR optimizacijskih ... nik na področju obdelave z odrezovanjem. ... prispevku so prav tako predstavljene osno... ptimizacije OJD. Medtem ko je bilo veliko tehni... zvojnega računanja izdelanih za reševanje sestavnih optimizacijskih problemov, je bila OJD izdelana za reševanje neprekinjenih problemov. Metoda OJD je lahko učinkovito orodje za reševanje nelinearnih neprekinjenih, sestavnih in kombiniranih - integriranih optimizacijskih problemov.

6 CONCLUSION AND FUTURE RESEARCH

This study has presented the multi-objective optimization of the milling process by using neural network modelling and particle swarm optimization. A neural network model was used to predict the cutting forces during the machining and the PSO algorithm was used to obtain the optimum cutting speed and feed rate. A set of seven constraints were used during the optimization. Next, the neural force model was used to predict the objective function. Next, the PSO algorithm was used to optimize both the feed and the speed for a typical case found in industry. The experimental results show that the MRR is improved by 28%. Machining time reductions of up to 20% were also observed. This paper opens the door for a new class of EC-based optimization techniques in the area of machining. This paper also presents the fundamentals of PSO optimization techniques. While a lot of evolutionary computation techniques have been developed for combinatorial optimization problems, the PSO has been basically developed for continuous optimization problems. The PSO can be an efficient optimization tool for solving nonlinear continuous optimization problems, combinatorial optimization problems, and mixed-integer nonlinear optimization problem.

7 LITERATURA
7 REFERENCES

[1] Župerl, U., F... š (2004) Določevanje značilnih tehnoloških in gospodarskih parametrov med postopkom odrezovanja ... determination of the characteristic technological and economic parameters during metal cutting. *Stro... stn.*, 5(2004), pp. 252-266.

[2] Čuš, F., J. B... (2003) Optimization of cutting process by GA approach. *Robot. Comput. Integr. Manuf.* 19(2003), pp ...-121.

[3] Župerl, U., ... uš (2003), Optimization of cutting conditions during cutting by using neural networks, *Robot. Com... Integr. Manuf.*, 19(2003), pp. 189-199.

[4] Balič, J. (20... A new NC machine tool controller for step-by-step milling, *Int. J. Adv. Manuf. Technol.* 8(2000) pp. ...403.

[5] Ozcan, E., C. Mohan (1998), Analysis of a simple Particle Swarm Optimization system, *Intelligent Engineering Systems Through Artificial Neural Networks*, 8(1998), pp. 253-258.

[6] Shi Y., R. Eberhart (1998), Parameter selection in Particle Swarm Optimization, *In Evolutionary Programming VII: Proc. EP98*, New York: Springer-Verlag (1998), pp. 591-600.

[7] Angeline, P.J. Evolutionary optimization versus Particle Swarm Optimization: philosophy and performance differences, *in: Proceedings of the 7th ICEC*, (1998), pp. 601-610.

[8] Kennedy, J., R.C. Eberhart (1995) Particle Swarm Optimization, *Proceedings of IEEE Int. Conference on Neural Networks*, 4(1995), pp. 1942-1948.

[9] Reynolds, C. (1990) Flocks, herds, and schools: a distributed behavioural model, *Computer Graphics*, 21(1990), pp. 25-34.

[10] R.C., Eberhart, Y. Shi (2003), Comparison between genetic algorithm and Particle Swarm Optimization, *Proceedings of the 7th ICEC*, (1998), pp. 611-616.

[11] Boyd, J. (2003), Thinking is Social: Experiments with the adaptive culture model, *Journal of Conflict Resolution*, 42(1998), pp. 56-76.

[12] Balic J. (2003), Optimization of cutting process by GA approach, *Robot. Comput. Integr. Manuf.* 19(2003) 113-121.

[13] He, Z., C. Wei, L. Yang, X. Gao, S. Yao, R. Eberhart, Y.: Shi, (1998), Extracting rules from fuzzy neural network by Particle Swarm Optimization, *Proc. of IEEE International Conference on Evolutionary Computation (ICEC'98)*, (1998), pp. 66-71.

[14] Cus, F., U. Zuperl, E. Kiker, M. Milfelner (2006), Adaptive controller design for feedrate maximization of machining process, *Journal of Achievements in Materials and Manufacturing Engineering* 17(2006),pp. 237-240.

[15] Abido M.A. (2002), Optimal power flow using Particle Swarm Optimization, *International Journal of Electrical Power & Energy Systems,* 24(2002), pp. 563-571.

[16] Kopač, J. (2002) Rezalne sile in njihov vpliv na gospodarnost obdelave = Cutting forces and their influence on the economics of machining. *Stroj. vestn*. 3(2002), pp. 121-132.

[17] Župerl, U., F. Čuš (2004) Tool cutting force modeling in ball-end milling using multilevel perceptron, *J. Mater. Process. Technol.*, (2004), pp. 67-72.

Naslova avtorjev: dr. Uroš Župerl
Prof. Dr. Franci Čuš
Univerza v Mariboru
Fakulteta za strojništvo
Smetanova 17
2000 Maribor
uros.zuperl@uni-mb.si

prof. dr. Valentina Gecevska
Univerza v Skopju
Fakulteta za strojništvo
P.P. 464
1000 Skopje, Makedonija

Authors' addresses: Dr. Uroš Župerl
Prof. Dr. Franci Čuš
University of Maribor
Faculty of Mechanical Eng.
Smetanova 17
SI-2000 Maribor, Slovenia
uros.zuperl@uni-mb.si

Prof. Dr. Valentina Gecevska
University in Skopje
Faculty of Mechanical Eng.
PO Box 464
1000 Skopje, Macedonia

Prejeto:
Received: 26.2.2007

Sprejeto:
Accepted: 25.4.2007

Odprto za diskusijo: 1 leto
Open for discussion: 1 year