

Optimization of Cutting Conditions in Multi-Pass Turning Using Hybrid Genetic Algorithm-Sequential Quadratic Programming

Abderrahim Belloufi^{1*}, Mekki Assas² and Imane Rezgui³

¹Mechanical Engineering Department, University Mohamed Kheider, Biskra, Algeria

²Engineering Production Laboratory, University HadjLakhdar, Batna, Algeria

³Mechanical Engineering Department, University HadjLakhdar, Batna, Algeria

Abstract

In this paper, a new, hybrid genetic algorithm-sequential quadratic programming is used for the resolution of cutting conditions. It used for the resolution of a multi-pass turning optimization case by minimizing the production cost under a set of machining constraints. The result indicates that the proposed hybrid genetic algorithm-sequential quadratic programming is effective when compared to other techniques carried out by different researchers.

Keywords: Multi-pass turning; Genetic algorithm; Sequential quadratic programming; Cutting conditions

Abbreviations: C_i (\$/piece): Machine idle cost due to loading and unloading operations and tool idle motion time; C_M (\$/piece): Cutting cost by actual time in cut; C_R (\$/piece): Tool replacement cost; C_T (\$/piece): Tool cost; d_r, d_s (mm): Depth of cut for each pass of rough and finish machining; d_{rL}, d_{rU} (mm): Lower and upper bound of depth of cut in rough machining; d_{sL}, d_{sU} (mm): Lower and upper bound of depth of cut in finish machining; d_f (mm): Depth of material to be removed; D, L (mm): Diameter and length of work piece; $f_{r,s}$ (mm/rev): Feed rates in rough and finish machining; f_{rL}, f_{rU} (mm/rev): Lower and upper bound of feed rate in rough machining; f_{sL}, f_{sU} (mm/rev): Lower and upper bound of feed rate in finish machining; F_r, F_s (kgf): Cutting forces during rough and finish machining; F_u (kgf): Maximum allowable cutting force; h_1, h_2 (mm): Constants relating to cutting tool travel and approach/departure time; k_o (\$/mm): Direct labor cost plus overhead; k_e (\$/edge): Cutting edge cost; k_1, μ, v : Constants of cutting force equation; k_2, τ, ϕ, δ : Constants related to chip-tool interface temperature equation; k_3, k_4, k_5 : Constants for roughing and finishing parameter relations; λ, v : Constants related to expression of stable cutting region; n : Number of rough cuts (an integer); N_U, N_L : Upper and lower bounds of n ; p, q, r, C_p : Constants of tool-life equation; P_r, P_s (kW): Cutting power during rough and finish machining; P_U (kW): Maximum allowable cutting power; Q_r, Q_s (°C): Chip-tool interface rough and finish machining temperatures; Q_U (°C): Maximum allowable chip-tool interface temperature; q : A weight for T_p [0,1]; R (mm): Nose radius of cutting tool; SC : Limit of stable cutting region constraint; SR_U (mm): Maximum allowable surface roughness; T_r, T_s (mm): Tool life, expected tool life for rough machining and expected tool life for finish machining; T_p (mm): tool life of weighted combination of T_r and T_s ; T_U, T_L (mm): Upper and lower bounds for tool life; UC \$: Unit production cost except material cost; V_r, V_s (M/mm): Cutting speeds in rough and finish machining; V_{rL}, V_{rU} (M/mm): Lower and upper bound of cutting speed in rough machining; V_{sL}, V_{sU} (M/mm): Lower and upper bound of cutting speed in finish machining

Introduction

The selection of optimal cutting parameters, like the number of passes, depth of cut for each pass, feed and speed, is a very important issue for every machining process [1]. Several cutting constraints must be considered in machining operations. In turning operations, a cutting process can possibly be completed with a single pass or by multiple passes. Multi-pass turning is preferable over single-pass

turning in the industry for economic reasons [2]. The optimization problem of machining parameters in multi-pass turnings becomes very complicated when plenty of practical constraints have to be considered [3]. Traditionally, mathematical programming techniques like graphical methods [4], linear programming [5], dynamic programming [6,7] and geometric programming [8,9] had been used to solve optimization problems of machining parameters in multi-pass turnings. However, these traditional methods of optimization do not fare well over a broad spectrum of problem domains. Moreover, traditional techniques may not be robust. Numerous constraints and multiple passes make machining optimization problems complicated and hence these techniques are not ideal for solving such problems as they tend to obtain a local optimal solution. Thus, meta-heuristic algorithms have been developed to solve machining economics problems because of their power in global searching. There have been some works regarding optimization of cutting parameters [2,3,10-14] for different situations, authors have been trying to bring out the utility and advantages of genetic algorithm, evolutionary approach and simulated annealing. It is proposed to use the hybrid genetic algorithm-sequential quadratic programming for the machining optimization problems.

The current paper focuses on the application of a new optimization technique, hybrid genetic algorithm-sequential quadratic programming, to determine the optimal machining parameters that minimize the unit production cost in multi-pass turnings.

Cutting process model

Decision variables: In the constructed optimization problem, six decision variables are considered: cutting speeds in rough and finish machining (V_r, V_s), feed rates in rough and finish machining (f_r, f_s) and depth of cut for each pass of rough and finish machining (d_r, d_s).

Objective function: Based on the minimum unit production

*Corresponding author: Abderrahim Belloufi, Mechanical Engineering Department, University Hadj Lakhdar, Batna, Algeria, Tel: +213 661 57 04 24, E-mail: abelloufi@yahoo.fr

Received January 03, 2012; Accepted February 21, 2012; Published February 23, 2012

Citation: Belloufi A, Assas M, Rezgui I (2012) Optimization of Cutting Conditions in Multi-Pass Turning Using Hybrid Genetic Algorithm-Sequential Quadratic Programming. J Applied Mechanic Engg 1:101. doi:10.4172/2168-9873.1000101

Copyright: © 2012 Belloufi A, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

cost, UC , criterion, the objective function for a multi-pass turning operation can be given by the equation [10]:

$$UC = C_M + C_I + C_R + C_T \quad (1)$$

$$C_M = k_0 \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right], \quad (2)$$

$$C_I = k_0 \left[t_c + (h_1 L + h_2) \left(\frac{d_t - d_s}{d_r} + 1 \right) \right], \quad (3)$$

$$C_R = k_0 \frac{t_c}{T_p} \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right], \quad (4)$$

$$C_T = \frac{k_t}{T_p} \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right]. \quad (5)$$

Constraints

There are some constraints which affect the selection of the optimal cutting conditions and will be taken into account. The constraints in rough and finish machining are as outlined below [10].

Rough machining:

I. Parameter bounds

Due to the limitations on the machine and cutting tool and due to the safety of machining the cutting parameters are limited with the bottom and top permissible limit.

$$\text{Cutting speed: } V_{rL} \leq V_r \leq V_{rU} \quad (6)$$

$$\text{Feed rate: } f_{rL} \leq f_r \leq f_{rU} \quad (7)$$

$$\text{Depth of cut: } d_{rL} \leq d_r \leq d_{rU} \quad (8)$$

II. Tool-life constraint

The constraint on the tool life is taken as:

$$T_L \leq T_r \leq T_U \quad (9)$$

III. Cutting force constraint

The maximum amount of cutting forces F_u should not exceed a certain value as higher forces produce shakes and vibration. This constraint is given below.

$$F_r = k_1 (f_r)^u (d_r)^v \leq F_u \quad (10)$$

IV. Power constraint

The nominal power of the machine P_U limits the cutting process:

$$P_r = \frac{F_r V_r}{6120\eta} \leq P_U \quad (11)$$

Efficiency $\eta = 0.85$

V. Stable cutting region constraint

This constraint is given as:

$$(V_r)^2 f_r (d_r)^v \geq SC \quad (12)$$

VI. Chip-tool interface temperature constraint

This constraint is given as:

$$Q_r = k_2 (V_r)^z (f_r)^{\phi} (d_r)^{\delta} \leq Q_u \quad (13)$$

Finish machining: All the constraints other than the surface finish constraint are similar for rough and finish machining [15].

I. Surface finish constraint

In the finishing operations, the obtained surface roughness must be smaller than the specified value, SR_U , given by technological criteria, so that the following equation is satisfied:

$$\frac{f_s^2}{8R} \leq SR_U \quad (14)$$

Constraints for roughing and finishing parameter relations

$$V_s \geq k_3 V_r \quad (15)$$

$$f_r \geq k_4 f_s \quad (16)$$

$$d_r \geq k_5 d_s \quad (17)$$

II. The number of rough cuts

The possible number of rough cuts is restricted by

$$n = \frac{d_t - d_s}{d_r} \quad (18)$$

Where $n_L \leq n \leq n_U$

$$n_L = (d_t - d_{sU}) / d_{rU} \quad (19)$$

$$n_U = (d_t - d_{sL}) / d_{rL} \quad (20)$$

The optimization problem in multi-pass turnings can be divided into $m = (n_U - n_L + 1)$ sub-problems, in each of which the number of rough cuts n is fixed. So the solution of the whole optimization problem is divided into searching the optimal results of m sub-problems and the minimum of them is the objective of whole optimization problem.

Genetic Algorithm (GA)

Genetic algorithm is a global optimization method, is developed to solve the general optimization problem. GA simulate biological evolution process; Darwin's theory of survival of the fittest.

The solution of an optimization problem with genetic algorithm begins with a set of potential solutions or chromosomes (usually in the form of bit strings) that are randomly selected. The entire set of these chromosomes comprises a population. The chromosomes evolve during several iterations or generations. New generations are generated utilizing the crossover and mutation technique. Crossover involves splitting two chromosomes and then combining one-half of each chromosome with the other pair. Mutation involves flipping a single bit of a chromosome. The chromosomes are then evaluated employing a certain fitness criteria and the best ones are kept while the others are discarded. This process repeats until one chromosome has the best fitness and is taken as the optimum solution of the problem [16].

The computational procedure involved in maximizing the fitness function $f(x_1, x_2, x_3, \dots, x_n)$ in the genetic algorithm can be described by the following steps [17].

1. Choose a suitable string length $l = nq$ to represent the n design variables of the design vector X . Assume suitable values for the following parameters: population size n , crossover probability p_c , mutation probability p_m , permissible value of standard deviation of fitness values of the population $(S_f)_{max}$ to use as a convergence criterion, and maximum number of generations i_{max} to be used as a second convergence criterion.

2. Generate a random population of size m , each consisting of a string of length $l = nq$. Evaluate the fitness values F_i , $i = 1, 2, \dots, m$, of the m strings.
3. Carry out the reproduction process.
4. Carry out the crossover operation using the crossover probability p_c .
5. Carry out the mutation operation using the mutation probability p_m to find the new generation of m strings.
6. Evaluate the fitness values F_i , $i = 1, 2, \dots, m$, of the m strings of the new population. Find the standard deviation of the m fitness values.
7. Test for the convergence of the algorithm or process. If $s_f \leq (s_f)_{\max}$, the convergence criterion is satisfied and hence the process may be stopped. Otherwise, go to step 3.
8. Test for the generation number. If $i \geq i_{\max}$, the computations have been performed for the maximum permissible number of generations and hence the process may be stopped. Otherwise, set the generation number as $i = i + 1$ and go to step 3.

Basic genetic algorithm operations

There are three basic operators found in every genetic algorithm: reproduction, crossover and mutation [18].

Reproduction: The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's fitness value, calculated from a fitness function. For each generation, the reproduction operator chooses strings that are placed into a mating pool, which is used as the basis for creating the next generation.

There are many different types of reproduction operators:

1. Proportional selection: This method will only work with fitness values above zero (non-negative) and scaling may sometimes be necessary. It has been shown that proportional selection performs poorly compared with other selection schemes in many GA problems.
2. Tournament selection: Choose t individuals at random from the population and copy the best individual from this group into the new population. Repeat N times.
3. Truncation selection: With truncation selection that has a threshold of T between 0 and 1, only the fraction T best individuals can be selected. They all have the same selection probability.
4. Linear ranking selection: The individuals are sorted according to their fitness values and the rank N is assigned to the best individual, the rank 1 assigned to the worst. The selection probability is linearly assigned to the individuals according to their rank and a selection equation.
5. Exponential ranking selection: This follows the same methodology of linear ranking selection, the only difference being that the probabilities of the ranked individuals are exponentially weighted.

One always selects the fittest and discards the worst, statistically selecting the rest of the mating pool from the remainder of the population. There are hundreds of variants of this scheme. None are right or wrong.

Crossover: Once the mating pool is created, the next operator in the GA arsenal comes into play.

The GA selects two strings at random from the mating pool. The strings selected may be different or identical, it does not matter. The GA then calculates whether crossover should take place using a parameter called the crossover probability. If the GA decides not to perform crossover, the two selected strings are simply copied to the new population.

If crossover does take place, then a random splicing point is chosen in a string, the two strings are spliced and the spliced regions are mixed to create two new strings. These child strings are then placed in the new population.

Mutation : Selection and crossover alone can obviously generate a staggering amount of differing strings. However, depending on the initial population chosen, there may not be enough variety of strings to ensure the GA sees the entire problem space. Or the GA may find itself converging on strings that are not quite close to the optimum it seeks, due to a bad initial population. Some of these problems are overcome by introducing a mutation operator into the GA. The GA has a mutation probability, which dictates the frequency at which mutation occurs. Mutation can be performed either during selection or crossover. For each string element in each string in the mating pool, the GA checks to see if it should perform a mutation. If it should, it randomly changes the element value to a new one.

Encodings and optimization problems: Usually there are only two main components of most genetic algorithms that are problem dependent: the problem encoding and the evaluation function. Consider a parameter optimization problem where we must optimize a set of variables either to maximize some target such as profit, or to minimize cost or some measure of error. We might view such a problem as a black box with a series of control dials representing different parameters; the only output of the black box is a value returned by an evaluation function indicating how well a particular combination of parameter settings solves the optimization problem.

Sequential Quadratic Programming (SQP)

The sequential quadratic programming, solve the general nonlinear programming problem. The problem is stated as follows:

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to } g_i(x) = 0, \quad i = 1, \dots, n \\ h_j(x) \geq 0, \quad j = n + 1, \dots, m \\ x^{\text{inf}} \leq x \leq x^{\text{sup}} \end{cases} \quad (21)$$

Where all problem functions are assumed to be continuously differentiable. The method, based on the iterative formulation and solution of quadratic programming subproblems, obtains subproblems by using a quadratic approximation of the Lagrangian and by linearizing the constraints. That is,

$$\begin{cases} \min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B_k d + \nabla f(x^k)^T d \\ \text{subject to } \nabla g_i(x^k)^T d + g_i(x^k) = 0, \quad i = 1, \dots, n \\ \nabla h_j(x^k)^T d + h_j(x^k) \geq 0, \quad j = n + 1, \dots, m \\ x^{\text{inf}} - x^k \leq d \leq x^{\text{sup}} - x^k \end{cases} \quad (22)$$

Where B_k is a positive definite approximation of the Hessian and

x_k is the current iterate. Let d_k be the solution of the subproblem. A line search is used to find a new point x_{k+1} ,

$$x^{k+1} = x^k + \alpha d_k, \quad \alpha \in [0,1] \tag{23}$$

Such that a “merit function” will have a lower function value at the new point.

Hybrid GA-SQP

SQP requires a smaller number of objective and constraint function calls than GA. It can also find accurate optimum results as it is a deterministic algorithm. However, due to the fact that SQP uses gradient information in its search algorithm, it tends to be trapped in the local optimum and suffers from noise in objective or constraint functions [19]. In contrast, GA search more globally and have more chance to find a global optimum. J. H. Holland [20] suggested that GA should be used to perform the initial global search. The results are used to guide the local search.

In order to benefit the global search ability of a GA and the accurate local search of a SQP, they are used as a complement of each other [21,22]. To do so, the GA stopping criteria are set so that the GA would stop prematurely e.g. with a low generation, a low population or a high tolerance. It is assumed that the GA should find its optimal results near the true global optimum. The GA results are therefore used as an initial point for the SQP algorithm. The SQP proceeds the local search and find its local optimum, which is the global optimum sought (Figure 1).

Advantages and Drawbacks of the Optimization Methods

The advantages and drawbacks of the optimization methods are mentioned in (Table 1).

Application Example

Now an application example is considered to demonstrate and validate the hybrid GA-SQP method for the optimization of process parameters of the multi-pass turning operation. The parameters used for the numerical application are mentioned in (Table 2).

Results and Discussion

The genetic algorithm was run with these parameters:

- Population size: 20,
- Scaling function: Rank (The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function).
- Selection function: Roulette.
- Reproduction: Elite count: 2,
- Crossover fraction: 0.8,
- Mutation: randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation. A step length is chosen along each direction.
- Crossover: Scattered (creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child).
- Migration fraction: 0.2,

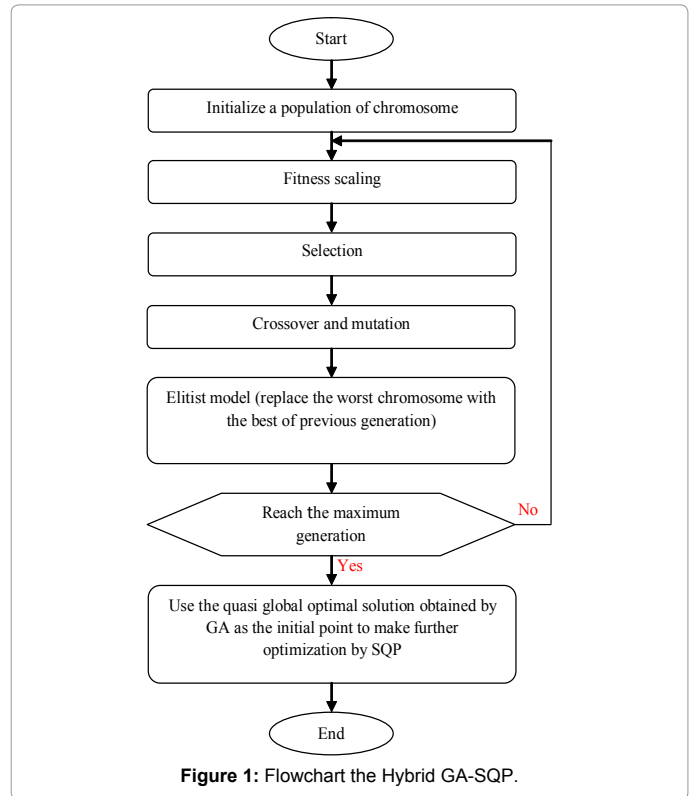


Figure 1: Flowchart the Hybrid GA-SQP.

Method	Advantages	Drawbacks
SQP	<ul style="list-style-type: none"> - High accuracy - Small number of function evaluations - Performs well with constrained problems 	<ul style="list-style-type: none"> - Local method - Needs gradient information - Continuous variables - Needs initial point
Meta-heuristic method's	<ul style="list-style-type: none"> - Global method - Robust, no gradient need - Continuous and discrete variables 	<ul style="list-style-type: none"> - Low accuracy - Large number of functions evaluation
Hybrid GA-SQP	<ul style="list-style-type: none"> - Able to find global optimum - High accuracy - Less evaluations than GA alone 	<ul style="list-style-type: none"> - Needs gradient information - Continuous variables

Table 1: Advantages and drawbacks of the optimization methods.

- Migration interval: 20,
- Number maximal of iterations: 100.

Several GA generations are performed in order to identify the most promising areas and then the SQP optimization algorithm is applied using as an initial guess the best individual found by the GA. It should be noted that in this approach the GA is used to specify a good initial guess for the SQP algorithm.

The results found by the hybrid GA-SQP are mentioned on (Table 3).

We find that the lowest value is 1.9308\$ under which the minimum number of rough cuts ($n=1$) is taken.

The performance of the hybrid GA-SQP and others can be seen in (Table 4).

According to Table 4 one notices that the hybrid GA-SQP yield much better results than the others algorithms. Thus the hybrid GA-

Parameter	Values	Parameter	Values	Parameter	Values
$D(mm)$	50	$L(mm)$	300	$D_r(mm)$	6
$V_{rL}(m/min)$	500	$V_{rR}(m/min)$	50	$f_{rL}(mm/rev)$	0.9
$f_{rL}(mm/rev)$	0.1	$D_{rL}(mm)$	3.0	$D_{rR}(mm)$	1.0
$V_{sU}(m/min)$	500	$V_{sL}(m/min)$	50	$f_{sU}(mm/rev)$	0.9
$f_{sL}(mm/rev)$	0.1	$D_{sU}(mm)$	3.0	$D_{sL}(mm)$	1.0
P	5	q	1.75	R	0.75
K	108	μ	0.75	ν	-1
η	0.85	λ	2	φ	0.95
k_2	132	τ	0.4		0.2
$\bar{\sigma}$	0.105	$R(mm)$	1.2	$K_0(\$/mm)$	0.5
C_0	6×10^{11}	h_1	7×10^{-4}	h_2	0.3
$T_L(min)$	25	$t_L(min/piece)$	0.75	$t_e(min/edge)$	1.5
$P_U(kW)$	5	$T_U(min)$	45	$F_U(Kgf)$	200
SC	140	$SR_U(\mu m)$	10	$Q_U(^\circ C)$	1000
k_3	1.0	k_4	2.5	k_5	1.0
$k_1(\$/edge)$	2.5				

Table 2: Machining data [10].

N	Cutting parameters (Rough machining)			Cutting parameters (Finish machining)			UC(\$)
	V_r (m/min)	f_r (mm/rev)	d_r (mm)	V_s (m/min)	f_s (m/rev)	D_s (mm)	
1	94.4640	0.8660	3.0000	162.2890	0.2580	3.0000	1.9308
2	182.9710	0.4520	2.4996	217.3229	0.1794	1.0009	2.5840
3	145.6160	0.9000	1.6670	191.3630	0.2580	1.0000	2.6450
4	157.2560	0.9000	1.2430	171.6070	0.2580	1.0260	3.1230
5	166.5110	0.9000	1.0000	191.3630	0.2580	1.0000	3.4585

Table 3: The optimized turning parameters.

Algorithms		Unit cost (\$)
FEGA	[11]	2.3084
SA/SP	[10]	2.2795
PSO	[12]	2.2721
GA	[13]	2.2538
SS	[14]	2.0754
GA-based approach	[3]	2.0298
GA-SQP		1.9308

Table 4: Results of optimization using different algorithms.

SQP can tackle the optimization of multi-pass turning operations problem efficiently to achieve better results in reducing the unit production cost.

Conclusion

This paper presents a hybrid GA-SQP optimization for solving the multi-pass turning operations problem. The results obtained from comparing the hybrid GA-SQP with those taken from recent literature prove its effectiveness.

The results of the hybrid GA-SQP are compared with results of genetic algorithms, simulated annealing, particle swarm intelligence, scatter search, ant colony approaches.

The hybrid GA-SQP obtain near optimal solution, it can be used for machining parameter selection of complex machined parts that require many machining constraints. Also, it can be extended to solve the other metal cutting optimization problems such as milling, drilling etc.

References

- Sardiñas RQ, Santana MR, Brindis EA (2006) Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. Eng Appl Artif Intell 19: 127–133.
- Wang YC (2007) A note on optimization of multi-pass turning operations using ant colony system. International Journal of Machine Tools and Manufacture 47: 2057-2059.
- Shutong XIE, Yinbiao GUO (2011) Intelligent selection of machining parameters in multi-pass turnings using a GA-based approach. Journal of Computational Information Systems 5: 1714-1721.
- Kilic SE, Cogun C, Sen DT (1993) A computer-aided graphical technique for the optimization of machining conditions. Computers in Industry 22: 319-326.
- Ermer DS, Patel DC (1974) Maximization of production rate with constraints by linear programming and sensitivity analysis. Proceeding of Second North American Metalworking Research Conference WI.
- Agapiou JS (1992) The optimization of machining operations based on a combined criterion. I. The Use of Combined Objectives in Single-Pass Operations. Journal of Engineering for Industry 114: 500-507.
- Shin YC, Joo YS (1992) Optimization of machining conditions with practical constraints. Int J Production Research 30: 2907-2919.
- Ermer DS (1971) Optimization of the constrained machining economics problem by geometric programming. ASME Journal of Engineering for Industry 93: 1067-1072.
- Petropoulos PG (1973) Optimal selection of machining rate variable by geometric programming. Int J Production Research 11: 305-314.
- Chen MC, Tsai DM (1996) A simulated annealing approach for optimization of multi-pass turning operations. Int J Production Research 34: 2803–2825.
- Chen MC, Chen KY (2003) Optimization of multi-pass turning operations with genetic algorithms: a note. Int J Production Research 41: 3385-3388.
- Srinivas J, Giri R, Yang SH (2009) Optimization of multi-pass turning using particle swarm intelligence. Int J Adv Manuf Technol 40: 56-66.
- Sankar RS, Asokan P, Saravanan R, Kumanan S, Prabhakaran G (2007) Selection of machining parameters for constrained machining problem using evolutionary computation. Int J Adv Manuf Technol 32: 892-901.
- Chen MC (2004) Optimizing machining economics models of turning operations using the scatter search approach. Int J Production Research 42: 2611-2625.
- Vijayakumar K, Prabhakaran G, Asokan P, Saravanan R (2003) Optimization of multi-pass turning operations using ant colony system. Int J Adv Manuf Technol 43: 1633–1639.
- Oktem H, Erzurumlu T, Erzincanli F (2006) Prediction of minimum surface roughness in end milling mold parts using neural network and genetic algorithm. Materials and Design 27: 735–744.
- Rao SS (2009) Engineering optimization theory and practice. Fourth edition, ISBN 978-0-470-18326-6 (cloth).
- Cus F, Balic J (2003) Optimization of cutting process by GA approach. Robot Comput Integr Manuf 19: 113-121.
- Keuawan S (2009) Modelling and optimal design in railway applications. These of doctorate Ecole Centrale de Lille.
- Holland JH (1992) Adaptation in natural and artificial systems. MIT Press Cambridge, MA, USA.
- Hiwa S, Hiroyasu T, Miki M (2007) Hybrid optimization using direct GA and SQP for global exploration. IEEE Congress on Evolutionary Computation 1709-1716.
- Lakshmanan V (2000) Using a genetic algorithm to tune a bounded weak echo region detection algorithm. J Appl Meteor 39: 222-230.