Cutting Parameters Optimization and Constraints Investigation for Turning Process by GA with Self-Organizing Adaptive Penalty Strategy*

Nafis AHMAD**, Tomohisa TANAKA** and Yoshio SAITO**

For efficient use of machine tools at optimum cutting condition, it is necessary to find a suitable optimization method, which can find optimum feasible solution rapidly and explain the constraints as well. As the actual turning process parameter optimization is highly constrained and nonlinear, a modified Genetic Algorithm with Self Organizing Adaptive Penalty (SOAP) strategy is used to find the optimum cutting condition and to get clear idea of constraints at the optimum condition. Unit production cost is the objective function while limits of the cutting force, power, surface finish, stability condition, tool-chip interface temperature and available rotational speed in the machine tool are considered as the constraints. The result shows that our approach of GA with SOAP converges quickly by focusing on the boundary of the feasible and infeasible solution space created by constraints and also identifies the critical and non-critical constraints at the optimum condition.

Key Words: Turning, Cutting Parameters, Optimization, GA, SOAP

1. Introduction

Due to the high cost of NC/CNC machine tools, compared with their conventional counterparts, there is an economic need to operate these machine tools as efficiently as possible in order to obtain the required payback. Even in case of conventional machine tools, by using optimum cutting parameters, efficiency and profitability can be improved significantly. Thus the cutting conditions play an important role in the efficient use of these machine tools. Since the cost of turning on these machines is sensitive to the cutting conditions, it is necessary to determine the optimum cutting parameters before a part is put into production.

Many research works have been conducted to optimize the turning process parameters before. GA is widely used by the researchers as it performs better than other traditional approaches. Ahmad et al.⁽²⁾, Amiolemhen et al.⁽⁴⁾, Saravanan et al.⁽⁵⁾ Onwubolu et al.⁽⁷⁾, Reddy et al.⁽⁹⁾

used GA while Meng et al.⁽⁸⁾ used machining theory to optimize the cutting parameters. Saravnan et al. also used Simulated Annealing (SA) and Vijayakumar et al.⁽⁶⁾ used Ant Colony approach. Optimization is performed by minimizing the unit production cost or production time. Optimization is also done considering single or multipass turning operation. They determined optimum cutting parameters for different combination of constraints and situations. However a set of optimum cutting parameters does not reflect the actual cutting condition properly. We need clear picture of the constraints at the optimum condition for better understanding and to identify the critical and non-critical constraints as explained by Ahmad et al.⁽¹⁾

It is also necessary to reduce time for optimization process, which can be performed by focusing search on the boundary of the feasible and infeasible solution space because the optimum parameters lay on the boundary of constrained optimization problem like turning process⁽⁸⁾.

As turning parameter optimization is highly constrained, penalty is applied when any solution violates any constraint during optimization process by GA. Traditionally the penalty applied is proportional to the degree of constraint violation, so that all the infeasible solutions in future generation moves toward the feasible region. However when the solutions move inside the feasible region, they move away from the optimum solution, because in

* Received 28th October, 2005 (No. 05-4232)

** Department of Mechanical and Control Engineering, Tokyo Institute of Technology, 2-12-1-I1-33 O-okayama, Meguro-ku, Tokyo 152-8552, Japan.
E-mail: ahmadn@mep.titech.ac.jp;
tanakat@mep.titech.ac.jp;
saitoy@mep.titech.ac.jp

constraint problems like turning parameter optimization, the optimum point lies on the border of the feasible and infeasible solution space as mentioned earlier.

As a result the traditional approach needs more time to reach the optimum solution. In this work we investigated a Self-Organizing Adaptive Penalty (SOAP)⁽³⁾ approach suitable for such constrained and nonlinear problem. This approach is more suitable than the traditional approach, because it keeps the feasible and infeasible solutions through out the GA process in such a way that focus for the optimum value is maintained on the border between feasible and infeasible region.

We investigated the optimum cutting parameters and constraints values obtained by two different approaches: GA with SOAP and GA without SOAP. The result shows that GA with SOAP gives better result than the traditional approach and also indicates the constraint(s) which is critical and which is redundant for a specific problem. The feasible ratios of the solutions at each generation for different constraints show the proportion of the solution feasible at different stages of searching process.

It is also important to note that, optimization by previous work^{(2), (4)–(7), (9)} does not ensure that the optimum solution will not violate any constraint. The optimum parameters obtained by these work will be very close to the actual optimum condition, but there is chance that it may be on the infeasible solution space. Thus a small violation of the constraint is allowed in the previous works. However our approach will ensure that the optimum solution will always be feasible and will not violate any constraint.

2. Nomenclature

Nomenclature with constraint limits used in this work and constants values are also provided here. Constants and exponents are taken from the work of Onwubolu et al.⁽⁷⁾

UC : unit production cost except material cost (\$/piece)
 C_M : cutting cost by actual time in cut (\$/piece)
 C_I : idle cost due to loading and unloading operations and tool idle time (\$/piece)
 C_R : tool replacement cost (\$/piece)
 C_T : tool cost (\$/piece)
 V, V_L, V_U : cutting speeds, lower and upper bound of cutting speed (m/min), [$V_L = 50, V_U = 500$]
 f, f_L, f_U : feed rates, lower and upper bound of feed rate (mm/rev), [$f_L = 0.1, f_U = 1.5$]
 d : depth of cut (mm), [$d = 1.0$]
 D, L : diameter and length of the workpiece (mm), [$D = 50, L = 500$]
 Wp : workpiece number [$Wp = 100$]
 k_o : direct labor cost + overhead (\$/min), [$k_o = 0.5$]
 k_t : cutting edge cost (\$/edge), [$k_t = 2.5$]
 t_m : machining times (min)
 t_c, t_i : preparation time for loading/unloading, idle tool motion time such as, tool travel and tool ap-

proach/departure time, and total machine idle time (min), [$t_c = 0.75$]

t_e, t_r : tool exchange and tool replacement times (min), [$t_e = 1.5$]

h_1, h_2 : constants relating to tool travel, approach/departure time (min), [$h_1 = 7 \times 10^{-4}, h_2 = 0.3$]

T, T_U, T_L : tool life, upper and lower bounds for tool life (min) [$T_U = 45, T_L = 25$]

p, q, r, C_o : constants of tool-life equation [$p = 5, q = 1.75, r = 0.75$ and $C_o = 6 \times 10^{11}$] [refer to Eq. (2)]

$SR, SR_{\max}$: surface roughness, maximum allowable surface roughness (μm) [$SR_{\max} = 100$]

R : nose radius of the cutting tool (mm), [$R = 1.2$]

F, F_U : cutting force and maximum allowable cutting force (N), [$F_U = 1980$]

k_1, μ, ν : constants of cutting force equation, [$k_1 = 108, \mu = 0.75, \nu = 0.95$] [refer to Eq. (13)]

P, P_U, η : power, power of the machine tool (kW), power efficiency, [$P_U = 5, \eta = 0.85$]

λ, ν : constants related to expression of stable cutting region, [$\lambda = 2, \nu = -1$]

SC : limit of stable region cutting constraint, [$SC = 140$]

Q, Q_U : chip-tool interface temperature and its maximum allowable limit ($^{\circ}\text{C}$), [$Q_U = 1000$]

k_2, τ, ϕ, δ : constants related to Q , [$k_2 = 132, \tau = 0.4, \phi = 0.2, \delta = 0.105$] [refer to Eq. (13)]

N, N_L, N_U : rpm and its lower and upper limit in the machine tool, [$N_{\min} = 100, N_{\max} = 4000$]

X_t, X_{soap} : penalty function for traditional approach and SOAP strategy

Z : a big constant used to calculate the fitness for traditional approach

$G, G_{\max}$: generation, maximum generation, [$G_{\max} = 1000$]

p_{cross}, p_{mut} : probability of crossover and mutation, [$p_{cross} = 0.8, p_{mut} = 0.02$]

m : number of constraints

Δg : constraint violation

QR_{obj}, QR_{con-j} : interquartile range of the objective function and the j th constraint

r_j^G, r_j^0 : the penalty parameter for the j th constraint at generation G and initial population

fr_j^G : feasible ratio for the j th constraint at generation G

3. Mathematical Model for Unit Manufacturing Cost

The objective of this work is to determine the optimal cutting parameters including feed rate and cutting speed for turning a cylindrical workpiece by a single pass operation. As we are considering a single pass operation, the depth is constant in this case. The pass may be a rough pass or a finish pass depending on the surface fin-

Fig. 1 Total cost at different feed rate and cutting speed for depth of cut 0.5 mm

ish constraint. Different approaches are available to define the objective function for optimization of turning parameters. The objective function may be the unit manufacturing cost, manufacturing time or a combination of these two parameters. In this work we have considered the manufacturing cost as the objective function, which is calculated by Eq. (1).

Cost components that are involved in unit production cost (UC) are the cost during actual machining (C_M), machine idle time cost (C_I), tool replacement cost (C_R), tool cost (C_T)⁽⁷⁾. As we are considering a single pass turning operation, time for loading/unloading the workpiece, approach and over travel are not included in the cost calculation. This cost component is constant for both the penalty approach and has no influence on the optimization process. For a depth of cut of 0.5 mm UC varies with feed rate and cutting speed as shown in Fig. 1. Cost increases while both feed rate and cutting speed are approaching near their respective limits.

Cutting cost is involved only when the cutting process is performed. This covers the cost for the labor and the machine usage. Tool replacement cost depends on the tool life and machining time. In this work the extended Taylor tool-life equation is used⁽⁷⁾. The tool replacement cost can be expressed in terms of tool life (T), time required to exchange a tool (t_e) and machining time (t_m). Tool cost (C_T) occurs due to the wear of the tool. In this work, unit production cost is

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

Where

$$C_M = k_o t_m, \quad C_R = k_o \frac{t_e t_m}{T}, \quad C_T = k_T \frac{t_m}{T},$$

$$t_m = \frac{\pi DL}{1000 V f} \text{ and } T = \frac{C_o}{V^p f^p d^r} \quad (2)$$

4. Process Parameters Optimization Model

As mentioned earlier, the objective function in this work is the machining cost calculated by Eq. (1). The

Fig. 2 Roughness at different feed rate

constraints that are considered in the optimization of the turning parameters are: limits of tool-life, maximum allowable cutting force, power, stable cutting region, chip-tool interface temperature, surface finish and available rotational speed in the machine tool. Constraint limits and constant values are presented in section 2 with respective nomenclatures.

The optimization model can be expressed by Eq. (3) to Eq. (13). Here Eq. (3) is for the objective function, Eqs. (4)–(10) are for constraints and Eqs. (11)–(12) are for the independent variables.

$$\text{Minimize } UC(f, V) \quad (3)$$

Subject to

$$T_L \leq T \leq T_U \quad (4)$$

$$F \leq F_u \quad (5)$$

$$P \leq P_u \quad (6)$$

$$Q \leq Q_u \quad (7)$$

$$N_L \leq N \leq N_u \quad (8)$$

$$\frac{f^2}{8R} \leq SR \quad (9)$$

$$V^\lambda f d^\nu \geq SC \quad (10)$$

$$V_L \leq V \leq V_U \quad (11)$$

$$F_L \leq f \leq f_U \quad (12)$$

Other relations are

$$F = k_1 f^\mu d^\nu, \quad P = \frac{FV}{6120\eta} \text{ and } Q = k_2 V^\tau f^\phi d^\delta \quad (13)$$

The model formulated above is non-linear, constrained optimization problem with multiple continuous variables referred to as the machining parameters i.e. feed rate and cutting speed in this work. Relations of these machining parameters to the constraints such as surface roughness, cutting force, tool-chip interface temperature and power are presented in Figs. 2–5 respectively.

For a fixed nose radius, surface roughness depends only on the feed rate, while cutting force depends of the

Fig. 3 Cutting force at different feed rate and depth of cut

Fig. 4 Tool chip interface temperature at different feed rate and cutting speed for depth of cut 1.0 mm

Fig. 5 Power at different feed rate and cutting speed for depth of cut 1.0 mm

depth of cut and feed rate. Tool-chip interface temperature depends on all three cutting parameters: feed rate, cutting speed and depth of cut. Power also depends on these three cutting parameters. However Figs. 4 and 5 represent for a specific depth of cut of 1.0 mm. To optimize such a constrained problem we proposed genetic algorithm optimization technique with SOAP, which is described in the following section.

5. Genetic Algorithm Methodology

Genetic Algorithms (GAs) are search algorithms based on the conjecture of natural selection and genetics. GAs start with generating a random population, check the fitness of solutions in the population, create new population by selection, perform cross-over and mutation operation. The process continues until the termination condition is met. As GAs are explained in other previous works in detail^{(2), (5)–(7)}, we are focusing our discussion on determining penalty and selecting 'elite' in both GA without SOAP and GA with SOAP in the following paragraphs.

5.1 Penalty Calculation Procedure in GA process

During the process of GA when any constraint is violated, the solution becomes infeasible. To keep the solution feasible, penalty is applied to the objective function when any constraint is violated, thus the fitness degrades in relation to the degree of constraint violation.

In the traditional approach, penalty (X_t) is proportional to the degree of constraint violation by any solution in the population. The proportional constant varies from problem to problem. Penalty [Eq. (14)] and fitness function [Eq. (15)] are calculated as described by Goldberg⁽⁹⁾. Here Z is big constant. As UC and X_t decreases the value of Φ increases. When any solution exceeds the limit of any constraint the penalty increases. As a result the chance for that solution to survive for the next generation decreases. In traditional approach, defining fitness function is difficult because the value of constant Z will vary from problem to problem.

$$X_t = 100 \times \sum \Delta g_j \quad j=1 \text{ to } m \quad (14)$$

$$\text{Maximize } \Phi(f, V) = Z - UC(f, V) - X_t \quad (15)$$

$$\text{for maximum limits } \Delta g_j = 1 - g_{\max}/g_j, \text{ for minimum limits } \Delta g_j = 1 - g_j/g_{\min} \quad (16)$$

In GA for turning parameters optimization we propose SOAP as a penalty strategy, because it is adaptive to each generation, independent penalty adjustability for each constraint, problem dependent parameter free and self maintaining feasible and infeasible solution ratio in the population. After adding the penalty function, the objective function becomes:

$$\text{Minimize } \Phi(f, V, G) = UC(f, V) + X_{\text{soap}}(f, V, G) \quad (17)$$

here

$$X_{\text{soap}}(f, V, G) = \frac{100 + G}{100} \times \frac{1}{m} \times \sum r_j^G \times \Delta g_j \quad (18)$$

Here, Δg_j is calculated by using Eq. (16). The penalty parameter, r_j^G is updated by using Eqs. (19) and (20) in the following manner:

$$r_j^G = r_j^{G-1} \times \left[1 - \frac{(fr_j^G - 0.5)}{5} \right] \quad G \geq 1 \quad (19)$$

Table 1 Optimum parameters at different generation with constraints and total cost value

Generation	Method*	f (mm/rev) $0.1 \leq f \leq 1.5$	V (m/min) $50 \leq V \leq 500$	F (kgf) $F \leq 200$	P (kW) $P \leq 5kW$	SR (μ m) $SR \leq 100$	Q ($^{\circ}$ C) $Q \leq 1000$	SC $SC \leq 140$	rpm $100 \leq rpm \leq 4000$	T (min)	t_m (min)	LC (\$)
10	S	0.9767	115.10	106.10	2.348	99.363	876.91	12939	732.75	30.95	69.867	42.269
	T	0.9353	121.19	102.72	2.393	91.122	887.46	13737	771.52	25.80	69.291	43.374
50	S	0.9767	115.10	106.10	2.348	99.363	876.91	12939	732.75	30.95	69.866	42.269
	T	0.9621	119.29	104.91	2.406	96.411	886.86	13691	759.42	26.58	68.437	42.587
500	S	0.9793	119.16	106.32	2.435	99.889	889.62	13904	758.60	25.91	67.308	42.097
	T	0.9786	114.45	106.27	2.338	99.764	875.28	12819	728.61	31.73	70.121	42.243
1000	S	0.979	119.00	106.30	2.435	99.889	889.62	13905	758.60	25.91	67.308	42.097
	T	0.9750	119.20	106.00	2.427	99.030	888.85	13845	758.60	26.10	67.599	42.216

*S-GA with SOAP, T-GA without SOAP

At the first generation ($G = 1$), the initial penalty parameter for j th constraint is defined by using interquartile range of the objective function values and interquartile range of j th constraint of the initial population as Eq. (20). To find the interquartile range, all the solutions in the initial population is sorted based on the objective function value, then the difference between the third quartile value and first quartile value for objective function and constraint values are taken.

$$r_j^0 = \frac{QR_{obj}}{QR_{con-j}} \quad (20)$$

Feasible ratio, fr used in Eq. (19) is the ratio of the feasible solutions in the population to the population size at any generation for different constraints. For example, if at any generation 50% of the solutions in the population exceed the maximum limit of cutting force then feasible ratio will be 0.5 for cutting force. In this work, at each generation the value of fr for different constraint is calculated to find the value of penalty parameter, r by Eq. (19). The penalty parameter at any generation also depends on the penalty parameter of the previous generation. If the current generation $fr = 0.5$, for any constraint, r will be the same as the previous generation, if $fr > 0.5$ the value of r will decrease and if $fr < 0.5$ the value of r for that constraint will increase. Thus r for all the constraint will be calculated and added to the Eq. (18). Depending on the problem, some of the constraints become critical and some of them become non-critical (redundant). For example, while optimizing the turning parameters, if we want very low surface roughness, then surface finish will be a critical constraint. In such cases the fr of surface finish constraint will tend to be 0.5. For other constraints the fr will tend to be 1.0 during the GA process.

Penalty for a solution in a population, calculated by

Eq. (18), depends on the deviation from the feasible space Δg , current generation number G and penalty function r . As the generation increases, for the same deviation from the feasible space will add more penalty at the final stage. Detail procedure for determining the feasible ratio and penalty function is available in another work by Lin et al.⁽³⁾

5.2 Selecting the 'elite'

To ensure feasibility of the optimum solution, we propose to select the 'elite' at each generation not based on the best fitness only (as happens in traditional approach of GA), but also on the feasibility of that solution. If the best fitness solution violates any constraint we go for the next best, which does not violate any constraint and keep it for the next generation. After performing all GA operators like crossover, mutation at each generation, we select the worst performing solution and replace it with the best i.e. 'elite' selected at previous generation. Thus at every generation we will get one best solution which will not violate any constraint. For two consecutive generation we compare the 'elite's and keep the better one among them. This process continues until the termination condition (1 000 generation in our case) is reached and thus we can find the best performing solution among all the solutions generated from the beginning to the end of the computation process.

6. Results

We have performed the computation using GA with SOAP and GA without SOAP for 1 000 generation with a population size of 400. The optimum result after 10, 50, 500 and 1 000 generation is presented in Table 1. The optimum cutting parameters, feed rate and cutting speed in these cases along with the objective function, total ma-

Fig. 6 Feasible ratios at different generation for different constraints

Fig. 7 Unit manufacturing cost at different generation

chining cost, machining time and the constraints value are also presented in Table 1.

As the fitness function is a combination of the actual cost and penalty, an optimum solution may be infeasible or may violate some of the constraints. For this reason we took the solution which is feasible considering all the constraint limits and whose cost is also minimum at that generation.

Figure 6 shows the feasible ratios at different generations up to 150 generations. As explained earlier, feasible ratio is the ratio of the feasible solutions to the total solutions in the population at a specific generation for a specific constraint. As the feasible ratios for cutting force, stability condition and rotational speed are 1.0 i.e. 100% of the solutions of the population was within the limits of these constraints, we did not included them in Fig. 6.

Figure 7 (a) and (b) shows the mean cost and the minimum cost for machining the parts at different generation. Figure 7 (b) represents the best solutions in the population at different generations, which do not violate any constraint.

The data for constraint limits, exponents and constants are provided with nomenclature in section 2. The Genetic Algorithm Optimization Toolbox (GAOT) devel-

oped for Matlab environment by Chris H. et al.⁽¹²⁾ is used after necessary modification to fit in this problem.

7. Discussion

Table 1 shows the constraint values for the optimum solution with their respective limits. Force, power, stability condition, rotational speed and tool-chip interface temperature constraints are well inside their limits. After 1000 generation the optimum feed rates (0.979 mm/rev and 0.975 0) and cutting speed (119 m/min and 119.2 m/min) are almost same and give almost same machining time and unit production cost while using GA with SOAP and without SOAP respectively. On the other hand at the early stage of computation, GA with SOAP gives better result than GA without SOAP. That means GA with SOAP converges to the optimum faster than GA without SOAP. Moreover, we took the best feasible solution in both the approaches at each generation to ensure the feasibility of the optimum solution. So the optimum results presented in Table 1, for both the cases do not violate any constraint. Traditionally the best taken at each generation has the lowest fitness value, but it is not possible to ensure that the best will not violate any constraint. If we do not use our approach to ensure the feasibility, the

traditional approach may result optimum solution, which will violate some constraint.

If we look into the constraints at different stages of the optimization process we find that surface roughness value is very close to its limit (100 μm). GA with SOAP finds optimum parameters with surface roughness closed to its limit. Tool life also moves from 30min at generation 10 to its lower limit [25 min] at generation 1 000 for GA with SOAP. From Fig. 6 also this fact becomes evident.

In Fig. 6 the feasible ratios for different constraint at different generations are presented. Feasible ratios for the constraints other than power, surface finish, tool-chip interface temperature and tool life are 1.0 from the beginning stage. In case of traditional approach, feasible ratios of power, surface finish and tool-chip interface temperature gradually becomes closes to 1.0 but feasible ratio for tool life slowly increased over the generations. However when we apply SOAP as the penalty strategy, feasible ratio for power and tool-chip interface temperature become 1.0 rapidly. On the contrary for surface finish and tool life, ratios try to maintain themselves between 0.0 and 1.0. It implies that the actual optimum lies on the boundary of the feasible and infeasible solution space created by these two constraints.

As in case of SOAP the search for the optimum is concentrated on the boundary of feasible and infeasible solution space created by these two constraints, the SOAP gets better result. In case of SOAP, at early stages of evolution in the genetic searches, the penalty resulting form a small degree of constraint violation is not too large to eliminate the infeasible solution, which may contain critical genes needed for an optimum solution. Near the end of the genetic search, the penalties for infeasible solution need to be increased to a level that allows the selection operations to push solutions back to feasible region so as to increase the chance of locating the true optimum.

The mean cost at different generations is shown in Fig. 7 (a). As the mean cost is calculated by using the feasible and infeasible solutions, it gives us an overall idea of the optimization process. In general the mean cost becomes higher when the solutions are far inside the feasible regions from the constraint boundary instead of being close to constraint boundary. As shown in Fig. 7 (a) the mean cost for GA with SOAP decreases faster than the mean cost in traditional approach because, in the first case the solutions move close to the boundary comparatively faster than the other approach. As the target is to get a single best solution instead of getting a mean value, we have to check the improvement of best feasible solutions at different generations. Figure 7 (b) represents the nature of improvement of the best feasible solutions at different generations, where GA with SOAP performed better than the traditional approach.

As GA with SOAP need extra computational process

to calculate penalty parameters and feasible ratios, it takes more time compared to traditional approach to finish the same number of generation with similar population size. Instead of this disadvantage, it performs faster than traditional approach when we consider the required number of generations to get a similar feasible optimum solution. GA with SOAP also gives us better idea of the constraints at the optimum condition.

8. Conclusion

This work shows that in constrained optimization problem like turning process, appropriate penalty strategy is necessary to get the optimum solution faster. SOAP can reduce the time for searching the best solution. It also performs better than traditional penalty approach by focusing on the boundary region of the feasible and infeasible solution space. From feasible ratios of constraints, the process planner gets better idea of the critical constraints for the optimum set of cutting parameters. This work can be extended for multipass turning operation and then can be continued for other machining operations like milling.

References

- (1) Ahmad, N., Tanaka, T. and Saito, Y., Investigation of Constraints for Optimization of Turning Process by GA with Self-Organizing Penalty Strategy, Proc. of the 3rd International Conference on Leading Edge Manufacturing in 21st Century, Nagoya, Japan, Vol.2 (2005), pp.319–324.
- (2) Ahmad, N., Tanaka, T. and Saito, Y., Optimization of Turning Process Parameters by Genetic Algorithm (GA) Considering the Effect of Tool Wear, Proc. of the 5th Manuf. and Machine Tool Conf. of JSME, Osaka, Japan, No.04-3, (2004), pp.305–306.
- (3) Lin, C.-Y. and Wu, W.-H., Self-Organizing Adaptive Penalty Strategy in Constrained Genetic Search, Struct Multidisc. Optim., Vol.26, No.6 (2004), pp.417–428.
- (4) Amiolemhen, P.E and Ibadode, A.O.A., Application of Genetic Algorithms—Determination of the Optimal Machining Parameters in the Conversion of a Cylindrical Bar Stock into a Continuous Finished Profile, Int. J. of Machine Tools and Manuf., Vol.44, No.12-13 (2004), pp.1403–1412.
- (5) Saravanan, R., Asokan, P. and Vijayakumar, K., Machining Parameters Optimization for Turning Cylindrical Stock into a Continuous Finished Profile Using Genetic Algorithm (GA) and Simulated Annealing (SA), Int. J. Adv. Manuf. Technology, Vol.21, No.1 (2003), pp.1–9.
- (6) Vijayakumar, K., Prabhakaran, G., Asokan, P. and Saravanan, R., Optimization of Multi-Pass Turning Operations Using Ant Colony System, Int. J. of Machine Tools and Manuf., Vol.43 (2003), pp.1633–1639.
- (7) Onwubolu, G.C. and Kumalo, T., Optimization of Multipass Turning Operations with Genetic Algorithms, Int. J. Prod. Res., Vol.39, No.16 (2001), pp.3727–3745.
- (8) Meng, Q., Arsecularatne, J.A. and Mathew, P., Calculation of Optimum Cutting Conditions for Turning Oper-

- ations Using a Machining Theory, *Int. J. Machine Tool and Manuf.*, Vol.40, No.12 (2000), pp.1709–1733.
- (9) Goldberg, D.E., *Genetic Algorithm in Search, Optimization & Machine Learning*, 3rd Indian Edition, (2000), Addison Wesley.
- (10) Reddy, B.S.V., Shunmugam, M.S. and Narendran, T.T., Optimal Sub-Division of the Depth of Cut to Achieve Minimum, Production Cost in Multi-Pass Turning Using a Genetic Algorithm, *J. of Mat. Proc. Tech.*, Vol.79, No.1-3 (1998), pp.101–108.
- (11) Kee, P.K., Development of Constrained Optimization Analyses and Strategies for Multi-Pass Rough Turning Operations, *Int. J. Mach. Tools Manuf.*, Vol.36, No.1 (1996), pp.115–127.
- (12) Chris, H., Jeff, J. and Mike, K., A Genetic Algorithm for Function Optimization: A Matlab Implementation, NCSU-IE TR 95-09, (1995).
-