Int J Adv Manuf Technol (2004) 24: 727–732
DOI 10.1007/s00170-003-1789-5

ORIGINAL ARTICLE

Z.G. Wang · Y.S. Wong · M. Rahman

Optimisation of multi-pass milling using genetic algorithm and genetic simulated annealing

Received: 22 January 2003 / Accepted: 6 May 2003 / Published online: 21 April 2004

Abstract The selection of optimal machining parameters plays an important part in computer-aided manufacturing. The optimisation of machining parameters is still the subject of many studies. Genetic algorithm (GA) and simulated annealing (SA) have been applied to many difficult combinatorial optimisation problems with certain strengths and weaknesses. In this paper, genetic simulated annealing (GSA), which is a hybrid of GA and SA, is used to determine optimal machining parameters for milling operations. For comparison, basic GA is also chosen as another optimisation method. An application example that has previously been solved using geometric programming (GP) method is presented. The results indicate that GSA is more efficient than GA and GP in the application of optimisation.

Keywords Genetic algorithm · Genetic simulated annealing · Milling

1 Introduction

The determination of efficient machining parameters has been a problem confronting manufacturing industries for nearly a century, and is still the subject of many studies. Optimal machining parameters are of great concern in manufacturing environments, where economy of machining operation plays a key role in the competitive market, and computer numerically controlled (CNC) machines are extensively employed. Most work done on the optimisation of cutting conditions in machining is mainly focused on turning operations, while multi-pass milling has received relatively little attention with regard to the optimisation of cutting parameters.

Although the importance of using optimal cutting parameters was identified in the early 1900s, advances in the development of optimisation strategies have been very slow [1], since the problem is too complex due to the nonlinear dependence of machining variables. Several types of methods have been used for the optimisation of cutting parameters. The direct search method is one of the most popular mathematical optimisation methods. It computes the first derivative of an objective function and sets it to zero, where the function becomes a maximum or minimum [2, 3]. The second derivative of the objective function is then used to determine the maxima or minima. Clearly, therefore, the objective function in classical and direct optimisation must be continuous and twice differentiable. However, this requirement is generally not met in real-world problems. Rad-Tolouei and Bidhendi [4], Wang [5–7], Armarego et al. [8], and Kilic et al. [9] used graphical techniques for the optimisation of machining conditions by mapping the relevant constraints and objective functions in the planes. The constrained optimisation strategies were also used for selecting the optimal cutting conditions [8, 10]. Kayacan et al. [11] and Agapiou [12] used knowledge base and machining topology; Jang [13] presented a unified optimisation approach for the selection of the machining parameters. Wang [14] used a neural network based approach to optimise cutting parameters. However, optimisation using these methods often ends in local minima or fails to converge on a result. Sönmez et al. [1], Ermer [15], Wang [16], Agapiou [12] and Shin and Joo [17] used the dynamic programming (DP) optimisation method. DP can solve both continuous and discrete variables and yield a global optimal solution. However, if the optimisation problem involves a large amount of independent parameters with a wide range of values such as cutting parameters, the use of DP is limited. In addition, the geometric programming (GP) method was used for optimisation by Sönmez et al. [1], Wang [16] Petropoulos [18] and Jha [19]. Jha has attempted to optimise cutting parameters in milling by GP and concluded that the GP-based program was very slow to produce good results [19]. Dereli et al. [20] indicated that it needed long execution times for good outcomes of the objective function by Sönmez et al., who developed a system for the constrained optimisation of cutting parameters to be used in the multi-pass plain and face milling operations using DP plus GP. The abovementioned optimisation techniques either tend to result in local

Z.G. Wang · Y.S. Wong (✉) · M. Rahman
Department of Mechanical and Production Engineering,
National University of Singapore,
10 Kent Ridge Crescent, Singapore, 119260
E-mail: mpewys@nus.edu.sg

minima or take a long time to converge on a reasonable result. In order to overcome the long execution time using DP plus GP, Dereli et al. [20] used genetic algorithm to develop an optimisation system that had better performance.

Genetic algorithm (GA) and simulated annealing (SA) are two new optimisation approaches that have been developed independently. Both of these algorithms are probabilistic search algorithms that are capable of finding globally optimal results to complicated optimisation problems. For application in machining processes, besides Dereli et al., Shunmugam et al [21] used GA to optimise the multi-pass face-milling and obtained optimal cutting parameters. Liu et al. [22] improved the convergence speed of traditional GA and obtained good results by defining and changing the operating domain of GA. GA and SA have their strengths and weaknesses. In this paper, a new method generic simulated annealing (GSA) that combines the recombinative power of GA and local selection of SA is presented to optimise the cutting parameters for milling operations. The following section describes and compares GA and GSA.

2 State of the art

2.1 Genetic algorithms

In a GA approach to solve combinatorial optimisation problems, a population of candidate solutions is maintained. To generate a new population, candidate solutions are randomly paired. For each pair of solutions, a crossover operator is first applied with a moderate probability (crossover rate) to generate two new solutions. Each new solution is then modified using a mutation operator with a small probability (mutation rate). The resulting two new solutions replace their parents in the old population to form a temporary new population. Each solution in the temporary population is ranked against other solutions based on a fitness criterion. A roulette wheel process is then used to determine a new population identical in size to the previous population, such that higher-ranked candidates are allowed to assume higher priority in the new population. GA iterates over a large number of generations and, in general, as the algorithm executes, solutions in the population become fitter, resulting in better candidate solutions. Last but not least, GA is a search strategy that is well suited for parallel computing.

2.2 Genetic simulated annealing

GA and SA are both independently valid approaches toward problem solving with certain strengths and weaknesses. While GA can begin with a population of solutions in parallel, it suffers from poor convergence properties. SA, by contrast, has better convergence properties, but it cannot easily exploit parallelism.

In order to retain the strengths of GA and SA, GSA blends both approaches into a single approach. GSA is naturally parallel by exploiting the population-based model and crossover-mutation operator of GA by creating a multiplicity of candidate solutions. At the same time, GSA employs the temperature gradient property of SA by using a local acceptance policy based on the fitness of a new solution compared to its parent, and a probability based on a global temperature gradient [23]. By using this approach, the algorithm maintains the parallelism of GA while overcoming its poor convergence weakness by following a temperature schedule as in SA. It has been shown that GSA can perform better for optimisation problems than either GA or SA separately [24].

GSA has been successfully applied to optimisation problems like wire routing, scheduling, transportation, etc. However, there is no clear understanding on how to blend GA and SA together and use it to find global optima for problems with many nonlinear constraints. This paper presents the use of GSA to optimise cutting parameters of milling process with nonlinear constraints.

3 Optimisation of end milling

3.1 Objective function

The minimum production time has been chosen as the objective function. Production time for a component is the total time required to produce a component and is composed of the following items: (i) Preparation time T_p, (ii) Loading/unloading time T_L, (iii) Process adjustment and quick return time T_a, (iv) Machining time T_m and (v) Tool change time per component T_c (In the single-tool optimisation approach, tool changing time is considered to be related to the frequency of tool replacement). Therefore, the total production time per component can be represented by the following equations: $T_{pr} = T_p + T_L + T_a + T_m + T_c$.

For a multi-pass operation in which N_p passes are required to remove the total depth of cut:

$$T_{pr} = \frac{T_s}{N_b} + T_L + \sum_{i=1}^{N_p} \left(T_{a_i} + T_{m_i} + T_d \frac{T_{m_i}}{T} \right)$$

If machining time (T_m) and tool life (T) is expressed in terms of cutting speed and feed rate, then T_{pr} can be represented in the following form [1].

$$T_{pr} = \frac{T_s}{N_b} + T_L + N_p T_a + \sum_{i=1}^{N_p} \left(\frac{\pi D L}{f_{Z_i} z 1000 v_i} + \frac{T_d \pi L v_i^{1/m-1} a_i^{e_v/m} f_{z_i}^{u_v/m-1} B^{r_v/m} z^{n_v/m-1} \lambda_s^{q_v/m}}{1000 C_v^{1/m} D^{b_v/m-1} (B_m B_h B_p B_r)^{1/m}} \right)$$

where B_m, B_t, B_h, B_p, are correction coefficients of tool life equation; D, outer diameter of the cutter (mm); L, length of cut (mm); v, cutting speed (m/min); z, number of teeth on the cutter; f_z, feed rate per tooth (mm/tooth); C_v, a constant taking into account the influence of all factors that appear separately in the tool life formula: λ_s, cutting inclination angle (degrees); a, depth of cut (mm); N_b, total number of parts in the batch.

Table 1. Constraints and their expressions in terms of common variables

	Constraint	Expression in variables
1	Feed rate constraint	$f_{z\min} \leqslant f_z \leqslant f_{zmaz} = \frac{f_{\max}}{zN_{\max}}$
2	Cutting speed constraint	$v_{\min} \leqslant v \leqslant v_{\max} = \frac{\pi DN_{\max}}{1000}$
3	Horsepower constraint	$C_{zp} Bz D^{bz} a^{ez} v f^{uz} \leqslant P_m \eta 6120$
4	Arbor strength constraint	$C_{zp} Bz D^{bz} a^{ez} f^{uz} \leqslant \frac{0.1 k_b d_a^3}{0.08 L_a + 0.65\sqrt{(0.25L_a)^2 + (0.5\alpha D)^2}}$
5	Arbor deflection	$C_{zp} Bz D^{bz} a^{ez} f^{uz} \leqslant \frac{4Eed_a^4}{L_a^3}$

where P_m, nominal motor power (kW); η, overall efficiency; n, spindle speed (rpm); B, milling width (mm); C_{zp}, Constant of the cutting force equation; d_a, arbor diameter (mm); e, permissible values of arbor deflection; E, Modulus of elasticity of arbor material (kg/mm^2); k_b, permissible bending stress of the arbor material (kg/mm^2); L_a, arbor length between supports (mm)

Since the machining parameters do not influence the set-up time, loading and unloading time and process adjustment and quick return time, the final optimisation model per operation becomes:

$$T'_{pr} = \frac{\pi DL}{f_{Z_i} z 1000 v_i} + \frac{T_d \pi L v_i^{1/m-1} a_i^{e_v/m} f_{z_i}^{u_v/m-1} B^{r_v/m} z^{n_v/m-1} \lambda_s^{q_v/m}}{1000 C_v^{1/m} D^{b_v/m-1} (B_m B_h B_p B_r)^{1/m}}$$

where $b_v, b_z, e_v, e_z, m, n_v, n_z, q, q_v, r_v, r_z, u_v, u_z,$ are determined empirically.

3.2 Constraints

For a meaningful optimisation of the machining process, there exists a number of constraints that must be satisfied. The constraints are presented in Table 1 [2].

3.3 An application example [1]

We now illustrate the proposed methodology based on the GA and GSA using the example taken from Sönmez et al. [1]. Specifications of the required parameters and values of the constants are given as follows: Type of machining is plain milling, $P_m = 5.5$ kW, $\eta = 0.7$, $D = 27$ mm, $L_s = 210$ mm, $k_b = 140$ MPa $= 14.27$ kg/mm^2, $k_l = 120$ MPa $= 12.23$ kg/mm^2, $E = 200$ GPa $= 20,387$ kg/mm^2, Spindle speed ranges: $31.5 \sim 2000$ rpm, feed rate ranges: $14 \sim 900$ mm/min, Tool material is HSS, $D = 63$ mm, $z = 8$, workpiece material is structural carbon steel ($C \leqslant 0.6\%$), tensile strength is 750 MPa, brinell hardness is 150, $L = 160$ mm, $B = 50$ mm, $a = 5$ mm, $T_L = 1.5$ min, $T_s = 10$ min, $T_c = 5$ min, $T_a = 0.1$ min/part, $N_b = 100$, $\lambda_s = 30°$.

4 The algorithms and results

4.1 Genetic algorithm procedure

In the optimisation, GA uses the objective function (production time) as the fitness function to measure the goodness of the chromosomes. New chromosomes of feed and cutting speed are generated from the initial population by transitionally using the genetic operators, crossover and mutation. The pseudo code in Fig. 1 gives an overview of a traditional GA.

Following selection, crossover, and mutation, the new population is ready for its next evaluation. This evaluation is used to build the probability distribution (for the next selection process), i.e., for a construction of a roulette wheel with slots sized according to current fitness values. The rest of the evolution is just cyclic repetition of the above steps.

4.2 Genetic simulated annealing algorithm

One of the key problems in using GA is the handling of problem constraints. There are three common ways of handling GA constraints [25]: feasible individuals, repair algorithms and the use of penalties. The last approach was used successfully for numeri-

```
begin
  t = 0
  initialize P(t)
  evaluate P(t)
  while (not termination condition) do
  begin
    t = t + 1
    select P(t) from P(t-1)
    reproduce P(t)
    mutate P(t)
    evaluate P(t)
  end
end
```

Fig. 1. The pseudo code of genetic algorithms

cal optimisation problems, and so it is chosen as GSA constraints handing method in this paper.

GSA introduces the concept of an SA temperature to the traditional GA operations of crossover and mutation, as illustrated in Fig. 2.

In GA, two chromosomes are replaced by a pair of their offspring after the crossover operator, while in GSA, after the crossover operation, there are four chromosomes from which two chromosomes are then selected to form the new population. The selection criterion assumes that strings with higher fitness value should have a greater probability of surviving to the next generation. Individuals that have a fitness value less than the best obtained are not necessarily discarded, but are instead selected with a probability proportional to the current temperature (as in simulated annealing).

4.3 Optimisation results

It is believed that a large population size helps in better schema processing and thus reduces the chances of premature convergence. Therefore the population size in this optimisation is selected as 512, and the crossover probability and mutation probability are selected to be 0.7 and 0.01 respectively, with the initial temperature $T = 100$ and cooling rate $\alpha = 0.7$. The results for the example are summarised in Tables 2, 3 and 4, which show the best cutting parameters at different depth of cut by using three optimisation methods.

From Table 2, it is clear that the results computed by using GSA are better than those by GA or GP, especially when the depth of cut is 1 mm or 4 mm. In [1], Sönmez et al. used GP

```
begin
  t = 0
  initialize P(t) and temperature
  evaluate P(t); Penalty function are used in the evaluation
  while (not termination condition) do
  begin
    t = t + 1
    select P(t) from P(t-1)
    select individuals for reproduction from P(t)
      do reproduction
        select two unused individuals as parents P1, P2
        create two children C1, C2 by applying crossover to P1, P2
        evaluate C1, C2, P1,P2
        select two optimal individuals using SA
        mutate each one of these two optimal individuals
        evaluate them and select two optimal individuals using SA
      until all selected parents reproducing their children
    temperature = temperature × α, 0 <α < 1
  end
end
```

Fig. 2. Algorithm of genetic simulated annealing

Table 2. Computed optimal results using GP, GA and GSA

Depth of cut (mm)	Cutting speed (m/min)			Feed rate (mm/rev)			Machining time (min)		
	GP	GA	GSA	GP	GA	GSA	GP	GA	GSA
$a = 1$ mm	32.25	31.30	30.19	0.7044	0.9104	0.9584	0.2097	0.1746	0.1703
$a = 2$ mm	25.16	25.40	25.18	0.5700	0.5570	0.5695	0.3150	0.3193	0.3150
$a = 3$ mm	26.40	24.71	24.20	0.3380	0.3557	0.3670	0.4979	0.4994	0.4930
$a = 4$ mm	30.95	29.16	27.37	0.1490	0.2031	0.2187	0.9421	0.7476	0.7321

Table 3. Objective function results using GP, GA and GSA

Cutting strategy (pass distribution) (mm)	Total production time per component (min)		
	GP	GA	GSA
4-1	4.543	4.315	4.294
3-2 (Optimal strategy)	4.205	4.210	4.199
3-1-1	6.009	5.935	5.921
2-2-1	5.927	5.900	5.886
2-1-1-1	7.731	7.625	7.608
1-1-1-1-1	9.535	9.350	9.330

Table 4. Computation time using GA and GSA

Optimisation method	Calculation time (sec)			
	$a = 1$ mm	$a = 2$ mm	$a = 3$ mm	$a = 4$ mm
GA	6.821	7.982	8.442	6.479
GSA	1.552	1.622	1.763	1.653

where a means depth of cut per path

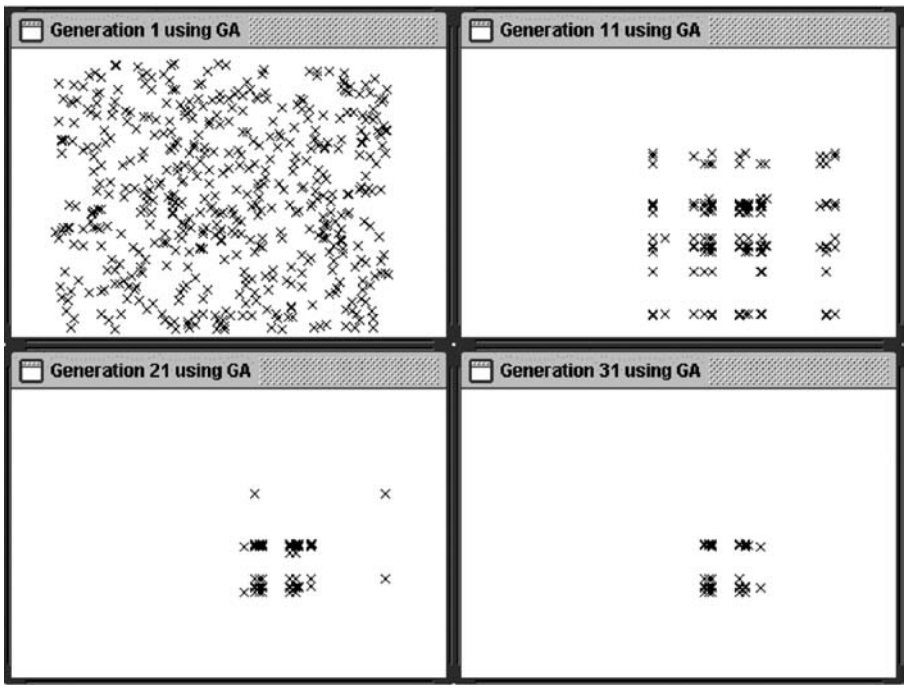

Fig. 3. Simulation of genetic algorithm

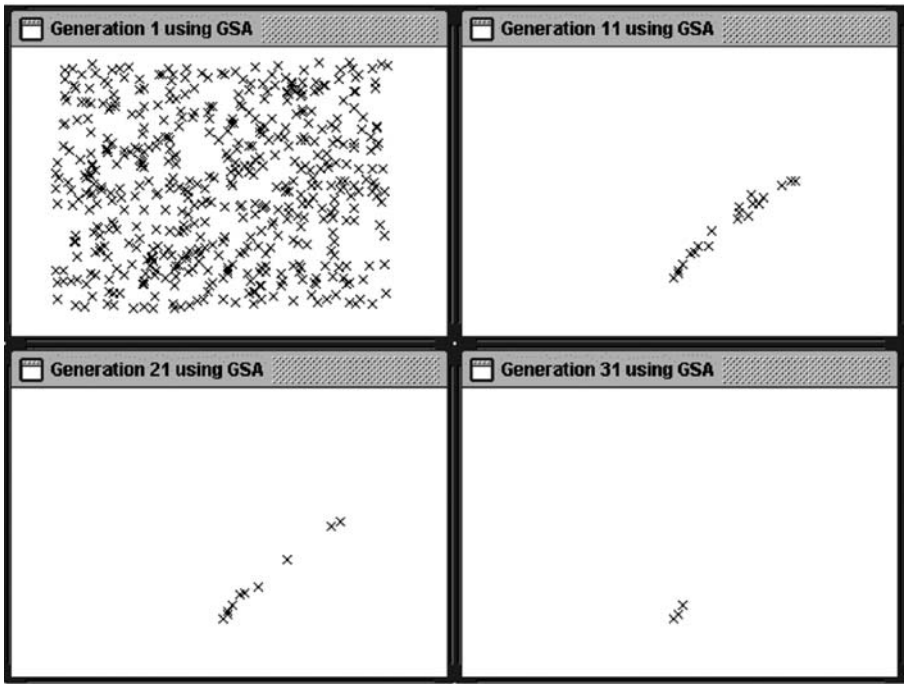

Fig. 4. Simulation of genetic simulated annealing

to optimise cutting parameters in the multi-pass plain and face milling operations, but they found that long execution times were needed to achieve good results of the objective function [20]. From Table 3, the same optimal solution can be obtained using three different optimisation methods, and it means that all these methods have the ability to find global optima. In Table 4, the computing time using GSA is less than that by using GA, because GA has certain drawback and requires very intensive computation. The computation time was recorded on a Pentium III-450 PC with 320 MB of memory.

Figures 3 and 4 depict the typical movements toward the optimal solutions. Generation 1 shows the random initialisation of the solutions in solution space, which are tracked as $\times$. While the repeatability of finding the solution is established, it is found that a reasonable solution can be reached when the generation number is about 31 by GSA. On the contrary, with GA, even when

the generation number reaches 31, the optimal solutions are still not obtainable. So it is obvious that more generations are needed to reach an optimal solution using GA than by using GSA. The fast convergence for GSA is due to the selection decisions after crossover and mutation, which can ensure that good candidates exist in the next generation so that the search space is narrowed for the next generation as shown in Figs. 3 and 4.

5 Conclusions

Although GA and SA are two most popular methods for global optimisation, they both have their strengths and weaknesses. In order to exploit their strengths and overcome their weaknesses, this paper presents a hybrid algorithm GSA to optimise the cutting conditions in plain milling (feed rate and speed) subject to a set of constraints. The proposed GSA algorithm uses a population of solutions and crossover and mutation operations among population members (as in GA) and a selection mechanism based on a temperature cooling schedule (as in SA). From the given results, it can be seen that GSA is more suitable for optimising the cutting parameters for milling operation than GA, as well as GP.

References

1. Sönmez AI, Baykasoglu A, Dereli T, Filiz IH. (1999) Dynamic optimization of multipass milling operations via geometric Programming. Int J Mach Tools Manuf 39(2):297–320
2. Gray P, Hart W, Painton L, Phillips C, Trahan M, Wagner J (2002) A survey of global optimization methods. http://www.cs.sandia.gov /opt/survey/, cited June 2002
3. Duffuaa SO, Shuaib AN, Alam M (1993) Evaluation of optimization methods for machining economics models. Comput Oper Res 20(2):227–237
4. Tolouei-Rad M, Bidhendi IM (1997) On the optimization of machining parameters for milling operations. Int J Mach Tools Manuf 37(1):1–16
5. Wang J, Armarego EJA (2001) Computer-aided optimization of multiple constraint single pass face milling operations. Mach Sci Technol 5(1):77–99
6. Wang J (1998) Computer-aided economic optimization of end-milling operations. Int J Prod Econ 54(3):307–320
7. Wang J, Kuriyagawa T, Wei XP, Guo DM (2002) Optimization of cutting conditions for single pass turning operations using a deterministic approach. Int J Mach Tools Manuf 42(9):1023–1033
8. Armarego EJA, Smith AJR, Wang J (1993) Constrained optimization strategies and CAM software for single-pass peripheral milling. Int J Prod Res 31(9):2139–2160
9. Kilic SE, Cogun C, Sen DT (1993) A computer-aided graphical technique for the optimization of machining conditions. Comput Ind 22(3):319–326
10. Kee KP (1996) Development of constrained optimization analyses and strategies for multi-pass rough turning operations. Int J Mach Tools Manuf 36(1):115–127
11. Kayacan MC, Filiz IH, Sönmez AI, Baykasolu A, Dereli T (1996) OPPS-ROT: an optimized process planning system for rotational parts. Comput Ind 32(2):181–195
12. Agapiou JS (1992) Optimization of machining operations based on a combined criterion, part 2: multipass operations. J Eng Ind Trans ASME 114(4)::508–513
13. Jang DY (1992) A unified optimization model of a machining process for specified conditions of machined surface and process performance. Int J Prod Res 30(3):647–663
14. Wang J (1993) Multiple-objective optimization of machining operations based on neural net-works. Int J Adv Manuf Technol 8(4):235–243
15. Ermer DS (1971) Optimization of the constrained machining economics problem by geometric programming. J Eng Ind 93(4):1067–1072
16. Wang J (1993) Constrained optimization of rough peripheral and end milling operations. PhD Thesis, University of Melbourne, Australia
17. Shin YC, Joo YS (1992) Optimization of machining conditions with practical constraints. Int J Prod Res 30(12):2907–2919
18. Petropoulos PG (1973) Optimal selection of machining rate variable by geometric programming. Int J Prod Res 11(4):305–314
19. Jha NK (1990) A discrete data base multiple objective optimization of milling operation through geometric programming. J Eng Ind 112(4):368–374
20. Dereli T, Filiz IH, Baykasoglu A (2001) Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms. Int J Prod Res 39(15):3303–3328
21. Shunmugam MS, Bhaskara Reddy SV, Narendran TT (2000) Selection of optimal conditions in multi-pass face-milling using a genetic Algorithm. Int J Mach Tools Manuf 40(3):401–414
22. Liu YM, Wang CJ (1999) Modified genetic algorithm based optimization of milling parameters. Int J Adv Manuf Technol 15(11):796–799
23. Shroff P, Watson DW, Flann NS, Freund RF (2002) Genetic simulated annealing for scheduling data-dependent tasks in heterogeneous environments. http://www.cs.ucsd. edu/users/berman/hcw.papers /hcwpc13_watson.ps, cited June 2002
24. Chen H, Flann NS (1994) Parallel simulated annealing and genetic algorithms: a space of hybrid methods. Proceedings of International Conference on Evolutionary Computation–PPSN III, pp428–438
25. Michalewicz Z (1996) Genetic algorithms+data structures=evolution Programs. Springer, Berlin Heidelberg New York