Chapter 5 - Simulated Annealing Based Optimization of Machining Process

In simulated annealing (SA) method, an exponential cooling schedule based on Newtonian cooling process is employed and experimentation is done on choosing the number of iterations (m) at each step. The SA approach is applied to predict the influence of tool geometry (nose radius) and cutting parameters (feed, speed and depth of cut) on surface roughness in dry turning of SS 420 materials conditions based on Taguchi’s orthogonal array method.

5.1. Simulated Annealing method (SA)

Simulated annealing was developed in 1983 to deal with highly nonlinear problems. SA appears rapidly to be becoming an algorithm of choice when dealing with financial instruments [83]. Standard nested regression and local-search methods usually are applied to develop hybrid securities, e.g. combining markets in interest rates, foreign exchange, equities and commodities by linking them via options, futures, forwards, and swaps, to increase profits and reduce risks in investments as well as in trading [84]. However, simulated annealing has been reasonably successfully used in the solution of a complex portfolio selection model [85,86]. The algorithm was able to handle more classes of constraints than most other techniques. One study has used SA on a set of several econometric problems [87], including cost functions arising in the monetary theory of exchange rate determination, a study of firm production efficiency, and a neural net model which generates chaos reputed to mirror some economic and financial series.

SA approaches the global maximization problem similar to using a bouncing ball that can bounce over mountains from valley to valley. It begins at a high "temperature" which enables the ball to make very high bounces, which helps it to bounce over any mountain to access any valley, given enough bounces. As the temperature declines the ball cannot bounce so high and it can also settle to become trapped in relatively small ranges of valleys. A generating distribution generates possible valleys or states to be explored. An acceptance distribution is also defined, which depends on the difference between the function value of the present generated valley to be
explored and the last saved lowest valley. The acceptance distribution decides probabilistically whether to stay in a new lower valley or to bounce out of it. All the generating and acceptance distributions depend on the temperature. It has been proved that by carefully controlling the rate of cooling of the temperature, SA can find the global optimum. SA's major advantage over other methods is an ability to avoid becoming trapped in local minima.

The general SA algorithm involves the following three steps. First, the objective function corresponding to the energy function must be identified. Second, one must select a proper annealing scheme consisting of decreasing temperature with increasing of iterations. Third, a method of generating a neighbor near the current search position is needed. In single objective optimization problems, the transition probability scheme is generally selected by the Metropolis and logistic algorithms [88, 89,90]. Simulated annealing (SA) is based on an analogy with the homonymous thermo dynamical process. For slowly cooled thermo dynamical systems (e.g., metals), nature is able to find the minimum state of energy, while the system may end in an amorphous state of higher energy if it is cooled quickly. This principle is expressed by the Boltzmann probability distribution.

The energy of a system in thermal equilibrium at a given temperature T is probabilistically distributed among all different states E. The parameter K is the Boltzmann constant and the exponential term is the Boltzmann coefficient. With the decrease of temperature, the Boltzmann distribution focuses on a state with lowest energy and finally as the temperature comes close to zero, this becomes the only possible state (see Fig.5.1). The system may switch to a new energy state with probability p, irrespective of whether it is higher or lower. Therefore, nature’s minimization strategy is to allow the system sometimes to go uphill as well as downhill, so that it has a chance to escape from a local energy minimum in favor of finding a better, more global minimum. However, the lower the temperature, the less likely is a significant uphill step.

![Fig. 5.1 Distribution of probability for three different temperatures](image-url)
Simulated annealing presents an optimization technique that can: (a) process cost functions possessing quite arbitrary degrees of nonlinearities, discontinuities, and stochasticity; (b) process quite arbitrary boundary conditions and constraints imposed on these cost functions; (c) be implemented quite easily with the degree of coding quite minimal relative to other nonlinear optimization algorithms; (d) statistically guarantee finding an optimal solution. Simulated annealing combines a downhill search with a random search. In order not to be trapped in a locally optimum region, this procedure sometimes accepts movements in directions other than steepest ascend or descend. The acceptance of an uphill rather than a downhill direction is controlled by a sequence of random variables with a controlled probability. Simulated annealing (SA) is a powerful stochastic search method applicable to a wide range of problems for which little prior knowledge is available. It can produce high-quality solutions for hard combinatorial optimization.

The process of slow cooling is known as annealing in metallurgical process. The simulated annealing procedure simulates this process of slow cooling of molten metal to achieve the minimum function value of surface roughness in the problem of minimization. It is a point-by-point method. The algorithm begins with an initial point and a high temperature $T$. A second point is taken at random in the vicinity of the initial point and the difference in the function values ($\Delta E$) at these two points is calculated. Suppose that initially we have a point $x_k$ in the search space and that the cost at that point is $f(x_k)$. A new point $x_{k+1}$ is randomly generated that is "nearby" in some sense; we will call this a "trial point". The cost there is $f(x_{k+1})$. Next we decide whether to move to $x_{k+1}$, that is whether to replace $x_k$ by $x_{k+1}$ as the current approximation. If $f(x_{k+1}) < f(x_k)$ then the move is definitely accepted. If $f(x_{k+1}) \geq f(x_k)$ then also the move is accepted with a probability of

$$P_{r(move_{\text{accepted}})} = \exp \left( \frac{f(x_{k+1}) - f(x_k)}{T} \right)$$

(5.1)
This completes an iteration of this simulated annealing procedure. In the next generation another point is created at random in the neighborhood of the current point and the Metropolis algorithm is used to accept or reject it. In order to simulate the thermal equilibrium at every temperature the number of points (n) is usually tested at a particular temperature before reducing the temperature. The algorithm is terminated when a sufficiently small temperature is obtained are a small enough change in function value is obtained. The structure of the proposed simulated annealing algorithm (SA) is as follows and is shown in figure 5.2.

**Step-1: Initialization**

Choose a start point (x) and set a high starting temperature (T), number of iterations to be performed at a particular temperature K; (K=1 to n)

**Step-2: Generation of neighborhood seed and evaluation** (Evaluate objective function $E = f(x)$)

**Step-3: Find new point** $X_{i(k+1)} = X_{i(k)} + \lambda_i (X_{i_{\text{max}}} - X_{i_{\text{min}}})$ (5.2)

$\lambda_i \in (-1,1)$

$$\lambda_i = \text{sign}(\bigcup_i -0.5) * T_i \left[ (1 + \frac{1}{T_i})^{|2U_i -1|}-1 \right]$$ (5.3)

$U_i =$ random variable between 0 and 1

Select $\Delta_i$ with probability determined by $g(\Delta_i, T)$. Set the new point $X_{\text{new}} = X + \Delta_x$

$$g(\Delta_i, T) = \Phi \prod T \sim^{n/2} \exp \left[ -\frac{|\Delta_i|^2}{2T} \right]$$ (5.4)

$n =$ dimension of space under exploration. The new point should be between the maximum and minimum limit.
Step-4: Calculate the new value of the objective function using fitness equation.

\[ E_{\text{new}} = f(X_{\text{new}}) \]  

(5.5)

Step-5: Calculation of uphill and downhill move acceptance parameter \( \Delta E \)

Set \( X \) to \( X_{\text{new}} \) and \( E \) to \( E_{\text{new}} \) with probability determined by acceptance function \( h(\Delta E, T) \)

\[
h(\Delta E, T) \approx \frac{1}{1 + \exp \left( \frac{\Delta E}{T} \right)}
\]

(5.6)

\( T = \) Current temperature, \( \Delta E = E_{\text{new}} - E \)

Step-6: Increment the iteration count \( K \), if \( K \) reaches the maximum stop iteration; otherwise go back to STEP-3.

Step-6: Reduce the temperature according to annealing schedule \( T = T_0 \times \alpha \), \( \alpha = \) cooling rate [91], usually between 0 and 1 and when \( T \) is small, terminate; Else go to step 2.

The cooling schedule is an important feature of this algorithm, in the generalized approach, \( \alpha \) may vary with respect to the temperature which is as follows:

\[
\left( \frac{\min T}{T_0} \right)^{1/\text{max no. of iteration}}
\]

(5.7)
Fig. 5.2 Simulated Annealing Structure
In order to optimize the present problem using simulated Annealing algorithms (SAs), the constrained optimization problem is stated as follows:

From the observed data for surface roughness, the response function has been determined using RSM and fitness function, defined as

Minimize,

\[ R_a = -4.89 + 2.49F - 38.0D + 0.599V + 3.27R - 5.38F * D + 0.0140F * V - 18.2F * R + 0.0097D * V + 15.8F * R - 0.232F * V + 80.5F^2 + 16.5D^2 - 0.00318V^2 \]  \( \text{(5.8)} \)

subject to

\[ 39.269 \text{ m/min} \leq V \leq 94.247 \text{ m/min} \]

\[ 0.059 \text{ mm/rev} \leq F \leq 0.26 \text{ mm/rev} \]

\[ 0.4 \text{ mm} \leq D \leq 1.2 \text{ mm} \]

\[ 0.4 \text{ mm} \leq R \leq 1.2 \text{ mm} \]

\[ x_{il} \leq x_i \leq x_{iu} \]

where \( x_{il} \) and \( x_{iu} \) are the upper and lower bounds of process variables \( x_i \). \( x_1, x_2, x_3, x_4 \) are the cutting speed, feed, depth of cut and nose radius respectively. In order to optimize the present problem using IGAs, the following parameters have been selected to obtain optimal solutions with less computational effort.

Initial Temperature \( T_i = 1^{\circ}C \)

Final Temperature \( T_f = 1*10^{-20}^{\circ}C \)

Maximum no. of iterations = 10000
5.2. Simulation Studies and Performance Evaluation

The SA code was developed using MATLAB. The input machining parameter levels were fed to the SA program. Table 5.1 shows the minimum values of surface roughness with respect to input machining parameters SA. It is possible to determine the conditions at which the turning operation has to be carried out in order to get the optimum surface finish. Fig. 5.3 shows the Performance of SAA and Fig. 5.4 shows the Cooling diagram of SAA. Hence, it can be concluded from the optimization results of the SA program that it is possible to select a combination of cutting speed, feed, depth of cut and nose radius to achieve the better surface finish.

Table 5.1 Output values of simulated annealing algorithms with respect to input machining parameters

<table>
<thead>
<tr>
<th>Machining Parameters</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feed, F(mm/rev)</td>
<td>0.12722</td>
</tr>
<tr>
<td>Depth of cut, D(mm)</td>
<td>1.19947</td>
</tr>
<tr>
<td>Cutting Velocity, (m/min)</td>
<td>47.4072</td>
</tr>
<tr>
<td>Nose Radius, R(mm)</td>
<td>0.45422</td>
</tr>
<tr>
<td>Min. Surface Roughness, Ra(microns)</td>
<td>4.94068*10^-7</td>
</tr>
</tbody>
</table>
Fig. 5.3. Performance of SAA

Fig. 5.4. Cooling diagram of SAA
5.3. Summary

In this part, the problem of minimizing the surface roughness in turning operation has been investigated. To model the machining process, several important operational constraints have been considered. These constraints were taken into account in order to make the model more realistic. To optimally determine machining parameters (cutting speed, feed rate, depth of cut and tool nose radius), a simulated annealing method has been employed. The computational results clearly demonstrated that the proposed solution procedure is quite capable in solving such complicated problems effectively and efficiently. A major advantage of SA is its flexibility and robustness as a global search method and good performance will be obtained when the size of problem is small. It can deal with highly nonlinear problems and non-differentiable functions as well as functions with multiple local optima. Even though SA gives good result, for better performance different algorithm named PSO is applied, which is discussed in the next chapter.