OPTIMIZING CNC TURNING PROCESS USING REAL CODED GENETIC ALGORITHM AND DIFFERENTIAL EVOLUTION

Pinkey Chauhan, Kusum Deep, Millie Pant (2011)

Zhou Yidong
2016.02.23
Contents

1. Introduction
2. Machining optimization model
3. methodology
4. Computational Results And Comparisons
5. Conclusions
Introduction

Problem review

- Develop efficient and economic CNC machining
  - Control all process by computer
  - Meet the highly accuracies
  - Introduces flexibility in processing
- Optimizing CNC turning machine parameter
  - Cutting speed
  - Feed rate
  - Depth of cutting
Introduction

- **Problem description**
  - Problem
    - Optimizing CNC turning process parameters
  - Objective
    - Minimum production time
  - Decision variable
    - Feed rate
    - Cutting speed
  - Algorithms
    - An real coded genetic algorithm
    - Differential evolution
  - constraints
    - Decision variables’ lower and upper bound
    - Cutting force
    - Cutting power
    - Tool chip Interface temperature
    - Surface roughness
## Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Significance</th>
<th>Numerical value</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>diameter of the workpiece (mm)</td>
<td>152</td>
</tr>
<tr>
<td>L</td>
<td>length of the workpiece (mm)</td>
<td>203</td>
</tr>
<tr>
<td>V</td>
<td>cutting speed (m/min)</td>
<td>-</td>
</tr>
<tr>
<td>$V_{\text{min}}$</td>
<td>minimum allowable cutting speed</td>
<td>30</td>
</tr>
<tr>
<td>$V_{\text{max}}$</td>
<td>maximum allowable cutting speed</td>
<td>200</td>
</tr>
<tr>
<td>f</td>
<td>feed rate (mm/rev)</td>
<td>-</td>
</tr>
<tr>
<td>$f_{\text{min}}$</td>
<td>minimum allowable feed rate</td>
<td>0.254</td>
</tr>
<tr>
<td>$f_{\text{max}}$</td>
<td>maximum allowable feed rate</td>
<td>0.762</td>
</tr>
<tr>
<td>$R_{\text{s}}$</td>
<td>surface roughness (μm)</td>
<td>-</td>
</tr>
<tr>
<td>$R_{\text{max}}$</td>
<td>max. surface roughness of rough and finished cut</td>
<td>50</td>
</tr>
<tr>
<td>$P_{\text{max}}$</td>
<td>max. power of the machine (kW)</td>
<td>5</td>
</tr>
<tr>
<td>$F_{\text{max}}$</td>
<td>max. cutting force (N)</td>
<td>900</td>
</tr>
<tr>
<td>$\Theta_{\text{max}}$</td>
<td>max. temperature of tool workpiece interface (°C)</td>
<td>550</td>
</tr>
<tr>
<td>doc</td>
<td>depth of cut (mm)</td>
<td>-</td>
</tr>
<tr>
<td>$\text{doc}_{\text{min}}$</td>
<td>minimum allowable depth of cut (finish)</td>
<td>2.0</td>
</tr>
<tr>
<td>$\text{doc}_{\text{max}}$</td>
<td>maximum allowable depth of cut (finish)</td>
<td>5.0</td>
</tr>
<tr>
<td>T</td>
<td>tool life (min)</td>
<td>-</td>
</tr>
<tr>
<td>$t_{\text{m}}$</td>
<td>machining time (min)</td>
<td>-</td>
</tr>
<tr>
<td>$t_{\text{cs}}$</td>
<td>tool change time (min/edge)</td>
<td>0.5</td>
</tr>
<tr>
<td>$t_{\text{l}}$</td>
<td>loading and unloading time (min/pass)</td>
<td>1.5</td>
</tr>
<tr>
<td>$t_{\text{r}}$</td>
<td>quick return time (min/pass)</td>
<td>0.13</td>
</tr>
<tr>
<td>$T_{\text{u}}$</td>
<td>total production time (min)</td>
<td>-</td>
</tr>
<tr>
<td>$C_{\text{o}}$</td>
<td>operating cost (Rs/piece)</td>
<td>3.5</td>
</tr>
<tr>
<td>$C_{\text{t}}$</td>
<td>tool cost per cutting edge (Rs/edge)</td>
<td>17.5</td>
</tr>
<tr>
<td>$C_{\text{T}}$</td>
<td>total production cost (Rs/edge)</td>
<td>-</td>
</tr>
<tr>
<td>a1, a2, a3, K</td>
<td>constants used in tool life equation</td>
<td>0.29 ; 0.35 ; 0.25 ; 193.3</td>
</tr>
</tbody>
</table>
The machining optimization model

- Machining processing is shaping of metal parts by removing unwanted material. Should satisfy given quality specifications as accuracy, surface finish and surface integrity with an objective of minimum production time.

• The objective function:

\[ T_u = t_m + t_{cs} \left( \frac{t_m}{T} \right) + t_r + t_h z \]

where cutting time per pass is

\[ (t_m) = \frac{\pi DL}{1000 Vf} \]

Taylor’s tool life equation is given by

\[ V f^{a_1} d o c^{a_2} T^{a_3} = K \]
The machining optimization model

- Constraints during machining:
  - The bound of parameters depth of cut, feed rate and cutting speed:
    1. \( d_{oc_{\min}} \leq d_{oc} \leq d_{oc_{\max}} \)
    2. \( f_{\min} \leq f \leq f_{\max} \)
    3. \( V_{\min} \leq V \leq V_{\max} \)
  - Constraints imposed on required machining features:
    I. Cutting force (preventing tool chatter or deflection)
    \[ F = 844 \cdot V^{-0.10133} \cdot f^{0.725} \cdot d_{oc}^{0.75} \leq F_{\max} \]
    II. Cutting power
    \[ P = 0.0373 \cdot V^{0.91} \cdot f^{0.78} \cdot d_{oc}^{0.75} \leq P_{\max} \]
    III. Working temperature
    \[ 74.96 \cdot V^{0.4} \cdot f^{0.2} \cdot d_{oc}^{0.105} \leq \theta_{\max} \]
    IV. Surface roughness
    \[ 14.785 \cdot V^{-1.52} \cdot f^{1.004} \cdot d_{oc}^{0.25} \leq R_{\max} \]
LXPM: A real coded GA

- **Chromosomes representation**
  Decision variables are encoded as real numbers
  Individual: $X^1 = (x_1^1, x_2^1, x_3^1, ..., x_n^1)$

- **Laplace crossover**
  Parents: $X^1 = (x_1^1, x_2^1, x_3^1, ..., x_n^1)$ and $X^2 = (x_1^2, x_2^2, x_3^2, ..., x_n^2)$
  Offspring: $Y^1 = (y_1^1, y_2^1, y_3^1, ..., y_n^1)$ and $Y^2 = (y_1^2, y_2^2, y_3^2, ..., y_n^2)$

A random number $\beta_i = \begin{cases} a - b \log(u_i), & r_i \leq 0.5; \\ a + b \log(u_i), & r_i \geq 0.5, \end{cases}$

Where $a$ is a location parameter and $b > 0$ is a scaling parameter, uniform random numbers $u_i, r_i \in [0,1]$

Crossover operation:

$$y_i^1 = x_i^1 + \beta_i |x_i^1 - x_i^2|$$
$$y_i^2 = x_i^2 + \beta_i |x_i^1 - x_i^2|$$
LXPM: A real coded GA

• **Power mutation:**
  Let $\bar{x}$ be a parent solution then $x$ is a mutated solution.

  $$
  x = \begin{cases} 
  \bar{x} - s(\bar{x} - x^l), & t < r; \\
  \bar{x} + s(x^u - \bar{x}), & t \geq r. 
  \end{cases}
  $$

  $r$ are uniform random number $\in [0,1]$, $s$ follow the power distribution $s = (s_1)^p$, where $s_1$ is a random number between 0 and 1, $p$ is the index of mutation. $t = \frac{x - x^l}{x^u - \bar{x}}$ and $x^l, x^u$ being the lower and upper bound of Decision variables.

• **Constraint handling**
  Penalty function:

  $$
  \text{fitness } (X_i) = \begin{cases} 
  f(X_i), & \text{if } X_i \text{ is feasible} \\
  f_{\text{worst}} + \sum_{j=1}^{m} \left| \Phi_j(X_i) \right|, & \text{otherwise}
  \end{cases}
  $$

  $F_{\text{worst}}$ is the objective value of the worst feasible solution currently available in the population. $\phi_j(x_i)$ refer to value of the left hand side of the inequality constraint($\phi_j(x_i) = g_j(x_i)$).

  $$
  g_j(\bar{x}) \geq 0, \quad j = 1, \ldots, J,
  $$
methodology

**LXPM Algorithm:**

**Step 1** (Initialization):
- Initialize population;
- Set Generation=0;

**Step 2** (Evaluation): Evaluate the fitness for each individual

**Step 3** (Termination check): Check the termination criteria, set to Maximum number of generations.
If satisfied stop; else goto 4.

**Step 4** (GA Operations)
- Select individuals according to Tournament selection to build a mating pool
- Apply Laplace Crossover to the population in mating pool with given crossover probability
- Apply Power Mutation to the current population with given mutation probability

**Step 5** (Replacement): Replace the old population with new population while retaining the best individual for next generation

**Step 6**
- Evaluate the best fitness and find optimal individual
- Increment generation; go to step 3.

- **The other parameter setting**
  a. Population = D*10
  b. Generation = 200
  c. Run = 100
  d. Crossover rate(CR) = varies from 0.86 to 0.9
  e. Mutation rate($p_m$) = varies from 0.006 to 0.06
Differential evolution (DE)

**Mutation**
- Produce a trial vector $u_i(t)$ corresponding to each individual of the current population by mutating a target vector $X_i(t)$ with a weighted differential. This trial vector is then used by crossover operator to produce offspring.
  - Select a target vector $X_{i_1}$ from population, random select two individuals $X_{i_2}, X_{i_3}$, such that $i \neq i_1 \neq i_2 \neq i_3$. The trial vector is calculated as:
  $$u_i(t) = X_{i_1}(t) + \beta(X_{i_2}(t) - X_{i_3}(t))$$
  Where $\beta \in (0, i+)$ is the scale factor.

**Crossover**
- Combines the trial vector $u_i(t)$ and the parent vector $X_i(t)$, to produce offspring as:
  $$X_{ji}'(t) = \begin{cases} u_{ji}(t) & \text{if } \text{randb}(j) \leq CR \text{ or } j = \text{rnbr}(i) \\ X_{ji}(t) & \text{if } \text{randb}(j) > CR \text{ or } j \neq \text{rnbr}(i) \end{cases}$$
  Where $\text{randb}(j) \in [0,1]$ is the $j$th evaluation of random number generator.$\text{rnbr}(i)$ is a randomly chosen index $\in [1,2,\ldots,d]$, which ensures that offspring, has at least one component from trial vector $u_i(t)$.

**Selection**
- Selection operator decides which individual should be forwarded to next generation.
  $$X_i(t+1) = \begin{cases} X_{ji}'(t) & \text{if } f(X_{ji}'(t)) \leq f(X_i(t)) \\ X_i(t) & \text{otherwise} \end{cases}$$
methodology

**DE Algorithm**

Step 1. Initialization:
- Set the generation number \( G = 0 \)
- Randomly initialize a population of \( NP \) individuals.

Step 2. DE operations:
WHILE the stopping criterion is not satisfied
Do
  For \( i = 1 \) to \( NP \) //do sequentially for each individual

  **Step 2.1 Mutation Step**
  Generate a donor vector corresponding to the \( i \)-th target vector using mutation schemes

  **Step 2.2 Crossover Step**
  Generate a trial vector for the \( i \)-th target vector through binomial crossover

  **Step 2.3 Selection Step**
  Evaluate the trial vector and target vector to compare the best one to move to next generation

End For

**Step 2.4 Increase the Generation Count**
\[ G = G + 1 \]

END WHILE

---

- **The other parameter setting**
  a. Population = \( D \times 10 \)
  b. Generation = 200
  c. Run = 100
  d. Scaling factor(\( \beta \)) = 0.5
  e. Crossover rate(\( CR \)) = 0.08

---

Production and Logistics Information
Computational Results And Comparisons

Computational experiment

- Experiment environment
  - VC++ on Celeron PC
  - 1.4 GHz, 1.256 GB RAM
- best parameter value obtained using LXPM and DE for different values of depth of cut

<table>
<thead>
<tr>
<th>Depth of cut (d)</th>
<th>LXPM</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$V^*(m/min)$</td>
<td>$f^*(mm/rev)$</td>
</tr>
<tr>
<td>2.0</td>
<td>139.26</td>
<td>0.762</td>
</tr>
<tr>
<td>2.5</td>
<td>129.07</td>
<td>0.762</td>
</tr>
<tr>
<td>3.0</td>
<td>122.72</td>
<td>0.686</td>
</tr>
<tr>
<td>3.5</td>
<td>122.43</td>
<td>0.585</td>
</tr>
<tr>
<td>4.0</td>
<td>134.54</td>
<td>0.517</td>
</tr>
<tr>
<td>4.5</td>
<td>127.92</td>
<td>0.454</td>
</tr>
<tr>
<td>5.0</td>
<td>132.15</td>
<td>0.410</td>
</tr>
</tbody>
</table>
Computational Results And Comparisons

- Computational experiment
  - An Analysis for LXPM and DE for 100 runs

<table>
<thead>
<tr>
<th>Depth of cut (d)</th>
<th>LXPM</th>
<th>DE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean (obj func value)</td>
<td>Std. deviation</td>
<td>Avg fun Eval</td>
</tr>
<tr>
<td>2.0</td>
<td>2.780401</td>
<td>4.135e-05</td>
<td>308</td>
</tr>
<tr>
<td>2.5</td>
<td>2.8733757</td>
<td>1.40e-04</td>
<td>338</td>
</tr>
<tr>
<td>3.0</td>
<td>3.0660640</td>
<td>0.0024726</td>
<td>405</td>
</tr>
<tr>
<td>3.5</td>
<td>3.336070</td>
<td>0.0294902</td>
<td>465</td>
</tr>
<tr>
<td>4.0</td>
<td>3.5686617</td>
<td>0.0108513</td>
<td>426</td>
</tr>
<tr>
<td>4.5</td>
<td>3.8364324</td>
<td>0.017543</td>
<td>474</td>
</tr>
<tr>
<td>5.0</td>
<td>4.0990033</td>
<td>0.012372</td>
<td>414</td>
</tr>
</tbody>
</table>
Computational Results And Comparisons

Computational experiment

- An Analysis for LXPM and DE for 100 runs

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>BSP</th>
<th>NMS</th>
<th>GA</th>
<th>SA</th>
<th>PSO</th>
<th>LXPM</th>
<th>DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S.no</td>
<td>doc</td>
<td>T_u</td>
<td>T_u</td>
<td>% dev.</td>
<td>T_u</td>
<td>% dev.</td>
<td>T_u</td>
</tr>
<tr>
<td>1</td>
<td>2.0</td>
<td>2.84</td>
<td>2.87</td>
<td>-1.06</td>
<td>2.85</td>
<td>-0.35</td>
<td>2.85</td>
</tr>
<tr>
<td>2</td>
<td>2.5</td>
<td>2.93</td>
<td>2.97</td>
<td>-1.37</td>
<td>3.12</td>
<td>-6.48</td>
<td>2.93</td>
</tr>
<tr>
<td>3</td>
<td>3.0</td>
<td>3.11</td>
<td>3.15</td>
<td>-1.29</td>
<td>3.13</td>
<td>-0.64</td>
<td>3.15</td>
</tr>
<tr>
<td>4</td>
<td>3.5</td>
<td>3.34</td>
<td>3.44</td>
<td>-2.99</td>
<td>3.46</td>
<td>-3.59</td>
<td>3.34</td>
</tr>
<tr>
<td>5</td>
<td>4.0</td>
<td>3.59</td>
<td>3.69</td>
<td>-2.79</td>
<td>3.51</td>
<td>+2.23</td>
<td>3.59</td>
</tr>
<tr>
<td>6</td>
<td>4.5</td>
<td>3.84</td>
<td>3.88</td>
<td>-1.04</td>
<td>3.96</td>
<td>-3.13</td>
<td>3.85</td>
</tr>
<tr>
<td>7</td>
<td>5.0</td>
<td>4.10</td>
<td>4.23</td>
<td>-3.17</td>
<td>4.14</td>
<td>-0.98</td>
<td>4.12</td>
</tr>
</tbody>
</table>

Analysis:
LXPM, PSO and DE perform better than other algorithms on the considered model.
for different values of depth of cut, LXPM and DE give significant improvement over Binary GA and other methods such as NMS, BSP, GA and SA.

*BSP (Boundary Search Procedure)
NMS (Nelder-Mead Simplex Method)
Binary GA,
SA (Simulated Annealing)
PSO (Particle Swarm Optimization)
Conclusion

• **Optimizing machining parameters for turning process**
  • Objective function is minimizing processing time
  • Take over the constrained machining environment ensure machining performance and product quality

• **Suggested**
  • A real coded genetic algorithm
  • Differential evolution algorithms

• **Adv & Disadv**
THANK YOU