Proceedings of the ASME 2014 12th Biennial Conference on Engineering Systems Design and Analysis
ESDA2014
June 25-27, 2014, Copenhagen, Denmark

ESDA2014-20520

CUTTING TOOLS OFFSET IN LATHE MACHINES USING A MODERN HEURISTIC TECHNIQUE

Dhafar Al-Ani
Mechanical Engineering Department
McMaster University
Hamilton, ON, Canada

Saeid Habibi
Mechanical Engineering Department
McMaster University
Hamilton, ON, Canada

ABSTRACT

In this paper, an optimal control strategy (i.e., offset settings system) based on the new Multi-objective Particle Swarm Optimization with Differential Evolution (MOPSO-DE) technique is developed and presented. The MOPSO-DE algorithm is used for calculating the optimal positions (i.e., offset settings) for the cutting tools in lathe machines. This optimal control strategy yields interesting results without a need to go through the complex mathematical modeling of the lathe system. The proposed technique is validated considering a real-world industrial system. This strategy is designed to take an action every 20 pieces, and it takes only 2.5 sec to run the code and optimally calculate the new settings. The control strategy is implemented using two high precision linear stepper motors. By implementing the new optimal control strategy, the estimated number of the defective pieces per day can be reduced by 85%.

INTRODUCTION

Particle swarm optimization (PSO) is a population-based metaheuristic optimization algorithm developed in 1995 [1]. Unlike other Evolutionary Algorithms (EA), including the Genetic Algorithms [1], the Genetic Programming [2], the Evolution Strategy [3], the Tabu-Search [4], the Ant Colony [5], the Harmony-Search [6], and the Memetic [7], in PSO, there is no selection and crossover operations where all particles in PSO are kept as individuals of the population through the entire course of the run. Further, PSO is the only algorithm that does not implement the survival of the fittest.

A basic variant of the PSO algorithm works by having the ideas of social behavior and swarm intelligent. Further, PSO has many features over the aforementioned EAs such as a concept for optimizing nonlinear functions, roots in artificial life and evolutionary computation, easy to code, computationally efficient, simpler to implement, flexible, faster to converge, and robust. A detailed comparative study of the performance of PSO and state-of-the-art EAs against a set of well-known and standard benchmark test functions was done in [8,9]. It was concluded that PSO outperformed all other EAs in terms of requiring less function evaluations to reach the find the global solution for a given problem.

To date, PSO has been successfully applied to many real-world applications such as biomedical engineering [10], robotic manipulators [11], control engineering [12], aerodynamic design [13], supply chain management [14], Medical decision making [15], etc. In this work, a new version of Multi-objective PSO algorithm is used as an alternative for a cutting tool offset system because of its powerful performance in several applications and the fact that it is simple to implement and test.

Offsetting two cutting tools in lathe machines is a crucial step for the optimal manufacturing-based processes. This optimization-based task, unique to optimal offset settings, is very difficult because of the complexity and dimensions of the problem as well as the high-computational effort required by any optimization algorithm to be implemented. A typical Single Optimization Problem (SOP) has first been suggested to solve the offsetting problems, but there is none in the literature (i.e., neither in the optimization community nor in the manufacturing associations) that could confirm the success of the optimization-based techniques in tackling this type of problems. Several systems exist nowadays for offsetting two cutting tools in lathe machines: offline programmed schemes and PID control schemes. No matter which of the benefits of these techniques a shop hopes to take advantage of, all of them will need to be programmed. If an operator cannot confidently and productively program the functionality, they may not be able to obtain the benefits. Moreover, these techniques permit the cutting tools to be offset properly but not optimally, where lateral and longitudinal misalignment can lead to significant loss in the manufacturing processes. This has been reported and investigated since the lathe machines are used in the workshops. Even today, accuracy remains the main goal for improvement. Previous studies have investigated the use of optimization methods for the tasks scheduling, but not for the offset of the cutting tools.

In this work, an offset cutting tool system, which can be used for regular lathe machines (NOT Computer Numerical Control (CNC) machines) and based on swarm optimization technique. Two cutting tools in a lathe machine will be tested, where the new developed version of Multi-objective PSO optimization algorithm is used for calculating their optimal position in two degrees of freedom. The offset system is supported by a friendly graphical user interface.

NOMENCLATURE

Symbol	Definition
B	the offset in x and y directions
c_1	the social factor
c_2	the local factor
c_3	the cognitive factor
FEs	fitness evaluations
f_1, f_2	the objective function 1 and 2
g_{best}^{t}	swarm best position
Li	the lower bound of the *n-th* dimensional
l_{best}^{t}	the swarm best position in the neighbor
M	the slope of the regression linear equation
$Maxiter$	the number of maximum iterations
N	the swarm size called also the population size)
Ns	the size of each sub-swarm
`n	decision variables
p_{best}^{t}	particle best position
r	a random number in the range of [0,1]
$rand$	random value [0,1]
std	standard deviation of an array
Ui	the upper bound of the *n-th* dimensional
v_t	the velocity of the particle (i) at current iteration (t)
$\vec{v}_{t+1}$	the velocity of the particle (i) at iteration (t+1)
w	inertia weight factor
X	the variable of the linear regression equation
$\vec{x}_{t+1}$	the position of the particle (i) at iteration (t+1)
x_i	the position of the *i-th* particle at iteration t
$\vec{x}_t$	the position of the particle *(i)* at current iteration (t)

MOPSO-DE ALGORITHM

The main advantage of the Particle Swarm Optimization (PSO) is its ability to successfully explore and then solve optimization problems that have both global and local domains.

In the local version of PSO, each particle's velocity is updated based on its personal best (p_{best}). While in the global version PSO, each particle updates its velocity according to its personal best (p_{best}) and the best performance achieved for the entire swarm (g_{best}).

Hence, the new dynamic model, proposed by the authors in [16], is incorporated both local and global techniques as well as introducing a new best performance achieved within its neighborhood, called (l_{best}) to successfully balance the exploration and exploitation ability of Multi-objective PSO with Differential Evolution (MOPSO-DE) is proposed. In addition, to further improve the performance of MOPSO-DE, the parallel MOPSO scheme, proposed by the authors in [17], is employed to enhance not only the computational complexity of the algorithm, but also its global performance in finding the optimal Pareto-front. Further, the Differential Evolution (DE) is adopted to update the particle's best (p_{best}) at each iteration.

The details of the proposed MOPSO-DE are provided in the following.

1. A NEW DYNAMIC MODEL

In this model, a new type of leader, referred to as the local best (l_{best}), is introduced to focus the search around small regions in the vicinity of the best fronts unlike other MOPSO algorithms in which particles could potentially result in a chaotic search behavior. Accordingly, the global leader (g_{best}) used in the original PSO is incorporated with the local best (l_{best}) to lead the swarm toward Pareto-front.

In the MOPSO-DE algorithm, particles "remember" (i.e. keep record of) their own best positions (p_{best}), the local best position (l_{best}), and the swarm best position (g_{best}). Hence, particles flight directions are determined and then guided by the particles' self-experience, the best position found so far by the neighbor, and the best position in the swarm found so far during the search process. The new dynamic model introduces a new leader between the two positions (p_{best} and g_{best}) to bring them closer in the search-space. This adds less pressure to the particles to slowly move towards the swarm best resulting in avoiding premature convergence.

Based on the new model, each particle adjusts its position based on the following:

- The current particle position,
- The current particle velocity,
- The distance between the current particle position and its best position (p_{best}), and
- The distance between the current particle position and the closest best position in the neighbor (l_{best}).
- The distance between the current particle position and the closest best position in the swarm (g_{best}).

In this work, a new dynamic model is given in Eqs. (1) and (2) by incorporating the combination of three guides p_{best}, l_{best} and g_{best} in updating the particle velocity and position in order to steer its flight direction towards the global optimum regions.

$$\begin{aligned}\vec{v}_{t+1} = w \times \vec{v}_t + c_1 \times rand(\quad) \times \left(\vec{g}_{best,i} - \vec{x}_t\right) \\ +c_2 \times rand(\quad) \times \left(\vec{l}_{best,i} - \vec{x}_t\right) \\ +c_3 \times rand(\quad) \times \left(\vec{p}_{best,i} - \vec{x}_t\right)\big]\end{aligned} \tag{1}$$

$$\vec{x}_{t+1} = \vec{x}_t + \vec{v}_{t+1} \tag{2}$$

2. PARALLEL ISLANDS MODEL

In MOPSO algorithms, parallel islands models have been found to be better in search performance than single swarm models in terms of the quality of the solutions and the reduced computational time [18]. In this work, the parallel islands model is implemented using the Parallel Computing toolbox and MATLAB Distributed Computing Server.

In the proposed MOPSO-DE, the swarm is divided into several sub-swarms. Each sub-swarm is assigned to a different processor (*island*). Semi-isolated sub-swarms help maintain MOPSO diversity (as shown in Figure 1). Therefore, the swarm of each island can explore a different part of the search-space. Each processor runs a sequential MOPSO on its swarm. Parallel islands models allow migration (i.e. periodic exchange) of good candidate solutions from one island to another after every fitness evaluation.

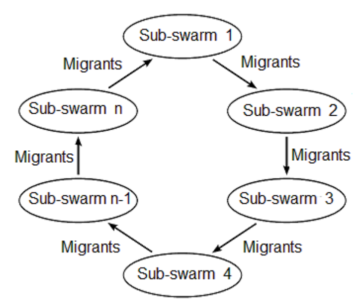

FIGURE 1: THE ISLANDS MODEL OF *n* SEMI-ISOLATED SUB-SWARMS

3. DIFFERENTIAL EVOLUTION

In order to improve the global search capability of the proposed MOPSO, DE is incorporated to update the personal best (p_{best}) as follows:

The vector of best particles $\left(\overrightarrow{p_{best}}_i^t\right)$ is updated at every iteration. As a result, a corresponding trail vector $(\vec{u}_i^t)$ is yielded. Then, $\overrightarrow{p_{best}}_i^t$ is updated only if the trail vector $(\vec{u}_i^t)$ is better (i.e., wins). The process of updating the personal best by DE is depicted in Table 2. It should be noted that Munoz-Zavala et al. [19] also exploited a similar approach by perturbing the personal best (p_{best}) of each particle; however, only the mutation operator in DE is used.

As mentioned in the previous section (Adaptive Mutation Technique), inducing randomness to the particles (i.e., by altering the memory of the personal best in the search-space) will enhance the overall the performance of the swarm to locally exploit the global region. In addition, incorporating DE to the MOPSO can also results in better exploration, where which each sub-swarm is encouraged to search for more promising region which might not be reached by the other sub-swarms during the evolution.

As illustrated in [20], if the variable value $u_{i,j}^t$ of the trial vector $(\vec{u}_i^t)$ violates the boundary constraint, the violated variable value is set to the boundary value using the following rule:

$$u_{i,j}^t = \begin{cases} L_j & if \quad (p \le 0.5) \wedge \left(u_{i,j}^t < L_j\right) \\ U_j & if \quad (p \le 0.5) \wedge \left(u_{i,j}^t > U_j\right) \\ 2L_j - u_{i,j}^t & if \quad (p > 0.5) \wedge \left(u_{i,j}^t < L_j\right) \\ 2U_j - u_{i,j}^t & if \quad (p > 0.5) \wedge \left(u_{i,j}^t > L_j\right) \end{cases} \qquad (3)$$

The simplicity of DE is best explained via the pseudo-code given in Table 1.

TABLE 1: PSEUDO-CODE OF UPDATING P_{best} BY DE

$p_{best} = \{\overrightarrow{p_{best\,1}}, \ldots, \overrightarrow{p_{best\,N}}\}$ (at iteration *t*)
for i (1 to *N*)
DO
Randomly select three different numbers $r_1,\ r_2$, and r_3 from $N \backslash i$
$v_i = \overrightarrow{p_{best\,r1}} + F\,(\overrightarrow{p_{best\,r2}} - \overrightarrow{p_{best\,r3}})$
for j (1 to *N*)
$u_{i,\,j} = \begin{cases} v_{i,j} & if\ randj \le Cr \quad or \quad j-jrand, \\ x_{i,j} & Otherwise, \end{cases}$
end for j
Compare u_i with $\overrightarrow{p_{best\,i}}$ according to the ε*-based rule
if ($\overrightarrow{u_i}$ wins)
Update $\overrightarrow{p_{best\,i}}$
end if
end fori
Obtain p_{best}

* *ε-based is the constraint-handling technique used in the paper.*

AUTOMATIC OFFSET SYSTEM

No discussion about CNC high technology and sophisticated multitasking programming to move the tools inside the machine in three-controlled directions at once on its axes in a prices, safe, and reliable manner without the need of post processing. However, this is not the case in the general (small) workshops and plants whereas the regular lathe machines are still parts of everyday manufacturing processes and will continue in use in the coming future. Further, skilled operators are required to perform manual offset adjustments to the tools into the machines.

The productivity of offline programming and/or manual adjustment is seemed to be inaccurate and unreliable for the regular types of lathe machines. Therefore, it is considered a very challenging task for many companies to work closely with a number of machine tool builders and their customers to develop both robust and accurate offset system to support the comprehensive and accurate programming and simulation of this new breed of highly capable automatic offset systems.

In this paper, the general layout of the proposed automatic offset system is illustrated in Figure 2.

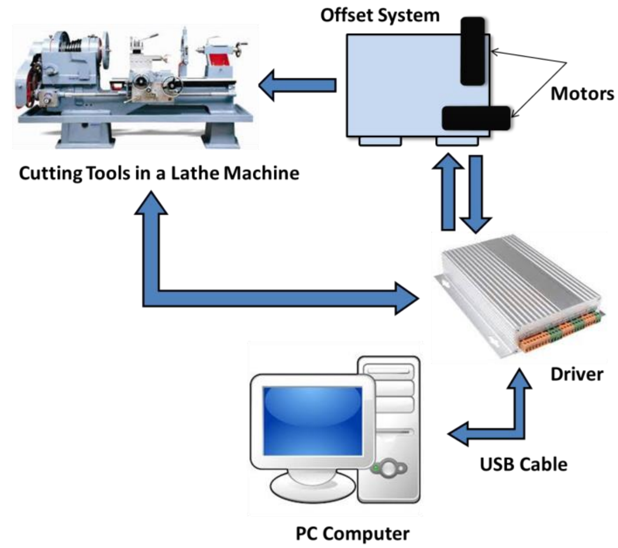

FIGURE 2: OFFSET SYSTEM LAYOUT

The positioning system is comprised of two high precision linear stepper motors as two motorized degrees of freedom x and y are considered. High quality stepper motors can yield high resolutions, allowing a fine-tuning in the order of the micrometer of the position of the cutting tools. The motors are mounted perpendicular to each other. The offset system is directly connected to the lathe machine, which is also connected to the drive.

CASE STUDY

The case study chosen in this research is a workshop in Hamilton, ON, Canada; which has a line product consists of several lathe machines to produce thousands of thin steel rings used in the automotive industry. The problem identified in this workshop is that there are plenty (i.e., huge number) of daily defective pieces due to inaccurate offsetting of the cutting tools. This is done by measuring the diameter as well as the right and left thickness using three dial-gages. The tolerance specified by the customer is set to be as follows: thickness = 0.1575in +/- 0.0012in and diameter = 4.752in +/- 0.006in.

1. PHASE 1: PROBLEM FORMULATION

The tool geometry offset can be defined as an offset that adjusts machine components to compensate for the unique shape of a particular cutting tool. Each cutting tool has two offsets, which serve to locate the tool tip with respect to some standard position The objective function (i.e., function of x and y position of the cutting tools) is identified as a multi-objective optimization problem (MOOP). Solving this problem by using the MOPSO-DE will allow obtaining optimal coordinates, and then, the proposed automatic offset system can be implemented directly to the lathe machine. A sweep of the entire two-dimensional slop functions of x and y coordinates is done. The step size between each position is set to 5 µm.

Sample of the data provided is shown in Figure 3. The cost function is formulated as a linear regression for every 10 to 20 data points. This will give us a linear line curve, which is represented by

$$Y = B + MX \tag{4}$$

For the thickness: there are two measurements H1 & H2. Therefore, the first objective function that minimizes the slope for the thickness is expressed as shown in Eqs. (5) and (6).

$$f_1 = B + MX \tag{5}$$

where

$$B\,(offset) = |H1 - H2| \tag{6}$$

For the diameter: there is only one measurement available. Hence, the cost function that minimizes the slop of the linear regression equation is described as shown in Eqs. (7) and (8).

$$f_2 = B + MX \tag{7}$$

where

$$B\,(offset) = 0 \tag{8}$$

Then, the objective function of this problem can be defined as shown in Eq. (9).

$$\text{Minimize } f = [f_1, f_2] \tag{9}$$

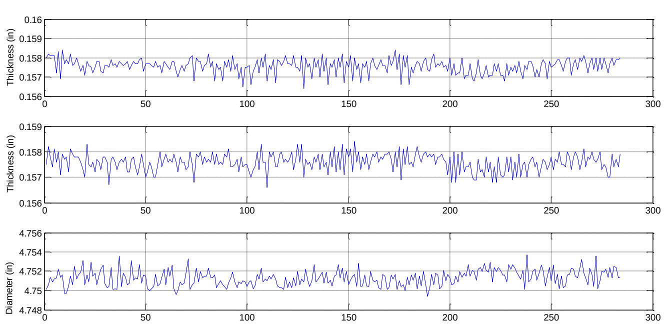

FIGURE 3: SAMPLE OF DATA SHOWING THE RIGHT/LEFT THICKNESS AND DIAMETER FOR 280 PIECES

2. PHASE 2: TESTING

The goal of the testing phase is to determine the performance, the feasibility, and the reliability of implementing the proposed automatic offset system to the lathe machines. Two main criteria are considered in this phase. The first criterion is the number of iterations (also known as the number of function evaluations) required by the system to find the optimal settings. This number can then be used as an indicator to trace the progress of the search process as well as acts as one of the stopping criteria to stop the execution of the optimization algorithm. The second criterion is the error between the expected coordinates and the coordinates found by the MOPSO-DE algorithm using different population sizes. This error is used as an indicator to evaluate the best performance.

After conducting several tests, the best value for the number of iterations is found to be 20. This value seems to yield very good results for this application. Moreover, the best performance (i.e., smallest error) is investigated by using several population sizes. For each configuration, the root mean square error is recorded, and then, the best population size is found to be 10. Table 2 presents the configurations considered as well as the calculated error target coordinates of x and y.

TABLE 2: COORDINATES ERROR FOR DIFFERENT POPULATION SIZES

Population Size	X-Error (%)	Y-Error (%)
8	0.021	0.020
10	0.015	0.015
12	0.022	0.021
14	0.022	0.023

EXPERIMENTAL RESULTS

The number of iterations and the population size are the two factors that evaluate the time it takes to find a new set of coordinates using the proposed automatic offset system. As a factor of the physical setup, on average, 2.5 seconds is required to run the MOPSO-DE algorithm and finds the optimal offset settings over the 20 iterations. This is due to the mathematical complexity of the cost function, the computational complexity of the MOPSO-DE algorithm, and finally, the mechanical setup of the offset system. In addition, it is important to note here that the algorithm did function correctly when enough write and read time is given to the communication routine.

CONCLUSION

To be effective in global/local markets, demands for higher accuracy and productivity require manufacturers to implement fast, reliable, and robust offset systems to their regular lathe machines. In this paper, a new automatic offset system is developed and presented. A search-based algorithm, referred to as the MOPSO-DE, is used to find an optimal set of coordinates for the cutting tools in lathe machines. This offset system is designed to take an action every 20 pieces, and it takes only 2.5 sec to run the code and then find the new settings. The experimental results presented show that the best population size to use is the one that returns the lowest error in relation to the expected coordinates (i.e., in this work, a swarm of 10 particles is used). Further, the number of iterations has significant effect on the computational time that the algorithm could take to find the best solution at all (i.e., smaller number is faster, and vice versa). Thus, 20 iterations as the maximum iteration (i.e., as a stopping criterion) frequently returned consistent results. The error percentages are small and the average time is acceptable. The advantages of the proposed optimization-based offset versus the exist control systems are accuracy, reliability, simplicity, and increased materials utilization. The estimated number of defects pieces per day can be reduced by 85%.

REFERENCES

[1] R. Eberhart and Y. Shi, "Comparison Between Genetic Algorithms and Particle Swarm Optimization," *Computer Science*, p. 616–661, 1998.

[2] D. B. Fogel, *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*. New York, U.S.A.: IEEE Press, 2000.

[3] I. Rechenberg, "Evolutions Strategie – Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution," PhD thesis, 1971.

[4] F. Glover, "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers and Operations Research*, vol. 13, no. 5, p. 533–549, 1986.

[5] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies," in *the Proceedings of the Europain Conference of the Artifical Life*, Paris, France, 1991, pp. 134-142.

[6] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*. Springer, 2009.

[7] R. Dawkins, *Universal Darwinism: Evolution from Molecules to Man*. Cambridge, United kingdom: Cambridge University Press, 1983.

[8] D. Al-Ani and S. Habibi, "A New Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization Algorithm - Part I," *Artificial Intelligence Research*, 2013.

[9] D. Al-Ani and S. Habibi, "A New Adaptive Parallel Clustering-based Multi-objective Particle Swarm Optimization Algorithm - Part II," *Artificial Intelligence Researc*, 2013.

[10] M. K. Ahirwal, A. Kumar, and G. K. Singh, "Analysis and Testing of PSO Variants Through Application in EEG/ERP Adaptive Filtering Approach," *Biomedical Engineering Letters* , vol. 2, no. 3, pp. 186-197, Sep. 2012 .

[11] G. Alici, R. Jagielskib, Y. Ahmet, and B. Shirinzade, "Prediction of Geometric Errors of Robot Mmanipulators with Particle Swarm Optimisation Method," *Robotics and Autonomous Systems*, vol. 54, pp. 956-966, Aug. 2006.

[12] J. Hamidi, "Control System Design Using Particle Swarm Optimization (PSO)," *Soft Computing and Engineering*, vol. 1, no. 6, pp. 2231-2307, Jan. 2012.

[13] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A Regularity Model-based Multi-objective Estimation of Distribution Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 41-63, 2008.

[14] J. T. Dickersbach, *Supply Chain Management with APO*. Springer, 2005.

[15] L. Shao and M. Ehrgott, "Finding Representative Non-dominated Points in Multi-objective Linear Programming," in *the Proceedings of IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making*, 2007, p. 245–252.

[16] D. Al-Ani and S. Habibi, "A New Energy Optimization Strategy for Pumping Operation in Water Distribution Systems," in *the Proceedings of the ASME Congress Exposition 2012 International Mechanical Engineering*, Huston, Texas-USA.

[17] D. Al-Ani and S. Habibi, "A New Hybrid Particle Swarm Optimization for Solving Pump Scheduling Problems," in *the Proceedings of the AWWA Annual Conference and Exposition (ACE12)*, Dallas, Texas-USA, 2012.

[18] L. D. Whitley and T. Starkweather, "GENITOR II: A Distributed Genetic Algorithm," *Experimental and Theoretical Artificial Intelligence*, 1990.

[19] A. E. Munoz-Zavala, A. Hernandez-Aguirre, E. R. Villa-Diharce, and S. Botello-Rionda, "PESO+ for Constrained Optimization," in *the Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2006)*, 2006.

[20] J. Brest, V. Zumer, and M. S. Maucec, "Self-adaptive Differential Evolution Algorithm in Constrained Real-parameter Optimization," in *the Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2006)*, 2006.