Journal of Production Engineering
Vol.18 No.2

JPE (2015) Vol.18 (2) Original Scientific Paper

Petrović, M., Petronijević, J., Mitić, M., Vuković, N., Plemić, A., Miljković, Z., Babić, B.

THE ANT LION OPTIMIZATION ALGORITHM FOR FLEXIBLE PROCESS PLANNING

Received: 27 September 2015 / Accepted: 20 November 2015

Abstract: *Obtaining an optimal process plan according to all alternative manufacturing resources has become very important task in flexible process planning problem research. In this paper, we use a novel nature-inspired algorithm called Ant Lion Optimizer (ALO) to solve this NP-hard combinatorial optimization problem. The network representation is adopted to describe flexibilities in process planning and mathematical model for the minimization of the total production time and cost is presented. The algorithm is implemented in Matlab environment and run on the 3.10 GHz processor with 2 GBs of RAM memory. The presented experimental results show that the proposed algorithm performs better in comparison with other bio-inspired optimization algorithms.*

Key words: *flexible process planning, ant lion optimizer, particle swarm optimization, genetic algorithms, simulated annealing*

Algoritam zasnovan na koloniji mrava za optimizaciju projektovanja fleksibilnih tehnoloških procesa. *Dobijanje optimalnog sadržaja tehnološkog procesa na osnovu dostupnih proizvodnih resursa postaje veoma važan zadatak u istraživanju problema koji se odnose na projektovanje fleksibilnih tehnoloških procesa. U ovom radu, koristi se novi prirodno orijentisan algoritam (Ant Lion Optimizier-ALO) kako bi se rešili NP-teški nedeterministički poloinomni kombinatorni optimizacioni problemi. Mrežno prikazivanje je usvojeno u cilju opisivanja fleksibilnosti kod projektovanja tehnoloških procesa, dok je matematički model usvojen u cilju postizanja minimalnih ukupnih troškova i vremena. Algoritam je implementiran u okviru programskog okruženja Matlab i pokreće ga procesor 3.1 GHz sa 2 GB RAM memorije. Eksperimentalni rezultati ukazuju na činjenicu da predloženi algoritam bolje rešava zadatke u odnosu na druge biološke orijentisane optimizacione algoritme.*

Ključne reči: *Fleksibilno projektovanje tehnološkog procesa, optimizacija kolonije mrava, optimizacija roja čestica, genetski algoritmi, simulirano kaljenje.*

1. INTRODUCTION

Computer Aided Process Planning (CAPP) represents one of the most important research directions in Computer Integrated Manufacturing (CIM). It was developed at the end of the 20th century for the purpose of integrating computer aided design (CAD) and computer aided manufacturing (CAM). Its aim consists of determining detailed methods for manufacturing a part economically and concurrently starting from the initial phase (drawing of the finished part) up to the final phase (the desired shape of the finished part).

As opposed to traditional manufacturing systems, most jobs produced in today's manufacturing systems may have a large number of alternative process plans due to the variety of alternative manufacturing resources. This problem is NP hard (*non deterministic polynomial optimization problems*) which means that time exponentially increases with increase of alternatives. Conventional nonheuristic methods are not able to find optimal solution for this combinatorial problem. In recent years metaheuristic algorithms have been used as primary techniques for obtaining the optimal solutions of process planning problem. Some of the most popular algorithms in this field are: genetic algorithms (GA), genetic programming (GP), simulated annealing (SA), tabu search (TS), swarm intelligence (ant colony optimization (ACO), particle swarm optimization (PSO)) or hybrid algorithms.

Li et al. [1] proposed GP-based approach to optimize flexible process planning with minimum total processing time as criteria. Network representation was adopted to describe flexibility of process plans and efficient genetic representations and operator schemes were also considered. Using the same optimization objective and representation, Shao et al. [2] presented a modified GA-based approach for generating optimal and near optimal process plans. Lv and Qiao [3] proposed new approach called cross-entropy (CE) to optimize flexible process planning. They used AND/OR network to represented flexibility of process planning and established mathematical model for minimization of total processing time and total cost. Hybrid GA-SA algorithm used to solve flexible process planning problem with the objective of minimizing the production time was presented in [4].

Although these algorithms are able to solve optimization problems, the so-called No Free Lunch theorem [5] allows researchers to propose new algorithms. This paper proposes a new algorithm called Ant Lion Optimizer (ALO) as an alternative approach for solving flexible process planning problem.

The structure of this paper consists of the following sections. In the Section 2 we briefly introduce a flexible process planning problem and describe its representation. Mathematical model of the problem with two objective functions is formulated in Section 3.

Section 4 outlines ant lion optimization concept. Section 5 shows comparative results and Section 6 gives concluding remarks. Finally, acknowledgements and references are stated in Section 7 and Section 8, respectively.

2. FLEXIBILITY AND REPRESENTATION

2.1 The flexible process planning problem

In this paper, we consider the following types of flexibility for process planning optimization: machine flexibility, process flexibility, and sequencing flexibility. Machine flexibility relates to the possibility of performing one operation on different alternative machines, with various processing times and costs. Process flexibility refers to the possibility of producing the same part in different ways with alternative operations or sequences of operations. Sequencing flexibility implies the possibility to interchange the ordering of required manufacturing operations.

2.2 Representation of flexible process plans

The flexible process plans representation consists of the information about alternative machines, processing times, operation sequences, and all the operations needed to manufacture the part. Networks, Petri nets, OR network [1, 2], AND/OR network graphs [3] are some of numerous representation methods used to describe aforementioned types of flexibilities. AND/OR network methodology, described in details in literature [1, 2, 4], is employed to represent flexible process plans.

3. MATHEMATICAL MODELING OF OPTIMIZATION PROBLEM

In this paper, minimization of the total production time and total production cost are two objectives of the optimization problem to be considered. Production time is the one of commonly used minimization criterion in process planning optimization (see e.g. [1, 2]). In this research production time comprises processing time and transportation time.

Processing time (*TW*) is the time needed to machine a material under some operation(s) and can be computed as:

$$TW = \sum_{i=1}^{n} TWI(i,j) \tag{1}$$

where *n* is the (total) number of operations in process plan, *TWI(i,j)* the processing time of operation *i* on the alternative machine *j*.

Transportation time (*TT*) is the time spent for transport of raw materials, goods, or parts between machines and can be computed as:

$$TT = \sum_{i=1}^{n-1} TTI((i,j_1),(i+1,j_2)) \tag{2}$$

where $TTI(i,j_1),(i+1,j_2)$ is the transportation time between the alternative machine j_1 and j_2 for two consecutive operations.

The total production time (*TPT*) is defined as:

$$TPT = TW + TT \tag{3}$$

The production cost is another criterion commonly used to select flexible process plans and to quantitatively measure its quality. The total production cost contains the machine cost and the machine change cost, where each cost factor can be described and computed as follows.

Machine cost (*MC*) is the total cost of the machines selected in flexible process plan and it is computed as:

$$MC = \sum_{i=1}^{n} MCI_i \tag{4}$$

where n is the total number of operations and MCI_i is the predetermined machine cost index for using machine *i,* which is a constant for a particular machine.

Machine change cost (*MCC*) is required to be considered when two consecutive operations are performed on different machines, and it can be computed as:

$$MCC = MCCI \times \sum_{i=1}^{n} \Omega(M_{i+1} - M_i) \tag{5}$$

where *MCCI* is the machine change cost index and M_i is the identity of the machine used for operation *i*.

All this costs are included in the total cost (*PC*) according to the following equation:

$$PC = MC + MCC \tag{6}$$

4. THE ANT LION OPTIMIZER

Antlions belong to group of insects in the family Myrmelentidae. The two main phases of the antlions lifecycle are larval stage and adult stage. The antlion larva is often called "doodlebug" because of the trails it leaves in the sand while looking for a good location to build its trap, see Fig. 1. During the process of hunting, antlion makes funnel pits in soft sand and then waits patiently at the bottom of the pit, Fig. 2. Slipping to the bottom, the prey is immediately seized by the antlion. Or, if prey attempts to escape from the trap, antlion throw sands towards the edge of the pit to slide the pray into the bottom of the pit. By throwing up loose sand from the bottom of the pit, the larva also undermines the sides of the pit, causing them to collapse and bring the prey with them. Mathematical modeling of the behavior of antlions and ants is given in the following section [6].

Fig. 1. Pits made by the antlion in soft sand [7]

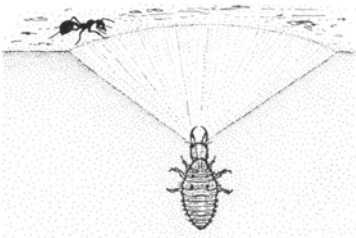

Fig. 2. Hunting behaviour of antlions [8]

4.1 Operators of the ALO algorithm

Random walks of ants when searching food in nature can be described as follows:

$$X(t) = [0, cumsum(2r(t_1)-1), cumsum(2r(t_2)-1), ..., \\ cumsum(2r(t_n)-1)] \quad (7)$$

where *cumsum* calculates the cumulative sum, *n* is the maximum number of iterations, *t* shows the step of random walk, and *r(t)* is stochastic function defined according to the following equation:

$$r(t) = \begin{cases} 1 \; if \; rand > 0.5 \\ 0 \; if \; rand \le 0.5 \end{cases} \quad (8)$$

where *t* shows the step of random walk and rand is a random number generated according to uniform distribution in the range [0,1].

The position of ants is presented with the matrix:

$$M_{Ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,d} \end{bmatrix} \quad (9)$$

where M_{Ant} is the matrix for each ant position, $A_{i,j}$ present the value of *j-th* variable of *i-th* ant, *n* is the number of ants, and *d* is the number of variables.

Fitness function of each ant is saved in the following matrix M_{OA}:

$$M_{OA} = \begin{bmatrix} f([A_{1,1}, A_{1,2}, ..., A_{1,d}]) \\ f([A_{2,1}, A_{2,2}, ..., A_{2,d}]) \\ \vdots \\ f([A_{n,1}, A_{n,2}, ..., A_{n,d}]) \end{bmatrix} \quad (10)$$

where *f* is objective function (see eq. (3) and (6)).

$$M_{Antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \cdots & AL_{1,n} \\ AL_{2,1} & AL_{2,2} & \cdots & AL_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ AL_{n,1} & AL_{n,2} & \cdots & AL_{n,d} \end{bmatrix} \quad (11)$$

where $M_{Antlion}$ is the matrix for each antlion position, $AL_{i,j}$ present the value of *j-th* variable of *i-th* antlion, *n* is the number of antlions, and *d* is the number of variables.

Analogously, fitness function of each antlion is saved in the following matrix M_{OAL}:

$$M_{OAL} = \begin{bmatrix} f([AL_{1,1}, AL_{1,2}, ..., AL_{1,d}]) \\ f([AL_{2,1}, AL_{2,2}, ..., AL_{2,d}]) \\ \vdots \\ f([AL_{n,1}, AL_{n,2}, ..., AL_{n,d}]) \end{bmatrix} \quad (12)$$

In order to keep random walks of ants inside the search space, they are normalized using the following equation:

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i - c_i^t)}{(d_i^t - a_i)} + c_i \quad (13)$$

where a_i is the minimum of random walk of *i-th* variable, b_i is the maximum of random walk in *i-th* variable, c_i^t is the minimum of *i-th* variable at *t-th* iteration, and d_i^t is the maximum of *i-th* variable at *t-th* iteration.

Mathematical modelling of ants trapping in antlion's pits is given by the following equations:

$$c_i^t = Antlion_j^t + c^t \quad (14)$$

$$d_i^t = Antlion_j^t + d^t \quad (15)$$

where c^t is the minimum of all variables at *t-th* iteration, d^t is the maximum of all variables at *t-th* iteration, and $Antlion_j^t$ is the position of the selected *j-th* antlion at *t-th* iteration.

Antlion's hunting capability is modelled by fitness proportional roulette wheel selection. The mathematical model that describes the way how the trapped ant slides down towards antlion is given as follows:

$$c^t = \frac{c^t}{I} \quad (16)$$

$$d^t = \frac{d^t}{I} \quad (17)$$

where *I* is a ratio calculated as:

$$I = 10^w \cdot \frac{t}{T} \quad (18)$$

where *t* is the current iteration, T is the maximum number of iterations, *w* is the constant that depends on current iteration as follows:

$$w = \begin{cases} 2 \; if \; t > 0.1T \\ 3 \; if \; t > 0.5T \\ 4 \; if \; t > 0.75T \\ 5 \; if \; t > 0.9T \\ 6 \; if \; t > 0.95T \end{cases} \quad (19)$$

Finally, elitism is applied in the following way: the best antlion in each iteration is considered to be elite. It means that every ant randomly walks around selected antlion and has position according to:

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (20)$$

where R_A^t is the random walk around the antlion selected by the roulette wheel at *t-th* iteration, and R_E^t is the random walk around the elite antlion at *t-th* iteration.

5. EXPERIMENTAL RESULTS

In order to evaluate the performance and illustrate the effectiveness of the ALO approach, the algorithm procedure is coded in Matlab software and implemented on a personal computer with a 3.10 GHz processor. In this experiment, job 1 (see Fig. 3) is considered and the transportation time between the machines is given in Table 1. The parameters of algorithm are set as follows: the size of the population is 40 and the number of iterations is 30. The optimization objective is to get an optimal operation sequence that results in minimum production time (eq. (3)) and cost (eq. (6)). Fig. 4a illustrates the convergence curves for the GA, hybrid GA-SA, PSO, and ALO algorithm and Fig. 4b shows error bar plot of production time after 10 runs. The obtained optimal sequence according to minimal production time is (1,2)-(5,3)-(8,7)-(9,8), *TPT=117*, and according to minimal production cost is (1,2)-(5,3)-(8,7)-(9,8), *TPC = 545.*

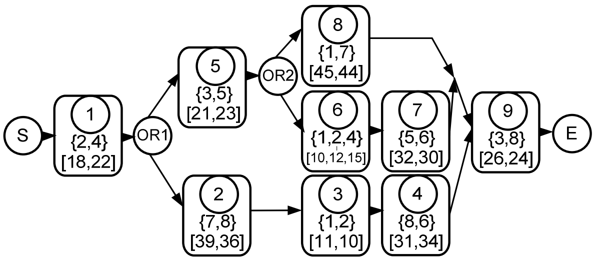

Fig. 3. Alternative process plans network [3]

machine	1	2	3	4	5	6	7	8
1	0	3	7	10	3	5	8	12
2	3	0	4	7	5	3	5	8
3	7	4	0	3	8	5	3	5
4	10	7	3	0	10	8	5	3
5	3	5	8	10	0	3	7	10
6	5	3	5	8	3	0	4	7
7	8	5	3	5	7	4	0	3
8	12	8	5	3	10	7	3	0

Table 1. Transportation time between machines [3]

machine	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
cost	30	10	30	40	100	60	10	15
MCCI	160							

Table 2. Machine cost and machine change cost

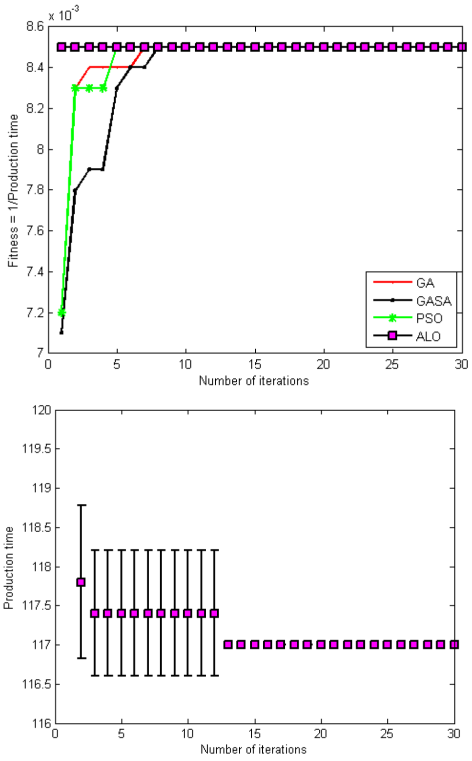

Fig. 4. (a) Comparison of the GA, hybrid GA-SA, PSO and ALO algorithm; (b) Error bar plot of production time

6. CONCLUSION

In this paper, new approach based on Ant Lion Optimization (ALO) algorithm is proposed to optimize combinatorial NP-hard flexible process planning problem. The network representation method is adopted to describe process flexibility, sequencing flexibility, and machine flexibility. The main steps of ALO algorithms are implemented on process planning problem. The performance of the presented ALO algorithm are verified and evaluated in comparison with the results obtained with GA, SA, and PSO standalone algorithms as well as hybrid GA-SA algorithm. Experimental results indicate that the proposed algorithm performs better in comparison with other bio-inspired optimization algorithms.

7. REFERENCES

[1] Li, X.Y., Shao, X.Y., Gao, L.: *Optimization of flexible process planning by genetic programming*, International Journal of Advanced Manufacturing Technology 38(1-2), pp. 143–153, 2008.

[2] Shao, X., Li, X., Gao, L., Zhang, C.: *Integration of process planning and scheduling—a modified genetic algorithm-based approach*, Computers and Operations Research 36(6), pp. 2082–2096, 2009.

[3] Lv, S., Qiao, L.: *A cross-entropy-based approach for the optimization of flexible process planning*, International Journal of Advanced Manufacturing Technology 68(9-12), pp. 2099–2110, 2013.

[4] Petrović, M., Miljković, Z., Babić, B.: *Optimization of Operation Sequencing in CAPP Using Hybrid Genetic Algorithm and Simulated Annealing Approach*, In: Proceedings of the 11th International Scientific Conference MMA 2012 – Advanced Production Technologies, Novi Sad, Serbia, pp. 285–288, 2012.

[5] Wolpert, D.H., Macready, W.G.: *No free lunch theorems for optimization*, Evolutionary Computation, IEEE Transactions, 1(1), pp. 67-82, 1997.

[6] Mirjalili, S.: *The ant lion optimizer*, Advances in Engineering Software, 83, pp. 80-98, 2015.

[7] http://www.asknature.org/media/image/3305

[8] http://oldblockwriter.blogspot.com/2012/02/ant-lions-and-angle-of-rest.html

ACKNOWLEDGMENT

This paper is part of the project: *An innovative, ecologically based approach to the implementation of intelligent manufacturing systems for the production of sheet metal parts*, financed by the Ministry of Education, Science and Technological Development of the Serbian Government, Grant *TR-35004* (2011-2015).

Authors: **M.Sc. Milica Petrović, M.Sc. Jelena Petronijević, Dr. Marko Mitić, Dr. Najdan Vuković, Aleksandar Plemić, prof. dr. Zoran Miljković, prof. dr. Bojan Babić**, University of Belgrade, Faculty of Mechanical Engineering, Production Engineering Department, Kraljice Marije 16, 11120 Belgrade 35, Serbia, Phone: +381 11 3302-264, Fax: +381 11 3302-274.

E-mail: mmpetrovic@mas.bg.ac.rs;
jpetronijevic@mas.bg.ac.rs;
mmitic@mas.bg.ac.rs;
nvukovic@mas.bg.ac.rs;
aleksandarplemic@gmail.com;
zmiljkovic@mas.bg.ac.rs;
bbabic@mas.bg.ac.rs;