2012 2nd International Conference on Management and Artificial Intelligence
IPEDR Vol.35 (2012) © (2012) IACSIT Press, Singapore

Solving Multi-Stage Multi-Machine Multi-Product Scheduling Problem Using Bat Algorithm

Ponnapa Musikapun and Pupong Pongcharoen[1 +]

[1] Department of Industrial Engineering, Faculty of Engineering,
Naresuan University, Pitsanulok, 65000, Thailand.

Abstract. Bat Algorithm is one of the recently-developed nature-inspired methods in the field of computational intelligence. This work presents the development of the Bat Algorithm based Scheduling Tool (BAST) that used to solve multi-stage multi-machine multi-product scheduling problems. The algorithm takes into account the Just-in-Time production philosophy by aiming to minimise the combination of earliness and tardiness penalty costs. The computational experiment on the BAST was conducted using data obtained from a collaborating company engaged in capital goods industry. The experimental results indicated that the BA performance can be improved up to 8.37% after adopting the appropriate parameters' setting.

Keywords: Computational Intelligence, Bat Algorithm, Production Scheduling, Design of Experiment.

1. Introduction

Nature is always being a source of inspiration, there has been increasing interests in the development of computational models or methods that iteratively conduct stochastic search process inspired by natural intelligence. Computational intelligence can be categorised into three groups: physically-based inspiration for instance Simulated Annealing; socially-based inspiration such as Tabu Search; and biologically-based inspiration e.g. Neural Network, Genetic Algorithm, Shuffled Frog Leaping, Swarm Intelligence, Ant System, Artificial Immune System, Artificial Bee Colony, Firefly Algorithm and Bat Algorithm [1].

Bat Algorithm (BA) is relatively new metaheuristic compared with other biological inspired optimisation methods. Although Bat Algorithm seems promising for dealing with optimisation problems, very few research works related to the BA have been reported in literature. The algorithm has been completely applied to solve continuous mathematical functions [2,3] and industrial problems [4-6]. However, there is no reported on international scientific databases related to the application of the BA to solve the multi-stage multi-machine multi-product (MMM) scheduling problems.

The objectives of this paper were to: i) demonstrate the use of the Bat Algorithm based Scheduling Tool (BAST) for solving MMM scheduling problems using data obtained from a collaborating company, which manufactures complex capital goods with deep and complex product structures; and ii) compare the algorithm's performance with and without using the optimised parameter setting.

The remaining sections firstly present a brief introduction on scheduling problem in MMM environment followed by the procedure of the Bat Algorithm and it pseudo code for scheduling the manufacture and assembly of complex products, which describes in section 3. Section 4 presents the experimental design and provides a statistical analysis on the experimental results. Section 5 draws the conclusions.

2. Production Scheduling Problem

Scheduling is defined as "the allocation of resources over time to perform a collection of tasks" [7]. A schedule specifies sequence and timing, normally expressed as a set of start and due times [8]. Scheduling is a combinatorial optimisation problem that is classified as an NP hard problem [9], which means that the

[+] Corresponding author. Tel.: +66 55 964201 or 964256; Fax: +66 55 964003
E-mail address: pupongp@nu.ac.th or pupongp@gmail.com

amount of computation required to find solutions increases exponentially with problem size. Scheduling is important because companies seek to minimise lead-times and simultaneously achieve high resource utilisation.

Various assumptions have been made in order to simplify, formulate and solve scheduling problems. The most common assumptions can be summarised as follows [10]: a successor operation is performed immediately after its predecessor has finished, providing that the machine is available; each machine can handle only one operation at a time; each operation can only be performed on one machine at a time; there is no interruption of operations; there is no rework; setup and transfer times are of zero or uniform duration; and tasks are independent. Classical job shop and flow shop scheduling problems generally consist of a set of independent tasks. This is known as single stage scheduling, which means that there are no precedence constraints arising from assembly requirements.

Production scheduling in the capital goods industry is difficult for several reasons. Firstly, demand is highly variable and uncertain. The products (e.g. steam turbine generators and power station boilers) are complex and are produced from components that require a large number of operations on machines with high capital and operating cost. Different control approaches should be applied to the resources with high and low utilisation [8,11]. A final product requires assemblies, subassemblies, parts and components, in each of which a sequence of operations to be performed on multiple machines is specified. There are many operation and assembly dependency relationships. There are also multiple finite capacity resource constraints and the performance objectives may vary for different product families. Finally, the nature of production leads to large variations in product mix. Feasible schedules must correctly sequence operations and satisfy precedence constraints, and assembly relationships.

3. Bat Algorithm for Scheduling

Bat Algorithm (BA) is recently introduced by Yang [1], who is inspired by bat behaviours. Bats are the only mammals with wings and advance capability of echolocation. The development of bat-inspired algorithm was based on three idealised rules: i) Bat uses echolocation to sense a distance and differentiate between food/prey and background barriers even in the darkness; ii) Bats fly randomly to search prey with velocity (v_i), fixed frequency (f_i) and loudness (A_i); iii) the loudness varies from a large loudness (A_0) to minimum loudness (A_{min}).

The main steps of the Bat Algorithm start from initialising a swarm of bats, each of which is determined initial position (initial solution), random rate of pulse, random loudness and find frequency. During the loop, all bats will move from initial solutions toward global best solution(s). After moving, if any bats find better solution, then the bat will be update rate of pulse emission and loudness. During flying iteration, the best so far solution is updated. The flying process is repeated until termination criteria are satisfied. Finally, the best so far solution is visualised. The pseudo code of Bat Algorithm (BA) applied to solve the production scheduling problem is shown in Fig 1.

```
Objective function f(x), x = (x_1,...,x_d)^T
Initialize the bat population x_i (i = 1,2,...,n) and v_i
Define pulse frequency f_i at x_i
Initialize pulse rates r_i and the loudness A_i
while (t < Max number of iterations)
Generate new solutions by adjusting frequency, and updating velocities and location/solutions
if (rand > r_i)
Select a solution among the best solutions
Generate a local solution around the selected best solution
end if
Generate a new solution by flying randomly
if (rand < A_i & f(x_i) < f(x_*))
Accept the new solutions
Increase r_i and reduce A_i
end if
Rank the bats and find the current best x_*
end while
Postprocess results and visualization
```

Fig. 1: Pseudo code of the Bat Algorithm adopted from [1].

The BA based scheduling program was coded in modular style using a general purpose programming language called C#. The scheduling program developed can be categorised into three phases: i) input phase, in which product's information and its manufacturing data including operation no., machining time, part code, product structure and resource were uploaded into the program; ii) scheduling phase, where the proposed algorithms were used to generate and evaluate schedules constrained by precedence relationships and finite resource capacity; and iii) output phase including information on the best production schedule found and its penalty cost. Graphic user interface was considered during the development of the program to allow users to manipulate data, set parameters and choose outputs from the program.

4. Experimental Design and Analysis

This section presents the design and analysis on the computational experiments conducted using the Bat Algorithm (BA) to solve production scheduling problem obtained from a collaborating company engaged in capital goods industry. Scheduling data including resources profile, products' information, manufacturing and operational data as well as customer due date obtained from a collaborating company engaged in make/engineer to order capital good industry were experimented as a case study. The considered problem consisted of two products, each of which has four levels of product structure. To manufacture both products, it involves a combined requirement of 118 machining operations and 17 assembly operations to be performed on 17 non-identical machines (resources).

The experiment was aimed to systematically investigate the appropriate setting of the BA parameters via the statistical design and analysis. Full factorial experimental design (3^k) shown in table 1 was adopted in this work. Table 1 shows the BA parameters and its levels considered in this work. The first parameter is the combination of the amount of bats (n) and the number of maximum iterations (I). This combination factor plays an important role on the amount of search in the solution space conducted by the BA. Higher values of both parameters increase the probability of finding the best solutions but require longer computational time. If there is no timing constraint and limitation on computing resources, the values of both parameters should be defined as high as possible. In this work, the experiment was based on timing constraint scenario. The amount of search (the combination of nI) for the instants problem is predefined at 2,500. The remaining factors are the rate of pulse emission coefficient (γ) and the loudness coefficient (α) [12]. Both factors were considered in the range between zero to one.

Table 1. Bat Algorithm's parameters and its levels considered.

Factors	**Levels**	**Uncoded Values**		
		Low(-1)	***Medium(0)***	***High(1)***
Combination of nI	3	25*100	50*50	100*25
Rate of pulse emission coefficient (γ)	3	0	0.5	1
Loudness coefficient (α)	3	0	0.5	1

The proposed design was computationally experimented with five replications and simulated on personal computers with AMD Turion^{TM} 1.9 GHz CPU and 2.0 GB DDRII RAM. The computational results obtained from 135 (3^3*5) runs were analysed using a general linear model form of analysis of variance (ANOVA). Table 2 shows ANOVA table consisting of Source of Variation, Degrees of Freedom (DF), Sum of Square (SS), Mean Square (MS), F and P values. A factor with value of $P \leq 0.05$ was considered statistically significant with 95% confidence interval.

It can be seen that the combination of nI and the loudness coefficient (α) were statistically significant while the rate of pulse emission coefficient (γ), interaction between nI and rate of pulse emission coefficient (γ) and interaction between nI and loudness coefficient (α) were found insignificant for the range considered.

The main effect plots on the levels of the BA's parameter against the computational results obtained from the BA based scheduling program are illustrated in Fig 2. It can be seen that the combination of the amount of bats and the number of maximum iterations (nI) has large impact on the total penalty cost associated with the schedule received. With the limited amount of search of 2,500 candidate solutions, the best result was achieved when the nI combination was set at 25*100. This indicated that the higher number of iterations is more preferable than the number of maximum bats. Another significant factor, the rate of pulse emission coefficient (γ), was performed best with at the value of 0.5 or 1. The best condition of the loudness coefficient (α) was found at a value of 1.

Table 2. ANOVA on the BA's parameters.

Source	DF	SS	MS	F	P
nI	2	1867303704	933651852	21.66	0.000
γ	2	83003704	41501852	0.96	0.385
α	2	302137037	151068519	3.50	0.033
$nI*\gamma$	4	274607407	68651852	1.59	0.181
$nI*\alpha$	4	121240741	30310185	0.70	0.591
Error	120	5173022222	43108519		
Total	134	7821314815			

Fig. 2: Main effect plots on the BA parameters.

For studying the performance comparison on the best so far schedules produced by the BA based scheduling program with and without using the appropriate parameter setting mentioned above, the error bar of the penalty costs obtained from the program with and without using optimised setting is provided in Fig 3. It can be seen that the average total penalty cost associated with the schedules obtained using the Bat Algorithms (BA) optimised setting was 99,566.7 Baht whilst 108,667 Baht was the average total penalty cost for those without using the optimised setting. The proposed can therefore be improved by 8.37%. This emphasises that the BA performance can be improved significantly after adopting the optimised parameter setting.

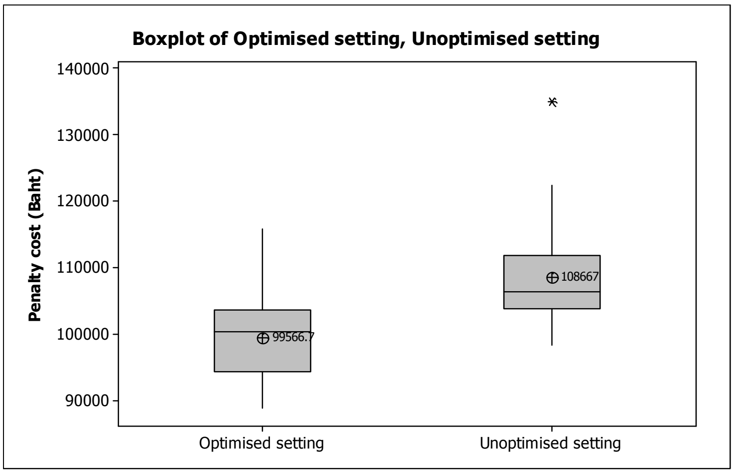

Fig. 3: Average total penalty cost associated with the schedules obtained.

5. Conclusions

This paper presents the development of Bat Algorithm based Scheduling Tool (BAST) for solving multi-stage multi-machine multi-product scheduling problem. The algorithm was designed to minimise the combination of earliness and tardiness penalties cost and correctly sequence operations required to manufacture components and also satisfy assembly precedence relationship. The statistical analysis on the experimental results indicated that the quality of the solutions obtained from the proposed algorithm can be improved magnificently after adopting the appropriate parameters' setting identified by the statistical tools. The BA's results using optimised setting outperformed those using non-optimised setting.

6. Acknowledgements

The authors would like to acknowledge Naresuan University for financial support.

7. References

[1] X. S. Yang. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, Berlin, 2010.

[2] P. W. Tsai, et.al. Bat Algorithm Inspired Algorithm for Solving Numerical Optimization Problems. *Applied Mechanics and Materials*. 2011, **148-149**: 134-137.

[3] X. S. Yang. Bat algorithm for multi-objective optimization. *International Journal of Bio-Inspired Computation*. 2011, **3**(5): 267-274.

[4] T.C. Bora, L.S. Coelho, L. Lebensztajn. Bat-Inspired Optimization Approach for the Brushless DC Wheel Motor Problem. IEEE *Transactions*. 2012, **48**(2): 947-950.

[5] K. Khan, A. Nikov, and A. Sahai. *Third International Conference on Software, Services and Semantic Technologies S3T 2011*. Springer, Berlin, 2011.

[6] A.L. Tamiru, and M.H. Fakhruldin. Application of Bat Algorithm and Fuzzy Systems to Model Exergy Changes in a Gas Turbine. In: Artificial Intelligence. *Evolutionary Computation and Metaheuristics (AIECM).* 2011.

[7] K.R. Baker. *Introduction to Sequencing and Scheduling*. Wiley and Sons, New York, 1974.

[8] C. Hicks, and P. Pongcharoen. Dispatching rules for production scheduling in the capital goods industry. *International Journal of Production Economics*. 2006, **104**(1): 154-163.

[9] J.R. King, and A.S. Spackis. Scheduling: bibliography and review. *International Journal of Physical Distribution and Materials Management*. 1980, **10**:105-32.

[10] P. Pongcharoen, A.Khadwilard, and C. Hicks. A genetic algorithm with a new repair process for solving multi-stage, multi-machine, multi-product scheduling problem. *Industrial Engineering and Management System*. 2008, **7**(3): 204-213.

[11] C. Hicks, and P. Pongcharoen. Applying different control approaches for resources with high and low utilisation: a case study of the production of complex products with stochastic processing times**.** *International Journal of Technology Management*. 2009, **48**(2): 202-218.

[12] X.S. Yang. *Stochastic Algorithms: Foundations and Applications*. Springer, Berlin, 2009.