

Computer based Scheduling Tool for Multi-product Scheduling Problems

Adirake Chainual, Tawatchai Lutuksin and Pupong Pongcharoen

Department of Industrial Engineering, Faculty of Engineering, Naresuan University,
Pitsanulok, Thailand 65000

Email: pupongp@yahoo.com and pupongp@nu.ac.th

Abstract

In this paper, the development of the Ant Colony Optimisation based Scheduling Tool (ACOST) for solving production scheduling problems in multi-product multi-stage multi-machine environment was described. The algorithms took into account the Just-in-Time philosophy aiming to minimise the costs of earliness and tardiness in the finite resource capacity scenario. A sequential experiment was adopted using data obtained from a collaborating company. The first experiment was designed to investigate appropriate parameters' setting, which was then used in the second experiment aiming to compare the performance of the algorithms based on the problem sizes.

Keywords: Product Scheduling, Ant Colony Optimisation, Metaheuristics.

1. Introduction

Scheduling was defined by Baker (1974) as "the allocation of resources over time to perform a collection of tasks". A schedule specifies sequence and timing, normally expressed in terms of a set of start and due times. Scheduling is a combinatorial optimisation problem and is classified as NP hard problem (King and Spackis, 1980), which means that the

amount of computation required to find the solutions increases exponentially with problem size. For example, if n independent jobs are to be performed on m resources (machines), there are potentially $(n!)^m$ sequences.

Production scheduling with multiple stages multiple machine environments is a problem usually faced by capital goods companies that produces complex products with deep and complex product structures, giving rise to many stages of assembly relationships. Feasible schedule must consider the sequences of precedence operation constraints, the assembly relationships based on the product structure and must avoid the deadlock situation.

Research on scheduling problems has mainly focused upon classical flow/job shop scheduling (Nagar et al., 1995; Ruiz and Maroto, 2005; Vallada et al., 2007). Research considering multi-stage multi-product scheduling problems with finite capacity multi-resource constraints is still limited. Reeja and Rajendran (2000) have highlighted the lack of production scheduling being concerned with assembly relationships and constraints.

The objective of this paper is to describe a computer aided scheduling software called the Ant Colony Optimisation

based Scheduling Tool (ACOST) that has been developed for scheduling multiple products with deep and complex product structures using various non-identical machines. The algorithms are used to generate feasible schedules based on just in time philosophy aiming to minimise the combination of earliness and tardiness penalties in the finite resource capacity scenario.

The next section of the present paper is section 2, which describes production scheduling problems in multi-product multi-stage multi-machine environment and its performance measure. Section 3 presents the architectural design of the Ant Colony Optimisation based Scheduling Tool (ACOST) program. A case study including two scheduling problems used for testing the program is explained in section 4. Section 5 presents the experimental design and provides an analysis of the results followed by the conclusions in the last section.

2. Production Scheduling Problem in Multi-Product Multi-Stage Multi-Machine Environment

Multi-stage multi-machine production scheduling of multi-product especially in capital goods industries is difficult for several reasons. First, the demand is highly variable and uncertain. Second, the industries usually produce capital products (e.g. steam turbine generators, power station boilers and transformer), most of which requires a large number of operations on machines with high capital and operating cost. There are many dependency relationships regarding the product structure, in which a final product requires assemblies, subassemblies and components. There are also multiple finite capacity resource constraints.

Manufacturing and assembly processes are diverse in terms of technology, capacities and production volumes. Each operation features a number of activities of varying duration such as set-up, machining and transfer time. These are characterised by multiple resources constraints and complex precedence relationships between operation and assembly sequence. The precedence relationships require the indicator (e.g. part number, operation number, etc.). Resources constraints also need indicators such as machine number. Assembly and operation line precedence is indicated by product structure identifiers and a product instance identifiers.

Due to the high capital and operating cost, many companies manufacture capital goods based on Just in Time philosophy by minimising earliness and tardiness delivery. A schedule with early completion may lead to high holding cost including inventory costs of finished goods and work-in-process whilst tardiness cost may be penalised for lateness delivery. In the present paper, the performance measure or objective function used for evaluating a schedule is to minimise total penalty costs of earliness and tardiness as given in equation (1).

$$\text{Total costs} = \text{earliness costs} + \text{tardiness costs} \\ = \sum \sum P_e(E_c + E_p) + \sum P_t(T_p) \quad (1)$$

Where

c = component c^{th} in a set of components (C)

p = product p^{th} in a set of products (P)

$E_c = \max(0, D_c - F_c)$

$E_p = \max(0, D_p - F_p)$

$T_p = \max(0, F_p - D_p)$

Notation E_c = Earliness of component (days)

E_p = Earliness of final product (days)

T_p = Tardiness of final product (days)

P_e = Penalty rate of earliness (£ per day)

P_t = Penalty rate of tardiness (£ per day)

D_c = Due date of component (date)

F_c = Finish time of component (date)

D_p = Due date of final product (date)

F_p = Finish time of final product (date)

3. Ant Colony Optimisation based Scheduling Tool (ACOST)

Ant colony optimisation (ASO) is one of the recent emerging metaheuristics, in which a colony of the collaborative artificial ants is initially imitated to find a shortest path between source and food (Colormi et al., 1994; Dorigo et al., 1996; Dorigo and Stutzle, 2004). The method includes two main phases: tour construction by each ant using probability calculation and heuristic information updating from pheromone evaporation for obsolete tour and pheromone deposition for a famous tour. ACO methods can be classified as Ant System (AS), Elitist Ant System (EAS), Rank based Ant System (AS_{Rank}), Max-Min Ant System (MMAS) and Ant Colony System (ACS) referred to Dorigo and Stutzle (2004) for more details.

In the present paper, computer aided production scheduling called Ant Colony Optimisation based Scheduling Tool (ACOST) including Ant System (AS), Ant Colony System (ACS), Elitist Ant System (EAS), Max-Min Ant System (MMAS) and Rank-based Ant System (AS_{rank}) was developed for scheduling the manufacture of multi-stage products, in which components require a number of operations to be performed on several machines.

ACOST program including graphic user interface and algorithms was written for more than 4,000 lines of codes in a modular style using Tcl/Tk programming language. The architectural design of the ACOST program (see Figure 1) can be categorised into three phases: i) input phase, in which product's information and its manufacturing data including part code, product structure and part instance

identifiers were imported into the program; ii) scheduling phase, where the embedded 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 (GUI) was considered during the development of the program to allow users to manipulate data, set parameters and choose outputs from the program.

4. A Case Study

Two sizes of scheduling problems including product's information and its manufacturing data obtained from a collaborating company currently engaged in make/engineer to order capital good industry were experimented as a case study (see Table 1). Both problems contain two products, each of which has four levels of product structure. The first problem (prob-I) involved 2 different products (229 and 451), with a combined requirement of 67 operations to be performed on 7 non-identical machines (resources). Whilst the second problem (prob-II), which is relatively larger, requires 135 operations on 17 machines.

Table 1 Industrial scheduling problems

Problem size	Part number	Characteristics of scheduling problems				
		Number of Products	Number of Components	Machining / Assembly Operations	Number of Resources	Levels of product structure
1	229 & 451	2	8	57/10	7	4
2	4 & 228	2	12	118/17	17	4

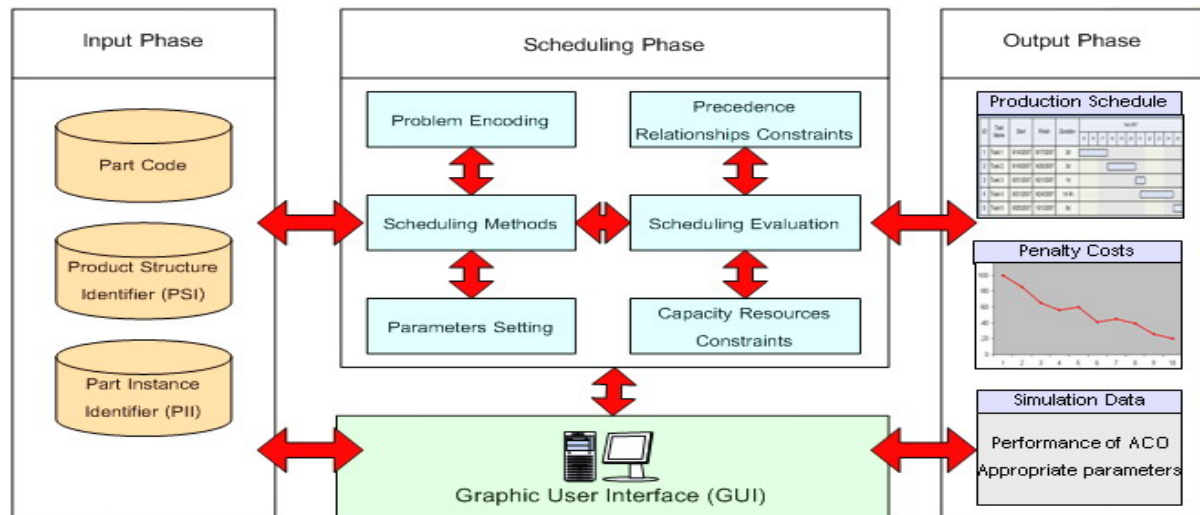


Figure 1 An architectural design of ACOST program.

5. Experimental Design and Analysis

A sequential experiment was adopted in this present work. The first experiment was designed to investigate appropriate parameters' setting required by the ACO methods using the small problem. ACO parameters considered as factors, each of which has three levels, are listed in Table 2.

Table 2 Experimental factors and its levels

Factors	Levels	Values		
		Low	Medium	High
Ants/Iterations (A/I)	3	45/20	30/30	20/45
Pheromone Weight (α)	3	0.5	1	1.5
Heuristic Information Weight (β)	3	2	3.5	5
Pheromone Evaporation Rate (ρ)	3	0.1	0.5	0.9

The first factor was the combination of a number of ants and a number of iterations (A/I), which influences on the exploration in the solution space and execution time spending on the ACO process. The amount of search for each level was fixed at 900. The remaining three factors (including pheromone weight (α), heuristic information (β) and pheromone evaporation rate (ρ) were specified based on previous research suggestions (Dorigo and Stutzle, 2004).

The one-third fractional factorial (3^{k-1}) experimental design was adopted to reduce the number of computational runs due to the combinatorial explosions of ACO parameters and its level (Montgomery, 2001). Using the proposed design, a total numbers of program executions were decreased from 81 to 27 runs for each ACO method per replication. A computational experiment was carried out using a personal computer with Pentium D 2.8 GHz CPU and 512 MB RAM. The experimental results obtained from five ACO methods, each of which was repeated 5 replicates by varying random seed number, were then analysed using a general linear form of analysis of variance (ANOVA) and main effect plots using the statistical software package called Minitab.

The F and p values of the main factors for each type of ACO are shown in Table 3. The significance of ACO parameters was varying case by case depending on the types of ACO used. However, pheromone weight (α) and heuristic information (β) were statistically significant with a 95% confident interval with the p value less than or equal to 0.05 for all five ACO methods. On the other hand, the combination of a number of

ants and a number of iterations (A/I) with the range considered were insignificant in all cases.

Table 3 Analysis of Variance (ANOVA) of five ACO methods

Source	DF	ACO Methods									
		AS		EAS		AS _{rank}		MMAS		ACS	
		F	p	F	p	F	p	F	p	F	p
A/I	2	0.04	0.965	0.33	0.718	1.55	0.215	0.69	0.501	2.30	0.105
α	2	7.44	0.001	8.01	0.001	7.67	0.001	3.36	0.038	3.42	0.036
β	2	27.98	0.000	14.25	0.000	20.79	0.000	23.46	0.000	10.78	0.000
ρ	2	2.06	0.132	1.99	0.141	1.65	0.196	3.39	0.037	23.40	0.000
Error	122										
Total	134										

Using main effect plot on significant factors, it suggested that the best results using AS and MMAS were obtained from the setting of α , β and ρ at 1.0, 2 and 0.1, respectively. For the remaining ACO methods, the appropriate parameters' setting of α , β and ρ were recommended at 0.5, 2 and 0.1, respectively. These findings were then used for the sequential experiment.

The second experiment was aimed to compare the performance of five ACO methods, each of which was based on their best appropriate parameters' setting discovered from the previous experiment.

This computational experiment was repeated five times using five ACO methods to solve both problem sizes. Table 4 shows the best solution found so far, the mean and standard deviation (SD) values with different problem sizes. It can be seen that the MMAS and EAS generally produced the schedules with the lowest penalty costs followed by the AS, AS_{rank} and ACS. However, the AS_{rank} produced results with lower variance. The average execution time required by ACO methods for prob-I was 18 minutes and 40 minutes for the larger problem (prob-II).

Table 4 The computational results of different size problems obtained from ACO methods

ACO Methods	Problem Size					
	Medium			Large		
	Best so far	Mean	SD	Best so far	Mean	SD
Ant System (AS)	12000	13600	1193.73	63000	65900	3008.32
Elitist Ant System (EAS)	11000	11600	418.33	62500	66100	3267.26
Rank-based Ant System (AS _{rank})	11500	12100	418.33	65000	66700	2539.69
Max-Min Ant System (MMAS)	11000	11400	547.72	59000	64600	4204.17
Ant Colony System (ACS)	12000	13600	1635.54	88000	91200	3094.35

5. Conclusions

In the present paper, the development of the Ant Colony Optimisation based Scheduling Tool (ACOST) for solving production scheduling problems in multi-product multi-stage multi-machine environment was described. The algorithms used the Just-in-Time philosophy aiming to minimise the costs of earliness and tardiness in the finite resource capacity scenario. A sequential experiment was adopted using data obtained from a collaborating company. The experimental results suggested that the performance of ACO methods depended on its parameter setting.

References

- Baker, K.R. (1974). *Introduction to Sequencing and Scheduling*, Wiley and Sons, New York.
- Colomi, A., Dorigo, M., Maniezzo, V., Tubian, M. (1994). "Ant System for job-shop scheduling". *JORBEL-Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1), 39-53.
- Dorigo, M., Stutzle, T. (2004). *Ant Colony Optimization*, A Bradford Book, Massachusetts.
- King, J.R., Spackis, A.S. (1980), "Scheduling: bibliography and review", *International Journal of Physical Distribution and Materials Management*, 10, 105-132.
- Montgomery, D.C. (2001). *Design and Analysis of Experiments*, Fifth edition, John Wiley and Sons, New York.
- Nagar, A., Haddock, J., Heragu, S. (1995). "Multiple and bicriteria scheduling: a literature survey", *European Journal of Operational Research*, 81, 88-104.
- Reeja, M.K., Rajendran, C. (2000). "Dispatching rules for scheduling in assembly job shops – Part I", *International Journal of Production Research*, 38(9), 2051-2066.
- Ruiz, R., Maroto, C. (2005). "A comprehensive review and evaluation of permutation flowshop heuristic", *European Journal of Operational Research*, 165, 479-494.
- Vallada, E., Ruiz, R., Minella, G. (2007), "Minimizing total tardiness in the m-machine flowshop problem: a review and evaluation of heuristics and metaheuristics", *Computers and Operations Research*, Article in press.