ELSEVIER

Int. J. Production Economics 77 (2002) 159–171

international journal of production economics

www.elsevier.com/locate/dsw

A list-based threshold accepting method for job shop scheduling problems

C.D. Tarantilis, C.T. Kiranoudis*

Department of Chemical Engineering, National Technical University of Athens, Zografou Campus, Athens 15780, Greece

Received 15 November 2000; accepted 13 December 2001

Abstract

Process plants typically produce a family of related products that require similar processing techniques. The most important problem encountered in such manufacturing systems is scheduling of operations so that demand is fulfilled within a pre-described time horizon imposed by production planning. The typical scheduling operation that process plants involve can be formulated as a general job shop scheduling problem. The aim of this study is to describe a new metaheuristic method for solving the job shop scheduling problem of process plants, termed as list-based threshold accepting (LBTA) method. The main advantage of this method over the majority of other metaheuristics is that it produces quite satisfactory solutions in reasonable amount of time by tuning only one parameter of the method. This property makes the LBTA a reliable and a practical tool for every decision support system designed for solving real life scheduling problems. The LBTA is described in detail, tested over classical benchmark problems found in literature and a while characteristic job shop scheduling case study for dehydration plant operations is presented. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Job shop scheduling; Metaheuristics; Threshold accepting

1. Introduction

Scheduling of process operations is of utmost importance for the operational integrity of a process plant. Scheduling operations focus on the customized problem of meeting demand in the form of orders that are related to the character of each plant. The majority of scheduling operations in process plants can be formulated in terms of job shop scheduling. Job shop problems are known to be problems of extreme computational complexity where exact algorithms (such as Branch and Bound techniques) are doomed to fail. An effective alternative is heuristic schemes. Nowadays, researchers strive to find modern local search heuristics, termed as metaheuristics, producing high quality solutions for large problems that unfortunately are of hard computational complexity for utilizing exact algorithms (that in turn guarantee optimality).

The development of such a metaheuristic method for solving the job shop scheduling problems of process plants, termed as list-based threshold accepting (LBTA), is the main contribution of this paper. The LBTA belongs to the class of threshold accepting algorithms. Its main

*Corresponding author. Tel.: +301-772-1503; fax: +301-772-2328.

E-mail address: kyr@chemeng.ntua.gr (C.T. Kiranoudis).

0925-5273/02/$ - see front matter
PII: S0925-5273(01)00231-6

difference over a typical threshold-accepting algorithm is that the threshold values used in the implementation of the move acceptance criterion are determined by a list that is rejuvenated and adapted according to the topology of the solution space of the problem. In this paper, the LBTA method is described in detail and its performance and characteristic case studies for the job shop scheduling problem of dehydration plants are presented.

2. Job shop scheduling

Production planning strategies in the case of most manufacturing systems decide upon certain operational variables that account for the level of production, the fulfillment of customer orders and the level of warehouse stock in specific periods within a time planning horizon. The information flow of a typical manufacturing system is shown in Fig. 1. The plant operator will take into consideration the objectives derived in the production planning case and subsequently will assign products and times to machines so that production will be achieved within the specific time horizon. Moreover, in the case of process plants, out-of-program orders may come up that demand rapid fulfillment. The procedure of assigning products and defining sequence of operations to machines that typically involve a series of different pre-treatment and process operations is called operation scheduling. One of the most important problems of scheduling operation tackled by Operations Research is the job shop problem that consists of scheduling a set of jobs on a set of machines with the objective to minimize the makespan, i.e. the maximum of completion time for processing all jobs, subject to the constraint that each job has a specified processing order through the machines and that each machine can process at most one job each time.

Job shop problems, described as $J/C_{\max}$ using the three-field notation of Graham et al. [1], are NP-hard [2]. Each instance of the job shop problem is defined by a set of jobs, a set of machines and a set of tasks. Each job consists of a sequence of tasks, each of which has to be performed on a given machine for a given time. A *schedule* is an allocation of the operations to time intervals on machines. The problem is to find the schedule that minimizes the *makespan*, the maximum of completion times for processing all jobs subject to the following constraints:

- (i) the precedence of tasks given by each job are to be respected,
- (ii) each machine can perform at most one task at a time,
- (iii) the operations cannot be interrupted.

Let $J = \{1, \dots, n\}$ denote the set of jobs, $M = \{1, \dots, m\}$ denote the set of machines, $V \subset M \times N \cup \{0, n+1\}$ denote the set of job tasks $[k, i]$, where $k \in M, i \in J$ and 0 and $n+1$ represent dummy start and finish tasks, respectively, $a^i = \{a^i_1, a^i_2, \dots, a^i_{m_i}), a^i_j \in M, a^i_k \neq a^i_l, k \neq l\}$, where $m_i \leqslant m$, $i = 1, 2 \dots, n$, and $[a^i_k, i]$ precedes $[a^i_{k+1}, i]$ as dictated by the precedence relations defined in (i) denote the series of machines of tasks performed by job i, $e^k = \{[k, i]/i \in J, k$, is an element of the sequence $\alpha^i\}$ where $k = 1, 2, \dots, m$ dictated by the sequence relations defined in (ii) denote the set of tasks performed by machine k, $A^i = \{([0], [\alpha^i_1, i]), ([\alpha^i_1, i], [\alpha^i_2, i]), \dots, ([\alpha^i_{m_i}, i], [n+1])\}$ where $i = 1, 2, \dots, n$ be the set of pairs of consecutive tasks to be performed by job i as specified in a^i, $E^k = \{\{[k, i], [k, j]\}/i \neq j$ and $[k, i], [k, j] \in e^k\}$ where $k = 1, 2, \dots, m$ be the set of all two element subsets of tasks to be performed by machine k and which therefore have to be sequenced as specified in (ii) and p_u, $u \in V$, denote the (fixed) processing time of the task u, respectively. The processing time of the 0 and $n+1$ tasks are equal to zero, i.e. $p_0 = p_{n+1} = 0$. In this framework, the job shop scheduling problem can be represented in terms of a disjunctive graph $G = (V, A, E)$, where $V = \bigcup_{k=1}^{m} e^k \cup \{0, n+1\}$ is the set of nodes of the graph (tasks), $A = \bigcup_{i=1}^{n} A^i$ is the set of ordinary arcs (conjunctive) of the graph (imposed by precedence constraints (i)) and $E = \bigcup_{k=1}^{m} E^k$ is the set of disjunctive arcs of the graph (to be sequenced as suggested by (ii)). The nodes of G correspond to tasks, the direct arcs to precedence relation between tasks of the same job and the disjunctive arcs to tasks to be performed on the same machine. Each disjunctive arc of E can be