Int. J. Production Economics 153 (2014) 253–267

Contents lists available at ScienceDirect

Int. J. Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships

Andrea Rossi *

Department of Civil and Industrial Engineering, Università di Pisa, Largo Lucio Lazzarino, Pisa 56126, Italy

ARTICLE INFO

Article history:
Received 14 November 2011
Accepted 8 March 2014
Available online 18 March 2014

Keywords:
Parallel machines
Metaheuristics
Swarm systems
Benchmark problems
Computation time

ABSTRACT

This paper proposes a swarm intelligence approach based on a disjunctive graph model in order to schedule a manufacturing system with resource flexibility and separable setup times. Resource flexibility assigns each operation to one of the alternative resources (assigning sub-problem) and, consequently, arranges the operation in the right sequence of the assigned resource (sequencing sub-problem) in order to minimize the makespan. Resource flexibility is mandatory for rescheduling a manufacturing system after unforeseen events which modify resource availability. The proposed method considers parallel (related) machines and enforces in a single step both the assigning and sequencing sub-problems. A neighboring function on the disjunctive graph is enhanced by means of a reinforced relation-learning model of pheromone involving more effective machine-sequence constraints and a dynamic visibility function. It also considers the overlap between the jobs feeding and the machine (anticipatory) setup times. It involves separable sequence-independent and dependent setup phases. The algorithm performance is evaluated by modifying the well-known benchmark problems for job shop scheduling. Comparison with other systems and lower bounds of benchmark problems has been performed. Statistical tests highlight how the approach is very promising. The performance achieved when the system addresses the complete problem is quite close to that obtained in the case of the classical job-shop problem. This fact makes the system effective in coping with the exponential complexity especially for sequence dependent setup times.

1. Introduction

A common aim in the practical job shop environment is to improve resource flexibility and setup lag times. Resource flexibility deals with flexible (or hybrid) job shop scheduling (FJS) where alternative resources are present to increase performance, to manage preventive maintenance or to tackle breakdown and other unforeseen events which modify resource availability. The FJS problem is thus to determine both an assignment of each operation to one of the alternative resources (*assignment sub-problem*) and an ordering of the operations on each assigned resource (*sequencing sub-problem*) with the aim of optimizing an objective function.

The FJS problem arises in at least two types of workshop. The first is a flexible manufacturing system where a small number of multi-purpose machines are equipped with different tools and a number of modes (*multiple modes*) are allowed to perform each operation. A FJS has a *total flexibility* if any operation can be processed by each machine present in the system. This method gives the required resource flexibility but the managing of machine capabilities to meet the tolerances of design (process planning) is a very difficult problem.

In the great majority of practical industrial applications, however, resource flexibility is combined with scheduling operations on alternative (identical) machines. It consists of workshops with a partition of available machines into groups of parallel machines tools. The machines of the same group (e.g. lathes, milling machines, washing/sterilization machines, measuring machines, assembly robots, etc.) are *related*: they group the manufacturing capability in order to process a set of technologically similar operations and hence, for example, they include equal processing and setup times (Stecke and Raman, 1995). If the jobs have an identical routing among the groups, the problem is the hybrid (non-permutation) flow-shop scheduling.

Job shop scheduling with the objective of makespan minimization $J_m||C_{max}$, which only deals with the sequencing problem, is strongly NP-hard (Garey et al., 1976). An extensive and rapidly growing series of approaches are proposed; nevertheless, only a

* Tel.: +39 50 2218123; fax: +39 50 2218140.
E-mail address: andrea.rossi@dimnp.unipi.it

http://dx.doi.org/10.1016/j.ijpe.2014.03.006
0925-5273/

few special cases can be optimally solved with effective computing times (see Jain and Meeran, 1999; Blazewicz et al., 1996 for a review). As it is an extension of job shop scheduling, the flexible job shop scheduling $FJ_m||C_{max}$ is NP-hard as well. Flexible job shop scheduling has only been treated in recent literature by a number of approaches where the resource flexibility is achieved by multiple modes with partial flexibility (Brucker and Schlie, 1990; Hurink et al., 1994; Brucker and Thiele, 1996; Dauzère-Pérès et al., 1998; Mastrolilli and Gambardella, 2000; Kacem et al., 2002; Kumar et al., 2003; Chan et al., 2006). In addition, Kacem et al. (2002) and Chan et al. (2006) consider FJS with total flexibility.

Compared with the extensive research on FJS where the resource flexibility is achieved by multiple modes, in the last years the systems and the applications to solve the parallel machine job shop scheduling problem has not received sufficient attention. Besides, the majority of FJS systems assume release date of jobs and resources, operation setup and job transportation (travel) times as negligible or part of the processing time. While these assumptions simplify the analysis in certain applications, they adversely affect the solution quality for many applications which require explicit treatment of these time lags. Such applications have motivated an increasing interest to include setup considerations, in order to reduce costs.

According to the $\alpha|\beta|\gamma$ notation of Graham et al. (1979), the problem under consideration can be denoted by $PJ_m(k)|s_{jk}, prec|C_{max}$, where field α denotes a job shop with k parallel resources per group and m groups, field β indicates the presence of sequence-dependent setup times and linear routings, i.e. the occurrence of simple precedence constraints in the job routing and, field γ denotes the makespan as the adopted measure of performance. In such a system, an operation is subjected to the following lag times: (i) sequence-dependent (*SD*) setup, the setup which depends on the previous operation processed on the resource; (ii) sequence-independent (*SI*) setup, the setup which depends on the previous operation in the job routing (i.e. job transportation) (Allahverdi et al., 2008). *Overlapping* among transportation and processing times and *anticipatory setup*, which causes the part not to be necessarily available on the resource during the setup period, involves separable *SI* and *SD* setup phases.

A number of job shop scheduling approaches assume the material handling system as a further resource where traveling operations involve non-negligible transportation times. Thus, the material handling system can be scheduled together with the resources, in order to avoid transportation costs which could influence the makespan (Artigues and Roubellat, 2001; Hurink and Knust, 2005). In such approaches, the scheduling algorithm must schedule twice the number of operations and one (or more) further resource included in the material handling system. Among the minor considerations, resource setup and transportation times can be seen as related to a single *operation setup phase* which includes separable *SD* and *SI* times. This approach reduces the number of operations in the system because no additional resource is used to model transportation times. Ivens and Lambrecht (1996) consider separable sequence-independent setup and travel times in the case of multi-stage multiprocessor flow-shop scheduling and non-linear routing. They extend the disjunctive graph (*digraph*) representation for job shop scheduling, originally proposed by Roy and Sussmann (1964). Blazewicz et al. (1996) state that the digraph model is becoming the standard model for scheduling applications because it is more efficient than Gantt diagrams to describe knowledge for optimization search techniques. Rossi and Dini (2001) propose a $PJ_m(k)|prec|C_{max}$ where separable setup and transportation times are related to a single operation setup phase and the problem knowledge for an evolutionary approach is still modeled by a Gantt diagram. A digraph approach to a case of study of the parallel machine job shop scheduling with setup lag times is proposed by Rossi and Dini (2007).

Artificial life methods have been developed in order to tackle the computational complexity of hard problems by means of a sort of implicit parallelism which offers a population-based iterative algorithm. This offers the possibility of obtaining a reactive, robust algorithm, which is basic for an industrial dynamic production process (De Jong and Spears, 1995). The Ant Colony Optimization (ACO: Bonabeau et al., 2000) is a promising metaheuristic and an emerging class of research, dealing with swarm intelligence, a set of artificial life methods which exploit the experience of an ant colony as a model of self-organisation in co-operative food retrieval by means of a proper pheromone trail model. The pheromone trail is the basic mechanism of communication among real ants and it is mimicked by the ACO in order to find the shortest path connecting source and destination on a weighted graph which represents the optimization problem. As soon as a path is generated, the artificial ant deposits on the arc a further amount of pheromone proportional to the path length and a pheromone decay routine is performed to prevent stagnation.

$J_m||C_{max}$ has been approached by an ant system (AS) proposed by Colorni et al. (1994). Kumar et al. (2003) propose an AS to approach the $FJ_m|s_{jk}, prec|C_{max}$ problem. Nevertheless, ant system has been improved by Ant Colony Optimization (ACO). Two main classes of ACO systems are proposed in literature in order to improve intensification and diversification mechanisms of ant systems: the Ant Colony System (ACS: Dorigo and Gambardella, 1997) and the MinMax Ant System (MMAS: Stutzle and Hoos, 2000). A MinMax Ant System was proposed by Blum and Sampels (2004) for solving a kind of job shop scheduling in which some routing constraints are removed. This ACO hybridizes some components of the current state-of-the-art system for job shop scheduling, the tabu search proposed by Nowicki and Smutnicki (1996), in order to outperform the *pure*-MMAS. Besides, no ACO has been extended to approach resource flexibility.

This paper describes an ACO approach to $PJ_m(k)|s_j = s_{jk}, prec|C_{max}$ problem, where the reconfiguration tasks of the resources of the same group j are standardized with a predetermined number of procedures Sj. This standardization is very important in real manufacturing systems for the efficient planning of the clamping or the batching of the parts to be produced. It is based on the digraph model of the flexible job shop scheduling problem with separable transportation and sequence-dependent setup times. The proposed system uses an algorithm similar to the list scheduler (originally proposed by Giffler and Thompson (1960), for classic $J_m||C_{max}$) to generate a feasible schedule on the digraph by visiting every operation once and only once. Here, the aim is to minimize the makespan, although an amount of results is independent of the selected objective function.

2. Job shop scheduling with resource flexibility and separable setup times

In FJS, n jobs have to be scheduled on m resources in accordance with its linear routing represented by a sequence of $l_i \le m$ operations, O_{ijr}; each of these has to be processed as the rth operation on a single resource selected within a set of resources M_{ij} ($|M_{ij}| \le m$), with a setup activity f_{ij}, which takes the time $t(f_{ij})$, and a processing time t_{ijr}; $st(O_{ijr})$ and $t(O_{ijr})$ denote respectively the starting and the completion time of the operation.

No resource can process more than one operation at a time; no operation O_{ijr} can start until O_{ijr-1} is completed or can stop after it starts; finally, an operation must be processed by one, and only one, resource.

In order to process the entire set of planned operations, the system includes F dedicated setup activities, grouped per resources, $F_1,\ldots,F_m$. A resource j includes all the equipments capable