European Journal of Operational Research 199 (2009) 55–67

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Scheduling interfering job sets on parallel machines

Hari Balasubramanian [a,*], John Fowler [b], Ahmet Keha [b], Michele Pfund [c]

[a] Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA, United States
[b] Department of Industrial Engineering, Arizona State University, Tempe, AZ, United States
[c] Department of Supply Chain Management, Arizona State University, Tempe, AZ, United States

ARTICLE INFO

Article history:
Received 12 October 2006
Accepted 28 October 2008
Available online 19 November 2008

Keywords:
Interfering job sets
Parallel machines
Bicriteria scheduling

ABSTRACT

We consider bicriteria scheduling on identical parallel machines in a nontraditional context: jobs belong to two disjoint sets, and each set has a different criterion to be minimized. The jobs are all available at time zero and have to be scheduled (non-preemptively) on *m* parallel machines. The goal is to generate the set of all non-dominated solutions, so the decision maker can evaluate the tradeoffs and choose the schedule to be implemented. We consider the case where, for one of the two sets, the criterion to be minimized is makespan while for the other the total completion time needs to be minimized. Given that the problem is NP-hard, we propose an iterative SPT–LPT–SPT heuristic and a bicriteria genetic algorithm for the problem. Both approaches are designed to exploit the problem structure and generate a set of non-dominated solutions. In the genetic algorithm we use a special encoding scheme and also a unique strategy – based on the properties of a non-dominated solution – to ensure that all parts of the non-dominated front are explored. The heuristic and the genetic algorithm are compared with a time-indexed integer programming formulation for small and large instances. Results indicate that the both the heuristic and the genetic algorithm provide high solution quality and are computationally efficient. The heuristics proposed also have the potential to be generalized for the problem of interfering job sets involving other bicriteria pairs.

1. Introduction

Traditionally, multicriteria scheduling problems have been considered with the objective of minimizing criteria that apply to each of the jobs being scheduled. While motivation for such problems can frequently be found in practice, it is also possible to have situations in which jobs belong to disjoint classes or sets, with a criterion associated with each set. The job sets in such a situation are said to compete or *interfere* with each other for the same resources. Research in the area of interfering job sets is limited. In Hoogeveen's (2005) review of multicriteria scheduling problems, he mentions the scheduling of interfering job sets as one of the new developments in the area. The work of Peha (1995) is the earliest reference on the topic. He considers the lexicographic optimization of the weighted number of tardy jobs for one set and the total weighted completion time for the other under the assumption of unit processing times, integer release dates and identical parallel machines. Peha's (1995) research is motivated by real time systems and integrated-services networks. He provides polynomial time algorithms for the problem which exploit the assumption of unit processing times.

Agnetis et al. (2003) consider the problem of two users competing for common job shop resources. Their motivation comes from the decision by two major companies to propose a joint venture to construct a flexible manufacturing system for their products. Agnetis et al. (2003) further state that discussion with these companies indicated that a decision support system that enables negotiation between the two competing users of the manufacturing system would be useful. Agnetis et al. (2003) also mention other applications where users with different goals compete with each other for the same resources: (1) scheduling multiple flights of different airlines on a common set of runways, (2) scheduling berths and material/people movers (cranes, walkways, etc.) at a port for multiple ships, (3) scheduling clerical workers among different "bosses" in an office, and (4) scheduling a mechanical/electrical workshop for different users. Baker and Smith (2003) present an example in which a prototype shop is shared by both the Research and Development department and the Manufacturing Engineering department. The Research and Development department might have concerns on meeting due-dates while the Manufacturing Engineering department might have concerns about quick response times.

In a different paper Agnetis et al. (2004) present complexity results for generating non-dominated solutions for single machine

* Corresponding author. Tel.: +1 4802763269.
E-mail addresses: hbalasubraman@ecs.umass.edu (H. Balasubramanian), john.fowler@asu.edu (J. Fowler), ahmet.keha@asu.edu (A. Keha), michele.pfund@asu.edu (M. Pfund).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2008.10.038

and shop scheduling problems given that jobs belong to one of two sets. The objectives they consider include $\sum C_j$ and $\sum U_j$. Cheng et al. (2006b) show the NP-hardness of the high multiplicity encoding version of the problem of minimizing $\sum C_j$ on a single machine with a constraint on $\sum U_j$, while Cheng et al. (2006a) show the strong NP-hardness of problem where jobs belong to one of many sets and $\sum w_j U_j$ is to be minimized for each set. Agnetis et al. (2007) tackle the computational complexity of the single machine problem involving interfering job sets and with more generally defined cost functions. Baker and Smith (2003) also consider the single machine version of the problem and show the polynomial solvability of bicriteria problems involving (1) $C_{\max}$ (2) $\sum C_j$ (3) $\sum w_j C_j$ 4) $L_{\max}$, except for the $\sum w_j C_j$, $L_{\max}$ pair, which they show to be NP-hard. Their bicriteria optimization function is a linear combination of the criteria with weights on each criterion. We note that the Baker and Smith (2003) approach of minimizing a linear combination is an *a priori* approach: information is available beforehand on the weights of the two criteria.

The problem of scheduling interfering job sets has some unique structural properties in the single machine environment. We consider the case where jobs belong to one of two disjoint sets $C1$ and $C2$. Key results from Baker and Smith (2003) that can be proved easily for regular scheduling measures by contradiction arguments are:

Property 1. *If makespan is the criteria for one of the sets, then there is an optimal schedule in which all jobs belonging to the makespan set are processed consecutively.*

Property 2. *If total completion time is the criteria for one of the sets, then there exists an optimal schedule in which jobs in the total completion time set are processed in shortest processing time (SPT) order.*

Intuitively, Property 1 tells us that since only the latest finish time of the set of makespan jobs matters, processing the jobs non-consecutively can never increase the objective value of the jobs in the other set (true for regular criteria). Property 1 has an interesting implication on the optimal schedule of jobs. Since all the jobs for which makespan is to be minimized are to be processed consecutively, they can all be accumulated into a single "makespan job". The processing time of the makespan job is the sum of the processing times of the jobs in the makespan set. Minimizing a weighted linear combination of the two objectives then reduces to the total weighted completion time problem (there could be a different weight on the makespan job and equal weights on all the $\sum C_j$ jobs), solvable using the well-known weighted shortest processing time (WSPT) rule.

If the set of all non-dominated solutions were to be generated (not considered by Baker and Smith (2003)), the makespan job would be placed at each position in the schedule (position 1, position 2, ..., position n_2+1) preceded and/or followed by the $\sum C_j$ jobs ordered by the SPT rule (due to Property 1).

This research extends the single machine research of Baker and Smith (2003) by considering the problem of two interfering jobs sets in the identical parallel machine environment. We limit our study to two well-known classical scheduling criteria: makespan and total completion time. Jobs are divided into two disjoint sets: one for which makespan needs to be minimized, and the other for which total completion time needs to be minimized. The problem is NP-hard as the single criterion problem of minimizing makespan on parallel machines is NP-hard. We propose computationally efficient heuristic techniques that can be extended with modifications to other bicriteria pairs. Our goal is to generate the set of non-dominated solutions, so the decision maker can evaluate the tradeoffs in the criteria. This is the *a posteriori* approach in which the decision maker makes his choice only after a set of points is presented. Since makespan is equivalent to the decision problem involving a common deadline (i.e. whether a feasible schedule can be obtained such that all jobs finish before a common deadline) the set of non-dominated solutions can give the decision-maker important information on whether jobs for one set can be finished by a given time and the resulting compromise or effect on the sum of completion times of the other customer's jobs.

We propose two different heuristic techniques in the paper: an iterative SPT–LPT–SPT (S–L–S) heuristic and a bicriteria genetic algorithm. Both approaches are designed specifically to exploit the problem structure and the properties of a non-dominated solution. Over the last two decades, a number of different approaches have been proposed for using evolutionary algorithms for multicriteria problems. We point to the tutorial by Landa Silva and Burke (2004b) for a concise description of the main developments in the area. We note that the genetic algorithm proposed in this research, due to its problem specific nature, is quite different from the GA approaches proposed in the literature for traditional multicriteria parallel machine scheduling problems. We propose a special encoding scheme in this research and also a strategy to explore all portions of the non-dominated front; our ideas are built to capture the structural implications of interference between jobs sets, especially when one of the criteria is makespan.

The remainder of the paper is organized as follows. We define our problem in Section 2. In Section 4, we discuss key aspects of the problem structure and propose the S–L–S heuristic. In Section 5, we describe the bicriteria genetic algorithm. We then compare, in Sections 6 and 7, the performance of the S–L–S heuristic and the genetic algorithm to the true set of non-dominated solutions generated by an integer program. Finally, the paper is summarized, overall conclusions are given and future research directions are discussed in Section 8.

2. Problem definition

We consider the problem of scheduling n jobs on m identical parallel machines. The jobs belong to one of two disjoint sets $C1$ and $C2$ with n_1 and n_2 jobs in them, respectively, such that $n_1 + n_2 = n$. For the jobs in the set $C1$, we are interested in minimizing criterion $C_{\max}$ and for the jobs in the set $C2$, we are interested in minimizing criterion $\sum C_j$. Each job has a processing time p_j and we assume that all jobs are available for processing at time zero.

In the $\alpha|\beta|\gamma$ scheduling notation of Graham et al. (1979), we refer to this problem as $P|\text{inter}|\text{ND}(C_{\max}, \sum C_j)$, where *inter* is our notation to indicate that jobs of the different sets interfere with one another, and $\text{ND}(C_{\max}, \sum C_j)$ is our notation to indicate we are interested in generating the set of non-dominated points.

The goal is to generate a set of *non-dominated* or *Pareto optimal* solutions, so the decision maker can determine the tradeoffs involved in scheduling the two sets of jobs. Let S be the set of feasible solutions for a bicriteria optimization problem with interfering job sets. Let $z_1(x)$ and $z_2(x)$ be the objective values for criteria z_1 (corresponding to set $C1$) and z_2 (corresponding to set $C2$) for a feasible solution $x \in S$ (both z_1 and z_2 need to be minimized).

Definition 3. A solution x^* is Pareto optimal or non-dominated if there exists no other solution $x \in S$ for which $z_1(x) \leqslant z_1(x^*)$ and $z_2(x) \leqslant z_2(x^*)$ where at least one of the inequalities is strict.

The problem of generating the set of non-dominated solutions for $C_{\max}$ and $\sum C_j$, given that the jobs sets corresponding to the two criteria interfere with each other, is strongly NP-hard. Indeed solving any bicriteria problem with interfering job sets involving classical scheduling criteria in the parallel machine environment is strongly NP-hard. This is because only $P\|\sum C_j$ is polynomially solvable (for an extensive compilation of complexity results for