Computers & Industrial Engineering 67 (2014) 10–19

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Application of graph search and genetic algorithms for the single machine scheduling problem with sequence-dependent setup times and quadratic penalty function of completion times ☆

Viswanathan Kodaganallur [a], Anup K. Sen [b], Subrata Mitra [b,*]

[a] Seton Hall University, 400 South Orange Avenue, NJ 07079, USA
[b] Indian Institute of Management Calcutta, D.H. Road, Joka, Kolkata 700104, India

ARTICLE INFO

Article history:
Received 29 August 2011
Received in revised form 18 June 2013
Accepted 14 October 2013
Available online 22 October 2013

Keywords:
Single machine scheduling
Sequence-dependent setup
Quadratic penalty
Graph search
Genetic algorithm

ABSTRACT

In this paper, we consider the single machine scheduling problem with quadratic penalties and sequence-dependent (QPSD) setup times. QPSD is known to be NP-Hard. Only a few exact approaches, and to the best of our knowledge, no approximate approaches, have been reported in the literature so far. This paper discusses exact and approximate approaches for solving the problem, and presents empirical findings. We make use of a graph search algorithm, Memory-Based Depth-First Branch-and-Bound (MDFBB), and present an algorithm, QPSD_MDFBB that can optimally solve QPSD, and advances the state of the art for finding exact solutions. For finding approximate solutions to large problem instances, we make use of the idea of greedy stochastic search, and present a greedy stochastic algorithm, QPSD_GSA that provides moderately good solutions very rapidly even for large problems. The major contribution of the current paper is to apply QPSD_GSA to generate a subset of the starting solutions for a new genetic algorithm, QPSD_GEN, which is shown to provide near-optimal solutions very quickly. Owing to its polynomial running time, QPSD_GEN can be used for much larger instances than QPSD_MDFBB can handle. Experimental results have been provided to demonstrate the performances of these algorithms.

1. Introduction

Single machine scheduling problems have been widely studied (Pinedo, 1995). One version of the problem that has been extensively dealt with in the literature is the consideration of sequence-dependent setup times (Ang, Sivakumar, & Qi, 2009; Anghinolfi & Paolucci, 2009; Biskup & Herrmann, 2008; Choi & Choi, 2002; Gupta & Smith, 2006; Kim & Lee, 2009; Koulamas & Kyparisis, 2008; Liao & Juan, 2007; Lin & Ying, 2008; Luo & Chu, 2007; Luo, Chu, & Wang, 2006; Nekoiemehr & Moslehi, 2011; Tasgetiren, Pan, & Liang, 2009; Valente & Alves, 2008; Wang, 2008; Wang & Tang, 2010; Zhao & Tang, 2010). Another version of the problem that has drawn limited attention is the consideration of quadratic penalty functions of job completion times (Bagchi, Chang, & Sullivan, 1987b; Bagchi, Sullivan, & Chang, 1987a; Bagga & Kalra, 1980; Croce, Szwarc, Tadei, Baracco, & di Tullio, 1995; Gupta & Sen, 1984; Mondal & Sen, 2000a; Sen, Dileepan, & Ruparel, 1990; Szwarc, Posner, & Liu, 1988; Townsend, 1978). In this paper, we consider the single machine scheduling problem with quadratic penalties and sequence-dependent (QPSD) setup times. In QPSD, there are N jobs, J_i, $i = 1, \ldots, N$, all of them available at time 0. These jobs are to be processed on a machine one after the other. Associated with J_i are the processing times, a_i, penalty coefficients, p_i, and setup times, $s_{i,j}$ (being the setup time for J_j when it is immediately preceded by J_i). The objective is to minimize the total penalty across all jobs, i.e. to minimize the weighted sum of the squares of the completion times. When the setup times are sequence-independent, they can simply be added to the processing times for the corresponding jobs and thus possess no additional complexity over problems without setup times. However, when the setup times are sequence-dependent, the quadratic penalty problem becomes extremely difficult to solve. It is interesting to note that minimizing the weighted sum of the completion times (i.e. the linear penalty case) can be transformed and treated as if the setup times are sequence-independent (Sen & Bagchi, 1996). However, no such transformation is possible for the non-linear penalty case with sequence-dependent setup times. In this paper, we attempt to advance the state of the art for solving the QPSD problem.

Single machine scheduling problems with N jobs have $N!$ possible distinct sequences, and except for special cases, it is known that these problems are NP-Hard (Rinooy Kan, 1976), i.e. finding an optimal solution requires an implicit enumeration of

☆ This manuscript was processed by Area Editor Subhash C. Sarin, Ph.D.

* Corresponding author. Tel.: +91 33 24678300.
E-mail addresses: kodagavi@shu.edu (V. Kodaganallur), sen@iimcal.ac.in (A.K. Sen), subrata@iimcal.ac.in (S. Mitra).

0360-8352/$ - see front matter
http://dx.doi.org/10.1016/j.cie.2013.10.005

all possible sequences. QPSD has some similarities to the asymmetric travelling salesman problem (ATSP) (Choi, Kim, & Kim, 2003). However, unlike ATSP, rotations of permutations are not equivalent in QPSD, and the processing sequence affects the job completion times. It is, therefore, a more complex problem, and finding an optimal solution requires exploring a larger search space. GREC (Sen & Bagchi, 1996) is a general graph search algorithm that has been used to solve QPSD optimally. Although GREC has been shown to be faster than Depth-First Branch-and-Bound (DFBB) for this problem, it runs out of memory even for moderate-sized problems. To our knowledge, approximate approaches for solving large instances of QPSD have not been reported in the literature so far. In this paper, we present exact and approximate approaches for QPSD. In particular:

- We discuss the characteristics of QPSD. The presence of sequence-dependent setup times makes conventional graph-search algorithms like A* (Hart, Nilsson, & Raphael, 1968) and dynamic programming approaches (French, 1982) inapplicable.
- We describe QPSD_MDFBB, an application to QPSD of a Memory-Based Depth-First Branch-and-Bound (MDFBB) approach proposed by Mondal and Sen (2001). QPSD_MDFBB optimally solves QPSD instances up to 30 jobs.
- In order to propose approximate approaches for solving large instances, we first present a greedy approach for QPSD and describe the concept of greedy stochastic search (Viswanathan, Sen, & Chakraborty, 2011). We present an effective greedy stochastic algorithm, QPSD_GSA that can generate moderately good solutions rapidly even for large instances. The general idea of GSA is applicable to a wide range of combinatorial optimization problems like the travelling salesman, knapsack, combinatorial auction and other problems for which greedy solutions can be conceived.
- The major contribution of this paper is to combine the findings of QPSD_GSA with a new genetic algorithm formulation called QPSD_GEN for the problem. QPSD_GEN does not guarantee optimal solutions, but has been seen to generate near-optimal solutions in general. Our findings indicate that QPSD_GEN can be used to generate solutions within 2.2% of the optimal solutions for 100-job problem instances. We have used QPSD_GSA to generate a subset of the initial population for the QPSD_GEN, and found this to have a great impact on the quality of the solutions obtained. This way of integrating GSA into genetic algorithms is also more generally applicable.

The paper is organized as follows. Section 2 discusses the problem with special emphasis on the complexities introduced by sequence-dependent setup times. The algorithm QPSD_MDFBB and experimental results for it are presented in Section 3. Section 4 presents QPSD_GSA and experimental results for it. The implementation of the genetic algorithm, QPSD_GEN and the associated experimental results are given in Section 5. Section 6 concludes the paper and suggests areas for further work.

2. Effect of sequence-dependent setup times on QPSD

The QPSD problem may be formulated in IP (Integer Programming). However, such a formulation may not be efficient to solve in practice. Sen and Bagchi (1996) have shown that the search space for job sequencing problems can be modeled as a tree, or as a graph, and algorithms using the graph search space run faster. For the QPSD problem under the tree formulation, two nodes with the same set of jobs but in different orders and having the same last job will generally not have the same cost because the setup times for the jobs could differ. Nevertheless, the sub-trees below them are identical in terms of the structure. Algorithms using the tree search space cannot take advantage of this fact and might wastefully traverse these identical sub-trees more than once. The graph search space has far fewer nodes and offers the potential for faster search. The node count reduction results from the fact that unlike in the tree search space, there could be multiple paths from the root node to any given node, and this helps avoid replicating the identical sub-trees. However, sequence-dependent setup times complicate traditional graph search because the identical sub-trees may not have the same costs.

The main feature of graph search algorithms like the graph version of A* (Hart et al., 1968) is that when these reach the same node through different paths, they retain the path having the lowest cost, discarding any other paths from the root to the node. This approach works fine when the incremental cost from a given node to a goal node is independent of the path by which the node was reached. This is the same as the principle of optimality on which the dynamic programming formulations are based (French, 1982). However, this does not hold for sequence-dependent setup times. For example, consider the following 4-job problem given in Table 1 (Viswanathan et al., 2011).

In this example, it is assumed that the setup time for a job is zero if it is the first in the sequence. Consider the ordered sequence of jobs (1, 2, 3) and (2, 1, 3). Under the graph formulation, a node is represented by the set of completed jobs without regard to the ordering, except for the last job in the sequence. Because the set of jobs and the last job in the two ordered sequences in question are the same, the two are represented by a single node ({1, 2}, 3), where the first two jobs form a set (unordered) and the last job is shown separately. The cost when the node is reached through the sequence 1, 2, 3 is 182 and through the sequence 2, 1, 3 is 188. If a traditional graph search algorithm reaches the node through the two different paths considered, it would simply discard the higher cost path 2, 1, 3. However, if we look below this node, we see that the sequence 1, 2, 3, 4 has a cost of 1206, which is higher than the cost of the sequence 2, 1, 3, 4 which is 1088. A traditional graph search algorithm thus runs the risk of missing the optimal solution.

Mondal and Sen (2001) have proposed the MDFBB search algorithm which is a memory constrained graph search algorithm. MDFBB is unaffected by situations when the costs of paths from a node to a goal node depend on the path through which the node was reached, as occurs in the presence of sequence-dependent setup times. It has been proven to provide optimal solutions and can therefore be applied to QPSD to take advantage of the graph formulation and yet guarantee optimal solutions. The next section describes the QPSD_MDFBB algorithm. The algorithm works on the graph formulation.

3. QPSD_MDFBB algorithm and empirical results

Algorithm QPSD_MDFBB is shown in Fig. 1. It uses the recursive function QPSD_REC. Unlike DFBB which stores only nodes on the current path and does not exploit the available additional memory on today's computers, QPSD_MDFBB uses a fixed predetermined

Table 1
4-Job QPSD problem.

Job	Setup times				Proc. times	Penalty coeff.
	1	2	3	4		
1	–	1	1	3	1	2
2	1	–	3	2	4	1
3	5	4	–	10	3	1
4	3	6	9	–	10	1