Applied Mathematics and Computation 260 (2015) 188–203

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times

Tamer F. Abdelmaguid*

Department of Mechanical Design and Production, Faculty of Engineering, Cairo University, Giza 12613, Egypt

ARTICLE INFO

Keywords:
Flexible job shop scheduling
Sequence-dependent setup time
Neighborhood search
Tabu search
Algorithm design

ABSTRACT

This paper addresses the makespan minimization problem in scheduling flexible job shops whenever there exist separable sequence-dependent setup times. An extension to the neighborhood search functions of Mastrolilli and Gambradella, developed for the flexible job shop scheduling problem (FJSP), is provided. It is shown that under certain conditions such an extension is viable. Accordingly, a randomized neighborhood search function is introduced, and its best search parameters are determined experimentally using modified FJSP benchmark instances. A tabu search approach utilizing the proposed neighborhood search function is then developed, and experimentations are conducted using the modified instances to benchmark it against a lower bound. Experimental results show that on average, the tabu search approach is capable of achieving optimality gaps of below 10% for instances with low average setup time to processing time ratios.

1. Introduction

The general job shop scheduling problem (JSP) was introduced to deal with the sequencing decisions of manufacturing operations in low volume-high variety manufacturing systems in which part routes are fixed while differ considerably from one part to the other. With the advent of flexible manufacturing systems (FMSs), it became necessary to relax the assumption of fixed part routes in order to be able to deal with the arising scheduling problems in these systems. FMSs are characterized by the existence of general purpose computer numerically controlled (CNC) machines that can perform various types of manufacturing operations. This made it possible for a manufacturing operation of a given part to be conducted on any of a set of different machines with possibly different processing times. This means that the operations' routing became a decision variable instead of being fixed. In the 1980s, the flexible job shop scheduling problem (FJSP) was introduced to address this situation.

The FJSP, denoted $FJc|rcrc|$ according to the three-field notation in [1], is composed of two interacting problems. The first is concerned with selecting a machine for each operation, and the second is a regular JSP with possible recirculation, i.e. a given part may visit the same machine more than once. As indicated in [2], there are two approaches that are followed in dealing with this problem. The first is a hierarchical or a decomposition approach which solves the machine assignment problem first; and based on its solution, a JSP is formulated and solved. The second is a concurrent one which looks for a solution to both problems simultaneously. Early studies of the FJSP developed both hierarchical and concurrent approaches that are based on mixed integer linear programming (MILP) models and dispatching rules [3–6]. In the early 1990s, many researchers found the FJSP fertile for emerging metaheuristic approaches since exact algorithms are computationally prohibitive and the capabilities of traditional

* Tel.: +20 106 2689333; fax: +20 2 35693025.
E-mail address: tabdelmaguid@alumni.usc.edu, tabdelmaguid@eng.cu.edu.eg

http://dx.doi.org/10.1016/j.amc.2015.03.059
0096-3003/

heuristic approaches are limited. The metaheuristics that are frequently applied to the FJSP are tabu search (TS) and genetic algorithm (GA).

In the TS literature, a hierarchical TS approach for $FJc|rcrc|C_{\max}$ and $FJc|rcrc|\Sigma w_j T_j$ was developed in [2]. Simple neighborhood search rules adopted from the JSP literature are used for the scheduling part. Later, more sophisticated neighborhood search strategies were developed in [7] and used within a TS approach to solve the problem $FJc|rcrc|C_{\max}$ in which an operation's processing time is not machine dependent. Dauzère-pérès and Paulli [8] developed another neighborhood search structure that allows either changing the position of an operation in the processing sequence of its already assigned machine or placing it in the sequence of another machine. They used TS for solving the problem $FJc|rcrc|C_{\max}$ with their neighborhood search structure. Their comparison of results with [7] shows significant improvement. Later, Mastrolilli and Gambradella [9] developed two new neighborhood search functions for the same problem and were capable of achieving better results using TS. Up till now, the TS approach of Mastrolilli and Gambradella [9] remains a competitive method for the problem $FJc|rcrc|C_{\max}$.

In the GA literature, Mesghouni et al. [10] are the first to use genetic (evolutionary) algorithm for solving the problem $FJc|rcrc|C_{\max}$. Different GA representations were then developed in [11–14]. Later, Gao et al. [15] developed a hybrid GA which integrates GA with local neighborhood search structures similar to that of Mastrolilli and Gambradella [9]. Their computational results indicate that their GA implementation is comparable to the best known TS implementation by Mastrolilli and Gambradella [9]. Pezzella et al. [16] used the same GA coding in [13] with two additional methods for solving the machine assignment subproblem. However, the GA of Gao et al. [15] demonstrates better results on standard benchmark instances. Wang et al. [17] presented a similar GA coding with immune and entropy principle for solving a multi-objective FJSP. Recently, Zhang et al. [18] proposed an improved GA representation which reduces the computational time and produces competitive results with new upper bounds for some standard benchmark instances. They demonstrated the importance of utilizing an efficient initialization method for a better performance of the GA.

Recently, new metaheuristic techniques were applied to the $FJc|rcrc|C_{\max}$ problem. The parallel variable neighborhood search technique in [19] utilizes the solution representation in [13] and implements six different neighborhood search operations within a parallel variable neighborhood search structure. Better results are achieved compared to the GA technique of Pezzella et al. [16]; however their results are inferior to those of Gao et al. [15]. Hmida et al. [20] utilized a discrepancy search technique with extending the neighborhood search functions in [7]. Their results on standard benchmark instances show that their technique provides competitive results. Bożejko et al. [21] presented hybrid metaheuristics working on two levels. The first level works on the machine selection decisions; while the second level solves the operations scheduling part of the problem. The second level uses tabu search in a parallel fashion that allows for the utilization of several processors simultaneously. They were capable of achieving new best solutions on some benchmark instances. Recently, Yuan and Xu [22] developed a hybrid differential evolution algorithm for the $FJc|rcrc|C_{\max}$ problem, and reported competitive results on standard benchmark instances. They utilized the neighborhood search of Mastrolilli and Gambradella [9] for improving solutions at each iteration.

One of the important factors being addressed in the scheduling literature is the setup time needed to get a machine ready for the processing of a given part. Setup times are classified into separable and inseparable. Separable setups are conducted on the machine without the requirement of having the unprocessed part engaged. On the other hand, the unprocessed part must be available during inseparable setups. The time needed by the inseparable setup can be easily added to the processing time and the resultant summation can then be treated as the processing time in a traditional way. In many situations, separable setup times depend on the sequence by which parts are processed on a machine. Such sequence dependent-setup times are apparent in FMSs as they are related to tool changes between different parts. Recent literature reviews on the different types of scheduling and lot sizing problems with sequence-dependent setup times are provided in [23,24].

This paper is concerned with the FJSP with separable sequence-dependent setup times (FJSP-SDST) with a single objective of minimizing the makespan, denoted $FJc|rcrc, s_{ijk}|C_{\max}$. As a generalization of the FJSP, the studied problem is known to be NP-hard which necessitates the development of efficient solution techniques. Few papers have addressed the FJSP-SDST. To the best of our knowledge, Choi and Choi [25] are the first to address this problem with the objective of minimizing the makespan. They provided an MILP model and developed a local search algorithm that can incorporate different dispatching rules. However, their experimental results showed that a tabu search algorithm adapted from the JSP literature demonstrates better performance in most of the test instances. Guimarães and Fernandes [26] developed a GA approach for the problem $FJc|rcrc, s_{ijk}|$ considering four different objectives simultaneously including the minimization of the makespan. Minor experiments were conducted using four instances from the FJSP literature in [13] with a modification that adds setup times.

Saidi-Mehrabad and Fattahi [27] developed a TS approach for the problem $FJc|rcrc, s_{ijk}|C_{\max}$ with a special structure that exactly specifies two machines for each operation. They compared the TS results for specially designed small size instances (up to 3 jobs and 3 machines) with the optimal solutions obtained by a commercial MILP solver. Their TS was capable of achieving optimal solutions for the small size instances; while, there is no clue about its relative performance with larger size instances. Defersha and Chen [28] developed a parallel GA approach for solving a variant of the FJSP-SDST in which jobs take the form of batches and a job can be moved to a next machine whenever a pre-specified number of parts are completed on the current machine. They used the GA coding of Kacem et al. [13], and they conducted minor experiments to show the capability of using parallel search techniques in escaping local minima, especially with large-sized problem instances, when compared to a single-population GA.

Unlike the FJSP, the studied problem is relatively newly addressed in the literature and there is no standard set of benchmark instances that can be used to compare the different solution approaches. Moreover, there is an apparent inconsistency between the different contributions cited in the previous two paragraphs in terms of the problem structure and the experimentation