Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 66 (2018) 127–134 www.elsevier.com/locate/endm

Using a variable neighborhood search to solve a bi-objective identical parallel machine scheduling problem

Leandro Resende Mundim [a,1] Thiago Alves de Queiroz [b,2]

[a] *Institute of Mathematical and Computer Sciences, University of São Paulo, Av. Trabalhador São-carlense, 400, 13566-590, São Carlos - SP, Brazil*

[b] *Institute of Mathematics and Technology, Federal University of Goiás - Catalão, Av. Dr. Lamartine P. de Avelar, 1120, 75704-020, Catalão - GO, Brazil*

Abstract

We developed a variable neighborhood search heuristic and a mixed integer programming model for the identical parallel machine scheduling problem with sequence dependent setup time. For this problem, we consider minimizing two objectives, which are the makespan and the flow time. The heuristic proposed has a constructive procedure to build initial solutions, five neighborhood structures, and a local search based on the variable neighborhood descent. Computational experiments indicate that the heuristic is very fast and can return better solutions than the model since it found 90% of the best solutions. It also outperformed all solutions computed with the longest processing time and the shortest processing time rules, both commonly adopted for scheduling problems.

Keywords: variable neighborhood search, parallel machines, makespan, flow time.

[1] Email: mundim@icmc.usp.br

[2] Email: taq@ufg.br

https://doi.org/10.1016/j.endm.2018.03.017

1571-0653/© 2018 Elsevier B.V. All rights reserved.

1 Introduction

Determining the best sequence to schedule n jobs, each job with a given processing time and sequence dependent setup time, on m identical parallel machines, is a practical problem which has applications in manufacturing, air traffic control, project management, among others. This problem is NP-hard if we are interested in minimizing the total time when scheduling all jobs [11]. An extensive survey of scheduling problems involving setup times and costs was presented in [1].

In the three-field notation, the identical parallel machine scheduling problem is denoted as $Pm|S_{ij}|C_{max}$ if minimizing the makespan (the completion time of the last job). On the other hand, it is denoted as $Pm|S_{ij}|\sum C_i$ if minimizing the total flow time (the sum of the completion time of all jobs). In this paper, we consider both objectives simultaneously, such that the notation $Pm|S_{ij}|C_{max}, \sum C_i$ is adopted. By considering these two objectives together, there is a reduction in the time in which the shop should operate, in addition to making better use of available resources, then there will be a reduction in the production costs [9].

Regarding the minimization of the flow time, Gupta and Ho [6] proposed an optimization algorithm considering two identical machines when knowing the optimal makespan value in advance. Under a hierarchical approach, Gupta and Ruiz-Torres [5] considered the minimization of the makespan but with the optimal flow time known for which they proposed a heuristic combining a list-scheduling procedure with a backtracking algorithm. A multi start simulated annealing algorithm for a permutation flow shop problem with the minimization of the makespan and the total flow time was proposed in [9]. It is important to mention that there are few studies involving parallel machine problems solved with variable neighborhood search based heuristics [2,3,4].

2 Mathematical Model

A mixed integer programming (MIP) model for the $Pm|S_{ij}|C_{max}, \sum C_i$ is presented next. We consider the minimization of the makespan and the flow time together in which a factor α is used to weigh one in relation to the other, forming a linear combination. The parameters used in the model are: J, the set of $n+1$ jobs in which "0" is a fictitious job; M, the set of m identical machines; p_i, the processing time of job i, for all $i \in J$, except for $p_0 = 0$; and, s_{ij}, the setup time when processing job j immediately after job i, for all $i, j \in J$. If $i = j$ or conversely $i \neq j$ but $i = 0$ ($j = 0$), then the setup

Download English Version:

https://daneshyari.com/en/article/8903358

Download Persian Version:

https://daneshyari.com/article/8903358

Daneshyari.com