Omega 64 (2016) 115–125

Contents lists available at ScienceDirect

Omega

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

Optimization of makespan for no-wait flowshop scheduling problems using efficient matheuristics ☆

Shih-Wei Lin [a,b], Kuo-Ching Ying [c,*]

[a] Department of Information Management, Chang Gung University, Taoyuan, Taiwan
[b] Department of Medical Research and Development, Linkou Chang Gung Memorial Hospital, Taoyuan, Taiwan
[c] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 14 July 2015
Accepted 5 December 2015
Available online 13 December 2015

Keywords:
Scheduling
No-wait flowshop
Makespan
Matheuristics

ABSTRACT

The no-wait flowshop scheduling problem (NWFSP) with makespan minimization is a well-known strongly *NP*-hard problem with applications in various industries. This study formulates this problem as an asymmetric traveling salesman problem, and proposes two matheuristics to solve it. The performance of each of the proposed matheuristics is compared with those of the best existing algorithms on 21 benchmark instances of Reeves and 120 benchmark instances of Taillard. Computational results show that the presented matheuristics outperform all existing algorithms. In particular, all tested instances of the problem, including a subset of 500-job and 20-machine test instances, are solved to optimality in an acceptable computational time. Moreover, the proposed matheuristics can solve very hard and large NWFSPs to optimality, including the benchmark instances of Vallada et al. and a set of 2000-job and 20-machine problems. Accordingly, this study provides a feasible means of solving the *NP*-hard NWFSP completely and effectively.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The flowshop scheduling problem (FSP) has been one of the most intensively discussed classes of problems in operations research literature over the past five decades [1–5]. Of particular practical interest is variants of FSPs, called no-wait FSPs (NWFSPs), that are widely applied in various industries, such as the chemicals, plastics, metals, electronics, pharmaceuticals, and food-processing industries [6,7]. For technological reasons, in these industries, no in-process waiting is allowed between any two consecutive machines, such that once the processing of a job begins, subsequent processing must be continuously carried out on all machines with no interruption until completion. This paper focuses on the NWFSP with the objective of minimizing the makespan, which can be written as $F_m|nwt|C_{\max}$ using the standard 3-tuple notation of Graham et al. [8], where F_m is a flowshop with m machines, *nwt* denotes the no-wait restriction and $C_{\max}$ indicates that the objective is to minimize the makespan. This problem is a member of the set of strongly *NP*-hard problems for three or more machines [9].

In view of the significance of the $F_m|nwt|C_{\max}$ problem in both theory and engineering applications, effective and efficient algorithms for solving it are required. Gilmore and Gomory solved the two-machine case of the $F_m|nwt|C_{\max}$ problem using an $O(n \log n)$ time algorithm with a sub-tour patching technique [10]. Reddi and Ramamoorthy [11] and Wismer [12] were the first to address the $F_m|nwt|C_{\max}$ problem with three or more machines. Many researchers have since attempted to develop effective and efficient algorithms for solving this problem. An early comprehensive survey of the $F_m|nwt|C_{\max}$ problem can be found in Hall and Sriskandarajah [13].

With respect to exact methods, Selen and Hott [14] presented a mixed integer goal programming model for solving the multi-objective NWFSP. Van der Veen and Van Dal [15] have proven that some special cases of NWFSPs are solvable using polynomial time solution algorithms if the processing times on all but two machines are fixed. To the best of our knowledge, no exact method has yet been proposed for solving the $F_m|nwt|C_{\max}$ problem. Given the *NP*-nature of this problem, all previous studies of this topic have focused on developing heuristic algorithms in order to find good (although not necessarily optimal) solutions to this intractable problem in a relatively short time.

The heuristic algorithms that are available for solving the $F_m|nwt|C_{\max}$ problem can be classified into two main categories: constructive heuristics and meta-heuristics. Table 1 summarizes the various constructive heuristics and meta-heuristics in

☆This manuscript was processed by Associate Editor Sterna.

* Corresponding author. Tel.: +886 2 2771 2171; fax: +886 2 2731 7168.
E-mail address: kcying@ntut.edu.tw (K.-C. Ying).

http://dx.doi.org/10.1016/j.omega.2015.12.002
0305-0483/© 2015 Elsevier Ltd. All rights reserved.

Table 1
Constructive heuristics and meta-heuristics for the $F_m|nwt|C_{max}$ problem.

Year	Author(s)	Acronym	Type[a]	Performance be superior to
1976	Bonney and Gundry [16]	S/M	C	Palmer's method, Gupta's algorithm
1980	King and Spachis [17]	LBJD(sc), LBJD(sc)[a]), LBJD(mc), MLSS(mc), MCL(mc)	C	MCL(mc) is better than other four compared heuristics
1993	Gangadharan and Rajendran [18]	GAN-RAJ	C	S/M, SC, MC
1994	Rajendran [19]	RAJ	C	S/M, SC, MC
1995	Gonzalez et al. [22]	GA	M	Palmer's method, Gupta's algorithm, RAJ
2003	Aldowaisan and Allahverdi [23]	SA, SA-1, SA-2, GEN, GEN-1, GEN-2	M	GAN-RAJ, RAJ
2005	Grabowski and Pempera [26]	DS, DS+M, TS, TS+M, TS+MP	M	RAJ, VNS, GASA
2006	Schuster [27]	FTS	M	GASA
2007	Liu et al. [28]	HPSO	M	VNS, GASA
2008	Pan et al. [29]	DPSO	M	HPSO, RAJ, VNS, GASA
2008	Pan et al. [30]	HDPSO	M	HPSO, DPSO
2008	Pan et al. [31]	IIG	M	RAJ, TS, TS+M, TS+MP, DPSO
2008	Li et al. [20]	CH	C	GAN-RAJ, RAJ
2009	Laha and Chakraborty [21]	LC	C	GAN-RAJ, RAJ and two compared heuristics
2009	Qian et al. [32]	HDE	M	HPSO
2010	Tseng and Lin [33]	HGA	M	RAJ, VNS, GASA, TS, HPSO
2011	Jarboui et al. [34]	GA-VNS	M	SA, TS, VNS, DPSO
2012	Samarghandi and ElMekkawy [35]	TS-PSO	M	VNS, GASA, DS, DS+M, TS, TS+M, TS+MP
2013	Davendra et al. [36]	DSOMA	M	DPSO
2015	Ding et al. [37]	TMIIG	M	DPSO, IIG, HDE, HGA, GA-VNS, TS-PSO

[a] C: constructive heuristic, M: meta-heuristic.

chronological order by publication. Several noteworthy constructive heuristics have been proposed for solving the $F_m|nwt|C_{max}$ problem. Bonney and Gundry [16] and King and Spachis [17] pioneered constructive heuristic algorithms to solve the $F_m|nwt|C_{max}$ problem. In 1976, Bonney and Gundry [16] developed a slope matching (S/M) method which used geometrical relationships between the cumulative process times. In the same year, King and Spachis [17] proposed two single-chain heuristics (LBJD(sc) and LBJD(sc)*) and three multiple-chain heuristics (LBJD(mc), MLSS (mc) and MCL(mc)), to solve the $F_m|nwt|C_{max}$ problem. Their computational results revealed that the overall performance of the MCL(mc) heuristic to be excellent. Gangadharan and Rajendran [18] and Rajendran [19] presented additional heuristics, named GAN-RAJ and RAJ, for solving the same problem; their heuristics were shown to be superior to the S/M [16], SC and MC heuristics [17]. Li et al. [20] introduced a composite heuristic (CH), based on an objective increment method, which outperformed GAN-RAJ [18] and RAJ [19] and had the lowest CPU time of all the algorithms to which it is compared. Laha and Chakraborty [21] presented a constructive heuristic, called the LC heuristic, for solving the $F_m|nwt|C_{max}$ problem, based on the principle of job insertion. The empirical results demonstrated that the solutions found using the LC heuristic were significantly better than those using the GAN-RAJ [18], RAJ [19] and two other compared heuristics. To the best of the authors' knowledge, the LC heuristic is currently the state-of-the-art constructive heuristic for solving the $F_m|nwt|C_{max}$ problem.

Some remarkable meta-heuristics have been developed for solving the $F_m|nwt|C_{max}$ problem. Gonzalez et al. [22] developed a hybrid genetic algorithm (GA) for solving the $F_m|nwt|C_{max}$ problem; it produced comparable or better solutions to benchmark problems than known heuristic algorithms. Aldowaisan and Allahverdi [23] proposed six meta-heuristics (SA, SA-1, SA-2, GEN, GEN-1, GEN-2) based on simulated annealing (SA) and GA to solve the problem. Their computational results showed the best two of the six algorithms to be SA-2 and GEN-2, which outperformed GAN-RAJ [18] and RAJ [19], but required significantly more processing time.

In the same year that Aldowaisan and Allahverdi [23] proposed their six meta-heuristics, Schuster and Framinan [24] provided two algorithms, including a variable neighborhood search (VNS) algorithm and a hybrid algorithm that used both SA and GA (GASA) for solving the no-wait jobshop scheduling problem. The authors showed that the VNS and GASA algorithms were superior to the RAJ [19], even though they were not specifically designed for solving the $F_m|nwt|C_{max}$ problem. Framinan and Schuster [25] improved upon the results of Schuster and Framinan [24] in the jobshop case, using a meta-heuristic called complete local search with memory (CLM).

Subsequently, more meta-heuristics have been developed for solving the $F_m|nwt|C_{max}$ problem. Grabowski and Pempera [26] developed and compared two variants of descending search (DS, DS+M) and three Tabu search (TS)-based algorithms (TS, TS+M, TS+MP) which were more effective in finding high quality solutions than all other previous methods, including RAJ [19], VNS [24] and GASA [24]. Schuster [27] implemented a fast Tabu search (FTS) algorithm for solving the no-wait jobshop scheduling problem and the $F_m|nwt|C_{max}$ problem, and find that it compared extremely well to the GASA algorithm [24]. Liu et al. [28] presented an effective hybrid particle swarm optimization (HPSO) for solving the problem. Their comparisons of HPSO with other algorithms, such as the VNS [24] and GASA [24] algorithms, demonstrated the effectiveness of the HPSO algorithm. Pan et al. conducted a series of studies and proposed a discrete particle swarm optimization (DPSO) algorithm [29], a hybrid discrete particle swarm optimization (HDPSO) algorithm [30] and an improved iterated greedy (IIG) algorithm [31]. Their computational results showed that these meta-heuristic algorithms to be superior to several of the best heuristics reported in the literature, in terms of quality of the search, robustness and efficiency. Qian et al. [32] proposed an effective hybrid differential evolution (HDE) algorithm for solving the same problem; the simulation results demonstrated that it was superior to the HPSO algorithm [28]. Tseng and Lin [33] proposed a hybrid genetic algorithm (HGA), which hybridized the genetic algorithm and a novel local search scheme. Their computational results, based on two well-known benchmarks, showed that the proposed HGA yielded better results than those obtained using the RAJ [19], VNS [24], GASA [24], TS [26] and HPSO [28] algorithms. Jarboui et al. [34] propose a hybrid genetic algorithm (GA-VNS) that applied VNS as an improvement procedure in the final step of the genetic algorithm. Their computational results show that GA-VNS provided competitive results and better upper bounds (UBs), while the VNS algorithm [24] was better than the GA-VNS for large test instances. Samarghandi and ElMekkawy [35] developed a

Download English Version:

https://daneshyari.com/en/article/1032358

Download Persian Version:

https://daneshyari.com/article/1032358

Daneshyari.com