Computers & Industrial Engineering 124 (2018) 276–292

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Multi objective two-stage assembly flow shop with release time

Shaya Sheikh[a,*], G.M. Komaki[b], Vahid Kayvanfar[c]

[a] School of Management, New York Institute of Technology, 1855 Broadway, New York, NY 10023, United States
[b] Department of Marketing and Business Analytics, College of Business, Texas A&M University-Commerce, 2200 Campbell St, Commerce, TX 75428, United States
[c] Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Ave., 15875-4413 Tehran, Iran

ARTICLE INFO

Keywords:
NSGA-III
MOPSO
Assembly flow shop
Multi-objectives

ABSTRACT

Two-stage assembly flow shops are integral part of several manufacturing systems such as computer and engine manufacturing lines. This paper explores three objectives of makespan, total tardiness, and total completion times for two-stage assembly flow shop with release time. To the best of our knowledge, these performance measures have not been addressed simultaneously in assembly flow shops. We derive polynomial optimal solutions for special cases of this problem with a single objective and then develop heuristics with promising starting solutions for the multi-objective case. Due to NP-hardness of the problem, we apply a customized reference-based Non-dominated Sorting Genetic Algorithm (NSGA-III) and Multi-Objective Particle Swarm Optimization (MOPSO) as solution procedures. Finally, we present extensive computational analysis to compare the performance of employed heuristic and metaheuristics on randomly generated instances. Results show that both NSGA-III and MOPSO generate competitive solutions for the presented problem. However, NSGA-III generates significantly better results than MOPSO based on one of the three performance metrics.

1. Introduction

Two-stage assembly flow shop, AF2, is a classical model where components are processed on parallel machines in the first stage and then these components are assembled on one assembly machine in the second stage. AF2 has diverse applications in manufacturing and service industry. Among several applications of AF2, we can refer to engine assembly plant by Lee, Cheng, and Lin (1993), computer manufacturing by Potts, Sevast'janov, Strusevich, Van Wassenhove, and Zwaneveld (1995), queries scheduling on distributed data by Allahverdi and Al-Anzi (2006), food and fertilizer production by Hwang and Lin (2012), and paint manufacturing by Chen, Huang, Luo, and Wang (2015).

Except one publication by the author of this article, Komaki and Kayvanfar (2015), all publications in the literature assume that components are available for production at the beginning of the process. Obviously, this assumption may not hold in practice all the time and it would be more realistic to assume non-negative release times for components. Considering components' release times in AF2 model is not only important but necessary in many practical cases. We also consider three simultaneous objectives of total completion time, makespan, and total tardiness. The goal is to find the sequence of jobs that minimizes these objectives. To the best of our knowledge, all these specifications have not been considered altogether in previous researches. According to the triple notation adopted by Graham, Lawler, Lenstra, and Rinnooy Kan (1977), our proposed problem is identified as $AF(m,1)|r_j|(C_{\max}, TT, TC)$.

Assembly flow shop with single objective of makespan or completion time or tardiness is known to be NP-hard in strong sense (Koulamas, 1994; Lee et al., 1993; Tozkapan, Kirca, & Chung, 2003). Therefore, multi-objective optimization of this problem is also NP-hard in strong sense. Prior to the rise of metaheuristic methods, a handful of branch and bound algorithms was offered for AF2 problems, see Lee et al. (1993) and Hariri and Potts (1997). Due to NP-hardness of this problem and computational burden of branch and bound algorithm, special efforts have been taken to find efficient solutions using metaheuristic methods. Algorithms such as Particle Swarm Optimization and Tabu Search by Allahverdi and Al-Anzi (2006) and Allahverdi and Aydilek (2015), Grey Wolf by Komaki and Kayvanfar (2015), and Artificial Immune System by Komaki, Teymourian, and Kayvanfar (2016) are among those.

This paper is organized as follows. A brief review of multi-objective AF models and their proposed solutions are presented in Section 2. Problem definition, assumptions, and special cases are explained in Section 3. Section 4 discusses the proposed lower bounds, heuristic, and metaheuristics. Section 5 contains the experimental results for proposed

* Corresponding author.
E-mail addresses: Ssheik11@nyit.edu (S. Sheikh), mohamed.komaki@tamuc.edu (G.M. Komaki), v.kayvanfar@aut.ac.ir (V. Kayvanfar).

https://doi.org/10.1016/j.cie.2018.07.023
Received 20 March 2018; Received in revised form 13 June 2018; Accepted 14 July 2018
Available online 20 July 2018
0360-8352/

heuristic and metaheuristic algorithms where we evaluate the performance of NSGA-III, MOPSO and other benchmarks using randomly generated instances and Section 6 concludes this research.

2. Literature review

Multi-objective assembly flow shop problem has been addressed by several researchers in recent years. However, majority of these researches consider two objectives of makespan and total completion time as the most frequently used performance measures. For instance, Allahverdi and Al-Anzi (2008) applied simulated annealing (SA), Ant colony optimization, and self-adaptive differential evolution to solve AF $(m,1)||(C_{\max}, TC)$ with SA outperforming other algorithms. Later, variable neighborhood search by Javadian et al. (2009), cloud theory based simulated annealing by Torabzadeh and Zandieh (2010), and imperialist competitive algorithm (ICA) by Shokrollahpour, Zandieh, and Dorri (2011) generated superior results than SA. Al-Anzi and Allahverdi (2009) presented tabu search (TS), particle swarm optimization (PSO), and self-adaptive differential evolution for AF$(m, 1)||(C_{\max}, L_{\max})$ where PSO outperformed the others. Azadeh, Jeihoonian, Shoja, and Seyedmahmoudi (2012) proposed a simulation approach coupled with multi-layer neural network to solve AF2 model with set-up time, machine breakdown, and stochastic activity times. Later, Seidgar, Zandieh, and Mahdavi (2016), Seidgar, Zandieh, Fazlollahtabar, and Mahdavi (2016) and Tian, Liu, Yuan, and Wang (2013) proposed NSGA and discrete particle swarm optimization for the same problem. Seyedi and Maleki-Daronkolaei (2013) investigated AF $(m,1)||(TE, TT)$ and developed variable neighborhood search (VNS), SA, and GA algorithms with VNS outperforming the others. Mozdgir, Fatemi Ghomi, Jolai, and Navaei (2013) developed hybrid variable neighborhood search for AF$(m, n)||(C_{\max}, TC)$ problem with setup costs. Komaki and Kayvanfar (2015) offered grey wolf optimizer algorithm for AF$(m, 1)|\ r_j\ |(C_{\max}, TC)$.

Majazi Dalfard, Ardakani, and Nazalsadat Banihashemi (2011) proposed a hybrid genetic algorithm for AF model with sequence dependent set-up and transportation times and objectives of total weighted squared tardiness, earliness, makespan, and number of tardy jobs. Navaei, Fatemi Ghomi, Jolai, and Mozdgir (2014) proposed SA and ICA for AF(m_1, m_2) considering set-up times and non-identical machines in assembly stage. Authors used holding and delay costs as the performance measures. Yan, Wan, and Xiong (2014) proposed hybrid electromagnetism-like algorithm and VNS for AF$||(C_{\max}, E_{\max}, L_{\max})$. Chen et al. (2015) proposed a bi-objective nonlinear program for AF problem with the objectives of minimizing maximum waiting time and average earliness and tardiness and performance measures of production simultaneity and shipment punctuality. Authors utilized linear weighted sum method to balance the two criteria and proposed modified genetic algorithm as the solution procedure. Wang, Ma, Luo, and Qin (2016) developed a coordinated scheduling system for production and transportation in a two-stage AF with batching as their last stage. Their objective was to decrease the total delivery cost while reducing the average arrival times. Kazemi, Mazdeh, and Rostami (2017) implemented ICA and a hybrid ICA to solve AF problem with minimizing batch delivery and total tardiness times. A nomenclature of abbreviations used in triple notations and a list of multi-objective assembly flow shop papers with their main specifications and solution methods are summarized in Tables 1 and 2, respectively.

Multi-Objective Evolutionary Algorithms (MOEAs) are categorized into methods with no tool for preservation of good solutions (elitism), such as Non-dominated Sorting Genetic Algorithm (NSGA), and methods with the elitism mechanism, such as NSGA-II by Deb, Agrawal, Pratap, and Meyarivan (2000 & 2002). NSGA is a popular evolutionary algorithm with a non-dominated sorting procedure that applies a ranking method with emphasis on superior solutions, see Coello (1999).

Table 1
Nomenclature of abbreviations used in triple notations $\alpha|\beta|\gamma$.

β		γ	
Notation	Explanation	Notation	Explanation
SD	Sequence-dependent set-up	C_{max}	Makespan
SI	time	*TE*	Total (weighted) Earliness
r	Sequence Independent set-	*FDC*	Fixed Delay Costs
NW	up time	L_{max}	Maximum lateness where
B	Non-zero release time	*SC*	$L_j = C_j - d_j$
PM	No wait between	*TT*	Shipping (delivery) costs
ML	subsequent processes	*TU*	Total (weighted) tardiness
LI	Non-zero buffer between	*UAS*	Total number of tardy jobs
	machines	*VDC*	Unavailability of System
	Preventive Maintenance	*VHC*	Variable Delay Costs
	Multiple Lot	*W*	Variable Holding Costs
	Limited Inventory Size	*TC*	Waiting time
		LT	Total (weighted) completion
		FC	times
			Order Lead Time
			Shop Floor Costs

This algorithm maintains the diversity in the population by utilizing a sharing method and exploring different regions in Pareto front. Nevertheless, NSGA's drawbacks such as lack of elitism has limited its usage in recent years. NSGA-II is a modified version of NSGA that utilizes a fast non-dominated sorting genetic algorithm with more computational efficiency and less dependency on sharing parameter for diversity preservation, see Deb et al. (2000) and Deb, Pratap, Agarwal, and Meyarivan (2002). Recently, a reference-point based multi-objective NSGA-II algorithm (called NSGA-III) is proposed by Deb and Jain (2014) with superior performance for problems with more than two objectives. The novelty of NSGA-III is in using the reference points to preserve the diversity of the population. Reference points are either provided by experts or generated by a systematic method such as the one developed by Das and Dennis (1998). Also, NSGA-III has been recently used by some researchers in different areas such as Kayvanfar, Husseini, Karimi, and Sajadieh (2017).

In addition to the evolutionary algorithms, some multi-objective metaheuristic approaches, such as Multi-Objective Particle Swarm Optimization (MOPSO), have also been used to solve the multi-objective optimization problems, see Moore and Chapman (1999). This metaheuristic is inspired from the social behavior of birds within a flock where particle represents each potential solution of the problem and swarm represents the population of solutions. In Particle Swarm Optimization (PSO), each particle searches the solution space based on its current position and velocity direction. Due to efficiency and fast convergence of the PSO in solving single objective problems, it has been extended to solve multi-objective problems. Since the inception of MOPSO by Moore and Chapman in 1999, several versions of MOPSO have been proposed in the literature. MOPSO has been shown to outperform NSGA-II in solving various benchmark problems (Kennedy, Kennedy, Eberhart, & Shi, 2001).

In this study, two metaheuristic algorithms, NSGA-III (Deb and Jain, 2014) and MOPSO (Coello, Pulido, & Lechuga, 2004) to solve the addressed problem in affordable computational time are developed. These algorithms are customized to handle the constraints of the problem. After obtaining the optimal solutions in Pareto frontier utilizing NSGA-III and MOPSO, several metrics are calculated to compare the solutions.

3. Problem definition and special cases

The first stage in AF2 consists of m non-identical parallel machines and the second stage has only one assembly machine. There exist n jobs, each is made of m components and $m + 1$ operations where the first m operations are processed in the first stage and the assembly operation in the second stage. We assume that preemption is not allowed on any