ARTICLE IN PRESS
JID: KNOSYS [m5G;September 23, 2017;12:54]
Knowledge-Based Systems 000 (2017) 1–19
Contents lists available at ScienceDirect
Knowledge-Based Systems
journal homepage: www.elsevier.com/locate/knosys

Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms

Weishi Shao[a], Dechang Pi[a,b,1,*], Zhongshi Shao[a]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[b] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China

ARTICLE INFO

Article history:
Received 16 March 2017
Revised 25 July 2017
Accepted 18 September 2017
Available online xxx

Keywords:
Distributed no-wait flow shop scheduling problem
Makespan
Iterated greedy algorithm
Neighborhood structures
Local search

ABSTRACT

The distributed production lines widely exist in modern supply chains and manufacturing systems. This paper aims to address the distributed no-wait flow shop scheduling problem (DNWFSP) with the makespan criterion by using proposed iterated greedy (IG) algorithms. Firstly, several speed-up methods based on the problem properties of DNWFSP are investigated to reduce the evaluation time of neighborhood with $O(1)$ complexity. Secondly, an improved NEH heuristic is proposed to generate a promising initial solution, where the iteration step of the insertion step of NEH is applied to the factory after inserting a new job. Thirdly, four neighborhood structures (i.e. *Critical_swap_single, Critical_insert_single, Critical_swap_multi, Critical_insert_multi*) based on factory assignment and job sequence adjustment are employed to escape from local optima. Fourthly, four local search methods based on neighborhood moves are proposed to enhance local searching ability, which contains *LS_insert_critical_factory*1, *LS_insert_critical_factory*2, *LS_swap*, and *LS_insert*. Finally, to organize neighborhood moves and local search methods efficiently, we incorporate them into the framework of variable neighborhood search (VNS), variable neighborhood descent (VND) and random neighborhood structure (RNS). Furthermore, three variants of IG algorithms are presented based on the designed VNS, VND and RNS. The parameters of the proposed IG algorithms are tuned through a design of experiments on randomly generated benchmark instances. The effectiveness of the initialize phase and local search methods is shown by numerical comparison, and the comparisons with the recently published algorithms demonstrate the high effectiveness and searching ability of the proposed IG algorithms for solving the DNWFSP. Ultimately, the best solutions of 720 instances from the well-known benchmark set of Naderi and Ruiz for the DNWFSP are proposed.

1. Introduction

Intelligent production system is an important part of an intelligent factory, meanwhile the intelligent production should consider production, scheduling, logistics, and management of all enterprises [1]. Along with the increase of cooperation between enterprises, the distributed production system has widely existed in the manufacturing systems [2–7]. The distributed manufacturing production system leads to high quality production and other benefits such as reduced production costs, decreased management risks and more [8]. In real production, many industrial applications, e.g. metal, plastic, chemical, and pharmaceutical industries, require that jobs cannot have any delay between consecutive machines or operations. For example, in steel factories, the heated metal must be processed through a continuous sequence of operations before it is cooled to prevent defects in the steel. Due to the competitive situation and globalization, the above industrial enterprises have switched the single factories to distributed factories. Therefore, this paper studies the distributed no-wait flow shop scheduling (DNWFSP) with the makespan criterion presented by Lin and Ying [4]. In the classic permutation flow shop scheduling problem (PFSP), n jobs are processed on m machines in the same processing order. Under the distributed manufacturing production system, Naderi and Ruiz [2] proposed a distributed permutation flow shop scheduling problem (DPFSP). In the DPFSP, there exist f identical factories, and each factory has a same permutation flow shop. Each job needs to be first assigned to one of factories, and then this problem must solve f PFSPs to optimize scheduling objectives. Therefore, the task of DPFSP is to reasonably assign jobs to factories and determine the jobs sequence in each factory.

* Corresponding author.
E-mail addresses: shaoweishi@hotmail.com (W. Shao), nuaacs@126.com (D. Pi), shaozhongshi@hotmail.com (Z. Shao).
[1] Present address: No. 169 Sheng Tai West Road, Jiang Ning District, Nanjing Jiangsu Province, People's Republic of China.

https://doi.org/10.1016/j.knosys.2017.09.026
0950-7051/

Besides the constraints of PFSP, the DPFSP also assumes that once a job is assigned to a factory, it should be completed here and no reassignments are possible. The distributed no-wait flow shop scheduling problem (DNWFSP) is an extension of DPFSP [4], which considers the no-wait constraint in the permutation flow shop in each factory. The no-wait refers to that no interruption is permitted either on or between any consecutive machines in an assigned factory in the route. Like the DPFSP, the combinations of factories and jobs are not changed, so the DNWFSP has the same number of solutions with the DPFSP. Therefore, we can conclude that the total number of solutions of DNWFSP is $\binom{n-1}{f-1}n!$ according to literature [2] that demonstrates the DPFSP has $\binom{n-1}{f-1}n!$ solutions totally. The number of solutions of the DNWFSP is significantly greater than those of the regular NWFSP (that has $n!$ solutions). According to the literature [4], if the number of factories equals one, or if all the jobs are assigned to single factory, then the DNWFSP with the makespan criterion reduce to a NWFSP with the makespan criterion. According to literature [9], the NWFSP with makespan criterion is a strongly NP-hard when the numbers of machines is more than two. In addition, each factory in the DNWFSP has a no-wait flow shop. Therefore, we can conclude that DNWFSP is a NP-hard problem with minimization of makespan when $m > 2$ and $n > f$.

Regarding the approaches for solving the DNWFSP, an iterated cocktail greedy (ICG) was first proposed by Lin and Ying [4], which contains two self-tuning mechanisms and a cocktail destruction mechanism. Then, Komaki and MalaKooti [10] proposed a general variable neighborhood search algorithm (GVNS) to solve it. Compared to the DNWFSP, the literature about the DPFSP is relatively rich. Naderi and Ruiz [2] first proposed six mixed integer linear programs models for the DPFSP, and designed two factory assignment rules, and 14 heuristics based on dispatching rules, constructive heuristics, and variable neighborhood decent (VND) methods. After that, many effective and efficient heuristic-based algorithms were developed to deal with the DPFSP. Liu and Gao [11] first proposed an electromagnetism-like mechanism (EM) algorithm which introduces the concept of component force and redefines the movement of election based on insertion and swap operators. The EM contains four neighborhood structures to enhance the search ability, i.e. insertion within the critical factory, swap in the critical factory and general insertion and swap. Gao and Chen [12] proposed a hybrid genetic algorithm with local search (IG_LS), in which the crossover and mutation operators were developed to make it suitable for the representation of DPFSP. A local method integrated Insert_Jobs, Exchange_Jobs, and Move_Jobs was developed to explore neighborhood solutions. After that, Gao et al. [13] presented a knowledge-based genetic algorithm to improve the GA_LS. The authors regarded the partial job sequences as knowledge, and employed the infection operator to improve individuals in the genetic algorithm. Gao and Chen [14] proposed a modified NEH2, i.e. NEH_{df}, which inserts f jobs instead of one job at a time. Meanwhile, they employed a multi-insertion through an unspecified branch and bound procedure. Gao et al. [15] proposed an improved VNS algorithm, which combines VND(a) of [2] with NEH_{df} method presented in [14]. Gao et al. [16] proposed a tabu search (TS) algorithm, in which the neighborhood structure is defined by exchanging sub-sequence between factories rather than by moving a single job to other positions. An enhanced local search method by combing Insert_Jobs, Exchange_Jobs, and Move_Jobs [13] is embed into TS to enhance local searching ability. Lin and Ying [17] proposed a modified iterated greedy (MIG) algorithm for solving the DPFSP. Four variants of MIG algorithms are tested and the best one IG_{VST} is compared against the HGA and TS. In the MIG algorithm, the destruction size is adjusted dynamically. Fernandez-Viagas and Framinan [18] proposed a bounded-search iterated greedy algorithm (BIG) for the DPFSP. The BIG improves a solution in each iteration by three local search methods including LS proposed by Neardri and Ruiz [2], simple relative local search, and relative local search based on exchange. Moreover, the authors defined lower and upper bounds for the makespan when a job is assigned to a specific factory. Wang et al. [19] proposed an estimation of distribution algorithm (EDA), in which the earliest completion factory (ECF) was proposed for a permutation based encoding to generate a feasible schedule. The EDA employs a probability model to describe the solution space, and generates new solutions by sampling the probability model. Xu et al. [20] proposed a hybrid immune algorithm (HIA), in which the ECF rule considering both factory dispatching and job sequence was used to transfer a job permutation to a feasible schedule. The authors designed the crossover operator, mutation, and vaccination operator to perform immune search. Naderi and Ruiz [21] proposed a scatter search which consists of five components, i.e. diversification generation method, improvement method, reference set update, subset generation method, and solution combination method. The authors compared the existing algorithms (EM, BIG, TS, and VNS) and the experimental results showed that the SS has the best performance so far. As regards the NWFSP, there are a significant amount of existing results. A number of algorithms were presented for solving the NWFSP, such as hybrid differential evolution (HDE) algorithm [22], discrete particle swam optimization (DPSO) [23], genetic algorithms (GA) [24,25], teaching-learning based optimization algorithm (TLBO) [26], iterative local search (ILS) algorithm (ILS) [27], iterated greedy (IG) algorithm [28], simulated annealing (SA) algorithm [29], etc.

The iterated greedy (IG) algorithm is a simple and effective algorithm presented by Ruiz and Stützle [30]. The IG has very few parameters and does not adopt specific problem knowledge. The IG starts with a random solution or a problem-specific heuristic solution generated by NEH [31], and then the destruction-construction is applied to this solution, which contains two phases, i.e. removing *ds* jobs and reinserting them into remaining sequence. Finally, the local search is applied to the current solution. During recent years, the IG algorithm has shown to be powerful in solving the complex optimization problems. Of course, the IG also has been applied to other scheduling problems, such as the blocking flow shop scheduling [32], the no-idle flow shop scheduling [33], the SDST-PFSP [34] etc. Motivated by the successfully applications of IG, this paper develops three different variants of iterated greedy algorithms that combine different neighborhood structures to solve the DNWFSP with the makespan criterion. The innovation of this paper is summarized as follows. (1) An improved NEH heuristic is proposed to generate a promising initial solution, where the iteration step of the insertion step of NEH is applied to the factory after inserting a new job. (2) Four local search methods (including *LS_insert_critical_factory*1, *LS_insert_critical_factory*2, *LS_swap, and LS_insert*) based on neighborhood moves are proposed to enhance searching ability. (3) Three different variants of IG algorithms (including IG_VNS, IG_VND, IG_RNS) are presented based on the designed VNS, VND and RNS to solve the DNWFSP. Regarding the contribution of this paper, three different neighborhood moves are firstly defined according to the no-wait constraint and the characteristic of distribution of the DNWFSP, and we develop accelerated techniques motivated by the no-wait constraint for computing the makespan of candidate solutions generated by neighborhood moves. Then, to escape from local optima, this paper designs four neighborhood structures according to the literature [35], which consist of *Critical_swap_single, Critical_insert_single, Critical_swap_multi, Critical_insert_ multi*. These neighborhood moves exist in the critical factory and between the critical factory and other factories. Finally, based on 720 benchmark instances proposed by Naderi and Ruiz [2], numerical simulations and comparisons with the existing algorithms show