Applied Soft Computing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system

Q1 Sana Abdollahpour, Javad Rezaian*

Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

ARTICLE INFO

Q3 *Article history:*
Received 31 October 2013
Received in revised form 8 November 2014
Accepted 21 November 2014
Available online xxx

Keywords:
Flow shop scheduling problem
Limited buffers
Artificial immune system
Iterated greedy algorithm

ABSTRACT

In this study, three new meta-heuristic algorithms artificial immune system (AIS), iterated greedy algorithm (IG) and a hybrid approach of artificial immune system (AIS-IG) are proposed to minimize maximum completion time (makespan) for the permutation flow shop scheduling problem with the limited buffers between consecutive machines. As known, this category of scheduling problem has wide application in the manufacturing and has attracted much attention in academic fields. Different from basic artificial immune systems, the proposed AIS-IG algorithm is combined with destruction and construction phases of iterated greedy algorithm to improve the local search ability. The performances of these three approaches were evaluated over Taillard, Carlier and Reeves benchmark problems. It is shown that the AIS-IG and AIS algorithms not only generate better solutions than all of the well-known meta heuristic approaches but also can maintain their quality for large scale problems.

1. Introduction

Scheduling is one of the most important concerns in operation research. As a typical manufacturing and scheduling problem with strong industrial background, flow shop scheduling has appealed wide attention both in academic and engineering fields [1].

Flow shop scheduling problem consists in scheduling some independent jobs on the some consecutive machines. If the jobs have same sequence on all machines, the problem is called permutation flow shop scheduling problem and if the processing sequence on each machine can be altered, the problem is called non-permutation flow shop.

A special case of flow shop scheduling problem is existence of buffer between consecutive machines, capacity of buffers can be finite or infinite (depend on the size of semi-manufactured). When the capacity of buffers is zero, the problem is called blocking flow shop; in this case, when the next machine is busy the job is blocked on the current machine. But in general cases capacity of buffers are finite. In this situation while next machine is busy, jobs will be pasted to buffers.

For determining the computational complexity of the flow shop with intermediate buffers, Papadimitriou and Kanellakis [2] proved that it is NP-hard even only for two machines. Therefore, in practice, using approximation methods near optimal solutions can be obtained taking reasonable time.

Currently, the approximation algorithms for solving flow shop problems can be classified into two categories: heuristic methods and meta-heuristic methods.

As samples of heuristic methods we can cite Ronconi [3] where the authors presented three constructive heuristics called minmax (MM), combination of MM and NEH (MME), and combination of permutation flow shop and NEH (PFE), for the blocking flow shop problem. Leisten [4] proposed that some priority-based heuristics for both the permutation and general flow shop problems with intermediate buffers.

Recently, extensive computational requirements of flow shop problems have resulted in numerous attempts to develop efficient heuristics, leading to a significant interest in meta-heuristics [5]. As examples, Norman [6] proposed the Tabu Search (TS) algorithm for flow shop scheduling Problems with sequence dependent setup time and finite buffers restrictions. Smutnicki [7] also proposed different TS approach for two machines flow shop environment with limited buffers. Nowicki [8] developed Smutnicki's idea with more than two machines. Later, Brucker et al. [9] applied Nowicki's approach for solving non-permutation flow shop with limited

Q2 * Corresponding author. Tel.: +98 9111571257; fax: +98 22190259.
E-mail address: J_rezaeian@ustmb.ac.ir (J. Rezaian).

http://dx.doi.org/10.1016/j.asoc.2014.11.022
1568-4946/© 2014 Published by Elsevier B.V.

buffers. Recently, Hsieh et al. [10] proposed an artificial immune system approach for the same problem and this algorithm obtained very good solutions.

Many researchers have concluded that hybrid approaches for scheduling problems could end up with high quality results [11–13]. As examples of hybrid meta-heuristics for the proposed problem, Wang et al. [14] developed a hybrid genetic algorithm (HGA) for the general flow shop scheduling problem with limited buffers to minimize the makespan. Liu et al. [15] proposed a hybrid particle swarm optimization (HPSO) algorithm for this problem. Pan et al. [16] applied discrete differential evolution (P-DDE) for flow shop with finite capacity of buffers and after that developed P-DDE algorithm and proposed hybrid discrete differential evolution (HDDE) [1] algorithm for the same problem. Pan et al. [17] used New Harmony Search (CHS) approach for solving intermediate buffers constraints in flow shop problem.

The artificial immune system algorithm (AIS) is one of the newest nature-inspired algorithms which is created based on the pattern taken from the body immune system for recognizing invader factors and eliminating them. In comparison to the other meta-heuristics in the literature, the AIS algorithm has fewer complexity and mathematical requirements and due to discrete nature, can be easily used for solving various kinds of scheduling problems. As samples of AIS algorithm applications in various scheduling problems, Ying [18] studied multistage hybrid flow shop problem and solved it by hybrid AIS. Zandieh et al. [19] also proposed hybrid approach of AIS for hybrid flow shop environment with sequence dependent setup time constraint. Naderi [20] implemented hybrid AIS for Realistic variant of job shops with makespan criteria. Liu, Huang, and Chang [21] applied AIS algorithm for solving grid scheduling problems. Chen, Chang and Lin [22] proposed an AIS algorithm with T-cells and B-cells for solving permutation flow shop problems.

AIS algorithm also has wide use in other science category. For example, Kephart [23] studied on computer viruses and applied a control way for them by artificial immune system.

Among the modern meta-heuristic-based algorithms, the AIS algorithm and simulated annealing (SA) algorithm have emerged as highly effective and efficient algorithmic approaches to NP-hard combinatorial optimization problems [24–26]. The iterated greedy (IG) algorithm is one of the simplest local search methods which in terms of concept and performance is similar to SA algorithm. The IG has two central phases: destruction and construction. The IG starts with an initial solution and generates a new solution by using the destruction and construction steps. The IG has been successfully applied to different variant of scheduling problems. Pan [27] proposed an improved IG algorithm for no-wait flow shop. Ying [5] applied IG with the NEH heuristic approach for non-permutation flow shop. Fanjul-Peyro and Ruiz [28] coined IG algorithm as a local search mechanism and proposed it for unrelated parallel machines scheduling problem.

In this study, three meta-heuristic approaches AIS, IG and a hybrid approach of artificial immune system (AIS-IG) algorithm by combining clonal selection of artificial immune system and destruction and construction phases of iterated greedy algorithm for solving the permutation flow shop scheduling problem with limited buffers are presented. Experimental results and comparisons show that the proposed AIS-IG algorithm is effectiveness approach for solving the flow shop scheduling problem with finite buffers for makespan criterion.

The reminder of this paper is organized as follows. Definition of problem and mathematical model are presented in section 2, section 3 deals with the proposed AIS-IG algorithm. The computational results are provided in Section 4 and the conclusions are presented in Section 5.

2. Problem definition

The permutation flow shop problem considered in this study can be defined as follow. There are n independent jobs ($i = 1 \ldots n$) that should be processed on m machines ($k = 1 \ldots m$) and B_k is intermediate buffer between two consecutive machines. The order of machines is given and the jobs are processed on m machines in the same sequence. Processing time of job i on machine k is shown by p_{ik}. When a particular type of machine is busy, the incoming parts will be moved to next machine, buffer (if the buffer is not full) or will be blocked on the same machine. The following assumptions are considered:

- The setup times are included in the processing times.
- All jobs are available at time zero and there is no precedence constraint.
- At any time, each machine can processes at most one job and each job can be processed only one machine.
- The number of jobs and their processing times on each machine are given.

For describing the mathematical model let a job permutation $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$ represents the sequence of jobs to be processed, and $d_{\pi(i),k}$ be the departure time of operation $o_{\pi(i),k}$ from machine k which will be calculated by following relations proposed by Pan et al. [1]:

$$d_{\pi(1),1} = p_{\pi(1),1} \quad (1)$$

$$d_{\pi(1),k} = p_{\pi(1),k-1} + p_{\pi(1),k} \quad k = 1, \ldots, m \quad (2)$$

$$d_{\pi(i),1} = d_{\pi(i-1),1} + p_{\pi(i),1} \quad i = 1, \ldots, B_1 + 1 \quad (3)$$

$$d_{\pi(i),k} = \max(d_{\pi(i-1),k}, d_{\pi(i),k-1}) + p_{\pi(i),k} \quad i = 2, \ldots, B_1 + 1 \quad k = 2, \ldots, m-1 \quad (4)$$

$$d_{\pi(i),1} = \max(d_{\pi(i-1),1} + p_{\pi(i),k}, d_{\pi(i-B_1-1),2} \quad i > B_1 + 1 \quad (5)$$

$$d_{\pi(i),k} = \max(\max(d_{\pi(i-1),k}, d_{\pi(i),k-1}) + p_{\pi(i),k}, d_{\pi(i-B_k-1),k+1}) \quad i > B_k + 1 \quad k = 2, \ldots, m-1 \quad (6)$$

$$d_{\pi(i),m} = \max(d_{\pi(i-1),m}, d_{\pi(i),m-1}) + p_{\pi(i),m} \quad i = 2, \ldots, m \quad (7)$$

In the above relations, the departure time of the first job on each machine is calculated at first, then the second job, and so on. Eqs. (1) and (2) define the departure time of job $\pi(1)$ through machine 1 to machine m. Eqs. (3) and (4) determine the departure time of job $\pi(i)$ on machine 1 ($i = 2, 3, \ldots, B_1 + 1$) or on machine $k = 2, 3, \ldots, m-1$ ($i = 2, 3, \ldots, B_k + 1$). Eqs. (5) and (6) calculate departure time of job $\pi(i)$ on the last machine ($i > B_1 + 1$) or on machine $k = 2, 3, \ldots, m-1$ ($i > B_k + 1$). In this case, the buffer capacities should be considered. In Eq. (6) $\max(d_{\pi(i-1),k}, d_{\pi(i),k-1}) + p_{\pi(i),k}$ represents the completion time of operation $o_{\pi(i),k}$ and $d_{\pi(i-B_k-1),k+1}$ is associated with limited buffer capacities. Eq. (7) gives the departure time of job $\pi(i)$ ($i = 2, 3 \ldots n$) on the last machine. It can be easily concluded from the above formulations that the buffer constraint makes no effect on makespan and the above problem becomes the classical permutation flow shop scheduling problem when $B_k > n - 1$. In the above equations B_k is defined as an intermediate buffer after machine k, but not as the size of buffer.

Please cite this article in press as: S. Abdollahpour, J. Rezaian, Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.11.022