Computers & Industrial Engineering 88 (2015) 273–283

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

A new hybrid island model genetic algorithm for job shop scheduling problem ☆

Mohamed Kurdi *

Computer Engineering Department, Adnan Menderes University, 09010 Aydin, Turkey

ARTICLE INFO

Article history:
Received 26 January 2015
Received in revised form 12 July 2015
Accepted 13 July 2015
Available online 30 July 2015

Keywords:
Job shop scheduling
Island model genetic algorithm
Tabu search
Parallel hybrid metaheuristics

ABSTRACT

This paper presents a new hybrid island model genetic algorithm (HIMGA) to solve the well-known job shop scheduling problem (JSSP) with the objective of makespan minimization. To improve the effectiveness of the island model genetic algorithm (IMGA), we have proposed a new naturally inspired self-adaptation phase strategy that is capable of striking a better balance between diversification and intensification of the search process. In the proposed self-adaptation phase strategy, the best individuals are recruited to perform a local search using tabu search (TS), and the worst ones are recruited to perform a global search using a combination of 3 classical random mutation operators. The proposed algorithm is tested on 76 benchmark instances, with the proposed self-adaptation strategy, and without it using the classical alternatives, and also compared with other 15 algorithms recently reported in the literature. Computational results verify the improvements achieved by the proposed self-adaptation strategy, and show the superiority of the proposed algorithm over 13 of the compared works in terms of solution quality, and validate its effectiveness.

1. Introduction

Job shop scheduling problem (JSSP) is an NP-hard problem, and one of the most intractable combinatorial optimization problems considered to date. This intractability and importance for industrial engineering in terms of improving machine utilization and reducing cycle-time, made it so widely studied for more than fifty years.

Earlier works on JSSP were centered on exact methods such as branch and bound to find optimal solutions for small size problems (Carlier & Pinson, 1989; Lageweg, Lenstra, & Rinnooy Kan, 1977); however, they failed in solving problems of a bigger size in practical computational cost. For that reason, research focus was shifted towards approximation methods, which do not guarantee finding optimal solutions, but there is a considerable probability of finding near optimal solutions in practical computational cost. Initially, they were limited to simple heuristic methods such as dispatching rules (Blackstone, Phillips, & Hogg, 1982) and shifting bottleneck procedure (Adams, Balas, & Zawack, 1988) which were distinguished in terms of efficiency, but undistinguished in terms of effectiveness. Therefore, research focus was shifted again towards more sophisticated approximation methods called metaheuristics, which explore the search space more effectively by employing more intelligence in escaping from local optimums. The most common ones that have been used for JSSP include genetic algorithm (GA) (Watanabe, Ida, & Gen, 2005), simulated annealing (SA) (Satake, Morikawa, Takahashi, & Nakamura, 1999; Van Laarhoven, Aarts, & Lenstra, 1992), tabu search (TS) (Nowicki & Smutnicki, 1996; Zhang, Li, Guan, & Rao, 2007), ant colony optimization (ACO) (Fığlalı, Özkale, Engin, & Fığlalı, 2009), and particle swarm optimization (PSO) (Lian, Jiao, & Gu, 2006; Lin et al., 2010). However, due to the stubborn nature of JSSP, sole metaheuristic methods left a considerable space of improvements; consequently, recently, most of researchers tend to develop hybrid methods that combine the complementary strengths of different metaheuristic. Actually, to the best of our knowledge, the best-so-far method for JSSP is a tabu search/path relinking method proposed by Peng, Lü, and Cheng (2015). An overview of JSSP techniques can be found in Zobolas, Tarantilis, and Ioannou (2008), while a comprehensive survey of them can be found in Jain and Meeran (1999).

GA has gained a well-earned reputation in being one of the best methods in solving JSSP, but it still has its own shortcomings like premature convergence, and the lack of intensification capabilities, i.e. searching in the small regions of the search space that are likely close to the optimal solutions (Zobolas et al., 2008). Various approaches have been developed during the last 4 decades to overcome these shortcomings, the most common ones include

☆ This manuscript was processed by Area Editor T.C. Edwin Cheng.

* Tel.: +90 256 213 7503x3577; fax: +90 256 213 6686.
E-mail address: mohamed.kurdi@adu.edu.tr

http://dx.doi.org/10.1016/j.cie.2015.07.015
0360-8352/

parallelization of GA into multiple sub-populations such as the island model GA (IMGA) to delay the premature convergence and improve the search diversification capabilities (Asadzadeh & Zamanifar, 2010; Gu, Gu, & Gu, 2009; Park, Choi, & Kim, 2003; Qi, Burns, & Harrison, 2000; Yusof, Khalid, Hui, Md Yusof, & Othman, 2011), and hybridization with local search methods (LSMs) to add intensification capabilities to it such as TS (Amirghasemi & Zamani, 2015; Cheng, Peng, & Lü, 2013; Meeran & Morshed, 2014; Ombuki & Ventresca, 2004), SA (Tamilarasi & kumar, 2010; Wang & Zheng, 2001), and new LSMs (Asadzadeh, 2015; Gao, Zhang, Zhang, & Li, 2011; Qing-dao-er-ji & Wang, 2012; Zhou, Feng, & Han, 2001). A tutorial survey of JSSP using GAs can be found in Cheng, Gen, and Tsujimura (1996), while a tutorial survey of JSSP using hybrid GAs can be found in Cheng, Gen, and Tsujimura (1999).

Recently, it has been shown that combining both of parallelization and hybridization in one framework is advantageous. Kalantari and SanieeAbadeh (2013) proposed an IMGA hybridized with a LSM based on pair-wise interchanged method, but this kind of LSMs acts like a random swap mutation (Gen & Cheng, 1997), cannot improve the makespan, and may generate infeasible solutions since there is no guarantee that the interchanged operations do belong to the critical path (Taillard, 1994); Asadzadeh (2014) proposed an IMGA hybridized with variable neighborhood search (VNS), but the VNS is applied on each individual after the end of each generation, and involves just two random mutation operators. Therefore, both of them may not achieve a proper balance between diversification and intensification.

It is known that there are two phases of evolution in GA: the cooperation phase implemented by crossover, and self-adaptation phase implemented by mutation. Whereas cooperation phase means individuals evolve by exchanging their information about the search space, self-adaptation phase means individuals evolve independently using only their own information (Hertz & Kobler, 2000).

In this work, in order to overcome the limitations of the previously discussed hybrid IMGA models, and design a new effective algorithm that can strike a better balance between diversification and intensification, an IMGA hybridized with TS (which is one of the best LSMs for JSSP) (HIMGA) is proposed. The proposed algorithm utilizes a new naturally inspired self-adaptation phase strategy, in which the best individuals are recruited to perform a local search using TS, and the worst ones are recruited to perform a global search using a combination of 3 classical random mutation operators.

The remainder of this paper is organized as follows. In the next section, JSSP definition is presented. In Section 3, the HIMGA framework is explained. Section 4 presents the computational results. The conclusions are made in Section 5.

2. Problem definition

The classical JSSP with the objective of makespan minimization, which is represented by $J||C_{\max}$ using the classification scheme of Graham, Lawler, Lenstra, and Kan (1979), consists of a set of n jobs $\{J_j\}1 \leqslant j \leqslant n$ needs to be processed on a set of m machines $\{M_r\}1 \leqslant r \leqslant m$. The processing of job J_j on machine M_r is called the operation O_{jr}, and lasts for an uninterrupted specified time period called processing time P_{jr} (preemption is not allowed). Two constraints are imposed on the problem: the precedence constraint which specifies that each job J_j should be processed on each machine M_r according to a predefined sequence called topological sequence, and the capacity constraint which specifies that each machine M_r can process only one job J_j at a time. The start time and completion time of operation O_{jr} are denoted as S_{jr}, C_{jr} respectively. A schedule (or solution) is the set of completion times for all operations; a feasible schedule is a schedule that satisfies the problem constraints. The time needed for the completion of all the operations is called makespan and denoted as $C_{\max}$, where $C_{\max} = \max_{1 \leqslant j \leqslant n,\ 1 \leqslant r \leqslant m} C_{jr}$. The objective of the problem becomes finding a feasible schedule that minimizes $C_{\max}$ as much as possible.

An example of a 3×3 JSSP is given in Table 1. The data include the topological sequence of all jobs with their processing times, for example, job 2 is processed in this order $O_{21} \to O_{22} \to O_{23}$, i.e. it is processed on machine 1 for 4 time units, then on machine 2 for 5 units, then on machine 3 for 3 units. A possible solution of the 3×3 JSSP represented by a Gantt chart is given in Fig. 1.

3. The hybrid island model genetic algorithm

Current implementations of parallel GAs (PGAs) are classified into three models: master–slave, fine-grained, and IMGA. In the first, there is one population, and computation is carried out by many processors; therefore, the quality of solutions is preserved while reducing the computational cost. In the second, there is also one population, but each individual is assigned to one processor to carry out only its operations, selection and reproduction are limited to neighbors. In the third, (which is also called multi-population or coarse-grained) the whole population is split into multiple independent islands (sub-populations) that are assigned to different processors, and explore different parts of the search space using their own evolution processes, with occasionally exchange of information via a migration policy (Engelbrecht, 2007); this imitates the nature in a better way, delays the premature convergence, and enhances the search diversification (Konfrst, 2004).

In this work, we adopt IMGA, because it is easy to implement in a serial manner (pseudo-parallel) on a single processor system, which serves our objectives that are centered on effectiveness rather than efficiency.

For the sake of simplicity, the LSM adopted is the classical TS algorithm, based on the one proposed by Nowicki and Smutnicki (1996), which has been found to be particularly successful approach for the JSSP. However, our method does not integrate its diversification procedure (e.g., long-term memory and the back-jump tracking procedure) which is used to restart the search process when gets trapped in a local optimum, because the proposed algorithm will count on TS for doing only the intensification part and leave the task of diversification to IMGA, i.e. IMGA will identify the promising regions, whose local optimums will be subsequently located by TS. The general framework of the proposed algorithm HIMGA is described in Fig. 2.

3.1. Migration policy

It plays a crucial role in the performance, since it controls the process of information exchange between islands. Its mechanism depends mainly on four factors: communications topology, migration rate, selection method, and replacement method (Engelbrecht, 2007).

Table 1
An example of a 3×3 JSSP.

Job	Machine/processing time		
J1	M1/3	M3/4	M2/9
J2	M1/4	M2/5	M3/3
J3	M2/4	M3/6	M1/4