Mathematical and Computer Modelling 52 (2010) 1957–1965

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

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

An agent-based parallel approach for the job shop scheduling problem with genetic algorithms

Leila Asadzadeh *, Kamran Zamanifar

Department of Computer Engineering, University of Isfahan, Isfahan, Iran

ARTICLE INFO

Article history:
Received 23 September 2009
Received in revised form 25 December 2009
Accepted 31 January 2010

Keywords:
Job shop scheduling problem
Parallel genetic algorithms
Agents
Multi-agent systems

ABSTRACT

The job shop scheduling problem is one of the most important and complicated problems in machine scheduling. This problem is characterized as NP-hard. The high complexity of the problem makes it hard to find the optimal solution within reasonable time in most cases. Hence searching for approximate solutions in polynomial time instead of exact solutions at high cost is preferred for difficult instances of the problem. Meta-heuristic methods such as genetic algorithms are widely applied to find optimal or near-optimal solutions for the job shop scheduling problem. Parallelizing the genetic algorithms is one of the best approaches that can be used to enhance the performance of these algorithms. In this paper, we propose an agent-based parallel approach for the problem in which creating the initial population and parallelizing the genetic algorithm are carried out in an agent-based manner. Benchmark instances are used to investigate the performance of the proposed approach. The results show that this approach improves the efficiency.

1. Introduction

The job shop scheduling problem is one of the most important problems in machine scheduling, and it is an important task for the manufacturing industry in terms of improving machine utilization and reducing product cycle-times. This problem can be described as follows. A set of n jobs $\{J_i\}_{1\leq i\leq n}$ and a set of m machines $\{M_r\}_{1\leq r\leq m}$ are given. Each job consists of a sequence of m_i operations $O_{i1}, O_{i2}, \ldots, O_{im_i}$ that must be processed in a specified order. O_{ij} is the jth operation of job i that must be processed on machine M_k for a processing time period t_{ik}. For each operation O_{ij}, M_k and t_{ik} are predefined and preemption is not allowed. Each machine can process only one job and each job can be processed by only one machine at a time. The duration in which all operations for all jobs are completed is referred to as the makespan. A schedule determines the execution sequence of all operations for all jobs on machines. The objective is to find the optimal schedule. The optimal schedule is the schedule that minimizes the makespan. Due to factorial explosion of possible solutions, job shop scheduling problems are considered to be a member of a large class of intractable numerical problems known as NP-hard [1]. The high complexity of the problem makes it hard and in some cases impossible to find the optimal solution within reasonable time. Hence, searching for approximate solutions in polynomial time instead of exact solutions at high cost is preferred for difficult instances of the problem.

Historically, the job shop scheduling problem has been primarily treated using the branch and bound method [2–4], heuristic rules [5–7] and the shifting bottleneck procedure [8]. In recent years, meta-heuristic methods have been widely applied to this problem. These methods, such as taboo search [9–11], simulated annealing [12–15], genetic algorithms [16–20], neural networks [21] and ant colony optimization [22–25], are well suited to solving complex problems with high costs. A survey of job shop scheduling techniques can be found in [1].

Compared with other meta-heuristic methods, genetic algorithms are widely used to find the best or near-best solutions for various optimization problems. Many genetic algorithm-based approaches have been proposed for job shop

* Corresponding author.
E-mail addresses: leila_asadzadeh_cs@yahoo.com (L. Asadzadeh), zamanifar@eng.ui.ac.ir (K. Zamanifar).

0895-7177/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.04.019

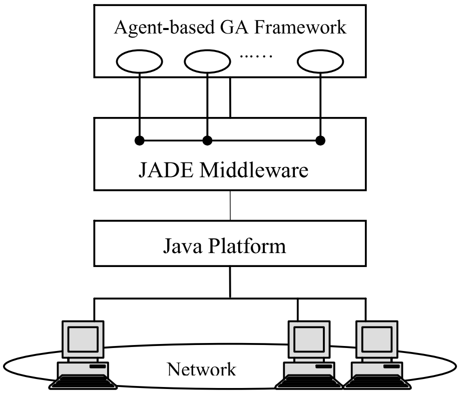

Fig. 1. The layers of the proposed agent-based architecture.

scheduling problems. In [26,27], the authors introduce an approach that uses load balancing of machines as an important parameter in job assignment. An advantage of these approaches is that they maximize machine utilization while minimizing the makespan. Ombuki and Ventresca [28] proposed a local search genetic algorithm that uses an efficient solution representation strategy in which both checking of the constraints and repair mechanism can be avoided. In their approach, at the local search phase a new mutation-like operator is used to improve the solution quality. They also developed a hybrid strategy using the genetic algorithm reinforced with a taboo search for the problem. Lin et al. [29] introduced a hybrid model consisting of coarse-grain genetic algorithms connected in a fine-grain style topology. Their method can avoid premature convergence, and it produced excellent results on standard benchmark job shop scheduling problems. Meta-heuristic methods such as neural networks, simulated annealing and local search were used with genetic algorithms to provide high performance in finding solutions for the problem [17,19,20]. Chen et al. [30] gave a tutorial survey of recent works on various hybrid approaches of the genetic algorithms proposed so far for the job shop scheduling problem. Wang and Zheng [20], by combining simulated annealing and genetic algorithms, developed a general, parallel and easily implemented hybrid optimization framework, and applied it to the job shop scheduling problem. Based on an effective encoding scheme and some specific optimization operators, some benchmark job shop scheduling problems are well solved by the hybrid optimization strategy. In [17], the authors proposed a hybrid genetic algorithm. In their approach, the schedules are constructed using a priority rule in which the priorities are defined by the genetic algorithm. Schedules are constructed using a procedure that generates parameterized active schedules. After a schedule is obtained, a local search heuristic is applied to improve the solution. In [19], a hybrid method is proposed to obtain a near-optimal solution within a reasonable amount of time. This method uses a neural network approach to generate initial feasible solutions and then a simulated annealing algorithm to improve the quality and performance of the initial solutions in order to produce the optimal/near-optimal solution. Chen et al. [31] proposed an agent-based genetic algorithm that accelerates the creation of the initial population. In this approach, the processing of selection, crossover and mutation can be controlled in an intelligent way.

In this paper, we propose an agent-based parallel genetic algorithm for the job shop scheduling problem. In our approach, the initial population is created in an agent-based way by using the method proposed in [31], and then an agent-based method is used to parallelize the genetic algorithm.

The reminder of this paper is organized as follow. In Section 2, we give the details of our proposed agent-based architecture and parallel genetic algorithm. In Section 3, we discuss the implementation and experimental results of the proposed approach. Our conclusion is given in Section 4.

2. Agent-based approach for job shop scheduling problem

Agents and multi-agent systems have wide application in parallel and distributed systems. One of the important features of agents is their capability in parallel implementing of genetic algorithms. We used this feature in our approach and propose a parallel and distributed model for the job shop scheduling problem. We developed a multi-agent system containing some agents with special actions that are used to parallelize the genetic algorithm and create its population.

In this section we introduce our agent-based method and briefly describe its structure. The architecture of our method and different layers of it are introduced in Section 2.1. In Section 2.2, we describe how the initial population can be produced in an agent-based way. Our proposed method that is used to parallelize the genetic algorithm for solving the problem is explained in Section 2.3.

2.1. Agent-based architecture

The layers of our proposed agent-based architecture for the job shop scheduling problem are shown in Fig. 1. The computer network that provides a context for developing and implementation of the proposed model is at the lower layer