Applied Soft Computing 13 (2013) 1440–1447

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

# Hybrid Evolutionary Algorithm for job scheduling under machine maintenance

Ruhul Sarker[a,*], Mohd Omar[b], S.M. Kamrul Hasan[a], Daryl Essam[a]

[a] School of Engineering and Information Technology, University of New South Wales, ADFA Campus, Canberra 2600, Australia
[b] Institute of Mathematical Sciences, University of Malaya, Lembah Pantai, Kuala Lumpur 50603, Malaysia

ARTICLE INFO

*Article history:*
Received 9 December 2011
Received in revised form 13 April 2012
Accepted 18 April 2012
Available online 15 May 2012

*Keywords:*
Job scheduling
Machine maintenance
Hybrid Evolutionary Algorithm

ABSTRACT

The job scheduling problem (JSP) belongs to the well-known combinatorial optimization domain. After scheduling, if a machine maintenance issue affects the scheduled processing of jobs, the delivery of jobs must be delayed. In this paper, we have first proposed a Hybrid Evolutionary Algorithm (HyEA) for solving JSPs. We have then analyzed the effect of machine maintenance, whether preventive or breakdown, on the job scheduling. For the breakdown maintenance case, it is required to revise the algorithm to incorporate a rescheduling option after the breakdown occurs. The algorithm has been tested by solving a number of benchmark problems and thence comparing them with the existing algorithms. The experimental results provide a better understanding of job scheduling and the necessary rescheduling operations under process interruption.



## 1. Introduction

Job scheduling is a well-known task in the manufacturing industry. In this paper, we consider Job Scheduling Problems (JSPs) with $n$ jobs and $m$ machines that involve active constraints such as machine capacity and precedence requirements. It is considered that a machine can only perform a particular type of operation. Thus, a machine is able to execute just a single operation of a job. The operations are non-preemptive; an operation can neither be paused nor resumed after it is started. The execution time for each operation is known. The setup time or cost is assumed to be negligible. An operation can only be started if the preceding operations of the same job are complete. This description is similar to traditional job-shop scheduling problems [1,2].

Although a considerable amount of research has been carried out on how to develop effective solution approaches for JSPs, no single algorithm is well-accepted for all kinds of JSPs [3]. The deterministic scheduling algorithms that converge to the optimal solution are suitable for small-scale problems. They are, however, incapable of handling complex and large-scale problems. In contrast, the stochastic and heuristic algorithms can handle large-scale problems, and provide near optimal solutions with a small computational effort [4]. As a result, these algorithms are popular when solving complex and large-scale scheduling problems.

In almost all research on JSPs, schedules are produced under ideal conditions, assuming there will be no disruptions of any type. However, machine unavailability is a common event on the shop floor due to both preventive and breakdown maintenance of machineries and their supporting equipments [4]. The inclusion of such unavailability of processing resources with the JSPs makes the resulting problem not only more practical, but also more complex and challenging. The preventive maintenance schedules, which are usually known in advance, can easily be incorporated in generating a job scheduling. However, in the case of sudden machine failure, the operations scheduled on the broken machine cannot be resumed until the machine is either appropriately repaired or replaced by a new one. During and after the machine repair/replacement, the continued implementation of the schedule generated earlier would delay the completion of some jobs due to active precedence constraints. In order to minimize the delay, it is very important to re-optimize the remaining operations at the time of machine breakdown. Even after re-optimization, it is expected that there will be some delay in the completion of some or all jobs.

In this research, we have proposed a Hybrid Evolutionary Algorithm (HyEA) for solving JSPs, that combines a genetic algorithm with a local search heuristic technique. We have considered makespan minimization as the fitness measure, as it is widely used in job scheduling problems. The total time between the starting of the first operation and the ending of the last operation is termed as the *makespan*. For a candidate solution, the local search heuristic technique identifies any gap left between any two consecutive operations (/jobs) on a machine. A job from the right of the gap can then be placed in it, if the gap is big enough for the operation without violating any precedence constraints. In addition, the job can also be placed in the gap, even if the gap is not big enough, but within a certain tolerance limit of the operation time, if it improves the overall fitness value. In this case it may need to shift other jobs

* Corresponding author. Tel.: +61 2 6268 8051.
*E-mail address:* r.sarker@adfa.edu.au (R. Sarker).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.04.032

to the right. We have extended this algorithm to study JSPs under sudden machine breakdowns. In the case of machine breakdown, the breakdown information is known after the actual breakdown, in fact, when the schedule is under implementation. In such a case, it is necessary to re-optimize the remaining operations by taking into account the machine downtime.

We have solved 40 benchmark test problems using HyEA and have experimented by varying the tolerance limit for the local search heuristic. We have shown that the inclusion of the heuristic not only improves the performance of the EAs, but also reduces the overall computational requirements. To study JSPs with machine maintenance, we have chosen a manufacturing shop with ten machines. For experimentation, we have used probability distributions to generate the breakdown scenarios. We have found that the revised solution is able to recover from most of the breakdowns, regardless of when they occur.

The novelties of our research presented in this paper can be summarized as follows. Although there exist a number of approaches for solving JSPs, no single algorithm is well-accepted for all kinds of JSPs. So there is a scope to develop new efficient algorithms. So our first goal was to develop a new Hybrid Evolutionary Algorithm (HyEA) that provides effective solutions for JSPs. Secondly almost all researches in JSPs dealt with the problem under ideal condition. However the real life situation faces many different disruptions. So we extended our study to deal with the disruption cases when the job processing is underway according to the job scheduling solution. This is a novel concept that would improve the efficiency in many manufacturing industry. Finally, the solution approach proposed in this research does not solve the corresponding mathematical model of the problem. Instead it considers the solution vector and the constraints from the problem description directly.

The paper is organized as follows. After the introduction, a brief outline of a standard job scheduling problem with and without disruption is given. The evolutionary algorithm to solve JSPs is discussed in Section 3. Section 4 provides an experimental study for JSPs with and without random machine breakdowns. Finally, conclusions are given in Section 5.

## 2. Job scheduling with machine maintenance

The standard job scheduling problem is defined with the following assumptions:

- Each job consists of a finite number of operations.
- The processing time for each operation in a particular machine is defined.
- There is a pre-defined sequence of operations that has to be maintained to complete each job.
- Delivery times of the products are not defined.
- There is no setup cost or tardiness cost.
- A machine can process only one job at a time.
- Each job visits each machine only once.
- No machine can deal with more than one type of operation.
- The system cannot be interrupted until each operation of each job is finished.
- No machine can halt a job and start another job before finishing the previous one.
- Each and every machine has full efficiency.

The objective of the problem is to minimize the maximum time taken to complete each and every job while satisfying the required operational sequence and the start time of each job.

The objective value is calculated as follows. Let $C_{ij}$ be the completion time of operation $O_{ij}$ of job $j$ on machine $i$. Note that $t_{ij}$ is the processing time of operation $O_{ij}$. The objective is to find a schedule that minimizes the maximum completion time, $C_{max} = \max_{ij} C_{ij}$. The value of $C_{max}$ is also called makespan or the length of the schedule.

In practice, the operations may be interrupted and delayed for many reasons, such as machine breakdown (including cutting tools and fixtures), unexpected machine setting up costs, arrival of a new priority job, process time variation, change of job priorities, defective materials, delay in transfer line between machines, and order cancellation [5,6]. Machine breakdown is considered as one of the most challenging issues in production scheduling. The unavailability of machines, due to planned preventive maintenance, can be incorporated as a constraint when solving for an optimal schedule. In the case of sudden breakdowns, the easiest solution is to apply some dispatching rules which help to select an operation immediately after the breakdown occurs [7].

Recently, reactive scheduling has been introduced to deal with the unexpected interruptions. Liu et al. [8] have proposed to divide the scheduling process into two non-overlapping parts: *predictive* before the breakdown occurs, and *reactive* when the machine is recovered after the breakdown. Fahmy et al. [5] have suggested inserting dummy operations to remove the affected operations from the schedule and to later reschedule them. The duration of the dummy operation being equal to the recovery time of the broken machine. Abumaizar and Svestka [9] have proposed to repair the reactive schedules using a *right shifting* process. Their process shifts each and every operation to its right after the machine breakdown. Wu et al. [10] developed a genetic algorithm with a pairwise-swapping heuristic and proposed using the right shifting technique to re-optimize. The drawback in this technique is the presence of uniform shifting for every operation, which increases the machine idle times between consecutive operations.

## 3. A new Hybrid Evolutionary Algorithm

According to the problem definition, the sequence of machines used (which is also the sequence of operations) by each job is given. That means, each operation is linked to one particular machine. In this case, if we know either the starting or finishing time of each operation, then we can calculate the makespan for each job and generate the whole schedule. In JSPs, the main problem is to find the order of jobs to be operated on each machine that minimizes the overall makespan. As indicated earlier, we have proposed a new Hybrid Evolutionary Algorithm in this paper. The details of the algorithm are discussed below.

Evolutionary Algorithms (EAs) are a population based stochastic global search algorithm. Over the last few decades, these algorithms have shown tremendous success in solving simple as well as complex optimization problems [11]. EAs use crossover and/or mutation as their basic search mechanism. However, for some complex problem scenarios, EAs are found to be slow in convergence and even trapped in local optima. The addition of a local search method to EAs helps to accelerate the convergence and escaping from the local optima. The incorporation of a local search method with EAs is recognized as either hybrid EAs or memetic algorithm. EAs and hybrid EAs are widely used in solving combinatorial optimization problems including job scheduling problems [12–14]. Interestingly, in most job scheduling cases, EAs use a special type of chromosome representation that allows mapping the problem instead of dealing with its detailed mathematical formulation. Such representation can also be applied in other combinatorial optimization problems [15]. The chromosome representation and the local search applied in this research are described in the next two sections.