Computers & Industrial Engineering 56 (2009) 179–185

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Multi-criteria sequence-dependent job shop scheduling using genetic algorithms

Andrew Manikas [a,*], Yih-Long Chang [b]

[a] The College of Business Administration, University of Wisconsin, Oshkosh, 800 Algoma Boulevard., Oshkosh, WI 54901, USA
[b] The College of Management, Georgia Institute of Technology, 800 West Peachtree Street, Atlanta, GA 30332-0520, USA

ARTICLE INFO

Article history:
Received 24 February 2007
Received in revised form 2 May 2008
Accepted 2 May 2008
Available online 11 May 2008

Keywords:
Job shop scheduling
Genetic algorithm
Sequence-dependent setup times

ABSTRACT

Real world job shops have to contend with jobs due on different days, material ready times that vary, reentrant workflows and sequence-dependent setup times. The problem is even more complex because businesses often judge solution goodness according to multiple competing criteria. Producing an optimal solution would be time consuming to the point of rendering the result meaningless. Commonly used heuristics such as shortest processing time (SPT) and earliest due date (EDD) can be used to calculate a feasible schedule quickly, but usually do not produce schedules that are close to optimal in these job shop environments. We demonstrate that genetic algorithms (GA) can be used to produce solutions in times comparable to common heuristics but closer to optimal. Changing criteria or their relative weights does not affect the running time, nor does it require programming changes. Therefore, a GA can be easily applied and modified for a variety of production optimization criteria in a job shop environment that includes sequence-dependent setup times.

1. Introduction

Sequence-dependent setup times in job shops are very common. Moving from one family of parts to another can involve a significant setup time compared to producing similar parts to those that were just run. This type of problem is mathematically complex to solve optimally, therefore simplistic heuristics like shortest processing time (SPT), earliest due date (EDD), first-in-first-out (FIFO) or even ad hoc scheduling are often used because management can readily understand them. Additional complexity arises because management routinely wants to consider multiple criteria when evaluating schedule goodness.

In many manufacturing environments, the sequence of jobs run on a particular machine affects the setup times. The author had the opportunity to work at a sheet metal fabricator in 1991 where the job shop environment had machines where the setup times would vary according to the sequence of jobs processed. For example, the turret punch could use the same punch for jobs of similar products while a different product would require the punch to be changed on the machine. The plant simply batched together alike jobs until customers called and complained about their orders being late, then the order of jobs queued in front of the machines was altered to reflect which customer had called most recently. The setup times for each order are different due to different previous job order. The similar job grouping by type was used in front of the numerical controlled (NC) machine. This situation is common in many environments with flexible manufacturing, where setting up once to run a particular major class of parts is done, then minor setups to alter specific products are done within the same major part class. The setup time for a particular major class of parts typically depends upon the previous part family.We consider a job shop environment with multiple machines where multiple jobs consisting of a set number of operations each must be scheduled according to various criteria. We will demonstrate that a simple genetic algorithm (GA) can produce a good result quickly for managers for a complex set of sequence-dependent job shop scheduling problems. A GA is a viable approach to solving optimization problems. The principles of a GA proposed by Holland (1975) are the foundation of all GAs. A GA simulates evolution via natural selection on a model of a problem. The idea is to evolve a population of candidate solutions to a given problem using operators inspired by natural genetic variation and natural selection (Mitchell, 1996).

The basic structure of a GA is:

- Create an initial set of solutions.
- Score the initial set of solutions.
- Loop for reproduction.
  - Copy the best solution to a new population.
  - Randomly Mutate a small subset of the population to introduce solution variety.
  - Mating to combine solutions in the population according to their fitness, in terms of score goodness, to create new solutions.
  - Score the new set of solutions.
- Until termination criteria.

* Corresponding author.
E-mail addresses: manikasa@uwosh.edu (A. Manikas), YihLong.Chang@mgt.gatech.edu (Y.-L. Chang).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.05.002

This paper is organized into the following sections; Section 2 is a literature review of job shop scheduling, sequence-dependent setups in scheduling, and GAs as an optimization tool. Section 3 is a description of the problem we are trying to solve. Section 4 is the mechanism behind how GAs work in general and specifics on our GA. Section 5 explains how the tests are set up. Section 6 is a robustness test of the GA for various types of job shop problems. Section 7 is the summary and discussion of the implications of the results.

2. Literature review

Job shops provide a unique scheduling problem because the routings are based upon the jobs that need to be processed and therefore the resource requirements are not based upon the quantity as in a flow shop but rather the routings for the products being produced. Job shop scheduling belongs to the class of intractable problems known as NP-hard (Jain & Meeran, 1999).

Flow shop problems with sequence-dependent setups are extensively covered in the current literature (Gupta, 1986; Kochhar & Morris, 1987; Osman & Potts, 1989; Reeves, 1995; Rios-Mercado & Bard, 1999; Ruiz, Maroto, & Alcaraz, 2005). Very little has been done in job shops with sequence-dependent setups. Solution methods for job shop scheduling range from simple greedy heuristics such as SPT and EDD to more complex heuristics such as branch and bound (Brucker, Jurisch, & Sievers, 1994), tabu search (Nowicki & Smutnicki, 1996), and genetic algorithms as will be discussed in this section.

Finding a sequence of operations, even for a single machine with sequence-dependent setup times and a single measure such as makespan is equivalent to the traveling salesman problem (TSP) and is therefore NP-hard (Pinedo, 2002). Schutten (1998) discusses extending the shifting bottleneck procedure (Adams, Balas, & Zawack, 1988) to accommodate family setup times, staggered release and due dates and other more realistic constraints one extension at a time. However, they conclude that to alter the shifting bottleneck to handle these features results in large computation times and is therefore not usable in practical situations. Sun and Noble (1999) solve a job shop scheduling problem with the objective of minimizing the weighted sum of squared tardiness by decomposing it into a series of single machine scheduling problems. They find superior results to simulated annealing and tabu search with their method.

Classical job shop scheduling problems have also been solved using a GA. Candido, Khator, and Barcia (1998) add realistic constraints and use multiple objectives for their GA. The GA is used to find the initial solutions and then these solutions are enhanced with a local hill climbing routine. Finally a modified GA is used to further refine the schedule. However, this routine was not applied to sequence-dependent setup times. The ability of GAs to handle complex constraints is further shown by allowing dual-resource constraints in a scheduling problem (ElMaraghy, Patel, & Abdallah, 2000). They achieve better solutions by forcing feasible solutions from birth rather than allowing infeasible solutions to exist in the population. They choose to use a random initial population rather than heuristic based. The paper also shows that linear order crossover (LOX) performs better than partially matched crossover for their particular problem. The LOX crossover method preserves relative position between genes (Falkenauer & Bouffouix, 1991). Cai, Wu, and Yong (2000) devise a GA specifically designed for job shop problems. The offspring from the mutation and crossover are not used directly in the next generation but in a local search routine. Given these initial successes with applying genetic algorithms, we choose to apply a genetic algorithm as the method to solve our series of job shop problems. Our problems are made more realistic and more complex by allowing multiple criteria in the objective function.

Having a single criterion to solve for does not give the production planner much control to differentiate among many competing requirements or constraints. Richter (2002) argues that the use of a multiple objective (criteria) fitness function has a positive effect on convergence speed. Our job shop scheduling problems involve earliness, tardiness, customer/job ranking and $C_{\max}$ (makespan) as criteria to form the objective function. However any number of criteria can be easily included in the weighted objective function in our GA solution. We use weighted multiple criteria similar to Fulya, Mitsuo, Lin, and Turan (2006).

3. Problem description

In many manufacturing environments, the sequence of jobs run on a particular machine affects the setup times. The author had the opportunity to work at a sheet metal fabricator in 1991 where the job shop environment had machines where the setup times would vary according to the sequence of jobs processed. For example, the turret punch could use the same punch for jobs of similar products while a different product would require the punch to be changed on the machine. The plant simply batched together alike jobs until customers called and complained about their orders being late, then the order of jobs queued in front of the machines was altered to reflect which customer had called most recently. The setup times for each order are different due to different previous job order. The similar job grouping by type was used in front of the numerical controlled (NC) machine. This situation is common in many environments with flexible manufacturing, where setting up once to run a particular major class of parts is done, than minor setups to alter specific products are done within the same major part class. The setup time for a particular major class of parts typically depends upon the previous part family.

In manufacturing environments a feasible good solution is a must. An optimal solution is a goal that cannot often be obtained given time constraints. Commonly used heuristics such as SPT and EDD can quickly create feasible solutions, but as problem complexity increases the solutions may be far from optimal. In this paper, to obtain not only a feasible solution, but to also find a better one according to management's criteria, we will develop a GA procedure and compare its results against common heuristics.

To make our job shop scheduling problems more realistic, in addition to sequence setup times, the environment contains staggered release dates and recirculation. Staggered release dates means that we do not assume all jobs are ready to start at the same time. This is common in real life since material availability is often based upon supplier deliveries of components or raw materials. Recirculation is the ability of a job routing to visit the same machine more than once. The job shop scheduling problems used in this paper are non-delay, meaning that a machine must process a job if it is available for a waiting job. The job shop allows no preemption of jobs (a job being processed on a machine must finish completely before the next job starts, interruptions are not allowed). Precedence relationships among operations of the same job are respected. Each machine resource processes at most one job at a time.

We first use three criteria and weight them to form our objective function; (1) earliness, (2) tardiness, and (3) customer and/or job rank. We then alter the objective function to include a fourth term, $C_{\max}$. A schedule is evaluated according to the weighted cri-