Solving the Reentrant Permutation Flow-Shop Scheduling Problem with a Hybrid Genetic Algorithm

Jen-Shiang Chen¹, Jason Chao-Hsien Pan², and Chien-Min Lin³

¹Department of Business Administration
Far East University
Tainan, Taiwan (744), R.O.C.
Email: {Jen-Shiang Chen, jschenc@ms25.hinet.net}

²Department of Business Administration
Takming University of Science and Technology
Taipei, Taiwan (114), R.O.C.
Email: {Jason Chao-Hsien Pan, jcpan@tamking.edu.tw}

³Department of Industrial Management
National Taiwan University of Science and Technology
Taipei, Taiwan (106), R.O.C.

Most production scheduling-related research assumes that a job visits certain machines once at most, but this is often untrue in practical situations. A reentrant permutation flow-shop (RPFS) describes situations in which a job must be processed on machines in \( M_1, M_2, \ldots, M_m, M_1, M_2, \ldots, M_m, \ldots \), and no job is allowed to pass a previous job. This study minimizes makespan by using the genetic algorithm to move from local optimal solutions to near-optimal solutions for RPFS scheduling problems. In addition, the hybrid genetic algorithm (HGA) improves the genetic algorithm’s performance in solving RPFS.

Significance: This paper studies the reentrant permutation flow-shop scheduling problem with the objectives of minimizing the makespan of jobs. The genetic algorithm and hybrid genetic algorithm are presented for the proposed problem.

Keywords: Scheduling, genetic algorithm, hybrid genetic algorithm, reentrant permutation flow-shop.

(Received 13 September 2005; Accepted in revised form 7 January 2009)

1. INTRODUCTION

In production management, a scheduling problem is defined as some specific work time hypothesis regarding assignment of resources, including equipment and human resources (labor), in order to complete work in a certain amount of time. In the complex and dynamic world of manufacturing systems, scheduling is an extremely important issue. Scheduling deals with the allocation of scarce resources to tasks over time. In different machine environments, these problems can be categorized into single machine, parallel machines, flow-shop, job-shop and open-shop.

In manufacturing and assembly facilities, many operations must be completed for every job. Often, these operations must be done on all jobs in the same order, implying that the jobs must follow the same route. One assumes these machines are set up in series, and this environment is referred to as a flow-shop. Classical flow-shop scheduling problems assume that each job visits each machine only once (Baker 1974), but this assumption is sometimes violated in practice. A new type of manufacturing shop, the reentrant flow-shop, has recently attracted attention. The basic characteristic of a reentrant shop is that a job visits certain machines more than once. In a reentrant flow-shop (RFS), all jobs have the same route through the shop machines and the same sequence is executed several times (levels) to complete the jobs. For example, in semiconductor manufacturing, each wafer re-visits the same machines for multiple processing steps (Vargas-Villamil and Rivera 2001). The wafer traverses flow lines several times to produce a different layer in each circuit (Bispo and Tayur 2001).

The reentrant permutation flow-shop problem (RPFS) is a special case of the RFS problem. In a RFS, if job ordering is the same on any machine at each level, there is no passing since no job is allowed to pass a previous job (Pan and Chen 2003). Finding an optimal schedule that minimizes the makespan in RPFS is never easy. In fact, flow-shop scheduling, the sequencing problem in which \( n \) jobs must be processed on \( m \) machines, is known to be NP-hard (Kubiak et al. 1996, Pinedo 2002, Wang et al. 1997), except when the number of machines is smaller than or equal to two; the makespan can be minimized by Johnson’s (1954) rule.

Because of their intractability, this study presents genetic algorithms (GA) to solve the RPFS scheduling problems. GA has been widely used to solve classical flow-shop problems and has performed well. In addition, hybrid genetic algorithms (HGA) are proposed to improve the GA performance and the heuristic methods proposed by Pan and Chen (2004) for solving RPFS.