Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 36 (2009) 329 – 343
computers & operations research
www.elsevier.com/locate/cor

Pure cycles in flexible robotic cells

Hakan Gultekin, Oya Ekin Karasan, M. Selim Akturk*

Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey

Available online 10 October 2007

Abstract

In this study, an m-machine flexible robotic manufacturing cell consisting of CNC machines is considered. The flexibility of the machines leads to a new class of robot move cycles called the pure cycles. We first model the problem of determining the best pure cycle in an m-machine cell as a special travelling salesman problem in which the distance matrix consists of decision variables as well as parameters. We focus on two specific cycles among the huge class of pure cycles. We prove that, in most of the regions, either one of these two cycles is optimal. For the remaining regions we derive worst case performances of these cycles. We also prove that the set of pure cycles dominates the flowshop-type robot move cycles considered in the literature. As a design problem, we consider the number of machines in a cell as a decision variable. We determine the optimal number of machines that minimizes the cycle time for given cell parameters such as the processing times, robot travel times and the loading/unloading times of the machines.

Keywords: Flexible manufacturing systems; CNC; Industrial robots; Cellular automation; Optimization; Production control

1. Introduction

A manufacturing cell which consists of a number of machines and a material handling robot is called a robotic cell. An m-machine robotic cell can be seen in Fig. 1. Such manufacturing cells are used extensively in chemical, electronic and metal cutting industries. In this study, we will restrict ourselves with the metal cutting applications in an environment in which the machines are predominantly CNC machines so that the machines and the robot can communicate in a real-time basis. These machines are highly flexible and capable of performing several different operations by fast and inexpensive tool changes as long as the required tools are loaded in their tool magazines. There are no buffers at or between the machines. Hence, at any time instant, a part is either on one of the machines, on the robot or at the input or output buffer. Each of the identical parts to be produced is assumed to have a number of operations to be performed on the machines. As a consequence of the flexibility of the machines, these operations can be performed in any order on each of the machines. Furthermore, each operation can be assigned to any one of the machines. In order to use such systems efficiently, problems including the scheduling of the robot moves and the determination of the machines to perform each operation of each part should be solved. Throughout this study, these problems will be tackled with the objective of maximizing the throughput rate.

There is an extensive literature on robotic cell scheduling problems as summarized in the surveys of Crama et al. [1] and Dawande et al. [2]. Most of the research on this area assumed the cell to work as a flowshop-type system. More formally, each part is assumed to visit all of the machines in the same order, machine 1 through machine m in

* Corresponding author. Fax: +90 312 266 4054.
E-mail address: akturk@bilkent.edu.tr (M.S. Akturk).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2007.10.007

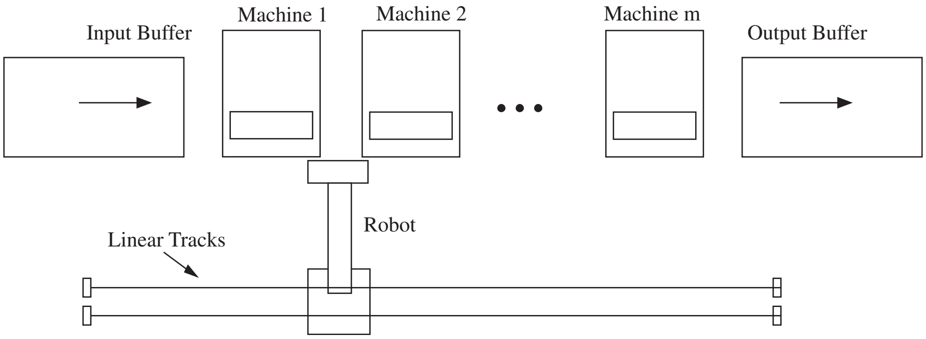

Fig. 1. m-Machine robotic cell.

an m-machine cell. Although this assumption might be valid for chemical or electroplating operations, it unnecessarily limits the number of alternative solutions in a flexible manufacturing cell (FMC), such as the one studied in this paper. Sethi et al. [3] developed the necessary framework for these scheduling problems and proved that for two machine producing identical parts, the optimal solution is a 1-unit cycle, where an n-unit cycle is defined to be a robot move cycle in which, starting with an initial state, the robot loads and unloads all of the machines exactly n times and returns back to the initial state. Note that in an n-unit cycle exactly n parts are produced. A similar result for three-machine case was proved by Crama and Van de Klundert [4]. However, the optimal solution is not necessarily a 1-unit cycle when the number of machines is greater than three [5]. Flexible robotic cells have recently been a topic of research. For example, in Akturk et al. [6], a robotic cell with two identical CNC machines possessing operational and process flexibility was considered. Operational flexibility is defined as the ability to interchange the ordering of several operations and process flexibility is defined as the ability to perform multiple operations on the same machine. For this problem, they proved that the optimal solution is either one of the two 1-unit cycles or the only 2-unit cycle.

In this study, we consider a new class of robot move cycles, named the pure cycles, resulting from the flexibility of the machines. Pure cycles are defined as the robot move cycles in which the robot loads and unloads all m machines with a different part during one repetition of the cycle. The terminology "pure" is to reflect the fact that each part is completely performed by only one machine and no part is transferred from one machine to another. Part movement is from the input buffer to one of the m machines and from this machine to the output buffer. A different sequence of loading and unloading operations leads to a different pure cycle. In earlier studies, we defined these cycles and showed that they perform efficiently in two-machine [7] and three-machine cells [8] in comparison to flowshop-type robot move cycles. These results are achieved by comparing one of the most simple and practical cycles among the class of pure cycles with the flowshop-type robot move cycles. However, the general problem of determination of the best pure cycle in an m-machine robotic cell was not tackled before. In this study we consider this problem.

This problem is somehow related with the parallel machine scheduling problem with a common server which can be reviewed in Hall et al. [9]. However, in that literature the setup time of the machines is arbitrary for each job and is given as a problem parameter. On the contrary, in our study, the setup time (transporting the part to the machine from the input buffer and loading it) is a variable depending on the robot move sequence. Additionally, different from that literature, the robot also performs the unloading of the machines. Finally, in that literature, it is assumed that a finite number of parts is to be produced and typically the objective function is either the minimization of the makespan or the total completion time. However, since we assume identical parts to be produced indefinitely and since the robot repeatedly follows a computer program, we consider cyclic scheduling. In a related study from this literature, Abdekhodaee et al. [10] considered scheduling of n different jobs on two parallel machines with the objective of minimizing the makespan where each job has its own processing and setup times given as problem parameters. Knowing that the general problem is NP-hard in the strong sense, they considered special cases of equal processing times and equal setup times. For the case of equal processing times, which is more related to our study, the authors prove that the problem is NP-hard in the ordinary sense when the setup times are small in comparison to the processing times and trivially solvable otherwise.

CNC machines possess several types of flexibilities. Such flexibilities are achieved by considering alternative tool types for operations and loading multiple tools to the tool magazines of the machines. This study focuses on the consequences of introducing such machine flexibilities to our system. We show that two specific pure cycles among