Computers and Operations Research 90 (2018) 161–172

Contents lists available at ScienceDirect

Computers and Operations Research

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

Cyclic scheduling of parts and robot moves in m-machine robotic cells

Hakan Gultekin[a,*], Betul Coban[a], Vahid Eghbal Akhlaghi[b]

[a] Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, Turkey
[b] Department of Industrial Engineering, Middle East Technical University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 8 November 2016
Revised 3 June 2017
Accepted 17 September 2017
Available online 19 September 2017

Keywords:
Robotic cell scheduling
Multiple parts
Mixed integer linear programming formulation
Hybrid metaheuristic
Throughput maximization

ABSTRACT

We consider a flow shop type manufacturing cell consisting of m machines and a material handling robot producing multiple parts. The robot transfers the parts between the machines and loads/unloads the machines. We consider the cyclic scheduling of the parts and the robot moves with the objective of maximizing the throughput rate. We develop a mixed integer linear programming formulation of the problem. The formulation is improved with several valid inequalities and reformulations of the constraints. We also develop a hybrid metaheuristic algorithm for this strongly NP-Hard problem. The algorithm is modified to handle both 1-unit and multi-unit robot cycles. Multi-threading is used to parallelize the algorithm in order to improve its efficiency. After calibrating the parameters of the heuristic algorithm, an extensive computational study is performed to evaluate its performance. The results of this study revealed that the developed heuristic provides near-optimal solutions in reasonable solution times. The effects of parallelization and the benefits of considering multi-unit cycles instead of 1-unit cycles are also quantified. Our computational tests show that multi-unit cycles improve the throughput rate by 9% on the average. The improvement can reach to 20% depending on the problem parameters.

1. Introduction

The use of industrial robots increases continuously among the manufacturing firms. According to Robotic Industries Association, 14,583 robots valued at \$ 817 million were ordered from North American robotics companies in the first half of 2016, an increase of two percent in units over the same period in 2015 (Robotics Online, 2016). The companies utilize industrial robots in order to increase productivity, safety, product quality, consistency, and flexibility. However, in order to get the maximum benefit from this high-cost investment, some complex operational problems must be solved. One of the most important operational problems is the sequencing and scheduling of the parts to be processed and the robot moves. In this study, we consider a flow shop type production system that consists of an input device denoted by M_0, a number of machines denoted by $M_1, \ldots, M_m$, an output device denoted by M_{m+1}, and a material handling robot as can be seen in Fig. 1. Such a production system is called a robotic cell. Multiple parts with different processing times are to be processed in this cell. The problem is to determine the part sequence and the robot move sequence that jointly maximize the throughput or equivalently minimize the cycle time. The cycle time is defined as the long run average time to produce a single part.

Such cells are used in many diverse industries including semiconductor manufacturing, printed circuit boards, glass products, textile mills and engine block manufacturing as reported in Dawande et al. (2007). Sethi et al. (1992) provided a real-life application that consists of two drilling and one boring machines and a single robot. The cell produces same or different castings to be used in truck differential assemblies. In another real-life application provided by Dawande et al. (2007), 28 varieties of hydraulic pump cover castings are produced through milling, rough and finish boring, drilling, and facing operations in a robotic cell. In the literature mostly the makespan and the cycle time minimization objectives are considered for robotic cells. In the makespan minimization case, it is assumed that the production is not repetitive. All machines are initially idle, a specific number of parts are to be processed, and at the end, all machines become idle again. Soukhal and Martineau (2005) developed an integer programming model and a Genetic Algorithm (GA) based heuristic for this problem. Carlier et al. (2010) proposed an exact branch-and-bound algorithm and developed a GA. Elmi and Topaloglu (2014) considered a hybrid flow shop type robotic cell in which there are different speed parallel machines at each stage. They developed an integer programming formulation and proposed a Simulated Annealing (SA) based algorithm. Liu and Kozan (2016) considered the job

* Corresponding author.
E-mail addresses: hgultekin@etu.edu.tr, hakan.gultekin@gmail.com (H. Gultekin), bcoban@etu.edu.tr (B. Coban), vahid.akhlaghi@metu.edu.tr (V.E. Akhlaghi).

http://dx.doi.org/10.1016/j.cor.2017.09.018
0305-0548/

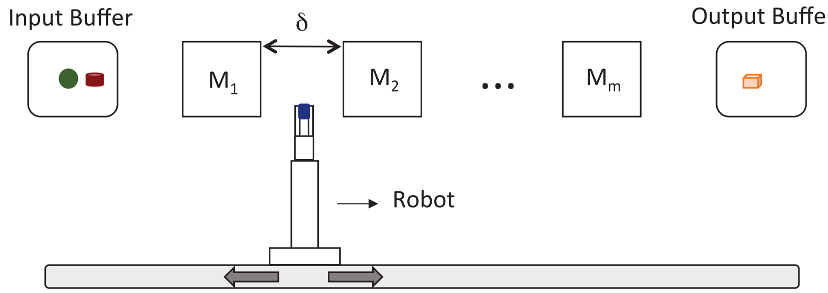

Fig. 1. Robotic cell.

shop version of the problem and developed a hybrid metaheuristic algorithm.

In the cycle time minimization case, it is assumed that the same set of parts is to be processed indefinitely. Therefore, the system is not required to start with an empty state in which all machines are idle. The state of the system can be identified with the position of the robot and the status of all machines (whether they are idle or loaded with a part). In order to satisfy repetitiveness, the final state of a cycle must be identical to its initial state. Such cyclic schedules are used extensively especially in mass production where the same set of parts are produced in large volumes or in the case of high demand diversity (Bożejko et al., 2015). In the latter case, a certain mix of items is produced in fixed intervals of time, cycle time. In a robotic cell, cyclic production refers to the repetition of the same sequence of robot moves indefinitely. Cyclic schedules are easy to implement and control for real life applications. Furthermore, Geismar et al. (2005) proved that such schedules are optimal for robotic flow shops producing identical parts. Sethi et al. (1992) defined an *n-unit* cycle as a sequence of robot activities in which the system returns to the same state after producing n parts. In these cycles, each machine is loaded and unloaded exactly n times and n parts are produced at the end.

Most studies on robotic cell scheduling problems assume the parts to be identical. With this assumption, the part sequencing problem vanishes and the robot move sequencing remains as the only problem. Sethi et al. (1992) proved that 1-unit cycles are optimal for two-machine cells producing identical parts. Crama and van de Klundert (1997) extended this result for three-machine cells. However, for m-machine cells where $m \geq 4$, Brauner and Finke (2001) showed that a multi-unit cycle can lead to a cycle time that is smaller than the smallest 1-unit cycle time. There are several studies that consider multi-unit cycles and develop exact or heuristic solution procedures for the identical parts case (see e.g. Kats et al., 1999; Che et al., 2011; Zhou et al., 2012; Li and Fung, 2014; Elmi and Topaloglu, 2016).

In the multiple part-types case, the repetitive production of a *Minimal Part Set (MPS)* is considered. An MPS is the smallest possible set of parts having the same proportions as the production target. For example, if the demands for three products for some fixed period of time are given as 400 for product A, 250 for product B, and 150 for product C, then the MPS consists of 8 product A's, 5 product B's, and 3 product C's that makes a total of 16 products. Given an MPS of n parts, an *MPS cycle* is equivalent to an n-unit robot move cycle. The cycle time can be defined as the total time to produce all n parts in the MPS. We utilize this definition of the cycle time throughout this study,. The per unit cycle time can be found easily by dividing this total time by n. A *Concatenated Robot Move (CRM)* cycle is a special class of MPS cycles in which the same 1-unit cycle is repeated n times (Sriskandarajah et al., 2004). The reason for considering CRM cycles lies in the easiness of implementation and control of these cycles. In most studies in the literature, the robot move sequence is fixed to a specific CRM cycle and the corresponding optimal part sequence is determined accordingly (Dawande et al., 2007). Sethi et al. (1992) solved the part sequencing problem for a given CRM cycle in a two-machine cell. Hall et al. (1998) proved that finding the optimal part sequence in two of the six possible CRM cycles in a three-machine robotic cell is strongly NP-Hard. Sriskandarajah et al. (1998) generalized these results for the m-machine cells and categorized the part sequencing problem for the CRM cycles into four categories depending on their complexity status. Kamalabadi et al. (2008) considered the part sequencing problem when the CRM cycle is fixed in a three-machine robotic cell and proposed a Particle Swarm Optimization (PSO) method. Abdulkader et al. (2013) considered the same problem in a four-machine robotic cell and proposed a GA. On the other hand, Hall et al. (1997) proved that the optimal cycle is not generally a CRM cycle even for the two-machine cells. They developed an algorithm with $O(n^4)$ complexity to determine the optimal MPS cycle for the two-machine case. Aneja and Kamoun (1999) improved the time complexity to $O(nlogn)$ by formulating the problem as a special case of the *Traveling Salesman Problem (TSP)*. However, the problem is strongly NP-Hard for the m-machine robotic cells when $m \geq 3$ and the robot sequence is not fixed (Hall et al., 1998).

Zahrouni and Kamoun (2012) developed a constructive heuristic inspired by the NEH algorithm of Nawaz et al. (1983) for the three-machine case. Batur et al. (2012) formulated the two-machine problem in which the processing times are also decision variables as a variation of the TSP. Since the model size increases drastically with respect to m, extending this formulation to the m-machine case is not practical. Fazel Zarandi et al. (2013) developed a branch and bound algorithm and proposed an SA algorithm for the two-machine case that includes sequence-dependent setup times. Che et al. (2010) considered the cyclic hoist scheduling with multiple parts, fixed processing times, and no-wait constraints. They developed an MILP formulation which makes use of the no-wait constraints and developed a dynamic branch and bound procedure. Lei et al. (2014) considered the cyclic scheduling of multiple parts and the robot where the process times on the machines must satisfy time-window constraints. That is, the actual process time of part j on machine i must be within the interval $[a_{ij}, b_{ij}]$. Note that, if b_{ij} is set to ∞, time window constraints become identical with the current study. For the problem, the authors developed an MILP formulation with the assumption that the part sequence is given. They made use of this formulation to develop a Branch and Bound (B&B) procedure where all possible part sequences are enumerated to determine the optimal MPS schedule. This procedure reduces the solution time for this complex problem. However, it can only solve instances with a smaller number of machines and parts in reasonable times. Test results with only 2 and 3 different parts are performed in that study where 2 parts and 18 machines take 3.8 seconds whereas 3 parts and 18 machines take 2242.6 seconds on a machine with 3.0 GHZ Pentium IV processor. This drastic increase in the solution times