Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-12 (2016) 215–220

Good Production Cycles for Circular Robotic Cells

Florence Thiard, Nicolas Catusse, Nadia Brauner

Univ. Grenoble Alpes, G-SCOP, F-38000 Grenoble, France
CNRS, G-SCOP, F-38000 Grenoble, France
e-mail: {florence.thiard, nicolas.catusse, nadia.brauner}@g-scop.grenoble-inp.fr

Abstract:
In this paper, we study cyclic production for throughput optimization in robotic flow-shops. We are focusing on simple production cycles. Robotic cells can have a linear or a circular layout: most classical results on linear cells cannot be extended to circular cells, making it difficult to quantify the potential gain brought by the latter configuration. Moreover, though the problem of finding the best one part production cycle is polynomial for linear cells, it is NP-hard for circular cells.
We consider the special case of circular balanced cells. We first consider three basic production cycles, and focus on one which is specific to circular cells, for which we establish the expression of the cycle time. Then, we provide a counter-example to a classical conjecture still open in this configuration. Finally, based on computational experiments, we make a conjecture on the dominance of a family of cycle, which could lead to a polynomial algorithm for finding the best 1-cycle for circular balanced cells.

Keywords: scheduling algorithms, throughput, robotic flow-shop, identical parts, balanced cell, circular layout

1. POSITIONING OF THE PROBLEM

Robotic cells consist in a flowshop where the machines are served by a robot. Present in many industries, they are frequently used in semi-conductor manufacturing and electroplating (Dawande et al., 2007). The model was first introduced by Asfahl (1985) to describe a production cell for truck differentials, while Sethi et al. (1992) provided the first formal study for small dimension cells. The robotic flow-shop problem for the production of multiple part types has been proven NP-complete for 3 machines by Hall et al. (1998).

As robotic cells constitute an adequate environment for large-scale production of a few different types of products, the throughput – the number of part produced per time unit – is a natural measure for their performance.Dawande et al. (2007) present a survey on throughput optimization in robotic cells.

1.1 Notations and problem specification

Formally, a robotic cell consists of m machines, denoted by $M_1, M_2, ..., M_m$, and a robot in charge of the handling of the parts in-between machines. The cell is also equipped with an input buffer, which provides the parts to be produced in infinite quantity, and an output buffer, also of infinite capacity. These buffers are modeled by two additional machines, respectively M_0 and M_{m+1}. As in a classical flow-shop, all parts must be processed successively on machines $M_1 \dots M_m$ in that order.

The input of the problem consists of travel times, processing times and loading/unloading times. In the general case, travel times and processing times are machine-dependent: $\delta_{i,j}$ denotes the travel time between machines M_i and M_j while $p_{i,j}$ represents the processing time of a part j on machine i. Loading and unloading times are generally assumed identical and denoted by ϵ.

However, if the robot travels at a constant speed, with no acceleration in-between machines, then the only information needed for travel times is the time between two consecutive machines δ_i. In this case, the travel times are called additive: this is a fairly common assumption. Additionally, if the time between any two consecutive machines is the same (if the machines are regularly disposed), the cell is called regular. In cases where the cell is used to group operations of similar length, it is relevant to consider machine-independent processing times. The cell is then called balanced.

Note that for a regular balanced cell producing one type of part, the problem input consists of only 4 numbers, m, δ, p, ϵ.

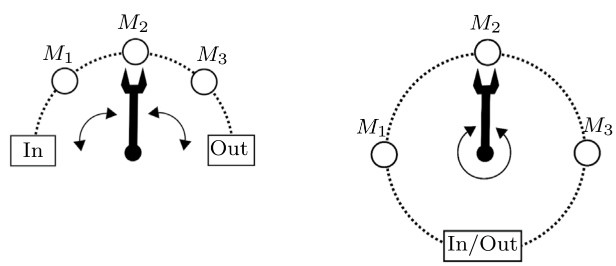
Depending on the type of robot used, the machines and the input and output stations can be disposed in several ways. Two main configurations are studied in the literature: on the one hand, linear or semi-circular layouts (Figure 1(a)), where the input and output buffers are separated and located respectively at each end of the line (Crama and van de Klundert, 1997), and on the other hand, circular layouts (Figure 1(b)), where the machines are arranged in

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.599

a circle, with the input and output buffers either occupying the same spot ($M_0 = M_{m+1}$), or very close (Rajapakshe et al., 2011; Jung et al., 2015).

This paper focuses on the classical robotic cell model, which means that the cell is served by a single robot which can hold a single part at a time, and the machines are bufferless: as a consequence, a machine cannot be loaded with a new part until processing is finished and the part transferred onto the next machine.

We will also assume that parts may stay on a given machine as long as necessary after processing is finished (this is referred to as unbounded waiting-times or free-pickup criterion). Other policies include no-wait (the part must be retrieved as soon as the processing is finished), and interval (waiting times are bounded), also called Hoist Scheduling Problems (HSP).



(a) semi-circular layout (b) circular layout

Fig. 1. Three-machine robotic cells

1.2 Identical parts production

In the general case, where multiple part types must be produced, two types of decisions must be made: sequencing the parts, and scheduling the robot moves. In the case where only one type of part is to be produced, the part sequencing is of course trivial: the problem reduces to finding an optimal robot move schedule. Brauner (2008) presents a survey on single part-type production in robotic cells.

In this case, robot move sequences can be described using the concept of activities, introduced by Crama and van de Klundert (1999). Activities are elementary robot moves defined as such: for $i \in \{0 \dots m\}$, activity A_i refers to the following sequence of events:

(1) The robot unloads a part from M_i;
(2) The robot travels to M_{i+1};
(3) The robot loads the part onto M_{i+1}.

1.3 Cyclic programmation

For large-scale production, it is operationally relevant to prefer a cyclic programmation. This means that the robot repeats indefinitely the same move sequence, each iteration leaving the cell in the same state, with the same machines loaded and the same machines empty. Moreover, cyclic programmations are dominant (Dawande et al., 2005b), which means that for any set of parameters, there always exists an optimal programmation which is cyclic. The elementary sequence is called a cycle.

One-cycles are particular cycles which produce one part exactly: during one iteration, exactly one part enters the cell at M_0, and one processed part leaves the cell. More generally, a k-cycle is a cycle of production of k parts. One-cycles are easy to describe and enumerate using the concept of activities, as they are exactly the permutations of the $m+1$ activities (Crama and van de Klundert, 1999). They are also easier to implement operationally.

As a consequence, it is convenient to restrict the possible move sequences to 1-cycles only. But does this allow to find an optimal sequence? Sethi et al. (1992) formulate the 1-cycle conjecture:

Conjecture 1. (1-cycle conjecture Sethi et al. (1992)).
The set of 1-cycles is dominant (for any set of parameters, there always exists a 1-cycle which is optimal).

Unfortunately, this conjecture has been proven false on the general case for more than 4 machines (Dawande et al., 2005a; Brauner and Finke, 2001), meaning that 1-cycles are not generally optimal. However, it is interesting to consider their performance compared to general cycles and their *dominance for special cases of robotic cells*, as well as *finding the best 1-cycle*. In this paper, we are interested in the last two problems.

1.4 Impact of the layout

The answers to these questions depend on the cell layout. In fact, although requiring more sophisticated robots, circular layout can improve the travel time, as the robot takes the shortest path around the cell. For example, in a regular balanced cell, travel time between machine M_i and M_j is $\delta_{i,j} = |i-j|\delta$, while on a similar cell with circular layout, it is $\delta_{i,j} = \min(|i-j|, m+1-|i-j|)\delta$.

In order to make a decision regarding the layout of the cell or quantify the potential gain of a circular layout, it is necessary to study the best programmation for either layout. However, dominant sequences for linear cells might not be dominant with a circular layout (Geismar et al., 2005). Most studies on circular cells consider models which relax the blocking constraints one way or another: robot with swapping ability (Jolai et al., 2012), dual-gripper robots (Sethi et al., 2001; Jung et al., 2015; Drobouchevitch et al., 2006), or machine buffers (Drobouchevitch et al., 2010). On the contrary, circular classical single gripper cells, studied by (Geismar et al., 2005; Rajapakshe et al., 2011; Jung et al., 2015) are not as well understood yet as their linear counterparts.

1-cycle conjecture The 1-cycle conjecture is valid for 2-machine cells regardless of the layout (Sethi et al., 1992). For linear layouts, it is valid for 3-machine cells (Crama and van de Klundert, 1997), and false for 4-machine cells (Brauner and Finke, 2001). For the special case of regular balanced cell, it is valid up to 15 machines (Brauner, 2008). However, for circular cells with more than 2 machines, it is still open, even for the regular balanced case.

In Section 3 we provide a counter-example to the 1-cycle conjecture for 6-machine regular balanced cells.

Best 1-cycle problem Finding the best 1-cycle is polynomial in linear additive cells: Crama and van de Klundert