Productivity Gains in Flexible Robotic Cells

H. Neil Geismar *

Suresh P. Sethi *

Jeffrey B. Sidney †

Chelliah Sriskandarajah *

* University of Texas at Dallas, USA, {ngeismar,sethi,chelliah}@utdallas.edu

† University of Ottawa, Canada, sidney@management.uottawa.ca

March 27, 2004

Abstract

Flexible robotic cells combine the capabilities of robotic flow shops with those of flexible manufacturing systems. In an m-machine flexible cell, each part visits each machine in the same order. However, the m operations can be performed in any order, and each machine can be configured to perform any operation. We derive the maximum percentage increase in throughput that can be achieved by changing the assignment of operations to machines. We find that no increase can be gained in two-machine cells, and the gain in three- and four-machine cells each is at most $14\frac{2}{7}\%$. For this calculation, the assignment remains constant throughout a lot's processing. Additionally, we determine the throughput increase that can be gained by varying the assignment of operations to machines for successive parts within a processing lot.

Key Words and Phrases: flexible robotic cells, robotic open shop, scheduling, sequencing.

1 Introduction

Robots are used in a wide range of applications in manufacturing companies (Asfahl 1985, Miller and Walker 1990). One important application of robots in manufacturing is their use for material handling in robotic cells. In such a cell, the robot is located at the approximate center of the workcell, and a number of machines (M_1, M_2, $\ldots$, M_m) and an input/output (I/O) hopper are arranged around it. A real-world example of a three-machine robotic cell is given in Asfahl (1985). In this example, a robotic cell processes castings for truck differential assemblies. The cell includes a drilling machine (operation: cross pin drill), a boring machine (operation: bore pinion holes), another drilling machine (operation: ear hole drill), and an input/output hopper. Although these operations can be done any order, for operational convenience the cell operates as a flow shop in which the robot moves the parts from I/O to machine M_1, M_2, M_3, and finally to I/O. There are many manufacturing environments such as the above example that can operate as an open shop: the order of the operations is immaterial.

We consider a circularly-configured robotic cell in which each job (part) visits the machines in the following sequence: (I/O, M_1, M_2,$\ldots$, M_m, I/O). Each job requires m operations $(o_1, o_2, \ldots , o_m)$ which can be performed in any order. Furthermore, we assume that an operation o_i requiring processing time of p_i can be assigned to any of the m machines by relatively fast and inexpensive tool changes. This assumption is valid in the current modern manufacturing environment because the use of flexible manufacturing machines which have quick tool-changing capability is growing in industry. We shall refer to such a system as a *flexible robotic cell*.

A two-machine robotic cell is illustrated in Figure 1. The machines are served by a central robot. The robot arm rotates and moves to handle inter-machine movements of parts. A part is picked up at the hopper (I/O), processed once on each machine, and finally dropped at the hopper (I/O), i.e., after operations have been assigned to machines, the cell operates as a flow shop. The processing of any part on a machine is nonpreemptive. Each machine can process at most one part at a time and has neither an input nor an output buffer. Thus, any part in the cell is always either on one of the machines, at I/O, or being handled by the robot. Moreover, even if a part has completed processing on a machine, no other part can be loaded onto that

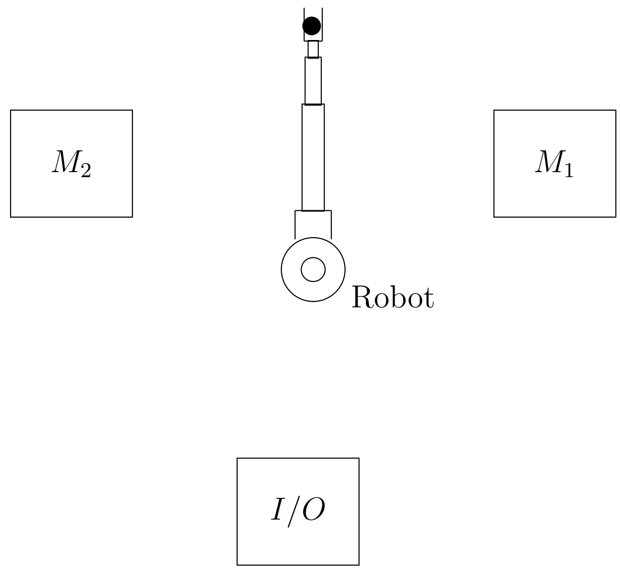

Figure 1: A Two Machine Robotic Cell.

machine until the robot has removed that part from the machine (this is a *blocking* condition). After loading a part onto a machine, the robot either waits at that machine for the part to finish processing, moves to another machine to unload a different part (when that part finishes processing on that machine), or moves to the I/O hopper to pick up a new part.

At any given instant of time, the state of the system is specified by the list of parts that are in process, where each of these parts is residing (either on the robot or on a machine), exactly how much work remains to be done at each machine for each part currently in progress, and the location of the robot. A cyclic schedule is one in which the same sequence of states is repeated over and over again (with the obvious proviso that the specific jobs in the system are changing). A cycle in such a schedule begins at any given state and ends when that state is encountered next. In each cycle of a cyclic schedule, one or more parts will be completed. If k parts are produced in a cycle, we call the cycle a k-unit cycle. In this paper we confine our discussion to cyclical schedules which are one-unit cycles, and we shall be interested in the steady state operations of the system under various cyclic scheduling schemes.

The robotic cells in this study belong to the class of *flexible manufacturing systems (FMS)*. An FMS contains computer numerically controlled machine tools that are connected by an automated material handling system and are controlled by a central computer (Stecke 1989, Askin and Standridge 1995). What distinguishes a flexible robotic cell from other robotic cells is that the machines are automatically reprogramable to perform different operations, and such a change in a machine's operation can be performed quickly. This capability is called *machine flexibility*. Furthermore, because the m fixed operations can be performed on each part in any order, they have *operation flexibility* (Sethi and Sethi 1990).

We investigate the productivity gains in flexible robotic cells by comparing their throughputs for various assignments of operations to machines. Since all parts produced are identical, for a given assignment of operations to machines, we need only to determine the sequence of moves performed by the robot. The objective is the maximization of the steady state throughput.

The degree to which the throughput can be improved by allowing the robotic cell to be operated as a flexible manufacturing system with the ability to assign operations to machines will be investigated under two different scenarios. In the first, the assignment of operations to machines, once chosen, remains fixed throughout a lot's processing. For example, if $m = 3$, then the operations could be processed in the order (o_2, o_3, o_1) in a three-machine flow shop where each job is first processed on M_1 (operation o_2), then on M_2 (operation o_3), and finally on M_3 (operation o_1). In the second scenario we allow *on-line tool changes*. In this case, each machine can change its tooling quickly so that a particular machine can perform different operations on successive parts without causing a delay. For example, the first part is processed in order (o_1, o_2, o_3), and the second part is processed in order (o_2, o_3, o_1), even though each visits the machines in the same order. In both cases, we assume that any of the six permutations of the operations of a job is feasible with respect to the processing order. It is obvious that there is potential for improvement of throughput if all permutations are possible.

The remainder of this paper is organized as follows. In Section 2 we provide a brief literature review. In Section 3 we specify notation for robotic cell scheduling problems. In Section 4 we define one-unit cycles and show how to derive their cycle time formulas for robotic flow shop cells. In Section 5 we examine the throughput gains that can be achieved in a flexible robotic cell by choosing an assignment of operations to machines that remains fixed. In Section 6 we discuss throughput gains achieved by assignment changes in flexible robotic cells that use on-line tool changes. Finally, Section 7 concludes the study and provides recommendations for future research.

2 Literature Review

There have been several studies devoted to the problem of scheduling in robotic flow shop cells. Detailed reviews of the literature can be found in surveys by Crama et al. (2000) and Dawande et al. (2002b). The performance of a robotic cell depends directly on the sequence of robot moves. As a result, finding an efficient robot move sequence has recently attracted considerable research attention.

Sethi et al. (1992) consider the problem of minimizing the cycle time in robotic flow shop cells with $m = 2$ and $m = 3$ that produce a single part-type. For two-machine cells, the optimal solution is given by a simple cycle producing one unit. For three-machine cells, they identify six one-unit cycles and conditions under which each is optimal in the class of one-unit cycles.

Sethi et al. (1992) also prove that a one-unit cyclic solution is optimal over the class of all solutions, cyclic or otherwise, in a two-machine cell. In the three-machine case, Crama and van de Klundert (1999) and Brauner and Finke (1999) show that the best one-unit cycle is optimal among the class of all cyclic solutions. Brauner and Finke (1997, 2001) show that in m-machine cells (for $m \geq 4$), the conjecture is not true.

One-unit cycles are attractive from a practical point of view because of their conceptual simplicity and their ease of implementation. Crama and van de Klundert (1997) provide a polynomial-time dynamic programming algorithm for minimizing cycle time over all one-unit cycles in an m-machine cell producing a single part-type. The complexity of the problem for k-unit cycles remains an open question for $k \geq 2$.

A thorough survey of the literature concerning flexible manufacturing systems can be found in Sethi and Sethi (1990). In this article that expands on the work of Browne et al. (1984), they define different types of flexibility, e.g., process, routing, product, volume, expansion, material handling, machine, and operation. Stecke (1985) defines and describes design, planning, scheduling, and control problems for FMS. Reviews of scheduling applications in FMS include Harmonosky and Robohn (1991), Basnet and Mize (1994), and Rachamadugu and Stecke (1994).

3 Notation

The following notation used to describe a robotic cell is similar to that in Sethi et al. (1992):

$M_1, \ldots, M_m$: the machines in the robotic cell in the processing order.

I/O : the input/output hopper, also called M_0 or M_{m+1} (and referred to as a machine).

$o_1, \ldots, o_m$: the operations required for a part.

p_j : the processing time of operation o_j.

σ : permutation of operations $o_1, \ldots, o_m$.

vector $(o_{\sigma(1)}, \ldots, o_{\sigma(m)})$: order of processing the operations, where $o_{\sigma(i)}$ is the i^{th} operation, and it is performed on machine M_i.

δ : the time taken by a rotational robot movement when traveling between two consecutive machines M_{j-1} and M_j, $1 \leq j \leq m+1$, where both M_0 and M_{m+1} mean I/O.

ϵ : the time taken by the robot to pick up or drop off a part at I/O, or the time taken by the robot to load or unload a part at any machine.

$E = (\chi_1, \ldots, \chi_m, M_h)$: the current state of the system, where $\chi_i = \phi$ (respectively, Ω) if machine M_i is free (resp., occupied by a part), and the robot has just loaded machine M_h, for $1 \leq h \leq m+1$.

$S_{i,m}$: robot move cycle i for a cell having m machines.

T_i: the cycle time for robot move cycle $S_{i,m}$ (it will be apparent from context how many machines are in the cell being discussed).

The robot travel time between locations x and y is denoted by the symmetric function $\ell(x, y)$. For example, in a two-machine robotic cell, $\ell(I/O, M_1) = \delta$, $\ell(M_1, M_2) = \delta$, and $\ell(M_2, I/O) = \delta$. In a three-machine robotic cell, $\ell(I/O, M_1) = \delta$, $\ell(M_1, M_2) = \delta$, $\ell(M_2, M_3) = \delta$, $\ell(M_3, I/O) = \delta$, $\ell(I/O, M_2) = 2\delta$, and $\ell(M_1, M_3) = 2\delta$.

The standard classification scheme for scheduling problems (Graham et al., 1979) as updated for robotic cells by Dawande et al. (2002b) denotes the robotic flow shop scheduling problem by $RF_m|(blocking,A,cyclic\text{-}1)|C_t$. The three fields indicate the scheduling environment (RF_m: m-machine robotic flow shop), restrictive requirements (*blocking,A,cyclic*-1: the cell has blocking, additive travel-time, and we seek one-unit cyclic solutions), and the objective function to be minimized (C_t : the per unit cycle time under a steady state repetitive manufacture of parts).

For flexible robotic cells, the classification scheme uses a different value in the first field. Because we consider two different problems for flexible robotic cells, we have two notations. The scheduling problem for flexible robotic cells in which the assignment of operations to machines remains fixed throughout a lot's processing is denoted by $FRC_m|(blocking,A,cyclic\text{-}1)|C_t$. The scheduling problem for a flexible robotic cell using on-line tool changes is denoted by $OFRC_m|(blocking,A,cyclic\text{-}1)|C_t$.

4 One-unit Cycles

The concept of *activity* is very useful in the study of robotic cells. Activity A_i, $i = 0, \ldots, m$, consists of the following sequence:

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

The activity sequence (A_i, A_k) implies that after completing activity A_i by loading machine M_{i+1}, the robot travels to machine k to begin activity A_k. Note that when using activity notation, we can never instruct a robot that is currently holding a part to unload a machine. Similarly, we can never instruct an empty robot to load a machine.

Define the function $F(A_i, t)$ to represent the time of completion of the t^{th} execution of any activity A_i, for fixed i. Given a feasible infinite sequence of activities and a compatible initial state, we can define the *long-run average throughput*, or simply *throughput*, of that sequence to be

$$\mu = \lim_{j \to \infty} \frac{t}{F(A_m, t)}.$$

Intuitively, this quantity represents the long-term average number of completed parts placed into the input/output hopper per unit time (Crama and van de Klundert 1997).

Obtaining a feasible infinite sequence of activities that maximizes throughput is a fundamental problem of robotic cell scheduling. Such a sequence of robotic moves is called *optimal*. Most studies focus on infinite sequences of activities in which a sequence of $m + 1$, or some integral multiple of $m + 1$, activities is repeated cyclically.

The study of cyclic production is motivated by its prevalence in industrial implementations. Additionally, Dawande et al. (2002a) show that it is sufficient to consider only cyclic solutions in order to maximize throughput. Cyclic production employs a repeatable sequence of activities. For example, $(A_0, A_2, A_4, A_3, A_1)$ is a sequence of activities that produces a part in a four-machine cell. Such a sequence can be repeated, with each repetition producing a single part. Since a part must be processed on all m machines and then placed into the input/output hopper, $m+1$ different activities (exactly one of each of the $m+1$ activities $A_0, A_1, \ldots, A_m$) are required to produce a part. More precisely, we have the following definition:

Definition: A *k-unit cycle* is the performance of a feasible sequence of robot moves which loads and unloads each machine exactly k times in a way which leaves the cell in exactly the same state as its state at the beginning of those moves.

To be *feasible*, a sequence of activities must satisfy two criteria:

- The robot cannot be instructed to load an occupied machine.
- The robot cannot be instructed to unload an unoccupied machine.

All one-unit cycles are feasible.

A description of the state of the robotic cell at any given instant of time includes where the robot is, where the semifinished parts are, and to what extent each part has been processed. Such a precise mathematical statement of the state is not required in this paper, because it deals only with a steady state analysis.

In a k-unit cycle, let A_i^j denote the j^{th} instance of activity A_i, $i = 1, \ldots, m$; $j = 1, \ldots, k$.

Definitions: (Crama and van de Klundert 1997) A robotic cell repeatedly executing a k-unit cycle π of robot moves is operating in *steady state* if there exist constants $T(\pi)$ and N such that for every A_i^j, $i = 0, \ldots, m$, and for every $t \in \mathbb{Z}^+$ such that $t > N$, $F(A_i^j, t+1) - F(A_i^j, t) = T(\pi)$. $T(\pi)$ is called the *cycle time* of π.

The per unit cycle time of a k-unit cycle π is $T(\pi)/k$. This is the reciprocal of the throughput and is easier to calculate directly. Therefore, rather than maximizing throughput, we minimize per unit cycle time. In this study we consider only one-unit cycles. Obviously, in this case the cycle time equals the per unit cycle time.

Brauner and Finke (2001) show that repeating a k-unit activity sequence will enable the robotic cell to reach a steady state (or cyclic solution) in finite time. Therefore, since we are maximizing the long-run average throughput, i.e., assuming that the cells operate in steady state for an infinite time, there is no contribution from the initial transient phase. Hence, there is no loss of generality by studying only the steady state behavior.

4.1 Two-machine Robotic Cells

It has been proven (Sethi et al. 1992) that in an m machine robotic cell there are $m!$ one-unit cycles, corresponding to the $m!$ permutations of $\{A_1, \ldots, A_m\}$. Thus, the two robot move cycles in a two-machine robotic cell are $S_{1,2} = (A_0, A_1, A_2)$ and $S_{2,2} = (A_0, A_2, A_1)$. We now derive the cycle times for producing parts in a robotic flow shop ($RF_2|(blocking, A, cyclic\text{-}1)|C_t$) using these two cycles. These results will be directly applicable to larger cells and to flexible robotic cells.

For $S_{1,2}$, if we start from the initial state $E = (\phi, \Omega, M_2)$, where the robot has just loaded part P_i onto M_2 and M_1 is free, the robot move cycle includes the following activities: wait until P_i is processed: (p_2), unload P_i from M_2: (ϵ) move to I/O: (δ), drop P_i at I/O: (ϵ), pick up P_{i+1} at I/O: (ϵ), move to M_1: (δ), load P_{i+1} on M_1: (ϵ), wait until P_{i+1} is processed: (p_1), unload P_{i+1} from M_1: (ϵ), move to M_2: (δ), and load P_{i+1} on M_2: (ϵ). Thus,

$$T_1 = 3\delta + 6\epsilon + p_1 + p_2. \tag{1}$$

For $S_{2,2}$, if we start from the initial state $E = (\Omega, \Omega, M_1)$, where the robot has just loaded part P_{i+1} onto M_1 and M_2 is occupied by part P_i, the robot move sequence includes the following activities: move to M_2: (δ), if necessary wait until P_i is processed at M_2: (w_2), unload P_i from M_2: (ϵ), move to I/O: (δ), drop P_i at I/O: (ϵ), move to M_1: (δ), if necessary wait until P_{i+1} is processed at M_1: (w_1), unload P_{i+1} from M_1: (ϵ), move to M_2: (δ), load P_{i+1} on M_2: (ϵ), move to I/O: (δ), pick up part P_{i+2} at I/O: (ϵ), move to M_1: (δ), load P_{i+2} on M_1: (ϵ). Therefore, $T_2 = 6\delta + 6\epsilon + w_1 + w_2$, where

$w_1 = \max\{0, p_1 - w_2 - 3\delta - 2\epsilon\}$, and

$w_2 = \max\{0, p_2 - 3\delta - 2\epsilon\}$.

By combining the expressions for w_1 and w_2, we obtain the following result:

$$T_2 = \max\{6\delta + 6\epsilon, p_1 + 3\delta + 4\epsilon, p_2 + 3\delta + 4\epsilon\}. \tag{2}$$

Lemma 1 *In $RF_2|(blocking, A, cyclic\text{-}1)|C_t$, cycle $S_{1,2}$ is optimal if $\delta \geq (p_1 + p_2)/3$, whereas cycle $S_{2,2}$ is optimal if $\delta \leq (p_1 + p_2)/3$.*

Proof: Follows from equations (1) and (2). See also Sethi et al. (1992). ∎

4.2 Three-machine Robotic Cells

In a three-machine robotic cell, the six one-unit cycles are

$$S_{1,3} = (A_0, A_1, A_2, A_3), \qquad S_{2,3} = (A_0, A_2, A_1, A_3),$$
$$S_{3,3} = (A_0, A_1, A_3, A_2), \qquad S_{4,3} = (A_0, A_3, A_1, A_2),$$
$$S_{5,3} = (A_0, A_2, A_3, A_1), \qquad S_{6,3} = (A_0, A_3, A_2, A_1)$$

Their cycle times are presented in the following lemma.

Lemma 2 *For problem $RF_3|(blocking, A, cyclic\text{-}1)|C_t$, the cycle times of the six one-unit cycles $(S_{1,3}, \ldots, S_{6,3})$ are given by:*

$T_1 = 4\delta + 8\epsilon + p_1 + p_2 + p_3$

$T_2 = \max\{8\delta + 8\epsilon, 4\delta + 6\epsilon + p_1, 4\delta + 4\epsilon + p_2, 4\delta + 6\epsilon + p_3, 2\delta + 4\epsilon + \frac{p_1+p_2+p_3}{2}\}$

$T_3 = \max\{8\delta + 8\epsilon + p_1, 4\delta + 6\epsilon + p_1 + p_2, 4\delta + 4\epsilon + p_3\}$

$T_4 = \max\{8\delta + 8\epsilon + p_2, 4\delta + 6\epsilon + p_2 + p_3, 4\delta + 6\epsilon + p_1 + p_2\}$

$T_5 = \max\{8\delta + 8\epsilon + p_3, 4\delta + 6\epsilon + p_2 + p_3, 4\delta + 4\epsilon + p_1\}$

$T_6 = \max\{12\delta + 8\epsilon, 4\delta + 4\epsilon + p_1, 4\delta + 4\epsilon + p_2, 4\delta + 4\epsilon + p_3\}$.

Proof: Similar to the derivation of equations (1) and (2) and Sethi et al. (1992). ∎

4.3 Four-machine Robotic Cells

In a four-machine robotic cell, the twenty-four one-unit cycles are

$$S_{1,4} = (A_0, A_1, A_2, A_3, A_4), \qquad S_{2,4} = (A_0, A_1, A_3, A_2, A_4),$$
$$S_{3,4} = (A_0, A_2, A_1, A_3, A_4), \qquad S_{4,4} = (A_0, A_2, A_3, A_1, A_4),$$
$$S_{5,4} = (A_0, A_3, A_1, A_2, A_4), \qquad S_{6,4} = (A_0, A_3, A_2, A_1, A_4),$$
$$S_{7,4} = (A_0, A_2, A_3, A_4, A_1), \qquad S_{8,4} = (A_0, A_3, A_2, A_4, A_1),$$
$$S_{9,4} = (A_0, A_3, A_4, A_1, A_2), \qquad S_{10,4} = (A_0, A_4, A_1, A_2, A_3),$$
$$S_{11,4} = (A_0, A_2, A_4, A_1, A_3), \qquad S_{12,4} = (A_0, A_4, A_1, A_3, A_2),$$
$$S_{13,4} = (A_0, A_1, A_3, A_4, A_2), \qquad S_{14,4} = (A_0, A_3, A_1, A_4, A_2),$$
$$S_{15,4} = (A_0, A_3, A_4, A_2, A_1), \qquad S_{16,4} = (A_0, A_4, A_2, A_1, A_3),$$

$$S_{17,4} = (A_0, A_1, A_4, A_2, A_3), \qquad S_{18,4} = (A_0, A_4, A_2, A_3, A_1),$$
$$S_{19,4} = (A_0, A_1, A_2, A_4, A_3), \qquad S_{20,4} = (A_0, A_2, A_1, A_4, A_3),$$
$$S_{21,4} = (A_0, A_2, A_4, A_3, A_1), \qquad S_{22,4} = (A_0, A_4, A_3, A_1, A_2),$$
$$S_{23,4} = (A_0, A_1, A_4, A_3, A_2), \qquad S_{24,4} = (A_0, A_4, A_3, A_2, A_1)$$

The cycle times for these cycles are presented in the following lemma. Note that in a four-machine cell, when the robot moves from M_4 to M_1 (or vice versa), it travels via I/O (requiring time 2δ) rather than via M_2 and M_3 (which would require time 3δ).

Lemma 3 *For problem $RF_4|(blocking,A,cyclic\text{-}1)|C_t$, the cycle times of the twenty-four one-unit* *cycles are given by:*

$$
\begin{aligned}
T_1 &= 5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4 \\
T_2 &= \max\{9\delta + 10\epsilon + p_1, 4\delta + 4\epsilon + p_3, 5\delta + 8\epsilon + p_2 + p_1, 5\delta + 8\epsilon + p_4 + p_1, \\
&\qquad (5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4)/2\} \\
T_3 &= \max\{9\delta + 10\epsilon + p_4, 4\delta + 4\epsilon + p_2, 5\delta + 8\epsilon + p_1 + p_4, 5\delta + 8\epsilon + p_3 + p_4, \\
&\qquad (5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4)/2\} \\
T_4 &= \max\{10\delta + 10\epsilon + p_3, 5\delta + 6\epsilon + p_2 + p_3, 5\delta + 6\epsilon + p_1, 5\delta + 8\epsilon + p_4 + p_3, \\
&\qquad (5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4)/2\} \\
T_5 &= \max\{10\delta + 10\epsilon + p_2, 5\delta + 6\epsilon + p_3 + p_2, 5\delta + 8\epsilon + p_1 + p_2, 5\delta + 6\epsilon + p_4, \\
&\qquad (5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4)/2\} \\
T_6 &= \max\{13\delta + 10\epsilon, 4\delta + 4\epsilon + p_3, 4\delta + 4\epsilon + p_2, 5\delta + 6\epsilon + p_1, 5\delta + 6\epsilon + p_4, \\
&\qquad (5\delta + 10\epsilon + p_1 + p_2 + p_3 + p_4)/3\} \\
T_7 &= \max\{9\delta + 10\epsilon + p_3 + p_4, 5\delta + 8\epsilon + p_2 + p_3 + p_4, 4\delta + 4\epsilon + p_1\} \\
T_8 &= \max\{13\delta + 10\epsilon, 4\delta + 4\epsilon + p_3, 5\delta + 6\epsilon + p_2, 9\delta + 8\epsilon + p_4, 4\delta + 4\epsilon + p_1\} \\
T_9 &= \max\{10\delta + 10\epsilon + p_4 + p_2, 5\delta + 8\epsilon + p_3 + p_4 + p_2, 5\delta + 6\epsilon + p_1 + p_2\} \\
T_{10} &= \max\{9\delta + 10\epsilon + p_2 + p_3, 5\delta + 8\epsilon + p_4 + p_2 + p_3, 5\delta + 8\epsilon + p_1 + p_2 + p_3\}
\end{aligned}
$$

$$
\begin{aligned}
T_{11} &= \max\{10\delta + 10\epsilon, 5\delta + 6\epsilon + p_2, 5\delta + 6\epsilon + p_4, 5\delta + 6\epsilon + p_1, 5\delta + 6\epsilon + p_3\} \\
T_{12} &= \max\{13\delta + 10\epsilon, 5\delta + 6\epsilon + p_4, 9\delta + 8\epsilon + p_1, 4\delta + 4\epsilon + p_3, 9\delta + 8\epsilon + p_2, 5\delta + 6\epsilon + p_1 + p_2\} \\
T_{13} &= \max\{10\delta + 10\epsilon + p_1 + p_4, 5\delta + 6\epsilon + p_3 + p_4, 5\delta + 6\epsilon + p_2 + p_1\} \\
T_{14} &= \max\{15\delta + 10\epsilon, 10\delta + 8\epsilon + p_3, 10\delta + 8\epsilon + p_1, 10\delta + 8\epsilon + p_4, 10\delta + 8\epsilon + p_2, 5\delta + 6\epsilon + p_3 + p_4, \\
&\qquad 5\delta + 6\epsilon + p_3 + p_2, 5\delta + 6\epsilon + p_1 + p_2\} \\
T_{15} &= \max\{13\delta + 10\epsilon + p_4, 5\delta + 6\epsilon + p_3 + p_4, 4\delta + 4\epsilon + p_2, 4\delta + 4\epsilon + p_1\} \\
T_{16} &= \max\{13\delta + 10\epsilon, 9\delta + 8\epsilon + p_4, 4\delta + 4\epsilon + p_2, 5\delta + 6\epsilon + p_1, 9\delta + 8\epsilon + p_3, 5\delta + 6\epsilon + p_4 + p_3\} \\
T_{17} &= \max\{10\delta + 10\epsilon + p_1 + p_3, 5\delta + 6\epsilon + p_4 + p_3, 5\delta + 8\epsilon + p_2 + p_1 + p_3\} \\
T_{18} &= \max\{13\delta + 10\epsilon + p_3, 5\delta + 6\epsilon + p_4 + p_3, 5\delta + 6\epsilon + p_2 + p_3, 4\delta + 4\epsilon + p_1\} \\
T_{19} &= \max\{9\delta + 10\epsilon + p_1 + p_2, 4\delta + 4\epsilon + p_4, 5\delta + 8\epsilon + p_3 + p_1 + p_2\} \\
T_{20} &= \max\{13\delta + 10\epsilon, 4\delta + 4\epsilon + p_2, 9\delta + 8\epsilon + p_1, 4\delta + 4\epsilon + p_4, 5\delta + 6\epsilon + p_3\} \\
T_{21} &= \max\{13\delta + 10\epsilon, 9\delta + 8\epsilon + p_2, 4\delta + 4\epsilon + p_4, 9\delta + 8\epsilon + p_3, 4\delta + 4\epsilon + p_1, 5\delta + 6\epsilon + p_2 + p_3\} \\
T_{22} &= \max\{13\delta + 10\epsilon + p_2, 4\delta + 4\epsilon + p_4, 5\delta + 6\epsilon + p_3 + p_2, 5\delta + 6\epsilon + p_1 + p_2\} \\
T_{23} &= \max\{13\delta + 10\epsilon + p_1, 4\delta + 4\epsilon + p_4, 4\delta + 4\epsilon + p_3, 5\delta + 6\epsilon + p_2 + p_1\} \\
T_{24} &= \max\{15\delta + 10\epsilon, 4\delta + 4\epsilon + p_4, 4\delta + 4\epsilon + p_3, 4\delta + 4\epsilon + p_2, 4\delta + 4\epsilon + p_1\}
\end{aligned}
$$

Proof: Similar to the derivation of equations (1) and (2) and Sethi et al. (1992). ■

5 Flexible Robotic Cells: Throughput Comparison

We examine the throughput gains in flexible robotic cells for all possible assignments of operations to machines. In this section the assignment of operations to machines remains fixed throughout a lot's processing (problem $FRC_m|(blocking,A,cyclic\text{-}1)|C_t$ for $m = 2, 3, 4$).

This particular version of the throughput problem and the one of Section 6 are based upon two assumptions:

1. Any of the $m!$ processing orders for the operations of a part in an m-machine cell is feasible.

2. The machines are converted according to the specified order of operations. For example, if $m = 3$ and the operations are processed in the order (o_3, o_1, o_2), then o_3 is processed for p_3 time on M_1, o_1 is processed for p_1 time on M_2, and o_2 is processed for p_2 time on M_3.

5.1 Comparison of Performance: Two-Machine Robotic Cells

We now show that in problem $FRC_2|(blocking,A,cyclic\text{-}1)|C_t$ the flexibility to assign operations to machines does not allow for an increase in throughput. We will denote a permutation of operations by a 2-vector of the form (o_i, o_j). This notation indicates that operations o_i and o_j are done on machines M_1 and M_2, respectively.

Lemma 4 *The flexibility to assign operations to machines in a two-machine flexible robotic cell ($FRC_2|(blocking,A,cyclic\text{-}1)|C_t$) provides no increase in throughput.*

Proof: That the two permutations (o_i, o_j) and (o_j, o_i) will have the same time performance can be seen by interchanging values of p_1 and p_2 in equations (1) and (2). ∎

5.2 Comparison of Performance: Three-Machine Robotic Cells

Since the optimal cyclic solution in a three-machine cell is given by a one-unit cycle (Crama and van de Klundert 1999, Brauner and Finke 1999), we consider only one-unit cycles ($FRC_3|(blocking,A,cyclic\text{-}1)|C_t$). As before, we will denote a permutation of operations by a 3-vector of the form (o_i, o_j, o_k). In other words, operations o_i, o_j, o_k are done on machines M_1, M_2, M_3, respectively.

For convenience, we denote order (o_i, o_j, o_k) as simply (i, j, k) in the rest of the paper. As a result of the two assumptions stated at the start of Section 5, the formulas for T_i $(1 \le i \le 6)$ stated in Lemma 2 may be applied to any of the six permutations of the operations. In fact, for any order $\alpha = (i, j, k)$ of operations, let $T_i(\alpha)$, $1 \le i \le 6$, denote the corresponding value of the i^{th} cycle time measure from Lemma 2 if the operations are processed in the order α. For example, if $\alpha = (2, 1, 3)$ then p_1 and p_2 are simply interchanged in each of the formulas, e.g., $T_3(\alpha) = \max\{8\delta + 8\epsilon + p_2, 4\delta + 6\epsilon + p_1 + p_2, 4\delta + 4\epsilon + p_3\}$. For the sake of efficiency in the remainder of this section, T_i will denote the cycle time value when the order of operations is $(1, 2, 3)$.

Let $OPT(\alpha) = \min\{T_i(\alpha) \mid 1 \leq i \leq 6\}$. Let $T_{i,j}(\alpha)$ be the j^{th} term in the maximization expression which gives the value of $T_i(\alpha)$, where the terms are ordered as in the formulas for T_i given in Lemma 2. For example, $T_{2,4}(1, 2, 3) = 4\delta + 6\epsilon + p_3$ and $T_{2,4}(2, 3, 1) = 4\delta + 6\epsilon + p_1$. The following theorem states that $OPT(\alpha) \geq (6/7)OPT(\beta)$ for all permutations α and β, i.e., at most a $14\frac{2}{7}\%$ decrease in OPT can be obtained by changing the order of operations.

Theorem 1 *Let α and β be two different orders of the operations for a three-machine flexible robotic cell. Then $OPT(\beta) \leq (7/6)OPT(\alpha)$, and this bound is tight.*

Proof: The proof will be presented in the form of six lemmas. The basic approach is to assume that an overall optimal solution (over all orders of operations (i, j, k)) is obtained using the order $(1, 2, 3)$. This leaves six possibilities as to which of the six expressions $T_i, 1 \leq i \leq 6$, yields the optimal value OPT. For each $i = 1, \ldots, 6$, we assume $T_i = OPT$ and then show that for any order of operations $\beta \neq (1, 2, 3)$, there exists at least one j such that $T_j(\beta) \leq (7/6)OPT = (7/6)T_i$.

Lemma 5 *$T_2 \leq T_4$.*

Proof: Clearly, $T_{2,1} \leq T_{4,1}, T_{2,2} \leq T_{4,3}, T_{2,3} \leq T_{4,3}$, and $T_{2,4} \leq T_{4,2}$. Finally,

$$T_{2,5} = 2\delta + 4\epsilon + (\sum p_i/2) \leq \frac{1}{2}(T_{4,2} + T_{4,3}),$$

which implies that either $T_{2,5} \leq T_{4,2}$ or $T_{2,5} \leq T_{4,3}$. Hence, for $j = 1, \ldots, 5$, $T_{2,j}$ is less than or equal to at least one of the $T_{4,k}$'s. ∎

As a result of Lemma 5, we need not analyze the case for which $T_4 = OPT$, since this case will be covered by the case $T_2 = OPT$. In other words, $OPT = \min\{T_1, T_2, T_3, T_5, T_6\}$. Our result can be easily proven for two of these cycles:

Lemma 6 *If $T_1 = OPT$ or $T_6 = OPT$, then Theorem 1 holds.*

Proof: $T_1(\alpha)$ is the same for all permutations α, and $T_6(\beta)$ is the same for all permutations β. Therefore, if $T_1 = OPT$, then $OPT(\alpha) \leq T_1(\alpha) = T_1 = OPT$; if $T_6 = OPT$, then $OPT(\beta) \leq T_6(\beta) = T_6 = OPT$. ■

Lemma 7 *$T_1(\alpha) = OPT, \forall\alpha$, if $p_1 + p_2 + p_3 \leq 4\delta$. $T_6(\alpha) = OPT, \forall\alpha$, if $\delta = 0$.*

Proof: Trivial. ■

Hence, we may assume for the remainder of the proof that $0 < 4\delta < p_1 + p_2 + p_3$.

Lemma 8 *If $0 < 4\delta < p_1 + p_2 + p_3$, then $T_2 = OPT$ implies that Theorem 1 holds.*

Proof: $T_2 = \max\{8\delta + 8\epsilon, 4\delta + 6\epsilon + p_1, 4\delta + 4\epsilon + p_2, 4\delta + 6\epsilon + p_3, 2\delta + 4\epsilon + (\sum p_i/2)\}$.
Suppose that for some permutation β and some number $\mu > 1$, we have

$$\begin{aligned} OPT(\beta) &= \min\{T_1(\beta), T_2(\beta), T_3(\beta), T_5(\beta), T_6(\beta)\} \\ &\geq \mu \cdot OPT = \mu T_2. \end{aligned} \tag{3}$$

First, note that relation (3) implies $T_2(\beta) > T_2$. Necessary conditions for $T_2(\beta) > T_2$ are $p_2 > \max\{p_1, p_3\}$ and that p_2 is processed first or last among the operations of β, in which case $T_2(\beta) = 4\delta + 6\epsilon + p_2$.

Relation (3) also implies many other inequalities, some of which will be used to prove that $\mu \leq 7/6$. First, we need to prove that $\mu < 5/4$. Suppose $\mu \geq 5/4$. Then,

$$\begin{aligned} T_2(\beta) &= 4\delta + 6\epsilon + p_2 \geq \frac{5}{4}T_2 \\ &\geq \frac{5}{4}\max\{T_{2,1}, T_{2,3}\}. \end{aligned}$$

Thus,

$$\begin{aligned} 4\delta + 6\epsilon + p_2 &\geq \frac{5}{4}T_{2,1} = 10\delta + 10\epsilon \\ &\iff p_2 \geq 6\delta + 4\epsilon, \end{aligned} \tag{4}$$

and

$$\begin{aligned} 4\delta + 6\epsilon + p_2 &\geq \frac{5}{4}T_{2,3} = 5\delta + 5\epsilon + \frac{5}{4}p_2 \\ &\iff p_2 \leq -4\delta + 4\epsilon. \end{aligned} \tag{5}$$

Since $\delta > 0$, (4) and (5) contradict each other, and we may conclude that $\mu < 5/4$.

We now apply three other inequalities:

$$\begin{aligned} T_2(\beta) &\geq \mu T_2 \geq \mu T_{2,1} \\ 4\delta + 6\epsilon + p_2 &\geq \mu(8\delta + 8\epsilon) \\ &\iff p_2 \geq (8\mu - 4)\delta + (8\mu - 6)\epsilon \\ &\iff (\mu - 1)p_2 \geq (8\mu^2 - 12\mu + 4)\delta + (8\mu^2 - 14\mu + 6)\epsilon, \end{aligned} \tag{6}$$

and

$$\begin{aligned} T_2(\beta) &\geq \mu T_{2,3} \\ 4\delta + 6\epsilon + p_2 &\geq \mu(4\delta + 4\epsilon + p_2) \\ &\iff (\mu - 1)p_2 \leq (4 - 4\mu)\delta + (6 - 4\mu)\epsilon. \end{aligned} \tag{7}$$

Combining (6) and (7) yields

$$\begin{aligned} (8\mu^2 - 12\mu + 4)\delta + (8\mu^2 - 14\mu + 6)\epsilon &\leq (\mu - 1)p_2 \\ &\leq (4 - 4\mu)\delta + (6 - 4\mu)\epsilon \\ &\iff (8\mu - 8)\delta + (8\mu - 10)\epsilon \leq 0. \end{aligned}$$

$$\text{Since } 1 < \mu, \text{ we have} (8\mu - 8) > 0 \quad \text{and} \quad \frac{\delta}{\epsilon} \leq \frac{10 - 8\mu}{8\mu - 8}. \tag{8}$$

Note that

$$\begin{aligned} T_6 \geq \mu T_{2,3} &\Rightarrow T_6 = 12\delta + 8\epsilon. \quad \text{Thus,} \\ T_6(\beta) &= T_6 \geq \mu T_{2,1} \quad \text{implies} \\ 12\delta + 8\epsilon &\geq \mu(8\delta + 8\epsilon) \\ &\iff (12 - 8\mu)\delta + (8 - 8\mu)\epsilon \geq 0. \end{aligned}$$

$$\text{Since } \mu < 5/4, \text{ we have } (12 - 8\mu) > 0 \text{ and } \frac{\delta}{\epsilon} \geq \frac{8\mu - 8}{12 - 8\mu}. \tag{9}$$

Combining (8) with (9) yields

$$\frac{8\mu - 8}{12 - 8\mu} \leq \frac{10 - 8\mu}{8\mu - 8},$$

which when solved yields $\mu \leq 7/6$. ∎

Lemma 9 *The bound of* $(7/6)$ *in Lemma 8 (and therefore in Theorem 1) is tight.*

Proof: The Lemma 8 case of $T_2 = OPT$ yields the result. Use $\delta = 1, \epsilon = 2, p_1 = p_3 = 8$, and $p_2 = 12$ with $\beta = (2, 1, 3)$. ■

Consider the two orders of operation $\alpha = (i, j, k)$ and $\beta = (k, j, i)$, where β is the reverse of α. It is easy to prove that $T_i(\alpha) = T_i(\beta)$ for $i = 1, 2, 4, 6$, and that

$$T_3(\alpha) = T_5(\beta) \text{ and } T_5(\alpha) = T_3(\beta).$$

Hence,

$$\begin{aligned} OPT(\alpha) &= T_3(\alpha) \Longleftrightarrow OPT(\beta) = T_5(\beta), \\ \text{and } OPT(\alpha) &= T_5(\alpha) \Longleftrightarrow OPT(\beta) = T_3(\beta). \end{aligned}$$

From these results (symmetries), it follows that we need to consider only one of the two cases $T_3 = OPT$ and $T_5 = OPT$. Therefore, to prove Theorem 1, the only remaining case that we need to consider is $T_5 = OPT$.

Lemma 10 *If* $0 < 4\delta < p_1 + p_2 + p_3$*, then* $T_5 = OPT$ *implies that Theorem 1 holds.*

Proof: Either T_2 is also optimal, i.e., $T_2 = T_5 = OPT$, in which case Lemma 8 applies, or $T_2 > T_5$. We therefore assume $T_2 > T_5$.

We use the same approach as in Lemma 8. First note that

$$\left.\begin{array}{c} T_{2,1} \leq T_{5,1},\ T_{2,3} \leq T_{5,2},\ T_{2,4} \leq T_{5,2}, \\ \text{and } \ T_{2,5} \leq \frac{1}{2}(T_{5,2} + T_{5,3}), \quad \text{which implies that} \\ \text{either } \ T_{2,5} \leq T_{5,2} \quad \text{or} \quad T_{2,5} \leq T_{5,3}. \end{array}\right\} \tag{10}$$

Since we are assuming that $T_2 > T_5$, relations (10) imply $T_{2,2} = T_2 > T_5 = OPT$. Thus,

$$T_{2,2} = 4\delta + 6\epsilon + p_1 > T_{5,2} = 4\delta + 6\epsilon + p_2 + p_3 \text{ so } p_1 > p_2 + p_3.$$

It follows that for any β,

$$T_2(\beta) \leq 4\delta + 6\epsilon + p_1 = T_2. \tag{11}$$

Assume that $\mu > 7/6$ is achievable using some permutation β. Then $T_6(\beta) = T_6 > (7/6)T_5$. If $T_6 = 4\delta + 4\epsilon + p_1$, then $4\delta + 4\epsilon + p_1 > (7/6)T_{5,3} = (7/6)(4\delta + 4\epsilon + p_1)$, which is a contradiction. If $T_6 = 12\delta + 8\epsilon$, then we have

$$\begin{aligned} T_6 &> \frac{7}{6}T_{5,1}, \\ 12\delta + 8\epsilon &> \frac{7}{6}(8\delta + 8\epsilon + p_3) \geq \frac{7}{6}(8\delta + 8\epsilon) \\ &\iff 2\delta > \epsilon, \end{aligned} \tag{12}$$

and

$$\begin{aligned} T_2 &\geq T_2(\beta) > \frac{7}{6}T_{5,3}, \\ 4\delta + 6\epsilon + p_1 &> \frac{7}{6}(4\delta + 4\epsilon + p_1) \\ &\iff -4\delta + 8\epsilon > p_1, \end{aligned} \tag{13}$$

and

$$\begin{aligned} T_2 &\geq T_2(\beta) > \frac{7}{6}T_{5,1}, \\ 4\delta + 6\epsilon + p_1 &> \frac{7}{6}(8\delta + 8\epsilon), \\ 6p_1 &> 32\delta + 20\epsilon. \end{aligned} \tag{14}$$

(13) and (14) imply

$$\begin{aligned} 32\delta + 20\epsilon &< -24\delta + 48\epsilon, \\ &\iff 2\delta < \epsilon. \end{aligned} \tag{15}$$

Since (12) contradicts (15), the assumption that $\mu > \frac{7}{6}$ is achievable must be wrong. This completes the proof of Lemma 10 and of Theorem 1. ∎

5.3 Comparison of Performance: Four-Machine Robotic Cells

Although the optimal cycle may not be a one-unit cycle if $m = 4$, we limit our analysis to only one-unit cycles $(FRC_4|(\textit{blocking},A,\textit{cyclic}\text{-}1)|C_t)$. We will denote a permutation of operations by a 4-vector of the form (o_i, o_j, o_k, o_l). In other words, operations o_i, o_j, o_k, o_l are done on machines M_1, M_2, M_3, M_4, respectively.

For convenience, we denote order (o_i, o_j, o_k, o_l) as simply (i, j, k, l) in the rest of the paper. The formulas for one-unit cycles T_i ($1 \leq i \leq 24$) were developed in Lemma 3. For any order $\alpha = (i, j, k, l)$ of operations, let $T_i(\alpha)$, $1 \leq i \leq 24$, denote the corresponding value of the i^{th} cycle time measure, and let $OPT(\alpha) = \min\{T_i(\alpha) \mid 1 \leq i \leq 24\}$. Note that the cycle time formula has the following form: $T_i(\alpha) = \max\{T_{i,j}(\alpha) \mid 1 \leq j \leq k_i\}$, where $T_{i,j}(\alpha)$ is the j^{th} term in the maximization expression for T_i, which has k_i arguments.

We must consider all twenty-four permutations of operations. These permutations are denoted $\alpha_1, \alpha_2, \ldots, \alpha_{24}$, with $\alpha_1 = (1, 2, 3, 4)$. As before, we assume that an overall optimal solution is obtained using α_1.

We now present a mixed integer program $[MIP]$ that finds the ratio $\mu = \max\limits_{2 \leq r \leq 24} \{OPT(\alpha_r)/OPT(\alpha_1)\}$. The first constraints, $(C1)$, $(C2)$, and $(C3)$, derive the values for $T_1(\alpha_r), \ldots, T_{24}(\alpha_r)$, for $r = 1, \ldots, 24$, and arbitrary values of δ, ϵ, and $p_i, i = 1, \ldots, 4$. The next three constraints, $(C4)$, $(C5)$, and $(C6)$, find $OPT(\alpha_1) = \min\limits_{1 \leq i \leq 24} \{T_i(\alpha_1)\}$ and set it equal to 1. Constraint $(C7)$ and the objective function find $U = \min\limits_{2 \leq r \leq 24} \{OPT(\alpha_r)\} = \min\limits_{2 \leq r \leq 24} \{\min\limits_{1 \leq i \leq 24} \{T_i(\alpha_r)\}\}$ and force μ to be as large as possible.

$[MIP]$

Maximize U

subject to:

$$T_i(\alpha_r) \geq T_{i,j}(\alpha_r), \quad r = 1, \ldots, 24; \quad i = 1, \ldots, 24; \quad j = 1, \ldots, k_i \qquad (C1)$$

$$T_i(\alpha_r) \leq T_{i,j}(\alpha_r) + M(1 - W^r_{i,j}), \quad r = 1, \ldots, 24; \quad i = 1, \ldots, 24; \quad j = 1, \ldots k_i \qquad (C2)$$

$$\textstyle\sum_{j=1}^{k_i} W^r_{i,j} = 1, \quad r = 1, \ldots, 24; \quad i = 1, \ldots, 24 \qquad (C3)$$

$$T_i(\alpha_1) \geq 1, \quad i = 1, \ldots, 24 \qquad (C4)$$

$$T_i(\alpha_1) Y_i \leq 1, \quad i = 1, \ldots, 24 \qquad (C5)$$

$$\textstyle\sum_{i=1}^{24} Y_i = 1 \qquad (C6)$$

$$U \leq T_i(\alpha_r), \quad r = 2, \ldots, 24; \quad i = 1, \ldots, 24 \qquad (C7)$$

$$\delta, \epsilon, U, T_1(\alpha_r), \ldots, T_{24}(\alpha_r) \geq 0, \quad r = 1, \ldots, 24$$

$p_i \geq 0, \quad i = 1, \ldots, 4$

$W^r_{i,j}, Y_i \in \{0, 1\}, \quad r = 1, \ldots, 24; \quad i = 1, \ldots, 24; \quad j = 1, \ldots k_i,$

where M is a large positive number.

Theorem 2 *The mixed integer program [MIP] states that* $(\mu)OPT(\alpha_1) \geq OPT(\alpha_r)$, $2 \leq r \leq 24$, *and* $\mu > 1$, *i.e.,* μ *is the largest ratio of increase in* OPT *that can be obtained by changing the order of operations.*

Proof: This result follows directly from the explanation that precedes $[MIP]$. It is easy to see that $\min_{1 \leq i \leq 24}\{T_i(\alpha_1)\} \leq \min_{2 \leq r \leq 24}\{\min_{1 \leq i \leq 24}\{T_i(\alpha_r)\}\} \leq \mu \min_{1 \leq i \leq 24}\{T_i(\alpha_1)\}$ if and only if there exists an optimal solution to program $[MIP]$ with $U > 1$. ∎

Because the number of cycles that produce two or more units is large, we have not enumerated them. Of course, the methodology we describe here could be used to check whether such cycles could provide an improvement in cycle time. In view of the simplicity of one unit-cycles and the results in this section which provide easy comparability of their cycle times, we recommend their use. Note that we can estimate μ for cycles that produce any number of units by running the program $[MIP]$. The optimal solution gives a tight upper bound, and the variable values δ, ϵ, and p_i, $i = 1, \ldots, 4$, achieve this bound.

We now show that for $m = 4$, at most a $14\frac{2}{7}\%$ increase in throughput can be obtained by changing the order of operations.

Theorem 3 *Let* α *and* β *be two different orders of the operations for a four-machine flexible robotic cell. Then* $OPT(\beta) \leq (7/6)OPT(\alpha)$, *and this bound is tight.*

Proof: The proof will be presented in a structure similar to that of Theorem 1: we use eight lemmas and consider cases that are divided by the value of i, where $T_i = OPT$ and T_i denotes $T_i(1, 2, 3, 4)$, $1 \leq i \leq 24$, for the remainder of this section. We first disqualify eight cycles because they are dominated by other cycles.

Lemma 11 *For a given assignment of operations to machines, we have the following dominance relationships: Cycles* $S_{4,4}$, $S_{5,4}$, $S_{9,4}$, $S_{13,4}$, *and* $S_{17,4}$ *are dominated by Cycle* $S_{11,4}$. *Cycles* $S_{12,4}$, $S_{14,4}$, *and* $S_{16,4}$ *are dominated by Cycle* $S_{6,4}$.

Proof: For cycles $S_{4,4}$ and $S_{11,4}$, $T_{11,1} \leq T_{4,1}$, $T_{11,2} \leq T_{4,2}$, $T_{11,3} \leq T_{4,4}$, $T_{11,4} \leq T_{4,3}$, and $T_{11,5} \leq T_{4,2}$. The proofs for the other pairs are similar. ∎

We now show symmetry for five pairs of cycles. These results imply that for each pair we need only consider cases in which one of them is optimal.

Lemma 12 *Regarding Theorem 3, we have the following equivalences:*

- *Theorem 3 holds for $T_2 = OPT$ if and only if it holds for $T_3 = OPT$.*
- *Theorem 3 holds for $T_7 = OPT$ if and only if it holds for $T_{19} = OPT$.*
- *Theorem 3 holds for $T_8 = OPT$ if and only if it holds for $T_{20} = OPT$.*
- *Theorem 3 holds for $T_{15} = OPT$ if and only if it holds for $T_{23} = OPT$.*
- *Theorem 3 holds for $T_{18} = OPT$ if and only if it holds for $T_{22} = OPT$.*

Proof: It is straightforward to verify that $\min\{T_h(i,j,k,l)|1 \leq h \leq 24\} = \min\{T_h(l,k,j,i)|1 \leq h \leq 24\}$. That result along with the following equalities yields the result:

$$\begin{aligned} T_3(i,j,k,l) &= T_2(l,k,j,i), \qquad T_{19}(i,j,k,l) = T_7(l,k,j,i), \\ T_{20}(i,j,k,l) &= T_8(l,k,j,i), \qquad T_{22}(i,j,k,l) = T_{18}(l,k,j,i), \\ T_{23}(i,j,k,l) &= T_{15}(l,k,j,i). \end{aligned}$$

∎

Hence, we need not consider cases in which cycles $S_{3,4}$, $S_{19,4}$, $S_{20,4}$, $S_{22,4}$, or $S_{23,4}$, are optimal.

Lemma 13 *Theorem 3 holds if $T_1 = OPT$, $T_{11} = OPT$, or $T_{24} = OPT$.*

Proof: T_1, T_{11}, and T_{24} are each independent of the assignment of operations to machines. Hence, $T_i(\beta) = T_i < (7/6)T_i$, $\forall \beta$, for $i \in \{1, 11, 24\}$. ∎

Lemma 14 *Regarding Theorem 3, we have the following implications:*

- *If Theorem 3 holds for $T_2 = OPT$, then it holds for $T_7 = OPT$.*

- *If Theorem 3 holds for* $T_6 = OPT$, *then it holds for* $T_8 = OPT$.
- *If Theorem 3 holds for* $T_2 = OPT$, *then it holds for* $T_{10} = OPT$.
- *If Theorem 3 holds for* $T_6 = OPT$, *then it holds for* $T_{15} = OPT$.
- *If Theorem 3 holds for* $T_6 = OPT$, *then it holds for* $T_{18} = OPT$.
- *If Theorem 3 holds for* $T_6 = OPT$, *then it holds for* $T_{21} = OPT$.

Proof:

$$\begin{aligned} T_2(k,j,i,l) &\leq T_7(i,j,k,l), & T_6(j,i,k,l) &\leq T_8(i,j,k,l), \\ T_2(k,j,i,l) &\leq T_{10}(i,j,k,l), & T_6(k,j,i,l) &\leq T_{15}(i,j,k,l), \\ T_6(l,j,k,i) &\leq T_{18}(i,j,k,l), & T_6(j,i,l,k) &\leq T_{21}(i,j,k,l). \end{aligned}$$

■

Lemma 15 $T_1(\alpha) = OPT, \forall\alpha$, *if* $p_1 + p_2 + p_3 + p_4 \leq 4\delta$. $T_{24}(\alpha) = OPT, \forall\alpha$, *if* $\delta = 0$.

Proof: Trivial. ■

Our task now is to show that Theorem 3 holds for cells in which $0 < 4\delta < \sum p_i$ and either $OPT = T_2$ or $OPT = T_6$.

Lemma 16 *If* $0 < 4\delta < \sum p_i$, *then* $T_6 = OPT$ *implies that Theorem 3 holds.*

Proof:

Case 1: $T_{24}(\beta) = 15\delta + 10\epsilon$, $\forall\beta$. $T_6 \geq 13\delta + 10\epsilon$, $\forall\beta$, implies

$$\frac{T_{24}(\beta)}{T_6} \leq \frac{15\delta + 10\epsilon}{13\delta + 10\epsilon} \leq \frac{15}{13} < \frac{7}{6}.$$

Case 2: $T_{24}(\beta) = 4\delta + 4\epsilon + \max p_i$, $\forall\beta$. $T_6 \geq 4\delta + 4\epsilon + \max p_i$, $\forall\beta$, implies

$$\frac{T_{24}(\beta)}{T_6} \leq \frac{4\delta + 4\epsilon + \max p_i}{4\delta + 4\epsilon + \max p_i} \leq 1 < \frac{7}{6}.$$

■

Lemma 17 *If $0 < 4\delta < \sum p_i$, then $T_2 = OPT$ implies that Theorem 3 holds.*

Proof:

Case 1: $T_{11} = 10\delta + 10\epsilon$. $T_2 \geq p_1 + 9\delta + 10\epsilon$ implies that

$$\frac{T_{11}(\beta)}{T_2} \leq \frac{10\delta + 10\epsilon}{p_1 + 9\delta + 10\epsilon} \leq \frac{10}{9} < \frac{7}{6}.$$

Case 2: $T_{11} = 5\delta + 6\epsilon + \max p_i$. First, it is easy to see that $T_{11} > T_2$ implies that $T_{11}(\beta) = 5\delta + 6\epsilon + p_3,\ \forall\beta$. Furthermore, $T_6 > T_2$ implies that $T_6(\beta) = 13\delta + 10\epsilon,\ \forall\beta$.

a) $T_2 = 9\delta + 10\epsilon + p_1$. This implies that $p_3 \leq 5\delta + 6\epsilon + p_1$, so

$$OPT(\beta) \leq \min\{T_6(\beta), T_{11}(\beta)\} \leq \min\{13\delta + 10\epsilon, 10\delta + 12\epsilon + p_1\}$$

Therefore,

$$\frac{OPT(\beta)}{T_2} \leq \min\left\{\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon + p_1}, \frac{10\delta + 12\epsilon + p_1}{9\delta + 10\epsilon + p_1}\right\} \leq \min\left\{\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon}, \frac{10\delta + 12\epsilon}{9\delta + 10\epsilon}\right\}.$$

Now assume that the result does not hold. This would imply that both terms in the minimization exceed 7/6:

$$\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon} > 7/6 \Rightarrow \epsilon < \frac{3}{2}\delta \quad \text{and} \quad \frac{10\delta + 12\epsilon}{9\delta + 10\epsilon} > 7/6 \Rightarrow \epsilon > \frac{3}{2}\delta.$$

This contradiction implies that at least one of these two terms must be less than 7/6.

b) $T_2 = 4\delta + 4\epsilon + p_3$. This implies that $p_3 \geq 5\delta + 6\epsilon$. Hence,

$$\frac{OPT(\beta)}{T_2} \leq \min\left\{\frac{13\delta + 10\epsilon}{4\delta + 4\epsilon + p_3}, \frac{5\delta + 6\epsilon + p_3}{4\delta + 4\epsilon + p_3}\right\} \leq \min\left\{\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon}, \frac{10\delta + 12\epsilon}{9\delta + 10\epsilon}\right\} \leq \frac{7}{6},$$

as in a) above.

c) $T_2 = 5\delta + 8\epsilon + p_1 + p_2$. This implies that $p_2 \geq 4\delta + 2\epsilon$ and that $p_3 \leq \delta + 4\epsilon + p_1 + p_2$, so $T_{11} \leq 6\delta + 10\epsilon + p_1 + p_2$. Hence,

$$\begin{aligned}\frac{OPT(\beta)}{T_2} &\leq \min\left\{\frac{13\delta + 10\epsilon}{5\delta + 8\epsilon + p_1 + p_2}, \frac{6\delta + 10\epsilon + p_1 + p_2}{5\delta + 8\epsilon + p_1 + p_2}\right\} \\ &\leq \min\left\{\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon}, \frac{10\delta + 12\epsilon}{9\delta + 10\epsilon}\right\} \leq \frac{7}{6},\end{aligned}$$

as in a) above. This argument applies to $T_2 = 5\delta + 8\epsilon + p_1 + p_4$, too.

d) $T_2 = (5\delta + 10\epsilon + \sum p_i)/2$. This implies that $p_3 < -(3/2)\delta + \epsilon + \sum p_i/2$. Therefore,

$$T_{11} < \frac{7\delta + 14\epsilon + \sum p_i}{2}.$$

$T_2 > p_1 + 9\delta + 10\epsilon$ implies $\sum p_i > 2p_1 + 13\delta + 10\epsilon \geq 13\delta + 10\epsilon$. Hence,

$$\frac{OPT(\beta)}{T_2} \leq \min\left\{\frac{26\delta + 20\epsilon}{5\delta + 10\epsilon + \sum p_i}, \frac{7\delta + 14\epsilon + \sum p_i}{5\delta + 10\epsilon + \sum p_i}\right\}$$

$$\leq \min\left\{\frac{26\delta + 20\epsilon}{18\delta + 20\epsilon}, \frac{20\delta + 24\epsilon}{18\delta + 20\epsilon}\right\} = \min\left\{\frac{13\delta + 10\epsilon}{9\delta + 10\epsilon}, \frac{10\delta + 12\epsilon}{9\delta + 10\epsilon}\right\} \leq \frac{7}{6},$$

as in a) above. ■

Lemma 18 *The bound of* (7/6) *in Lemma 17 (and therefore in Theorem 3) is tight.*

Proof: Let $\delta = 2$, $\epsilon = 3$, $p_1 = 0$, $p_2 = p_4 = 8$, $p_3 = 28$, and $\beta = (2, 3, 1, 4)$. $OPT = T_2 = 48$. $OPT(\beta) = T_3(\beta) = T_4(\beta) = T_6(\beta) = T_8(\beta) = T_{11}(\beta) = T_{16}(\beta) = T_{18}(\beta) = T_{20}(\beta) = 56$. Therefore, $OPT(\beta)/OPT = 7/6$.

This completes the proof of Theorem 3. ■

6 On-Line Tool Changes

We now consider the case in which the operation performed by a particular machine changes during the processing of a lot. This assumes that tool changes can be made very quickly on each machine. Each part will still visit the machines in order $M_1, M_2, \ldots, M_m$, but operations performed by a particular machine will vary from part to part. Each part must still be subject to each of the m operations. Our concern is whether varying the order in which the processes are performed can increase throughput. We first examine two-machine cells, then generalize to larger ones.

Recall that for $m = 2$ the order of operations (i, j) means that operation o_i is performed on machine M_1 and that operation o_j is performed on machine M_2. We can specify the order of operations for each part, e.g., the first part is processed in order (i, j), the second in order (j, i), etc. Obviously, if robot cycle $S_{1,2}$ is used, then the throughput does not change if the order of operations changes: $T_1 = 3\delta + 6\epsilon + p_1 + p_2$, independent of the assignment of operations to machines. However, if robot cycle $S_{2,2}$ is used, then the throughput may be improved by using a varying order of operations. This problem is designated $OFRC_2|(blocking,A,S_{2,2})|C_t$.

To study problem $OFRC_m|(blocking,A,S_{q,m})|C_t$, we define the notation $S_{q,m}[\beta_1, \beta_2, \ldots, \beta_k]$ to be the k-unit cycle generated by repeating k times the one-unit robot move cycle $S_{q,m}$, with the assignment of operations to machines being β_1 for the first part, β_2 for the second part, ... , and β_k for the k^{th} part. In the cases studied here, $k = m$. If the k-unit cycle under consideration uses the same assignment of operations throughout its performance, we simply have the one-unit cycle $S_{q,m}$ of Section 4. Similarly, to maintain consistency with the notations of Section 3, we define $T_q[\beta_1, \beta_2, \ldots, \beta_k]$ to be the cycle time of $S_{q,m}[\beta_1, \beta_2, \ldots, \beta_k]$.

When performing cycle $S_{2,2}[(i,j)(j,i)]$, the waiting times differ from those of cycle $S_{2,2}$, so $T_2[(i,j)(j,i)]/2 \neq T_2$ (as defined in equation (2)). In cycle $S_{2,2}[(i,j)(j,i)]$, the robot's movements are identical to those described for $S_{2,2}$ in Section 4.1. If part h is produced in sequence (i,j) and part $h+1$ is produced in sequence (j,i), then part h will undergo operation o_j on machine M_2 while part $h+1$ is undergoing operation o_j on M_1. Similarly, part $h+1$ will undergo operation o_i on machine M_2 while part $h+2$ is undergoing operation o_i on M_1. If w_g^r indicates the robot's waiting time at machine M_g when it is performing operation r, then we have the following values for waiting times:

$$\begin{aligned} w_1^i &= \max\{0, p_i - 3\delta - 2\epsilon - w_2^i\} \quad \text{(first part)} \\ w_2^j &= \max\{0, p_j - 3\delta - 2\epsilon\} \quad \text{(first part)} \\ w_1^j &= \max\{0, p_j - 3\delta - 2\epsilon - w_2^j\} \quad \text{(second part)} \\ w_2^i &= \max\{0, p_i - 3\delta - 2\epsilon\} \quad \text{(second part)}, \end{aligned}$$

so the time required to complete the two parts is

$$\begin{aligned} T_2[(i,j),(j,i)] &= 12\delta + 12\epsilon + w_1^i + w_2^j + w_1^j + w_2^i \\ &= 12\delta + 12\epsilon + \max\{0, p_i - 3\delta - 2\epsilon\} + \max\{0, p_j - 3\delta - 2\epsilon\} \\ &= \max\{6\delta + 6\epsilon, p_i + 3\delta + 4\epsilon\} + \max\{6\delta + 6\epsilon, p_j + 3\delta + 4\epsilon\}. \end{aligned}$$

Hence, $T_2[(i,j),(j,i)]/2 \leq T_2$, and alternating the operations on the machines will reduce the per unit cycle time by $\min\{|p_j - p_i|, \max\{0, p_i - 3\delta - 2\epsilon, p_j - 3\delta - 2\epsilon\}\}/2$. Gantt charts for $S_{2,2}$ and $S_{2,2}[(i,j)(j,i)]$ can be found in Figure 2.

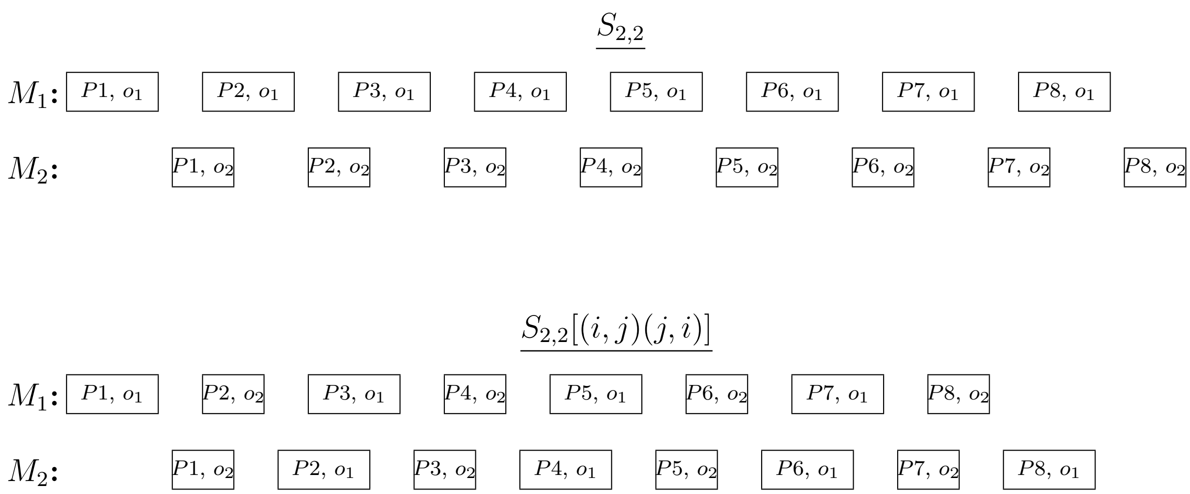

Figure 2: Gantt charts for $S_{2,2}$ and $S_{2,2}[(i,j)(j,i)]$

With these results, we have the following lemma, which is analogous to Lemma 1:

Lemma 19 *In* $ORF_2|(blocking,A,cyclic\text{-}2)|C_t$, *cycle* $S_{1,2}$ *is optimal if* $\delta \geq (p_1 + p_2)/3$, *while cycle* $S_{2,2}[(i,j),(j,i)]$ *is optimal if* $\delta \leq (p_1 + p_2)/3$.

Proof: Same as for Lemma 1. ■

Similarly, if in a three-machine cell we perform $S_{6,3}$ ($OFRC_3|(blocking,A,S_{6,3})|C_t$) so that the order in which operations are performed rotates in order $(i,j,k),(j,k,i),(k,i,j)$, then the time required to produce three successive units is

$$\begin{aligned} T_6[(i,j,k),(j,k,i),(k,i,j)] &= 36\delta + 24\epsilon + w_1^i + w_2^j + w_3^k + w_1^j + w_2^k + w_3^i + w_1^k + w_2^i + w_3^j \\ &= 36\delta + 24\epsilon + \max\{0, p_i - 8\delta - 4\epsilon\} + \max\{0, p_j - 8\delta - 4\epsilon\} \\ &\quad + \max\{0, p_k - 8\delta - 4\epsilon\} \\ &= \max\{12\delta + 8\epsilon, p_i + 4\delta + 4\epsilon\} + \max\{12\delta + 8\epsilon, p_j + 4\delta + 4\epsilon\} \\ &\quad + \max\{12\delta + 8\epsilon, p_k + 4\delta + 4\epsilon\} \leq 3T_6. \end{aligned}$$

These results can be easily extended to $OFRC_m|(blocking,A,S_{m!,m})|C_t$, $m \geq 4$.

7 Conclusions and Recommendations for Future Study

We have examined the productivity gains that can be achieved in flexible robotic cells by changing the assignment of operations to machines. If this assignment remains constant throughout a lot's processing, flexibility provides no throughput increase for a two-machine cell. Furthermore, for both three- and four-machine cells, the maximum productivity increase is $14\frac{2}{7}\%$. We have also shown that flexible cells that support on-line tool changes can be scheduled in a way that increases throughput. Both results should be very useful to robotic cell designers and to those considering the purchase of such a system, because flexible robotic cells are more expensive than robotic flow shops.

It is curious that the results of Section 5 show that the maximum throughput increase in a flexible robotic cell is $14\frac{2}{7}\%$ for both $m = 3$ and $m = 4$. It is not clear whether this trend would continue for $m \geq 5$. Examining this trend would be a challenging and useful question for future research. Another interesting line of inquiry would be to quantify the overall productivity gains for cells having $m \geq 4$, as the best one-unit cycle may not be optimal among the class of all cyclic solutions (Brauner and Finke 1997, 2001). Another direction for future research is towards finding similar results for cells that produce different part-types. Foundational work for such cells can be found in Hall et al. (1997), Hall et al. (1998), Kamoun et al. (1999), and Sriskandarajah et al. (1998). Dual gripper robotic cells, too, may prove to be a profitable field for the examination of flexible robotic cells. Cyclic solutions for dual gripper robotic cells have been studied in Su and Chen (1996), Sethi et al. (2001), Sriskandarajah et al. (2004), and Geismar et al. (2003).

Acknowledgments: The research was supported in part by the Natural Sciences and Engineering Research Council of Canada (Grant number OGP0002507) and the University of Texas at Dallas.

References

Asfahl, C.R., *Robots and Manufacturing Automation*, John Wiley & Sons, New York (1985).

Askin, R.G. and Strandridge, C.P., *Models and Analysis of Manufacturing Systems*, John Wiley & Sons, New York (1995).

Basnet C. and Mize, J.H., "Scheduling and Control of Flexible Manufacturing Systems: A Critical Review," *International Journal of Computer Integrated Manufacturing*, Vol 7, No. 6, pp. 340-355 (1994).

Brauner, N. and Finke, G., "Final Results on the One-cycle Conjecture in Robotic Cells," Internal note, Laboratoire LEIBNIZ, Institut IMAG, Grenoble, France (1997).

Brauner, N. and Finke, G., "On the Conjecture in Robotic Cells: New Simplified Proof for the Three-Machine Case," *INFOR*, Vol. 37, pp. 20-36 (1999).

Brauner, N. and Finke, G., "Cycles and Permutations in Robotic Cells," *Mathematical and Computer Modeling*, Vol. 34, pp. 565-591 (2001).

Browne, J., Dubois, D., Rathmill, K., Sethi, S.P., and Stecke, K.E., "Classification of Flexible Manufacturing Systems," *The FMS Magazine*, Vol. 2, No. 2, pp. 114-117 (April 1984).

Crama, Y., Kats, V., van de Klundert, J., and Levner, E., "Cyclic Scheduling in Robotic Flow Shops," *Annals of Operations Research: Mathematics of Industrial Systems* (2000).

Crama, Y. and van de Klundert, J., "Cyclic Scheduling of Identical Parts in a Robotic Cell," *Operations Research*, Vol. 45, pp. 952-965 (1997).

Crama, Y. and van de Klundert, J., "Cyclic Scheduling in 3-machine Robotic Flow Shops," *Journal of Scheduling*, Vol. 2, pp. 35-54 (1999).

Dawande, M., Geismar, N., and Sethi, S., "Dominance of Cyclic Solutions and Challenges in the Scheduling of Robotic Cells," under review in *SIAM Review* (2002a).

Dawande, M., Geismar, N., Sethi, S., and Sriskandarajah, C., "Sequencing and Scheduling in Robotic Cells: Recent Developments," under review in *Journal of Scheduling* (2002b).

Geismar, H.N., Dawande, M., and Sriskandarajah, C., "Scheduling Constant Travel-Time Dual Gripper Robotic Cells with Parallel Machines," Working paper, University of Texas at Dallas (2003).

Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G., "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey," *Annals of Discrete Mathematics*, Vol. 5, pp. 287-326 (1979).

Hall, N.G., Kamoun, H., and Sriskandarajah, C., "Scheduling in Robotic Cells: Classification, Two and Three Machine Cells," *Operations Research*, Vol. 45, pp. 421-439 (1997).

Hall, N.G., Kamoun H., and Sriskandarajah, C., "Scheduling in Robotic Cells: Complexity and Steady State Analysis," *European Journal of Operational Research*, Vol. 109, pp. 43-63 (1998).

Harmonosky, C.M. and Robohn, S. F., "Real-Time Scheduling in Computer Integrated Manufacturing: A Review of Recent Research," *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 6, pp. 331-340 (1991).

Kamoun, H., Hall, N.G., and Sriskandarajah, C., "Scheduling in Robotic Cells: Heuristics and Cell Design," *Operations Research*, Vol. 47, pp. 821-835 (1999).

Miller, R.K. and Walker, T.C., *FMS/CIM Systems Integration Handbook*, The Fairmont Press, Lilburn, GA (1990).

Rachamadugu, R. and Stecke, K.E. "Classification and Review of FMS Scheduling Procedures," *Production Planning and Control*, Vol. 5, No. 1, pp. 2-20 (1994).

Sethi, A.K. and Sethi, S.P., "Flexibility in Manufacturing: A Survey," *International Journal of Flexible Manufacturing Systems*, Vol. 2, pp. 289-328 (1990).

Sethi, S.P., Sidney, J.B., and Sriskandarajah, C., "Scheduling in Dual Gripper Robotic Cells for Productivity Gains," *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 324-341 (2001).

Sethi, S.P., Sriskandarajah, C., Sorger, G., Błażewicz, J., and Kubiak, W., "Sequencing of Parts and Robot Moves in a Robotic Cell, " *International Journal of Flexible Manufacturing Systems*, Vol. 4, pp. 331-358 (1992).

Sriskandarajah, C., Drobouchevitch, I. G., Sethi, S., and Chandrasekaran, R., "Scheduling Multiple Parts in a Robotic Cell Served by a Dual Gripper Robot," *Operations Research*, Vol. 52, No. 1, pp. 65-82 (2004).

Sriskandarajah, C., Hall, N.G., and Kamoun, H., "Scheduling Large Robotic Cells without Buffers", *Annals of Operations Research*, Vol. 76, pp. 287-321 (1998).

Stecke, Kathryn E., "Design, Planning, Scheduling, and Control Problems of Flexible Manufacturing Systems," *Annals of Operations Research*, Vol. 3, pp. 3-12 (1985).

Stecke, Kathryn E., "FMS Design and Operating Problems and Solutions," *Proceedings of the Second Intelligent FA Symposium*, pp. 17-32 (July 19-21, 1989)

Su, Q and Chen, F., "Optimal Sequencing of Double-Gripper Gantry Robot Moves in Tightly-Coupled Serial Production Systems," *IEEE Transactions on Robotics and Automation*, Vol. 12, pp. 22-30 (1996).