SCHEDULING IN TWO-MACHINE ROBOTIC CELL: AN OPTIMAL ALGORITHM BASED ON BUFFERING

RADHA PENEKELAPATI

A THESIS

IN

THE DEPARTMENT

OF

MECHANICAL ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

JANUARY 2000

0-612-47838-6

Abstract

Scheduling in Two-Machine Robotic Cell:
An Optimal Algorithm Based on Buffering

Radha Penekelapati

Two important trends in developing innovative and efficient approaches for improving plant productivity are cellular manufacturing and robotics. The proliferation of robot technology is an outcome of increasing industrial automation especially in engineering and electronics. Robots offer substantial gains in manufacturing productivity, particularly when integrated into an automated system. Robotic cells involve the use of robots to feed machines in manufacturing cells. The factors affecting the performance of such systems include sequencing robot moves, sequencing parts, buffering, and cell design. This thesis addresses the problem of sequencing robot moves in a two machine manufacturing cell in the presence of a buffer.

We develop cycle time formulae using a state space approach. We adopt analytical methods for determining the optimal cycle time of a two-machine robotic cell with a single buffer producing identical parts. We also evaluate the effectiveness of buffering in reducing the cycle time. We extend our research to the robotic cell producing multiple part types. We consider the production of a quantity known as minimal part set (MPS) for multiple part types, to be compatible with the recent trend toward just-in-time manufacturing. Our objective is to identify optimal robot move sequences for a pre-determined arrangement of parts in a minimal part set. We accomplish our goal by developing a branch-and-bound algorithm. We also provide a comparative analysis for scenarios with and without a buffer to establish the usefulness of a buffer.

To my parents.

Acknowledgments

I am grateful to my supervisors, Dr. S.V. Amiouny and Dr. K. Demirli, for their support and continued encouragement. Dr. Amiouny and Dr. Demirli were instrumental in the completion of this work. I sincerely appreciate them for providing guidance, thoughtful suggestions and valuable comments. I acknowledge the financial assistance provided by them.

I gratefully acknowledge the Graduate Partial Tuition Scholarship awarded by Concordia University. I would like to thank the faculty, staff and graduate students of the Mechanical Engineering Department for making my stay at Concordia a pleasant one.

I am indebted to Dharma for his incessant support through out the course of this work. I am thankful to my parents and my family for their unwavering support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robotic cells are an outcome of increased levels of automation in manufacturing industries. Although manufacturing cells can have human operators to perform material handling, the full benefits to productivity from the use of cells are only reaped by robot handling. Among various issues pertaining to robotic cells, robot move scheduling, part sequencing and buffering are important as they affect the performance. This research provides optimal sequence of robot moves for a two-machine robotic cell with one buffer and single part type and supplies a program to identify optimal sequences for multiple part types in a minimal part set.

1.1 Cellular Manufacturing

Cellular Manufacturing (CM) is one of the most innovative and efficient approaches to improve plant productivity. Cellular manufacturing involves the identification of parts requiring similar processing requirements (same machine setups and same machine routings) and grouping of such parts into part families and the organization of corresponding machines in to cells called manufacturing cells. A variety of definitions are offered for a manufacturing cell. Dennis E. Wisnowsky, president of Wizdom Systems, Inc. (Naperville, IL) defines a manufacturing cell as "*the grouping of people and processes into a specific area dedicated to the production of a family of parts or products*" [Mar89].

Manufacturing cells have assumed importance as they offer a compromise between stand-alone machine tools and full blown manufacturing systems. Manufacturing cells have proved advantageous as they increase throughput and machine utilization, reduce work-in-process, shorten lead and turn around times, and offer better quality, increased flexibility and increased control and predictability in the manufacturing process. Staggering figures have been offered in the industry to prove the efficiency of cellular manufacturing. One such example is a plant named Deere Tech Services (Moline, IL), producing hydraulic cylinders. A move to CM in 1988 reduced the part numbers from 405 to 75, inventory was reduced from 21 days to 10 days supply, setup time was cut by 75%, lead times and material handling was cut by 42%, and scrap was reduced by 80% [Mar89].

1.2 Robotic Cells

The proliferation of robot technology is an outcome of increasing industrial automation especially in engineering and electronics. Robots are used to perform various tasks ranging from assembly to testing and inspection [AW87]. Robots offer substantial gains in manufacturing productivity, particularly when integrated in to an automated system. This is because the robot can make precise movements repeatedly and rapidly; it can operate effectively in hazardous environments; and it can perform its functions without fatigue, scheduling difficulties, or other labour related problems [NH88]. It has also been observed that in a wide variety of industrial settings, material handling within a cell was accomplished very efficiently by the use of robots. Where this is the case,

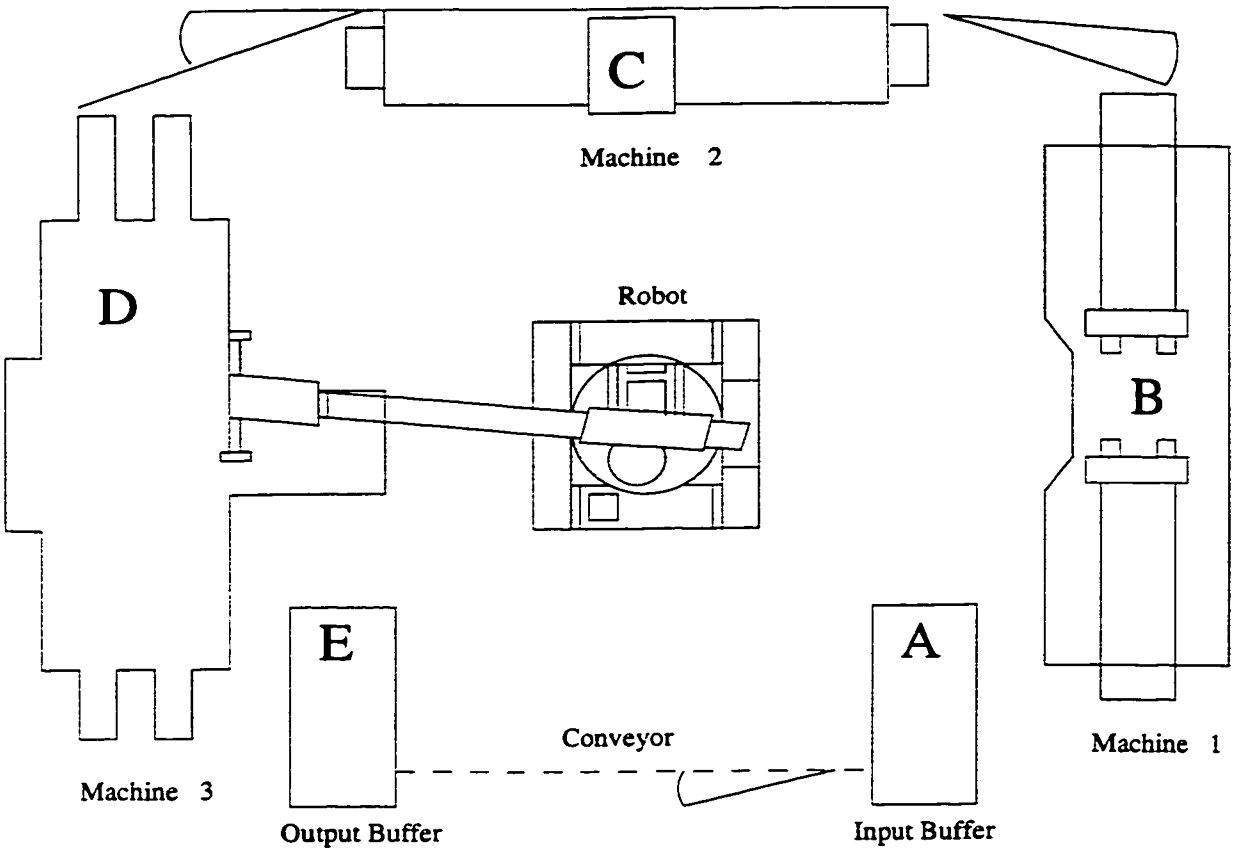

Figure 1: An Example of a Robotic Cell.

the manufacturing cell is known as a *robotic cell.* An implementation at the Fanuc factory in Japan increased cell production from 90 motors per day when manually served, to 300 per day when robot-served [IS82]. It has been pointed out that a cell system is ideally suited to robot handling, especially where lot sizes are small [Har83].

A robotic cell is a manufacturing cell which contains a robot system and programmable units such as CNC (computer numerically controlled) machine tools or other automated equipment [NH88]. The robotic work cell has few machines and is arranged so that a robot can load and unload the machines and change the tools in them if necessary [BS93]. Robotic cells can be classified in to three categories depending on the location and the mobility of the robot in the manufacturing cell. The three different types of robotic cell layouts [Gro87] are:

- Robot Centered Cell

- Mobile-Robot Cell
- In-line Robot Cell.

The Robot centered cellular layout, which is widely accepted in the literature, is adopted in the present work. It is a layout where the robot operates as the central unit in the manufacturing cell. Figure 1 shows a typical example of a robot centered cell [KS84]. The alphabetical order of letters in Figure 1 indicates the flow of work. The robot is located in the center of the manufacturing cell and the three machines, and the input and the output are located around the robot in the arc of a circle. The machines are arranged in a semi-circle around the robot to take advantage of the range of motion of the robot. The movement of the robot arm is rotational. The robot base is mounted on the floor and is stationary. Such floor-mounted, pedestal robots can serve one or more machines and are preferred for local loading tasks. Floor-mounted robots often use a pallet pick-and-place or a conveyor feeder to feed parts to and from the location. The robot can be programmed to pick or place parts on a geometrically designed surface, putting the parts in rows or palletizing them in a given area. A robotic cell with a conveyor not only improves the consistency of operations but also helps in reducing work-in-process costs significantly. Among the interrelated issues to be considered in using robotic cells are their design, the scheduling of robot moves, and the sequencing of parts to be produced [Kam94].

1.3 Scheduling

Scheduling is an important aspect of production activity control. The scheduling problem comprises of the allocation of resources (machines) to a set of tasks (jobs) with respect to time [Bak74]. Concisely, the scheduling problem is one of timetabling the processing of jobs or batches on to machines or workstations so that a given measure of performance achieves its optimal value. The performance measure or the criteria could be anything ranging from the usage of machine, space, material etc., to product quality, throughput, timeliness and the robustness to changes due to machine breakdowns [Fre82]. These performance measures or objectives, which vary from one manufacturing environment to another and sometimes from day to day, are numerous, complex and often conflicting.

1.3.1 Cyclic Schedules

Cyclic schedules are an integral part of repetitive manufacturing environments such as Just-In-Time (JIT). A cyclic schedule can be defined as a schedule which repeats itself after every C units of time, C being the cycle length.

Cyclic schedules are particularly suited for Just-in-Time systems, which provide exactly the required products at the required times in the required quantities. The natural objective in such a system is to maintain a constant production rate of all products such that the demand is met "just in time". A forecast of the demand for various part types is made and the product mix ratios are determined. The minimum number of parts of each type that satisfy the required ratios are identified and the minimal part set is produced in a cyclic fashion. The minimal part set is the smallest representative set of the production target. For instance, if the total demand for one week includes 200 units of part type A, 300 units of part type B and 500 units of part type C. The minimal part set (MPS) includes 2 units of part type A, 3 units of part type B and 5 units of part type C. Thus the weekly production would consist of producing one MPS followed by another MPS and so on until the total weekly demand is met. The success of Japanese industries especially automobile industries can be accrued to cyclic scheduling. The repetitive nature of cyclic schedule stimulates improvement of both the product and the processes [Hal88].

1.4 Research Goals

The use of robotic cells in manufacturing industries has helped in successfully accomplishing the production targets. The issues which arise in the wake of increase in the use of robotic cell technology are effective cell formation, optimal sequence of robot moves, and parts and the use of buffer storage within the cell. The problem of deciding how to form parts and machines into manufacturing cells or rather cell formation has been researched by many. There has been several studies of the scheduling of different types of robotic cells with different criteria of performance. However the problem of a robotic cell with limited buffer storage between the machines is not well studied in the literature.

This research delves into the problem of a two-machine robotic cell with a single buffer between the machines with an attempt to minimize the cycle time. It addresses the issue of determining the effectiveness of buffer in reducing the cycle time. It first identifies the six potentially optimal one unit robot move cycles for a two machine robotic cell with a single buffer, producing a single part-type. This work provides sufficient conditions under which each of these six cycles is optimal and thus outlines the conditions under which a buffer effectively reduces the cycle time. It also quantifies the reduction in cycle time achieved by a buffer. This study can be used to improve the efficiency of robotic cells which manufacture the same part repetitively over some period of time before switching to another part type.

This research is extended to the robotic cell producing multiple part types. There are two scheduling issues that need to be resolved in this problem which are:

- Order of processing of parts in MPS
- Sequence of robot moves for each part.

In some cases the order of processing of parts is set by other considerations especially in *Kanban* systems where product mix has to be highly stable and variability causes disruption [SB97] [VBW92]. The order of processing of parts in MPS is fixed in our study. Hence, the problem is investigated with a goal to identify optimal sequences for a pre-determined order of parts in a MPS. It also provides a program for suggesting optimal sequences to be used to process the parts in a given MPS, preserving the cyclic nature of the schedules. In the case where the processing order of parts is to determined along with robotic move sequences, the results of our study can still be used by executing the program for all the possible arrangements of parts and choosing the optimal order of processing of parts. This total enumeration may not be efficient and a search procedure can be built to arrange the parts in the MPS in an optimal fashion or near-optimal fashion. However, we do not address this issue in this research. Further, a comparative analysis between bufferized and non-bufferized scenarios is attempted to establish the usefulness of a buffer.

Although a robot centered cell is used throughout this work, other types of robotic cells such as mobile-robot cell or in-line robot cell can be treated in a similar fashion. In a mobile robot cell there is an additional component of robot moving time representing the linear movement of the robot. This additional component of time can be combined with the rotational component of time while computing the cycle time of the sequences.

1.5 Thesis Outline

Chapter 2 reviews the literature pertaining to robotic cells. It identifies the important research directions in this field such as cell formation, cyclic scheduling, scheduling of robot moves, sequencing of parts, complexity and steady state analysis. It also summarizes some significant results concerning scheduling in robotic cells. It explains the motivation behind this research. It also describes the two robotic cell models adopted in this research work. It concludes with an outline of the assumptions made in the robotic cell models.

Chapter 3 includes a description of all the entities and their feasible states in case of a robotic cell system producing identical parts. It presents the state space representation of the system and demonstrates how to identify and describe all the possible robotic sequences using the state space approach. It then provides the expressions for cycle times with an attempt to find out the optimal sequence. It also illustrates graphically which sequence is optimal under which conditions, and the conditions where reduction in cycle time is accomplished with a buffer.

Chapter 4 gives an introduction to combination sequences, which are an output of switchings from one cyclic sequence to another. It provides an outline of the common states between the six cyclic sequences and includes descriptions of the robotic moves and computations of sequence times for the combination sequences. It then provides expressions for cycle times of cyclic sequences in this two machine robotic cell set up manufacturing multiple part types. It illustrates the preceding and succeeding sequences for all the feasible sequences and provides the constraints to be imposed to preserve the cyclic nature of the schedules thus establishing a framework to identify the optimal sequences to be used to process the parts in a MPS.

Chapter 5 describes a branch-and-bound algorithm which identifies optimal sequences for a given processing order of parts in the MPS. An algorithm is presented and its corresponding program written in C++ is used to establish the need for a buffer, and to observe the amount of reduction in processing time with the help of a buffer. It compares the observations with the two-machine cell with identical parts problem. Chapter 6 concludes the thesis with an outline of its contribution and the future goals.

Chapter 2

Background

The research issues pertaining to Robotic cells range from scheduling to cell design. Many researchers including Sethi, Hall, Kamoun, Sriskandarajah and Logendran have worked on Scheduling in different types of Robotic cells such as two-machine cell, three machine-cell and large robotic cells. Their research focuses on identifying optimal robotic moves and part sequences to minimize the cycle time under various conditions and assumptions. The problem of robot move sequencing in a two machine robotic cell with a buffer is not studied by the researchers and thus forms the topic of this research. Two models of a two machine robotic cell are considered in this work.

2.1 Introduction

There has been extensive research on different issues concerning robotic cells including cell formation, cyclic scheduling, scheduling of robot moves, sequencing of parts, complexity and steady state analysis [HKS95]. A broad review of the literature could be found in Sethi *et al.* [SSS+92]. However, this chapter lays emphasis on the literature pertaining to sequencing of parts and robot activities in robotic cells and the relevance of this work. Section 2.2 presents a general review of the problem domain. Section 2.3 provides a detailed review of literature that bears a direct influence on this thesis. Section 2.4 discusses the motivation behind this work. Section 2.5 describes vividly the two robotic cell models adopted in this research work and Section 2.6 outlines the assumptions made in the robotic models which are being studied.

2.2 Problem Domain

Cell formation, one of the first problems faced in designing a cellular manufacturing system, involves grouping similar parts into families and corresponding machines into cells. The problem has attracted great attention both from academia as well as industry. Many researchers, including McAuley [McA72], Burbridge [Bur75], Rajagopalan and Batra [RB75], King [Kin80], Chan and Milner [CM82], Seiffodini and Wolfe [SW86], Kusiak [Kus87], and Askin and Chiu [AC90] have studied this problem. They have suggested mostly heuristic procedures of a clustering type for forming the cells. Chao-Hsien Chu [Chu93] has suggested a neural network approach to handle this problem.

One notable result in cyclic Scheduling is the polynomial time algorithm developed by Gilmolre and Gomory [GG64] for solving a special case of the traveling salesman problem. Hall [Hal88] describes the application of cyclic scheduling in a repetitive manufacturing environment such as Just In Time. Crama and Klundert [CK97] proved that the identical parts cyclic scheduling in a m machine robotic cell can be solved in $O(m^3)$ time. Hall and Sriskandarajah provide a survey of machine scheduling problems with blocking and no-wait in process [HS96].

Heuristic procedures for optimizing the working cycle of an industrial robot were developed by Bedini *et al.* [BLS79]. Kondoleon [Kon79] analyses the effects of various robot assembly system configurations on the cycle time. Baumann *et al.* [BBH+81] study robot and machine utilization in a robotic cell. Asfahl [Asf85] uses simulation to compute the cycle times. Claybourn and Hewit [CH82], Badalamenti and Bao [BB86], and Noh and Herring [NH88] have described simulation studies of robotic cells. A branch-and-bound based algorithm was developed by Hitomi and Yoshimura [HY86] for obtaining an optimum robot transport sequence that minimizes the makespan. Seidmann *et al.* [SSN85] present a predictive model for describing the productive capacity of multi-product robotic cells with stochastic times.

2.3 Scheduling in Robotic Cells

This section summarizes some important results pertaining to scheduling in two machine, three machine and large robotic cells producing either identical parts or multiple parts under different assumptions.

2.3.1 Sequencing of Parts and Robot Moves in a Robotic Cell

Sethi *et al.* [SSS+92] used a state space approach to determine a sequence of robot moves to minimize the cycle time of a robot centered cell, for the following cases

- Single part type with two machines
- Single part type with three machines
- Multiple part types with two machines

For a single part type with two machines they identify the two feasible optimal cycles which are one part cycles (one part cycle: cycle which produces a single part) and provide expressions for these cycle times. For a single part type with three machines they find the six optimal one part cycles. They show that two of these cycles are dominated by the other four. They conjecture that one part cycle is optimal over all other cycles which are formed as a combination of several one part cycles. They formulate multiple part type problem with two machines as a traveling salesman problem.

They fix one of the two optimal cycles and they reduce the problem to sequencing of parts of the minimal part set.

The assumptions made by them are as follows:

- Time for loading and unloading is same for all types of parts on all the machines.
- Traveling time between any two consecutive machines is same.
- There are no buffers between the machines.

2.3.2 Classification, Two and Three Machine Cells

While maintaining the assumptions and the objective stated by Sethi *et al.* [SSS+92], Hall *et al.* [HKS97] consider a mobile robot cell. They provide a classification scheme for scheduling problems in robotic cells. They consider a two machine, multiple part type problem and propose an algorithm Min cycle, which optimizes the robot move cycle and the part sequence. They made an observation that the repetition of the best one-unit robot move cycle is not optimal for multiple part type problem and that the cycle times can be reduced using both the feasible sequences (S_1 and S_2). They noted that there can be a switch from one cycle to another at a state common to both the cycles and that in any schedule the sequence S_1 or S_2 is followed by either S_1 or $S_{1,2}$. Similarly S_2 is followed by either S_2 or $S_{2,1}$. For a single part type with three machines they provided a proof for the conjecture stated in Sethi *et al.* [SSS+92] which is that, one unit cycles dominate more complicated policies that produce two units. However, the problem where cycles producing three or more units can provide improvement has not been solved by them. There are certain unresolved issues in their work such as

- whether one unit cycles in single part type problems with more than three machines dominate other cycles?
- In multiple part type problems do complicated cycles offer an improvement for more than two machines?

For a three machine cell with multiple part types the optimal part sequencing problems associated with four of the six optimal cycles provided by Sethi *et al.* [SSS+92] are shown by them to be polynomially solvable. They observed that if processing times are short relative to robot travel times, it is advantageous to wait at a machine during processing rather than drop the part and move elsewhere.

2.3.3 Heuristics and Cell Design

Kamoun *et al.* [KHS95] consider a two machine, multiple part type problem with the same objective, assumptions and the cell layout as Hall *et al.* [HKS97]. They found that the optimizing algorithm given by Hall *et al.* [HKS97] was inefficient in terms of computational time especially for larger problems. So, they provided a faster heuristic which performs better as the number of parts increase although it is not optimal. In the two machine, multiple part type problem, multiple minimal part sets were considered. They found that by using multiple minimal part sets the cycle time could be reduced but with an increase of inventory in the output buffer. For three machine cell with multiple part types, they described and tested heuristic procedures for several scheduling problems which were shown by Hall *et al.* [HKS97] as intractable. They also described and tested Heuristics for cell design problems.

2.3.4 Scheduling Large Robotic Cells

For an m machine cell where $m \geq 2$ there are $m!$ potentially optimal robot move cycles that produce one unit. The cycle time minimization problem is reduced to a unique part sequencing problem once one of these cycles is chosen. Sriskandarajah *et al.* [SHKW95] classified the part sequencing problems associated with these robot move cycles as

- Sequence independent (trivially solvable).
- Capable of formulation as a traveling salesman problem (TSP), but polynomially solvable.
- Capable of formulation as a TSP and unary NP-hard.
- Unary NP-hard but not having TSP structure.

They also proved that the part sequencing problems associated with $2m - 2$ of the $m!$ available robot cycles are polynomially solvable. They identified that the remaining cycles have associated part sequencing problems which are unary NP-hard.

2.3.5 Sequencing of Robot Activities and Parts in Two-Machine Cells

Analytical methods were proposed by Logendran *et al.* [LS96] to obtain optimal sequences of robot activities and parts in a two machine robotic cell to minimize the cycle time. Both single part type and multiple part type problems were investigated under each of the three robotic cell layouts: robot-centered cell, mobile robot cell and in-line robot cell. Several assumptions were relaxed such as the robot travel time between two consecutive machines being considered different. For the problem of multiple part types with two machines, the analysis for a single MPS was extended to include the cyclic production of multiple MPSs.

2.4 Motivation

Scheduling in two machine robotic cell is extensively studied by the researchers. But the use of limited buffer storage within the robotic cell as a means of reducing cycle time remains an open problem. Hence, we consider the problem of finding optimal cycle time in a two machine robotic cell with a buffer. The problem of two machine robotic cell with a buffer bears some similarity with the three machine robotic cell producing identical parts studied by Sethi *et al.* [SSS⁺92]. The buffer in the two machine cell can be considered as a second machine in the three machine cell. The robot can wait at the second machine but doesn't wait at the buffer. Hence, the three machine problem can be transformed to a two machine cell with a buffer by setting the processing time on the second machine to zero. The loading and unloading operations on the second machine in a three machine cell can be compared with the dropping and picking operations associated with the buffer in a two machine cell. However, the study of Sethi *et al.* [SSS⁺92] is limited to identical parts. Hall *et al.* [HKS97] study the multiple part-types problem in a three machine cell. But their study is restrictive in the sense that all the parts in the MPS are forced to be processed by same sequence. In the multiple part-types version of our problem we remove this restriction.

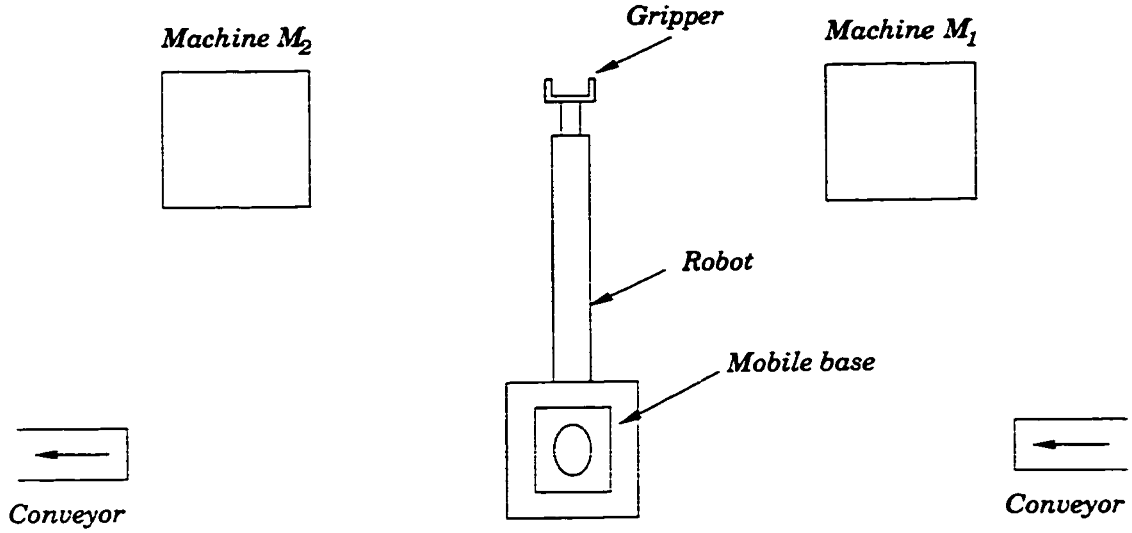

Figure 2: Two-Machine Robotic Cell.

2.5 Models

The two models adopted in this research work are basically two machine robot centered cells. It should be noted that our analysis can be easily extended to other robotic cellular layouts. The extension of this work to a mobile robot cell is as trivial as adding a linear component of time to the robot travel time; which can be combined with rotational component and be considered simply as a unit of time.

Figures 2 and 3 give a typical setup of the robotic cells which are being studied. Figure 2 shows the setup of a robotic cell with out a buffer. It has two machines, a robot which handles the material handling operations and a conveyor. The material handling operations comprise of picking the part from the input, dropping the part at the output, loading and unloading of machines. Input and output are two distinct points on the conveyor. It is assumed that the conveyor always supplies an unlimited number of parts. In other words, the robot does not wait at the input to pick a part and that there will always be a part on the conveyor. This robotic cell is synonymous with a flow shop; i.e., all the parts have the same number as well as sequence of operations. All the parts go to both machines. They first go to machine M_1 and then to machine M_2.

The setup shown in Figure 3 is similar to the one in Figure 2 except that a storage buffer is placed

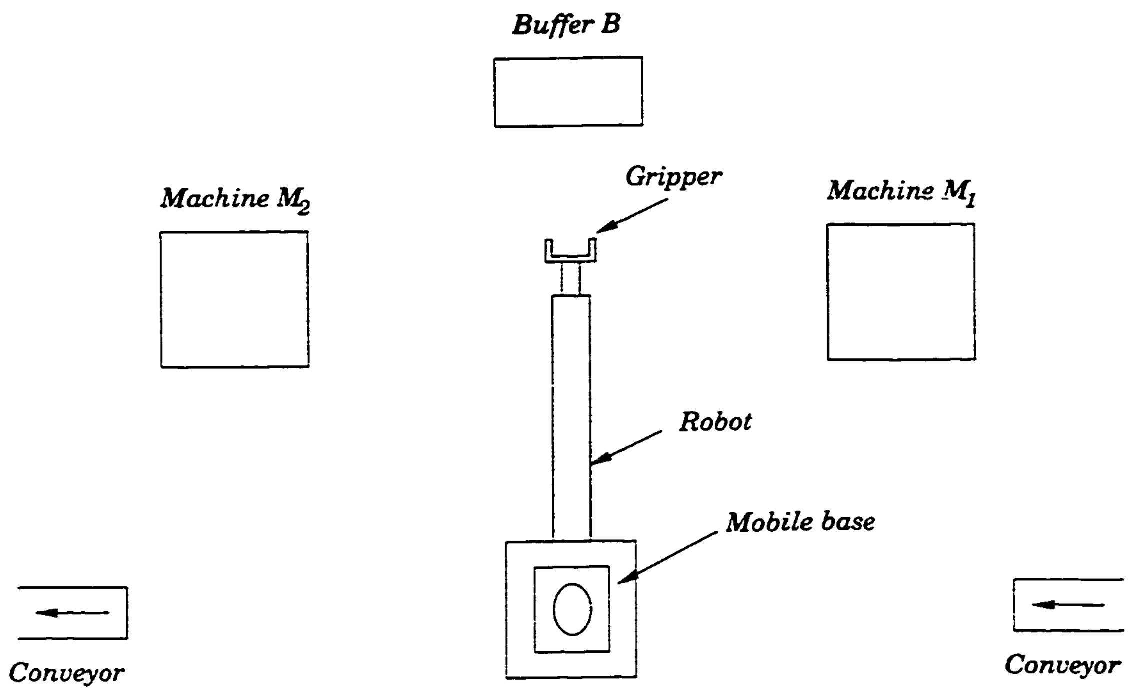

Figure 3: Two-Machine Robotic Cell with a Buffer.

between the machines M_1 and M_2. We place the buffer in such a way that the time taken to move from M_1 to the buffer is same as the time taken to move from M_2 to the buffer. We observed that the buffer could be placed anywhere between the machines and its specificity of location between the machines would not affect the cycle time. The capacity of the buffer is assumed to be one part.

2.6 Assumptions

Certain assumptions have been made in the set-up of the robotic cell model being considered, such as:

1. Time to load a machine is same as to unload the machine.
2. Time to pick-up a part is same as to drop the part.
3. Robot does not wait at the conveyor. There is always a part available for the robot
4. Preemption of any operation on either of the machines is not allowed.

5. Input I, machines M_1 and M_2, buffer B and the output O, are located on the arc of a circle with the robot at the center of the circle.

6. Input and output nodes, and the buffer are treated as machines separated by a distance.

7. Any part in the cell is always either on one of the machines, or in the buffer or being handled by the robot.

Chapter 3

Optimal Robot Sequences for a Two-Machine Cell with Single Part Type

---

Scheduling of robot moves in a two machine manufacturing cell, producing a family of identical parts is studied to investigate the possible reduction in cycle time with the use of a buffer. State space representation for the system is provided and is later adopted for identification of all feasible sequences. Optimal cycle times are identified under various conditions, and reduction in cycle time with the use of a buffer is quantified.

3.1 Introduction

The increase in the level of automation in manufacturing industries especially in engineering and electronics has brought forth cellular manufacturing and robotics. Various studies have proved that, in a wide variety of industrial settings, material handling within a cell can be accomplished efficiently with the use of robots. Among the different issues to be considered in the use of robotic cells are their design, the scheduling of robotic moves, the sequencing of parts and the need for a buffer between the machines.

In this chapter we address the problem of providing an optimal sequence of robot moves for a two machine robotic cell with a buffer and single part-type. Section 3.2 discusses all the entities of the system and their possible states. In Section 3.3 we provide state space representation for the system. In Section 3.4 we adopt state space approach to identify and describe all the feasible robotic sequences for the two robotic cell models discussed in Chapter 2. We also compute the cycle times for the six cyclic sequences. Finally, in Section 3.5 we use pair-wise comparison to suggest the optimal robotic move cycle with an attempt to minimize cycle time and also to establish the conditions where reduction in cycle time is possible with the help of a buffer.

3.2 System State Representation for a Robotic Cell

System state representation for a robotic cell captures the various possible states of a system. It is imperative to investigate all the entities of the system and their possible states, before proposing a definition. We therefore adopt a state-based representation to model the robotic cell. The objects in the system are Machine M_1, Machine M_2, Input I, Output O, Robot R and Buffer B. The possible states that these entities can be in are as follows.

- Machine M_1 can be occupied by part p or else the machine can be free.
- Machine M_2 can be occupied by part p or else the machine can be free.
- Buffer B can contain only one part at a time. It can either contain a part p or can be empty.

- Robot R can do the following tasks and thus can assume the states as below.
  - Robot loading part p on machine M_1.
  - Robot loading part p on machine M_2.
  - Robot unloading part p from machine M_1.
  - Robot unloading part p from machine M_2.
  - Robot picking up part p from Input I.
  - Robot dropping part p at Output O.
  - Robot picking part p from buffer B.
  - Robot dropping part p in buffer B.

3.3 Formal State Representation

We adapted the state space representation provided by Sethi *et al.* [SSS+92] to suit our models. The following definition models the two machine mobile robot cell with a single buffer and a capacity to store one part. The state is modeled as a 4-tuple ($\mathcal{M}_1^i$, $\mathcal{M}_2^i$, $\mathcal{B}^i$, $\mathcal{R}_r^i$) which encapsulates all the pertinent information, where

- $\mathcal{M}_1^i$: Machine M_1, where i belongs to the set $\{0, p\}$.
  - 0 when M_1 is free and is not occupied by a part.
  - p when M_1 is occupied by a part.
- $\mathcal{M}_2^i$: Machine M_2, where i belongs to the set $\{0, p\}$.
  - 0 when M_2 is free and is not occupied by a part.
  - p when M_2 is occupied by a part.
- $\mathcal{B}^i$: Buffer, where i belongs to the set $\{0, p\}$.
  - 0 when buffer is empty.

- p when buffer contains a part.

- $\mathcal{R}_x^i$: Robot, where
  - $i = p$.
  - x belongs to the set $\{M_1^+, M_1^-, M_2^+, M_2^-, B^+, B^-, I, O\}$.
    * M_1^+ : Robot has just unloaded the part from Machine M_1.
    * M_1^- : Robot has just loaded the part on Machine M_1.
    * M_2^+ : Robot has just unloaded the part from Machine M_2.
    * M_2^- : Robot has just loaded the part on Machine M_2.
    * B^+ : Robot has just picked the part from Buffer.
    * B^- : Robot has just dropped the part in the Buffer.
    * I : Robot picks the part from the Input.
    * O : Robot drops the part in the output.

3.4 Cyclic Sequences

Different cyclic sequences of robot moves are possible for this model. Some of these sequences involve the buffer and some others do not. For the sequences where buffer is not used the problem is reduced to a two machine cell with identical parts studied by Sethi *et al.* [SSS+92]. Hence, the sequences S_1 and S_2 which do not use a buffer are adapted from Sethi *et al.* [SSS+92]. Sethi *et al.* [SSS+92] showed that there are six potentially optimal robot move cycles (S_1,...., S_6) to consider in the three machine cell. By setting the processing time on the second machine in the three machine cell to zero, the three machine problem can be transformed to a two machine cell with a buffer. The second machine in the three machine cell is considered as the buffer in the two machine cell . The loading and unloading operations on the second machine in a three machine cell can be considered as dropping and picking operations of the buffer. Under these conditions the sequences S_1 and S_4 proposed by Sethi *et al.* [SSS+92] are reduced to our S_1 and S_2 respectively. The other four sequences provided by Sethi *et al.* [SSS+92] are equivalent to our sequences S_3, S_4, S_5 and S_6. All

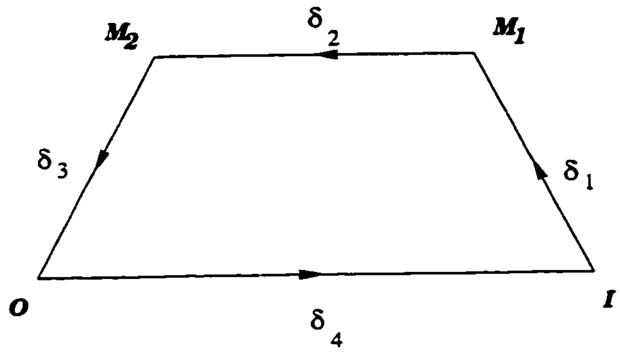

Figure 4: Cyclic Sequence S_1.

the cyclic sequences given below are one unit cycles. State space representation is provided for all these sequences. Initial state of the system is a state when Robot is at the Input and both the machines M_1 and M_2 are idle and the buffer is empty. This initial state can be formally represented as $(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p)$. In the initial state the system is empty. The robotic moves preceding the cycles S_2, S_3, S_4, S_5 and S_6 when the system is empty are given in Appendix A.

3.4.1 S_1 - Cyclic Sequence without a Buffer

The cyclic sequence S_1 comprises of the following robotic moves. Cycle S_1 is illustrated in Figure 4. The sequence along with the state space representation is given below.

1. Robot travels to the Input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till the part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

4. Robot carries the part to machine M_2 and loads it on M_2:

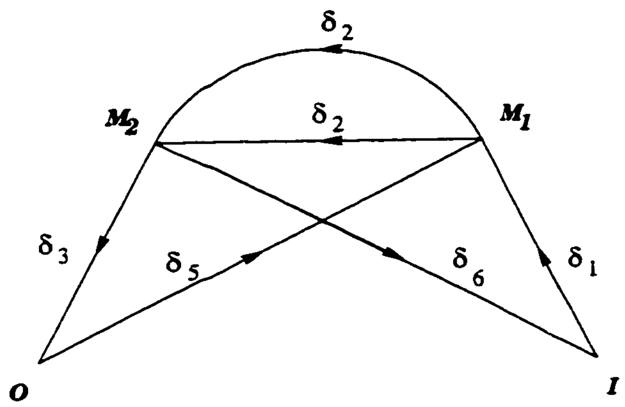

Figure 5: Cyclic Sequence S_2.

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}^p_{M_2^-}).$$

5. Robot waits at machine M_2 till part is processed and unloads it from M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}^p_{M_2^+}).$$

6. Robot carries the part to output, drops the part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}^p_O).$$

3.4.2 S_2 - Cyclic Sequence without a Buffer

The initial steps which precede the sequence S_2 are given in Appendix A. Figure 5 gives a vivid representation of the cyclic sequence S_2. In the Figure 5, the straight line from M_1 to M_2 corresponds to robotic move number 2 and the curved line corresponds to robotic move number 5. The cyclic sequence S_2 consists of the following robotic moves. The sequence along with the formal representation of its system states is given below.

1. Robot travels to M_1, waits if necessary and unloads the part from machine M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}^p_{M_1^+}).$$

2. Robot travels to machine M_2 and loads the part on M_2:

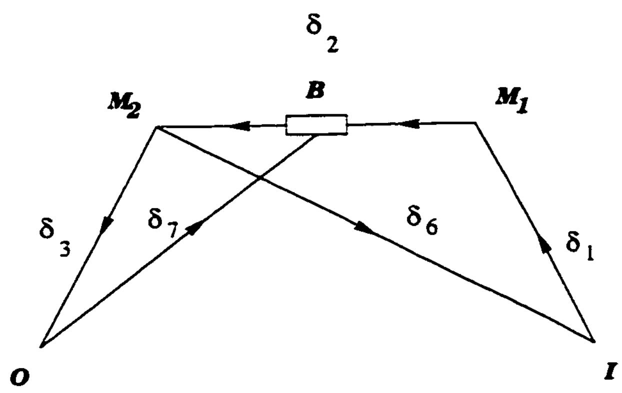

Figure 6: Cyclic Sequence with a Buffer S_3.

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

3. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p).$$

4. Robot travels to M_1 and loads the part on machine M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

5. Robot travels to machine M_2, waits if necessary and unloads the part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

6. Robot carries the part to output and drops it at the output:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_O^p).$$

3.4.3 S_3 - Cyclic Sequence with a Buffer

The initial steps which lead us to this sequence are given in Appendix A. Figure 6 gives a clear representation of the cycle S_3. The cyclic sequence S_3 comprises of the following robotic moves. The buffer comes in to picture in this one unit cyclic sequence. The sequence along with its formal representation is given below.

1. Robot travels to the buffer and picks part from buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B+}^p).$$

2. Robot travels to machine M_2 and loads part on M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

3. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p).$$

4. Robot travels to machine M_1 and loads the part on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

5. Robot waits at machine M_1 until the part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

6. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B-}^p).$$

7. Robot travels to machine M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p).$$

8. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

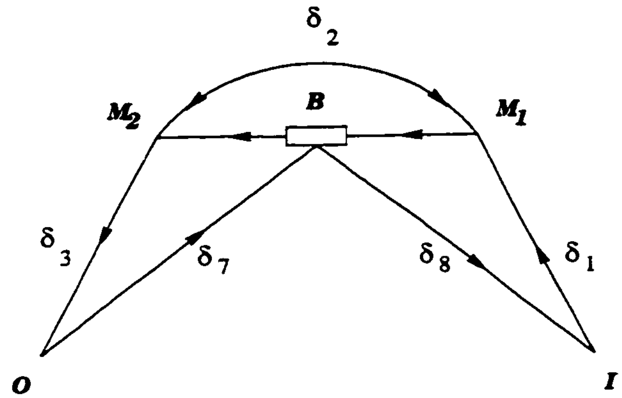

Figure 7: Cyclic Sequence with a Buffer S_4.

3.4.4 S_4 - Cyclic Sequence with a Buffer

The initial steps which precede this sequence are given in Appendix A. This sequence S_4 comprises of the following robotic moves. It also involves the usage of buffer as the sequence S_3. Sequence S_4 along with its formal representation is given below. Figure 7 provides a clear representation of the cyclic sequence S_4.

1. Robot travels to buffer and picks part from buffer:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^-}^p).$$

2. Robot carries part to machine M_2 and loads it on M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

3. Robot travels to M_1, waits if necessary and unloads part from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

4. Robot carries part to the buffer and drops it in the buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

5. Robot travels to the input and picks a part from I:

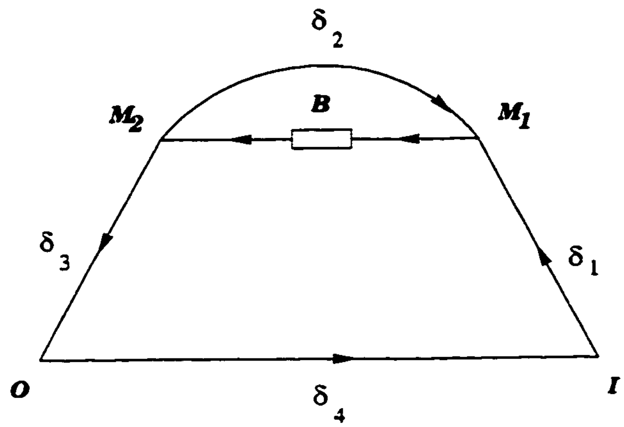

Figure 8: Cyclic Sequence with a Buffer S_5.

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_I^p).$$

6. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

7. Robot travels to M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p).$$

8. Robot travels to the output and drops part at O:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

3.4.5 S_5 - Cyclic Sequence with a Buffer

The initial steps which lead us to this sequence are given in Appendix A. This cyclic sequence S_5 consists of the following robotic moves. Sequence S_5 along with its formal representation is given below. Cyclic sequence S_5 is illustrated in the Figure 8.

1. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_I^p).$$

2. Robot travels to machine M_1 and loads the part on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

3. Robot travels to the buffer and picks the part from buffer:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p) .$$

4. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

5. Robot travels to machine M_1, waits if necessary and unloads the part from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

6. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

7. Robot travels to machine M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p).$$

8. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

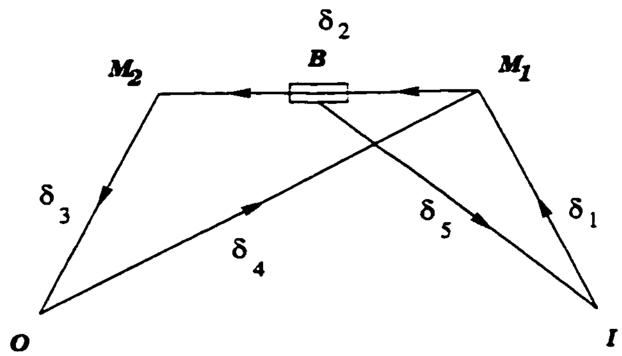

Figure 9: Cyclic Sequence with a Buffer S_6.

3.4.6 S_6 - Cyclic Sequence with a Buffer

The initial steps which lead us to this sequence are given in Appendix A. This cyclic sequence S_6 consists of the following robotic moves. Sequence S_6 along with its formal representation is given below. Figure 9 illustrates the cycle S_6.

1. Robot travels to machine M_1, waits if necessary and unloads the part from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

2. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

3. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_I^p).$$

4. Robot travels to machine M_1 and loads the part on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

5. Robot travels to the buffer and picks the part from buffer:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p).$$

6. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

7. Robot waits at machine M_2 till the part is processed and unloads part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^+}^p).$$

8. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{\mathcal{O}}^p).$$

3.4.7 Cycle Times of the Sequences

To make a decision as to which of the above described sequences are optimal, we need to compute their cycle times. The optimality depends on the cycle time; that is the cyclic sequence having the minimum cycle time would be the optimal sequence. Cycle times of the sequences S_1, S_2, S_3, S_4, S_5 and S_6 are represented by T_1, T_2, T_3, T_4, T_5 and T_6 respectively, are explicitly computed below.

Notation

The following notation is used to describe the robotic cell.

- Loading as well as unloading time for the part on either of the machines, M_1 and M_2 is given by ε.
- Dropping time as well as picking time for the part is given by γ.
- α is the processing time of the part on machine M_1.
- β is the processing time of the part on machine M_2.
- δ_1 is the time taken by the robot to travel from Input I to Machine M_1.
- δ_2 is the time taken by the robot to travel from Machine M_1 to Machine M_2.

- δ_3 is the time taken by the robot to travel from Machine M_2 to the Output O.
- δ_4 is the time taken by the robot to travel from Output O to Input I.
- δ_5 is the time taken by the robot to travel from Output O to Machine M_1.
- δ_6 is the time taken by the robot to travel from Machine M_2 to Input I.
- δ_7 is the time taken by the robot to travel from Output O to Buffer B.
- δ_8 is the time taken by the robot to travel from Buffer B to Input I.

Computation

To compute the cycle time T_1 of the sequence S_1, we follow S_1 from state $E = (\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^P)$ at time zero to time $t(E)$ when the system returns to state E the first time. Thus,

$$T_1 = \delta_4 + \gamma + \delta_1 + \varepsilon + \alpha + \varepsilon + \delta_2 + \varepsilon + \beta + \varepsilon + \delta_3 + \gamma = 2(2\varepsilon + \gamma) + \delta_1 + \delta_2 + \delta_3 + \delta_4 + \alpha + \beta.$$

Similarly, we can compute the cycle times of the remaining five sequences. Hence, the expressions of T_2, T_3, T_4, T_5 and T_6 are as follows.

- $T_2 = 4\varepsilon + 2\gamma + 2\delta_2 + \delta_1 + \delta_3 + \delta_5 + \delta_6 + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

$$w_1 = max(0, \beta - (\delta_1 + \delta_2 + \delta_6 + \varepsilon + \gamma))$$
$$w_2 = max(0, \alpha - (\delta_2 + \delta_3 + \delta_5 + \varepsilon + \gamma + w_1))$$

- $T_3 = 4(\varepsilon + \gamma) + 3\delta_2/2 + \delta_1 + \delta_3 + \delta_6 + \delta_7 + \alpha + w_1$, where w_1 is the waiting time and is given by the expression

$$w_1 = max(0, \beta - (\delta_1 + \delta_2 + \delta_6 + 2\varepsilon + 2\gamma + \alpha))$$

- $T_4 = 4(\varepsilon + \gamma) + 3\delta_2 + \delta_1 + \delta_3 + \delta_7 + \delta_8) + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

$$w_1 = max(0, \alpha - (5\delta_2/2 + \delta_3 + \delta_7 + 2\varepsilon + 2\gamma + w_2))$$

$$w_2 = max(0,\ \beta - (\delta_1 + 5\delta_2/2 + \delta_8 + 2\varepsilon + 2\gamma + w_1))$$

- $T_5 = 4(\varepsilon+\gamma) + 3\delta_2 + \delta_1 + \delta_3 + \delta_4 + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

$$w_1 = max(0,\ \alpha - (2\delta_2 + \varepsilon + \gamma))$$
$$w_2 = max(0,\ \beta - (2\delta_2 + \varepsilon + \gamma + w_1))$$

- $T_6 = 4(\varepsilon+\gamma) + 3\delta_2/2 + \delta_1 + \delta_3 + \delta_5 + \delta_8 + \beta + w_1$, where w_1 is waiting time and is given by the expression

$$w_1 = max(0,\ \alpha - (\delta_2 + \delta_3 + \delta_5 + 2\varepsilon + 2\gamma + \beta))$$

Assumptions

To simplify this considerably complex analysis, we make the following assumptions.

- We assume all the travel times to be equal; that is $\delta_1 = \delta_2 = \delta_3 = \delta_4 = \delta_5 = \delta_6 = \delta_7 = \delta_8 = \delta$
- We assume that the loading, unloading, dropping and picking times are equal; that is $\gamma = \varepsilon$

Lemma 1

This lemma is used in the computation of cycle time S_4.

Given

$$w_1 = max(0,\ \Delta_1 - w_2) \quad \cdots\cdots\cdots\cdots \quad (1)$$

$$w_2 = max(0,\ \Delta_2 - w_1) \quad \cdots\cdots\cdots\cdots \quad (2)$$

We can conclude

- if $\Delta_2 > \Delta_1$, then $w_1 = 0$,
- if $\Delta_1 > \Delta_2$, then $w_2 = 0$, and
- if $\Delta_1 = \Delta_2$, then either $w_1 = 0$ or $w_2 = 0$.

Proof of Lemma 1

Adding w_2 to expression (1) we have

$$w_1 + w_2 = max(w_2, \Delta_1).$$

If $\Delta_1 < w_2$, then $w_1 + w_2 = w_2$. Consequently,

$$\text{if } w_2 > \Delta_1, \text{ then } w_1 = 0, \quad \cdots\cdots\cdots\cdots \quad (1.1)$$

$$\text{if } w_2 < \Delta_1, \text{ then } w_1 + w_2 = \Delta_1. \quad \cdots\cdots\cdots\cdots \quad (1.2)$$

Consider expression (2). On substituting $w_1 = 0$ in (2) we have

$$w_2 = max(0, \Delta_2) \text{ when } w_2 > \Delta_1.$$

If $\Delta_2 > 0$, then $w_2 = \Delta_2$.

On substituting $w_2 = \Delta_2$ in (1.1), we have the following important result:

$$\text{if } \Delta_2 > \Delta_1, \text{ then } w_1 = 0.$$

Adding w_1 to expression (2) we have

$$w_2 + w_1 = max(w_1, \Delta_2).$$

If $\Delta_2 < w_1$, then $w_2 + w_1 = w_1$. Consequently,

$$\text{if } w_1 > \Delta_2, \text{ then } w_2 = 0, \quad \cdots\cdots\cdots\cdots \quad (2.1)$$

$$\text{if } w_1 < \Delta_2, \text{ then } w_2 + w_1 = \Delta_2. \quad \cdots\cdots\cdots\cdots \quad (2.2)$$

Consider expression (1). On substituting $w_2 = 0$ in (1) we have,

$$w_1 = max(0, \Delta_1) \text{ when } w_1 > \Delta_2.$$

If $\Delta_1 > 0$, then $w_1 = \Delta_1$.

On substituting $w_1 = \Delta_1$ in (2.1), we have the following important result:

$$\text{if } \Delta_1 > \Delta_2, \text{ then } w_2 = 0.$$

If $\Delta_1 = \Delta_2 = \Delta$, then expressions (1) and (2) can be written as

$$w_1 = max(0, \Delta - w_2), \quad \cdots\cdots\cdots\cdots \quad (1)$$

$$w_2 = max(0, \Delta - w_1). \quad \cdots\cdots\cdots\cdots \quad (2)$$

For any w_1 and w_2,

$w_1 = \Delta - w_2$ if $\Delta \geq w_2$, and

$w_2 = \Delta - w_1$ if $\Delta \geq w_1$.

Adding expressions (1) and (2), we have

$$w_1 + w_2 = max(0, \Delta - w_2, \Delta - w_1, 2\Delta - w_2 - w_1).$$

Assuming Δ is large, we have

$$w_1 + w_2 = 2\Delta - w_2 - w_1 \implies$$

$$w_1 + w_2 = \Delta.$$

If $w_1 = w_2$, then $w_1 = w_2 = \Delta/2$.

$w_1 + w_2 = \Delta$ can be rewritten as

$$w_1 + w_2 = w_2 + x$$

because $\Delta \geq w_2$, where $x \geq 0 \implies$

$$w_1 = x \implies$$

$$w_1 \geq 0.$$

Similarly,

$$w_1 + w_2 = w_1 + y$$

because $\Delta \geq w_1$, where $y \geq 0 \implies$

$$w_2 = y \implies$$

$$w_2 \geq 0.$$

Hence, when $\Delta_1 = \Delta_2 = \Delta$, either w_1 can be ≥ 0, or w_2 can be ≥ 0, as long as $w_1 + w_2 = \Delta$.

Cycle Time Expressions

The expressions for the cycle times are now reduced to

- $T_1 = 6\varepsilon + 4\delta + \alpha + \beta$
- $T_2 = 6\varepsilon + 6\delta + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

  $$w_1 = max(0,\ \alpha - (3\delta + 2\varepsilon + w_2))$$

  $$w_2 = max(0,\ \beta - (3\delta + 2\varepsilon))$$

  The expression for T_2 can be evaluated as below

  $$T_2 = \begin{cases} 6\varepsilon + 6\delta & \text{if } \beta < 3\delta + 2\varepsilon, \alpha < 3\delta + 2\varepsilon \\ 4\varepsilon + 3\delta + \alpha & \text{if } \beta < 3\delta + 2\varepsilon, \alpha > 3\delta + 2\varepsilon \\ 4\varepsilon + 3\delta + \beta & \text{if } \beta > 3\delta + 2\varepsilon, \alpha < \beta \\ 4\varepsilon + 3\delta + \alpha & \text{if } \alpha > \beta, \beta > 3\delta + 2\varepsilon \end{cases}$$

- $T_3 = 8\varepsilon + 5.5\delta + \alpha + w_1$, where w_1 is the waiting time and is given by the expression

  $$w_1 = max(0,\ \beta - (3\delta + 4\varepsilon + \alpha))$$

  The expression for T_3 can be evaluated as below

  $$T_3 = \begin{cases} 8\varepsilon + 5.5\delta + \alpha & \text{if } \beta < 3\delta + 4\varepsilon + \alpha \\ 4\varepsilon + 2.5\delta + \beta & \text{if } \beta > 3\delta + 4\varepsilon + \alpha \end{cases}$$

- $T_4 = 8\varepsilon + 7\delta + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

  $$w_1 = max(0,\ \alpha - (4.5\delta + 4\varepsilon + w_2)) = max(0,\ \Delta_1 - w_2)$$

  $$w_2 = max(0,\ \beta - (4.5\delta + 4\varepsilon + w_1))$$

  w_1 and w_2 can be rewritten as follows

  $$w_1 = max(0,\ \Delta_1 - w_2)$$

  $$w_2 = max(0,\ \Delta_2 - w_1)$$

where $\Delta_1 = \alpha - (4.5\delta + 4\varepsilon)$

$\Delta_2 = \beta - (4.5\delta + 4\varepsilon)$

According to Lemma 1, if $\Delta_1 > \Delta_2$ then $w_2 = 0$ else $w_1 = 0$.

Therefore, the expression for T_4 can be evaluated as below

$$T_4 = \begin{cases} 8\varepsilon + 7\delta & \text{if } \alpha < 4.5\delta + 4\varepsilon, \beta < 4.5\delta + 4\varepsilon \\ 4\varepsilon + 2.5\delta + \alpha & \text{if } \alpha > \beta, \alpha > 4.5\delta + 4\varepsilon \\ 4\varepsilon + 2.5\delta + \beta & \text{if } \beta > \alpha, \beta > 4.5\delta + 4\varepsilon \end{cases}$$

- $T_5 = 8\varepsilon + 6\delta + w_1 + w_2$, where w_1 and w_2 are waiting times and are given by these expressions.

  $w_1 = max(0,\ \alpha - (2\delta + 2\varepsilon))$

  $w_2 = max(0,\ \beta - (2\delta + 2\varepsilon + w_1))$

  The expression for T_5 can be evaluated as below

$$T_5 = \begin{cases} 8\varepsilon + 6\delta & \text{if } \alpha < 2\delta + 2\varepsilon, \beta < 2\delta + 2\varepsilon \\ 6\varepsilon + 4\delta + \beta & \text{if } \alpha < 2\delta + 2\varepsilon, \beta > 2\delta + 2\varepsilon \\ 6\varepsilon + 4\delta + \alpha & \text{if } \alpha > 2\delta + 2\varepsilon, \beta < \alpha \\ 6\varepsilon + 4\delta + \beta & \text{if } \alpha > 2\delta + 2\varepsilon, \beta > \alpha \end{cases}$$

- $T_6 = 8\varepsilon + 5.5\delta + \beta + w_1$, where w_1 is the waiting time and is given by the expression

  $w_1 = max(0,\ \alpha - (3\delta + 4\varepsilon + \beta))$

  The expression for T_6 can be evaluated as below

$$T_6 = \begin{cases} 8\varepsilon + 5.5\delta + \beta & \text{if } \alpha < 3\delta + 4\varepsilon + \beta \\ 4\varepsilon + 2.5\delta + \alpha & \text{if } \alpha > 3\delta + 4\varepsilon + \beta \end{cases}$$

3.5 Analysis

Cycle times T_1, T_2, T_3, T_4, T_5 and T_6 are found to depend on the factors ε, δ, α and β. These cycle times assume different expressions for distinct values of α and β. Among the sequences S_1, S_2, S_3, S_4, S_5 and S_6, which sequence is optimal depends on which of the respective cycle times T_1, T_2,

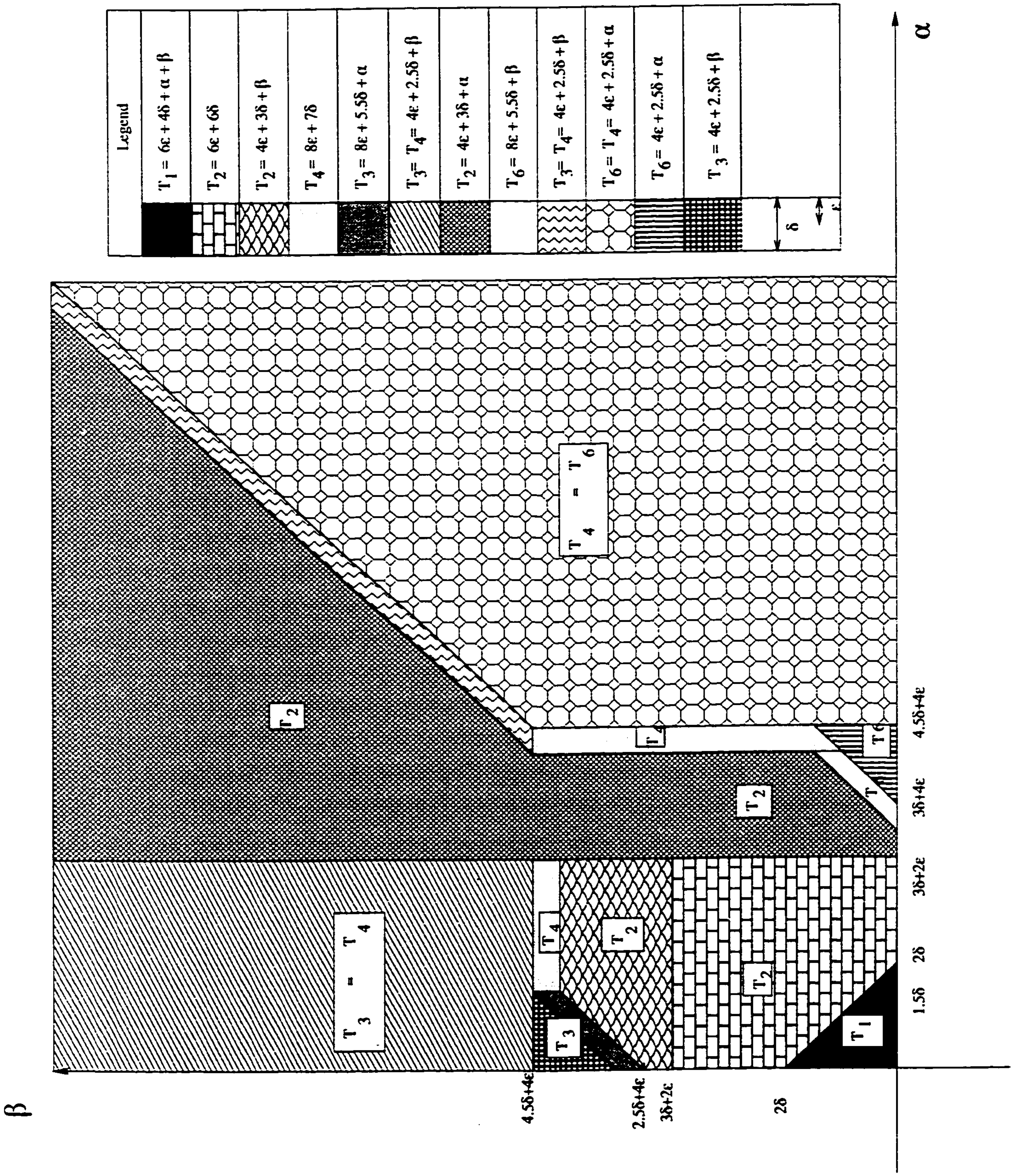

Figure 10: Optimal Cycle Times.

T_3, T_4, T_5 and T_6 is minimal. Hence, the cycle time of each sequence is compared with the cycle time of every other feasible sequence under same conditions to establish the optimal cycle under a given set of conditions. Sethi *et al.* [SSS+92] showed that two cycles are dominated by the other four cycles. We try to see if S_5 and S_2 are dominated by the other four cycles. It is interesting to note that the sequence S_5 is not optimal under any of these conditions and is dominated by the other sequences. But the same is not true for S_2. The comparison between these cycle times to establish optimal sequences under various conditions is better illustrated graphically in Figure 10.

It can be observed in Figure 10 that different sequences are optimal under different conditions. Meaningful conclusions can be drawn from Figure 10 and the results can be summarized as follows.

- Sequence S_1 with cycle time $T_1 = 6\epsilon + 4\delta + \alpha + \beta$ is optimal when the condition

  $$\alpha + \beta \leq 2\delta$$

  is satisfied.

- Sequence S_2 with cycle time $T_2 = 6\epsilon + 6\delta$ is optimal when the conditions

  $$\alpha + \beta \geq 2\delta, \beta \leq 3\delta + 2\epsilon \text{ and } \alpha \leq 3\delta + 2\epsilon$$

  are satisfied.

- Sequence S_2 with cycle time $T_2 = 4\epsilon + 3\delta + \beta$ is optimal when the conditions

  $$\beta \geq 3\delta + 2\epsilon, \alpha \leq 3\delta + 2\epsilon, \beta \leq 2.5\delta + 4\epsilon + \alpha \text{ and } \beta \leq 4\delta + 4\epsilon$$

  are satisfied.

- Sequence S_2 with cycle time $T_2 = 4\epsilon + 3\delta + \alpha$ is optimal when the conditions

  $$\alpha \geq 3\delta + 2\epsilon, \alpha \leq 2.5\delta + 4\epsilon + \beta, \alpha \leq 4\delta + 4\epsilon \text{ and } \beta \leq 4.5\delta + 4\epsilon$$

  are satisfied.

- Sequence S_2 with cycle time $T_2 = 4\epsilon + 3\delta + \alpha$ is optimal when the conditions

$$\alpha \geq 3\delta + 2\varepsilon,\ \beta \geq 0.5\delta + \alpha \text{ and } \beta \geq 4.5\delta + 4\varepsilon$$

are satisfied.

- Sequence S_3 with cycle time $T_3 = 8\varepsilon + 5.5\delta + \alpha$ is optimal when the conditions

$$\alpha \leq 1.5\delta,\ \beta \leq 3\delta + 4\varepsilon + \alpha \text{ and } \beta \geq 2.5\delta + 4\varepsilon + \alpha$$

are satisfied.

- Sequence S_3 with cycle time $T_3 = 4\varepsilon + 2.5\delta + \beta$ is optimal when the conditions

$$\alpha \leq 1.5\delta,\ \beta \geq 3\delta + 4\varepsilon + \alpha \text{ and } \beta \leq 4.5\delta + 4\varepsilon$$

are satisfied.

- Sequences S_3 and S_4 with cycle time $T_3 = T_4 = 4\varepsilon + 2.5\delta + \beta$ are optimal when the conditions

$$\alpha \leq 3\delta + 2\varepsilon \text{ and } \beta \geq 4.5\delta + 4\varepsilon$$

are satisfied. There is a saving of 0.5δ over S_2 with either S_3 or S_4, under these conditions.

- Sequences S_3 and S_4 with cycle time $T_3 = T_4 = 4\varepsilon + 2.5\delta + \beta$ are optimal when the conditions

$$\beta \geq \alpha,\ \beta \leq 0.5\delta + \alpha \text{ and } \beta \geq 4.5\delta + 4\varepsilon$$

are satisfied. There is a saving over S_2 which increases linearly from 0 to 0.5δ between the boundaries $\beta \leq 0.5\delta + \alpha$ and $\beta \geq \alpha$ with either S_3 or S_4, under these conditions.

- Sequence S_4 with cycle time $T_4 = 8\varepsilon + 7\delta$ is optimal when the conditions

$$\beta \geq 4\varepsilon + 4\delta,\ \alpha \geq 1.5\delta,\ \alpha \leq 3\delta + 2\varepsilon \text{ and } \beta \leq 4.5\delta + 4\varepsilon$$

are satisfied. There is a saving over S_2 which increases linearly from 0 to 0.5δ between the boundaries $\beta \geq 4\varepsilon + 4\delta$ and $\beta \leq 4.5\delta + 4\varepsilon$ with S_4, under these conditions.

- Sequence S_4 with cycle time $T_4 = 8\varepsilon + 7\delta$ is optimal when the conditions

$$\beta \leq 4\varepsilon + 4.5\delta \text{ , } \alpha \geq 4\delta + 4\varepsilon, \alpha \leq 4.5\delta + 4\varepsilon \text{ and } \alpha \leq 3\delta + 4\varepsilon + \beta$$

are satisfied. There is a saving over S_2 which increases linearly from 0 to 0.5δ between the boundaries $\alpha \geq 4\delta + 4\varepsilon$ and $\alpha \leq 4.5\delta + 4\varepsilon$ with S_4, under these conditions.

- Sequences S_4 and S_6 with cycle time $T_4 = T_6 = 4\varepsilon + 2.5\delta + \alpha$ are optimal when the conditions

  $$\alpha \geq \beta \text{ and } \alpha \geq 4.5\delta + 4\varepsilon$$

  are satisfied. There is a saving of 0.5δ over S_2 with either S_4 or S_6, under these conditions.

- Sequence S_6 with cycle time $T_6 = 4\varepsilon + 2.5\delta + \alpha$ is optimal when the conditions

  $$\alpha \geq 3\delta + 4\varepsilon + \beta \text{ and } \alpha \leq 4.5\delta + 4\varepsilon$$

  are satisfied. There is a saving of 0.5δ over S_2 with S_6, under these conditions.

- Sequence S_6 with cycle time $T_6 = 8\varepsilon + 5.5\delta + \beta$ is optimal when the conditions

  $$\alpha \leq 3\delta + 4\varepsilon + \beta, \alpha \leq 2.5\delta + 4\varepsilon + \beta \text{ and } \alpha \leq 4\delta + 4\varepsilon$$

  are satisfied.

- Sequences S_1 and S_2 which do not use a buffer are found to perform better for smaller values of processing times (for small values of α and β). So, no reduction in cycle time can be achieved or no advantage can be gained with the help of a buffer when the processing times are small.

- Sequences S_3, S_4 and S_6 which use a buffer are found to perform better than sequences S_1 and S_2 which do not use a buffer under certain conditions. However, the saving achieved with a buffer is not more than 0.5δ.

Chapter 4

Robot Sequences for a Two-Machine Cell with Multiple Part Types

---

Robot move sequencing in a two machine cell with multiple part types and a single buffer is investigated to suggest optimal sequences to process a MPS. All feasible sequences are identified including cyclic sequences, and combination sequences resulting from the switchings from one sequence to another. Cycle times are evaluated using analytical methods. A framework is established to identify the optimal sequences for the parts in a given MPS while preserving the cyclic nature of the schedules.

4.1 Introduction

In this chapter we consider the problem of finding an optimal sequence of robot moves in a two machine robotic cell with a buffer and multiple part-types. This problem is different from a two machine single part-type problem in the sense that optimal solution may not be given by a single sequence. The cycle time minimization problem in the multiple part-types case then becomes one of deciding which sequence to be used for processing each part and determining how to switch from one cycle to another. Section 4.2 introduces combination sequences which are a result of switchings from one sequence to another. Section 4.3 outlines the common states between the feasible sequences. Section 4.4 describes robotic moves and computation of sequence times for these combination sequences and outlines the expressions for cycle times of cyclic sequences. Section 4.5 outlines the preceding and succeeding sequences for each one of these feasible sequences and illustrates the sequence tree. Section 4.6 describes the constraints imposed to preserve the cyclic nature of the schedules.

4.2 Combination Sequences

Finding an optimal sequence in a multiple part type problem is different from that of a single part type in some ways. In a single part type problem, we find a sequence with minimum cycle time among the feasible sequences and repeatedly use that sequence to process the parts. However in a multiple part type problem, a single sequence may not be optimal for all the parts. There may be a different optimal cycle for each part. So, while processing different parts in a MPS we need to switch from one cycle to another as originally proposed by Kamoun [Kam94] in his study of two machine robotic cell problem with multiple part types. The starting and the end states of one sequence may not be same as another sequence. So after completing one cycle, we cannot start another sequence whose starting state is different from that of its previous sequence without some wasteful robotic moves. These moves defy the cyclic nature of these schedules. So, if we want to switch from one sequence to another it is appropriate to switch in a state which is common to both the sequences. This gives rise to combination sequences which are a result of switches from one

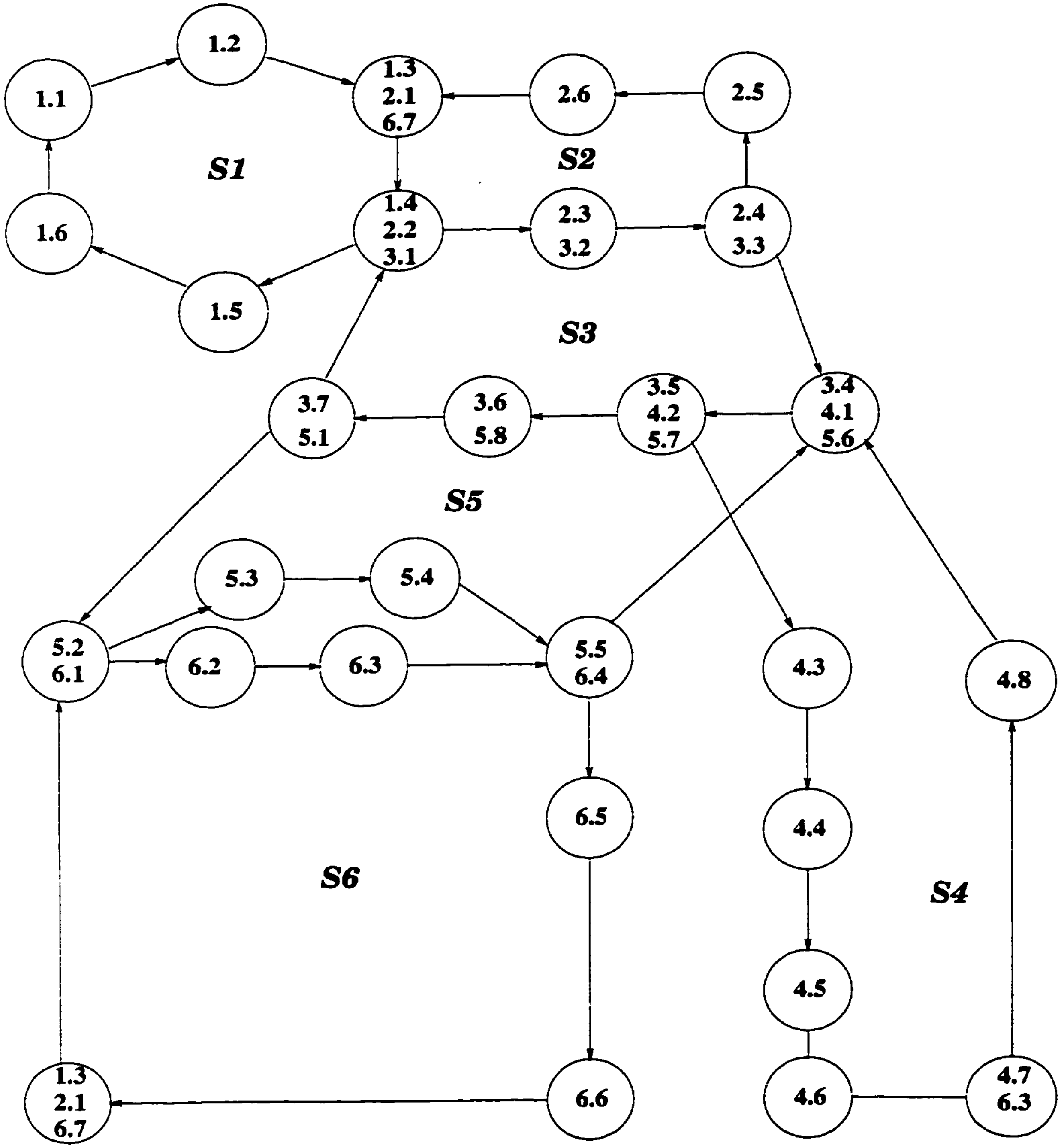

Figure 11: State Transition Diagram for a Two-Machine Robotic Cell with a Buffer.

State	State Space Representation
1.1	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p)$
1.2	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p)$
1.3, 2.1, 6.7	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p)$
1.4, 2.2, 3.1	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$
1.5	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^+}^p)$
1.6	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_O^p)$
2.3, 3.2	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p)$
2.4, 3.3	$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p)$
2.5	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^+}^p)$
2.6	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_O^p)$
3.4, 4.1, 5.6	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p)$
3.5, 4.2, 5.7	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p)$
3.6, 5.8	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p)$
3.7, 5.1	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p)$
4.3	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_I^p)$
4.4	$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p)$
4.5	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p)$
4.6	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p)$
4.7, 6.3	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p)$
4.8	$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$
5.2, 6.1	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_I^p)$
5.3	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p)$
5.4	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p)$
5.5, 6.4	$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$
6.2	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p)$
6.5	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^+}^p)$

Table 1: Different States in the Cycles and their Representation.

sequence to another in their common state. These combination sequences are one unit sequences, but they are not cyclic.

The common states between the cycles are explicitly shown in Figure 11. For instance, the state represented by 1.4, 2.2 and 3.1 is common to the sequences S_1, S_2 and S_3. The 17 combination sequences can be easily identified in Figure 11. The numbers 1.1, 1.2, ..., 6.6 etc. are the different states in the respective sequences. The prefix 1, 2, ..., 6 indicates the sequence number. For example, the number 1 in 1.2 indicates sequence S_1 and the number 2 indicates a state in S_1. The legend for Figure 11, comprising of the state space representation for the different states shown in Figure 11 is provided in Table 1. It is to be noted that the node 6.3 appears twice in Figure 11. Thus, we have 17 combination sequences and 6 basic cyclic sequences which makes the total number of feasible sequences 23. The combination sequences S_{12} and S_{21} are adopted from Kamoun's two machine cell, multiple part type analysis [Kam94].

4.3 Common States

A common state between any two sequences is a state which exists in both the sequences and in which state these sequences can switch from one to another without any wasteful robotic moves and thus preserving the cyclic nature of the schedules. Table 2 gives the cyclic sequences, their common states and their potential combination sequences. We observe that there are 10 common states which should give us ideally 20 combination sequences.

For instance the common state between S_1 and S_2 is ($\mathcal{M}_1^0$, $\mathcal{M}_2^p$, $\mathcal{B}^0$, $\mathcal{R}_{M_2^-}^p$), where Machine M_1 is empty, Machine M_2 is loaded, Buffer B is empty and Robot R has just loaded a part on M_2. There can be a switching from S_1 to S_2 and S_2 to S_1 in this state which results in the two combination sequence S_{12} and S_{21} respectively. To help comprehend the combination sequences we attempt to write the robotic moves for these combination sequences and also compute their cycle time. i is the position of the part in the sequence of parts in the MPS.

Sequences	Common State	Combination Sequences
S_1 and S_2	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$	S_{12} and S_{21}
S_1 and S_3	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$	S_{13} and S_{31}
S_2 and S_3	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p)$	S_{23} and S_{32}
S_3 and S_4	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p)$	S_{34} and S_{43}
S_1 and S_6	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p)$	S_{16} and S_{61}
S_2 and S_6	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p)$	S_{26} and S_{62}
S_4 and S_6	$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p)$	S_{46} and S_{64}
S_3 and S_5	$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p)$	S_{35} and S_{53}
S_4 and S_5	$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p)$	S_{45} and S_{54}
S_5 and S_6	$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p)$	S_{45} and S_{54}

Table 2: Common States.

4.4 Combination Sequences and Sequence Times

Notation

The notation used in Chapter 3 to describe the cell data is still applicable here. However, keeping in consideration the multiple part-type scenario we need to define a few additional parameters which we would be using in our computation of sequence times and cycle times.

- α_i is the processing time of the i^{th} part on machine M_1.
- β_i is the processing time of the i^{th} part on machine M_2.

- i is the position of the part in the sequence of parts or in the MPS. i belongs to $1, 2, \ldots n$ where, n = total number of parts in a MPS.

Sequence S_{12}

- Sequence S_{12} involves the following robotic moves.

  1. Robot travels to the Input and picks up part i.
  2. Robot carries the part to machine M_1 and loads it on M_1.
  3. Robot waits at machine M_1 till the part is processed and unloads it from M_1.
  4. Robot carries the part to machine M_2 and loads it on M_2.
  5. Robot travels to the input and picks up part $i+1$.
  6. Robot travels to M_1 and loads the part on machine M_1.
  7. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  8. Robot carries the part to output and drops it at the output.

- Sequence time of S_{12}:

$$T_{12} = 8\varepsilon + 7\delta + \alpha_i + w_1,$$

where $w_1 = max(0, \beta_i - (3\delta + 2\varepsilon))$.

Sequence S_{21}

- Sequence S_{21} involves the following robotic moves.

  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot carries the part to machine M_2 and loads it on M_2.
  3. Robot waits at machine M_2 till the part is processed and unloads it from M_2.
  4. Robot carries the part to output and drops it at the output.

- Sequence time of S_{21}:

$$T_{21} = 4\varepsilon + 3\delta + \beta_i + w_1,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 2\varepsilon + w_2))$,

and $w_2 = max(0,\ \beta_{i-1} - (3\delta + 2\varepsilon))$.

Sequence S_{13}

- Sequence S_{13} involves the following robotic moves.

  1. Robot travels to the Input and picks up part i.
  2. Robot carries the part to machine M_1 and loads it on M_1.
  3. Robot waits at machine M_1 till the part is processed and unloads it from M_1.
  4. Robot carries the part to machine M_2 and loads it on M_2.
  5. Robot travels to the input and picks up part $i+1$.
  6. Robot travels to machine M_1 and loads the part on M_1.
  7. Robot waits at machine M_1 until the part is processed and unloads it from M_1.
  8. Robot travels to the buffer and drops the part in buffer.
  9. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  10. Robot travels to the output and drops part at the output.

- Sequence time of S_{13}:

$$T_{13} = 10\varepsilon + 7\delta + \alpha_i + \alpha_{i+1} + w_1,$$

where $w_1 = max(0,\ \beta_i - (3\delta + 4\varepsilon + \alpha_{i+1}))$.

Sequence S_{31}

- Sequence S_{31} involves the following robotic moves.

1. Robot travels to the buffer and picks part i from buffer.
2. Robot travels to machine M_2 and loads part on M_2.
3. Robot waits at machine M_2 till part is processed and unloads it from M_2.
4. Robot carries the part to output, drops the part.

- Sequence time of S_{31}:

$$T_{31} = 4\varepsilon + 2.5\delta + \beta_i.$$

Sequence $\mathbf{S_{16}}$

- Sequence S_{16} involves the following robotic moves.

1. Robot travels to the Input and picks up part i.
2. Robot carries the part to machine M_1 and loads it on M_1.
3. Robot waits at machine M_1 till the part is processed and unloads it from M_1.
4. Robot travels to the buffer and drops the part in buffer.
5. Robot travels to the input and picks up part $i+1$.
6. Robot travels to machine M_1 and loads the part on M_1.
7. Robot travels to the buffer and picks part i from buffer.
8. Robot travels to machine M_2 and loads the part on M_2.
9. Robot waits at machine M_2 till the part is processed and unloads part from M_2.
10. Robot travels to the output and drops part at the output.

- Sequence time of S_{16}:

$$T_{16} = 10\varepsilon + 6.5\delta + \alpha_i + \beta_i.$$

Sequence $\mathbf{S_{61}}$

- Sequence S_{61} involves the following robotic moves.

  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot carries the part to machine M_2 and loads it on M_2.
  3. Robot waits at machine M_2 till part is processed and unloads it from M_2.
  4. Robot carries the part to output, drops the part.

- Sequence time of S_{61}:

$$T_{61} = 4\varepsilon + 3\delta + \beta_i + w_1,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 4\varepsilon + \beta_{i-1}))$.

Sequence S_{23}

- Sequence S_{23} involves the following robotic moves.

  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot carries the part to machine M_2 and loads it on M_2.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to M_1 and loads the part on machine M_1.
  5. Robot waits at machine M_1 until the part is processed and unloads it from M_1.
  6. Robot travels to the buffer and drops the part in buffer.
  7. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  8. Robot travels to the output and drops part at the output.

- Sequence time of S_{23}:

$$T_{23} = 8\varepsilon + 6\delta + \alpha_{i+1} + w_1 + w_3,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 2\varepsilon + w_2))$,
$w_2 = max(0,\ \beta_{i-1} - (3\delta + 2\varepsilon))$,
and $w_3 = max(0,\ \beta_i - (3\delta + 4\varepsilon + \alpha_{i+1}))$.

Sequence S_{32}

- Sequence S_{32} involves the following robotic moves.

  1. Robot travels to the buffer and picks part i from buffer.
  2. Robot travels to machine M_2 and loads part on M_2.
  3. Robot travels to the input and picks up part $i \div$.
  4. Robot travels to machine M_1 and loads the part on M_1.
  5. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  6. Robot travels to the output and drops part at the output.

- Sequence time of S_{32}:

$$T_{32} = 6\varepsilon + 5.5\delta + w_1,$$

where $w_1 = max(0, \beta_i - (3\delta + 2\varepsilon))$.

Sequence S_{26}

- Sequence S_{26} involves the following robotic moves.

  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot travels to the buffer and drops the part in buffer.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to machine M_1 and loads the part on M_1.
  5. Robot travels to the buffer and picks part i from buffer.
  6. Robot travels to machine M_2 and loads the part on M_2.
  7. Robot waits at machine M_2 till the part is processed and unloads part from M_2.
  8. Robot travels to the output and drops part at the output.

- Sequence time of S_{26}:

$$T_{26} = 8\varepsilon + 5.5\delta + \beta_i + w_1,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 2\varepsilon + w_2))$,

and $w_2 = max(0,\ \beta_{i-1} - (3\delta + 2\varepsilon))$.

Sequence S_{62}

- Sequence S_{62} involves the following robotic moves.
  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot travels to machine M_2 and loads the part on M_2.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to M_1 and loads the part on machine M_1.
  5. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  6. Robot carries the part to output and drops it at the output.
- Sequence time of S_{62}:

$$T_{62} = 6\varepsilon + 6\delta + w_2 + w_1,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 4\varepsilon + \beta_{i-1}))$,

and $w_2 = max(0,\ \beta_i - (3\delta + 2\varepsilon))$.

Sequence S_{34}

- Sequence S_{34} involves the following robotic moves.
  1. Robot travels to the buffer and picks part i from buffer.
  2. Robot travels to machine M_2 and loads part on M_2.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to machine M_1 and loads the part on M_1.

5. Robot waits at machine M_1 till the part is processed and unloads it from M_1.
6. Robot carries the part to the buffer and drops it in the buffer.
7. Robot travels to the input and picks up part $i+2$.
8. Robot carries the part to M_1 and loads it on M_1.
9. Robot travels to M_2, waits if necessary and unloads part i from M_2.
10. Robot carries part to the output and drops it at O.

- Sequence time of S_{34}:

$$T_{34} = 10\varepsilon + 8\delta + \alpha_{i+1} + w_1,$$

where $w_1 = max(0, \beta_i - (5.5\delta + 6\varepsilon + \alpha_{i+1}))$.

Sequence S_{43}

- Sequence S_{43} involves the following robotic moves.

1. Robot travels to buffer and picks part i from buffer.
2. Robot carries part to machine M_2 and loads it on M_2.
3. Robot travels to M_1, waits if necessary and unloads part $i+1$ from M_1.
4. Robot carries part to the buffer and drops it in the buffer.
5. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
6. Robot travels to the output and drops part at the output.

- Sequence time of S_{43}:

$$T_{43} = 6\varepsilon + 4.5\delta + w_1 + w_2,$$

where $w_1 = max(0, \alpha_{i+1} - (4.5\delta + 4\varepsilon + w_3))$,

and $w_2 = max(0, \beta_i - (2\delta + 2\varepsilon + w_1))$.

The expression for w_3 which is the waiting time of the robot at M_2 before unloading $i-1^{th}$ part, and thus the value of T_{43} depends on the sequence preceding S_{43}. If S_{34} precedes S_{43}, then w_3 is given by the expression

$$w_3 = max(0, \beta_{i-1} - (5.5\delta + 6\epsilon + \alpha_i)).$$

If S_{54} or S_{64} precedes S_{43}, then w_3 is given by the expression

$$w_3 = max(0, \beta_{i-1} - (4.5\delta + 4\epsilon + w_4)),$$

where $w_4 = max(0, \alpha_i - (2\delta + 2\epsilon))$.

If S_4 precedes S_{43} then w_3 is equal to the waiting time w_2 of the preceding sequence S_4.

Sequence S_{35}

- Sequence S_{35} involves the following robotic moves.
  1. Robot travels to the buffer and picks part i from buffer.
  2. Robot travels to machine M_2 and loads part on M_2.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to machine M_1 and loads the part on M_1.
  5. Robot waits at machine M_1 until the part is processed and unloads it from M_1.
  6. Robot travels to the buffer and drops the part in buffer.
  7. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  8. Robot travels to the output and drops part at the output.
- Sequence time of S_{35}:

$$T_{35} = 8\epsilon + 5.5\delta + \alpha_{i+1} + w_1,$$

where $w_1 = max(0, \beta_i - (3\delta + 4\epsilon + \alpha_{i+1}))$.

Sequence S_{53}

- Sequence S_{53} involves the following robotic moves.

  1. Robot travels to the input and picks up part $i+1$.
  2. Robot travels to machine M_1 and loads the part on M_1.
  3. Robot travels to the buffer and picks part i from buffer.
  4. Robot travels to machine M_2 and loads the part on M_2.
  5. Robot travels to machine M_1, waits if necessary and unloads part $i+1$ from M_1.
  6. Robot travels to the buffer and drops the part in buffer.
  7. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
  8. Robot travels to the output and drops part at the output.

- Sequence time of S_{53}:

$$T_{53} = 8\varepsilon + 6\delta + w_1 + w_2,$$

where $w_1 = max(0, \alpha_{i+1} - (2\delta + 2\varepsilon))$,

and $w_2 = max(0, \beta_i - (2\delta + 2\varepsilon + w_1))$.

We observe that sequence S_{35} is exactly same as S_3 and sequence S_{53} is exactly same as S_5. Hence we eliminate the notations S_{35} and S_{53} from the list of feasible sequences and simply use their equivalents S_3 and S_5.

Sequence S_{45}

- Sequence S_{45} involves the following robotic moves.

  1. Robot travels to buffer and picks part i from buffer.
  2. Robot carries part to machine M_2 and loads it on M_2.
  3. Robot travels to M_1, waits if necessary and unloads part $i+1$ from M_1.

4. Robot carries part to the buffer and drops it in the buffer.

5. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.

6. Robot travels to the output and drops part at the output.

- Sequence time of S_{45}:

$$T_{45} = 6\varepsilon + 4.5\delta + w_1 + w_2,$$

where $w_1 = max(0, \alpha_{i+1} - (4.5\delta + 4\varepsilon + w_3))$,
and $w_2 = max(0, \beta_i - (2\delta + 2\varepsilon + w_1))$.

The expression for w_3 which is the waiting time of the robot at M_2 before unloading $i-1^{th}$ part, and thus the value of T_{45} depends on the sequence preceding S_{45}. If S_{34} precedes S_{45}, then w_3 is given by the expression

$$w_3 = max(0, \beta_{i-1} - (5.5\delta + 6\varepsilon + \alpha_i)).$$

If S_{54} or S_{64} precedes S_{45}, then w_3 is given by the expression

$$w_3 = max(0, \beta_{i-1} - (4.5\delta + 4\varepsilon + w_4)), \text{ and}$$

$$w_4 = max(0, \alpha_i - (2\delta + 2\varepsilon)).$$

If S_4 precedes S_{45}, then w_3 is equal to the waiting time w_2 of the preceding sequence S_4.

We notice that S_{43} is exactly same as S_{45}. Hence, it is sufficient to consider only one of these two sequences and we choose to represent S_{43}. Thus the total number of feasible sequences are reduced to 23 from 26 after eliminating 3 sequences.

Sequence S_{54}

- Sequence S_{54} involves the following robotic moves.

1. Robot travels to the input and picks up part $i+1$.

2. Robot travels to machine M_1 and loads the part on M_1.

3. Robot travels to the buffer and picks part i from buffer.

4. Robot travels to machine M_2 and loads the part on M_2.

5. Robot travels to machine M_1, waits if necessary and unloads part $i+1$ from M_1.

6. Robot travels to the buffer and drops the part in buffer.

7. Robot travels to the input and picks part $i+2$ from I.

8. Robot carries the part to machine M_1 and loads it on M_1.

9. Robot travels to M_2, waits if necessary and unloads part i from M_2.

10. Robot travels to the output and drops part at O.

- Sequence time of S_{54}:

$$T_{54} = 10\varepsilon + 8.5\delta + w_1 + w_2,$$

where $w_1 = max(0, \alpha_{i+1} - (2\delta + 2\varepsilon))$,

and $w_2 = max(0, \beta_i - (4.5\delta + 4\varepsilon + w_1))$.

Sequence S₄₆

- Sequence S_{46} involves the following robotic moves.

1. Robot travels to buffer and picks part i from buffer.

2. Robot carries part to machine M_2 and loads it on M_2.

3. Robot waits at machine M_2 till the part is processed and unloads part from M_2.

4. Robot carries the part to output and drops it at the output.

- Sequence time of S_{46}:

$$T_{46} = 4\varepsilon + 2.5\delta + \beta_i.$$

Sequence S_{64}

- Sequence S_{64} involves the following robotic moves.

  1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
  2. Robot travels to the buffer and drops the part in buffer.
  3. Robot travels to the input and picks up part $i+1$.
  4. Robot travels to machine M_1 and loads the part on M_1.
  5. Robot travels to the buffer and picks part i from buffer.
  6. Robot travels to machine M_2 and loads the part on M_2.
  7. Robot travels to M_1, waits if necessary and unloads part $i+1$ from M_1.
  8. Robot carries part to the buffer and drops it in the buffer.
  9. Robot travels to the input and picks part $i+2$ from I.
  10. Robot carries the part to machine M_1 and loads it on M_1.
  11. Robot travels to M_2, waits if necessary and unloads part i from M_2.
  12. Robot travels to the output and drops part at O.

- Sequence time of S_{64}:

$$T_{64} = 12\varepsilon + 10\delta + w_1 + w_2 + w_3,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 4\varepsilon + \beta_{i-1}))$,

$w_2 = max(0,\ \alpha_{i+1} - (2\delta + 2\varepsilon))$,

and $w_3 = max(0,\ \beta_i - (4.5\delta + 4\varepsilon + w_2)$.

Sequence S_{56}

- Sequence S_{56} involves the following robotic moves.

  1. Robot travels to the input and picks up part $i+1$.
  2. Robot travels to machine M_1 and loads the part on M_1.
  3. Robot travels to the buffer and picks part i from buffer.

4. Robot travels to machine M_2 and loads the part on M_2.

5. Robot waits at machine M_2 till the part is processed and unloads part from M_2.

6. Robot travels to the output and drops part at the output.

- Sequence time of S_{56}:

$$T_{56} = 6\varepsilon + 4\delta + \beta_i.$$

Sequence S_{65}

- Sequence S_{65} involves the following robotic moves.

1. Robot travels to machine M_1, waits if necessary and unloads part i from M_1.
2. Robot travels to the buffer and drops the part in buffer.
3. Robot travels to the input and picks up part $i+1$.
4. Robot travels to machine M_1 and loads the part on M_1.
5. Robot travels to the buffer and picks part i from buffer.
6. Robot travels to machine M_2 and loads the part on M_2.
7. Robot travels to M_1, waits if necessary and unloads part $i+1$ from M_1.
8. Robot travels to the buffer and drops the part in buffer.
9. Robot travels to machine M_2, waits if necessary and unloads part i from M_2.
10. Robot travels to the output and drops part at the output.

- Sequence time of S_{65}:

$$T_{65} = 10\varepsilon + 7.5\delta + w_1 + w_2 + w_3,$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 4\varepsilon + \beta_{i-1}))$,

$w_2 = max(0,\ \alpha_{i+1} - (2\delta + 2\varepsilon))$,

and $w_3 = max(0,\ \beta_i - (2\delta + 2\varepsilon + w_2)$.

4.5 Sequence Times for Cyclic Sequences

The expressions for the cycle times of the 6 cyclic sequences in a multiple part type setup is slightly different from a single part type problem. Although the robotic moves are the same as in Chapter 3, the cycle times depend on the position of the part in the sequence of parts.

Sequence S_1

- Sequence time of S_1:

$$T_1 = 6\varepsilon + 4\delta + \alpha_i + \beta_i.$$

Sequence S_2

- Sequence time of S_2:

$$T_2 = 6\varepsilon + 6\delta + w_1 + w_2.$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 2\varepsilon + w_3))$,

$w_3 = max(0,\ \beta_{i-1} - (3\delta + 2\varepsilon))$,

and $w_2 = max(0,\ \beta_i - (3\delta + 2\varepsilon))$.

Sequence S_3

- Sequence time of S_3:

$$T_3 = 8\varepsilon + 5.5\delta + \alpha_{i+1} + w_1.$$

where $w_1 = max(0,\ \beta_i - (3\delta + 4\varepsilon + \alpha_{i+1}))$

Sequence S_4

- Sequence time of S_4:

$$T_4 = 8\varepsilon + 7\delta + w_1 + w_2,$$

where $w_1 = max(0,\ \alpha_{i+1} - (4.5\delta + 4\varepsilon + w_3)) = max(0,\ \Delta_{1i} - w_3)$
and $w_2 = max(0,\ \beta_i - (4.5\delta + 4\varepsilon + w_1)) = max(0,\ \Delta_{2i} - w_1)$.

The expression for w_3 which is the waiting time of the robot at M_2 before unloading $i-1^{th}$ part, and thus the value of T_4 depends on the sequence preceding S_4.

If S_{34} precedes S_4 then w_3 is given by the expression

$$w_3 = max(0,\ \beta_{i-1} - (5.5\delta + 6\varepsilon + \alpha_i)).$$

If S_{54} or S_{64} precedes S_4 then w_3 is given by the expression

$$w_3 = max(0,\ \beta_{i-1} - (4.5\delta + 4\varepsilon + w_4)),$$

where $w_4 = max(0,\ \alpha_i - (2\delta + 2\varepsilon))$.

If S_4 precedes S_4 then w_3 is equal to the waiting time w_2 of the preceding sequence S_4.

If S_4 precedes S_4 and if that preceding S_4 is the first sequence in the MPS then the following steps are used to compute the cycle time T_4.

- Note the first occurrence of sequence S_4 and set i = 1.
- Compute Δ_{1i} and Δ_{2i} for each i^{th} part processed by sequence S_4 successively, where $\Delta_{1i} = \alpha_{i+1} - (4.5\delta + 4\varepsilon)$ and $\Delta_{2i} = \beta_i - (4.5\delta + 4\varepsilon)$
- Set the minimum of these Δ_{1i} and Δ_{2i} to zero. The minimum of these Δ and its corresponding waiting time can be safely assumed to be zero because of the following reason. When the waiting time which is a positive quantity is subtracted from the minimum Δ the result would be either a negative waiting time or a very small positive number which is almost equal to zero. The resulting waiting time cannot be negative and hence is equal to zero.
- The waiting time w corresponding to the minimum Δ evaluates to zero.
- If Δ_{1i} is the minimum its corresponding w_{1i} would be zero and w_{2i} evaluates to Δ_{2i}. Having known the values of w_{1i} and w_{2i} its cycle time can be computed. Using this w_{2i} the cycle times of the succeeding and later preceding sequences are calculated (as their waiting times are dependent on the waiting time of their preceding sequence).

– If Δ_{2i} is the minimum its corresponding w_{2i} would be zero. It's cycle time cannot be computed yet as the value of w_{1i} is unknown. Using this w_{2i} the waiting time w_{1i} and w_{2i} and thus the cycle time of the succeeding sequence can be computed. This will help in computing its succeeding and then its preceding sequences and also that sequence with the minimum Δ_{2i}.

The above steps can be illustrated with an example explained below. Consider a MPS of two parts where both parts are processed using S_4 and their processing times on M_1 are α_1 and α_2 and on M_2 are β_1 and β_2. Let the waiting times for the first part at M_1 and M_2 be w_{11} and w_{21} respectively. Similarly, let w_{12} and w_{22} be the waiting times for the second part.

$$w_{11} = max(0, \alpha_2 - (4.5\delta + 4\varepsilon + w_{22})) = max(0, \Delta_{11} - w_{22}),$$

$$w_{21} = max(0, \beta_1 - (4.5\delta + 4\varepsilon + w_{11})) = max(0, \Delta_{21} - w_{11}),$$

$$w_{12} = max(0, \alpha_1 - (4.5\delta + 4\varepsilon + w_{21})) = max(0, \Delta_{12} - w_{21}),$$

$$w_{22} = max(0, \beta_2 - (4.5\delta + 4\varepsilon + w_{12})) = max(0, \Delta_{22} - w_{12}).$$

Consider $min(\Delta_{11}, \Delta_{21}, \Delta_{12}, \Delta_{22})$. If Δ_{11} is minimum then set it to zero and w_{11} evaluates to zero and $w_{21} = max(0, \Delta_{21})$. Having known the values of w_{11} and w_{21}, T_4 of the first part can be computed; the waiting times and cycle time of the second part can also be computed. If Δ_{21} is minimum then set it to zero and w_{21} evaluates to zero and $w_{12} = max(0, \Delta_{12})$. Having known the values of w_{12}, the value of w_{22} and T_4 of the second part can be computed; substituting w_{22}, the waiting times and cycle time of the first part can thus be computed.

Sequence S_5

- Sequence time of S_5:

$$T_5 = 8\varepsilon + 6\delta + w_1 + w_2.$$

where $w_1 = max(0, \alpha_{i+1} - (2\delta + 2\varepsilon))$,

and $w_2 = max(0, \beta_i - (2\delta + 2\varepsilon + w_1))$.

Sequence S_6

- Sequence time of S_6:

$$T_6 = 8\varepsilon + 5.5\delta + \beta_i + w_1.$$

where $w_1 = max(0,\ \alpha_i - (3\delta + 4\varepsilon + \beta_{i-1}))$.

4.6 Sequence Trees

To preserve the cyclic nature of the schedules, not every sequence is allowed to follow every other sequence. The sequence tree shown in Figure 12 illustrates the possible preceding and succeeding sequences for the 23 feasible sequences. We consider S_1 as the root of the tree just for illustration purposes. We can actually construct the tree with any one of the 23 sequences as the root. We observe that sequences S_3, S_{13}, S_{23} and S_{43} can be succeeded by the sequences S_5, S_{54} and S_{56} as there can be a switching from S_3 to S_5 right after the completion of a full cycle of S_3 with out any wasteful robotic moves. Table 3 simplifies the tree and outlines the preceding and succeeding sequences for every sequence.

4.7 Preserving the Cyclic Nature of Schedules

We observe that these combination sequences are not cyclic. That is their initial state is not same as their final state. For instance the initial state of S_{12} is different from its final state. So to preserve the cyclic nature of production in an MPS, the sequences should return to their initial state. If there is a transition from S_1 in to another sequence for instance S_3 through the combination sequence S_{13}, then there should be a transition back in to S_1 to bring the sequence back to its initial state. This can be translated into a mathematical proof as illustrated by Lemma 2.

Notation

B_k represents the set whose elements comprise of parts using sequence S_k, where

$$k \in \{1,2,3,4,5,6,12,21,13,31,16,61,23,32,26,62,34,43,46,64,54,56,65\}.$$

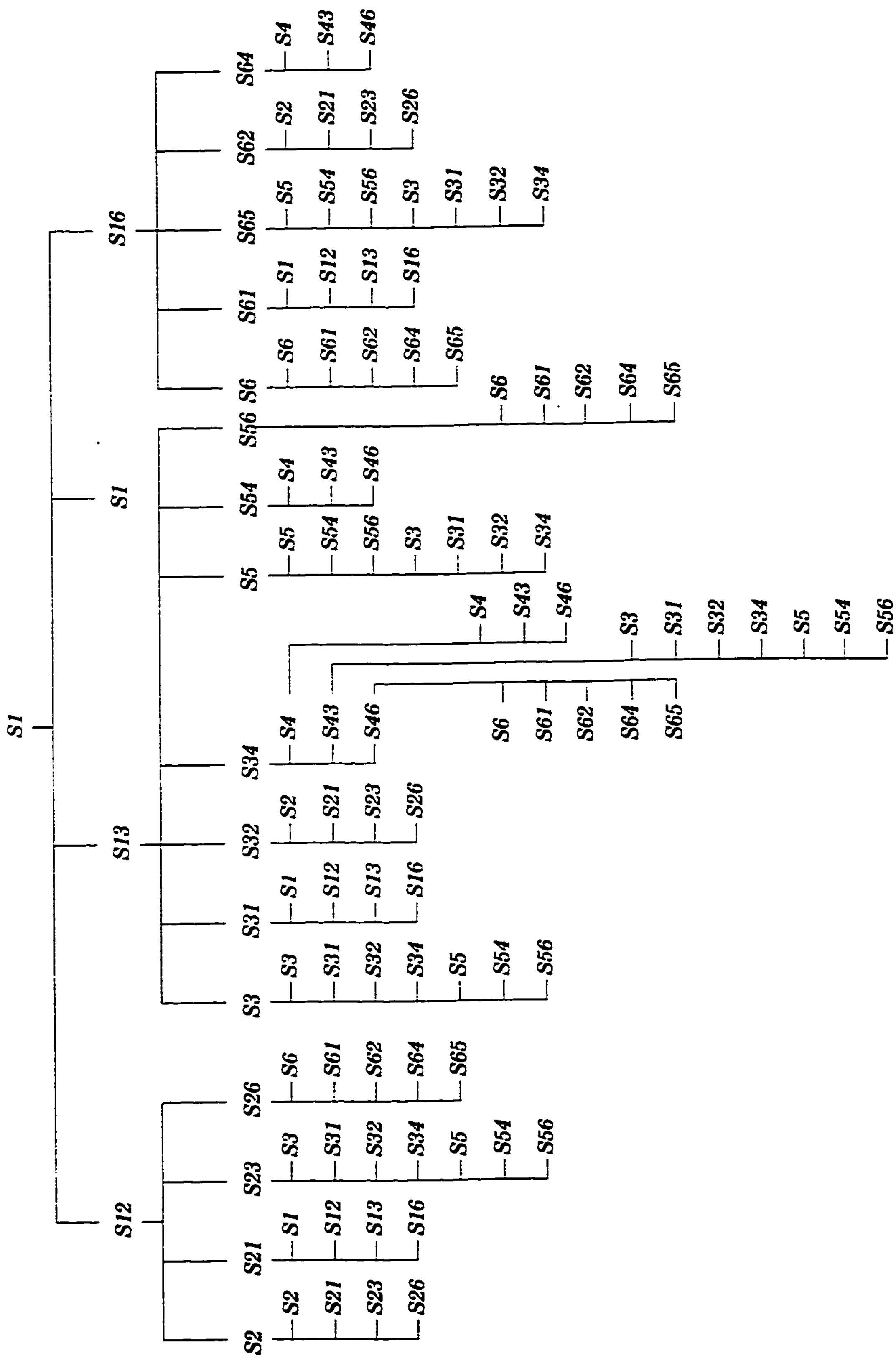

Figure 12: Sequence Tree.

Sequence	Preceding Sequence	Succeeding Sequence
S_1	$S_1, S_{21}, S_{31}, S_{61}$	$S_1, S_{12}, S_{13}, S_{16}$
S_2	$S_2, S_{12}, S_{32}, S_{62}$	$S_2, S_{21}, S_{23}, S_{26}$
S_3	$S_3, S_{13}, S_{23}, S_{43}, S_5, S_{65}$	$S_3, S_{31}, S_{32}, S_{34}, S_5, S_{54}, S_{56}$
S_4	$S_4, S_{34}, S_{54}, S_{64}$	S_4, S_{43}, S_{46}
S_5	$S_5, S_{65}, S_3, S_{43}, S_{13}, S_{23}$	$S_5, S_{54}, S_{56}, S_3, S_{31}, S_{32}, S_{34}$
S_6	$S_6, S_{16}, S_{26}, S_{46}, S_{56}$	$S_6, S_{61}, S_{62}, S_{64}, S_{65}$
S_{12}	$S_1, S_{21}, S_{31}, S_{61}$	$S_2, S_{21}, S_{23}, S_{26}$
S_{13}	$S_1, S_{21}, S_{31}, S_{61}$	$S_3, S_{31}, S_{32}, S_{34}, S_5, S_{54}, S_{56}$
S_{16}	$S_1, S_{21}, S_{31}, S_{61}$	$S_6, S_{61}, S_{62}, S_{64}, S_{65}$
S_{21}	$S_2, S_{12}, S_{32}, S_{62}$	$S_1, S_{12}, S_{13}, S_{16}$
S_{23}	$S_2, S_{12}, S_{32}, S_{62}$	$S_3, S_{31}, S_{32}, S_{34}, S_5, S_{54}, S_{56}$
S_{26}	$S_2, S_{12}, S_{32}, S_{62}$	$S_6, S_{61}, S_{62}, S_{64}, S_{65}$
S_{31}	$S_3, S_{13}, S_{23}, S_{43}, S_5, S_{65}$	$S_1, S_{12}, S_{13}, S_{16}$
S_{32}	$S_3, S_{13}, S_{23}, S_{43}, S_5, S_{65}$	$S_2, S_{21}, S_{23}, S_{26}$
S_{34}	$S_3, S_{13}, S_{23}, S_{43}, S_5, S_{65}$	S_4, S_{43}, S_{46}
S_{43}	$S_4, S_{34}, S_{54}, S_{64}$	$S_3, S_{31}, S_{32}, S_{34}, S_5, S_{54}, S_{56}$
S_{46}	$S_4, S_{34}, S_{54}, S_{64}$	$S_6, S_{61}, S_{62}, S_{64}, S_{65}$
S_{54}	$S_5, S_{65}, S_3, S_{43}, S_{13}, S_{23}$	S_4, S_{43}, S_{46}
S_{56}	$S_5, S_{65}, S_3, S_{43}, S_{13}, S_{23}$	$S_6, S_{61}, S_{62}, S_{64}, S_{65}$
S_{61}	$S_6, S_{16}, S_{26}, S_{46}, S_{56}$	$S_1, S_{12}, S_{13}, S_{16}$
S_{62}	$S_6, S_{16}, S_{26}, S_{46}, S_{56}$	$S_2, S_{21}, S_{23}, S_{26}$
S_{64}	$S_6, S_{16}, S_{26}, S_{46}, S_{56}$	S_4, S_{43}, S_{46}
S_{65}	$S_6, S_{16}, S_{26}, S_{46}, S_{56}$	$S_5, S_{54}, S_{56}, S_3, S_{31}, S_{32}, S_{34}$

Table 3: Preceding and Succeeding Sequences.

$|B_k|$ represents the cardinality of the set.

Lemma 2

In any feasible solution for a two machine, single buffer, multiple part-type problem,

$$|B_{12}| \leq |B_{21}| + |B_{31}| + |B_{61}|.$$

Proof of Lemma 2

Any feasible solution is a cycle and therefore should have the same initial and final state. However, the initial state of S_1 is not same as the final state of S_2 and vice-versa. It follows that the number of transitions from S_1 into any other cycle must be equal to the number of transitions into S_1 from any other any cycle. For every S_{12} there should be at least one S_{21} or one S_{31} or one S_{61} to preserve the cyclic nature. Hence

$$|B_{12}| \leq |B_{21}| + |B_{31}| + |B_{61}|, \text{ and}$$

$$|B_{12}| + |B_{13}| + |B_{16}| = |B_{21}| + |B_{31}| + |B_{61}|.$$

Extending the same reasoning to other sets, gives us the necessary conditions for a valid division of MPS into subsets $|B_1|$, $|B_2|$, $|B_3|$ and so on. The necessary conditions are enumerated below.

1. $|B_{21}| \leq |B_{12}| + |B_{32}| + |B_{62}|$, and

   $|B_{21}| + |B_{23}| + |B_{26}| = |B_{12}| + |B_{32}| + |B_{62}|.$

2. $|B_{13}| \leq |B_{21}| + |B_{31}| + |B_{61}|$, and

   $|B_{31}| + |B_{32}| + |B_{34}| + |B_3| = |B_{13}| + |B_{23}| + |B_{43}| + |B_5|.$

3. $|B_{31}| \leq |B_{13}| + |B_{23}| + |B_5| + |B_{43}|.$

4. $|B_{16}| \leq |B_{21}| + |B_{31}| + |B_{61}|.$

5. $|B_{61}| \leq |B_{16}| + |B_{26}| + |B_{46}| + |B_{56}|$, and

$$|B_{61}| + |B_{62}| + |B_{64}| + |B_{65}| = |B_{16}| + |B_{25}| + |B_{46}| + |B_{56}|.$$

6. $|B_{23}| \leq |B_{12}| + |B_{32}| + |B_{62}|$.

7. $|B_{32}| \leq |B_{13}| + |B_5| + |B_{23}| + |B_{43}|$.

8. $|B_{26}| \leq |B_{62}| + |B_{32}| + |B_{12}|$.

9. $|B_{62}| \leq |B_{16}| + |B_{26}| + |B_{46}| + |B_{56}|$.

10. $|B_{34}| \leq |B_{13}| + |B_5| + |B_{23}| + |B_{43}|$.

11. $|B_{43}| \leq |B_{34}| + |B_{54}| + |B_{64}|$, and

    $|B_{43}| + |B_{46}| = |B_{34}| + |B_{54}| + |B_{64}|$.

12. $|B_{56}| + |B_{54}| = |B_{43}| + |B_{65}|$.

13. $|B_{54}| \leq |B_3| + |B_{65}|$.

14. $|B_{64}| \leq |B_{16}| + |B_{26}| + |B_{46}| + |B_{56}|$.

15. $|B_{56}| \leq |B_3| + |B_{65}|$.

16. $|B_{65}| \leq |B_{16}| + |B_{26}| + |B_{46}| + |B_{56}|$.

4.8 Conclusion

In this chapter we provide all feasible one unit robotic move sequences both cyclic and combination for a two machine robotic cell with multiple part types and a single buffer. We use analytical methods to evaluate their cycle times. We employ a sequence tree to illustrate the preceding and succeeding sequences for each one of the 23 possible sequences. We also supply the necessary conditions for preserving the cyclic nature of the schedules. We thus provide a framework to identify the optimal sequences for each part in a given arrangement of parts in a minimal part set (MPS).

Chapter 5

Optimal Robot Sequences for a Two-Machine Cell with Multiple Part Types

---

In this chapter, we provide a branch-and-bound algorithm to determine the optimal sequences to be used for processing the parts in the minimal part set. We employ this algorithm to draw some meaningful conclusions on optimal scheduling in a two-machine robotic cell with multiple parts, and to compare the results with the identical parts two-machine cell problem. We establish the usefulness of a buffer in reducing the processing time of a minimal part set. We establish the condition where a buffer would not be useful in reducing the processing time.

5.1 Introduction

We apply the framework developed in Chapter 4 to develop a branch-and-bound algorithm. The algorithm identifies the best sequence for each part in the minimal part set (MPS), to optimize the manufacturing time for a given arrangement of the parts in the MPS. We implement the algorithm in C++ and apply it to an extensive test suite in order to gain insight into the role of a buffer in reducing the manufacturing time of a MPS. Section 5.2 describes the branch-and-bound algorithm, including the lower bound. Section 5.3 illustrates the branch-and-bound algorithm with an example. Section 5.4 establishes the need for a buffer, and illustrates it with an example. Section 5.5 analyzes the test results to observe the amount of saving that is possible with the buffer. Section 5.6 discusses the necessity of considering the sequence S_5. It provides an example to demonstrate that the combination sequence involving sequence S_5 dominates every other sequence. Section 5.7 concludes the chapter with a summary of the observations.

5.2 Branch-and-Bound Algorithm

We provide a *branch-and-bound* algorithm to determine the optimal sequence of cycles for processing a specific arrangement of parts in the MPS. The goal is to minimize the cycle time for the production of the MPS. The algorithm preserves the cyclic nature of the manufacturing process by considering only sequences where the first cycle is a valid successive cycle of the last cycle. The MPS can consist of any number of parts. The algorithm is implemented in C++. used in the algorithm is

Max(Sum of remaining α, Sum of remaining β).

The motivation for choosing this bound is the fact that the minimum processing time for the parts in a MPS is equal to the maximum of (i) the total processing time for the parts on machine M_1 and (ii) the total processing time for the parts on machine M_2. We have included a mechanism for observing the usefulness of the bound in eliminating some computation.

Algorithm A

The milestones of the *branch-and-bound* algorithm are enumerated below. We consider a MPS of four parts for illustration purposes.

1. We use the variables T_0, T_1 and T_2 to store the cumulative processing time as we compute the cycle times. Initially, the value of T_0 is 0.

2. Compute the processing time of the first part in the MPS, using each of the 21 sequences (S_1, S_2, ..., S_5, S_{12}, S_{62}) individually except S_4 and S_{43}.

3. Determine the sequence S_a with the minimum processing time, say S_2. Add the processing time of the first part using sequence S_a to T_0.

4. Consider the succeeding sequences of the first sequence S_a, which in this case are S_2, S_{21}, S_{23}, and S_{26}, to compute the processing time of the second part in the MPS.

5. Determine the sequence S_b with the minimum processing time for the second part, say S_{21}. Increment the cumulative processing time T_0 with the processing time of the second part using sequence S_b.

6. Consider the succeeding sequences of the second sequence S_b, which in this case are S_1, S_{12}, S_{13}, and S_{16} to compute the processing time of the third part in the MPS.

7. Determine the sequence S_c with the minimum processing time for the third part, say S_{12}. Increment the cumulative processing time T_0 with the processing time of the third part using sequence S_c.

8. Consider the succeeding sequences of the third sequence S_c, which in this case are S_2, S_{21}, S_{23}, and S_{26}. Since the fourth part is the last part in the MPS for this case, consider only the sequences that will bring the system back to its original state S_a (S_2 in this case), that is only sequences for which the first sequence S_a is a valid successor. Note that sequence S_2 itself is the only valid one in this case. In the absence of such a sequence we ignore the corresponding sub-tree.

9. Determine the sequence S_n used for the last part in the sequence. The sequence S_n should bring the system back to its original state to preserve the cyclic nature. In summary, we repeat step 4 until we reach the n–th part in the MPS.

10. Increment the cumulative processing time T_0 with the processing time of the fourth part using sequence S_n.

11. Store the manufacturing time T_0 of the MPS, and the sequence of sequences S_a, S_b, …, S_n, that is, S_2, S_{21}, S_{12}, and S_2 in this case.

12. Consider the $(n-1)$th part, the third part in this case, and use the second best sequence among the *unused* sequences for the $(n-1)$th part, say S_{13} in this case, for processing this part. Increment the cumulative processing time T_1 of the first $(n-2)$ parts, that is the first and second parts in this case, with the processing time of the $(n-1)$th part using the second best sequence, sequence S_{13} in this case. Let T_2 be defined as follows.

$$T_2 = T_1 + \text{Max(Sum of remaining } \alpha\text{, Sum of remaining } \beta\text{)}.$$

If T_2 is greater than T_0, then ignore the sub-tree whose root node is the second best sequence for the $(n-1)$th part, sequence S_{13} in this case. Otherwise, compute the best processing times for the succeeding parts using the sequences within this sub-tree, that is for the nth part, and increment T_1 with this amount. If T_1 is less than T_0, then set T_0 to T_1.

13. Repeat step 12 for the $(n-1)$th part, using the other *unused* sequences for the $(n-1)$th part, that is the third best sequence for the $(n-1)$th part and so on. In other words, use the other sub-trees whose root nodes are sequences at the $(n-1)$th level.

14. Repeat steps 12 and 13 for the $(n-2)$th, $(n-3)$th, …, *first* parts. Note that for the *first* part, the other 20 sequences (S_1, S_2, …, S_5, S_{12}, S_{62}, except sequence S_a) are used individually. The resulting sequence stored in S_a, S_b, …, S_n is the optimal sequence found by the branch-and-bound algorithm.

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	20	10
p_2	30	25
p_3	15	30
p_4	12	20

Table 4: Processing Time on Machines M_1 and M_2.

Algorithm B

In searching for an optimal sequence of cycles, whenever the first part in the sequence is processed using either cycle S_4 or cycle S_{43}, we find the *best* possible sequence using the following algorithm.

1. Construct a tree of sequences with S_4 (or S_{43}) as the root node, by generating a sub-tree for each possible successor sequence. The height of the tree is equal to the number of parts in the MPS.

2. For each path in the tree starting from the root node to a leaf node, such that the first sequence is a valid successor of the last sequence, compute the processing time of the MPS using the method suggested for computing the sequence time for S_4 in Chapter 4.

3. Determine the path with minimum processing time for the MPS. The sequences in this path represent the optimal sequence of sequences starting with S_4 (or S_{43}).

In the above case, we use total enumeration and do not apply the branch-and-bound algorithm to further optimize the sequence of sequences for the following reason. The computations of the processing times of sequences S_4 and S_{43} depend on their preceding sequence. In this case, since these sequences are being used for processing the first part, the preceding sequence corresponds to the sequence used for processing the last part. So the processing time for the first part cannot be computed prior to identifying the sequence of sequences. Hence, the processing time for the MPS is computed using the method suggested for S_4 in Chapter 4.

The processing time for the optimal sequence of sequences, obtained when the first part is processed using S_4 or S_{43} according to Algorithm B, is compared with the processing time for the optimal sequence of sequences obtained by applying the branch-and-bound algorithm. The sequence of sequences that yields the minimum processing time for the MPS is chosen as the optimal sequence of sequences to be used for processing the MPS.

5.3 Illustration of Branch-and-Bound Algorithm

We consider the problem of two machine robotic cell with buffer producing multiple parts. The two scheduling issues that need to be resolved in this problem are

- Sequence of robot moves for each part.
- Order of processing of parts in MPS.

The branch-and-bound algorithm is primarily used to identify the optimal robot move sequences for a pre-determined order of processing of parts in MPS. We show that the algorithm can be used for finding optimal robot move sequences when the processing order of parts is not fixed by totally enumerating different arrangement of parts in MPS. However it is to be noted that this is not an efficient way and we do it just for illustration purposes. We execute the program to observe a wide range of scenarios. We include a representative example in this section, and give the results of the sample runs of the program. Table 4 provides the data for the example being considered for discussion. The processing times α and β on Machines M_1 and M_2 respectively, for each part in the MPS are given in Table 4. In the example given in Table 3, the MPS comprises of four parts. The value of δ is 4, and the value of ε is 1.

Illustration of Branch-and-Bound Algorithm for a Fixed Order of Processing of Parts

In some manufacturing systems especially *Kanban* systems, the order of processing of parts is kept stable [SB97] [VBW92]. If the order of processing of parts is fixed to p_1, p_2, p_3, p_4 then the optimal sequences and the processing time of MPS given by the algorithm are shown in Table 5. The optimal

Sequence of Parts	With Buffer		Without Buffer		With Buffer – All Parts Same Sequence	
	Optimal Sequence of Cycles	Proc Time	Optimal Sequence of Cycles	Proc Time	Optimal Sequence of Cycles	Proc Time
p_1 p_2 p_3 p_4	S_{43} S_{54} S_4 S_4	161.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	163.00

Table 5: Optimal Robot Move Sequences and Optimal Cycle Time for a fixed Part Sequence.

Sequence of Parts	With Buffer: Optimal Sequence of Cycles	With Buffer: Proc Time	Without Buffer: Optimal Sequence of Cycles	Without Buffer: Proc Time	With Buffer – All Parts Same Sequence: Optimal Sequence of Cycles	With Buffer – All Parts Same Sequence: Proc Time
p_1 p_2 p_3 p_4	S_{43} S_{54} S_4 S_4	161.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	163.00
p_1 p_2 p_4 p_3	S_{43} S_{54} S_4 S_4	151.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	155.00
p_1 p_3 p_2 p_4	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00
p_1 p_3 p_4 p_2	S_{43} S_{54} S_4 S_4	153.00	S_2 S_2 S_2 S_2	164.00	S_4 S_4 S_4 S_4	155.00
p_1 p_4 p_2 p_3	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00
p_1 p_4 p_3 p_2	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00
p_2 p_1 p_3 p_4	S_4 S_{43} S_{54} S_4	153.00	S_2 S_2 S_2 S_2	164.00	S_4 S_4 S_4 S_4	155.00
p_2 p_1 p_4 p_3	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00
p_2 p_3 p_1 p_4	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00
p_2 p_3 p_4 p_1	S_{54} S_4 S_4 S_{43}	161.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	163.00
p_2 p_4 p_1 p_3	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00
p_2 p_4 p_3 p_1	S_{54} S_4 S_4 S_{43}	151.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	155.00
p_3 p_1 p_2 p_4	S_4 S_{43} S_{54} S_4	151.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	155.00
p_3 p_1 p_4 p_2	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00
p_3 p_2 p_1 p_4	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00
p_3 p_2 p_4 p_1	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00
p_3 p_4 p_1 p_2	S_4 S_4 S_{43} S_{54}	161.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	163.00
p_3 p_4 p_2 p_1	S_{54} S_4 S_4 S_{43}	153.00	S_2 S_2 S_2 S_2	164.00	S_4 S_4 S_4 S_4	155.00
p_4 p_1 p_2 p_3	S_4 S_{43} S_{54} S_4	161.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	163.00
p_4 p_1 p_3 p_2	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00	S_2 S_2 S_2 S_2	154.00
p_4 p_2 p_1 p_3	S_4 S_4 S_{43} S_{54}	153.00	S_2 S_2 S_2 S_2	164.00	S_4 S_4 S_4 S_4	155.00
p_4 p_2 p_3 p_1	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00	S_2 S_2 S_2 S_2	163.00
p_4 p_3 p_1 p_2	S_4 S_4 S_{43} S_{54}	151.00	S_2 S_2 S_2 S_2	169.00	S_4 S_4 S_4 S_4	155.00
p_4 p_3 p_2 p_1	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00	S_2 S_2 S_2 S_2	153.00

Table 6: **Optimal Robot Move Sequences and Optimal Cycle Time for Different Part Sequences.**

Sequence of Parts				Optimal Sequence of Cycles				Processing Time
p_1	p_2	p_4	p_3	S_{43}	S_{54}	S_4	S_4	151.00
p_2	p_4	p_3	p_1	S_{54}	S_4	S_4	S_{43}	151.00
p_3	p_1	p_2	p_4	S_4	S_{43}	S_{54}	S_4	151.00
p_4	p_3	p_1	p_2	S_4	S_4	S_{43}	S_{54}	151.00

Table 7: Optimal Sequence of Parts and Optimal Robot Move Sequences.

processing time with a buffer is 161 time units. In the absence of a buffer, the optimal sequences and their processing times given by a modified version of the algorithm are also included in Table 5. We observe that the optimal processing time without a buffer is 169 time units. The savings with a buffer over the processing of the MPS is 8 time units.

We know from the discussion in Chapters 2 and 3 that two machine cell with a buffer is a restricted version of three machine cell problem. Hall *et al.* [HKS97] in their study of multiple part-types problem in a three machine cell, force all the parts in the MPS to be processed by the same sequence. In our algorithm we relax this restriction and allow switching between sequences while processing the parts in the MPS. In Table 5 we show the optimal processing time when all the parts in the MPS are processed by same sequence. It is evident from Table 5 that it is disadvantageous to pre-determine robot move sequence and force all parts in MPS to be processed by that same sequence. We observe that with switching we save 2 time units. Therefore, switching is beneficial in reducing the processing time of MPS. We also looked into the usefulness of the bound. We found that the bound is successful in skipping up to 70% of the sub-trees that are to be searched for an optimal sequence.

Illustration of Branch-and-Bound Algorithm for a Varying Order of Processing of Parts

In the cases where the processing order of parts can be varied, the problem of scheduling in robotic cell becomes one of identifying optimal sequence of parts in addition to identifying optimal robot move sequences. We execute the program for all the possible arrangements of parts and choose the optimal order of processing of parts. This total enumeration may not be efficient and a procedure can be built to arrange the parts in the MPS in an optimal fashion. However in this research we do not address this issue. The four parts in the MPS can be arranged in ${}^4P_4 = 4!$ ways giving rise to twenty-four distinct part sequences. Table 6 includes the optimal sequences for processing the parts in the MPS, and the total processing time for the MPS, for each of these twenty-four part sequences with and without buffer. It also includes the optimal sequence and the processing time, for each of these twenty-four part sequences if the parts in MPS are produced using same sequence.

Therefore, by total enumeration we find the optimal robot move sequences for the optimal part sequences. Table 7 gives the sequence of parts for which the processing time is optimal, that is 151 time units. The table also includes the respective optimal sequences to be used for processing each of these part sequences. It is evident from Table 6 that the optimal cycle time that can be accomplished without a buffer is 153 time units by repeatedly processing the parts in the MPS using cycle S_2. On comparing this data with Table 7, we note that the savings accomplished with the buffer is 2 time units for a MPS. This translates as savings equal to 0.5δ for each part in the MPS which is the maximum savings that can be accomplished in a two-machine with identical parts problem. However, it remains open to see if the magnitude of savings can be more in the current problem of two machines with multiple parts. In all the examples in subsequent sections, we do not fix the order of processing of parts in MPS and arrange the parts in all feasible ways ($p!$ ways). We do this with the intent to optimize both robot move sequences and part sequences.

5.4 Necessity of a Buffer

In this section we establish the need for a buffer by providing an example where considerable savings in time are achieved with the help of a buffer. In this example, the MPS comprises of three parts. The value of δ is 3; and the value of ε is 1. Table 8 gives the processing times α and β on Machines M_1 and M_2 respectively, for each part in the MPS. The three parts in the MPS can be arranged in ${}^3P_3 = 3!$ ways giving rise to 6 distinct part sequences. Table 9 includes the optimal sequences for processing the parts in the MPS and the total processing time of the MPS, for each of these six part sequences with and without a buffer.

It is evident from Table 9 that all these 6 distinct part sequences use cycle S_4 repeatedly for

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	26.5	12.3
p_2	15.0	18.5
p_3	17.0	20.0

Table 8: Processing Time on Machines M_1 and M_2.

Sequence of Parts	Optimal Sequence of Cycles with Buffer	Processing Time with Buffer	Optimal Sequence of Cycles without Buffer	Processing Time without Buffer
p_1 p_2 p_3	S_4 S_4 S_4	96.0	S_2 S_2 S_2	99.0
p_1 p_3 p_2	S_4 S_4 S_4	96.0	S_2 S_2 S_2	102.5
p_2 p_1 p_3	S_4 S_4 S_4	96.0	S_2 S_2 S_2	102.5
p_2 p_3 p_1	S_4 S_4 S_4	96.0	S_2 S_2 S_2	99.0
p_3 p_1 p_2	S_4 S_4 S_4	96.0	S_2 S_2 S_2	99.0
p_3 p_2 p_1	S_4 S_4 S_4	96.0	S_2 S_2 S_2	102.5

Table 9: Optimal Sequence of Parts and Optimal Robot Move Sequences.

all the parts in the MPS and the optimal cycle time is 96 units with a buffer. We also observe that the optimal cycle time that can be accomplished with out a buffer is 99 time units by repeatedly processing the parts in the MPS using cycle S_2. We note that the savings accomplished with the buffer is 3 time units for a MPS.

Although the buffer helped in reducing the processing time of the MPS in the case discussed above, it does not always ensure reduction in the processing time as can be seen in the example illustrated below. In this example, the MPS comprises of three parts. The value of δ is 4; and the value of ε is 0.7. Table 10 gives the processing times α and β on Machines M_1 and M_2 respectively, for each part in the MPS. Table 11 includes the optimal sequences for processing the parts in the MPS and the total processing time of the MPS, for each of these six part sequences. We notice that the buffer has no effect on the cycle time and that the optimal cycle time is 80.6 units with and without a buffer. We can also infer that the buffer would not influence the cycle time when the processing times are relatively smaller than the robot move time. We know from Chapter 3 that a

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	2	4
p_2	3	5
p_3	4	3

Table 10: Processing Time on Machines M_1 and M_2.

Sequence of Parts			Optimal Sequence of Cycles with Buffer			Processing Time with Buffer	Optimal Sequence of Cycles without Buffer			Processing Time without Buffer
p_1	p_2	p_3	S_1	S_{12}	S_{21}	80.6	S_1	S_{12}	S_{21}	80.6
p_1	p_3	p_2	S_{12}	S_{21}	S_1	81.6	S_{12}	S_{21}	S_1	81.6
p_2	p_1	p_3	S_1	S_{12}	S_{21}	81.6	S_1	S_{12}	S_{21}	81.6
p_2	p_3	p_1	S_{12}	S_{21}	S_1	80.6	S_{12}	S_{21}	S_1	80.6
p_3	p_1	p_2	S_{21}	S_1	S_{12}	80.6	S_{21}	S_1	S_{12}	80.6
p_3	p_2	p_1	S_{21}	S_1	S_{12}	81.6	S_{21}	S_1	S_{12}	81.6

Table 11: Optimal Sequence of Parts and Optimal Robot Move Sequences.

buffer would not be necessary when the condition

$$\alpha + \beta \leq 2\delta$$

is satisfied. It is interesting to note that this condition is true in the above example.

5.5 Savings with a Buffer

In two machine with identical parts problem, we found that the maximum savings that can be accomplished with a buffer for each cycle is equal to 0.5δ. We do extensive testing to ascertain if the same would be true in two machine with multiple parts problem. We provide some examples to corroborate our remarks. In this example, the MPS comprises of three parts. The value of δ is 2; and the value of ε is 0.5. Table 12 gives the processing times α and β on Machines M_1 and M_2 respectively, for each part in the MPS. Table 13 includes the optimal sequences for processing the parts in the MPS and the total processing time of the MPS, for each of these six part sequences

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	10	20
p_2	15	5
p_3	8	20

Table 12: Processing Time on Machines M_1 and M_2.

Sequence of Parts			Optimal Sequence of Cycles with Buffer			Processing Time with Buffer	Optimal Sequence of Cycles without Buffer			Processing Time without Buffer
p_1	p_2	p_3	S_4	S_{43}	S_{34}	70.0	S_2	S_2	S_2	72.0
p_1	p_3	p_2	S_{34}	S_4	S_{43}	68.0	S_2	S_2	S_2	74.0
p_2	p_1	p_3	S_{43}	S_{34}	S_4	68.0	S_2	S_2	S_2	74.0
p_2	p_3	p_1	S_{43}	S_{34}	S_4	70.0	S_2	S_2	S_2	72.0
p_3	p_1	p_2	S_{34}	S_4	S_{43}	70.0	S_2	S_2	S_2	72.0
p_3	p_2	p_1	S_4	S_{43}	S_{34}	68.0	S_2	S_2	S_2	74.0

Table 13: Optimal Sequence of Parts and Optimal Robot Move Sequences.

with and without a buffer. It is evident from Table 13 that the optimal cycle time is 68 units with a buffer. We also observe that the optimal cycle time that can be accomplished with out a buffer is 72 time units by repeatedly processing the parts in the MPS using cycle S_2. We note that the savings accomplished with the buffer is 4 time units for a MPS which is a saving of 1.33 time units for each cycle. This saving is clearly greater than 0.5δ which is 1 time unit.

We consider a case where the MPS is equal to 3; δ is 2; and ε is 0.5. Table 14 gives the processing times α and β on Machines M_1 and M_2 respectively, for each part in the MPS. We observe that the processing time of the part on first machine is always greater than its respective processing time on the second machine. From Table 15, we note that the savings with buffer is equal to 3 time units which is a saving of 1 time unit for each cycle. This saving is equal to 0.5δ which is 1 time unit.

We consider a case where the processing time of the part on first machine is always less than its respective processing time on the second machine. In this example, the MPS is equal to 3; δ is 3.5; and ε is 0.2. Table 16 gives the processing times α and β on Machines M_1 and M_2 respectively, for

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	20	10
p_2	15	8
p_3	17	6

Table 14: Processing Time on Machines M_1 and M_2.

Sequence of Parts			Optimal Sequence of Cycles with Buffer			Processing Time with Buffer	Optimal Sequence of Cycles without Buffer			Processing Time without Buffer
p_1	p_2	p_3	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0
p_1	p_3	p_2	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0
p_2	p_1	p_3	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0
p_2	p_3	p_1	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0
p_3	p_1	p_2	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0
p_3	p_2	p_1	S_4	S_4	S_4	73.0	S_2	S_2	S_2	76.0

Table 15: Optimal Sequence of Parts and Optimal Robot Move Sequences.

each part in the MPS. It is obvious from Table 17, that the savings with buffer is equal to 1.75 time units which is a saving of 0.58 time units for each cycle. This saving is clearly less than 0.5δ which is 1.16 time units. Table 18 includes the optimal processing time with buffer, optimal processing time without buffer and the percentage saving in processing time with buffer for the examples of minimal part sets discussed in this chapter.

5.6 Necessity of Sequence S_5

In the identical parts two-machine cell problem, cycle S_5 was safely eliminated from the list of possible optimal cycles. S_5 was dominated by other cycles including S_1, S_2, S_3, S_4 and S_6. It would be interesting to study if the same would be true in a two-machine robotic cell with multiple parts problem. We test the program extensively to come up with a case where S_5 or one of the combination sequence involving S_5 such as S_{54}, S_{56}, S_{65} dominate every other sequence and become the optimal

Part	Processing Time on Machine M_1 α	Processing Time on Machine M_2 β
p_1	12	20
p_2	7	18
p_3	12	12.5

Table 16: Processing Time on Machines M_1 and M_2.

Sequence of Parts			Optimal Sequence of Cycles with Buffer			Processing Time with Buffer	Optimal Sequence of Cycles without Buffer			Processing Time without Buffer
p_1	p_2	p_3	S_{23}	S_{32}	S_2	82.65	S_2	S_2	S_2	84.40
p_1	p_3	p_2	S_4	S_4	S_4	83.20	S_2	S_2	S_2	84.40
p_2	p_1	p_3	S_4	S_4	S_4	83.20	S_2	S_2	S_2	84.40
p_2	p_3	p_1	S_{32}	S_2	S_{23}	82.65	S_2	S_2	S_2	84.40
p_3	p_1	p_2	S_2	S_{23}	S_{32}	82.65	S_2	S_2	S_2	84.40
p_3	p_2	p_1	S_4	S_4	S_4	83.20	S_2	S_2	S_2	84.40

Table 17: Optimal Sequence of Parts and Optimal Robot Move Sequences.

sequences. The example given in Table 4 and Table 7 testifies the necessity of considering the sequence S_5. It is obvious from the results shown in Table 7 that sequence S_{54} which is a combination of sequences S_5 and S_4 is an optimal sequence and dominates other feasible sequences.

5.7 Conclusion

The task of providing generalized conditions where a buffer is useful or not, like in Chapter 3 would be very complex owing to the number of variables involved. The number of variables is equal to

$$(\text{number of parts in MPS}) \; * \; (\alpha + \beta) + 2.$$

In this expression, the number 2 added represents the two variables δ and ε. However, the program can be executed to find out the optimal robot move sequences for the parts and thereby compute the optimal time for processing a pre-determined sequence of parts in the MPS. We also illustrate that our program can be used to simultaneously optimize both robot move sequences and part sequences,

MPS Data in Table	Optimal Processing Time with Buffer	Optimal Processing Time without Buffer	Percentage Saving with Buffer
3	151.00	153.00	1.30
6	96.00	99.00	3.03
8	80.60	80.60	0.00
10	68.00	72.00	5.55
12	73.00	76.00	3.95
14	82.65	84.40	2.07

Table 18: Percentage Saving with Buffer.

although in an inefficient manner. We find that it is not optimal to force all the parts in the MPS to use a single sequence and switching between the cycles while processing the MPS reduces the cycle time. It is a definite improvement over the three machine multiple part type results of Hall *et al.* [HKS97]. We observe that considerable savings in processing time can be accrued with the help of a buffer in certain cases. We also notice that a buffer is not beneficial when the processing times are relatively small compared to the robot move time. This observation is in conformance with a similar result in two-machine cell with identical parts problem. We also find that it is important to consider sequence S_5 unlike identical parts two-machine cell problem, where S_5 is dominated by other sequences.

Chapter 6

Conclusions and Future Work

We have investigated the usefulness of a buffer in reducing the cycle time in a two machine robotic cell producing identical parts. We also extended our research to the two machine robotic cell producing multiple parts. Several closely related problems remain as a topic for future research. Among these is the joint robot move and part sequencing problem. Extending the problem to different robotic cellular layouts, Alternative assumptions about the engineering characteristics of the robot and new performance measures provide several open research issues.

6.1 Conclusions

The important contribution of this thesis lies in the fact that it identifies and quantifies the effectiveness of a buffer in reducing the manufacturing time and thereby improving the efficiency of the robotic cells. We believe that both practitioners and robotized cell designers who are constantly on the look out for improving the manufacturing productivity will greatly benefit from this study.

In this thesis we studied the impact of the buffer with a capacity to store a single part on the cycle time of a two machine robotic cell. In the first part, we consider the problem of finding all feasible cyclic schedules in a two machine robotic cell producing identical parts with and with out a buffer. We adopt analytical methods to compute the cycle times for the six possible cyclic sequences. By comparing the cycle times we suggest the optimal cyclic sequences to be used under different conditions. We also provide conditions where a buffer is useful. We also quantify the reduction in cycle time achieved with the use of a buffer. The savings with buffer are low as expected; because in a single part type problem the only possible savings are in the robot move time as the processing times of all the parts are same.

In the second part of the thesis, we deal with the problem of identifying optimal sequence of robot moves in a two machine robotic cell producing multiple parts with and with out a buffer. We provide all 23 feasible sequences both cyclic and combination along with their sequence times. We provide and implement an algorithm that suggests the optimal sequence for each part in the minimal part set, thereby minimizing the cycle time needed for repetitive manufacture of a given arrangement of parts in a MPS. We also demonstrate that switching between sequences while processing the parts in MPS is beneficial in reducing the cycle time. We provide some conclusive remarks as to when a buffer is needed and also give an approximate percentage of savings in manufacturing time, accomplished with the buffer. Considerable savings are obtained by going to a buffer in multiple part type problem as savings are possible while balancing the processing times of different parts.

Material handling devices other than robots generate similar problems with related scheduling issues. There are a lot of similarities between material handling devices in a computer integrated manufacturing environment. Among these devices are cranes, mono-rails, and Automated Guided

Vehicles (AGV's). These devices are frequently used in industry with limited storage buffers. Important insights about various problems arising in environments served by these devices can be gained from a study of robot-served manufacturing cells.

6.2 Future Work

Some of the important directions for future work are the following:

- Optimal arrangement of parts in a MPS while optimally sequencing robot moves in a cell. The present work can be used as a basis to assess the reduction in the manufacturing time of a MPS, that can be accomplished by the optimal arrangement of parts in the MPS.
- The present work could be extended to a three machine robotic cell with and without a buffer.
- Scheduling in robotic cells under uncertainty (for example stochastic processing times).
- The trade offs between throughput rate and inventory holding cost resulting from limited buffer storage within the cell as a means of reducing cycle time.
- Alternative assumptions about the engineering characteristics of the robot demand new analysis. For instance a robot equipped with a dual gripper to unload a finished part and load a new part at a single visit to a machine would demand a new analysis beginning with the derivation of the cycle time.
- Robots are dealt in a flow shop environment where the routing of all part types is same that is all parts go to all the machines. It could be dealt in a job shop environment where the routing of different parts is different.
- Our study is based on an environment requiring the repetitive manufacture of a small variety of similar items, which makes the cycle time criterion a natural one to consider, other criteria such as minimizing the make span for a fixed set of parts could also be considered. Other objectives also become relevant, including those with due dates at which parts have been

promised to customers, those with varying times at which parts become available for processing, and those with different weights (or values) between the parts. Examples of objectives which can usefully be studied include the minimization of: the total weight of parts delivered late, the total lateness of the late parts, and the lateness of the latest part.

Bibliography

[AC90] R.G. Askin and K.S. Chiu. A graph partitioning procedure for machine assignment and cell formation in group technology. *International Journal of Production Research*, 28:1555–1572, 1990.

[Asf85] C.R. Asfahl. *Robots and Manufacturing Automation*. John Wiley & Sons, New York, 1985.

[AW87] E. Appleton and D.J. Williams. *Industrial robot applications*. Halsted Press; Milton Keynes: Open University Press, New York, 1987.

[Bak74] K. R. Baker. *Introduction to sequencing and scheduling*. Wiley, New York, 1974.

[BB86] J. Badalamenti and H. Bao. *Robotics and Material Flows*, chapter Simulation Modeling of Robotic Assembly Cell incorporating Carousel. Elsevier Press, New York, 1986. (Ed.) S.Y. Nof.

[BBH+81] W. Baumann, R. Birner, J. Haensler, R.P. Hartmann, and A.G. Stevens. Operating and idle times for cyclic multi-machine servicing. *The Industrial Robot*, pages 44–49, March 1981.

[BLS79] R. Bedini, G.G. Lisini, and P. Sterpos. Optimal programming of working cycles for industrial robots. *Journal of Mechanical Design, Transactions of the ASME*, 101:250–257, 1979.

[BS93] J.T. Black and B.J. Schroer. Simulation of an apparel assembly cell with walking workers and decouplers. *Journal of Manufacturing Systems*, 12:170–180, 1993.

[Bur75] J.L. Burbidge. *The Introduction of Group Technology*. John Wiley & Sons, New York, 1975.

[CH82] B.H. Claybourn and J.R. Hewit. Simulation of activity cycles for robot served manufacturing cells. In *Proceedings of the First International Conference on Flexible Manufacturing Systems*, pages 443–452, 1982.

[Chu93] C.H. Chu. Manufacturing cell formation by competitive learning. *International Journal of Production Research*, 31(4):829–843, 1993.

[CK97] Y. Crama and J. Klundert. Cyclic scheduling of identical parts in a robotic cell. *Operations Research*, 1997.

[CM82] H.M. Chan and D.A. Milner. Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*, 1:64–76, 1982.

[Fre82] S. French. *Sequencing and scheduling: an Introduction to the mathematics of the job-shop*. Wiley, Chichester, West Sussex: E Horwood; New York, 1982.

[GG64] P.C. Gilmore and R.E. Gomory. Sequencing a one state-variable machine: a solvable case of the travelling salesman problem. *Operations Research*, 12:655–679, 1964.

[Gro87] M.P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[Hal88] R.W. Hall. Cyclic scheduling for improvement. *International Journal of Production Research*, 26(3):457–472, 1988.

[Har83] J. Hartley. *Robots at Work*. North-Holland Publishing Company, 1983.

[HKS95] N.G. Hall, H. Kamoun, and C. Sriskandarajah. Scheduling in robotic cells: complexity and steady state analysis. Working Paper 95-21, 1995.

[HKS97] N. G. Hall, H. Kamoun, and C. Sriskandarajah. Scheduling in robotic cells: classification, two and three machine cells. *Operations Research*, 45(3):421–439, 1997.

[HS96] N.G. Hall and C. Sriskandarajah. A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44(3):510–525, 1996.

[HY86] K. Hitomi and M. Yoshimura. *Robotics and Material Flows*, chapter Operations scheduling for work transportation by industrial robots in automated manufacturing systems, pages 131–139. Elsevier Press, New York, 1986. (Ed.) S.Y. Nof.

[IS82] H. Inaba and S. Sakakibara. Flexible automation - unmanned machining and assembly cells with robots. In *Proceedings of the 1st International Conference on Flexible Manufacturing Systems*, 1982.

[Kam94] H. Kamoun. *Scheduling in Repetitive Manufacturing Systems: Complexity, Heuristic Algorithms and System Design*. PhD thesis, Department of Industrial Engineering, University of Toronto, Toronto, Canada, June 1994.

[KHS95] H. Kamoun, N.G. Hall, and C. Sriskandarajah. Scheduling in robotic cells: Heuristics and cell design. Working Paper 93-08, 1995.

[Kin80] J.R. King. Machine-component grouping in production flow analysis: an approach using rank order clustering algorithm. *International Journal of Production Research*, 18:213–232, 1980.

[Kon79] A.S. Kondoleon. Cycle time analysis of robot assembly systems. In *Proceedings of the Ninth Symposium on Industrial Robots*, pages 575–587, March 1979.

[KS84] E. Kafrissen and M. Stephans. *Industrial Robots and Robotics*. Reston Books, New York, 1984.

[Kus87] A. Kusiak. The generalized group technology concept. *International Journal of Production Research*, 25:561–569, 1987.

[LS96] R. Logendran and C. Sriskandaraja. Sequencing of robot activities and parts in two-machine robotic cells. *International Journal of Production Research*, 34(12):3447–3463, 1996.

[Mar89] J.M. Martin. Cells drive manufacturing strategy. *Manufacturing Engineering*, 1:49–54, 1989.

[McA72] J. McAuley. Machine grouping for efficient production. *The Production Engineer*, 51:53–57, 1972.

[NH88] Y.D. Noh and B. Herring. Simulation model for an individual robotic manufacturing cell. *International Journal of Production Research*, 26(1):63–79, 1988.

[RB75] R. Rajagopalan and J.L. Batra. Design of cellular production systems: a graph theoretic approach. *International Journal of Production Research*, 13:567–579, 1975.

[SB97] D. Sipper and R. Bulfin. *Production: Planning, Control and Integration*. McGraw-Hill series in industrial engineering and management science, 1997.

[SHKW95] C. Sriskandarajah, N.G. Hall, H. Kamoun, and H. Wan. Scheduling in large robotic cells. Working Paper 94-11, 1995.

[SSN85] A. Seidmann, J.P. Schweitzer, and S.Y. Nof. Performance evaluation of a flexible manufacturing cell with random multiproduct feedback flow. *International Journal of Production Research*, 23(6):1171–1184, 1985.

[SSS+92] S.P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak. Sequencing of parts and robot moves in a robotic cell. *The International Journal of Flexible Manufacturing Systems*, 4:331–358, 1992.

[SW86] H. Seifoddini and P.M. Wolfe. Application of the similarity coefficient method in group technology. *IIE Transactions*, 18:271–277, 1986.

[VBW92] T.E. Vollmann, W.L. Berry, and D.C. Whybark. *Manufacturing planning and control systems*. Richard D. Irwin, Inc, 1992.

Appendix A

Initial Steps of Cyclic Sequences

This chapter comprises of preliminary sequence of robot moves which precede the cycles S_2, S_3, S_4, S_5 and S_6. These steps start from the initial state of the system and end in the system state where the robotic move becomes the initial step of a cyclic sequence. The initial state of the system being the system state where both the machines and the buffer are empty and the robot is at the input. We outline the initial steps which lead to the cycles S_2, S_3, S_4, S_5 and S_6 discussed in Chapters 3 and 4.

A.1 Initial Steps of Cycle S_2

The initial steps which precede the cyclic sequence S_2 are given below.

1. Robot picks up a part at the Input:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

4. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

5. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p).$$

6. Robot travels to M_1 and loads the part on machine M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

7. Robot travels to machine M_2, waits if necessary and unloads the part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^+}^p).$$

8. Robot carries the part to output and drops it at the output:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_O^p).$$

A.2 Initial Steps of Cycle S_3

The initial steps which lead us to the cyclic sequence S_3 are:

1. Robot picks up a part at the Input:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

4. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

5. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p).$$

6. Robot travels to machine M_1 and loads the part on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

7. Robot waits at machine M_1 until the part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

8. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

9. Robot travels to machine M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p).$$

10. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

A.3 Initial Steps of Cycle S_4

The initial steps which precede the cycle S_4 are as follows:

1. Robot picks up a part at the Input:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till the part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

4. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

5. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_I^p).$$

6. Robot travels to machine M_1 and loads the part on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

7. Robot waits at machine M_1 till the part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

8. Robot carries the part to the buffer and drops it in the buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

9. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_I^p).$$

10. Robot carries the part to M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

11. Robot travels to M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^-}^p).$$

12. Robot carries part to the output and drops it at O:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

A.4 Initial Steps of Cycle S_5

The initial steps which lead us to the sequence S_5 are:

1. Robot picks up a part at the Input:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

4. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

5. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_I^p).$$

6. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

7. Robot travels to the buffer and picks the part from buffer:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^+}^p).$$

8. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

9. Robot travels to machine M_1, waits if necessary and unloads the part from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_1^+}^p).$$

10. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^p, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

11. Robot travels to machine M_2, waits if necessary and unloads part from M_2:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_2^+}^p).$$

12. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_O^p).$$

A.5 Initial Steps of Cycle S_6

The initial steps which lead us to the sequence S_6 are as follows:

1. Robot picks up a part at the Input:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_I^p).$$

2. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

3. Robot waits at machine M_1 till part is processed and unloads it from M_1:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_1^-}^p).$$

4. Robot travels to the buffer and drops the part in buffer:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{B^-}^p).$$

5. Robot travels to the input and picks up a part:

$$(\mathcal{M}_1^0, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_I^p).$$

6. Robot carries the part to machine M_1 and loads it on M_1:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^p, \mathcal{R}_{M_1^-}^p).$$

7. Robot travels to the buffer and picks the part from buffer:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{B^-}^p).$$

8. Robot travels to machine M_2 and loads the part on M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

9. Robot waits at M_2 till the part is processed and unloads part from M_2:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_{M_2^-}^p).$$

10. Robot travels to the output and drops part at the output:

$$(\mathcal{M}_1^p, \mathcal{M}_2^0, \mathcal{B}^0, \mathcal{R}_O^p).$$