Technical Research Report

Sequencing Wafer Handler Moves to Improve the Performance of Sequential Cluster Tools

by Jeffrey W. Herrmann, Manh-Quan T. Nguyen

T.R. 2000-3

Sequencing wafer handler moves to improve the performance of sequential cluster tools

Jeffrey W. Herrmann and Manh-Quan T. Nguyen
Department of Mechanical Engineering and Institute for Systems Research
University of Maryland
College Park, MD 20742
July 25, 2000

Abstract

Cluster tools are highly integrated machines that can perform a sequence of semiconductor manufacturing processes. The sequence of wafer handler moves affects the total time needed to process a set of wafers. Reducing this time can reduce cycle time, reduce tool utilization, and increase tool capacity. This paper introduces the cluster tool scheduling problem for sequential cluster tools and describes a branch-and-bound algorithm that can find an optimal sequence of wafer handler moves. In addition, we enumerate the set of 1-unit cyclic sequences for two- and three-stage sequential cluster tools. Experimental results show that the tool performance can be improved significantly if the wafer handler follows a cyclic sequence instead of using a dispatching rule.

1. Introduction

Manufacturing semiconductor devices involves three main steps: formation of p and n-type regions of the required conductivity within the semiconductor chip by doping; formation of reliable metal-semiconductor contacts on the surface of the chip; and encapsulation and packaging of the chip to provide protection and a convenient method of making electrical connection. In the first and second steps, the chips are processed together as wafers.

Most operations process each wafer individually. However, identical wafers move together from one process to the next. Each set of wafers is a lot. The container used to move and store the wafers in a lot is called a cassette.

A cluster tool is a manufacturing system with integrated processing modules linked mechanically. Typical cluster tools include load locks that store cassettes of wafers (cassette modules), process modules that modify the properties of the wafers, and single or multiple wafer handler(s) that transport the wafers (transport modules). These modules are linked together by an evacuated transfer space. Because it has multiple chambers, a cluster tool can process multiple wafers simultaneously.

After a lot enters the cluster tool, each wafer must undergo a series of activities. Each activity is performed in a different chamber. The wafer handler transports each wafer from one chamber to another. For example, the cluster tool shown in Figure 1 has one load lock (LL), which stores the cassette of wafers (unprocessed and completed), and three process stages. Each wafer starts in the LL and must visit the first-stage chamber, the second-stage chamber, the third-stage chamber, before returning to LL. Note that there are no locations to store wafers between process steps.

A sequential cluster tool has one chamber for each stage, so each wafer must visit every chamber. A hybrid cluster tool will have one or more stages with at least two identical chambers that are used in parallel. In a hybrid tool, a wafer visits only one of the chambers at each stage. A cluster tool can improve yield and device performance since wafers are exposed to fewer contaminates between process steps. The tool can include an in-situ metrology step that provides real-time feedback on process performance. A cluster tool with multiple parallel chambers can increase capacity and reduce cycle times by reducing the total time needed to process a lot of wafers. Moreover, it may be more reliable, since a single chamber's failure does not necessarily stop production. Semiconductor manufacturers are increasingly using cluster tools. Annual sales of cluster tools is projected to increase from $11.2 billion in 1997 to $21.9 billion in 2000 [1].

The sequence of wafers leaving the LL is not important, since the wafers are identical, and an activity's time is the same for every wafer. But the sequence of wafer handler moves, which determines when each activity starts, will change the total time needed to process a lot of wafers. We will call this the lot makespan. This paper addresses the problem of sequencing the wafer handler moves to minimize the lot makespan. Reducing the lot makespan can reduce cycle time, reduce tool utilization, and increase tool capacity. Moreover, the lot makespan is a necessary component for calculating overall equipment effectiveness (OEE) and cost-of-ownership (COO), which are usually used to evaluate cluster tool performance [10, 11].

Like machine tools, cluster tools use controllers that supervise the tool operations, monitor the tool status, and handle exceptions that abnormal events cause. Under normal operation, sequencing wafer handler moves is an important responsibility. In practice, cluster tools use a push dispatching rule or a pull dispatching rule to sequence the wafer handler moves. After completing one move, the wafer handler will wait where it is (if no more wafers are ready to move) or start another move (if at least one wafer is ready). If multiple wafers are ready to be moved, the cluster tool must decide which move the wafer handler will perform.

In this case, the dispatching rule selects the next move. The pull rule gives priority to the wafer that has fewer remaining process steps. The push rule gives priority to the wafer that has more remaining process steps. Consider the cluster tool in Figure 1. Suppose there are unprocessed wafers in the LL, the first stage chamber is empty, and the second and third stage chambers each have a wafer that has finished processing and needs to move to the next stage. The pull rule will give priority to the wafer in the third stage chamber. The push rule will give priority to the next unprocessed wafer in LL that needs to visit the first stage chamber. Note that the wafer in the second-stage chamber cannot be moved because the third-stage chamber is full.

Although these rules help the cluster tool sequence the wafer handler moves, the push and pull dispatching rules do not guarantee that the resulting sequence has the optimal lot makespan for the given lot size, tool configuration, and activity processing times. For instance, consider a three-stage cluster tool that has one chamber in each stage. Each first-stage activity requires 5 seconds. Each second-stage activity requires 10 seconds. Each third-stage activity requires 10 seconds. A wafer handler move requires 5 seconds. The lot has five wafers. Figure 2 presents the Gantt charts of three sequences. The first sequence is an optimal sequence, which has a lot makespan of 185. The second sequence is the sequence that the push dispatching rule generates. The third sequence is the sequence that the pull dispatching rule generates. Neither one has an optimal lot makespan. Each Gantt chart has four rows. The bottom row displays the wafer handler activities. The row above that displays the first-stage activities. The row above that displays the second-stage activities. The top row displays the third-stage activities.

Cluster tool performance can be improved by determining a good sequence of wafer handler moves and providing it to the cluster tool controller, which can then use this sequence to direct normal operations. We will treat the problem as a deterministic machine scheduling problem, since the processing and move times have little variation, and small variations do not invalidate a given sequence.

This paper presents a branch-and-bound algorithm that can find the optimal sequence of wafer handler moves. It also discusses and enumerates a special class of cyclic sequences for two- and three-stage sequential cluster tools. The remainder of this paper is organized as follows. Section 2 reviews the related literature. Section 3 formulates the problem. Section 4 presents the cyclic sequences for two-stage cluster tools. Section 5 presents the cyclic sequences for three-stage cluster tools. Section 6 describes the branch-and-bound algorithm. Section 7 presents experimental results on a range of problem instances. Section 8 summarizes our results and concludes the paper.

2. Related Literature

Wood [2] derives formulas that relate the total lot processing time to the number of wafers in the lot for ideal sequential and parallel tools. Considering the transitions at the beginning and the end of the lot, Perkinson *et al.* [3] derive a model that relates the total lot processing time to the number of wafers. Both papers present linear models and identify two operating regions: in one region, the total lot processing time is constrained by the wafer handling time; in the other region, by the module process time. Venkatesh *et al.* [4] analyze the throughput of a sequential cluster tool with a dual-blade wafer handler. They also identify conditions when the tool operation is constrained by the wafer handler. Srinivasan [5] presents more detailed Petri net models for sequential and parallel tools and uses these to determine the steady state behavior of the tool. Herrmann *et al.* [6] study the impact of process changes on cluster tool performance. They propose using a network model for a prespecified sequence of wafer moves and cluster tool simulation software when the controller uses a dispatching rule or scheduling algorithm to sequence the wafer moves. They choose the cluster tool performance measure of interest is the lot makespan. None of the previous work addresses the problem of reducing the total lot processing time (lot makespan) by sequencing the wafer handler moves.

Jeng *et al.* [7] study the problem of sequencing robot activities for a robot-centered parallel-processor workcell where n jobs and m identical processors exist in the cell. They provide a branch and bound algorithm to find an optimal sequence of robot activities, which minimizes the total completion times. Hall *et al.* [8] discuss the problem of scheduling activities in a serial two or three machine manufacturing cell that is served by a robot. For multiple part-type problems in a two-machine cell, they provide an algorithm that simultaneously finds sequences of parts and robot moves to minimize the steady state cycle time. They also address a conjecture about the optimality of repeating one-unit cycles for a three-machine cell with general data and identical parts. Restricted to a special problem where the number of machines is arbitrary, but all parts are of the same type, Crama and van de Klundert [9], relying on the concept of pyramidal permutation, present a dynamic programming approach that finds an minimum one-unit cycle time in $O(m^3)$ time.

3. Problem Statement

This paper focuses on single load lock, single wafer handler cluster tools. The following information about the cluster tool scheduling problem is given. The cluster tool has one load lock (LL) and S stages (S > 1). In a sequential tool, each stage has one chamber, so the chambers are numbered 1 to S. Each stage S_i has a wafer processing time p_i. The wafer handler move time is p_r. The lot has L identical wafers. Since each wafer must visit each stage and return to LL, the total number of moves is L(S+1).

The sequence of wafers leaving LL is not important, since the wafers are identical. However, the sequence of moves affects the lot makespan C_{max}, the total time needed to complete all moves. The scheduling objective is to minimize the lot makespan. By convention, scheduling problems are described by triplets of the form $\alpha \mid \beta \mid \gamma$. The α field describes the machine environment. We use α = CT1-1 to describe two-stage sequential cluster tools and CT1-1-1 to describe three-stage sequential cluster tools. For our problem, the objective function $\gamma = C_{max}$.

When processing begins, the wafer handler is at LL, and all of the wafers are unprocessed and in LL. For convenience, we will number the wafers in the order they leave LL. Let $R_{0,j}$ denote the move that takes wafer j from

LL to S_1. Let $R_{i,j}$ denote the move that takes wafer j from S_i to S_{i+1} (i = 1, ..., S-1). Let $R_{S,j}$ denote the move that takes wafer j from S_S to LL.

A feasible sequence of moves must satisfy the following constraints. All wafers must follow the fixed sequence of processing steps. Therefore, for all j = 1, …, L, and i = 0, ..., S-1, $R_{i,j}$ must precede $R_{i+1,j}$. Since there are no buffers (besides LL) to store wafers, the chamber at S_{i+1} must be free before $R_{i,j}$ begins. That is, the wafer handler must have moved the previous wafer to the next stage. Thus, $R_{i+1,j-1}$ must precede $R_{i,j}$. for all j = 2, …, L and i = 0, …, S-1.

The following facts describe the operation of the cluster tool. Each and every move requires the wafer handler. Since there is just one wafer handler, then, at any time, there is at most one move in process. The wafer handler cannot unload an empty or busy chamber and cannot load a busy or full chamber. (A full chamber has a wafer that has completed processing and is waiting to be moved.)

The chamber at stage S_i begins processing wafer j when move $R_{i-1,j}$ ends (i = 1, ..., S). This activity cannot be interrupted until the chamber is finished processing the wafer. For example, if S_i starts processing at time t, then the chamber is busy during the interval [t, t + p_i], and the wafer cannot be unloaded during that time.

Move $R_{i,j}$ starts when the chamber finishes processing wafer j and the wafer handler completes any previous move. $R_{i,j}$ requires p_r time units if the wafer handler is already at the chamber that processed wafer j (at LL if the move is $R_{0,j}$). $R_{i,j}$ requires $2p_r$ time units otherwise, for the wafer handler must move to the correct chamber at S_i before moving the wafer to a chamber at stage S_{i+1} (to LL if the move is $R_{S,j}$). The wafer handler cannot make anticipatory moves. That is, the wafer handler cannot move to the chamber before processing ends.

That is, $R_{0,1}$ requires p_r time units. For j ≥ 2, $R_{0,j}$ requires p_r time units if and only if the previous move is $R_{S,k}$ for some k < j. For i ≥ 1 and j ≥ 1, $R_{i,j}$ requires p_r time units if and only if the previous move is $R_{i-1,j}$.

Special cases. We can identify two special cases. If $p_r = 0$, there is no scheduling problem since moves require no time, and all wafers move as soon as they are ready. If all $p_i = 0$, then an optimal solution is R_{01}, R_{11}, …, R_{S1}, R_{02}, R_{12}, …, R_{S2}, …, R_{0L}, R_{1L}, …, R_{SL}. This sequence has a lot makespan of $L(S+1)p_r$.

Cyclic sequences. Unless the lot size L is very small, a typical sequence has three phases, which we label filling-up, steady state (or cyclic), and completion. The chambers are empty when processing begins. Until the first wafer is completed, the tool is filling up with wafers. Then the tool is in a steady-state phase as it completes wafers and loads new wafers. When there are no more wafers to start, the tool enters the completion phase and completes wafers until the last wafer is unloaded from the last stage. Then processing ends.

Let us define a λ-unit cycle as a subsequence that loads and unloads each stage λ times and thus completes λ wafers. Complete sequences formed by repeating a cycle in the steady state and completion phase we will call λ-unit cyclic sequences.

Consider optimal sequence presented in Figure 2. The cyclic phase, which starts at time 55, has three 1-unit cycles: $R_{0, p+1} – R_{2, p} – R_{1, p+1} – R_{3, p}$, for p = 2, 3, and 4. In the push sequence, the cyclic phase starts at time 65 and has two 1-unit cycles: $R_{2, q} – R_{1, q+1} – R_{0, q+2} – R_{3, q}$, for q = 2 and 3.

The order of events in the completion phase resembles that in the cyclic phase. The five moves in the completion phase of the push sequence form two incomplete cycles: the first is $R_{2, L-1} – R_{1, L} – R_{3, L-1}$; and the second is $R_{2, L} – R_{3, L}$. The cycles are incomplete because there are no unprocessed wafers and some moves are no longer needed.

Note that the cycle does not define the filling-up phase, which ends with the first wafer being completed. Although there may exist more than one feasible filling-up phase for a given cycle, the class of cyclic sequences is small enough to enumerate for CT1-1 and CT1-1-1.

4. Cyclic Sequences for CT1-1

In this section, we analyze the cycle time and lot makespan of the two 1-unit cyclic sequences that are feasible for a two-stage sequential cluster tool. There are two feasible cycles, which we label σ1 and σ2.

σ1: $R_{2,j-1} – R_{0,j} – R_{1,j} – R_{2,j}$ (j = 2,..., L),

σ2: $R_{2,j-1} – R_{1, j} – R_{0,j+1} – R_{2,j}$ (j = 2,..., L).

Each cycle has only one feasible filling-up phase and thus forms exactly one cyclic sequence. $R_{0, 1} – R_{1, 1}$ is the filling-up phase for σ1. $R_{0, 1} – R_{1, 1} – R_{0, 2}$ is the filling-up phase for σ2. For x = 1 and 2, let P_x be the cycle time of σx, and let MS_x be the lot makespan of the cyclic sequence formed by σx.

Theorem 1. $P_1 = 3p_r + p_1 + p_2$, $MS_1 = LP_1$. $P_2 = 4p_r + \max\{2p_r, p_1, p_2\}$, $MS_2 = 3p_r + p_1 + p_2 + (L-1)P_2$.

Proof. The proof can be found in Appendix A.

Theorem 2. For CT1-1, if $p_r \geq p_1$ and $p_r \geq p_2$, then $P_2 > P_1$ and $MS_2 > MS_1$.

Proof. Under these conditions, $P_2 = 4p_r + \max\{2p_r, p_1, p_2\} = 6p_r$, and $P_1 = 3p_r + p_1 + p_2 \leq 5p_r < P_2$. Then, $MS_2 > MS_1$ since $MS_2 - MS_1=(L-1)(P_2-P_1) > 0$.

5. Cyclic Sequences for CT1-1-1

In this section, we analyze the cyclic sequences formed by the 1-unit sequences for a three-stage sequential cluster tool. There are six 1-unit cycles, which we denote σ1 through σ6. (Note these are the same six cycles that Sethi *et al.* [12] identify.)

σ1: $R_{3,j-1} – R_{0,j} – R_{1,j} – R_{2,j} – R_{3,j}$.
σ2: $R_{3,j-1} – R_{0,j+1} – R_{2,j} – R_{1,j+1} – R_{3,j}$.
σ3: $R_{3,j-1} – R_{2,j} – R_{0,j+1} – R_{1,j+1} – R_{3,j}$.
σ4: $R_{3,j-1} – R_{1,j} – R_{2,j} – R_{0,j+1} – R_{3,j}$.
σ5: $R_{3,j-1} – R_{1,j} – R_{0,j+1} – R_{2,j} – R_{3,j}$.
σ6: $R_{3,j-1} – R_{2,j} – R_{1,j+1} – R_{0,j+2} – R_{3,j}$.

The cycle σ1 has one feasible filling-up phase. Each of the other five cycles has two feasible filling-up phases. Thus, there are eleven feasible cyclic sequences for CT1-1-1.

Theorem 3. For CT1-1-1, the cycle time and lot makespan of a 1-unit cyclic sequence that uses cycle σ1, σ3, σ4, or σ5 (and a feasible filling-up phase) will equal the expressions given in the following table.

Cycle	Cycle time	Feasible filling-up phase	Lot makespan
σ1	$P_1 = 4p_r + p_1 + p_2 + p_3$	f1 = $R_{0,1} – R_{1,1} – R_{2,1}$	$MS_{11} = LP_1 = 4p_r+p_1+p_2+ p_3+(L-1)P_1$
σ3	$P_3 = 4p_r + \max\{3p_r + p_1, p_3, p_r + p_1 + p_2\}$	f1 = $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2} – R_{1,2}$	$MS_{31} = 4p_r+p_1+p_2+p_3+(L-1)P_3$
		f2 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1} – R_{1,2}$	$MS_{32} = 9p_r + p_1 + p_3+ \max\{p_2, 2p_r\}+ \max\{p_1, p_2, p_3, 2p_r\}+ (L-2)P_3$
σ4	$P_4 = 5p_r+p_2+\max\{2p_r, p_3, p_1\}$	f1 = $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2}$	$MS_{41} = 4p_r+p_1+p_2+p_3+(L-1)P_4$
		f2 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1}$	$MS_{42} = 8p_r + p_1 + p_2 + p_3 + \max\{p_1, p_r + p_3 + \max(p_2, 2p_r)\}+ (L – 2)P_4$
σ5	$P_5 = 4p_r+\max\{p_r+p_2+p_3, 3p_r+p_3, p_1\}$	f1 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1}$	$MS_{51} = 4p_r+p_1+p_2+p_3+(L-1)P_5$
		f2 = $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2}$	$MS_{52} = 9p_r + p_1 + 2p_2 + p_3 + \max\{p_1, p_3, 2p_r\}+ (L - 2)P_5$

Proof. The proof can be found in Appendix B.

Theorem 4. For CT1-1-1, the cycle time and lot makespan of the 1-unit cyclic sequences that use cycle σ2 can be determined as follows. For filling-up phase f1 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1} – R_{1,2}$, the average cycle time is P_{21} and the lot makespan is MS_{21}.

$$P_{21} = \frac{1}{L-2}\sum_{j=2}^{L-1} P_{21j}.$$

$$MS_{21} = 13p_r + p_1 + p_3 + a_1 + b_1 + c_1 + (L-2)\,P_{21} + \max(p_2 - 2p_r - c_{L-1}, 0),$$

where

$$b_1 = \max(p_2 - 2p_r, 0) = \max(p_2, 2p_r) - 2p_r,$$
$$a_1 = \max\{p_1 - \max(p_2, 2p_r), 0\}, \text{ and}$$
$$c_1 = \max(p_3 - 2p_r - a_1, 0);$$

and, for j = 2, …, L-1,

$$b_j = \max(p_2 - 3p_r - c_{j-1}, 0),$$
$$a_j = \max(p_1 - 2p_r - b_j, 0),$$
$$c_j = \max(p_3 - 2p_r - a_j, 0), \text{ and}$$
$$P_{21j} = 7p_r + a_j + b_j + c_j.$$

For filling-up phase f2 = $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2} – R_{1,2}$, the average cycle time is P_{22} and the lot makespan is MS_{22}.

$$P_{22} = \frac{1}{L-2}\sum_{j=1}^{L-2} P_{22j} .$$

$$MS_{22} = 8p_r + p_1 + p_2 + p_3 + \max\{p_3, 3p_r + p_1\} + \max\{p_2 - 2p_r - g_{L-2}, 0\} + (L-2)P_{22},$$

where

$$e_1 = \max\{p_1 + p_2 - \max(p_3, p_1 + 3p_r), 0\},$$
$$d_1 = \max\{p_1 - 2p_r - e_1, 0\}, \text{ and}$$
$$g_1 = \max\{p_3 - 2p_r - d_1, 0\};$$

for j = 2, ..., L-2,

$$e_j = \max\{p_2 - 3p_r - g_{j-1}, 0\},$$
$$d_j = \max\{p_1 - 2p_r - e_j, 0\}, \text{ and}$$
$$g_j = \max\{p_3 - 2p_r - d_j, 0\};$$

and, for j = 1, ..., L-2,

$$P_{22j} = 7p_r + d_j + e_j + g_j.$$

Proof. The proof can be found in Appendix C.

Theorem 5. For CT1-1-1, the cycle time and lot makespan of the 1-unit cyclic sequences that use cycle σ6 can be determined as follows:

For filling-up phase f1 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1} – R_{1,2} – R_{0,3}$, define the following quantities:

$$w_2 = \max\{p_1 - \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\}, 0\}$$

For j = 3, …, L-2,

$$w_j = \max\{p_1 - \max(p_2, p_3 - w_{j-1}, 4p_r), 0\}.$$

For j = 2, …, L-2,

$$P_{61j} = 4p_r + w_j + \max\{p_2, p_3 - w_j, 4p_r\}.$$

Then, the average cycle time

$$P_{61} = \frac{1}{L-3}\sum_{j=2}^{L-2} P_{61j}.$$

The lot makespan

$$MS_{61} = 13p_r + p_1 + p_3 + w_{L-2} + \max(2p_r, p_1, p_2) + \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\} + \\ + \max\{2p_r, p_2, p_3 - w_{L-2}\} + (L-3)P_{61}.$$

For filling-up phase f2 = $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2} – R_{1,2} – R_{0,3}$, define the following quantities:

$$v_2 = \max\{p_1 - \max[p_2, 4p_r, p_3 - p_1 - p_r], 0\}$$

For all j = 3, ..., L-2,

$$v_j = \max\{p_1 - \max(p_2, p_3 - v_{j-1}, 4p_r), 0\}.$$

For all j = 2, ..., L-2,

$$P_{62j} = 4p_r + v_j + \max\{p_2, p_3 - v_j, 4p_r\}.$$

Then, the average cycle time

$$P_{62} = \frac{1}{L-3}\sum_{j=2}^{L-2} P_{62j}.$$

The lot makespan

$$MS_{62} = 12p_r + p_1 + p_2 + p_3 + v_{L-2} + \max\{p_r + p_1 + p_2, p_3, 5p_r + p_1\} + \max\{2p_r, p_2, p_3 - v_{L-2}\} + (L-3)P_{62}.$$

Proof. The proof can be found in Appendix D. □

Hall *et al.* [8] show that, if a mobile-robot cell has three machines and repeats cycle σ2, then the cycle time is constant. However, this does not hold for CT1-1-1. For example, Table 1 presents, for some instances, the length of each cycle, the average cycle time, and the lot makespan of the 1-unit cyclic sequence that use cycle σ2 and filling-up phase f1. Table 2 presents, for the same instances, the length of each cycle, the average cycle time, and the lot makespan of the 1-unit cyclic sequence that use cycle σ2 and filling-up phase f2. In all instances, $p_r = 1$, $p_1 = 5$, and $p_3 = 10$.

Table 1. One-unit cyclic sequences that use cycle σ2 and filling-up phase f1.

	P_{21j}								P_{21}	MS_{21}
	j = 2	j = 3	j = 4	j = 5	j = 6	j = 7	j = 8	j = 9		
p_2 = 20, L = 9	24	24	24	24	24	24	24	-	24	232
p_2 = 20, L = 10	24	24	24	24	24	24	24	24	24	256
p_2 = 10, L = 9	15	17	15	17	15	17	15	-	15.86	158
p_2 = 10, L = 10	15	17	15	17	15	17	15	17	16	173

Table 2. One-unit cyclic sequences that use cycle σ2 and filling-up phase f2.

	P_{22j}								P_{22}	MS_{22}
	j = 1	j = 2	j = 3	j = 4	j = 5	j = 6	j = 7	j = 8		
p_2 = 20, L = 9	30	24	24	24	24	24	24	-	24.85	237
p_2 = 20, L = 10	30	24	24	24	24	24	24	24	27.75	261
p_2 = 10, L = 9	20	15	17	15	17	15	17	-	16.57	160
p_2 = 10, L = 10	20	15	17	15	17	15	17	15	16.38	177

The following corollaries follow from Theorems 3, 4, and 5.

Corollary 1. For CT1-1-1, if $P_x \geq P_y$, then $MS_{x1} \geq MS_{y1}$ for all x, y = 1, 3, 4, and 5.

Proof. Directly from the values of MS_{11}, MS_{31}, MS_{41}, and MS_{51}.

Corollary 2. For CT1-1-1, MS_{x1} does not necessarily equal MS_{x2} for x = 2, 3, 4, 5, or 6.

Proof. Consider the following instance: L = 15, p_r = 13, p_1 = 18, p_2 = 14, and p_3 = 37. Table 3 presents the lot makespan for each of the eleven feasible 1-unit cyclic sequences. Note that, in this instance, MS_{22} is the lot makespan of the best 1-unit cyclic sequence.

Table 3. The eleven 1-unit cyclic sequences when L = 15, p_r = 13, p_1 = 18, p_2 = 14, p_3 = 37.

x	1	2	3	4	5	6
MS_{x1}	1815	1561	1647	1745	1913	1587
MS_{x2}		1556	1652	1757	1901	1593

Corollary 3. The shortest 1-unit cycle does not necessarily form an optimal sequence for CT1-1-1 | | C_{max}.

Proof. We will provide a counterexample. Let L = 10, p_r = 16, p_1 = 20, p_2 = 11, and p_3 = 18. Then σ2 is the shortest cycle, with $P_{21} = P_{22}$ = 112. The lot makespan of the 1-unit cyclic sequence that uses σ2 and filling-up phase f1 is MS_{21} = 1142. The lot makespan of the 1-unit cyclic sequence that uses σ2 and filling-up phase f2 is MS_{22} = 1141. Neither is optimal. The 1-unit cyclic sequence that uses σ1, whose cycle time P_1 = 113, has a lot makespan MS_{11} = 1130.

If the lot size L is large, then a shorter cycle will form a shorter cyclic sequence, since the lot makespan, as a function of L, has a slope equal to the cycle time (as shown in Theorems 3, 4, and 5). For instance, consider again the last counterexample. Table 4 shows MS_{11} and MS_{22} for different values of L. $MS_{11} = MS_{22}$ when L = 21. If L > 21, then MS_{22} is less than MS_{11}.

Table 4. MS_{11} and MS_{22} as functions of L.

L	19	20	21	22	23	24	25
MS_{11}	2147	2260	2373	2486	2599	2712	2825
MS_{22}	2149	2261	2373	2485	2597	2709	2821

Hall *et al.* [8] prove that, for a mobile-robot cell that produces a single part type, repeating a 1-unit cycle dominates more complicated 2-unit cycles. However, this result does not hold for CT1-1-1.

Theorem 6: The 1-unit cycles do not necessarily dominate 2-unit cycles for the problems CT1-1-1 | | C_t and CT1-1-1 | | C_{max}.

Proof. We will provide a counterexample. Let L = 8, p_r = 10, p_1 = 30, p_2 = 25, and p_3 = 5. The best 1-unit cycle is σ5, and the cycle time P_5 = 80. The best 1-unit cyclic sequence is formed by repeating σ5 and using filling-up phase f1 = $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1}$. The lot makespan MS_{51} = 660. However, there exists an optimal sequence that repeats the 2-unit cycle, $R_{3,j} – R_{0,j+2} – R_{2,j+1} – R_{3,j+1} – R_{1,j+2} – R_{0,j+3} – R_{2,j+2} – R_{1,j+3}$, which has an average cycle time of 77.5 time units. The optimal lot makespan is 645. Figure 3 presents the Gantt charts of this optimal sequence and the 1-unit cyclic sequence that uses σ5 and filling-up phase f1.

6. Branch-and-bound Algorithm

Although the cyclic sequences that use one-unit cycles are a natural class of solutions for the problem, they do not necessarily include an optimal sequence. Thus, we need to consider an algorithm that can generate an optimal sequence. This section presents a branch-and-bound algorithm, which is named Algorithm BB.

The algorithm begins by using the push dispatching rule to construct a feasible sequence and then using the pull dispatching rule to construct another feasible sequence. The smaller lot makespan becomes the initial upper bound on the optimal lot makespan. For each partial solution constructed, the algorithm creates a lower bound by calculating the completion time of the last scheduled activity.

4.1. Algorithm BB

Given S, p_r, p_1,...,p_S, and L, Algorithm BB proceeds as follows. Note that this algorithm calls Algorithm P to generate sequences using the push and pull dispatching rules. Section 4.2 describes Algorithm P.

Step 0.

Use the push dispatching rule (Algorithm P) to generate a feasible sequence. Use the pull dispatching rule (Algorithm P) to generate a second feasible sequence. Save the sequence with the smaller lot makespan as the current best sequence, and let the upper bound UB equal the lot makespan.

Step 1.

Initialize the cluster tool. All L unprocessed wafers are in LL, and the wafer handler is at LL. All of the chambers are free. The current sequence is a sequence with no moves. Set C = 0, n = L, and t_i = 0 for all stages S_i (i = 1, ..., S).

Step 2.

Based on the tool state, let t = C (the last move completion time) and identify all feasible moves. $R_{0,j}$ is feasible if n > 0 and j = L+1-n and S_1 is free. This can begin at time t. $R_{i,j}$ (0 < i < S) is feasible if wafer j is in S_i and S_{i+1} is free. This can begin at max{t, t_i}. $R_{S,j}$ is feasible if wafer j is in S_S. This can begin at max{t, t_S}. Use Theorem 7 to remove any dominated moves from consideration.

Step 3.

For each feasible move, form a new sequence with that move and perform one of the following steps (which calculates the lower bound LB and updates the tool state). Then go to Step 4.

- If the feasible move was $R_{0,j}$, then go to Step 3a.
- If the feasible move was $R_{i,j}$, 0 < i < S, then go to Step 3b.
- If the feasible move was $R_{S,j}$, then go to Step 3c.

Step 3a.

Reduce n by one. If the wafer handler was at LL, then the move completion time C = t + p_r. Otherwise, the move completion time C = t + 2p_r. The wafer handler is now at S_1, which now has wafer j, and t_1 = C + p_1. Let LB = t_1. Go to Step 3d.

Step 3b.

If the wafer handler was at S_i, then the move completion time C = max{t, t_i} + p_r. Otherwise, the move completion time C = max{t, t_i} +2p_r. S_i is now free. The wafer handler is now at S_{i+1}, which now has wafer j, and t_{i+1} = C + p_{i+1}. Let LB = t_{i+1}. Go to Step 3d.

Step 3c.

If the wafer handler was at S_S, then the move completion time $C = \max\{t, t_S\} + p_r$. Otherwise, the move completion time $C = \max\{t, t_S\} + 2p_r$. S_S is now free. LB = C. Go to Step 3d.

Step 3d.

If LB ≥ UB, then discard this new sequence. If this new sequence includes all L(S+1) moves, the lot makespan equals C. Consequently, if C < UB, save this new sequence as the current best sequence and set UB = C.

Step 4.

If any incomplete new sequences remain, select one, identify the corresponding tool state, and go to Step 2. Otherwise, stop and return the current best sequence and UB, its lot makespan.

For example, consider the following instance of CT1-1 || C_{max}: $p_r = 5$, $p_1 = 10$, $p_2 = 40$, and L = 3. Figure 4 shows a graph of all feasible sequences. Any path from the top node (in Level 1) to a node in a lower level corresponds to a feasible partial sequence. Branch A is the push sequence. The lot makespan is 185. Branch D is the pull sequence. The lot makespan is 195. The initial upper bound UB = 185. Algorithm BB will fully fathom Branch A and will prune the other three branches at Level 8 because their lower bounds are greater than or equal to 185.

4.2. Algorithm P

Given S, p_r, $p_1,...,p_S$, and L, Algorithm P proceeds as follows. Note that Algorithm P generates only non-delay schedules.

Step 1.

Initialize the cluster tool. All L unprocessed wafers are in LL, and the wafer handler is at LL. All of the chambers are free. $t_i = 0$ for all stages S_i (i = 1, ..., S). Set t = 0 and n = L.

Step 2.

Based on the tool state, identify any feasible moves that could begin at time t. $R_{0,j}$ can begin at time t if n > 0 and j = L+1-n, and S_1 is free. $R_{i,j}$ (0 < i < S) can begin at time t if wafer j is at S_i, $t_i \le t$, and S_{i+1} is free. $R_{S,j}$ can begin at time t if wafer j is at S_S and $t_S \le t$. If there is exactly one feasible move, then perform that move and go to Step 4. If there is more than one feasible move and the dispatching rule is push, select the feasible move $R_{i,j}$ with the smallest value of i and go to Step 4. If there is more than one feasible move and the dispatching rule is pull, select the feasible move $R_{i,j}$ with the largest value of i and go to Step 4. Otherwise, go to Step 3.

Step 3.

Let $t = \min\{ t_k : t_k > 0, k = 1, \ldots, S\}$. Go to Step 2.

Step 4.

Update the tool state.

- If the selected move was $R_{0,j}$, then reduce n by 1. If the wafer handler was at LL, then the move completion time $C = t + p_r$. Otherwise, the move completion time $C = t + 2p_r$. The wafer handler is now at S_1, which is now busy processing wafer j, and $t_1 = C + p_1$.
- If the selected move was $R_{i,j}$, 0 < i < S, and the wafer handler was at S_i, then the move completion time $C = t + p_r$. If the selected move was $R_{i,j}$, 0 < i < S, and the wafer handler was not at S_i, then the move completion time $C = t + 2p_r$. S_i is now free. $t_i = 0$. The wafer handler is now at S_{i+1}, which is now busy processing wafer j, and $t_{i+1} = C + p_{i+1}$.
- If the selected move was $R_{S,j}$ and the wafer handler was at S_S, then the move completion time $C = t + p_r$. If the selected move was $R_{S,j}$ and the wafer handler was not at S_S, then the move completion time $C = t + 2p_r$. S_S is now free. $t_S = 0$. The wafer handler is now at LL.

Step 5.

If all L(S+1) moves are complete, then stop. The lot makespan equals C. Otherwise, let t = C and go to Step 2.

4.3. Dominance criterion

Algorithm BB uses a dominance criterion to avoid unnecessary searching. Theorem 7 prohibits a move $R_{i,j}$ if there exists another move $R_{p,q}$ that can be done first without delaying the completion of $R_{i,j}$. Note that using this condition limits Algorithm BB to the set of active schedules.

Theorem 7: Given Q1, a feasible partial sequence for a sequential cluster tool, move $R_{p,q}$ dominates $R_{i,j}$ if both are feasible and the following conditions hold: The last move in Q1 ends at time t. The wafer handler is at chamber k after this move (k may be LL). $R_{i,j}$ can begin at time $t_a \geq t$ and wafer j is at chamber c_a, which is not chamber k. $R_{p,q}$ can begin at time $t_b \geq t$ and wafer q is at chamber c_q. Either $c_q = k$ and $t_b+p_r \leq t_a$ or c_q is not k and $t_b+2p_r \leq t_a$.

Proof. Consider a complete feasible permutation sequence Q that begins with Q1 and $R_{i,j}$. Since c_a is not k, $R_{i,j}$ requires $2p_r$ time units. Form a new sequence Q' by moving $R_{p,q}$ before $R_{i,j}$. Because $R_{p,q}$ remained feasible from the end of Q1 to its position in Q, Q' is also a feasible permutation sequence. If $c_q = k$ and $t_b+p_r \leq t_a$, the wafer handler can complete $R_{p,q}$ at t_b+p_r and still begin $R_{i,j}$ at t_a. Otherwise, c_q is not k and $t_b+2p_r \leq t_a$. Still, the wafer handler can complete $R_{p,q}$ at t_b+2p_r and still begin $R_{i,j}$ at t_a. Thus, no move must be delayed, and the lot makespan of Q' is not worse than the lot makespan of Q.

7. Experimental Results

We conducted experiments to determine the performance of Algorithm BB. Specifically, we wanted to know how much computational effort was required and whether the algorithm generated sequences that were much better than the cyclic sequences. In addition we wanted to determine how lot size and relative move time affect the performance of Algorithm BB. In effect we tested Algorithm BB against the unnamed heuristic that evaluates all feasible 1-unit cyclic sequences and selects the best one. (Note that evaluating the cyclic sequences requires little computational effort.) The push and pull dispatching rules were the benchmarks. (Note that Algorithm BB begins with these sequences).

We created 18 problem sets of instances. Each problem set had ten randomly generated instances. Each problem set used different parameter values to generate its instances. Table 5 lists the parameter values for each problem set. (Recall that S = 2 for CT1-1 and S = 3 for CT1-1-1.) Given S, L, and a, b, c, d, we used the following generation scheme to create the instances. Note that all data are integers, so X ~ U[a, b] implies that the random variable X has a discrete probability distribution and P{X = x} = 1/(b-a+1) for x = a, a+1, …, b.

$p_r \sim U[a, b]$.
$p_i \sim U[c, d]$, for i = 1, …, S.

The parameter values were chosen so that six problem sets (1, 2, 3, 10, 11, 12) contained instances with short move times and long processing times, another six problem sets (4, 5, 6, 13, 14, 15) contained instances with approximately equal move times and processing times, and the remaining six problem sets (7, 8, 9, 16, 17, 18) contained instances with a long move time and short processing times. We will refer to these three categories as short, equal, and long move times, respectively.

For each instance, we used Theorem 1 or Theorems 3, 4, and 5 to find the lot makespan of the best 1-unit cyclic sequence. We also used the branch-and-bound algorithm (Algorithm BB), the push dispatching rule (Algorithm P), and the pull dispatching rule (Algorithm P) to generate solutions. However, we did stop Algorithm BB if it reached 100,000 nodes and used the best current sequence at that point. Thus, for some instances, Algorithm BB may return a suboptimal sequence.

Tables 6 and 7 present the results for each problem set. The second column lists the average CPU time that Algorithm BB required. The third column shows the number of instances that Algorithm BB completely solved within 100,000 nodes. The next four columns show the average lot makespan of the best cyclic sequence and the other sequences that were generated. The last four columns show the average relative improvement from the push and pull sequences.

For the two-stage sequential cluster tools, Algorithm BB required little effort except when the lot size was large and the move times were approximately equal to the processing times. However, the best cyclic sequence was optimal for most of these two-stage sequential cluster tool instances. (In Problem Set 6 we cannot prove optimality.) As the move times increase (relative to the processing times), the optimal sequences were much better than those that the push and pull dispatching rules generated. Indeed, when the move times were long, the performance was dramatically better, as the relative improvement was approximately forty percent.

For the three-stage sequential cluster tools, Algorithm BB required little effort only when L = 5 (Problem Sets 10, 13, and 16). Otherwise, it was usually unable to complete the search in 100,000 nodes. In these cases, the best cyclic sequence was often better. For these three-stage sequential cluster tool instances, as the move times increase (relative to the processing times), the cyclic sequences were much better than those that the push and pull dispatching rules generated. Again, when the move times were long, the performance was dramatically better, as the relative improvement was approximately forty percent. Algorithm BB generated sequences that were also improvements, but they were not as good as the cyclic sequences, except when L = 5.

The computing effort for Algorithm BB increased as the lot size increased. Conducting longer searches or using more complicated lower bounds did not improve search performance significantly.

Table 5. Parameters for problem sets.

Problem Set	Parameters					
	S	L	a	b	c	d
1	2	5	1	10	20	40
2	2	10	1	10	20	40
3	2	15	1	10	20	40
4	2	5	10	20	10	20
5	2	10	10	20	10	20
6	2	15	10	20	10	20
7	2	5	20	40	1	10
8	2	10	20	40	1	10
9	2	15	20	40	1	10
10	3	5	1	10	20	40
11	3	10	1	10	20	40
12	3	15	1	10	20	40
13	3	5	10	20	10	20
14	3	10	10	20	10	20
15	3	15	10	20	10	20
16	3	5	20	40	1	10
17	3	10	20	40	1	10
18	3	15	20	40	1	10

Table 6. The performance of 1-unit cyclic sequences and Algorithm BB on instances of CT1-1.

Problem Set	CPU time (mm:ss)	Instances solved	Average lot makespan				% improvement over push		% improvement over pull	
			Cyclic	BB	Push	Pull	Cyclic	BB	Cyclic	BB
1	0:01	10	303.1	303.1	303.1	303.1	0	0	0	0
2	0:02	10	585.6	585.6	585.6	585.6	0	0	0	0
3	0:28	10	868.1	868.1	868.1	868.1	0	0	0	0
4	0:01	10	372.7	372.7	420.3	420.3	10.33	10.33	10.33	10.33
5	0:02	10	745.2	745.2	852.3	852.3	11.41	11.41	11.41	11.41
6	1:13	0	1117.7	1117.7	1284.3	1284.3	11.76	11.76	11.76	11.76
7	0:01	10	510.5	510.5	812.5	812.5	37.15	37.15	37.15	37.15
8	0:02	10	1021.0	1021.0	1700.5	1700.5	39.94	39.94	39.94	39.94
9	0:22	10	1531.5	1531.5	2588.5	2588.5	40.81	40.81	40.81	40.81

Table 7. The performance of 1-unit cyclic sequences and Algorithm BB on instances of CT1-1-1.

Problem Set	CPU time (mm:ss)	Instances solved	Average lot makespan				% improvement over push		% improvement over pull	
			Cyclic	BB	Push	Pull	Cyclic	BB	Cyclic	BB
10	0:01	10	369.0	369.0	369.0	371.1	0	0	0.44	0.44
11	1:04	5	677.5	677.5	677.5	683.1	0	0	0.62	0.62
12	1:44	0	986.0	986.0	986.0	995.1	0	0	0.68	0.68
13	0:03	10	521.7	521.7	591.8	591.8	11.10	11.10	11.21	11.21
14	1:37	0	1032.2	1083.0	1195.8	1195.8	13.00	9.07	13.00	9.07
15	2:10	0	1541.7	1687.0	1799.8	1799.8	13.68	6.03	13.71	6.06
16	0:01	10	670.0	670.0	1095.4	1071.6	38.63	38.63	37.29	37.29
17	1:28	0	1340.0	1585.6	2267.4	2267.4	40.70	29.84	40.70	29.84
18	1:54	0	2010.0	2757.6	3439.4	3415.6	41.35	19.67	40.95	19.12

8. Summary and Conclusions

This paper studied the sequential cluster tool scheduling problem. The goal is to improve tool performance by reducing the total lot processing time (which we call the lot makespan). This paper enumerated and analyzed the class of one-unit cyclic sequences. In addition, it presented a branch-and-bound algorithm that can find optimal solutions. The paper describes the results of experiments performed to understand the performance of the cyclic sequences and the branch-and-bound algorithm.

Our results show that, for two- and three-stage sequential cluster tools, identifying the best cyclic sequence can, in some cases, dramatically reduce the lot makespan and thus improve cluster tool performance. The practice of using a push dispatching rule or a pull dispatching rule yields inferior performance. This is especially true when the move time and processing times are approximately equal and when the move time is longer than processing times. Compared to the cyclic sequences, the branch-and-bound algorithm requires additional computational effort and does not yield better solutions on the instances considered. The computational effort is sensitive to the problem size (the number of stages and wafers).

We have also studied hybrid cluster tools that have at least one stage with two or more chambers. Those results have been reported separately [13]. Future work should consider more complex tool configurations and scheduling anticipatory moves that position the wafer handler at the next chamber before the chamber finishes processing the wafer. Such anticipatory moves should further improve the cluster tool performance.

Acknowledgements

This research was supported by the Semiconductor Research Corporation and the National Science Foundation under grant DMI 97−13720. We would like to thank Professor Gary W. Rubloff, Mr. Brian F. Conaghan, Mr. Praveen Mellacheruvu, and Mr. Rock Z. Shi of the Institute for Systems Research, University of Maryland, for their useful insights. The research was performed using the facilities of the Computer-Integrated Manufacturing Laboratory at the University of Maryland.

Bibliography

1. Semiconductor Business News. "Applied Materials, Novellus, LAM Research lead cluster tool market" CMP Media Inc. 26 March 1998. 14 Jan. 2000. <http://www.semibiznews.com/stories/8c26tools.htm>

2. Wood, Samuel C, “Simple performance models for integrated processing tools,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, n. 3, pp. 320-328, 1996.
3. Perkinson, Terry L., Peter K. McLary, Ronald S. Gyucsik, and Ralph K. Cavin, “Single-wafer cluster tool performance: an analysis of throughput,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, n. 3, pp. 369-373, 1994.
4. Venkatesh, Srilakshmi, Rob Davenport, Pattie Foxhoven, and Jaim Nulman, “A steady-state throughput analysis of cluster tools: dual-blade versus single-blade robots,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 10, n. 4, pp. 418-424, 1997.
5. Srinivasan, R. S, “Modeling and performance analysis of cluster tools using Petri nets,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 11, n. 3, pp. 394-403, 1998.
6. Herrmann, Jeffrey W., Niranjan Chandrasekaran, Brian F. Conaghan, Manh-Quan T. Nguyen, Gary W. Rubloff, and Rock Z. Shi, “Evaluating the impact of process changes on cluster tool performance,” *IEEE Transactions on Semiconductor Manufacturing*. Volume 13, Number 2, pages 181-192, 2000.
7. Jeng, Wu-De, James T. Lin, and Ue-Pyng Wen, “Algorithms for sequencing robot activities in a robot-centered parallel-processor workcell”, *Computers Ops. Res*. v20, n2, pp. 185-197, 1993
8. Hall, Nicholas G., Hichem Kamoun, and Chelliah Sriskandarajah, “Scheduling in robotic cells: classification, two and three machine cells,” *Operations Research*, vol. 45, n.3, pp. 421-439, 1997.
9. Crama, Yves and Joris van de Klundert, “Cyclic scheduling of identical parts in a robotic cell”, Operations Research, vol. 45, n. 6, Nov. – Dec. 1997.
10. Murphy, Robert, Puneet Saxena, William Levinson, “Use OEE; don’t let OEE use you,” *Semiconductor International*, pp. 125-132, Sep. 1996.
11. Dance, Daren L., Devid W. Jimenez, and Alan L. Levine, “Understanding equipment cost-of-ownership,” *Semiconductor International*, pp. 117-122, July 1998.
12. Sethi, S. P., C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak, “Sequencing of parts and robot moves in a robotic cell,” *Int. J. Production Res.*, vol. 4, pp. 331-358, 1992.
13. Nguyen, Manh-Quan T., and Jeffrey W. Herrmann, “Sequencing wafer handler moves to improve the performance of hybrid cluster tools,” Technical Report, Institute for Systems Research, University of Maryland, 2000.

Appendix A. Proof of Theorem 1.

We will analyze the cycle time and lot makespan for each 1-unit cyclic sequence.

Cycle σ1 is $R_{2, j-1} – R_{0,j} – R_{1,j} – R_{2,j}$. Figure A1 presents a Gantt chart of a complete sequence when L = 3. For convenience, we will consider the cycle $R_{0, j} – R_{1, j} – R_{2, j}$. The complete sequence has L such cycles. For i = 0, 1, 2, and for all j = 1, …, L, the length of $R_{i,j}$ is p_r. Thus, the cycle time $P_1 = 3p_r + p_1 + p_2$, and the lot makespan $MS_1 = LP_1$.

Cycle σ2 is $R_{2, j-1} – R_{1, j} – R_{0,j+1} – R_{2,j}$. Figure A2 presents a Gantt chart of a complete sequence when L = 4. For convenience, we will consider the cycle $R_{0, j+1} – R_{2, j} –R_{1, j+1}$. The complete sequence has L - 1 such cycles (j = 1, …, L-1). The sequence begins at time 0. The filling-up phase has two moves, $R_{0, 1}$ and $R_{1, 1}$, and ends at $t_1 = 2p_r + p_1$.

Let y be the idle time between $R_{0, j+1}$ and $R_{2, j}$. $y = \max\{p_2, 2p_r\} - 2p_r$. Let x be the idle time between $R_{2, j}$ and $R_{1, j+1}$. $x = \max\{p_1 - 2p_r - y, 0\} = \max\{p_1, 2p_r + y\} - 2p_r - y$. Thus, the cycle time

$$P_2 = 6p_r + x + y = 4p_r + \max\{p_1, 2p_r + y\} = 4p_r + \max\{2p_r, p_1, p_2\}.$$

The completion phase, which has one move $R_{2, L}$, starts at $t_2 = t_1 + (L - 1)P_2$ and ends at $t_2 + p_2 + p_r$. Thus, the lot makespan

$$MS_2 = 2p_r + p_1 + (L-1)P_2 + p_2 + p_r = 3p_r + p_1 + p_2 + (L-1)P_2.$$

Appendix B. Proof of Theorem 3.

We will analyze the cycle time and lot makespan for each of the seven 1-unit cyclic sequences.

Cycle σ1 is $R_{3,j-1}$– $R_{0,j}$ – $R_{1,j}$ – $R_{2,j}$ – $R_{3,j}$. Figure B1 presents a Gantt chart of a complete sequence when L = 3.

For the purposes of this proof, we will consider the cycle $R_{0,j}$ – $R_{1,j}$ – $R_{2,j}$ – $R_{3,j}$. The complete sequence consists of L such cycles. The length of $R_{i,j}$ is p_r for i = 0, 1, 2, 3, and for all j = 1, …, L. Thus, the cycle time $P_1 = 4p_r + p_1 + p_2 + p_3$, and the lot makespan $MS_{11} = LP_1$.

Cycle σ3 is $R_{3,j-1}$– $R_{2,j}$ – $R_{0,j+1}$ – $R_{1,j+1}$ – $R_{3,j}$. There are two feasible filling-up phases, which we denote f1 and f2.

Filling-up phase f1 is $R_{0,1}$ – $R_{1,1}$ – $R_{2,1}$ – $R_{0,2}$ – $R_{1,2}$. Figure B2 presents a Gantt chart of a complete sequence with this filling-up phase when L = 4. We will consider the cycle $R_{2,j}$– $R_{0,j+1}$ – $R_{1,j+1}$ – $R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$.

Let P_3 be the length of the interval between the start of $R_{2,j}$ and $R_{2,j+1}$, for j = 2, …, L-1. Note that the length of the interval between the start of $R_{2,1}$ and $R_{2,2}$ is $P_3 – p_r$. Thus, the first cycle starts at $t_1 = 2p_r + p_1 + p_2 + P_3 – p_r = p_r + p_2 + p_1 + P_3$.

In the cyclic phase, the length of $R_{1,j}$ is p_r, and the length of any other move is $2p_r$. Let t be the time that move $R_{2,j}$ starts (j = 2,…, L-1). The start and end of each move can be calculated as follows:

Move	Start	End
$R_{2,j}$	t	$t+2p_r$
$R_{0,j+1}$	$t+2p_r$	$t+4p_r$
$R_{1,j+1}$	$t+4p_r+p_1$	$t+5p_r+p_1$
$R_{3,j}$	$\max\{ t+5p_r+p_1, t+2p_r+p_3\}$	$t+4p_r+\max\{ 3p_r+p_1, p_3\}$
$R_{2,j+1}$	$\max\{ t+4p_r+\max\{ 3p_r+p_1, p_3\}, t+5p_r+p_1+p_2\}$ $= t+4p_r +\max\{ \max\{ 3p_r+p_1, p_3\}, p_r+p_1+p_2\}$ $= t+4p_r +\max\{3p_r+p_1, p_3, p_r+p_1+p_2\}$	$t+6p_r +\max\{ 3p_r+p_1, p_3, p_r+p_1+p_2\}$

Thus, the cycle time $P_3 = 4p_r +\max\{3p_r+p_1, p_3, p_r+p_1+p_2\}$. The completion phase starts at time $t_2 = t_1 + (L-2)P_3 = p_r + p_1 + p_2 + (L-1)P_3$ and has two moves, $R_{2,L}$ and $R_{3,L}$. The length of the completion phase is $2p_r+p_3+p_r$. Thus, the lot makespan $MS_{31} = p_r+p_1+p_2+(L-1)P_3 + 2p_r+p_3+p_r = 4p_r+p_1+p_2+p_3+(L-1)P_3$.

Filling-up phase f2 is $R_{0,1}$ – $R_{1,1}$ – $R_{0,2}$ – $R_{2,1}$ – $R_{1,2}$. Figure B3 presents the Gantt chart of a complete with this filling-up phase when L = 4. We will consider the cycle $R_{2,j}$– $R_{0,j+1}$ – $R_{1,j+1}$ – $R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$.

The start and end times of each move in the filling-up phase can be calculated as follows:

Move	Start	End
$R_{0,1}$	0	p_r
$R_{1,1}$	p_r+p_1	$2p_r+p_1$
$R_{0,2}$	$2p_r+p_1$	$4p_r+p_1$
$R_{2,1}$	$2p_r+p_1+\max\{p_2, 2p_r\}$	$4p_r+p_1+\max\{p_2, 2p_r\}$
$R_{1,2}$	$\max\{4p_r+2p_1, 4p_r+p_1+\max(p_2, 2p_r)\}$ $=4p_r+p_1+\max\{p_1, p_2, 2p_r\}$	$6p_r+p_1+\max\{p_1, p_2, 2p_r\}$
$R_{3,1}$	$\max\{4p_r+p_1+p_3+\max(p_2, 2p_r), 6p_r+p_1+\max(p_1, p_2, 2p_r)\}$ $=4p_r+p_1+\max\{p_3+\max(p_2, 2p_r), 2p_r+\max(p_1, p_2, 2p_r)\}$	$6p_r+p_1+\max\{p_3+\max(p_2, 2p_r), 2p_r+\max(p_1, p_2, 2p_r)\}$

Let t_1 be the time that the first cycle starts (the time that move $R_{2,2}$ starts).

$$\begin{aligned} t_1 &= \max\{6p_r + p_1 + p_2 + \max(p_1, p_2, 2p_r), 6p_r + p_1 + \max[p_3 + \max(p_2, 2p_r), 2p_r + \max(p_1, p_2, 2p_r)]\} \\ &= 6p_r + p_1 + \max\{p_2 + \max(p_1, p_2, 2p_r), p_3 + \max(p_2, 2p_r), 2p_r + \max(p_1, p_2, 2p_r)\} \\ &= 6p_r + p_1 + \max\{\max(p_2, 2p_r) + \max(p_1, p_2, 2p_r), p_3 + \max(p_2, 2p_r)\} \\ &= 6p_r + p_1+ \max\{p_2, 2p_r\} + \max\{p_1, p_2, p_3, 2p_r\} \end{aligned}$$

The steady-state and completion phases are the same as those for the first filling-up phase. Hence, the cycle time $P_3 = 4p_r + \max\{3p_r + p_1, p_3, p_r + p_1 + p_2\}$. The lot makespan

$$\begin{aligned} MS_{32} &= t_1 + (L\text{-}2)P_3 + [2p_r + p_3 + p_r] \\ &= 6p_r + p_1 + \max\{p_2, 2p_r\} + \max\{p_1, p_2, p_3, 2p_r\} + (L\text{-}2)P_3 + [2p_r + p_3 + p_r] \\ &= 9p_r + p_1 + p_3 + \max\{p_2, 2p_r\} + \max\{p_1, p_2, p_3, 2p_r\} + (L\text{-}2)P_3. \end{aligned}$$

Cycle σ4 is $R_{3,j-1} – R_{1,j} – R_{2,j} – R_{0,j+1} – R_{3,j}$. There are two feasible filling-up phases, which we denote f1 and f2.

Filling-up phase f1 is $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2}$. (See Figure B4.) We will consider the cycle $R_{1,j} – R_{2,j} – R_{0,j+1} – R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$. Let P_4 be the length of the interval between the start of $R_{1,j}$ and $R_{1,j+1}$, for j = 2, …, L-1. Note that the length of the interval between the start of $R_{1,1}$ and $R_{1,2}$ is $P_4 – p_r$. Thus, the first cycle starts at $t_1 = p_r + p_1 + P_4 – p_r = p_1 + P_4$.

In the cyclic phase, the length of $R_{2,j}$ is p_r, and the length of any other move is $2p_r$. Let t be the time that move $R_{1,j}$ starts (j = 2,…, L-1). The start and end of each move can be calculated as follows:

Move	Start	End
$R_{1,j}$	t	$t+2p_r$
$R_{2,j}$	$t+2p_r+p_2$	$t+3p_r+p_2$
$R_{0,j+1}$	$t+3p_r+p_2$	$t+5p_r+p_2$
$R_{3,j}$	$t+3p_r+p_2+\max\{2p_r, p_3\}$	$t+5p_r+p_2+\max\{2p_r, p_3\}$
$R_{1,j+1}$	$\max\{t+5p_r+p_2+\max\{2p_r, p_3\}, t+5p_r+p_2+p_1\}$ $= t+5p_r+p_2+\max\{2p_r, p_3, p_1\}$	$t+7p_r+p_2+\max\{2p_r, p_3, p_1\}$

Thus, the cycle time $P_4 = 5p_r + p_2 + \max\{2p_r, p_3, p_1\}$. The completion phase starts at time $t_2 = t_1 + (L - 2)P_4 = p_1 + (L - 1)P_4$. The completion phase has three moves, $R_{1,L}$, $R_{2,L}$, and $R_{3,L}$. The length of the completion phase is $2p_r + p_2 + p_r + p_3 + p_r$. Thus, the lot makespan $MS_{41} = [p_1 + (L - 1)P_4] + [2p_r + p_2 + p_r + p_3 + p_r] = 4p_r + p_1 + p_2 + p_3 + (L - 1)P_4$.

Filling-up phase f2 is $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1}$. See Figure B5. We will consider the cycle $R_{1,j} – R_{2,j} – R_{0,j+1} – R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$. The start and end times of each move in the filling-up phase can be calculated as follows:

Move	Start	End
$R_{0,1}$	0	p_r
$R_{1,1}$	p_r+p_1	$2p_r+p_1$
$R_{0,2}$	$2p_r+p_1$	$4p_r+p_1$
$R_{2,1}$	$2p_r+p_1+\max\{p_2, 2p_r\}$	$4p_r+p_1+\max\{p_2, 2p_r\}$
$R_{3,1}$	$4p_r+p_1+p_3+\max\{p_2, 2p_r\}$	$5p_r+p_1+p_3+\max\{p_2, 2p_r\}$

Let t_1 be the time that the first cycle starts (the time that move $R_{1,2}$ starts).

$$\begin{aligned} t_1 &= \max\{4p_r + 2p_1, 5p_r + p_1 + p_3 + \max(p_2, 2p_r)\} \\ &= 4p_r + p_1 + \max\{p_1, p_r+p_3+\max(p_2, 2p_r)\}. \end{aligned}$$

The steady-state and completion phases are the same as those for the first filling-up phase. Hence, the cycle time $P_4 = 5p_r + p_2 + \max\{2p_r, p_3, p_1\}$. The lot makespan

$$\begin{aligned} MS_{42} &= t_1 + (L\text{-}2)P_4 + 2p_r + p_2 + p_r + p_3 + p_r \\ &= [4p_r + p_1 + \max\{p_1, p_r+p_3 + \max(p_2, 2p_r)\} + (L - 2)P_4] + [2p_r + p_2 + p_r + p_3 + p_r] \\ &= 8p_r + p_1 + p_2 + p_3 + \max\{p_1, p_r+p_3 + \max(p_2, 2p_r)\} + (L - 2)P_4. \end{aligned}$$

Cycle $\sigma 5$ is $R_{3,j-1} – R_{1,j} – R_{0,j+1} – R_{2,j} – R_{3,j}$. There are two feasible filling-up phases, which we denote f1 and f2.

Filling-up phase f1 is $R_{0,1} – R_{1,1} – R_{0,2} – R_{2,1}$. See Figure B6. We will consider the cycle $R_{1,j} – R_{0,j+1} – R_{2,j} – R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$. Let P_5 be the length of the interval between the start of $R_{1,j}$ and $R_{1,j+1}$, for j = 2, …, L-1. Note that the length of the interval between the start of $R_{1,1}$ and $R_{1,2}$ is $P_5 – p_r$. Thus, the first cycle starts at $t_1 = p_r + p_1 + P_5 – p_r = p_1 + P_5$.

In the cyclic phase, the length of $R_{3,j}$ is p_r, and the length of any other move is $2p_r$. Let t be the time that move $R_{1,j}$ starts (j = 2,…, L-1). The start and end of each move can be calculated as follows:

Move	Start	End
$R_{1,j}$	t	$t+2p_r$
$R_{0,j+1}$	$t+2p_r$	$t+4p_r$
$R_{2,j}$	$t+2p_r+\max\{p_2, 2p_r\}$	$t+4p_r+\max\{p_2, 2p_r\}$
$R_{3,j}$	$t+4p_r+\max\{p_2, 2p_r\}+p_3$	$t+5p_r+\max\{p_2, 2p_r\}+p_3$
$R_{1,j+1}$	$\max\{ t+5p_r+\max\{p_2, 2p_r\}+p_3, t+4p_r +p_1\}$ $= t+4p_r+\max\{p_r+p_2+p_3, 3p_r+p_3, p_1\}$	$t+6p_r+\max\{p_r+p_2+p_3, 3p_r+p_3, p_1\}$

Thus, the cycle time $P_5 = 4p_r + \max\{p_r + p_2 + p_3, 3p_r + p_3, p_1\}$. The completion phase starts at time $t_2 = t_1 + (L - 2)P_5 = p_1 + (L - 1)P_5$. The completion phase has three moves, $R_{1,L}$, $R_{2,L}$, and $R_{3,L}$. The length of the completion phase is $2p_r + p_2 + p_r + p_3 + p_r$. Thus, the lot makespan $MS_{51} = [p_1 + (L - 1)P_5] + [2p_r + p_2 + p_r + p_3 + p_r] = 4p_r + p_1 + p_2 + p_3 + (L - 1)P_5$.

Filling-up phase f2 is $R_{0,1} – R_{1,1} – R_{2,1} – R_{0,2}$. See Figure B7. We will consider the cycle $R_{1,j} – R_{0,j+1} – R_{2,j} – R_{3,j}$, which repeats L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$. The start and end times of each move in the filling-up phase can be calculated as follows:

Move	Start	End
$R_{0,1}$	0	p_r
$R_{1,1}$	p_r+p_1	$2p_r+p_1$
$R_{2,1}$	$2p_r+p_1+p_2$	$3p_r+p_1+p_2$
$R_{0,2}$	$3p_r+p_1+p_2$	$5p_r+p_1+p_2$
$R_{3,1}$	$\max\{3p_r+p_1+p_2+p_3, 5p_r+p_1+p_2\}$	$5p_r+p_1+p_2+ \max\{p_3, 2p_r\}$

Let t_1 be the time that the first cycle starts (the time that move $R_{1,2}$ starts).

$$
\begin{aligned}
t_1 &= \max\{5p_r + 2p_1 + p_2, 5p_r + p_1 + p_2 + \max(p_3, 2p_r)\} \\
&= 5p_r + p_1 + p_2 + \max\{p_1, p_3, 2p_r\}.
\end{aligned}
$$

The steady-state and completion phases are the same as those for the first filling-up phase. Hence, the cycle time $P_5 = 4p_r + \max\{p_r + p_2 + p_3, 3p_r + p_3, p_1\}$. The lot makespan

$$
\begin{aligned}
MS_{52} &= t_1 + (L-2)P_5 + 2p_r + p_2 + p_r + p_3 + p_r \\
&= [5p_r + p_1 + p_2 + \max\{p_1, p_3, 2p_r\} + (L - 2)P_5] + [2p_r + p_2 + p_r + p_3 + p_r] \\
&= 9p_r + p_1 + 2p_2 + p_3 + \max\{p_1, p_3, 2p_r\} + (L - 2)P_5.
\end{aligned}
$$

Appendix C. Proof of Theorem 4.

We will analyze the cycle time and lot makespan for each of the two 1-unit cyclic sequences.

Cycle σ2 is $R_{3,j-1}$– $R_{0,j+1}$ – $R_{2,j}$ – $R_{1,j+1}$ – $R_{3,j}$. There are two feasible filling-up phases, which we denote f1 and f2.

Filling-up phase f1 is $R_{0,1}$ – $R_{1,1}$ – $R_{0,2}$ – $R_{2,1}$ – $R_{1,2}$. See Figure C1. We will consider the cycle $R_{0,j+1}$ – $R_{2,j}$ – $R_{1,j+1}$ – $R_{3,j}$, which occurs L-2 times (j = 2, …, L-1). Thus, we add to the filling-up phase the move $R_{3,1}$.

For j = 1, …, L-1, define a_j, b_j, and c_j as follows. Let a_j be the idle time between $R_{2,j}$ and $R_{1,j+1}$. Let b_j be the idle time between $R_{0,j+1}$ and $R_{2,j}$. Let c_j be the idle time between $R_{1,j+1}$ and $R_{3,j}$.

$$b_1 = \max(p_2 - 2p_r, 0) = \max(p_2, 2p_r) - 2p_r,$$
$$a_1 = \max(p_1 - 2p_r - b_1, 0) = \max\{p_1 - \max(p_2, 2p_r), 0\}, \text{ and}$$
$$c_1 = \max(p_3 - 2p_r - a_1, 0).$$

The filling-up phase ends and the first cycle starts at $t_1 = 10p_r + p_1 + a_1 + b_1 + c_1$.

Let P_{21j} be the length of the interval between the start of $R_{0,j}$ and the end of $R_{3,j}$, for j = 2, …, L-1. The length of $R_{0,j}$ is p_r, and the length of any other move is $2p_r$. In these cycles,

$$b_j = \max(p_2 - 3p_r - c_{j-1}, 0),$$
$$a_j = \max(p_1 - 2p_r - b_j, 0), \text{ and}$$
$$c_j = \max(p_3 - 2p_r - a_j, 0).$$

Thus $P_{21j} = 7p_r + a_j + b_j + c_j$. The average cycle time is

$$P_{21} = \frac{1}{L-2}\sum_{j=2}^{L-1} P_{21j}.$$

The last cycle ends at $t_2 = t_1 + (L - 2)P_{21} = 10p_r + p_1 + a_1 + b_1 + c_1 + (L-2) P_{21}$. The completion phase has two moves, $R_{2,L}$ and $R_{3,L}$. The first move can start at time $t_2 + \max(p_2 - 2p_r - c_{L-1}, 0)$. Thus, the lot makespan

$$\begin{aligned} MS_{21} &= 10p_r + p_1 + a_1 + b_1 + c_1 + (L-2) P_{21} + \max(p_2 - 2p_r - c_{L-1}, 0) + 3p_r + p_3 \\ &= 13p_r + p_1 + p_3 + a_1 + b_1 + c_1 + (L-2) P_{21} + \max(p_2 - 2p_r - c_{L-1}, 0). \end{aligned}$$

Filling-up phase f2 is $R_{0,1}$ – $R_{1,1}$ – $R_{2,1}$ – $R_{0,2}$ – $R_{1,2}$. See Figure C2. We will consider the cycle $R_{3,j}$ – $R_{0,j+2}$ – $R_{2,j+1}$ – $R_{1,j+2}$, which occurs L-2 times (j = 1, …, L-2). The first cycle starts at time $t_1 = 3p_r + p_1 + p_2 + \max\{p_3, 3p_r + p_1\}$.

For j = 1, …, L-2, define d_j, e_j, and g_j as follows. Let d_j be the idle time between $R_{2,j+1}$ and $R_{1,j+2}$. Let e_j be the idle time between $R_{0,j+2}$ and $R_{2,j+1}$. Let g_j be the idle time between $R_{1,j+2}$ and $R_{3,j+1}$.

$$e_1 = \max\{p_2 - 3p_r - \max(p_3 - p_1 - 3p_r, 0), 0\} = \max\{p_1 + p_2 - \max(p_3, p_1 + 3p_r), 0\},$$
$$d_1 = \max\{p_1 - 2p_r - e_1, 0\}, \text{ and}$$
$$g_1 = \max\{p_3 - 2p_r - d_1, 0\}.$$

For j = 2, …, L-2,

$$e_j = \max\{p_2 - 3p_r - g_{j-1}, 0\},$$
$$d_j = \max\{p_1 - 2p_r - e_j, 0\}, \text{ and}$$
$$g_j = \max\{p_3 - 2p_r - d_j, 0\}.$$

Let P_{22j} be the length of the interval between the start of $R_{3,\,j}$ and the start of $R_{3,j+1}$, for j = 1, …, L-2. The length of $R_{0,j}$ is p_r, and the length of any other move is $2p_r$. $P_{22j} = 7p_r + d_j + e_j + g_j$. The average cycle time is

$$P_{22} = \frac{1}{L-2}\sum_{j=1}^{L-2} P_{22j}.$$

The completion phase has three moves and starts at $t_2 = t_1 + (L-2)P_{22} = 3p_r + p_1 + p_2 + \max\{p_3, 3p_r + p_1\} + (L-2)P_{22}$. The start and end time of each move can be calculated as follows.

Move	Start	End
$R_{3,\,L\text{-}1}$	t_2	$t_2 + 2p_r$
$R_{2,\,L}$	$t_2 + 2p_r + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\}$	$t_2 + 4p_r + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\}$
$R_{3,\,L}$	$t_2 + 4p_r + p_3 + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\}$	$t_2 + 5p_r + p_3 + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\}$

Therefore the lot makespan

$$\begin{aligned} MS_{22} &= 3p_r + p_1 + p_2 + \max\{p_3, 3p_r + p_1\} + (L-2)P_{22} + 5p_r + p_3 + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\} \\ &= 8p_r + p_1 + p_2 + p_3 + \max\{p_3, 3p_r + p_1\} + \max\{p_2 - 2p_r - g_{L\text{-}2}, 0\} + (L-2)P_{22}. \end{aligned}$$

Appendix D. Proof of Theorem 5.

Cycle $\sigma 6$ is $R_{3,j-1} - R_{2,j} - R_{1,j+1} - R_{0,j+2} - R_{3,j}$. There are two feasible filling-up phases, which we denote f1 and f2.

Filling-up phase f1 is $R_{0,1} - R_{1,1} - R_{0,2} - R_{2,1} - R_{1,2} - R_{0,3}$. See Figure D1. We will consider the cycle $R_{2,j} - R_{1,j+1} - R_{0,j+2} - R_{3,j}$, which occurs L-3 times (j = 2, …, L-2). Thus we add the move $R_{3,1}$ to the filling-up phase. The start and end time of each move in this phase can be calculated as follows:

Move	Start	End
$R_{0,1}$	0	p_r
$R_{1,1}$	p_r+p_1	$2p_r+p_1$
$R_{0,2}$	$2p_r+p_1$	$4p_r+p_1$
$R_{2,1}$	$\max\{4p_r+p_1, 2p_r+p_1+p_2\}=2p_r+p_1+\max\{2p_r, p_2\}$	$4p_r+p_1+\max\{2p_r, p_2\}$
$R_{1,2}$	$\max\{4p_r+p_1+\max\{2p_r, p_2\}, 4p_r+2p_1\}$ $= 4p_r+p_1+\max\{2p_r, p_1, p_2\}$	$6p_r+p_1+\max\{2p_r, p_1, p_2\}$
$R_{0,3}$	$6p_r+p_1+\max\{2p_r, p_1, p_2\}$	$8p_r+p_1+\max\{2p_r, p_1, p_2\}$
$R_{3,1}$	$\max\{8p_r+p_1+\max(2p_r, p_1, p_2), 4p_r+p_1+\max(2p_r, p_2)+p_3\}$ $= 4p_r+p_1+\max\{4p_r+\max(2p_r, p_1, p_2), \max(2p_r, p_2)+p_3\}$	$6p_r+p_1+\max\{4p_r+\max(2p_r, p_1, p_2), \max(2p_r, p_2)+p_3\}$

The first cycle starts at

$$t_1 = \max\{6p_r + p_1 + \max[4p_r + \max(2p_r, p_1, p_2), \max(2p_r, p_2) + p_3], p_2 + 6p_r + p_1 + \max\{2p_r, p_1, p_2\}\}$$
$$= 6p_r + p_1 + \max(2p_r, p_1, p_2) + \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\}.$$

w_2 is the idle time between $R_{2,2}$ and $R_{1,3}$.

$$w_2 = \max\{p_1 + 8p_r + p_1 + \max(2p_r, p_1, p_2) - 8p_r - p_1 - \max(2p_r, p_1, p_2) - \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\}, 0\}$$
$$= \max\{p_1 - \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\}, 0\}.$$

In general, for j = 2, …, L-2, w_j is the idle time between $R_{2,j}$ and $R_{1,j+1}$, and P_{61j} is the interval from the start of $R_{2,j}$ to the start of $R_{2,j+1}$. In the cyclic phase, all moves take $2p_r$. Let t be the starting time of $R_{2,j}$ (j = 2, …, L-2). The start and end time of each move in the cycle can be calculated as follows:

Move	start	complete
$R_{2,j}$	t	$t+2p_r$
$R_{1,j+1}$	$t+2p_r+w_j$	$t+4p_r+w_j$
$R_{0,j+2}$	$t+4p_r+w_j$	$t+6p_r+w_j$
$R_{3,j}$	$\max\{t+2p_r+p_3, t+6p_r+w_j\}$ $= t+2p_r+\max\{p_3, 4p_r+w_j\}$	$t+4p_r+\max\{p_3, 4p_r+w_j\}$
$R_{2,j+1}$	$\max\{t+4p_r+w_j+p_2, t+4p_r+\max[p_3, 4p_r+w_j]\}$ $= t+4p_r+\max\{p_2+w_j, \max[p_3, 4p_r+w_j]\}$ $= t+4p_r+w_j+\max\{p_2, p_3-w_j, 4p_r\}$	$t+6p_r+w_j+\max\{p_2, p_3-w_j, 4p_r\}$

Thus, $P_{61j} = 4p_r+ w_j +\max\{p_2, p_3-w_j, 4p_r\}$, $w_{j+1} = \max\{p_1-\max(p_2, p_3 - w_j, 4p_r), 0\}$. The average cycle time

$$P_{61} = \frac{1}{L-3}\sum_{j=2}^{L-2} P_{61j}.$$

The completion phase starts at time $t_2 = t_1 + (L-3)P_{61}$. The completion phase is $R_{2,L-1}$- $R_{1,L}$ - $R_{3,L-1}$ - $R_{2,L}$ - $R_{3,L}$. The start and end time of each move can be calculated as follows.

Move	Start	Complete
$R_{2,L-1}$	t_2	t_2+2p_r
$R_{1,L}$	$t_2+2p_r+ w_{L-2}$	$t_2+4p_r+w_{L-2}$
$R_{3,L-1}$	$\max\{t_2+4p_r+ w_{L-2}, t_2+2p_r+p_3\}$ $= t_2+2p_r+\max\{2p_r+ w_{L-2}, p_3\}$	$t_2+4p_r+\max\{2p_r+ w_{L-2}, p_3\}$
$R_{2,L}$	$\max\{t_2+4p_r+\max[2p_r+ w_{L-2}, p_3], t_2 + 4p_r+ w_{L-2} + p_2\}$ $= t_2+4p_r+ w_{L-2} +\max\{2p_r, p_2, p_3- w_{L-2}\}$	$t_2+6p_r+ w_1 +\max\{2p_r, p_2, p_3-w_{L-2}\}$
$R_{3,L}$	$t_2+6p_r+p_3+ w_{L-2}+\max\{2p_r, p_2, p_3 - w_{L-2}\}$	$t_2+7p_r+p_3+w_{L-2}+\max\{2p_r, p_2, p_3- w_{L-2}\}$

Thus, the lot makespan

$$\begin{aligned} MS_{61} &= 6p_r + p_1 + \max(2p_r, p_1, p_2) + \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\} + (L-3)P_{61} + \\ &\quad + 7p_r + p_3 + w_{L-2} + \max\{2p_r, p_2, p_3- w_{L-2}\} \\ &= 13p_r + p_1 + p_3 + w_{L-2} + \max(2p_r, p_1, p_2) + \max\{4p_r, p_2, p_3 - \max[0, p_1 - \max(2p_r, p_2)]\} + \\ &\quad + \max\{2p_r, p_2, p_3- w_{L-2}\} + (L-3)P_{61}. \end{aligned}$$

Filling-up phase f2 is $R_{0,1} - R_{1,1} - R_{2,1} - R_{0,2} - R_{1,2} - R_{0,3}$. See Figure D2. We will consider the cycle $R_{2,j} - R_{1,j+1} - R_{0,j+2} - R_{3,j}$, which repeats L-3 times (j = 2, …, L-2). Thus we add the move $R_{3,1}$ to the filling-up phase. The start and end time of each move in this phase can be calculated as follows:

Move	Start	End
$R_{0,1}$	0	p_r
$R_{1,1}$	p_r+p_1	$2p_r+p_1$
$R_{2,1}$	$2p_r+p_1+p_2$	$3p_r+p_1+p_2$
$R_{0,2}$	$3p_r+p_1+p_2$	$5p_r+p_1+p_2$
$R_{1,2}$	$5p_r+2p_1+p_2$	$6p_r+2p_1+p_2$
$R_{0,3}$	$6p_r+2p_1+p_2$	$8p_r+2p_1+p_2$
$R_{3,1}$	$\max\{3p_r+p_1+p_2+p_3, 8p_r+2p_1+p_2\}$	$5p_r+p_1+p_2+\max\{p_3, 5p_r+p_1\}$

The first cycle starts at

$$\begin{aligned} t_1 &= \max\{6p_r + 2p_1 + 2p_2, 5p_r + p_1 + p_2 + \max(p_3, 5p_r + p_1)\} \\ &= 5p_r + p_1 + p_2 + \max\{p_r + p_1 + p_2, p_3, 5p_r + p_1\}. \end{aligned}$$

v_2 is the idle time between $R_{2,2}$ and $R_{1,3}$.

$$\begin{aligned} v_2 &= \max\{8p_r + 3p_1 + p_2 - t_1 - 2p_r, 0\} \\ &= \max\{8p_r + 3p_1 + p_2 - 5p_r - p_1 - p_2 - \max\{p_r + p_1 + p_2, p_3, 5p_r + p_1\} - 2p_r, 0\} \\ &= \max\{p_r + 2p_1 - \max(p_r + p_1 + p_2, p_3, 5p_r + p_1), 0\} \\ &= \max\{p_1 - \max[p_2, 4p_r, p_3 - p_1 - p_r], 0\}. \end{aligned}$$

In general, for j = 2, …, L-2, v_j is the idle time between $R_{2,j}$ and $R_{1,j+1}$, and P_{62j} is the interval from the start of $R_{2,j}$ to the start of $R_{2,j+1}$. The start and end time of each move in the cycle can be calculated as we did previously. $P_{62j} = 4p_r + v_j + \max\{p_2, p_3-v_j, 4p_r\}$. $v_{j+1} = \max\{p_1-\max(p_2, p_3 - v_j, 4p_r), 0\}$. The average cycle time

$$P_{62} = \frac{1}{L-3}\sum_{j=2}^{L-2} P_{62j}.$$

The completion phase starts at time $t_2 = t_1 + (L-3)P_{62}$. The completion phase is $R_{2,L-1}$- $R_{1,L}$ - $R_{3,L-1}$ - $R_{2,L}$ - $R_{3,L}$. We can compute the start and end times of each move as we did before. The lot makespan

$$\begin{aligned} MS_{62} &= t_2 + 7p_r+p_3+v_{L-2}+\max\{2p_r, p_2, p_3- v_{L-2}\} \\ &= 5p_r + p_1 + p_2 + \max\{p_r + p_1 + p_2, p_3, 5p_r + p_1\}+(L-3)P_{62}+7p_r+p_3+w_{L-2}+\max\{2p_r, p_2, p_3- v_{L-2}\} \\ &= 12p_r + p_1 + p_2 +p_3+v_{L-2}+ \max\{p_r + p_1 + p_2, p_3, 5p_r + p_1\}+\max\{2p_r, p_2, p_3- v_{L-2}\}+(L-3)P_{62}. \end{aligned}$$

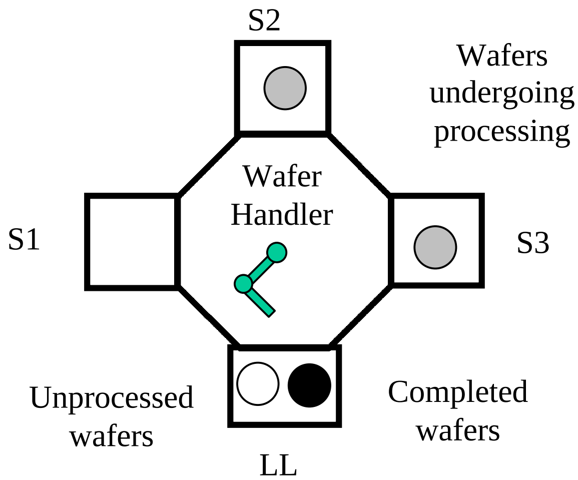

Figure 1. A three-stage sequential cluster tool.

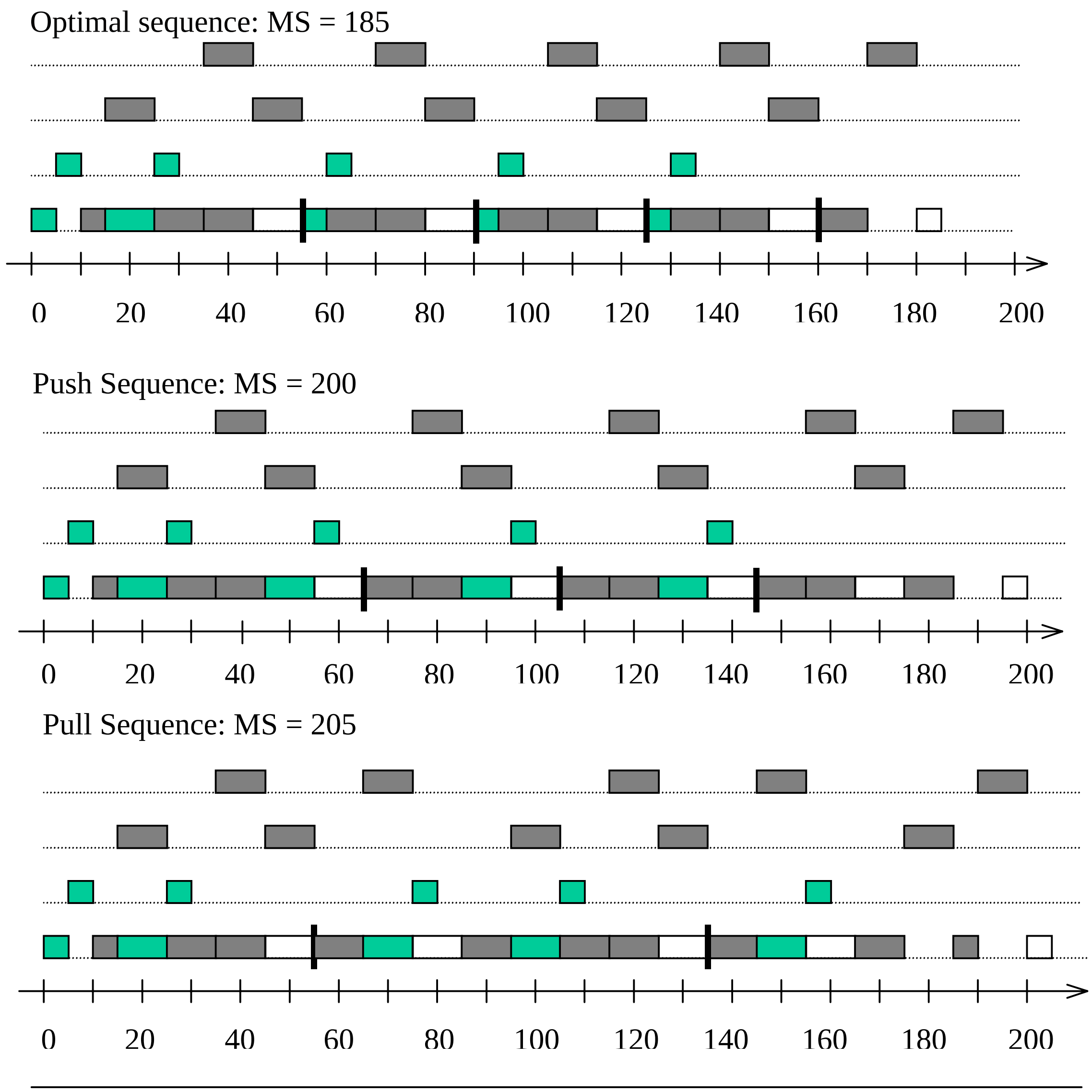

Figure 2. An optimal sequence, the push sequence, and the pull sequence.

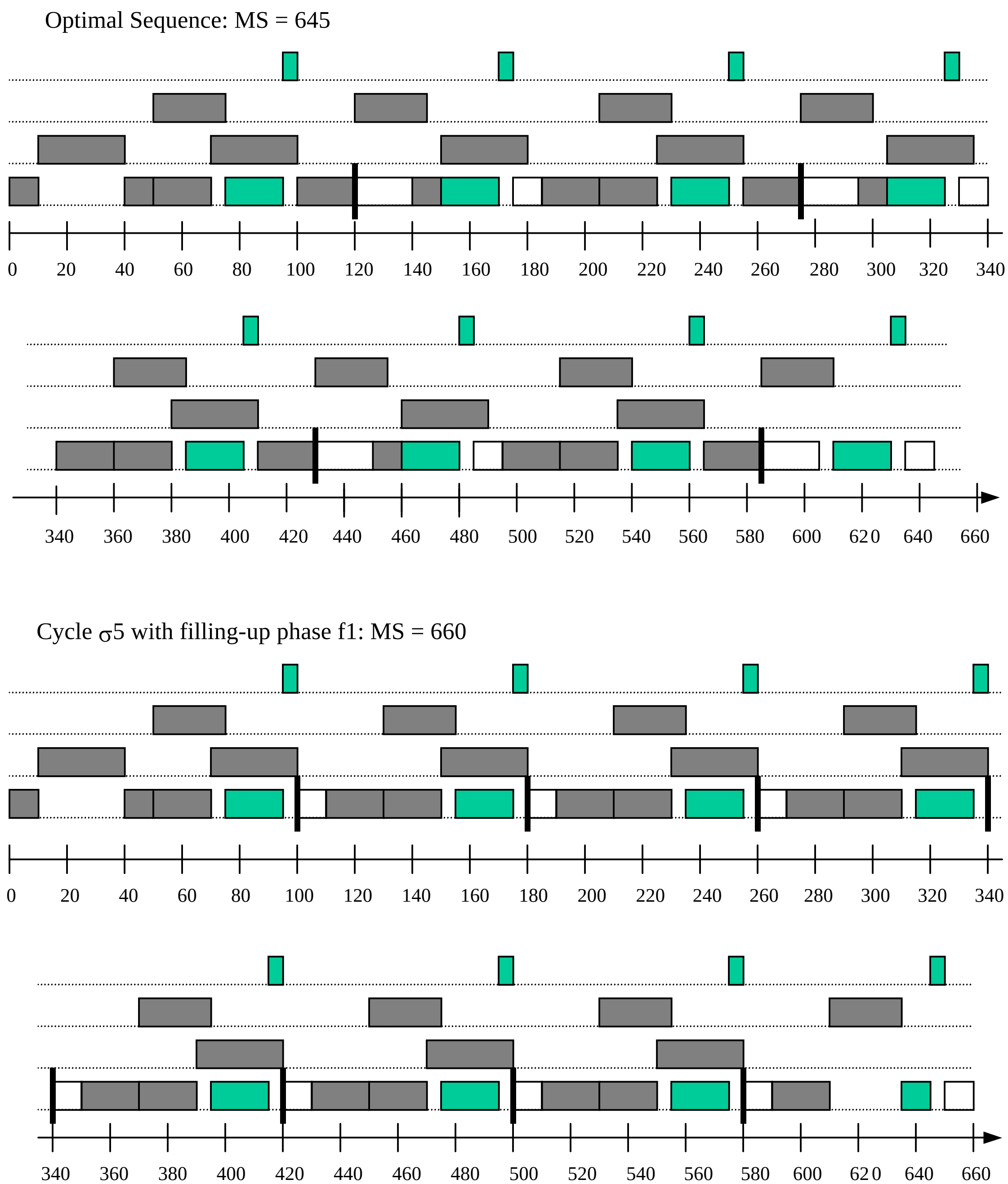

Figure 3. An optimal two-unit cyclic sequence and a one-unit cyclic sequence.

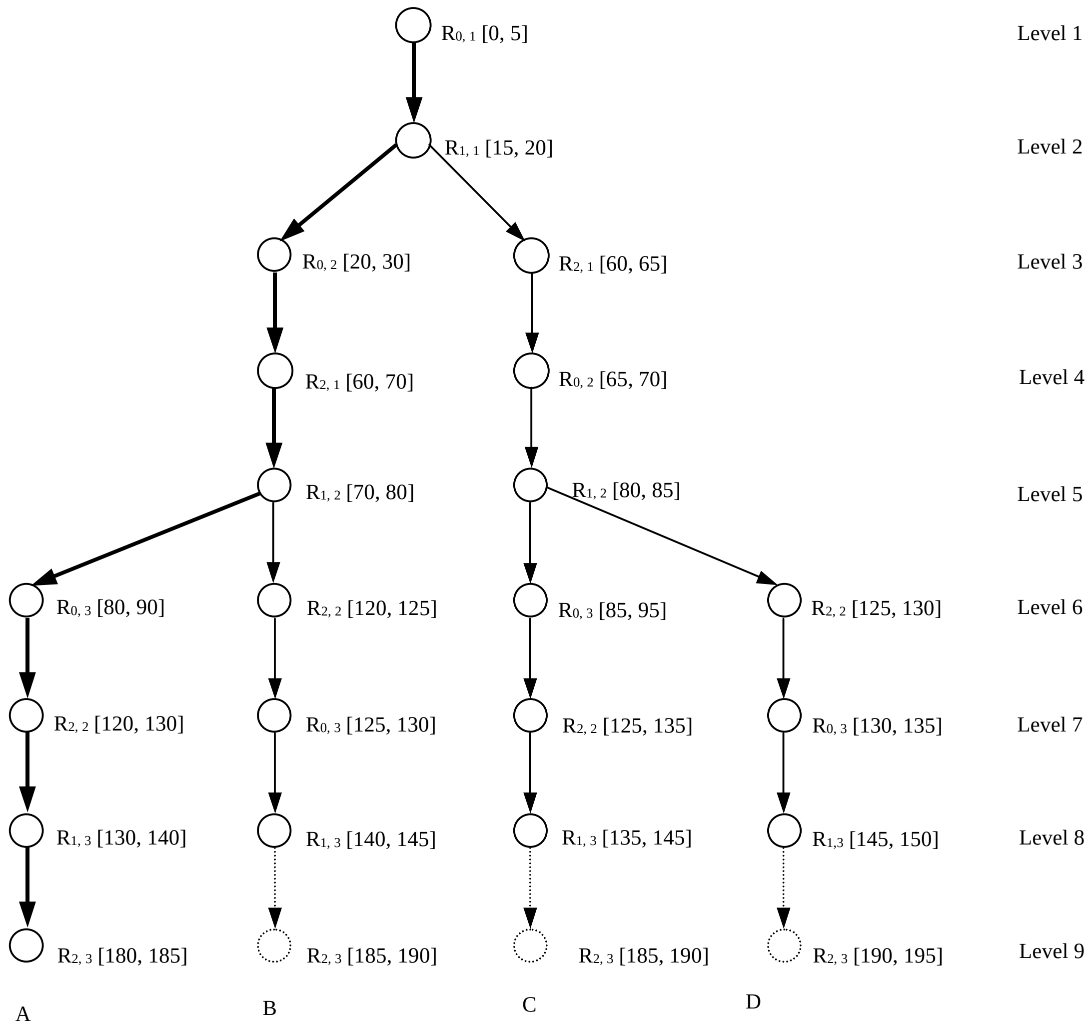

Figure 4. Search tree for CT1-1.
Note R_{ij} [x, y] denotes that move R_{ij} starts at time x and ends at time y.

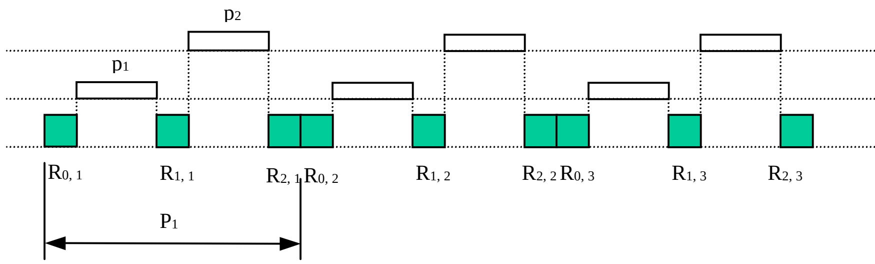

Figure A1

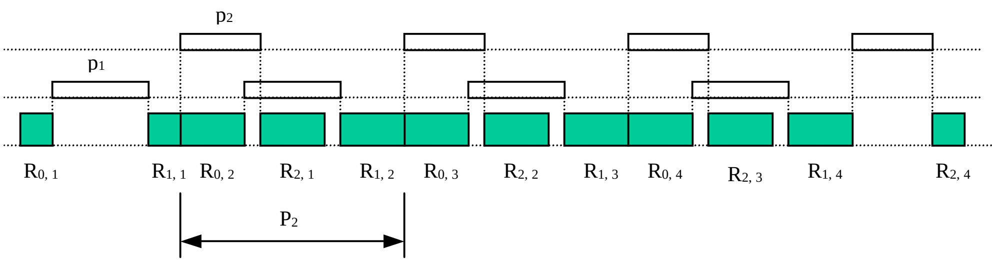

Figure A2

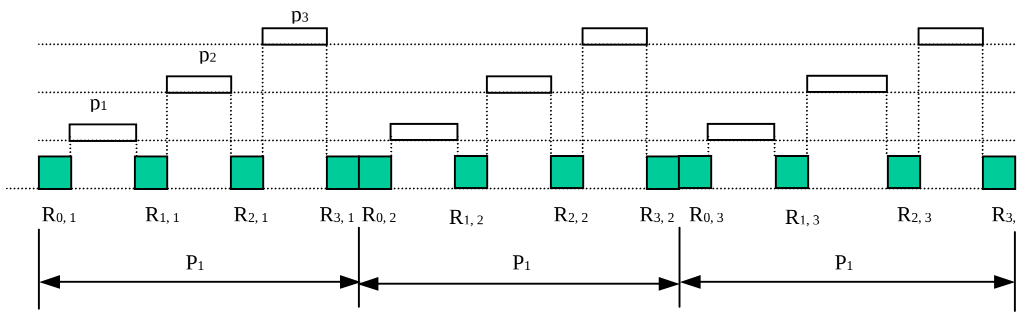

Figure B1.

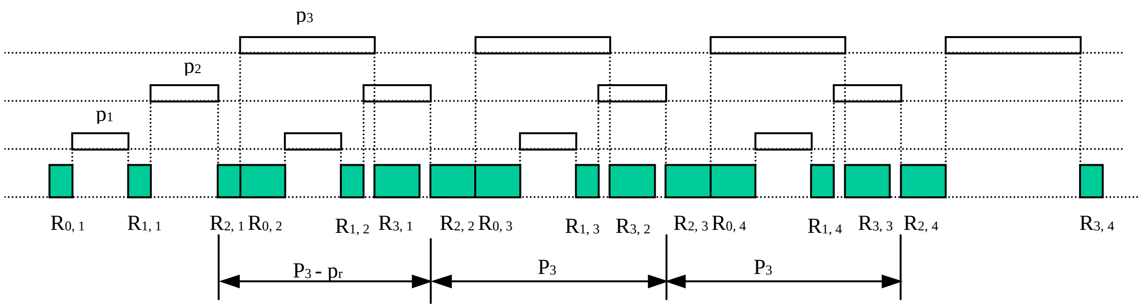

Figure B2.

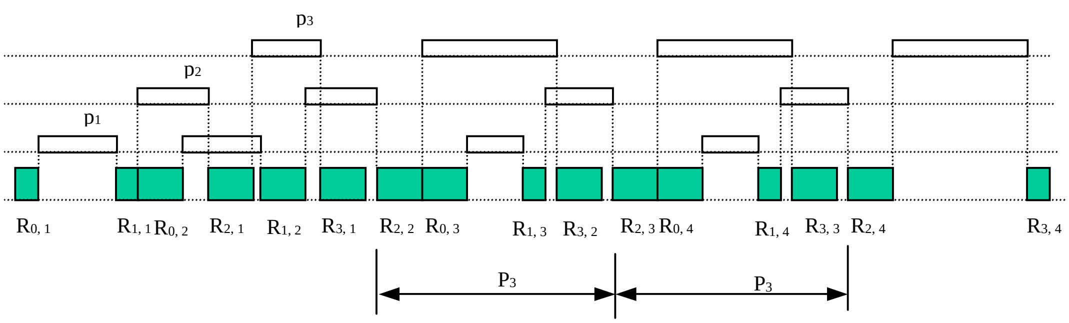

Figure B3.

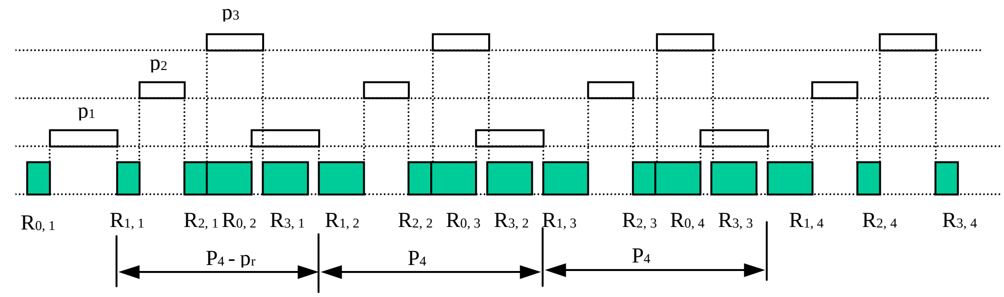

Figure B4.

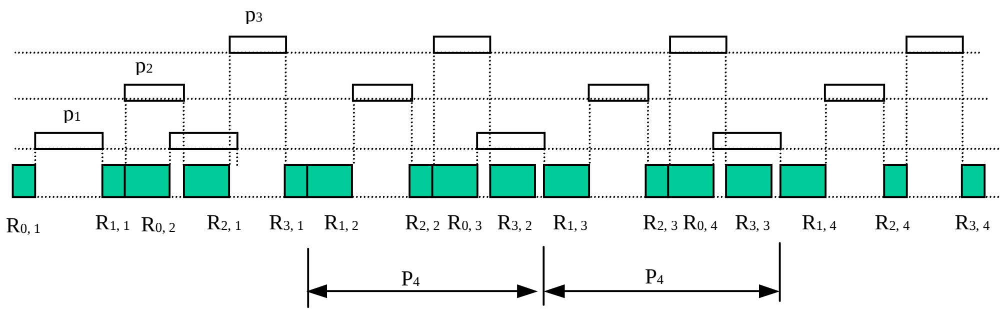

Figure B5.

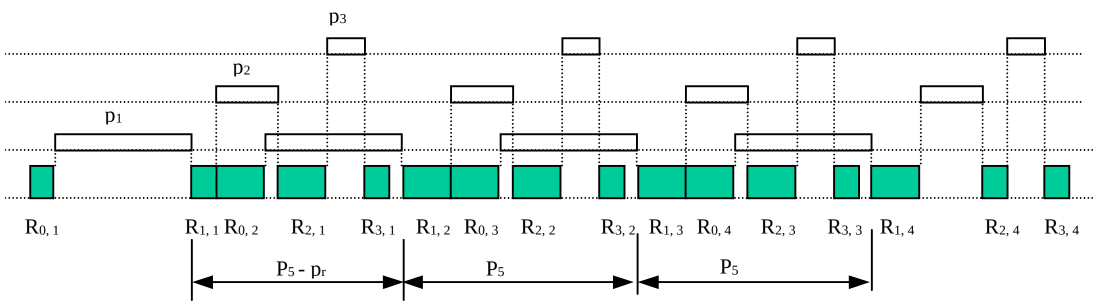

Figure B6.

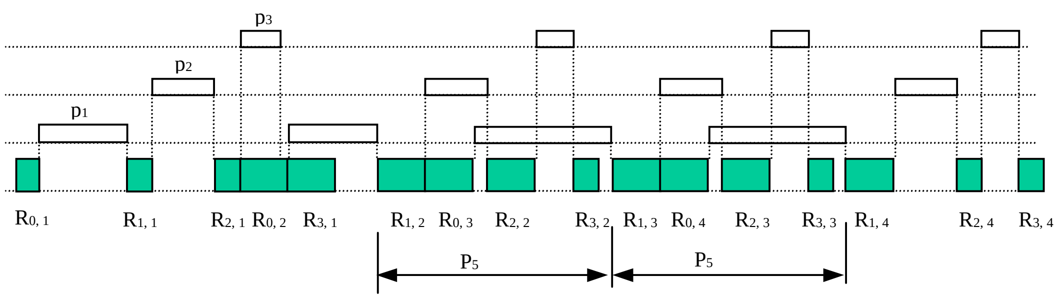

Figure B7.

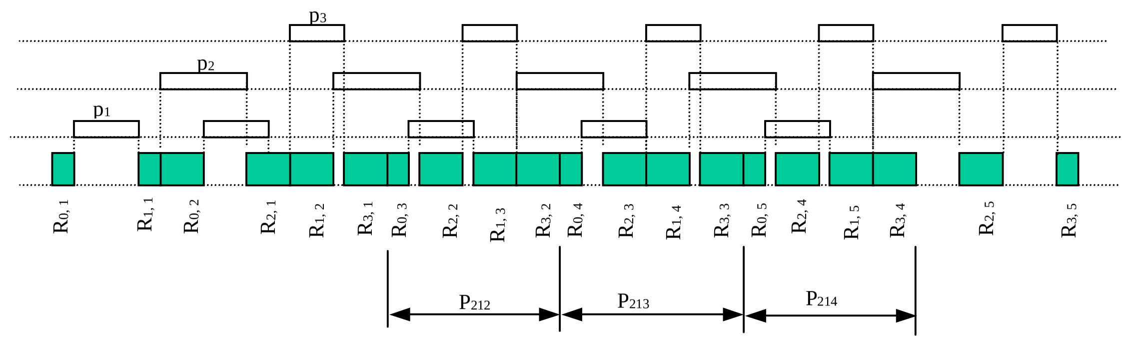

Figure C1.

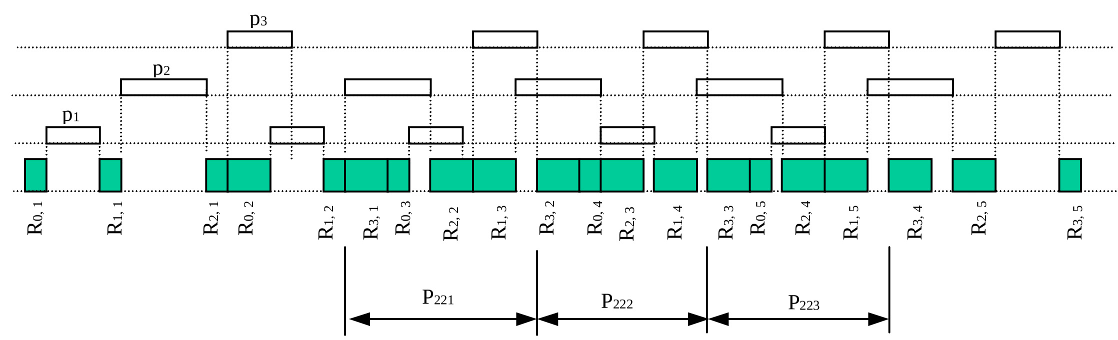

Figure C2.

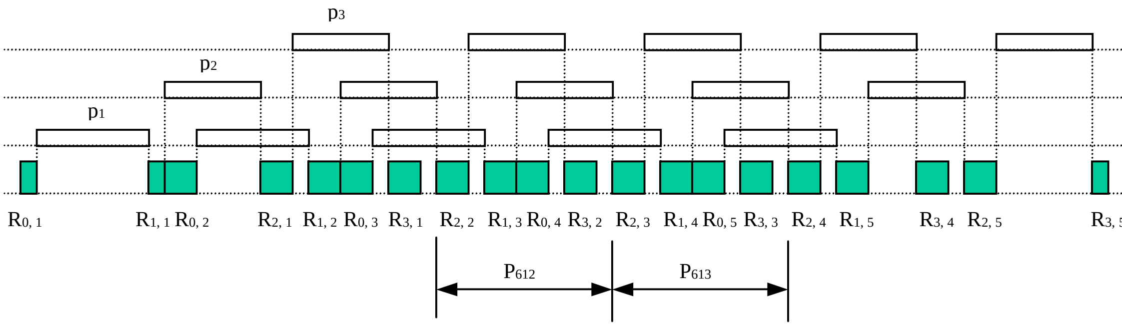

Figure D1.

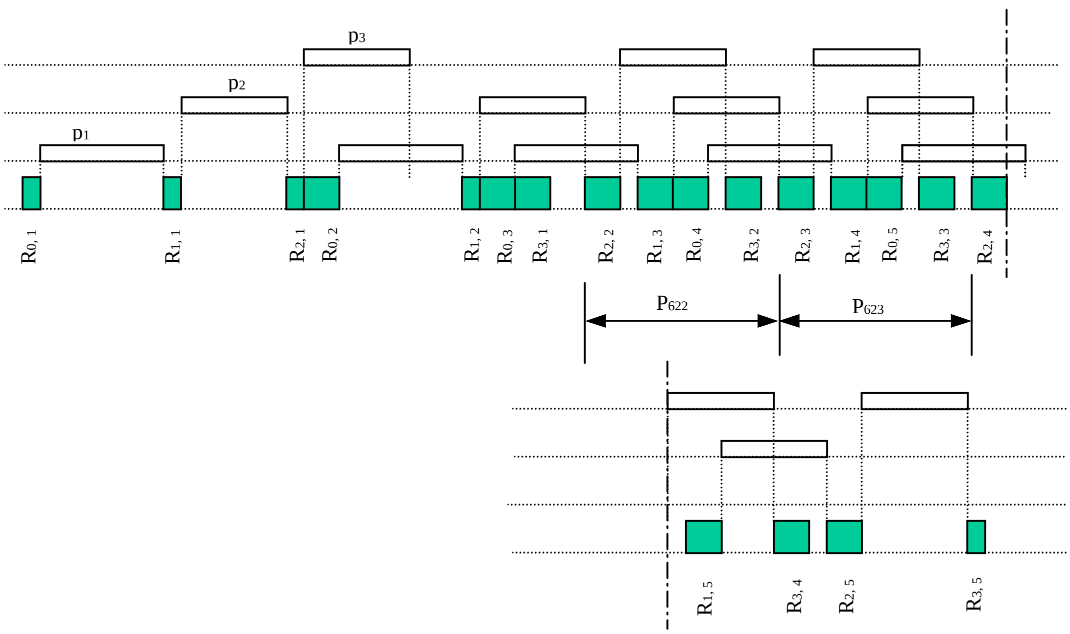

Figure D2.