AN ANALYSIS OF PURE ROBOTIC CYCLES

A THESIS
SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Serdar Yıldız
July, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. M. Selim Aktürk (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Oya Ekin Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Hakan Gültekin

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

AN ANALYSIS OF PURE ROBOTIC CYCLES

Serdar Yıldız
M.S. in Industrial Engineering
Supervisor: Prof. Dr. M. Selim Aktürk
July, 2008

This thesis is focused on scheduling problems in robotic cells consisting of a number of CNC machines producing identical parts. We consider two different cell layouts which are in-line robotic cells and robot centered cells. The problem is to find the robot move sequence and processing times on machines minimizing the total manufacturing cost and cycle time simultaneously. The automation in manufacturing industry increased the flexibility, however it is not widely studied in the literature. The flexibility of machines enables us to process all the required operations for a part on the same machine. Furthermore, the processing times on CNC machines can be increased or decreased by changing the feed rate and cutting speed. Hence, we assume that a part is processed on one of the machines and the processing times are assumed to be controllable. The flexibility of machines results in a new class of cycles named pure cycles. We determined efficient pure cycles and corresponding processing times dominating the rest of pure cycles in the specified cycle time regions. In addition, for in-line robotic cells, the optimum number of machines is determined for given parameters.

Keywords: In-line robotic cell, CNC, scheduling, bicriteria optimization, controllable processing times, robot centered cell, cycle time minimization, manufacturing cost minimization.

ÖZET

TEK ATAMALI ROBOTİK DÖNGÜLER ÜZERİNE BİR ANALİZ

Serdar Yıldız
Endüstri Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. M. Selim Aktürk
Temmuz, 2008

Bu tezin konusu, aynı tür parçalar üreten, belirli bir sayıda CNC makinadan oluşan robotik hücrelerde ortaya çıkan çizelgeleme problemleridir. Bu çalışmada, doğrusal robotik hücreler ve robot merkezli robotik hücreler göz önünde bulundurulmuştur. Bu problemdeki amacımız, birim başına düşen üretim maliyetini ve döngü süresini aynı anda enküçülten robot hareket döngüsünü ve bu döngüye karşılık gelen, makinalar üzerindeki üretim zamanlarını bulmaktır. Üretim endüstrisindeki otomasyonlar hücrelerin esnekliğini arttırdı. Ancak, robotik hücrelerde esneklik üzerine literatürde yeterince araştırma bulunmamaktadır. Bir ürün için gereken üretim işlemlerinin tümü bir CNC makinada gerçekleştirilebilir. Ayrıca, CNC makinalardaki üretim süreleri, besleme ve işleme hızına bağlı olarak azaltılıp arttırılabilir. Bu nedenle, bir parça üzerindeki üretim işlemlerinin tek bir makinada yapıldğını ve üretim sürelerinin kontrol edilebilir olduğunu varsaydık. Esnek makinalar, yeni bir döngü sınıfı olan tek atamalı döngülerin oluşturulmasına olanak vermiştir. Bu tezde tek atamalı döngüler göz önünde bulundurulmuştur. Belirtilen döngü zamanı alanlarında, diğer bütün tek atamalı döngüleri başatlayan tek atamalı döngüler ve üretim zamanlarını belirledik. Ayrıca, doğrusal robotik hücreler için, verilen değerlere göre en iyi makina sayısını belirledik.

Anahtar sözcükler: Doğrusal robotik hücre, CNC, robotik hücre çizelgelemesi, iki kriterli eniyileme, kontrol edilebilir üretim zamanları, döngü zamanı enküçültme, üretim maliyeti enküçültme.

Acknowledgement

I would like to express my gratitude to my supervisors Prof. Dr. M. Selim Aktürk and Assoc. Prof. Dr. Oya Ekin Karaşan for their instructive comments, guidance and encouragement in the supervision of the thesis.

I am also grateful to Assist. Prof. Hakan Gültekin for excepting to read and review this thesis and for his invaluable suggestions.

Finally, I would like to express my special thanks and gratitude to TÜBİTAK for the scholarship provided throughout the thesis study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The automation in manufacturing processes increased as the technology created special appliances for automation and improved the machines used in today's industry. Robots and Computer Numerically Controlled (CNC) machines are the important automation appliances that are considered in this thesis. Since robots increase the efficiency and reduce the labor cost, they are used in many diverse industries such as semiconductor manufacturing industry and electroplating applications, chemical operations, and metal cutting industry [9]. In this thesis, we focus on the metal cutting applications in which the machines are predominantly CNC machines. The robots have different duties in different industries. One of the most important applications of robots is using them as material handling instruments. The robot handling costs may constitute from 10% to 80% of total costs according to the type of manufacturing facility [33]. A robotic cell is defined as a manufacturing cell composed of a number of machines and a material handling robot. We assume that there are no buffers at or between the machines, thus, at any time, a part is either on one of the machines, at the input or output buffers or on the robot.

In this thesis, we focused on the robot move sequence, the processing times on machines and the design of the robotic cell which are important decisions that have to be made in robotic cells. For the design of the cell, we consider two different layouts. The first cell layout considered is an m-machine in-line robotic cell and the second cell layout is an m-machine robot centered cell. There is a single robot with a single gripper in both of these layouts. For the first cell layout, there is a bicriteria optimization problem of minimizing the cycle time and minimizing the manufacturing cost simultaneously. Additionally, as a design problem, the optimum number of machines in the cell is calculated. For the robot centered cell, the first problem considered is the minimization of cycle time that results in the maximization of throughput which is prominent in production planning. The second problem considered is the bicriteria optimization of minimizing cycle time and total manufacturing cost simultaneously.

Highly flexible CNC machines are used for metal cutting operations in robotic cells. The machines and the robot are used simultaneously in robotic cells. The cutting speed and the feed rate are controllable variables in CNC machines so that the processing times on these machines can be decreased at the expense of decreasing tool life and consequently increasing tooling cost. Considering fixed processing times is not convenient for real life problems in these robotic cells. For the bicriteria optimization problems considered in this thesis, the processing times on machines are assumed to be decision variables due to the controllability assumption.

As the flexibility of machines increase with the technological advance, new problems arise to be solved. The cyclic scheduling is widely studied in the literature and we focus on cyclic schedules. In this thesis, we restrict ourselves to pure cycles resulting from the flexibility of machines. Pure cycles are defined by Gultekin et al. [12] as the robot move cycles in which the robot loads and unloads all m machines with a different part during one repetition of the cycle, thus for each repetition a pure cycle produces m parts. Each part is completely performed by only one machine and no part is transferred from one machine to another one. Furthermore, since pure cycles are practical, easy to understand and easy to put in practice, they could have significant implementation possibilities in industry. Gultekin et al. [12] considered the case where the processing times of all machines are fixed and are the same. In this regard, they proved that the set of pure cycles dominate all flowshop type robot move cycles in terms of cycle time. Then, they showed that two specific pure cycles perform significantly better than the other robot move cycles among the class of feasible robot move cycles and they derived the regions of optimality for these two cycles. For the remaining region, they derived the worst case performances of these cycles. However, the bicriteria problem of finding the best pure cycle and processing times minimizing the cycle time and the manufacturing cost simultaneously in an m-machine robotic cell is not studied in the current literature. Hence, we set out to close this gap in the literature with this study.

For the bicriteria problems considered in this study, the processing times are decision variables which are determined according to two objectives and these objectives are minimization of cycle time and minimization of total manufacturing cost. In order to increase the throughput rate, the minimization of cycle time is more important and it is studied in the literature widely. Although the minimization of total manufacturing cost objective is one of the most fundamental objectives in the manufacturing literature, as far as authors know, in robotic cell literature, the only study considering this objective is by Gultekin et al. [13]. Since, the problem we are focusing is a bicriteria optimization problem, the efficient solution set is composed of nondominated solutions.

The bicriteria problem that we consider in this thesis is composed of two important objectives that are thoroughly investigated in the literature separately. These two objectives are minimizing cycle time and minimizing manufacturing cost. The complexity of the problems increases when the objective of the problem is changed from a single objective problem to a multicriteria problem. There are efficient ways of solving the bicriteria problems in the literature and some of these solution methods are summarized by Hoogeven [20]. In addition, most of the real life problems consist of more than one objective. The reason for that is single objective optimization results in loose solutions for the other performance measures when there is a trade-off between the performance measures. The single objective of cycle time minimization may result in a solution that performs inefficient in terms of manufacturing cost. Since these two objectives are as important as each other, the solution of bicriteria optimization problem results in solutions with cycle times and total manufacturing costs which are between the solutions of minimizing cycle time problem and minimizing manufacturing cost problem, separately. Thus, the focus of this thesis is on bicriteria objective of minimizing both the cycle time and the total manufacturing cost, simultaneously.

1.1 Motivation

Different from the current literature, for the bicriteria problems considered in this thesis, our problem is to determine the pure cycles and the corresponding nondominated processing time vectors in order to minimize total manufacturing cost and cycle time simultaneously in robotic cells. The considered objectives are prominent objectives in the literature and this problem is not studied in the literature. Most of the studies focus on one machine problems since they are easier to analyze. The complexity of problems increases as the number of machines in the cell increases. Although they are more complex, we focused on m-machine robotic cells where m is any positive integer value. In addition, the most popular cost function used in the literature is a linear cost function however in reality, the cost functions are mostly not in linear structure. Thus, the cost function considered in this thesis is a nonlinear, strictly convex and differentiable cost function. Furthermore, we solve the bicriteria problem for two different cell layouts which are an m-machine in-line robotic cell and an m-machine robot centered cell.

1.2 Organization of the Thesis

This thesis is organized as follows: Chapter 2 summarizes the studies in the literature on robotic cell scheduling, bicriteria scheduling and controllable processing times. In Chapter 3, the assumptions and definitions used throughout the thesis are explicitly presented. In Chapter 4, two problems considered in robotic cell scheduling literature are analyzed for m-machine in-line robotic cells. The first problem considered is the bicriteria analysis of pure cycles in m-machine in-line robotic cells. The second problem considered is finding the optimum number of machines in the cell as a design problem. In Chapter 5, there are two problems to be solved for m-machine robot centered cells. The layout of the cell is changed to robot centered cell. For an m-machine robot centered cell, the efficient pure cycles are investigated according to the objective of minimizing cycle time. In the second problem, controllable processing times are considered and the problem is investigating the efficient pure cycles minimizing both the cycle time and the total manufacturing cost simultaneously. Finally, in Chapter 6, the summary of the thesis and some future research directions are presented.

Chapter 2

Literature Review

In this thesis, we consider a bicriteria optimization problem in robotic cells where the production process is considered as cyclic. Gultekin et al. [12] and Gultekin et al. [14] studied on minimizing cycle time in robotic cells. Gultekin et al. [14] focused on process and operational flexibility and proposed a new class of robot move cycles named as pure cycles. They proved that this new class of cycle dominates all classical robot move cycles considered in the literature for $m = 2$. Furthermore, they proved that changing the layout from an in-line robotic cell to a robot-centered cell reduces the cycle time of the proposed cycle even further, whereas the cycle times of all other cycles remain the same. For the m-machine case, they found the regions where the proposed cycle dominates the classical robot move cycles. In addition, Gultekin et al. [12] proved that pure cycles dominate the flowshop type robot move cycles studied in the literature according to the single objective of minimizing cycle time. Therefore, we focus on pure cycles in this study. Furthermore, the new production environments give us the opportunity to decrease or increase the processing times of jobs in specified boundaries. The processing speed of machines can be altered to change the total cost of production as well as the processing times. Using this property, more economical ways of production can be determined which also maximizes the throughput rate. So, the processing times are assumed to be controllable in our study.

There are many studies on minimizing cycle time in robotic cell scheduling literature. The minimization of cost objective is one of the prominent objectives in manufacturing systems, however this objective is not well studied in the literature on robotic cell scheduling. So, our study is distinctive from the studies considering only minimization of cycle time or only minimization of total manufacturing cost in robotic cells. The robotic cell scheduling problems are classified according to machine environment, processing characteristics and objectives in Dawande et al. [9] which are described in the following three parts.

1. Machine environment

The robotic cells including single machines for each stage are named as *simple robotic cells* or *robotic flowshops*. If there are more than one machine at least in one stage, the cell is named as *robotic cell with parallel machines*. In order to increase the throughput rate, more than one robot can be placed in the cell. The robotic cells including one robot is called *single robot cells* and the cells containing more than one robot is named as *multiple robot cells*. The robots studied in the literature are single gripper robots and dual gripper robots. The single gripper robots can hold only one part at a time. The dual gripper robots are able to hold two parts at the same time. The robotic cell considered in this thesis includes a single robot with a single gripper.

2. Processing characteristics

Most of the studies in robotic cells assume that there is no buffers for intermediate storage. Thus, a part can be in the input device, in the output device, on the machine or on the robot at any time instant. According to the pickup criterion, robotic cells are classified in three groups. The main assumption is that a part unloaded from machine i, M_i, can be loaded to M_{i+1} if M_{i+1} is unoccupied. In *free-pickup* cells, a completed part may remain on M_i indefinitely. In *no-wait* cells, the part must be unloaded from M_i and loaded to M_{i+1} as soon as the process on M_i is finished. The no-wait pickup criterion is considered in Hall and Sriskandarajah [19] and Kats and Levner [22]. In *interval* robotic cells, each stage has a specific interval of time to be processed. The interval robotic cells are considered in hoist scheduling problems such that Lei and Wang [24]. The pickup criterion is assumed to be free-pickup criterion in this thesis.

The robot travel time is another important processing characteristic of the cell. The distance between machine i, M_i, and machine j, M_j, is denoted as $d(M_i, M_j)$. The robot's travel time between any consecutive machines can be equivalent as δ. For additive travel times in in-line robotic cells, the distance between any two machines M_i and M_j, $0 \leq i,\ j \leq m+1$, $d(M_i, M_j) = |i - j|\delta$. For certain cells (Dawande et al. [10]), the distance between any two machines can be assumed as equal and these travel times are named as constant travel times. The third travel time type is Euclidean travel times in which the travel time from a machine to itself is zero and the travel times satisfy the triangular inequality. We consider additive travel times in this thesis. There are also some studies which assume non Euclidean travel times in the literature.

The robotic cells producing identical parts are called as *single part-type* cells. In contrast, if the cell produces more than one type of parts, then the cell is named as *multiple part-type* cell. We focus on single part-type robotic cells in this thesis.

3. Objectives

The only objective dealt in the literature is maximizing the throughput. As far as authors know, there is only one study considering manufacturing cost in robotic cell literature. In general, dealing with single objective problems is simpler than dealing with multicriteria objectives. Hence, there are several papers studying on single objective problems in the literature. Our objective in this thesis is a bicriteria objective considering both of the objectives presented previously.

There is an extensive literature on robotic cell scheduling problems as summarized by the surveys in Dawande et al. [9] and Crama et al. [8]. In addition, TSP based approaches used for robotic cells are presented in the survey of Sriskandarajah et al. [2]. Furthermore, the bicriteria optimization literature is presented extensively in Hoogeven [20]. In the survey paper of Shabtay and Steiner [30], there is an extensive literature on scheduling with controllable processing times. From now on, the literature is going to be presented in three closely related subjects to this thesis. These subjects are presented in the following order. First, the robotic cell scheduling is presented in two subtopics: *i*) cyclic scheduling and *ii*) multiple part type problems. The second subject is the bicriteria optimization and the third subject is controllable processing times.

2.1 Robotic Cell Scheduling

The robotic cells are used in many diverse industries such as semiconductor manufacturing industry, hoist electroplating line, testing and inspection boards used in mainframe computers [9]. We can present some representative studies on these subjects as follows. Akcali et al. [1], Kumar et al. [23], Perkinson et al. [27], Perkinson et al. [28], and Wood [35] are some of the studies on robotic cell application in semiconductor manufacturing industry. An example of robotic cell study in hoist electroplating for printed circuits is Lei et al. [24]. Miller [26] studied for testing and inspecting boards used in mainframe computers. In the next part, we present the robotic cell scheduling literature on cyclic production and multiple part-types.

2.1.1 Cyclic Scheduling

Since cyclic schedules are easy to implement and control and are the primary way of specifying the operation of a robotic cell industry, the cyclic scheduling is a prominent study area in the literature. The definition of cycles is presented in Dawande et al. [9]. In order to define cycles, first the robot activities are defined, then *k-unit activity sequence* is defined and finally a *k-unit cycle* is defined. A robot activity is defined in Crama et al. [6] as follows:

Definition 2.1. *A_i is the robot activity defined as; robot unloads machine i, transfers part from machine i to machine $i+1$, loads machine $i+1$.*

The *k-unit activity sequence* is defined in Dawande et al. [9] as follows:

Definition 2.2. *A k-unit activity sequence is a sequence of robot moves which loads and unloads each machine exactly k times.*

In the light of this definition, the *k-unit cycle* is defined in Dawande et al. [9] as follows:

Definition 2.3. *A k-unit cycle is the performance of a feasible k-unit activity sequence in a way which leaves the cell in exactly the same state as its state at the beginning of those moves.*

From now on, the literature on cyclic scheduling is summarized and the results of important studies are presented as follows. The Sethi et al. [29] is a fundamental study on cyclic scheduling in robotic cells. They proved that 1-unit cycles result in the maximum throughput for 2-machine robotic flowshops. They used the free pick-up criterion and the robot travel times are assumed to be additive. They conjectured that the 1-unit cycles may also be the optimum cycles for $m \geq 3$ machine case. For the same problem but 3-machine case of maximizing throughput, Crama and van de Klundert [7] proved that the conjecture holds. However, Brauner and Finke ([3], [4]) found a counterexample which results in less per unit cycle time for 4-machine cell. This conjecture does not hold when $m \geq 4$.

In this thesis, we focus on pure cycles described by Gultekin et al. [12] and which are m-unit cycles. We analyzed pure cycles in Chapters 4 and 5. In the next part, we move to present the literature on multiple part-type studies.

2.1.2 Multiple Part-Types

Studying identical part type problems is easier in theoretical means, thus most of the studies in robotic cells are focused on identical part type problems. However, in real life industry, an important amount of manufacturing facilities produce different types of parts. The multiple part-type problems increase the complexity of problems tremendously.

One of the decisions to be made for multiple part-type problems is to decide the sequence of parts to be produced in the cell. For solving multiple part-type problems, the minimal part set (MPS) structure is commonly used. The proportions of the different part types in the lot have to be the same of the proportions of part types in the demand as in just in time (JIT) manufacturing systems [9]. For example, if the part type A constitutes %30 of demand and the part type B constitutes %70 of demand, then for a demand of 10 units lot size, the MPS has 3 parts of type A and 7 parts of type B. The other part type sequence considered in the literature is concatenated robot move sequences (CRM sequences). Indeed, it is a type of MPS cycles in which the robot move sequence is the same 1-unit cycle of robot move sequences repeated n times [31].

In order to summarize the results obtained from multiple part-type case literature in robotic cells, the following papers are useful to present. In 2-machine robotic flowshops, for the CRM sequence corresponding to reverse cycle $\pi_D = S_2 = (A_0, A_2, A_1)$, Logendran and Sriskandarajah [25] solved the the optimal part schedule problem where no-wait pick-up criterion is assumed and 1-unit cycles are considered only. They formulated the problem as a solvable type of TSP problem which is solved by using the algorithm in Gilmore and Gomory [11]. One another study analyzed during thesis is Hall et al. [18] where they developed a polynomial time algorithm to find the robot waiting times at different machines and the cycle time for a given part schedule for the specified robot move sequences.

Hall et al. [17] studied on 3-machine robotic flowshop cells in order to maximize the throughput and they made complexity analysis for the possible cycles. They showed that Gilmore and Gomory [11] algorithm can be used to find the optimum part schedule for the three CRM sequences based on three of the possible cycle. They showed that for one of the cycles the problem is trivial since the cycle time does not depend on part schedule for that cycle. For the remaining two cycles, Hall et al. [17] proved that finding the optimal part schedule for the CRM sequences based on these robot move cycles is NP-Hard, unless the special conditions on the data are met. Thus, even in 3-machine cells and even fixing the robot move cycle, finding the optimum part schedule can be an NP-Hard problem. In the next part, we present a summary of bicriteria optimization studies which are investigated during thesis study.

2.2 Bicriteria Optimization

In this part, the literature on bicriteria optimization is briefly presented. Since dealing with single objective problems is relatively easier, most of the studies are focused on single objectives. The optimum solutions for a single objective may perform poorly according to the other objectives because of trade-off relation between objectives. A review for multicriteria scheduling models is presented in Hoogeveen [20]. The multicriteria scheduling problems are more complex, thus it is helpful to use the well studied solution methods for this kind of problems. In Hoogeveen [20], there are different methods to solve bicriteria problems and we used two methods presented in Hoogeveen [20] to solve the bicriteria problems in this thesis.

Some important solution methods presented in Hoogeveen [20] to deal with bicriteria problems are presented as follows. Suppose that we have two performance measures f and g to be minimized. For the first problem, assume that performance measure f is far more important than g. In this problem, first, the optimum solution for performance measure f is determined. Then, from these optimal solutions for f, the one that results in minimum g is selected as the best solution for this bicriteria problem. This solution method is named as *hierarchical optimization* or *lexicographical optimization* and denoted as $Lex(f, g)$ in T'kindt and Billaut [32].

For the second problem, assume that no criteria is more dominant than the other. This problem is the bicriteria problem we consider in this study and the set of pareto-optimum solutions for this problem is achieved by using simultaneous optimization. There are three ways to solve this problem and we present the one which is used in our study. First, we compose the composite function $F(f(\sigma), g(\sigma))$ where σ is the considered robot move sequence. Since the two objectives in our problem are equally important, we use the *posteriori optimization* in this problem. The solution set obtained from this problem constitutes a nondominated set. A nondominated schedule is defined in Hoogeven [20] as follows:

Definition 2.4. *A feasible schedule σ is nondominated with respect to the performance criteria f and g if there is no feasible schedule π such that both $f(\pi) \leq f(\sigma)$ and $g(\pi) \leq g(\sigma)$, where at least the one of the inequalities is strict.*

This definition is used in our study in order to find the pure cycles and processing times dominating the rest of pure cycles. To find the nondominated points for this problem, we use the *epsilon-constraint approach* presented in the terminology of T'kindt and Billaut [32]. In this method, in the first step, the hierarchical optimization method is used where f is assumed to be the important performance measure. The minimum f value is found when an upper bound on g is given. By solving a series of subproblems of minimizing f subject to a given upper bound on g, the elements of nondominated solution set are determined.

As summarized previously, we use the *posteriori optimization* method where the *epsilon-constraint approach* is used to construct the nondominated solution set to solve the bicriteria optimization problem. There is only one study, Gultekin et al. [13] considering the bicriteria problem of minimizing the cycle time and the manufacturing cost in robotic cells.

2.3 Controllable Processing Times

Shabtay and Steiner [30] present an extensive literature review on scheduling with controllable processing times. Since analyzing linear cost functions is easier in theory, most of the current literature on controllable processing time problems focus on linear cost functions (Vickson [34], Cheng et al. [5]). Using linear cost functions does not reflect the law of diminishing returns. Thus, in our study, we use a nonlinear, strictly convex, and differentiable cost function.

The cost function we used in given bicriteria examples in Chapter 4 is modified from a cost function presented in Kayan and Akturk [21]. They determine the upper and lower bounds for the processing time of each job under controllable machining conditions. In this thesis, we modified a cost function as $Z_1 = \sum_{i=1}^{N}(O.P_i + TUP_i^{\alpha})$. T and α are specific constants for the tool. We consider the same single pass turning operation for every part. We assumed that T is the same for identical tools. It is assumed that U is a specific constant only depending on tools. In addition, as we assume the cost function is decreasing when processing time increases, α is a negative constant. The processing times are considered as controllable.

Gurel and Akturk [15] considered total manufacturing cost and total weighted completion time objectives simultaneously on a CNC machine. The decision of the appropriate processing times becomes as important as deciding the job sequence. After deducing some optimality properties, they proposed a heuristic algorithm to generate an approximate set of efficient solutions. In addition, Gurel and Akturk [16] considered the problem of minimizing total manufacturing cost subject to a given total completion time level and they gave an effective formulation for the problem. They found some optimality properties that facilitates designing an efficient heuristic algorithm to generate approximate non-dominated solutions. Gultekin et al. [13] considered the problem of finding the robot move sequence and the processing times minimizing total manufacturing cost and cycle time simultaneously in 2-machine and 3-machine flowshop robotic cells. They determined the sufficient conditions under which each of the cycles dominates the rest.

2.4 Summary

In this chapter, we reviewed the current literature. Most of the studies consider the robotic cell as a flowshop cell in which the parts are processed on each machine in the same order. The processing times are considered as fixed on all machines for all parts. However, the flexibility of machines, especially the CNC machines, enables us to process all operations required for a product on one machine. The speed, feed rate and cutting speeds in CNC machines can be altered in order to change the processing times. Most of the studies consider single objective problems, indeed the single objective solutions usually do not perform well for the other objectives. In general, the minimization of manufacturing cost is the most important objective for manufacturing industry, however it is not widely studied in the current literature. Furthermore, the bicriteria optimization problem considered in this thesis is not studied in the literature. Most of the studies are focused on single machine problems, however we focused on m-machine cells. In addition, in the literature, the linear cost functions are usually used to represent the cost functions. The cost function considered in this thesis is differentiable, strictly convex, and nonlinear. We considered m-machine in-line robotic cells and m-machine robot centered cells.

Chapter 3

Assumptions and Definitions

In this chapter, we review the standard terminology in the literature, assumptions and the notations used throughout this thesis. Firstly, pure cycles are defined and the necessary information on pure cycles are given. It is assumed that each machine is able to perform all of the operations of identical parts. Gultekin et al. [12], by using this flexibility, defined a new class of cycles named pure cycles and defined new robot activities to describe pure cycles as follows:

Definition 3.1. *L_i is the robot activity in which the robot takes a part from the input buffer and loads machine i, $i = 1, 2, \ldots, m$. Similarly, U_i , $i = 1, 2, \ldots, m$, is the robot activity in which the robot unloads machine i and drops the part to the output buffer. Let $\mathcal{A} = \{L_1, \ldots, L_m, U_1, \ldots, U_m\}$ be the set of all activities.*

There are m loading and m unloading activities in an m-machine robotic cell. Now, the definition of pure cycles in Gultekin et. al [12] can be presented as follows:

Definition 3.2. *Under a* **pure cycle***, starting with an initial state, the robot performs each of the $2m$ activities ($L_i, U_i,\ i = 1, \ldots, m$) exactly once and the final state of the system is identical with the initial state.*

In other words, any permutation of the m load and the m unload activities is a pure cycle. For example, in a 2-machine robotic cell, the robot activity set is $\mathcal{A} = \{L_1, L_2, U_1, U_2\}$ and the robot move sequence $L_1U_1L_2U_2$ is a pure cycle. Since there are m machines in a robotic cell that is considered in this thesis, each pure cycle produces m parts thus, each pure cycle is an m-unit cycle. A k-unit cycle, is defined by Dawande et al. [9] in Definition 2.3. The pure cycles are defined in Definition 3.2 and now, the feasible robot move sequences are defined in Crama et al. [8] as follows:

Definition 3.3. *A (possibly infinite) sequence π of robot activities is called a feasible robot move sequence if, in the course of executing the sequence,*

1. *the robot is never required to unload an empty machine and*

2. *the robot is never required to load a loaded machine.*

The definition of robot activities of pure cycles in Definition 3.1 implies that the robot never attempts to unload an empty machine and the robot never attempts to load an already loaded machine. The two requirements of feasibility for robot move sequences are satisfied in pure cycles thus, the pure cycles are feasible cycles in terms of feasibility requirements in Definition 3.3.

In this study, we use the notation of pure cycle as C_i^m which Gultekin et al. [12] defined as the i^{th} pure cycle in an m-machine robotic cell and they denoted the cycle time corresponding to the i^{th} pure cycle as $T_{C_i^m}$. Each of the identical parts are processed on one of the identical machines. All the operations performed on a part are processed only on one machine. In this study, P_i denotes the processing time on machine i for any identical part. Any part taken from the input buffer and loaded onto machine i is processed on that machine for P_i time units. A feasible processing time on the machine is bounded below by lower bound denoted as P^L and bounded above by the upper bound denoted as P^U and these bounds are the same for every machine. In other words, for any machine i, a feasible processing time can be stated as $P^L \leq P_i \leq P^U$. We denote a processing time vector as $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$ which is composed of processing times on machines. In a feasible processing time vector, all of the processing times on machines 1 to m have to take values between the upper bound and lower bound. Thus, we present the set of feasible processing time vectors as

$\mathcal{P}_{feas} = \{(P_1, P_2, \ldots, P_m) \in R^m : P^L \leq P_i \leq P^U, \ \ \forall i\}$.

The notations used throughout this thesis are described as follows:

ϵ :The load/unload times of the machines by the robot which are the same for all machines. The pick/drop times at input buffer, output buffer or at I/O station are also the same as ϵ time units.

K : Cycle time, the total time required to complete an m-unit pure cycle.

$f(P_i)$: The manufacturing cost incurred from processing time on machine i which is strictly convex, differentiable and monotonically decreasing for $P^L \leq P_i \leq P^U$, $\forall i$.

$F_1(C_i^m, \boldsymbol{P}) = \sum_{i=1}^m f(P_i)$: Total manufacturing cost depending only on the processing times.

$F_2(C_i^m, \boldsymbol{P})$: Cycle time corresponding to processing time vector $\boldsymbol{P}$ and the pure cycle C_i^m.

The only possible robot moves for a part are described as follows: any part which is taken from the input buffer is transferred to one of the m machines, after all of the operations are performed, the part is finally transferred to the output buffer. Between any two loadings of any machine, all other machines are loaded once. There are $(2m)!$ possible pure cycles and some of them represent the same cycle. For instance, in 2-machine case, $L_1U_1L_2U_2$ and $L_2U_2L_1U_1$ are different representation of the same cycle and there are $(2m-1)!$ pure cycles in an m-machine cell after removing the different representations.

The total manufacturing cost is the sum of tooling costs and machining costs. The machining cost is considered as a function of exact working times where the cost is incurred if and only if machine is working on a part. The machining cost increases as the processing times on parts increase but the tooling cost decreases simultaneously. Conversely, reducing processing times decreases machining cost, but increases tooling cost. We define $f(P_i)$ as the manufacturing cost incurred by the processing time of machine i, P_i. So, we define the total manufacturing cost of a repetition of a cycle as the sum of the manufacturing costs incurred by the processing times of all machines and it is denoted as $F_1(C_i^m, \boldsymbol{P}) = \sum_{i=1}^{m} f(P_i)$. The total manufacturing cost depends only on the processing times, but not on the robot move cycle. The cycle time is the time required to complete the activities in the cycle and finally return back to the initial state, which depends on both the robot move cycle and processing times and denoted as $F_2(C_i^m, \boldsymbol{P})$. In the next part, the solution method used to solve the bicriteria problems considered in Chapters 4.1 and 5.2 is defined.

3.1 Bicriteria Solution Procedure

There are two bicriteria problems considered to be solved in this study. One of these problems is solved in Chapter 4.1 and the other one is solved in Chapter 5.2. Both considers a bicriteria model but with different cell layouts. There are different solution methods for bicriteria problems as discussed in Hoogeven [20] and we use one of these solution methods which is described in this part. The bicriteria objective considered in both of these problems is identical and it is the minimization of the cycle time and the total manufacturing cost simultaneously. These two problems are solved by using the solution procedure presented in this part. The bicriteria problem is formulated as follows:

$$\begin{aligned} \text{minimize} \quad & \textit{Total manufacturing cost} \\ \text{minimize} \quad & \textit{Cycle time} \\ \text{Subject to} \quad & P^L \leq P_i \leq P^U, \quad \forall i \end{aligned}$$

There are different strategies presented in Hoogeven [20] to solve multicriteria problems. In our study, the nondecreasing composite function $F(f, g)$ is minimized where f stands for the total manufacturing cost and g stands for the cycle time. In this approach, all the nondominated points are generated and the decision maker indicates the preferable solution. Since it is hard to determine which performance measure is more important, it is useful to present all nondominated solutions and give the decision maker the opportunity of selecting the most appropriate solution for the situation. For each robot move sequence, the sufficient conditions for the processing time values minimizing the manufacturing cost are determined for a given cycle time level. In order to find all of the nondominated points, a series of problems are solved for each robot move sequence. Through this method, for each robot move sequence, the nondominated processing time vectors are found for all possible cycle time levels and finally these points are used to compose the solution set for minimizing $F(f,g)$. We will use the epsilon-constraint method denoted by $\epsilon(f|g)$ that finds the nondominated points by minimizing f given an upper bound for g. The epsilon constraint formulation of the problem is denoted as $\epsilon(F_1(C_i^m, \boldsymbol{P})|F_2(C_i^m, \boldsymbol{P}))$ that finds the processing time vector minimizing the total manufacturing cost $F_1(C_i^m, \boldsymbol{P})$ for a given level of cycle time $F_2(C_i^m, \boldsymbol{P})$. Thus for any given cycle time level, the following ECP is solved to find the nondominated processing time vector:

Epsilon-Constraint Problem(ECP)

$$\begin{aligned} \text{minimize} \quad & \textit{Total manufacturing cost} \\ \text{Subject to} \quad & \text{Cycle time} \leq K \\ & P^L \leq P_i \leq P^U, \quad \forall i \end{aligned}$$

Any feasible solution of the bicriteria problem corresponds to a feasible robot move sequence and a feasible processing time vector. This study is restricted to pure cycles, consequently the set of feasible cycles in an m-machine cell, which is denoted as $\mathcal{C}^m_{feas}$, is defined as the set of pure cycles in this cell. In the next definition, for a pure cycle, we define the efficient frontier consisting of nondominated points. The set of nondominated processing time vectors for an m-unit robot move cycle C_i^m and for a given cycle time level K is defined as follows:

Definition 3.4. *For a robot move sequence C_i^m and a given cycle time level K, the set of nondominated points is defined as $\mathbf{P}^*(C_i^m|K) = \{\boldsymbol{P} \in \mathcal{P}_{feas}$: There is no other $\mathbf{P}' \in \mathcal{P}_{feas}$ such that $F_1(C_i^m,\mathbf{P}') < F_1(C_i^m, \boldsymbol{P})$ where $F_2(C_i^m, \boldsymbol{P}) = K$ and $F_2(C_i^m,\mathbf{P}') = K\}$.*

We say that a cycle dominates another cycle by comparing the manufacturing costs incurred by these cycles. In order to decide which cycle dominates the other one, we compare $F_1(C_i^m, \tilde{\mathbf{P}})$ with $F_1(C_j^m, \hat{\mathbf{P}})$, for all $\tilde{\mathbf{P}} \in \mathbf{P}^*(C_i^m|K)$ and for all $\hat{\mathbf{P}} \in \mathbf{P}^*(C_j^m|K)$, for the same cycle time level K.

Definition 3.5. *We say that a cycle C_i^m dominates another cycle C_j^m for a given cycle time level K, if there is no $\hat{\mathbf{P}} \in \mathbf{P}^*(C_j^m|K)$ such that $F_1(C_j^m, \hat{\mathbf{P}}) < F_1(C_i^m, \tilde{\mathbf{P}})$ for all $\tilde{\mathbf{P}} \in \mathbf{P}^*(C_i^m|K)$, where $F_2(C_j^m, \hat{\mathbf{P}}) = K$ and $F_2(C_i^m, \tilde{\mathbf{P}}) = K$.*

In the next chapter, the efficient set of processing time vectors such that no other processing time vector gives both a smaller cycle time and a smaller manufacturing cost is presented. After that, it is proved that the proposed pure cycles in this study dominate the rest of pure cycles in the specified cycle time regions.

Chapter 4

Bicriteria Scheduling in In-Line Robotic Cells

The cell considered in this chapter is an m-machine in-line robotic cell consisting of a single gripper robot and identical CNC machines. In this chapter, we focus on two problems solved in two sections. In the first section, the problem is finding the robot move sequences and processing times on machines minimizing both cycle time and total manufacturing costs simultaneously. The minimizing cycle time and minimizing total manufacturing cost objectives are fundamental objectives studied in the scheduling literature. We propose that the robot move sequences C_1^m and C_2^m are efficient pure cycles according to the bicriteria objective of minimizing both cycle time and total manufacturing cost simultaneously. In the second section of this chapter, as a design problem, the optimum number of machines in the cell are determined for pure cycles C_1^m and C_2^m.

4.1 Bicriteria Analysis of C_1^m and C_2^m

In this section, the problem of determining the pure cycles and the corresponding cycle time regions where these cycles result in minimum cycle time and minimum total manufacturing cost is determined. We propose two pure cycles which are proved to result in minimum cycle time for fixed processing time in most of the processing time region by Gultekin et al. [12]. Firstly, the problem is defined and the necessary definitions are presented. Afterwards, the steps of solution method are presented. The cycle times of proposed cycles are determined when there is a given processing time vector. Then, the lower bound of cycle time for pure cycles is found when the number of machines and the processing time vector are given. The nondominated solutions of proposed pure cycles and the upper bound of processing time vectors are compared in order to prove that the proposed cycles result in minimum total manufacturing cost.

4.1.1 Problem Definition

In this problem, there is an in-line robotic cell consisting of m-machines and a robot performing handling operations. The in-line robotic cell is depicted in Figure 4.1. The problem is finding the processing times of the parts on machines that not only minimize the cycle time, but also simultaneously minimize the total manufacturing cost. We consider cyclic scheduling as most of the studies in robotic cell literature do, and we focus on pure cycles. The definitions and assumptions presented in the previous chapter are used in this section. An additional definition used in this section is presented as follows:
δ : Time taken by the robot to travel between two consecutive machines which is additive. Hence, the travelling time from machine i to machine j is equal to $|i-j|\delta$.

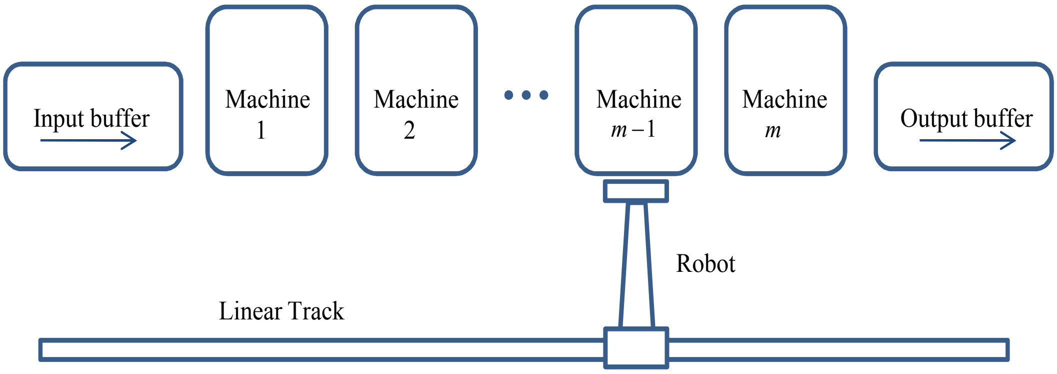

Figure 4.1: m-Machine In-Line Robotic Cell

4.1.2 Solution Procedure

In this section, the solution method of the bicriteria problem considered in this study is presented. First, the cycle times of proposed pure cycles C_1^m and C_2^m which are defined in Definitions 4.1 and 4.2 are determined when a processing time vector is given. After that, the lower bound of cycle time for a given processing time and number of machines is determined. Then, the processing time vector which results in the lower bound of total manufacturing cost for a given cycle time level is determined. After that, the nondominated solutions of C_1^m and C_2^m for a given cycle time level are determined. For each cycle time level, the nondominated solutions of C_1^m and C_2^m are compared with the processing time vector resulting in minimum total manufacturing cost for that cycle time level. It is observed that either C_1^m or C_2^m results in the processing time vector which minimizes total manufacturing cost for the specified cycle time regions. So, it is proved that either C_1^m or C_2^m dominates the rest of pure cycles in the described cycle time regions according to bicriteria objective of minimizing both cycle time and total manufacturing cost simultaneously. The proposed pure cycles are C_1^m and C_2^m which are defined by Gultekin et al. [12] as follows:

Definition 4.1. *C_1^m is the robot move cycle in an m-machine robotic cell with the following activity sequence: $L_1 L_m U_{m-1} L_{m-1} U_{m-2} L_{m-2} \ldots U_2 L_2 U_1 U_m$.*

Definition 4.2. *C_2^m is the robot move cycle in an m-machine robotic cell with* *the following activity sequence:* $L_1U_mL_mU_{m-1}L_{m-1}\ldots U_2L_2U_1$.

In the initial state of the cycle C_1^m, the machines 1 and m are idle and the rest of the machines 2 to $m-1$ are already loaded with a part. In the initial state of the cycle C_2^m, only machine 1 is idle and the rest of the machines 2 to m are loaded with a part.

The controllable processing times increase the solution flexibility such that they result in at most equal cost to fixed processing times for a given cycle time level K. The following example is useful to see the contribution of controllable processing times, in order to decrease the total manufacturing cost, compared to fixed processing times for cycle C_2^m. The total manufacturing cost of cycle C_2^m with controllable processing times, which is studied in this study, is compared to the total manufacturing cost of C_2^m in Gultekin et al. [12], where the processing times on machines are assumed to be fixed and same for all machines. In this example, we refer to some lemmas described in the further parts of this study.

Example 4.1 There is a 3-machine robotic cell. We will show that the cycle C_2^3 with controllable processing times results in less cost than cycle C_2^3 with fixed processing times for the same cycle time level. Let $\epsilon = 0.2$, $\delta = 0.1$, $P^L = 2.0$, $P^U = 4.0$. Now, we will compare the processing time vector obtained from controllable processing times with the processing time vector obtained from fixed processing times, for C_2^3.

The processing times are fixed and equivalent in the study of Gultekin et al. [12]. Let us take fixed processing time as $P = P_L = 2.0$, for all machines. Now we can state the processing time vector with fixed processing times as $\mathbf{P}_{fixed}(C_2^3) = (2.0, 2.0, 2.0)$. The cycle time of C_2^m is denoted by the following equation in Gultekin et al. [12]:

$T_{C_2^m} = 4m\epsilon + 2((m+1)^2 - 2)\delta + max\{0, P - ((4m-4)\epsilon + 2(m-1)(m+2)\delta)\}$

For the given parameters the cycle time of C_2^3 with fixed processing time $P = 2.0$ is calculated as:

$T_{C_2^3} = 12\epsilon + 28\delta + max\{0, 2.0 - (8\epsilon + 20\delta)\} = 5.2 = K$

For this cycle time level $K = 5.2$, the nondominated processing time vector giving the minimum total manufacturing cost for cycle C_2^3 is found by using Lemma 4.5. The nondominated processing time vector $(P_1^*, P_2^*, P_3^*) \in \mathbf{P}^*(C_2^3|5.2)$ is defined as follows:

$$\mathbf{P}^*(C_2^3|5.2) = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \end{bmatrix} = \begin{bmatrix} min\{P^U, K - (4\epsilon + (2m+2)\delta)\} \\ min\{P^U, K - (4\epsilon + (2m+2)\delta)\} \\ min\{P^U, K - (4\epsilon + (2m+2)\delta)\} \end{bmatrix} = \begin{bmatrix} min\{4.0, 3.6\} \\ min\{4.0, 3.6\} \\ min\{4.0, 3.6\} \end{bmatrix}$$

This simply leads to,

$$\mathbf{P}^*(C_2^3|5.2) = \begin{bmatrix} 3.6 \\ 3.6 \\ 3.6 \end{bmatrix}$$

Now we can compare the processing time vectors for these two cases as:

$$\mathbf{P}_{fixed}(C_2^3) = \begin{bmatrix} P \\ P \\ P \end{bmatrix} = \begin{bmatrix} 2.0 \\ 2.0 \\ 2.0 \end{bmatrix} < \begin{bmatrix} 3.6 \\ 3.6 \\ 3.6 \end{bmatrix} = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \end{bmatrix} = \mathbf{P}^*(C_2^3|5.2)$$

Since the nondominated processing time vector of cycle C_2^3 with controllable processing times is greater than the processing time vector with fixed processing times, the cycle C_2^3 with controllable processing times results in less total manufacturing cost.

From now on, we find the cycle times, in Lemmas 4.1 and 4.2, and the set of nondominated points obtained from these two cycles, in Lemmas 4.4 and 4.5, respectively for cycle C_1^m and C_2^m. Finally, the performances of these two prominent cycles are compared to the other pure cycles, in Theorems 4.2 and 4.3. Hence, the sufficient conditions under which one of these two cycles dominates the rest of pure cycles are found.

In the following lemma, the cycle time of the first pure cycle C_1^m is determined. When there is a given processing time vector, Lemma 4.1 determines the corresponding cycle time obtained from cycle C_1^m. Conversely, for a specified cycle time level, the highest processing times on machines that do not violate this cycle time level can be found. Since our aim is to determine the processing times giving minimum manufacturing cost and since cost decreases as processing time increases, this lemma is useful in finding the efficient set of solutions for C_1^m.

Lemma 4.1. *The cycle time of C_1^m for a given processing time vector is found as follows:*
$T_{C_1^m} = 4m\epsilon + 2m(m+1)\delta + max\{0, P_1 - ((4m-6)\epsilon + 2(m^2-2)\delta), P_m - ((4m-6)\epsilon + 2(m^2-2)\delta), P_{k_{max}} - ((4m-4)\epsilon + 2(m^2-1)\delta)\}$
$k_{max} = argmax\{P_i : i \in [2, \ldots, m-1]\}$.

Proof. Gultekin et al. [12] defined the cycle time of C_1^m as the total time required for all of the robot activities and the waiting times in front of the machines and denoted the cycle time as follows:

$$T_{C_1^m} = 4m\epsilon + (2m^2 + 2m)\delta + w_1 + w_2 + \ldots + w_m \tag{4.1}$$

For an m-machine cell, the robot travel time between consecutive machines (δ), the load/unload time of machines (ϵ), and the number of machines (m) are constant. Thus, we only have to find the total waiting times in front of the machines to calculate the cycle time. The time between loading machine i and the arrival time of the robot in front of the machine i to unload it is denoted as v_i. If the processing time on machine i, P_i, exceeds v_i, then the waiting time is the difference between P_i and v_i. Otherwise, the process on the machine is already finished when the robot comes to machine i to unload it. Hence, the waiting time of machine i is defined as $w_i = \{0, P_i - v_i\}$. We borrow the below definitions of v_i from Gultekin et al. [12].
$v_1 = T_{C_1^m} - (6\epsilon + (2m+4)\delta + w_1 + w_m) = (4m-6)\epsilon + (2m^2-4)\delta + w_2 + \ldots + w_{m-1}$.
$v_m = T_{C_1^m} - (6\epsilon + (2m+4)\delta + w_m) = (4m-6)\epsilon + (2m^2-4)\delta + w_1 + w_2 + \ldots + w_{m-1}$.
In addition, the v_i definition for $i \in [2, \ldots, m-1]$ is presented as:
$v_i = T_{C_1^m} - (4\epsilon + (2m+2)\delta + w_i) = (4m-4)\epsilon + (2m^2-2)\delta + w_1 + \ldots + w_m - w_i$.
If there is no waiting time on none of the machines, $w_1 + w_2 + \ldots + w_m = 0$.
If there is waiting time on machine 1, then:
$w_1 + w_2 + \ldots + w_m = P_1 - v_1 + \sum_{j \neq 1} w_j = P_1 - (4m-6)\epsilon - (2m^2-4)\delta + w_m$.
If there is waiting time on machine i where $i \in [2, \ldots, m-1]$, then:
$w_1 + w_2 + \ldots + w_m = P_i - v_i + \sum_{j \neq i} w_j = P_i - (4m-4)\epsilon - (2m^2-2)\delta$.
If there is waiting time on machine m, then:

$w_1 + w_2 + \ldots + w_m = P_m - v_m + \sum_{j \neq m} w_j = P_m - (4m-6)\epsilon - (2m^2-4)\delta$.
There are four different cases for the total waiting times and the sufficient conditions for these cases are determined as follows:

1. If $P_i \leq v_i$, $\forall i \in [1, \ldots, m]$, then $w_i = 0$, for $i = 1, \ldots, m$.

2. Else if $P_{k_{max}} > v_{k_{max}}$, then $w_{k_{max}} = P_{k_{max}} - v_{k_{max}} = P_{k_{max}} - (4m-4)\epsilon - (2m^2-2)\delta - \sum_{i \neq k_{max}} w_i$. Hence, $w_1 + w_2 + \ldots + w_m = P_{k_{max}} - (4m-4)\epsilon - (2m^2-2)\delta$.

3. Else if $P_m > v_m$, then $w_m = P_m - v_m = P_m - (4m-6)\epsilon - (2m^2-4)\delta - \sum_{i \neq m} w_i$. Hence, $w_1 + w_2 + \ldots + w_m = P_m - (4m-6)\epsilon - (2m^2-4)\delta$.

4. Else if only $P_1 > v_1$, then $w_m = 0$ and $w_1 = P_1 - v_1 = P_1 - (4m-6)\epsilon - (2m^2-4)\delta - \sum_{i \neq 1,m} w_i$. Hence, $w_1 + w_2 + \ldots + w_m = P_1 - (4m-6)\epsilon - (2m^2-4)\delta$.

As a consequence, total waiting time is calculated as:
$w_1 + w_2 + \ldots + w_m = max\{0, P_1 - ((4m-6)\epsilon + (2m^2-4)\delta), P_m - ((4m-6)\epsilon + (2m^2-4)\delta), P_{k_{max}} - ((4m-4)\epsilon + (2m^2-2)\delta)\}$.
The cycle time of C_1^m is obtained by replacing the total waiting time in the equation (4.1) with this max function. □

In Lemma 4.2, the cycle time of C_2^m is determined. When there is a given processing time vector, Lemma 4.2 gives the corresponding cycle time. Similar to Lemma 4.1, Lemma 4.2 can be used to determine the largest feasible processing times for a given cycle time level.

Lemma 4.2. *The cycle time of C_2^m for a given processing time vector is found as follows:*
$T_{C_2^m} = 4m\epsilon + 2((m+1)^2-2)\delta + max\{0, P_{k_{max}} - ((4m-4)\epsilon + 2(m-1)(m+2)\delta)\}$,
$k_{max} = argmax\{P_i : i \in [1, \ldots, m]\}$.

Proof. Gultekin et al. [12] defined the cycle time of the second pure cycle C_2^m as follows:

$$T_{C_2^m} = 4m\epsilon + (2m^2 + 4m - 2)\delta + w_1 + w_2 + \ldots + w_m. \tag{4.2}$$

The total waiting time have to be found in order to calculate the cycle time of C_2^m corresponding to a processing time. The waiting time on machine i is defined as $w_i = \{0, P_i - v_i\}$. The definition of v_i for all machines is identical for C_2^m and it is presented by Gultekin et al. [12] as:
$v_i = T_{C_2^m} - (4\epsilon + (2m+2)\delta + w_i) = (4m-4)\epsilon + 2(m-1)(m+2)\delta + w_1 + \ldots + w_m - w_i$.
If there is no waiting time on none of the machines, then $w_1 + w_2 + \ldots + w_m = 0$.
If there is some waiting time on some machine i, then:
$w_1 + w_2 + \ldots + w_m = P_i - v_i + \sum_{j \neq i} w_j = P_i - (4m-4)\epsilon - 2(m-1)(m+2)\delta$, $\forall i$.
There are two different total waiting time results and the sufficient conditions for these cases are determined as follows:

1. If $P_i \leq v_i$ for $\forall i \in [1, \ldots, m]$, then $w_i = 0$, for $i = 1, \ldots, m$
2. Else if $P_{k_{max}} > v_{k_{max}}$, then $w_{k_{max}} = P_{k_{max}} - v_{k_{max}} = P_{k_{max}} - (4m-4)\epsilon - 2(m-1)(m+2)\delta - \sum_{i \neq k_{max}} w_i$. Hence, the total waiting time is as follows: $w_1 + w_2 + \ldots + w_m = P_{k_{max}} - ((4m-4)\epsilon + 2(m-1)(m+2)\delta)$.

So, $w_1 + w_2 + \ldots + w_m = max\{0, P_{k_{max}} - (4m-4)\epsilon - 2(m-1)(m+2)\delta\}$ and the cycle time is obtained by replacing the total waiting time in the equation (4.2) with this max function. □

In the next theorem, the lower bound for the cycle time of pure cycles with controllable processing times is defined. The lower bound of cycle time for pure cycles is determined by using Theorem 4.1, when a processing time vector is given.

Theorem 4.1. *For an m-machine robotic cell with controllable processing times, the cycle time of any pure cycle is no less than*

$$\underline{T_{contr}} = max\{4m\epsilon + 2m(m+1)\delta, 4\epsilon + (2m+2)\delta + max\{P_i, i : 1, \ldots, m\}\}. \quad (4.3)$$

Proof. From the definition of pure cycles, we determine that the cycle time of a pure cycle has to be greater than or equal to two lower bounds. The first lower bound is obtained from the exact robot activity time and the second one is obtained from the given processing time vector. Since the robot has to perform an exact set of robot activities, the total time required for these activities constitutes a lower bound. Thus, the first lower bound is obtained as follows: The set of robot activities can be analyzed in two groups, the first group consists of robot loading and unloading times. First, a part is taken from the input buffer (ϵ) then loaded to one of the machines (ϵ) after the processing on the machine is finished, the part is unloaded (ϵ) and dropped to the output buffer (ϵ). This makes a total of $4m\epsilon$ time units for a cycle. The robot travel times constitute the second group of robot activities. For any part, the robot takes the part from input buffer to output buffer ($(m+1)\delta$). Then, the robot travels from the output buffer to input buffer to take a new part or to complete the cycle ($(m+1)\delta$). This makes a total of $2m(m+1)\delta$ time units for a cycle. Consequently, the first lower bound, which is the total time required to complete the set of robot activities, makes a total of $4m\epsilon$+$2m(m+1)\delta$.

The second lower bound is the minimum time between two consecutive loadings of any machine. The minimum time needed to unload machine i after loading it is P_i time units. After the processing on the part is finished, it is unloaded (ϵ), the part is transferred to output buffer ($(m+1-i)\delta$), and the part is dropped (ϵ). After that, the robot travels to the input buffer to take a new part to make the consecutive loading of machine i ($(m+1)\delta$), takes a new part part, (ϵ), brings the new part to machine i ($i\delta$) and finally loads the machine (ϵ). Hence, the minimum time required between two consecutive loadings of machine i is $4\epsilon+(2m+2)\delta+P_i$. However there are m machines and the processing times on these machines may be different from each other, due to controllability. Thus, the cycle time has to be at least equal to the minimum time required between two consecutive loadings of any machine in the cell. So, the second lower bound of the cycle time is $4\epsilon + (2m+2)\delta + max\{P_i, i : 1, \ldots, m\}$. $\square$

Our aim is to determine the processing time vector providing the minimum cost for a given cycle time level. The total manufacturing cost depends only on the processing times on machines. The manufacturing cost of a machine decreases as the processing time increases on that machine. A feasible processing time must satisfy $P^L \leq P_i \leq P^U$. Since P^L constrains processing time from below and our aim is to determine the largest feasible processing time vector, it is not necessary to involve P^L in analysis as a constraint. A processing time vector is composed of processing times on machines constrained by two bounds. For any pure cycle, the processing time on any machine is bounded above by the processing time upper bound P^U. In addition, from Theorem 4.1, processing times are bounded by cycle time level K. Consequently, for a given cycle time level, we can find the upper bounds of processing times for pure cycles that do not violate this cycle time level. After obtaining these two bounds for processing times on all machines, the upper bound of processing time vectors is determined for pure cycles. In other words, any pure cycle cannot have a greater processing time vector than the proposed processing time vector of the next lemma. Since this processing time vector is an upper bound for the processing time vectors obtained from pure cycles for a cycle time level K, it also results in the lower bound of total manufacturing cost that a pure cycle can result in. Let $\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \ldots, \overline{P}_m(K))$ denote the upper bound of processing time vectors. Now, $\overline{\mathbf{P}}(K)$ for a given cycle time level K is found as follows:

Lemma 4.3. *For a given cycle time level K, the upper bound of processing time vectors for pure cycles is represented as follows:*

$$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \ldots, \overline{P}_m(K)),\ \textit{where}\ \overline{P}_i(K) = min\{P^U, K-(4\epsilon+(2m+2)\delta)\}, \forall i.$$

Proof. The two bounds constraining processing time vectors are found in the following cases.

1. The processing time on any machine is less than P^U. This leads to: $\overline{P}_i(K) \leq P^U$, $\forall i$.

2. In addition, the processing times on the machines cannot exceed a specific value, since otherwise the cycle time level K will be exceeded. Now, we find the upper bound of processing time on machine i, P_i, for the cycle time level K. Theorem 4.1 determines the lower bound for the cycle time, when a processing time vector is given. The cycle time lower bound in Theorem 4.1 is presented as:

   $\underline{T_{contr}} = max\{4m\epsilon + 2m(m+1)\delta, 4\epsilon + (2m+2)\delta + max\{P_i, i : 1, \ldots, m\}\}.$

Let us set the cycle time K, then cycle time is at least equal to the cycle time lower bound :

$\underline{T_{contr}} = max\{4m\epsilon+2m(m+1)\delta, 4\epsilon+(2m+2)\delta+max\{P_i, i : 1, \ldots, m\}\} \leq K$

Now, the processing time upper bound for machine i is found as follows:

$max\{P_i, i : 1, \ldots, m\} \leq K-(4\epsilon+(2m+2)\delta)$, then $P_i \leq K-(4\epsilon+(2m+2)\delta$, $\forall i$. This implies that $\overline{P}_i(K) \leq K - (4\epsilon + (2m + 2)\delta$, $\forall i$.

Hence, the $\overline{\mathbf{P}}(K)$ is upper bound of processing time vectors satisfying the two bounds described above for a given cycle time level. □

In order to compare the total manufacturing costs obtained from cycles C_1^m and C_2^m to the remaining pure cycles, we construct Lemmas 4.4 and 4.5, respectively, to determine the processing times that give minimum total manufacturing cost for a given cycle time level K. In order to decrease the total manufacturing cost, the processing times have to take their maximum value without exceeding bounds. There are two bounds constraining the processing times: the first constraint is the processing time upper bound (P^U). The second constraint is the cycle time (K) constraint. For a given cycle time level K, the processing times are bounded such that the resulting cycle time value must not exceed this cycle time level. In the following lemma, the set of nondominated processing time vectors for cycle C_1^m, which is denoted as $\mathbf{P}^*(C_1^m|K)$ is determined. The nondominated solutions in $\mathbf{P}^*(C_1^m|K)$ are the optimum solutions of corresponding ECPs pertaining to cycle time level K. Let $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$, be a nondominated processing time vector. We can see from Lemma 4.1, the cycle C_1^m is feasible when the cycle time is $4m\epsilon + 2m(m+1)\delta \leq K$, and hence we consider this region in the next lemma.

Lemma 4.4. *Given any feasible cycle time level K, the nondominated processing time vector $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$ is defined as:*

$$\mathbf{P}^*(C_1^m|K) = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} = \begin{bmatrix} min\{P^U, K - (6\epsilon + (2m+4)\delta)\} \\ min\{P^U, K - (4\epsilon + (2m+2)\delta)\} \\ \vdots \\ min\{P^U, K - (4\epsilon + (2m+2)\delta)\} \\ min\{P^U, K - (6\epsilon + (2m+4)\delta)\} \end{bmatrix}$$

Proof. The two upper bounds for processing times are presented as follows:

1. Any feasible processing time is at most equal to P^U. Since $(P_1^*, P_2^*, \ldots, P_m^*)$ is also a feasible solution, then $P_i^* \leq P^U$, $\forall i$.

2. In addition, the processing times on machines cannot exceed a specific amount, since the cycle time is bounded by level K. In Lemma 4.1, it can be seen that cycle time depends on the processing times. Conversely, the cycle time level K constrains the processing times so that they do not exceed an upper bound. Now, we will find the processing times on machine 1 and m, e.g. P_1 and P_m, respectively, for the cycle time level K. The processing time upper bounds for cycle time level K are found by using Lemma 4.1 as follows:

   $K = 4m\epsilon + 2m(m+1)\delta + max\{0, P_1 - ((4m-6)\epsilon + 2(m^2-2)\delta), P_m - ((4m-6)\epsilon + 2(m^2-2)\delta), P_{k_{max}} - ((4m-4)\epsilon + 2(m^2-1)\delta)\}$, where $k_{max} = argmax\{P_i : i \in [2, \ldots, m-1]\}$.

   We find that $P_1 \leq K - (6\epsilon + (2m+4)\delta)$. Since P_1^* is also a feasible solution, it has to satisfy this condition as well, hence $P_1^* \leq K - (6\epsilon + (2m+4)\delta)$. Similarly, $P_m \leq K - (6\epsilon + (2m+4)\delta)$, thus $P_m^* \leq K - (6\epsilon + (2m+4)\delta)$.

   In addition, we find that $P_{k_{max}} \leq K - (4\epsilon + (2m+2)\delta)$, since $P_i \leq P_{k_{max}}$ for $i \in [2, \ldots, m-1]$, then $P_i \leq K - (4\epsilon + (2m+2)\delta)$. This simply leads to $P_i^* \leq K - (4\epsilon + (2m+2)\delta)$ for $i \in [2, \ldots, m-1]$.

   As a result, the processing time bounds pertaining to cycle time level K are presented as follows:

$$\begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} \leq \begin{bmatrix} K-(6\epsilon+(2m+4)\delta) \\ K-(4\epsilon+(2m+2)\delta) \\ \vdots \\ K-(4\epsilon+(2m+2)\delta) \\ K-(6\epsilon+(2m+4)\delta) \end{bmatrix}$$

The processing times on their maximum values, without violating the bounds found in the first and second arguments, compose the nondominated processing time vectors stated in Lemma 4.4. □

The next example is useful to understand the contribution of controllable processing times in order to decrease the total manufacturing cost of cycle C_1^m. Furthermore, we can see that the total manufacturing cost of nondominated solution of C_1^m with controllability is less than the total manufacturing cost of nondominated solution of C_1^m with fixed processing times.

Example 4.2 In this example, we will show that the cycle C_1^4 with controllable processing times results in less cost than C_1^4 with fixed processing times, for a 4-machine robotic cell. Let $\epsilon = 0.2$, $\delta = 0.1$, $P^U = 6.5$ and assume that the cycle time level is $K = 8.0$. By using Lemma 4.4, for cycle time level K, the nondominated processing time vector, $(P_1^*, P_2^*, P_3^*, P_4^*) \in \mathbf{P}^*(C_1^4|8.0)$, giving the minimum total manufacturing cost is found as follows:

$$\mathbf{P}^*(C_1^4|8.0) = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \\ P_4^* \end{bmatrix} = \begin{bmatrix} min\{P^U, K-(6\epsilon+(2m+4)\delta)\} \\ min\{P^U, K-(4\epsilon+(2m+2)\delta)\} \\ min\{P^U, K-(4\epsilon+(2m+2)\delta)\} \\ min\{P^U, K-(6\epsilon+(2m+4)\delta)\} \end{bmatrix} = \begin{bmatrix} min\{6.5, 5.6\} \\ min\{6.5, 6.2\} \\ min\{6.5, 6.2\} \\ min\{6.5, 5.6\} \end{bmatrix}$$

This simply leads to:

$$\mathbf{P}^*(C_1^4|8.0) = \begin{bmatrix} 5.6 \\ 6.2 \\ 6.2 \\ 5.6 \end{bmatrix}$$

Now, we will calculate the nondominated processing time vector for fixed processing times in the study of Gultekin et al. [12]. The cycle time corresponding to a fixed processing time P, on each machine, is found as:

$$T_{C_1^m} = 4m\epsilon + 2m(m+1)\delta + max\{0, P - ((4m-6)\epsilon + 2(m^2-2)\delta)\}$$

After a simple calculation we obtain $P \leq T_{C_1^m} - 6\epsilon - (2m+4)\delta$.

Our cycle time level is K, we can replace $T_{C_1^m}$ with K and the following result is obtained:

$P \leq K - 6\epsilon - (2m+4)\delta$. For the given set of data in this example, $P \leq 5.6$ for all machines. Now we can state the nondominated processing time vector with fixed processing times as $\mathbf{P}^*_{fixed}(C_1^4|8.0) = (5.6, 5.6, 5.6, 5.6)$.

The nondominated processing time vectors obtained from these two cases are compared as follows:

$$\mathbf{P}^*_{fixed}(C_1^4|8.0) = \begin{bmatrix} P \\ P \\ P \\ P \end{bmatrix} = \begin{bmatrix} 5.6 \\ 5.6 \\ 5.6 \\ 5.6 \end{bmatrix} \leq \begin{bmatrix} 5.6 \\ 6.2 \\ 6.2 \\ 5.6 \end{bmatrix} = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \\ P_4^* \end{bmatrix} = \mathbf{P}^*(C_1^4|8.0)$$

By comparing the processing times of $\mathbf{P}^*_{fixed}(C_1^4|8.0)$ and $\mathbf{P}^*(C_1^4|8.0)$, we see that $P = P_1^* = P_4^*$ and $P < P_2^* = P_3^*$. Thus, the nondominated processing time vector of C_1^4 with controllable processing times results in less total manufacturing cost.

In the following lemma, we determine $\mathbf{P}^*(C_2^m|K)$, the set of nondominated processing time vectors for cycle C_2^m that simultaneously minimize the cycle time and the total manufacturing cost. It can be seen from Lemma 4.2 that the cycle C_2^m is feasible when cycle time is $4m\epsilon + 2((m+1)^2 - 2)\delta \leq K$, thus the ECP problem is solved for cycle time level K in this boundary to construct the efficient frontier.

Lemma 4.5. *Given any feasible cycle time level K, the nondominated processing time vector $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_2^m|K)$ is defined as follows:*

$$\mathbf{P}^*(C_2^m|K) = \begin{bmatrix} P_1^* \\ \vdots \\ P_m^* \end{bmatrix} = \begin{bmatrix} min\{P^U, K-(4\epsilon+(2m+2)\delta)\} \\ \vdots \\ min\{P^U, K-(4\epsilon+(2m+2)\delta)\} \end{bmatrix}$$

Proof. For a given cycle time level K, a feasible processing time vector is composed of processing times on machines that satisfy two upper bounds.

1. The processing times of a feasible processing time vector has to be at most equal to P^U. Since $(P_1^*, P_2^*, \ldots, P_m^*)$ is also a feasible processing time vector, then $P_i^* \leq P^U$, $\forall i$.

2. In addition, the processing times are bounded to satisfy the cycle time level K. Now, we will determine the processing time on machine i, P_i, for the cycle time K. By using Lemma 4.2 the cycle time level K can be presented as follows:

   $K = 4m\epsilon+2((m+1)^2-2)\delta+max\{0, P_{k_{max}}-((4m-4)\epsilon+2(m-1)(m+2)\delta)\}$,
   $k_{max} = argmax\{P_i : i \in [1, \ldots, m]\}$.

   This leads to $P_{k_{max}} \leq K-(4\epsilon+(2m+2)\delta)$. Since $P_i \leq P_{k_{max}}$, it implies that $P_i \leq K-(4\epsilon+(2m+2)\delta)$. Since P_i^* is feasible, $P_i^* \leq K-(4\epsilon+(2m+2)\delta)$, $\forall i$.

   The processing time bounds pertaining to the cycle time level K are presented as follows:

$$\begin{bmatrix} P_1^* \\ \vdots \\ P_m^* \end{bmatrix} \leq \begin{bmatrix} K-(4\epsilon+(2m+2)\delta) \\ \vdots \\ K-(4\epsilon+(2m+2)\delta) \end{bmatrix}$$

The processing times on their maximum values, without violating the bounds found in the first and second arguments, compose the nondominated processing time vectors stated in Lemma 4.5. □

In the next two theorems, Theorems 4.2 and 4.3, we prove that the two prominent pure cycles, C_1^m and C_2^m, dominate the rest of pure cycles in the specified regions. The feasible cycle time region of pure cycles obtained from Theorem 4.1

is $4m\epsilon+2m(m+1)\delta \leq K$. We analyze this cycle time region in two parts. The first region is where the cycle C_2^m is feasible and the second region is where the cycle C_2^m is not feasible but C_1^m is feasible. We can see from Lemma 4.2 that the cycle C_2^m is feasible when cycle time is $4m\epsilon + 2((m+1)^2 - 2)\delta \leq K$. In Theorem 4.2, we show that pure cycle C_2^m dominates the rest of pure cycles in this region. In addition, from Theorem 4.1, the cycle time lower bound that can be attained from pure cycles is $4m\epsilon + 2m(m+1)\delta$. We can see that the only region that C_2^m is not feasible is the cycle time region $4m\epsilon+2m(m+1)\delta \leq K < 4m\epsilon+2((m+1)^2-2)\delta$. In Theorem 4.3, we show that C_1^m dominates the rest of pure cycles under specified conditions, in this region. So, it is seen that the proposed pure cycles C_1^m and C_2^m dominate rest of the pure cycles in most of the regions. In the following theorem, we prove that the second pure cycle C_2^m dominates the rest of the pure cycles in the described region. We see that the cycle C_2^m results in the lower bound of total manufacturing cost in the specified region. Since the lower bound of total manufacturing cost corresponds to the upper bound of processing time vectors, we show that C_2^m results in the upper bound of processing time vectors, for the cycle time level K. From Lemma 4.2, C_2^m is feasible when cycle time is $4m\epsilon + 2((m+1)^2 - 2)\delta \leq K$, thus we consider this cycle time region in the following theorem.

Theorem 4.2. *Whenever C_2^m is feasible, it dominates all other pure cycles.*

There are two possible cases that may arise according to processing time upper bound P^U.

1. The first case considers that $P^U \leq K - (4\epsilon + (2m+2)\delta)$. The nondominated processing time vector for cycle C_2^m is determined by using Lemma 4.5. Take any nondominated solution $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_2^m|K)$. The processing time of machine i, P_i^* is found by using Lemma 4.5 such that $P_i^* = min\{P^U, K - (4\epsilon + (2m+2)\delta)\}$. Since we assumed that $P^U \leq K - (4\epsilon + (2m+2)\delta)$, then $P_i^* = P^U$. We can present the nondominated processing time vector of C_2^m as follows:

   $(P_1^*, P_2^*, \ldots, P_m^*) = (P^U, P^U, \ldots, P^U)$

The upper bound of processing time vectors found by using Lemma 4.3 is presented as:

$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \overline{P}_2(K), \ldots, \overline{P}_m(K)) = (P^U, P^U, \ldots, P^U)$.

The processing time vector obtained from C_2^m is equal to the upper bound of processing time vectors for cycle time level K, $\mathbf{P}^*(C_2^m|K) = \overline{\mathbf{P}}(K)$, Since all processing times take the maximum value P^U, there is not any other pure cycle that can result in smaller total manufacturing cost.

2. The second possible case for the processing time upper bound is: $P^U > K - (4\epsilon + (2m+2)\delta)$.

The nondominated processing time vector of C_2^m is found from Lemma 4.5 as follows:

$$\mathbf{P}^*(C_2^m|K) = \begin{bmatrix} P_1^* \\ \vdots \\ P_m^* \end{bmatrix} = \begin{bmatrix} K - (4\epsilon + (2m+2)\delta) \\ \vdots \\ K - (4\epsilon + (2m+2)\delta) \end{bmatrix}$$

The upper bound of processing time vectors found by using Lemma 4.3 is presented as follows:

$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \ldots, \overline{P}_m(K))$ where $\overline{P}_i(K) = K - (4\epsilon + (2m+2)\delta)$, $\forall i$.

The processing time vector obtained from the cycle C_2^m is equal to the upper bound of processing time vectors for this cycle time level, $\mathbf{P}^*(C_2^m|K) = \overline{\mathbf{P}}(K)$, thus there is not any pure cycle that can result in less total manufacturing cost. □

In the next theorem, we prove that C_1^m dominates the rest of the pure cycles in the described region, under the specified condition. Since the lower bound of total manufacturing cost corresponds to the upper bound of processing time vectors, we show that C_1^m results in the upper bound of processing time vectors. C_1^m is found to be feasible when cycle time is $4m\epsilon + 2m(m+1)\delta \leq K$ by using Lemma 4.1. In addition, in Theorem 4.2, we show that C_2^m dominates all of the other pure cycles when cycle time is $4m\epsilon + 2((m+1)^2 - 2)\delta \leq K$. Now, we will show that C_1^m dominates the rest of the pure cycles in the remaining cycle time region; $4m\epsilon + 2m(m+1)\delta \leq K < 4m\epsilon + 2((m+1)^2 - 2)\delta$ under the specified condition of P^U.

Theorem 4.3. *For the remaining region* $4m\epsilon+2m(m+1)\delta \leq K < 4m\epsilon+2((m+1)^2-2)\delta$, C_1^m *dominates the rest of pure cycles if* $P^U \leq K-(6\epsilon+(2m+4)\delta)$.

Proof. In this case, we consider that $P^U \leq K-(6\epsilon+(2m+4)\delta)$. The nondominated processing time vector for cycle C_1^m is found by Lemma 4.4. Take any nondominated processing time vector $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$. We find the processing time of machine 1, P_1^*, as an example. From Lemma 4.4, $P_1^* = min\{P^U, K-(6\epsilon+(2m+4)\delta)\}$ and since, $P^U \leq K-(6\epsilon+(2m+4)\delta)$ is assumed, we have $P_1^* = P^U$. Similarly, we find all of the other processing times of nondominated processing time vector of C_1^m and the following result is obtained:

$$(P_1^*, P_2^*, \ldots, P_m^*) = (P^U, P^U, \ldots, P^U)$$

In addition, from Lemma 4.3, we find the upper bound of processing time vectors as follows:

$$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \overline{P}_2(K), \ldots, \overline{P}_m(K)) = (P^U, P^U, \ldots, P^U).$$

The processing time vector obtained from C_2^m is equal to the upper bound of processing time vectors $\mathbf{P}^*(C_1^m|K) = \overline{\mathbf{P}}(K)$. Since all the processing times on the machines take the maximum value, P^U, there is not any other pure cycle that can result in less total manufacturing cost. □

In the next lemma, we compare the total manufacturing cost of pure cycle C_1^m with the lower bound of total manufacturing cost. The region considered in Lemma 4.6 is the only region where neither C_1^m nor C_2^m dominates the rest of the pure cycles. Since C_1^m is feasible and C_2^m is not feasible in this region, only C_1^m is considered. The term $F_1(C_1^m, \mathbf{P}^*(C_1^m|K))$ denotes the total manufacturing cost incurred by nondominated processing time vector of the cycle C_1^m for the given cycle time level K. Similarly, the term $F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K))$ denotes the lower bound of total manufacturing cost of pure cycles for the given cycle time level K. We present the performance analysis below in which $f(\chi)$ gives the manufacturing cost when the processing time is equal to χ.

Lemma 4.6. *For the remaining region* $4m\epsilon+2m(m+1)\delta \leq K < 4m\epsilon+2((m+1)^2-2)\delta$ *and* $K-(6\epsilon+(2m+4)\delta) < P^U$, *the performance of the cycle* C_1^m *in* *this region is stated as below:*

$F_1(C_1^m, \mathbf{P}^*(C_1^m|K)) \le F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K)).\varrho$, *where* $\varrho = (1 - \frac{2}{m} + \frac{2f\big((4m-6)\epsilon + 2(m^2-2)\delta\big)}{mf\big((4m-4)\epsilon + 2(m^2-1)\delta\big)})$.

Proof. We analyze the performance in two regions according to the level of processing time upper bound P^U.

1. First, we consider the case where :
$K - (6\epsilon + (2m+4)\delta) < P^U \le K - (4\epsilon + (2m+2)\delta)$.
Take any nondominated processing time vector $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$. Now we will determine the processing time on machine 1, P_1^*, as an example. Lemma 4.4 implies that $P_1^* = min\{P^U, K - (6\epsilon + (2m+4)\delta)\}$. Since we assumed that $K - (6\epsilon + (2m+4)\delta) < P^U$, the processing time on this machine is $P_1^* = K - (6\epsilon + (2m+4)\delta)$. Similarly, all processing times of $\mathbf{P}^*(C_1^m|K)$ are found and the following result is obtained:
$$\mathbf{P}^*(C_1^m|K) = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} = \begin{bmatrix} K - (6\epsilon + (2m+4)\delta) \\ P^U \\ \vdots \\ P^U \\ K - (6\epsilon + (2m+4)\delta) \end{bmatrix}$$
We compare the total manufacturing cost of $\mathbf{P}^*(C_1^m|K)$ to the lower bound of total manufacturing cost which corresponds to the upper bound of processing time vectors for the cycle time level K and the upper bound of processing time vectors is found from Lemma 4.3 as follows:
$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \overline{P}_2(K), \ldots, \overline{P}_m(K)) = (P^U, P^U, \ldots, P^U)$.
The total manufacturing cost obtained from $\mathbf{P}^*(C_1^m|K)$ is equal to the total of manufacturing costs on machines and it is calculated as follows:
$F_1(C_1^m, \mathbf{P}^*(C_1^m|K)) = (m-2)f\big(P^U\big) + 2f\big(K - (6\epsilon + (2m+4)\delta)\big)$.
The lower bound of total manufacturing cost for cycle time level K is found as:
$F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K)) = \sum_{i=1}^m f\big(P_i\big) = mf\big(P^U\big)$.

Now, we can calculate the performance by dividing the total manufacturing cost obtained from cycle C_1^m to the lower bound of total manufacturing cost.
$\frac{F_1(C_1^m,\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(C_i^m,\overline{\mathbf{P}}(K))} = \frac{(m-2)f\big(P^U\big)+2f\big(K-(6\epsilon+(2m+4)\delta)\big)}{mf\big(P^U\big)} = 1-\frac{2}{m}+\frac{2f\big(K-(6\epsilon+(2m+4)\delta)\big)}{mf\big(P^U\big)}.$

In this case $P^U \leq K-(4\epsilon+(2m+2)\delta)$ is assumed. Hence, we have the following:
$\frac{2f\big(K-(6\epsilon+(2m+4)\delta)\big)}{mf\big(P^U\big)} \leq \frac{2f\big(K-(6\epsilon+(2m+4)\delta)\big)}{mf\big(K-(4\epsilon+(2m+2)\delta)\big)}$

We assumed that $4m\epsilon+2m(m+1)\delta \leq K$, then we can say that:
$\frac{2f\big(K-(6\epsilon+(2m+4)\delta)\big)}{mf\big(K-(4\epsilon+(2m+2)\delta)\big)} \leq \frac{2f\big((4m-6)\epsilon+2(m^2-2)\delta\big)}{mf\big((4m-4)\epsilon+2(m^2-1)\delta\big)}$, thus
$\frac{F_1(C_1^m,\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(C_i^m,\overline{\mathbf{P}}(K))} \leq 1-\frac{2}{m}+\frac{2f\big((4m-6)\epsilon+2(m^2-2)\delta\big)}{mf\big((4m-4)\epsilon+2(m^2-1)\delta\big)}.$

2. Now, we consider the second case where we assume that:
$K-(4\epsilon+(2m+2)\delta) < P^U$.

Take any nondominated processing time vector $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$. Now, we will determine the processing time on machine 2, P_2^*, as an example. From Lemma 4.4, we know that $P_2^* = min\{P^U, K-(4\epsilon+(2m+2)\delta)\}$. Since $K-(4\epsilon+(2m+2)\delta) < P^U$, the processing time on that machine is $P_2^* = K-(4\epsilon+(2m+2)\delta)$. Similarly, we find all of the other processing times of $\mathbf{P}^*(C_1^m|K)$ by using Lemma 4.4 and the following result is obtained:

$$\mathbf{P}^*(C_1^m|K) = \begin{bmatrix} P_1^* \\ P_2^* \\ \vdots \\ P_{m-1}^* \\ P_m^* \end{bmatrix} = \begin{bmatrix} K-(6\epsilon+(2m+4)\delta) \\ K-(4\epsilon+(2m+2)\delta) \\ \vdots \\ K-(4\epsilon+(2m+2)\delta) \\ K-(6\epsilon+(2m+4)\delta) \end{bmatrix}$$

We compare the total manufacturing cost of $\mathbf{P}^*(C_1^m|K)$ to the lower bound of total manufacturing cost. The upper bound of processing time vector is found from Lemma 4.3 as:
$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \ldots, \overline{P}_m(K))$,
where $\overline{P}_i(K) = min\{P^U, K-(4\epsilon+(2m+2)\delta)\}$, $\forall i$.

Now, let us compare the total manufacturing cost of $\mathbf{P}^*(C_1^m|K)$ to the lower bound of total manufacturing cost obtained from $\overline{\mathbf{P}}(K)$ for given cycle time level K. The total manufacturing cost obtained from $\mathbf{P}^*(C_1^m|K)$ is calculated as follows:
$F_1(C_1^m, \mathbf{P}^*(C_1^m|K)) = (m-2)f\big(K-(4\epsilon+(2m+2)\delta)\big)+2f\big(K-(6\epsilon+(2m+4)\delta)\big)$.
The lower bound of total manufacturing cost is found as:
$F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K)) = \sum_{i=1}^{m} f(P_i) = mf\big(K-(4\epsilon+(2m+2)\delta)\big)$.
Now, we can calculate the performance of C_1^m by dividing the total manufacturing cost obtained from cycle C_1^m to the lower bound.

$$\frac{F_1(C_1^m,\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(C_i^m,\overline{\mathbf{P}}(K))} = \frac{(m-2)f\Big(K-(4\epsilon+(2m+2)\delta)\Big)+2f\Big(K-(6\epsilon+(2m+4)\delta)\Big)}{mf\Big(K-(4\epsilon+(2m+2)\delta)\Big)}$$

$$= 1-\frac{2}{m}+\frac{2f\Big(K-(6\epsilon+(2m+4)\delta)\Big)}{mf\Big(K-(4\epsilon+(2m+2)\delta)\Big)}$$

We assumed that $4m\epsilon+2m(m+1)\delta \le K$, then we can say that:
$\frac{2f\Big(K-(6\epsilon+(2m+4)\delta)\Big)}{mf\Big(K-(4\epsilon+(2m+2)\delta)\Big)} \le \frac{2f\Big((4m-6)\epsilon+2(m^2-2)\delta\Big)}{mf\Big((4m-4)\epsilon+2(m^2-1)\delta\Big)}$, thus

$$\frac{F_1(C_1^m,\mathbf{P}^*(C_1^m|K))}{F_1^{LB}(C_i^m,\overline{\mathbf{P}}(K))} \le 1-\frac{2}{m}+\frac{2f\Big((4m-6)\epsilon+2(m^2-2)\delta\Big)}{mf\Big((4m-4)\epsilon+2(m^2-1)\delta\Big)} \qquad \square$$

In the previous lemma, the total manufacturing cost obtained from nondominated solutions of the cycle C_1^m is compared with the lower bound of total manufacturing cost obtained from Lemma 4.3. As can be seen from the statement in Lemma 4.6, the number of machines directly effects the difference in between the total manufacturing cost of C_1^m and the lower bound of total manufacturing cost that we could obtain for any pure cycle. The next example presents two cases which are useful in understanding Lemma 4.6. In this example, we consider 2-machine robotic cell case and an m-machine case where $m \to \infty$. The first case represents the highest difference rate between the lower bound and the total manufacturing cost of C_1^m. In the second case, the total manufacturing cost of C_1^m becomes equal to the lower bound. The cost function used in the next two examples is modified from the cost functions given in Kayan and Akturk [21]. The operation to be performed on identical parts is a single pass turning operation using a single cutting tool on identical CNC machines. The total manufacturing cost is the sum of manufacturing costs on each machine, thus we define the machining cost and the tooling cost on each machine. The machining cost for machine i is defined as $O.P_i$ where O is the operating cost which is identical for all machines. The tooling cost on machine i is defined as TUP_i^{α}, where $T > 0$ and $\alpha < 0$ are constants for identical tools and $U > 0$ is a specific constant for identical operations on identical tools. Consequently, the manufacturing cost for machine i is $f(P_i) = O.P_i + TUP_i^{\alpha}$.

Example 4.3 In this example, we show that C_1^m performs better as the number of machines increases. The total manufacturing cost of C_1^m gets closer to the lower bound of total manufacturing cost as the number of machines increases. Hence, we consider the case with infinite number of machines, where the total manufacturing cost of C_1^m equals to the lower bound. Similarly, the difference between the total manufacturing cost of C_1^m and the lower bound increases as the number of machines decreases. In order to demonstrate this case, we consider the two machine case. Let $\delta = 0.1$, $\epsilon = 0.1$, $\alpha = -1.2423$, $T = 1$, $U = 1$, and $O = 1$. The cost function corresponding to processing times on machines is described as $f(P_i) = O.P_i + TUP_i^{\alpha}$.

The performance measure for 2-machine case is calculated as $\varrho = 1.24$ by using Lemma 4.6. Thus, $F_1(C_1^2, \mathbf{P}^*(C_1^2|K)) \leq F_1^{LB}(C_i^2, \overline{\mathbf{P}}(K)).(1.24)$. This statement implies that C_1^m results in at most 1.24 times of the lower bound. In addition, the minimum difference in performance occurs for m-machine case where $m \to \infty$. The performance difference is measured as $\varrho = 1$ by using Lemma 4.6. Thus, $F_1(C_1^m, \mathbf{P}^*(C_1^m|K)) \leq F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K))$. Since the total manufacturing cost obtained from cycle C_1^m cannot be less than the lower bound, $F_1(C_1^m, \mathbf{P}^*(C_1^m|K)) = F_1^{LB}(C_i^m, \overline{\mathbf{P}}(K))$. Hence, we see that the cycle C_1^m performs better as the number of machines increases.

The next example presents the efficient frontiers of cycles C_1^m and C_2^m for given parameters and illustrates Theorem 4.2 and Lemma 4.6. We will see the region where C_2^m dominates the rest of pure cycles. Furthermore, C_1^m covers the region where cycle C_2^m is not feasible and in that region, the performance analysis of C_1^m is made by using Lemma 4.6.

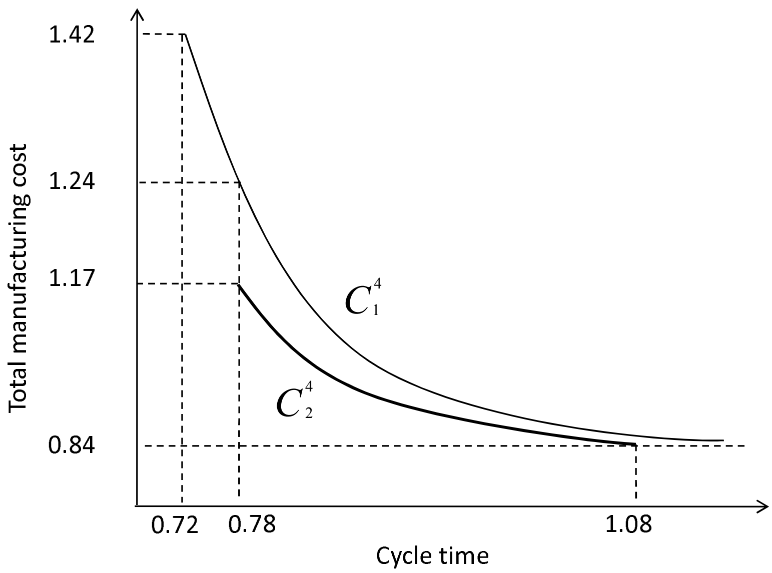

Figure 4.2: C_1^4 and C_2^4 pure cycles

Example 4.4 In this example, we consider a turning operation with machines using the same tool to produce identical parts in a 4-machine robotic cell. For this turning operation, let the parameters be given as follows: $T = 0.1$, $O = 0.1$, $U = 1$, $\alpha = -1.6423$, $\epsilon = 0.02$, $\delta = 0.01$ and $P^U = 0.90$.

The two curves in Figure 4.2 represent the efficient frontiers of C_1^4 and C_2^4, which are constructed by using Lemmas 4.4 and 4.5 respectively. In this figure, $4m\epsilon + 2((m+1)^2 - 2)\delta = 0.78$ is the point found from Lemma 4.2 where C_2^4 becomes feasible. It can be seen that C_2^4 is presented in bold line, from Theorem 4.2, C_2^4 dominates the rest of pure cycles when the cycle time is at least 0.78. In addition, $4m\epsilon + 2m(m+1)\delta = 0.72$ is the point where C_1^4 becomes feasible. The only region where C_2^4 is not feasible is the cycle time region $0.72 \leq K < 0.78$. In this region, the cycle C_1^4 is feasible. Since $K - (6\epsilon + (2m+4)\delta) < P^U$ in this region, we cannot say that C_1^4 dominates the rest of pure cycles by using Theorem 4.3. However, by using Lemma 4.6, we can calculate the performance of C_1^4 in this region as follows:

We find $\varrho = 1.08$, by using Lemma 4.6. Hence, the total manufacturing cost obtained from cycle C_1^4 is at most 0.08 percent higher than the lower bound of total manufacturing cost, in this region. As a result, we can conclude that C_2^4 dominates the rest of pure cycles in most of the regions. In the only region where C_2^4 is not feasible, we see that the difference between the performance of C_1^4 and the lower bound is less than 0.08 times of the lower bound.

4.1.3 Discussion

The minimization of cycle time and minimization of total manufacturing cost objectives are fundamental objectives in the scheduling literature. In this section, we analyzed both of these objectives simultaneously in in-line robotic cells. The problem is to find the robot move sequence and processing times on machines minimizing the cycle time and the total manufacturing cost. In this section, the robot move cycles C_1^m and C_2^m are proved to be efficient in most of the cycle time region according to the bicriteria objective. In the next section the optimum number of machines in the cell is determined for C_1^m and C_2^m as a design problem.

4.2 Optimum Number of Machines

Until now, we have dealt with operational problems in robotic cells. In this section, we solve a design problem where we consider the number of machines as a decision variable.

4.2.1 Problem Definition and Solution Procedure

The optimum number of machines which is useful to determine the equipment requirements of the cell is calculated when the cycle time, robot travel times, loading/unloading times and the processing time upper bound are given parameters. The following theorem determines the optimum number of machines for C_1^m for the given values of ϵ, δ, the cycle time K, and P^U.

Theorem 4.4. *For a given cycle time K, the optimum number of machines m^* for cycle C_1^m is one of the integers below giving less unit cycle time:*

$$\lfloor min\{\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 2\delta^2} - \epsilon, \sqrt{\epsilon^2 + \delta K/2 + \epsilon\delta + \delta^2/4} - \epsilon - \delta/2\}/\delta\rfloor \text{ or}$$

$$\lfloor min\{\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 2\delta^2} - \epsilon, \sqrt{\epsilon^2 + \delta K/2 + \epsilon\delta + \delta^2/4} - \epsilon - \delta/2\}/\delta\rfloor + 1.$$

Proof. Lemma 4.4 defines the nondominated processing time vector of C_1^m as $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_1^m|K)$ where $P_1^* = P_m^* = min\{P^U, K - (6\epsilon + (2m+4)\delta)\}$ and $P_2^* = \ldots = P_{m-1}^* = min\{P^U, K - (4\epsilon + (2m+2)\delta)\}$. The cycle time of C_1^m for this processing time vector is found by using Lemma 4.1 as follows:

$$4m\epsilon + 2m(m+1)\delta + max\{0, min\{P^U, K - (6\epsilon + (2m+4)\delta)\} - ((4m-6)\epsilon + 2(m^2-2)\delta)\}$$

The per unit cycle time of C_1^m at cycle time K is found by dividing the cycle time by m as follows:

$$max\{4\epsilon + (2m+2)\delta, min\{P^U + 6\epsilon + (2m+4)\delta, K\}/m\}. \tag{4.4}$$

In order to minimize per unit cycle time in equation (4.4), the two arguments in the max function have to be minimized. The minimum value of max function is the minimum value of two arguments or the intersection point of these two arguments. As the number of machines decreases, the first argument decreases but the second argument increases. The first argument takes its minimum value when $m = 0$ but then the second argument $min\{P^U + 6\epsilon + (2m+4)\delta, K\}/m \to \infty$, so it does not minimize the max function. The second argument takes its minimum value when $m \to \infty$. However, when $m \to \infty$, the first term $4\epsilon + (2m+2)\delta \to \infty$, thus this does not give the minimum value of the max function. Now, we will determine the only remaining minimizer of max function which is the intersection of the two arguments:

$$4\epsilon + (2m+2)\delta = min\{P^U + 6\epsilon + (2m+4)\delta, K\}/m.$$

This equation leads to:

$$min\{P^U + 6\epsilon + 4\delta - (2\delta m^2 + 4\epsilon m), K - (2\delta m^2 + (4\epsilon + 2\delta)m)\} = 0.$$

This min function results 0 if the two arguments in the min function are non-negative and one of them is equal to 0. The m values making these arguments nonnegative are found as follows:

1. The first argument is nonnegative when:
   $0 \leq P^U + 6\epsilon + 4\delta - (2\delta m^2 + 4\epsilon m)$.
   This inequality has two roots and one of them is negative, we only consider the positive root hence, the inequality holds when:
   $m \leq (\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 2\delta^2} - \epsilon)/\delta$.

2. The second argument is nonnegative when:
   $0 \leq K - (2\delta m^2 + (4\epsilon + 2\delta)m)$.
   This inequality has two roots and one of them is negative, we only consider the positive root hence the inequality holds when:
   $m \leq (\sqrt{\epsilon^2 + \delta K/2 + \epsilon\delta + \delta^2/4} - \epsilon - \delta/2)/\delta$.

Since, both of the two arguments in the min function have to be nonnegative and one of the arguments has to be 0, the optimum m is stated as follows:

$$m = min\{(\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 2\delta^2} - \epsilon)/\delta, (\sqrt{\epsilon^2 + \delta K/2 + \epsilon\delta + \delta^2/4} - \epsilon - \delta/2)/\delta\}$$

This equation may result in a fractional value but m^* is an integer, thus m^* is the upper rounded or the lower rounded values of the equation above which gives less per unit cycle time. □

The following theorem determines m^* of C_2^m for the given values of ϵ, δ, K, and P^U.

Theorem 4.5. *For a given cycle time K, the optimum number of machines m^* for cycle C_2^m is one of the integers below giving less unit cycle time:*

$$\lfloor min\{\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 9\delta^2/4} - \epsilon - \delta/2, \sqrt{\epsilon^2 + \delta K/2 + 2\epsilon\delta + 2\delta^2} - \epsilon - \delta\}/\delta\rfloor \; or$$

$$\lfloor min\{\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 9\delta^2/4} - \epsilon - \delta/2, \sqrt{\epsilon^2 + \delta K/2 + 2\epsilon\delta + 2\delta^2} - \epsilon - \delta\}/\delta\rfloor + 1.$$

Proof. Lemma 4.5 defines the nondominated processing time vector of C_2^m as $(P_1^*, P_2^*, \ldots, P_m^*) \in \mathbf{P}^*(C_2^m|K)$ where $P_1^* = \ldots = P_m^* = min\{P^U, K - (4\epsilon + (2m+2)\delta)\}$. The per unit cycle time of C_2^m is found by dividing the cycle time corresponding to the nondominated processing time vector, which is found by using Lemma 4.2, by m as follows:

$$max\{4\epsilon + (2m + 4 - 2/m)\delta, min\{P^U + 4\epsilon + (2m+2)\delta, K\}/m\}. \tag{4.5}$$

In order to minimize per unit cycle time in equation (4.5), the two arguments in the max function have to be minimized. The minimum value of max function is the minimum value of two arguments or the intersection point of these two arguments. As the number of machines decreases, the first argument decreases but the second argument increases. The first argument becomes minimum when $m = 0$ but then the second argument $min\{P^U + 4\epsilon + (2m+2)\delta, K\}/m \to \infty$. Hence, it does not minimize the max function. The second argument takes its minimum value when $m \to \infty$. However, when $m \to \infty$, the first term $4\epsilon + (2m + 4 - 2/m)\delta \to \infty$. Thus, this does not give the minimum value of the max function. Now, we will find the only remaining minimizer of the max function which is the intersection of the two arguments:

$$4\epsilon + (2m + 4 - 2/m)\delta = min\{P^U + 4\epsilon + (2m+2)\delta, K\}/m.$$

This equation leads to:

$$min\{P^U + 4\epsilon + 4\delta - (2\delta m^2 + (4\epsilon + 2\delta)m), K + 2\delta - (2\delta m^2 + (4\epsilon + 4\delta)m)\} = 0.$$

This min function results 0 if the two arguments in the min function are non-negative and one of them is equal to 0. The m values making these arguments nonnegative are found as follows:

1. The first argument is nonnegative when:
   $0 \le P^U + 4\epsilon + 4\delta - (2\delta m^2 + (4\epsilon + 2\delta)m)$.

   This inequality has two roots and one of them is negative, we only consider the positive root. Hence, the inequality holds when:
   $m \le (\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 9\delta^2/4} - \epsilon - \delta/2)/\delta$.

2. The second argument is nonnegative when:
   $0 \le K + 2\delta - (2\delta m^2 + (4\epsilon + 4\delta)m)$.

This inequality has two roots and one of them is negative. We only consider the positive root. Hence, the inequality holds when:
$m \leq (\sqrt{\epsilon^2 + \delta K/2 + 2\epsilon\delta + 2\delta^2} - \epsilon - \delta)/\delta$.

Since, both of the two arguments in the min function have to be nonnegative and one of the arguments has to be 0, the optimum m is stated as follows:

$$min\{(\sqrt{\epsilon^2 + \delta P^U/2 + 3\epsilon\delta + 9\delta^2/4} - \epsilon - \delta/2)/\delta, (\sqrt{\epsilon^2 + \delta K/2 + 2\epsilon\delta + 2\delta^2} - \epsilon - \delta)/\delta\}$$

This equation may result in a fractional value but m^* is an integer, thus m^* is the upper rounded or the lower rounded values of the equation above which gives less per unit cycle time. □

We will now conclude the section by discussing the results found.

4.2.2 Discussion

Another problem focused in the literature is a design problem of finding the optimum number of machines in the cell. In this section, we focus on the design of the cell. The problem to be solved is to determine the optimum number of machines for pure cycles C_1^m and C_2^m in the cell when the robot load/unload times and robot travel time are given parameters. Finally, we determined the optimum number of machines for the two pure cycle studied in this chapter for C_1^m and C_2^m.

Chapter 5

Pure Cycles in Robot Centered Cells

In the previous chapter, we considered the minimization of cycle time and minimization of total manufacturing cost simultaneously for pure cycles in in-line robotic cells. In this chapter, we analyze the pure cycles in a different cell layout. The cell layout considered in this chapter is the robot centered cell. This chapter is composed of two sections. In the first section, we determine the pure cycles resulting in minimum cycle time for the described processing time regions. It is assumed that the processing times on machines are fixed and the same for all machines. We propose two cycles that are efficient in order to minimize the cycle time. After that, we analyze the 3-machine case in order to find the processing time regions and the corresponding pure cycles resulting in minimum cycle time in these regions. In the second section, the processing times are assumed to be controllable which makes our problem closer to the real life problems. We propose that the same two pure cycles proposed in the first section result in the minimum cycle time and minimum total manufacturing cost for most of the region. The reason for this prediction is the efficiency of these cycles in order to minimize cycle time. After that, the 3-machine robot centered cell is analyzed to determine the pure cycles resulting in minimum cycle time and total manufacturing cost, and the corresponding cycle time regions. Finally, the summary and the results of this section are presented.

5.1 Minimizing Cycle Time with Fixed Processing Times

In this problem, we considered an m-machine robot centered cell composed of CNC machines and an output and an input buffer are combined in an I/O station. In the first part, the problem to be solved is a single objective problem where the objective is to minimize the cycle time. For this problem, the processing times are assumed to be fixed and identical for all machines. We find the lower bound of cycle time of pure cycles in robot centered cells for a given processing time. We propose two pure cycles that result in the minimum cycle time in a specified processing time region. For the remaining processing time region, the worst case performance of these two pure cycles is determined. After that, the 3-machine case is analyzed. It is observed that the proposed cycles results in the minimum cycle time in most of the processing time region.

5.1.1 Problem Definition

The cell considered in this problem is an m-machine robot centered cell composed of CNC machines producing identical parts. The robot arm includes one gripper and contains at most one part at a time. All of the operations performed on identical parts are performed by only one machine. Thus, a part taken from I/O station is transferred onto one of the machines and after all of the processes on the part are finished, the part is returned to I/O station. The objective is to maximize the throughput rate in other words to minimize the cycle time in pure cycles. The problem is to determine the robot move sequence minimizing the cycle time. We focused on pure cycles since they are practical, good performing and easy to implement. Since a pure cycle produces m parts throughout a cycle, a pure cycle is an m-unit cycle.

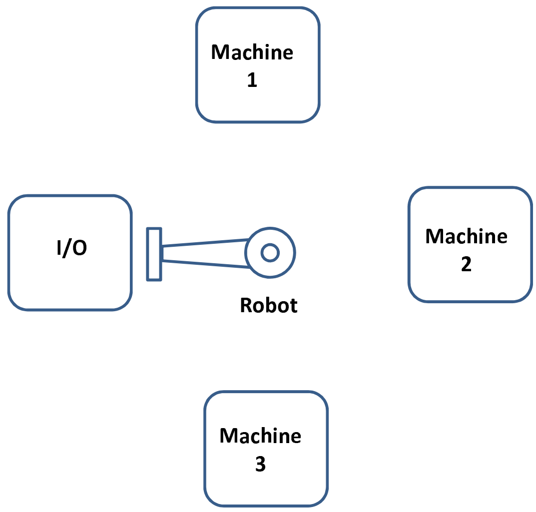

Figure 5.1: 3-Machine Robot Centered Cell

In this study, the prominent pure cycles resulting in minimum cycle time in an m-machine robot centered cell including an I/O station are determined. A 3-machine robot centered cell is presented in Figure 5.1. The time required to travel the distance between any two consecutive machines is considered as equal and δ time units. The m-machine robot centered cell produces identical parts on identical machines and the processing times of identical parts are assumed to be fixed as P. The definitions and assumptions presented in Chapter 3 are used in this section. Additional definitions used in this section and which are common in literature is presented as follows:

δ : The time required for rotational movement between two consecutive machines which is assumed to be additive such that the travelling time between machine i and j is $min\{|i-j|, m+1-|i-j|\}\delta$.

P : The fixed processing time on machines. The processing time is fixed and the same for every machine in this section.

I/O *station*: The I/O station consists of an Input device, from which parts are introduced into the cell, and an Output device, onto which the parts are dropped upon completion of their processing on the machines

In the next section, the solution of the problem considered in this section is presented.

5.1.2 Solution Procedure

In this part, the solution method of determining the pure cycles minimizing the cycle time is presented step by step. At first, the intuition of finding good performing pure cycles is presented. The number of different pure cycles is $(2m-1)!$ for an m-machine cell. For a 3-machine robotic cell, there are 120 pure cycles and for a 4-machine robotic cell, there are 5040 pure cycles. The number of possible pure cycles increases as a factorial function of the number of machines in the cell. So it is very useful to have an intuition to find good performing cycles among these numerous cycles. Furthermore, the cycles we proposed according to this intuition are proved to minimize cycle time as will be shown in the later parts of this study. After presenting the intuition of selecting good pure cycles among numerous pure cycles, we propose that two pure cycles are efficient cycles. Then, the cycle times of these pure cycles are determined when there is a given processing time. The lower bound of cycle time is found for the robot centered cell.

Consequently, the cycle times obtained from the proposed cycles are compared to the lower bound of cycle time and by this way, the processing time region where the proposed cycles result in minimum cycle time is determined. After that, the 3-machine analysis is presented.

5.1.2.1 Intuition for Efficient Pure Cycles

At this step, the intuition of finding good performing cycles according to the objective of minimizing cycle time is presented. First, we analyze the structure of cycle time equation of pure cycles. The cycle time of a pure cycle is composed of two parts. The first part is the total time required for the robot activities which are part transportation and load/unload activities. The second part is the total waiting time of robot in front of machines before unloading them. The time required for robot activities is calculated as follows. The robot activities are composed of two parts, the load/unload operation and transportation of parts by robot from I/O station to a machine and after the operations on that part is finished, transportation of the part from that machine to I/O station. Time required for robot load/unload times is calculated as follows. For each part, the part is taken from I/O station (ϵ), then loaded to machine i (ϵ), after all of the operations are finished the part is unloaded from machine i (ϵ) and finally the part is dropped into I/O station (ϵ) which makes a total of 4ϵ time units for one part. Since a pure cycle produces m parts, the total time required for loading and unloading is $4m\epsilon$ time units. It is obvious that the total robot load/unload times is $4m\epsilon$ time units and it is the same for of all pure cycles for an m-machine cell. However, the robot travel time and total waiting time differ according to the robot move sequence. Let the total robot travel time for pure cycle C_i^m be $a_i\delta$. Now, the cycle time of pure cycle C_i^m can be presented as:

$$T_{C_i^m} = 4m\epsilon + a_i\delta + w_1 + w_2 + \ldots + w_m$$

The values of the total robot travel time $a_i\delta$ and the total waiting time $w_1 + w_2 + \ldots + w_m$ differ according to robot move sequence. Thus, the sum of these terms has to be minimized in order to minimize cycle time. Waiting time of machine i is denoted as $w_i = max\{0, P - v_i\}$. v_i is defined by Gultekin et al. [12] as the amount of time between just after loading the machine i and the time robot returns back in front of machine i to unload it. The waiting times in the cycle time equation are written explicitly as follows:

$$T_{C_i^m} = 4m\epsilon + a_i\delta + max\{0, P - v_1\} + max\{0, P - v_2\} + \ldots + max\{0, P - v_m\} \quad (5.1)$$

There could be two different approaches to minimize the cycle time in equation 5.1. The first approach is to minimize the robot travel time. If the processing times are small, this approach is more efficient in order to minimize the cycle time. The second approach is minimizing the total waiting times in order to decrease the cycle time. This approach becomes more efficient when the processing times are greater. These results are obtained by observing the behavior of equation 5.1 as the processing time increases or decreases.

In this study, we focus on the second approach, minimizing total waiting times. Thus, it is expected that the resulting cycles are going to be more efficient in minimizing cycle time for higher processing times. The waiting time on machine i is denoted as $max\{0, P - v_i\}$ in equation 5.1. Since P is constant, in order to reduce waiting time, we have to find the pure cycles resulting in higher v_i values. Thus, we have to find the robot move sequence where v_i values take their maximum values. The v_i is defined as the amount of time between just after loading the machine and just after reaching in front of machine i to unload it. Let us define a new variable b_i as follows:

$$b_i = T_{C_i^m} - v_i \quad (5.2)$$

The equation above simply implies that b_i is the time between just reaching in front of machine i to unload it to the time just after loading machine i. In other words, b_i is the complement of cycle time for v_i. v_i can be calculated from this equation as $v_i = T_{C_i^m} - b_i$. Since our aim is to maximize v_i, we have to find the minimum value of b_i in equation 5.2. The minimum time between just coming in front of machine i to unload it and the time just after loading machine i is calculated as follows. The minimum robot activities that must be performed during this b_i time units are waiting to unload machine i (w_i), then unloading machine i (ϵ), then transporting part to the I/O station ($min\{i, m + 1 - i\}\delta$), dropping part to the I/O station (ϵ), after that picking a new part to load machine

i (ϵ), then transporting part to machine i ($min\{i, m+1-i\}\delta$) and finally loading machine i (ϵ). So, the minimum value of b_i is $4\epsilon + 2min\{i, m+1-i\}\delta$ time units. This means that the loading activity of machine i is immediately sequenced after unloading activity in the robot move sequence which means that U_iL_i is the activity sequence minimizing b_i. So, this robot activity sequence minimizes the waiting time on machine i. The lower bound of b_i is calculated in this paragraph so, the value of b_i for pure cycles is presented as follows:

$$4\epsilon + 2min\{i, m+1-i\}\delta + w_i \leq b_i \tag{5.3}$$

The value of b_i is calculated in U_iL_i sequence as follows. The robot waits for machine i to complete the processing of the part (w_i), then unloads the part (ϵ), after that transports the part to the I/O station ($min\{i, m+1-i\}\delta$), and then drops the part (ϵ) and takes a new part to load the machine i (ϵ), then transports the part to the machine ($min\{i, m+1-i\}\delta$) and finally loads the part to the machine (ϵ). This makes a total of $4\epsilon + 2min\{i, m+1-i\}\delta + w_i$, thus b_i is equal to its lower bound in equation 5.3. So, we showed that (U_iL_i) robot activity sequence minimizes waiting time on machine i. In order to minimize the total waiting time, all of the waiting times on all machines have to be minimized. Thus, for each machine the load activity have to be immediately sequenced after unloading activity. The resulting robot move sequence is:

$$U_{k_1}L_{k_1}U_{k_2}L_{k_2}\dots U_{k_m}L_{k_m}$$

where $k_i, k_j \in [1, 2, \dots m]$, $i, j \in [1, 2, \dots m]$, $k_i \neq k_j$ when $i \neq j$.

The resulting robot move sequence is proposed to be efficient in order to decrease total waiting time. However, there are $(m-1)!$ pure cycles in the structure defined above. Now, we are going to select one of those pure cycles in this structure which minimizes the robot travel time. Thus, we obtain the most efficient cycle in order to minimize the cycle time in the set of pure cycles minimizing total waiting time. The total robot travel time of the pure cycles in which the robot loads each machine immediately after unloading that machine is calculated as:

$(i+2\sum_{\forall i} min\{i, m+1-i\}+\sum_{i\neq m} min\{|k_i-k_{i+1}|, m+1-|k_i-k_{i+1}|\}+min\{k_m, m+$

$1 - k_m\})\delta$

The lower bound of the total robot travel time described above is found as follows. First, we present the lower bounds of components of the robot travel time equation above:

$1 \leq i$, $1 \leq min\{|k_i - k_{i+1}|, m + 1 - |k_i - k_{i+1}|\}$, $1 \leq min\{k_m, m + 1 - k_m\}$ and $2\sum_{\forall i} min\{i, m + 1 - i\} = \lceil m(m + 2)/2\rceil$.

Hence the lower bound is found as:

$$(\lceil m(m + 2)/2\rceil + m + 1)\delta \tag{5.4}$$

The cycle time of C_2^m and C_3^m are presented in Lemmas 5.1 and 5.2 respectively and for both of these cycles, the robot travel time is equal to the lower bound of robot travel time in equation 5.4 for the cycles in which the machines are loaded just after they are unloaded. Thus, we selected C_2^m and C_3^m, since they are the pure cycles minimizing total robot travel time among pure cycles minimizing total waiting time.

5.1.2.2 C_2^m and C_3^m robot move cycles

According to the inspiration presented in the previous part, we investigate the good cycles and we propose that two pure cycles, C_2^m and C_3^m results in the minimum cycle time for pure cycles for the specified processing time region. These two cycles are defined as follows:

Cycle C_2^m is defined in Definition 4.2 as the robot move cycle in an m-machine robotic cell with the following activity sequence: $L_1U_mL_mU_{m-1}L_{m-1}\ldots U_2L_2U_1$. The second proposed pure cycle for this part is defined as follows:

Definition 5.1. *C_3^m is the robot move cycle in an m-machine robotic cell with the following activity sequence: $L_1U_2L_2U_3L_3U_4L_4\ldots U_{m-1}L_{m-1}U_mL_mU_1$.*

The initial state of the cell is identical for both of C_2^m and C_3^m. All of the machines except machine 1 are loaded with a part and machine 1 is empty. The robot is in front of the I/O station and idle. The following lemma determines the cycle time of C_2^m for a given processing time value P.

Lemma 5.1. *The cycle time of C_2^m for a given fixed processing time P is represented as follows:*

$T_{C_2^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta\}$.

Proof. Let t_l be the completion time of activity $l \in \mathcal{A}$. The cycle time of C_2^m is found as follows:
$t_{L_1} = 2\epsilon + \delta$,
$t_{U_m} = t_{L_1} + 2\epsilon + 3\delta + w_m$,
$t_{L_m} = t_{U_m} + 2\epsilon + \delta$,
$t_{U_i} = t_{L_{i+1}} + 2\epsilon + \delta + min\{i, m+1-i\}\delta + w_i, \;\; i = m-1, m-2, \ldots, 2$,
$t_{L_i} = t_{U_i} + 2\epsilon + min\{i, m+1-i\}\delta, \;\; i = m-1, m-2, \ldots, 2$,
$t_{U_1} = t_{L_2} + 2\epsilon + 2\delta + w_1$.
The robot is at the I/O station at the end of the cycle thus, the cycle time is presented as:

$$T_{C_2^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + w_1 + w_2 + \ldots + w_m \tag{5.5}$$

The waiting time for any machine i is defined as $w_i = max\{0, P - v_i\}$. The waiting times depend on the v_i values and the v_i's are calculated as follows:
$v_i = T_{C_2^m} - (4\epsilon + 2min\{i, m+1-i\}\delta + w_i) = (4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2min\{i, m+1-i\})\delta + w_1 + w_2 + \ldots + w_m - w_i, \forall i$.

The total waiting time $\sum_{\forall i} w_i$ of cycle time of C_2^m in equation 5.5 is found as follows:

1. If $P \leq v_i, \forall i$ then, $w_1 + w_2 + \ldots + w_m = 0$
2. Else if $\exists k \in [1, \ldots, m]$ such that $v_k < P$, then $w_k = P - v_k = P - (4m-4)\epsilon - (\lceil m(m+2)\rceil + m + 1 - 2min\{k, m+1-k\})\delta - \sum_{i \neq k} w_i$. Hence, $w_1 + w_2 + \ldots + w_m = P - (4m-4)\epsilon - (\lceil m(m+2)/2\rceil + m + 1 - 2min\{k, m+1-k\})\delta$.

Now we can state that:

$T_{C_2^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2\rceil + m + 1 - 2min\{k, m+1-k\})\delta; \forall k \in [1, \dots, m]\}$.

Since $min\{k, m+1-k\}$ takes its maximum value when $k = \lceil m/2\rceil$, the equation turns into:

$T_{C_2^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta\}$. □

The following lemma determines the cycle time of C_3^m for a given processing time value P.

Lemma 5.2. *The cycle time of C_3^m for a given fixed processing time P is represented as follows:*

$T_{C_3^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta\}$.

Proof. Let t_l be the completion time of activity $l \in \mathcal{A}$. The cycle time of C_3^m is found as follows:
$t_{L_1} = 2\epsilon + \delta$,
$t_{U_i} = t_{L_{i-1}} + 2\epsilon + \delta + min\{i, m+1-i\}\delta + w_i, \quad i = 2, 3, \dots, m$,
$t_{L_i} = t_{U_i} + 2\epsilon + min\{i, m+1-i\}\delta, \quad i = 2, 3, \dots, m$,
$t_{U_1} = t_{L_m} + 2\epsilon + 3\delta + w_1$.
The robot is at the I/O station at the end of the cycle, thus the cycle time is presented as:

$$T_{C_3^m} = 4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta + w_1 + w_2 + \ldots + w_m \quad (5.6)$$

The waiting time for any machine i is defined as $w_i = max\{0, P - v_i\}$. The waiting times depend on the v_i values and the v_i's are calculated as follows:
$v_i = T_{C_3^m} - (4\epsilon + 2min\{i, m+1-i\}\delta + w_i) = (4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2min\{i, m+1-i\})\delta + w_1 + w_2 + \ldots + w_m - w_i, \forall i$.

The total waiting time $\sum_{\forall i} w_i$ of cycle time of C_3^m in equation 5.6 is found as follows:

1. If $P \leq v_i, \forall i$ then, $w_1 + w_2 + \ldots + w_m = 0$

2. Else if $\exists k \in [1, \ldots, m]$ such that $v_k < P$, then $w_k = P - v_k = P - (4m - 4)\epsilon - (\lceil m(m+2) \rceil + m + 1 - 2min\{k, m+1-k\})\delta - \sum_{i \neq k} w_i$. Hence, $w_1 + w_2 + \ldots + w_m = P - (4m-4)\epsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2min\{k, m+1-k\})\delta$.

Now we can state that:

$T_{C_3^m} = 4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2min\{k, m+1-k\})\delta; \forall k \in [1, \ldots, m]\}$.

Since $min\{k, m+1-k\}$ takes its maximum value when $k = \lceil m/2 \rceil$, the equation turns into:

$T_{C_3^m} = 4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + max\{0, P - (4m-4)\epsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2\lceil m/2 \rceil)\delta\}$. $\square$

In the next theorem, the cycle time lower bound of pure cycles for the robot centered cells with I/O station is determined.

Theorem 5.1. *For an m-machine robot centered cell, the cycle time of any pure cycle is no less than*

$$\underline{T_{I/O}} = max\{4m\epsilon + \lceil m(m+2)/2 \rceil \delta, 4\epsilon + 2\lceil m/2 \rceil \delta + P\}. \tag{5.7}$$

Proof. A lower bound for a pure cycle can be calculated by using two different definitions of the cycle time. The first lower bound is obtained from the exact robot activity time and the second is obtained from the given processing time vector. Since the robot has to perform an exact set of robot activities, the total time required for these activities constitutes a lower bound. Thus, the first lower bound is obtained as follows: The set of robot activities can be analyzed in two groups and the first group is robot loading and unloading times. First, a part is taken from the I/O station (ϵ), then loaded to one of the machines (ϵ), after the processing on the machine is finished, the part is unloaded (ϵ) and dropped to the I/O station (ϵ). This makes a total of $4m\epsilon$ for a repetition of cycle. The robot travel times constitute the second group of robot activities. The robot takes a part from I/O station and travels to machine i to load it ($min\{i, m+1-i\}\delta$), after the processing on the part is finished, robot unloads the machine and travels to the I/O station to drop the finished part ($min\{i, m+1-i\}\delta$).

1. Suppose the number of machines is even, then the total robot travel time is calculated as:
   $\sum_{i=1}^{m} min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \ldots + m\delta + m\delta + (m-2)\delta + (m-4)\delta + \ldots + 2\delta = \lceil m(m+2)/2 \rceil \delta$.

2. Suppose the number of machines is odd, then the total robot travel time is calculated as:
   $\sum_{i=1}^{m} min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \ldots + (m+1)\delta + (m-1)\delta + (m-3)\delta + \ldots + 2\delta = \lceil m(m+2)/2 \rceil \delta$.

Consequently, the total of robot activities require at least $4m\epsilon + \lceil m(m+2)/2 \rceil \delta$ time units.

The second definition of a cycle time that leads to another lower bound is the minimum time between two consecutive loadings of any machine. The minimum time needed to unload machine i after loading it is P time units. After processing of the part is finished, the part is unloaded (ϵ), it is transferred to I/O station ($min\{i, m+1-i\}\delta$), and dropped (ϵ). After that, the robot takes a new part to make the consecutive loading of machine i (ϵ), brings the new part to machine i ($min\{i, m+1-i\}\delta$) and finally loads the machine (ϵ). The total time between two consecutive loadings of machine i is at least $4\epsilon + 2min\{i, m+1-i\}\delta + P$. However there are m machines and the total time for consecutive loadings are different for each of them. Thus, the cycle time has to be greater than or equal to the minimum time required between two consecutive loadings of any machine in the cell. So, the second lower bound of the cycle time is $4\epsilon + 2max\{min\{i, m+1-i\}, i : 1, \ldots, m\}\delta + P$. □

As can be seen from Lemma 5.1 and Lemma 5.2, the pure cycles C_2^m and C_3^m result in the same cycle time for each processing time. Thus, the processing time region where they result in the minimum cycle time for pure cycles will be the same as well. The next theorem determines the processing time region where either C_2^m or C_3^m dominates the rest of pure cycles.

Theorem 5.2. *For an m-machine robot centered cell, either C_2^m or C_3^m dominates the rest of pure cycles in the processing time region:*

$$(4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta \leq P$$

Proof. If $(4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta \leq P$, then $T_{C_2^m} = T_{C_3^m} = 4\epsilon + 2\lceil m/2\rceil\delta + P = \underline{T_{I/O}}$. □

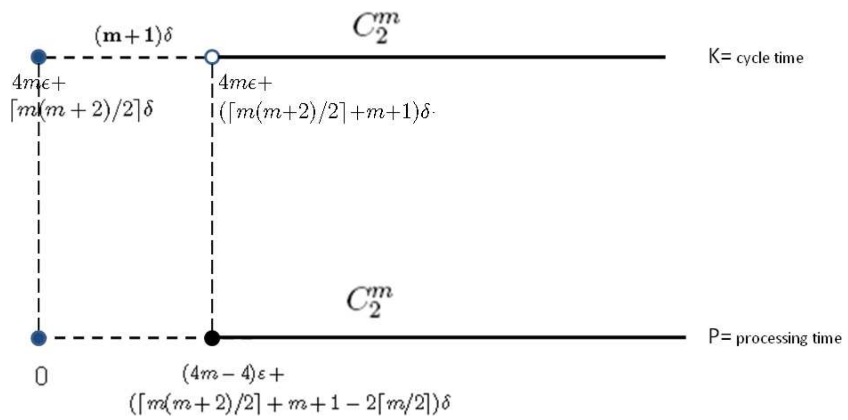

Figure 5.2: Cycle time-Processing time region where C_2^m dominates the rest of pure cycles

In Figure 5.2, we can see the processing time region where either C_2^m or C_3^m dominates the rest of pure cycles and the corresponding cycle time region. From Theorem 5.2, for the processing time region $0 < P < (4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta$, C_2^m does not dominate the rest of pure cycles. For the processing time region $(4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta \leq P$, either C_2^m or C_3^m results in the cycle time lower bound. This processing time region corresponds to the cycle time region $4m\epsilon + (\lceil m(m+2)/2\rceil + m + 1)\delta \leq K$. Consequently, it is observed that the only cycle time region where C_2^m and C_3^m is not optimum is the only region starting from cycle time lower bound and the size of this region is $(m+1)\delta$ time units. This region becomes smaller relative to whole cycle time region as the number of machines increases.

In the next lemma, for the remaining processing time region where neither C_2^m nor C_3^m does not result in minimum cycle time the worst case performances of the two cycles are calculated. The worst case performance is calculated by comparing cycle time are obtained from C_2^m and C_3^m to the cycle time lower bound in the mentioned processing time region. Since C_2^m and C_3^m result in the same cycle time for any processing time, their worst case performances are equal. Let T^* represents the minimum cycle time obtained from pure cycles in the described processing time region.

Lemma 5.3. *For the remaining processing time region:*

$$P < (4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta$$

the performance of either C_2^m or C_3^m in this region is stated as:

$$T_{C_2^m}, T_{C_3^m} \leq (1 + \frac{(m+1)\delta}{4m\epsilon + \lceil m(m+2)/2\rceil\delta}) \cdot T^*.$$

Proof. The cycle time of C_2^m and C_3^m are equal $T_{C_2^m} = T_{C_3^m}$ as can be seen from equation of $T_{C_2^m}$ in Lemma 5.1 and equation of $T_{C_3^m}$ in Lemma 5.2 . In the mentioned processing time region, the lower bound of cycle time is calculated from Theorem 5.1 as $4m\epsilon + \lceil m(m+2)/2\rceil\delta \leq \underline{T_{I/O}}$. Then,

$$\frac{T_{C_3^m}}{T^*} = \frac{T_{C_2^m}}{T^*} \leq \frac{T_{C_2^m}}{\underline{T_{I/O}}} \leq \frac{4m\epsilon+(\lceil m(m+2)/2\rceil+m+1)\delta}{4m\epsilon+\lceil m(m+2)/2\rceil\delta} = 1 + \frac{(m+1)\delta}{4m\epsilon+\lceil m(m+2)/2\rceil\delta}. \qquad \square$$

The only region where neither C_2^m nor C_3^m does not dominate the rest of pure cycles is the processing time region $P < (4m-4)\epsilon + (\lceil m(m+2)/2\rceil + m + 1 - 2\lceil m/2\rceil)\delta$. The difference between cycle time lower bound and the cycle time of either C_2^m or C_3^m in this region decreases as the number of machines increases and vice versa. In order to see the decrease in difference between cycle time lower bound and cycle time of either C_2^m or C_3^m as the number of machines increases, the 2-machine cell and m-machine cell where $m \to \infty$ are analyzed for this processing time region. Firstly, the maximum difference between lower bound and cycle time of either C_2^m or C_3^m is observed in 2-machine cell. The cycle time lower bound is calculated from Theorem 5.1 as $8\epsilon + 4\delta \leq \underline{T_{I/O}}$. The cycle time of either C_2^2 or C_3^2 is determined from Lemma 5.1 as $T_{C_3^2} = T_{C_2^2} = 8\epsilon + 7\delta$.

The difference between either $T_{C_2^m}$ or $T_{C_3^m}$ and the lowest cycle time that can be obtained from pure cycles T^* is determined by using Lemma 5.3 as $T_{C_3^2} = T_{C_2^2} \leq (1 + \frac{(m+1)\delta}{4m\epsilon + \lceil m(m+2)/2 \rceil \delta}) \cdot T^* = (1 + \frac{3\delta}{8\epsilon+4\delta}) \cdot T^*$. Hence, the worst case of difference between either $T_{C_2^2}$ or $T_{C_3^2}$ and T^* is $\frac{3\delta}{8\epsilon+4\delta} \cdot T^*$. The difference between $T_{C_2^m}$, $T_{C_3^m}$ and T^* takes its minimum value as $m \to \infty$ and it is determined by using Lemma 5.3 as $T_{C_3^m} = T_{C_2^m} \leq (1 + \frac{(m+1)\delta}{4m\epsilon + \lceil m(m+2)/2 \rceil \delta}) \cdot T^* = T^*$. Since, cycle time of either C_2^m or C_3^m is at least equal to $T^* \leq T_{C_2^m} = T_{C_3^m}$, the cycle time obtained from either C_2^m or C_3^m is equal to the minimum cycle time, $T_{C_3^m} = T_{C_2^m} = T^*$. Thus, we say that the difference between either $T_{C_2^m}$ or $T_{C_3^m}$ and T^* decreases as the number of machines increases.

5.1.3 3-Machine Analysis

In the Figure 5.3, the pure cycles resulting in minimum cycle time for pure cycles are presented among all possible 3-machine cell. There are 120 pure cycles in a 3-machine cell and we have calculated the cycle times of all of these pure cycles. However, they cannot be presented in the graph because of space limitation, only 12 of the pure cycles are presented. The cycle time of these pure cycles are presented as follows:

$C_1^3 = L_1L_3U_2L_2U_1U_3$ with cycle time $T_{C_1^3} = 12\epsilon + 12\delta + max\{0, P - 8\delta - 6\epsilon\}$,
$C_2^3 = L_1U_3L_3U_2L_2U_1$ with cycle time $T_{C_2^3} = 12\epsilon + 12\delta + max\{0, P - 8\delta - 8\epsilon\}$,
$C_3^3 = L_1U_2L_2U_3L_3U_1$ with cycle time $T_{C_3^3} = 12\epsilon + 12\delta + max\{0, P - 8\delta - 8\epsilon\}$,
$C_4^3 = L_1U_1L_2U_2L_3U_3$ with cycle time $T_{C_4^3} = 12\epsilon + 8\delta + 3P$,
$C_5^3 = L_1U_1L_2U_2U_3L_3$ with cycle time $T_{C_5^3} = 12\epsilon + 10\delta + 2P$,
$C_6^3 = L_1U_1L_2L_3U_2U_3$ with cycle time $T_{C_6^3} = 12\epsilon + 12\delta + P + max\{0, P - 4\delta - 2\epsilon\}$,
$C_7^3 = L_1U_2L_2U_1L_3U_3$ with cycle time $T_{C_7^3} = 12\epsilon + 10\delta + P + max\{0, P - 6\delta - 4\epsilon\}$,
$C_8^3 = L_1L_2L_3U_1U_2U_3$ with cycle time $T_{C_8^3} = 12\epsilon + 16\delta + max\{0, P - 8\delta - 4\epsilon\}$.
In addition the robot move sequences and cycle times of pure cycles C_{11}^3, C_{12}^3, C_{15}^3 and C_{19}^3 are presented in Table 5.1.
The cycle time lower bound is found by using Theorem 5.1 as:
$\underline{T_{I/O}} = max\{12\epsilon + 8\delta, 4\epsilon + 4\delta + P\}$.

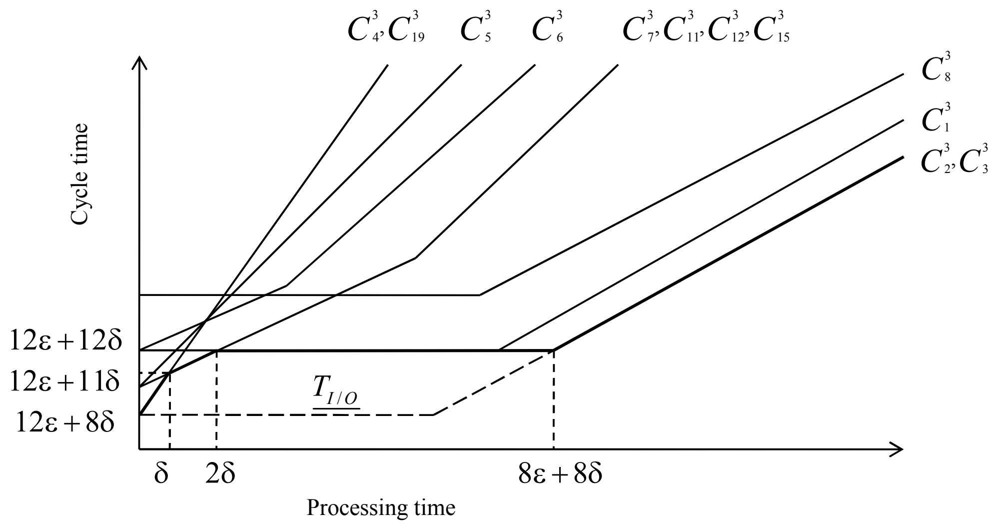

Figure 5.3: 3-machine cell analysis

The graph in Figure 5.3 is useful in order to see the efficiency of C_2^3 and C_3^3 in a 3-machine cell. The dashed line in the graph $(\underline{T_{I/O}})$ represents the cycle time lower bound found by using Theorem 5.1. The bold lines in the graph represents the minimum cycle time corresponding to processing times. Either C_2^3 or C_3^3 results in the minimum cycle time in the processing time region $2\delta \leq P$. The only region where neither C_2^3 nor C_3^3 does not dominate the rest is the region of $P < 2\delta$. Consequently, the efficiency of cycles C_2^3 and C_3^3 in order to minimize cycle time is presented in 3-machine cells.

In the next theorem, the pure cycles minimizing cycle time and corresponding processing time regions are presented for 3-machine cells.

Theorem 5.3. *For 3-machine case*

1. *If $P < \delta$, then either C_4^3 or C_{19}^3 results in minimum cycle time for pure cycles,*
2. *If $\delta \leq P < 2\delta$, then either C_7^3 or C_{11}^3 or C_{12}^3 or C_{15}^3 results in minimum cycle time for pure cycles,*
3. *If $P \geq 2\delta$, then either C_2^3 or C_3^3 results in minimum cycle time for pure*

cycles.

Proof. There are 120 different pure cycles and we calculated the cycle time of all these pure cycles. The proof consists of 4 parts and in order to obtain these results, the cycle time of 120 pure cycles are compared with the proposed cycles.

1. When $K < 12\epsilon+10\delta$, only C_4^3 and C_{19}^3 are feasible pure cycles in this region and they result in the same cycle time as can be seen in Table 5.1. The corresponding processing time region for this cycle time region is $P < 2\delta/3$. Thus, there is no other pure cycle that can result in less cycle time than either C_4^3 or C_{19}^3.

2. When $12\epsilon + 10\delta \leq K < 12\epsilon + 12\delta$, all of the 14 pure cycles presented in Table 5.1 are feasible. When we compare the cycle time of C_4^3 and C_{19}^3 with the other 12 pure cycles, either C_4^3 or C_{19}^3 results in the minimum cycle time in the processing time region $2\delta/3 \leq P < \delta$ which corresponds to the cycle time region $12\epsilon+10\delta \leq K < 12\epsilon+11\delta$. The cycle times of C_7^3, C_{11}^3, C_{12}^3 and C_{15}^3 are equal for any processing time as can be seen in Table 5.1. Similarly by comparing the cycle time of C_7^3, C_{11}^3, C_{12}^3 and C_{15}^3 with the cycle time of other 10 feasible pure cycles, the cycle time of C_7^3, C_{11}^3, C_{12}^3 and C_{15}^3 is the minimum cycle time for processing time region $\delta \leq P < 2\delta$ corresponding to cycle time region $12\epsilon + 11\delta \leq K < 12\epsilon + 12\delta$.

3. There are 24 feasible pure cycles at cycle time $K = 12\epsilon+12\delta$ when $2\delta \leq P$. When we compare cycle time of either C_2^3 or C_3^3 with the rest of pure cycles, the cycle time of either C_2^3 or C_3^3 is the minimum cycle time for the processing time region $2\delta \leq P < 8\epsilon + 8\delta$.

4. From Theorem 4.2, either C_2^3 or C_3^3 results in the minimum cycle time for the processing time region $8\epsilon + 8\delta \leq P$. □

Table 5.1 presents the robot move sequences and total cycle time for 14 pure cycles which are used in the proof of Theorem 5.3.

Cycle	Robot move sequence	Cycle Time ($T_{C_i^3}$)
C_4^3	$L_1U_1L_2U_2L_3U_3$	$12\epsilon + 8\delta + 3P$
C_5^3	$L_1U_1L_2U_2U_3L_3$	$12\epsilon + 10\delta + 2P$
C_7^3	$L_1U_2L_2U_1L_3U_3$	$12\epsilon + 10\delta + P + max\{0, P - 4\epsilon - 6\delta\}$
C_9^3	$L_1L_3U_3L_2U_2U_1$	$12\epsilon + 10\delta + 2P$
C_{10}^3	$L_1L_3U_3U_1L_2U_2$	$12\epsilon + 10\delta + 2P$
C_{11}^3	$L_1U_2L_3U_3L_2U_1$	$12\epsilon + 10\delta + P + max\{0, P - 4\epsilon - 4\delta\}$
C_{12}^3	$L_1U_1L_2U_3L_3U_2$	$12\epsilon + 10\delta + P + max\{0, P - 4\epsilon - 4\delta\}$
C_{13}^3	$L_1L_2U_2L_3U_3U_1$	$12\epsilon + 10\delta + 2P$
C_{14}^3	$L_1L_2U_2U_1L_3U_3$	$12\epsilon + 10\delta + 2P$
C_{15}^3	$L_1U_1L_3U_2L_2U_3$	$12\epsilon + 10\delta + P + max\{0, P - 4\epsilon - 6\delta\}$
C_{16}^3	$L_1U_1L_3L_2U_2U_3$	$12\epsilon + 10\delta + 2P$
C_{17}^3	$L_1U_1U_3L_2U_2L_3$	$12\epsilon + 10\delta + 2P$
C_{18}^3	$L_1U_1U_3L_3L_2U_2$	$12\epsilon + 10\delta + 2P$
C_{19}^3	$L_1U_1L_3U_3L_2U_2$	$12\epsilon + 8\delta + 3P$

Table 5.1: Some of the robot move sequences and corresponding cycle times in 3-machine robotic cells

5.1.4 Discussion

In this section, we analyzed the pure cycles in robot centered cells where the machines are CNC machines. The processing times are assumed to be fixed and the same on every machine. The problem was to determine the robot move sequences minimizing cycle time for a given processing time. Consequently, it is proved that either C_2^m or C_3^m results in the minimum cycle time for the specified processing time region in Theorem 5.2. For the remaining region where neither C_2^m nor C_3^m does not result in minimum cycle time, the worst case performance analysis made by comparing the cycle time of C_2^m and C_3^m to cycle time lower bound in that processing time region. In addition, the cycles resulting in minimum cycle time in 3-machine robot centered cell are determined. In 3-machine cell analysis, it is observed that either C_2^m or C_3^m results in the minimum cycle time for most of the processing time region.

Until now, we have focused on the cycle time objective and we find that either C_2^3 or C_3^3 results in minimum cycle time for a specified processing time region. From now on, the controllability is introduced to our problem so that the processing times can be increased or decreased. The efficient cycles that minimize both cycle time and total manufacturing cost simultaneously are investigated. Since C_2^m and C_3^m minimize cycle time in most of the processing time region, we propose that C_2^m and C_3^m are efficient pure cycles for this bicriteria problem. In the next section, the cycle time region where either C_2^m or C_3^m dominates the rest is determined.

5.2 Bicriteria Analysis of C_2^m and C_3^m

In this part, there is an m-machine robot centered cell and the processing times are assumed to be controllable. The objective is the bicriteria objective of minimizing cycle time and total manufacturing cost simultaneously. The pure cycles C_2^m and C_3^m are proved to result in minimum cycle time in a specified processing time region in the previous chapter. Thus, we propose that these two prominent cycles are effective pure cycles in terms of these two objectives. For the stated bicriteria optimization problem, the 3-machine case analysis according to this bicriteria objective is presented. In 3-machine robot centered cell, the pure cycles and processing time vectors resulting in both minimum cycle time and minimum total manufacturing cost for specified cycle time regions are determined.

5.2.1 Problem Definition

In this section, different from the previous section, the machines are considered as highly flexible CNC machines in which the processing times are controllable such that it enables us to increase or decrease the processing times. The feasible processing times on machines are assumed to be between a lower bound and an upper bound which is denoted by $P^L \leq P_i \leq P^U$. A processing time vector is denoted as $\boldsymbol{P} = (P_1, P_2, \ldots, P_m)$. Feasible processing time vectors are composed of feasible processing times and the set of feasible processing time vectors is denoted as $\mathcal{P}_{feas} = \{(P_1, P_2, \ldots, P_m) \in R^m : P^L \leq P_i \leq P^U, \ \forall i\}$.

5.2.2 Solution Procedure

In this part, we present the solution method for this problem. First, we find the cycle time of proposed pure cycles C_2^m and C_3^m when a processing time vector is given. Afterwards, we find the processing time vector resulting in lower bound of total manufacturing cost for pure cycles. The nondominated solutions of C_2^m and C_3^m are determined. The total manufacturing cost obtained from nondominated solutions of C_2^m and C_3^m are compared with the lower bound of total manufacturing cost. By this way, the cycle time region where either C_2^m or C_3^m dominates the rest of pure cycles is determined. The next lemma determines the cycle time of either C_2^m or C_3^m when there is a given processing time vector.

Lemma 5.4. *The cycle time of either C_2^m or C_3^m for a given processing time vector is presented as follows:*

$T_{C_2^m} = T_{C_3^m} = 4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + max\{0, max\{P_i - (4m - 4)\epsilon - (\lceil m(m+2) \rceil + m + 1 - 2min\{i, m+1-i\})\delta, i : 1, \ldots, m\}\}.$

Proof. The cycle time of C_2^m in equation (5.5) and the cycle time of C_3^m in equation (5.6) are the same and defined as follows:

$$4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + w_1 + w_2 + \ldots + w_m.$$

So, we can see that robot load/unload times and robot travel times are the same in $T_{C_2^m}$ and in $T_{C_3^m}$. The waiting times are the same in $T_{C_2^m}$ and in $T_{C_3^m}$ and defined as $w_i = max\{0, P_i - v_i\}$ where $v_i = (4m-4)\epsilon + (\lceil m(m+2)/2 \rceil + m + 1 - 2min\{i, m+1-i\})\delta + w_1 + w_2 + \ldots + w_m - w_i$ for all machines. So, the cycle time of C_2^m and C_3^m are equal for any processing time.

There are two different total waiting time results and the sufficient conditions for these cases are determined as follows:

1. If $P_i \leq v_i$ for $\forall i \in [1, \ldots, m]$, then $w_i = 0$, for $i = 1, \ldots, m$
2. Else if $\exists k \in [1, \ldots, m]$ such that $v_k < P_k$, then $w_k = P_k - v_k = P_k - (4m-4)\epsilon - (\lceil m(m+2) \rceil + m + 1 - 2min\{k, m+1-k\})\delta - \sum_{i \neq k} w_i$. Hence,

$w_1 + w_2 + \ldots + w_m = P_k - (4m-4)\epsilon - (\lceil m(m+2)/2 \rceil + m + 1 - 2min\{k, m + 1 - k\})\delta$.

So, $w_1 + w_2 + \ldots + w_m = max\{0, max\{P_k - (4m-4)\epsilon - (\lceil m(m+2) \rceil + m + 1 - 2min\{k, m+1-k\})\delta\}$ and the cycle time is obtained by replacing the total of waiting time in equations (5.5) and (5.6) with this max function. □

In the next theorem, the cycle time lower bound for pure cycles in robot centered cells is determined when a processing time vector is given.

Theorem 5.4. *For an m-machine robot centered cell with controllable processing times, the cycle time of any pure cycle is no less than:*

$$\underline{T_{I/0,contr}} = max\{4m\epsilon + \lceil m(m+2)/2 \rceil \delta, 4\epsilon + 2max\{min\{i, m+1-i\}\delta + P_i, i : 1, \ldots, m\}\}. \tag{5.8}$$

Proof. From the definition of pure cycles, we determined that the cycle time of a pure cycle has to be at least equal to two lower bounds. The first lower bound is obtained from the exact robot activity time and the second one is obtained from the given processing time vector. Since the robot has to perform an exact set of robot activities, the total time required for these activities constitutes a lower bound. Thus, the first lower bound is obtained as follows: The set of robot activities can be analyzed in two groups and the first group is robot loading and unloading times. First, a part is taken from the I/O station (ϵ), then loaded to one of the machines (ϵ), after the processing on the machine is finished, the part is unloaded (ϵ) and dropped to the I/O station (ϵ). This makes a total of $4m\epsilon$ for a repetition of cycle. The robot travel times constitute the second group of robot activities. The robot takes a part from I/O station and travels to machine to load it ($min\{i, m+1-i\}\delta$), after the processing on the part is finished, robot unloads machine and travels to the I/O station to drop the finished part ($min\{i, m+1-i\}\delta$).

1. Suppose the number of machines is even, then the total robot travel time is calculated as:

$\sum_{i=1}^{m} min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \ldots + m\delta + m\delta + (m-2)\delta + (m-4)\delta + \ldots + 2\delta = \lceil m(m+2)/2 \rceil \delta$.

2. Suppose the number of machines is odd, then the total robot travel time is calculated as:
$\sum_{i=1}^{m} min\{i, m+1-i\}\delta = 2\delta + 4\delta + 6\delta + \ldots + (m+1)\delta + (m-1)\delta + (m-3)\delta + \ldots + 2\delta = \lceil m(m+2)/2 \rceil \delta$.

Consequently, the total of robot activities require at least $4m\epsilon + \lceil m(m+2)/2 \rceil \delta$ time units.

The second lower bound is the minimum time required between two consecutive loadings of any machine. The minimum time needed to unload machine i after loading it is P_i time units. After the processing on the part is finished, it is unloaded (ϵ), transferred to the I/O station ($min\{i, m+1-i\}\delta$), and dropped (ϵ). After that, the robot takes a new part to make the consecutive loading of machine i (ϵ), brings the new part to machine i ($min\{i, m+1-i\}\delta$) and finally loads the machine (ϵ). The total time between two consecutive loadings of machine i is at least $4\epsilon + 2min\{i, m+1-i\}\delta + P_i$. However there are m machines and the total time for consecutive loadings are different from each other. Thus, the cycle time has to be greater than or equal to the minimum time required between two consecutive loadings of any machine in the cell. So, the second lower bound of the cycle time is $4\epsilon + max\{2min\{i, m+1-i\}\delta + P_i\}, \forall i$. □

In the next lemma, the upper bound of processing time vectors of pure cycles for a given cycle time level K is determined.

Lemma 5.5. *For a given cycle time level K, the upper bound of processing time vectors in robot centered cells for pure cycles is represented as follows:*

$\overline{\mathbf{P}}(K) = (\overline{P}_1(K), \ldots, \overline{P}_m(K))$,
where $\overline{P}_i(K) = min\{P^U, K - (4\epsilon + 2min\{i, m+1-i\}\delta)\}, \forall i$.

Proof. The two bounds constraining processing time vectors are found in the following cases.

1. The processing times are less than P^U which leads to $\overline{P}_i(K) \leq P^U$, $\forall i$.

2. In addition, the processing times on machines cannot exceed a specific value, since the cycle time is bounded by K. The cycle time level K constrains the processing times. Now, we find the upper bound of processing time on machine i, P_i, for the cycle time level K. The cycle time lower bound is determined by using equation (5.8) in Theorem 5.4 which is presented as:

   $\underline{T_{I/0,contr}} = max\{4m\epsilon + \lceil m(m+2)/2 \rceil \delta, 4\epsilon + max\{2min\{i, m+1-i\}\delta + P_i, i : 1, \ldots, m\}\}$.

   Let us set the cycle time K, then cycle time is at least equal to the cycle time lower bound :

   $\underline{T_{I/0,contr}} = max\{4m\epsilon + \lceil m(m+2)/2 \rceil \delta, 4\epsilon + max\{2min\{i, m+1-i\}\delta + P_i, i : 1, \ldots, m\}\} \leq K$

   Now, the processing time upper bound for machine i is found as follows:

   $max\{2min\{i, m+1-i\}\delta + P_i, i : 1, \ldots, m\} \leq K - 4\epsilon$, then $P_i \leq K - 4\epsilon - 2min\{i, m+1-i\}\delta$, $\forall i$. This implies that:

   $$\overline{P}_i(K) \leq K - (4\epsilon + 2min\{i, m+1-i\}\delta), \forall i.$$

Hence, the $\overline{\mathbf{P}}(K)$ is upper bound of processing time vectors satisfying the two bounds described above for a given cycle time level. □

The cycle times of C_2^m and C_3^m are equal for any processing time vector as defined in Lemma 5.4. Since the set of nondominated points is calculated only according to cycle time definition as presented in the proof of Lemma 5.6 and the cycle time of C_2^m and C_3^m are the same, C_2^m and C_3^m result in the same set of nondominated processing time vectors, $\mathbf{P}^*(C_2^m|K) = \mathbf{P}^*(C_3^m|K)$. In the next lemma, the processing time vector that gives the minimum total manufacturing cost obtained from either C_2^m or C_3^m for a given cycle time level K is determined.

Lemma 5.6. *Given any feasible cycle time level K, the nondominated processing time vector of either C_2^m or C_3^m is defined as $(P_1^*, P_2^*, \ldots, P_m^*) \in (\mathbf{P}^*(C_2^m|K) = \mathbf{P}^*(C_3^m|K))$ where $P_i^* = min\{P^U, K - (4\epsilon + 2min\{i, m+1-i\}\delta)\}, \forall i$.*

Proof. For a given cycle time level K, a feasible processing time vector is composed of processing times on machines that satisfy two upper bounds.

1. All processing times are less than P^U which leads to $P_i^* \leq P^U, \forall i$.

2. In addition, the processing times are bounded such that not to violate the cycle time level K. The processing time on machine i, P_i, is constrained according to the cycle time K. By using Lemma 5.4 and fixing cycle time to K, the cycle time can be presented as follows:

   $K = 4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta + max\{0, max\{P_i - (4m-4)\epsilon - (\lceil m(m+2) \rceil + m + 1 - 2min\{i, m+1-i\})\delta, i : 1, \ldots, m\}\}$

   This leads to $P_i \leq K - (4\epsilon + 2min\{i, m+1-i\}\delta)$. Since P_i^* is feasible, $P_i^* \leq K - (4\epsilon + 2min\{i, m+1-i\}\delta), \forall i$.

The processing times on their maximum values, without violating the bounds found in the first and second arguments, compose the nondominated processing time vectors in Lemma 5.6. □

In the next example, the nondominated processing time vector of C_2^m and C_3^m for a given cycle time level is determined as defined in Lemma 5.6.

Example 5.1 There is a 5-machine robot centered cell. Let $\delta = 0.1$, $\epsilon = 0.1$, $P^L = 3.0$, $P^U = 4.5$ and $K = 5.0$. For this cycle time level, the nondominated processing time vector $(P_1^*, P_2^*, P_3^*, P_4^*, P_5^*) \in (\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0))$ is calculated by using Lemma 5.6 as follows:

$$\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0) = \begin{bmatrix} P_1^* \\ P_2^* \\ P_3^* \\ P_4^* \\ P_5^* \end{bmatrix} = \begin{bmatrix} min\{P^U, K - (4\epsilon + 2\delta)\} \\ min\{P^U, K - (4\epsilon + 4\delta)\} \\ min\{P^U, K - (4\epsilon + 6\delta)\} \\ min\{P^U, K - (4\epsilon + 4\delta)\} \\ min\{P^U, K - (4\epsilon + 2\delta)\} \end{bmatrix}$$

$$\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0) = \begin{bmatrix} min\{4.5, 4.4\} \\ min\{4.5, 4.2\} \\ min\{4.5, 4.0\} \\ min\{4.5, 4.2\} \\ min\{4.5, 4.4\} \end{bmatrix} = \begin{bmatrix} 4.4 \\ 4.2 \\ 4.0 \\ 4.2 \\ 4.4 \end{bmatrix}$$

Hence, the nondominated processing time vector obtained from C_2^5 and C_3^5 for this cycle time level is $\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0) = (4.4, 4.2, 4.0, 4.2, 4.4)$.

The next theorem presents the cycle time region where either C_2^m or C_3^m dominates the rest of pure cycles according to the bicriteria objective of this problem. The feasible cycle time region of C_2^m and C_3^m is determined by using Lemma 5.4 as $4m\epsilon + (\lceil m(m+2)/2 \rceil + m + 1)\delta \leq K$, thus we consider this cycle time region in the following theorem.

Theorem 5.5. *Whenever either C_2^m or C_3^m is feasible, they dominate all other pure cycles.*

Proof. Since $\mathbf{P}^*(C_2^m|K) = \mathbf{P}^*(C_3^m|K) = (P_1, P_2, \ldots, P_m) = \overline{\mathbf{P}}(K)$ where $P_i = min\{P^U, K - (4\epsilon + 2min\{i, m+1-i\}\delta)\}, \forall i$, there is not any other processing time vector greater than the nondominated processing time vector obtained from either C_2^m or C_3^m, thus either C_2^m or C_3^m results in the lower bound of total manufacturing cost whenever it is feasible. □

In the next example, for a given feasible cycle time level for C_2^m and C_3^m, we show that the nondominated solution of either C_2^m or C_3^m results in the lower bound of total manufacturing cost. As processing times increase the manufacturing cost decreases. Thus, in order to show that either C_2^m or C_3^m results in the lower bound of total manufacturing cost, we show that $\mathbf{P}^*(C_2^m|K)$ and $\mathbf{P}^*(C_3^m|K)$ equals to the upper bound of processing time vectors $\overline{\mathbf{P}}(K)$ at the given cycle time level.

Example 5.2 There is a 5-machine robot centered cell with the same parameters in Example 1. In that example, the nondominated processing time vector of C_2^5 and C_3^5 is calculated as $\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0) = (4.4, 4.2, 4.0, 4.2, 4.4)$. The upper bound of processing time vector for cycle time level $K = 5.0$ is calculated from Lemma 5.5 as follows:

$$\overline{\mathbf{P}}(K)=\begin{bmatrix}\overline{P}_1(K)\\ \overline{P}_2(K)\\ \overline{P}_3(K)\\ \overline{P}_4(K)\\ \overline{P}_5(K)\end{bmatrix}=\begin{bmatrix}min\{P^U,K-(4\epsilon+2\delta)\}\\ min\{P^U,K-(4\epsilon+4\delta)\}\\ min\{P^U,K-(4\epsilon+6\delta)\}\\ min\{P^U,K-(4\epsilon+4\delta)\}\\ min\{P^U,K-(4\epsilon+2\delta)\}\end{bmatrix}=\begin{bmatrix}min\{4.5,4.4\}\\ min\{4.5,4.2\}\\ min\{4.5,4.0\}\\ min\{4.5,4.2\}\\ min\{4.5,4.4\}\end{bmatrix}$$

This simply implies that:

$$\overline{\mathbf{P}}(K)=\begin{bmatrix}4.4\\ 4.2\\ 4.0\\ 4.2\\ 4.4\end{bmatrix}$$

Hence, the upper bound of processing time vectors is found as $\overline{\mathbf{P}}(K) = (4.4, 4.2, 4.0, 4.2, 4.4)$. Since the nondominated processing time vector of C_2^5 and C_3^5 are equal to the upper bound of processing time vectors $\mathbf{P}^*(C_2^5|5.0) = \mathbf{P}^*(C_3^5|5.0) = \overline{\mathbf{P}}(K)$, there is no other pure cycle that can result in less total manufacturing cost than either C_2^5 or C_3^5.

5.2.3 3-Machine Case with Controllable Processing Times

In Figure 5.4, the efficient frontier of 3-machine problem is presented. The bold lines in the figure represent the efficient frontier such that the minimum total manufacturing cost obtained at the corresponding cycle time. The cycle time lower bound of pure cycles in 3-machine cell is calculated by using Theorem 5.4 as $12\epsilon + 8\delta$. Either C_4^3 or C_{19}^3 dominates the rest in the cycle time region $12\epsilon + 8\delta \leq K < K_1$. K_1 is always between $12\epsilon + 10\delta < K_1 < 12\epsilon + 12\delta$ and the exact value of K_1 differs according to the manufacturing cost function f. At the cycle time value of K_1, the total manufacturing cost of $\mathbf{P}^*(C_4^3|K_1)$, $\mathbf{P}^*(C_{19}^3|K_1)$, $\mathbf{P}^*(C_7^3|K_1)$ and $\mathbf{P}^*(C_{15}^3|K_1)$ are equal. After that, either C_7^3 or C_{15}^3 results in the minimum cycle time for the cycle time region $K_1 < K < 12\epsilon + 12\delta$. The pure cycles C_2^3 and C_3^3 are feasible in the region $12\epsilon + 12\delta \leq K$ and from Theorem 5.5, either C_2^3 or C_3^3 dominates the rest in this region. The cycle times corresponding to the 14 feasible pure cycles when $K < 12\epsilon + 12\delta$ are presented as follows:

$T_{C_4^3} = T_{C_{19}^3} = 12\epsilon + 8\delta + P_1 + P_2 + P_3$

$T_{C_5^3} = T_{C_{18}^3} = 12\epsilon + 10\delta + P_1 + P_2 + max\{0, P_3 - 8\epsilon - 8\delta - P_1 - P_2\}$

$T_{C_7^3} = 12\epsilon + 10\delta + max\{0, P_1 - 4\epsilon - 6\delta - w_2\} + max\{0, P_2 - 8\epsilon - 6\delta - w_1 - P_3\} + P_3$

$T_{C_9^3} = T_{C_{13}^3} = 12\epsilon + 10\delta + max\{0, P_1 - 8\epsilon - 8\delta - P_3 - P_2\} + P_2 + P_3$

$T_{C_{10}^3} = 12\epsilon + 10\delta + max\{0, P_1 - 4\epsilon - 4\delta - P_3\} + P_2 + P_3$

$T_{C_{11}^3} = 12\epsilon + 10\delta + max\{0, P_1 - 8\epsilon - 8\delta - P_3 - w_2\} + max\{0, P_2 - 4\epsilon - 4\delta - w_1\} + P_3$

$T_{C_{12}^3} = 12\epsilon + 10\delta + P_1 + max\{0, P_2 - 4\epsilon - 4\delta - w_3\} + max\{0, P_3 - 8\epsilon - 8\delta - w_2 - P_1\}$

$T_{C_{14}^3} = 12\epsilon + 10\delta + max\{0, P_1 - 4\epsilon - 6\delta - P_2\} + P_2 + P_3$

$T_{C_{15}^3} = 12\epsilon + 10\delta + P_1 + max\{0, P_2 - 8\epsilon - 6\delta - P_1 - w_3\} + max\{0, P_3 - 4\epsilon - 6\delta - w_2\}$

$T_{C_{16}^3} = 12\epsilon + 10\delta + P_1 + P_2 + max\{0, P_3 - 4\epsilon - 6\delta - P_2\}$

$T_{C_{17}^3} = 12\epsilon + 10\delta + P_1 + P_2 + max\{0, P_3 - 4\epsilon - 4\delta - P_1\}$

Cycle	Robot move sequence
C_4^3	$L_1U_1L_2U_2L_3U_3$
C_5^3	$L_1U_1L_2U_2U_3L_3$
C_7^3	$L_1U_2L_2U_1L_3U_3$
C_9^3	$L_1L_3U_3L_2U_2U_1$
C_{10}^3	$L_1L_3U_3U_1L_2U_2$
C_{11}^3	$L_1U_2L_3U_3L_2U_1$
C_{12}^3	$L_1U_1L_2U_3L_3U_2$
C_{13}^3	$L_1L_2U_2L_3U_3U_1$
C_{14}^3	$L_1L_2U_2U_1L_3U_3$
C_{15}^3	$L_1U_1L_3U_2L_2U_3$
C_{16}^3	$L_1U_1L_3L_2U_2U_3$
C_{17}^3	$L_1U_1U_3L_2U_2L_3$
C_{18}^3	$L_1U_1U_3L_3L_2U_2$
C_{19}^3	$L_1U_1L_3U_3L_2U_2$

Table 5.2: The feasible pure cycles in the cycle time region $K < 12\epsilon + 12\delta$ and corresponding robot move sequences

Cycle		$\mathbf{P}^*(C_i^3\|K)$	
	Machine 1	Machine 2	Machine 3
C_4^3	$(K-12\epsilon-8\delta)/3$	$(K-12\epsilon-8\delta)/3$	$(K-12\epsilon-8\delta)/3$
C_5^3	$(K-12\epsilon-10\delta)/2$	$(K-12\epsilon-10\delta)/2$	$K-4\epsilon-2\delta$
C_7^3	$4\epsilon+6\delta$	$K-4\epsilon-4\delta$	$K-12\epsilon-10\delta$
C_9^3	$K-4\epsilon-2\delta$	$(K-12\epsilon-10\delta)/2$	$(K-12\epsilon-10\delta)/2$
C_{10}^3	a_1	a_2	a_3
C_{11}^3	$K-4\epsilon-2\delta$	$4\epsilon+4\delta$	$K-12\epsilon-10\delta$
C_{12}^3	$K-12\epsilon-10\delta$	$4\epsilon+4\delta$	$K-4\epsilon-2\delta$
C_{13}^3	$K-4\epsilon-2\delta$	$(K-12\epsilon-10\delta)/2$	$(K-12\epsilon-10\delta)/2$
C_{14}^3	$(K-4\epsilon+2\delta)/2$	$(K-12\epsilon-10\delta)/2$	$(K-12\epsilon-10\delta)/2$
C_{15}^3	$K-12\epsilon-10\delta$	$K-4\epsilon-4\delta$	$4\epsilon+6\delta$
C_{16}^3	b_1	b_2	b_3
C_{17}^3	c_1	c_2	c_3
C_{18}^3	$(K-12\epsilon-10\delta)/2$	$(K-12\epsilon-10\delta)/2$	$K-4\epsilon-2\delta$
C_{19}^3	$(K-12\epsilon-8\delta)/3$	$(K-12\epsilon-8\delta)/3$	$(K-12\epsilon-8\delta)/3$

Table 5.3: The processing times of optimum solutions of feasible pure cycles in the cycle time region $K < 12\epsilon + 12\delta$

The values of a_j, b_j and c_j for $j = 1, 2, 3$ in Table 5.3 are presented as follows:
$a_1 \leq K - 8\epsilon - 6\delta$, $a_2 \leq K - 12\epsilon - 10\delta$, $a_3 \leq K - 12\epsilon - 10\delta$,
$b_1 \leq K - 12\epsilon - 10\delta$, $b_2 \leq K - 12\epsilon - 10\delta$, $a_3 \leq K - 8\epsilon - 4\delta$,
$c_1 \leq K - 12\epsilon - 10\delta$, $c_2 \leq K - 12\epsilon - 10\delta$, $c_3 \leq K - 8\epsilon - 6\delta$.

Now, the efficient frontier according to bicriteria objective of minimizing cycle time and total manufacturing cost simultaneously for 3-machine robot centered cells is presented in Figure 5.6. In this figure, the bold lines represent the minimum total cost obtained at the corresponding cycle time for any pure cycle.

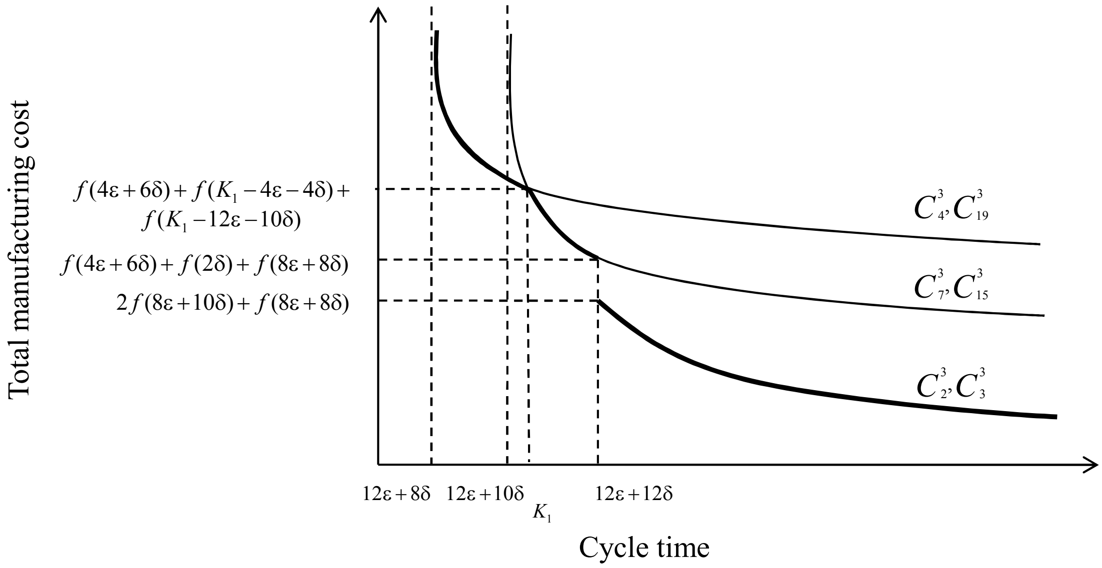

Figure 5.4: 3-machine cell with controllable processing times analysis

In this theorem, the pure cycles resulting in minimum cycle time and total manufacturing cost simultaneously and the corresponding cycle time regions where they dominate the rest of pure cycles are presented.

Theorem 5.6. *For 3-machine case*

1. *If* $K < K_1$, *then either* C_4^3 *or* C_{19}^3 *dominates the rest of pure cycles,*
2. *If* $K_1 \leq K < 12\epsilon + 12\delta$, *then either* C_7^3 *or* C_{15}^3 *dominates the rest of pure cycles,*
3. *If* $K \geq 12\epsilon + 12\delta$, *then either* C_2^3 *or* C_3^3 *dominates the rest of pure cycles,*

where K_1 *is the cycle time such that the resulting total manufacturing cost of* $\mathbf{P}^*(C_4^3|K_1)$, $\mathbf{P}^*(C_{19}^3|K_1)$, $\mathbf{P}^*(C_7^3|K_1)$ *and* $\mathbf{P}^*(C_{15}^3|K_1)$ *at* K_1 *are equal as follows:* $3f\big((K_1 - 12\epsilon - 8\delta)/3\big) = f\big(4\epsilon + 6\delta\big) + f\big(K_1 - 4\epsilon - 4\delta\big) + f\big(K_1 - 12\epsilon - 10\delta\big)$

Proof. There are 120 different pure cycles in a 3-machine cell. The cycle times and nondominated processing time vectors of all 120 pure cycles are calculated in order to compare the total manufacturing costs of these pure cycles in the specified cycle time regions.

1. The cycle times of all 120 pure cycles are calculated and it is observed that when $K < 12\epsilon + 10\delta$, only C_4^3 and C_{19}^3 are feasible pure cycles in this region. The nondominated processing time vectors of these cycles are presented in Table 5.3. In this cycle time region, from this table, the nondominated processing time vectors of C_4^3 and C_{19}^3 are equal $\mathbf{P}^*(C_4^3|K) = \mathbf{P}^*(C_{19}^3|K)$ for any given cycle time level K in this cycle time region. The total manufacturing cost of these pure cycles are calculated as total of manufacturing costs on machines as follows:

   $$F_1(C_4^3, \mathbf{P}^*(C_4^3|K)) = 3f\big((K - 12\epsilon - 8\delta)/3\big) = F_1(C_{19}^3, \mathbf{P}^*(C_{19}^3|K))$$

   Thus, they result in the same total manufacturing cost. There is not any other feasible pure cycle in this cycle time region other than C_4^3 and C_{19}^3 and these two pure cycles result in the same total manufacturing cost. Hence, there is not any other pure cycle that can result in less total manufacturing cost than C_4^3 and C_{19}^3.

2. From 120 different pure cycles, 14 of them are feasible in the cycle time region $12\epsilon + 10\delta \leq K < 12\epsilon + 12\delta$ and they are presented in Table 5.2. The nondominated processing time vectors of these 14 pure cycles for a given cycle time level K in this cycle time region are presented in Table 5.3. The total manufacturing cost of a processing time vector is equal to the sum of manufacturing costs obtained from the processing times in the vector. The total manufacturing costs of nondominated processing time vectors of C_7^3 and C_{15}^3 are calculated from $\mathbf{P}^*(C_7^3|K)$ and $\mathbf{P}^*(C_{15}^3|K)$ respectively as follows:

   $F_1(C_7^3, \mathbf{P}^*(C_7^3|K)) = f\big(4\epsilon + 6\delta\big) + f\big(K - 4\epsilon - 4\delta\big) + f\big(K - 12\epsilon - 10\delta\big) = F_1(C_{15}^3, \mathbf{P}^*(C_{15}^3|K))$.

   Thus, for any cycle time level K, the minimum total manufacturing cost obtained from C_7^3 and C_{15}^3 are equal.

   For the same cycle time level K in this processing time region, when we compare this total manufacturing cost with total manufacturing cost of

the nondominated solutions of other 12 feasible pure cycles, the result is presented as follows:
$F_1(C_7^3, \mathbf{P}^*(C_7^3|K)) = F_1(C_{15}^3, \mathbf{P}^*(C_{15}^3|K)) = f\big(4\epsilon + 6\delta\big) + f\big(K - 4\epsilon - 4\delta\big) + f\big(K - 12\epsilon - 10\delta\big) \leq F_1(C_i^3, \mathbf{P}^*(C_i^3|K))$,
where $i \in [4, 5, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19]$.

Hence, there is not any other pure cycle that can result in less total manufacturing cost than either C_7^3 or C_{15}^3 in the described cycle time region.

3. The feasible cycle time region of C_2^3 and C_3^3 is calculated from Lemma 5.1 as $K \geq 12\epsilon + 12\delta$. Hence, in the feasible cycle time region of C_2^2 and C_3^3, from Theorem 5.5, either C_2^m or C_3^m results in the lower bound of total manufacturing cost. □

5.2.4 Discussion

In this part, the machines are considered as highly flexible CNC machines such that the processing times can be controlled. The introduction of controllability to the problem puts forward the concept of total manufacturing cost minimization. In this part the problem to be solved is finding the robot move sequences minimizing the cycle time and minimizing the total manufacturing cost simultaneously. The cycle time region where either C_2^m or C_3^m results in the minimum total manufacturing cost for a given cycle time level in that region is determined. Additionally, the cycles minimizing total manufacturing costs for specified cycle time regions are determined in 3-machine robot centered cells.

Chapter 6

Conclusions and Future Work

This thesis is composed of two parts focusing on two different robotic cell layouts. In the first part, we considered an m-machine in-line robotic cell composed of CNC machines producing identical parts. The machines are assumed to be capable of performing all operations for identical parts. Since the machines are highly flexible, the processing times are assumed to be controllable. This study is restricted on a new class of cycles, called pure cycles, which result from the flexibility of machines and are easy and practical to implement. We consider the problem of finding the robot move sequence and processing times minimizing the cycle time and the total manufacturing cost simultaneously. Since the problem is a bicriteria problem, the optimal solution set is composed of nondominated solutions. The manufacturing cost is only incurred from processing times and the cost function is assumed to be strictly convex, nonlinear and differentiable. Another contribution of this study is the procedure of finding prominent pure cycles among numerous pure cycles. The two pure cycle proposed in Chapter 5 are determined by using this procedure and as explained in below, they perform efficiently in both of the two problems solved in Chapter 5.

We analyzed two specific pure cycles and determined their cycle times when a processing time vector is given in Lemmas 4.1 and 4.2. The lower bound of cycle time is determined in Theorem 4.1. Afterwards, we determined the nondominated solutions for these cycles in Lemmas 4.4 and 4.5. In Theorems

4.2 and 4.3, we proved that these two prominent cycles dominate the rest of pure cycles in the specified cycle time regions. For the remaining region, we compared the total manufacturing cost of the one of two specific cycles which is feasible in this region, to the lower bound of total manufacturing cost. The results show that these two prominent cycles are not only simple and practical, but also very efficient. As a design problem, for the two proposed cycles, we find the optimum number of machines that minimizes per unit cycle time when travel time between consecutive machines, loading/unloading times, and cycle time are given parameters in Theorems 4.4 and 4.5, respectively.

In the second part, we considered an m-machine robot centered cell producing identical parts on identical CNC machines. The input buffer and output buffer are in the same place in an I/O station. In the first part, the objective is minimizing cycle time. The processing times are assumed to be fixed and the same for every machine. We found the cycle time lower bound of pure cycles for robot centered cell in Theorem 5.1. We proposed two pure cycles and determined their cycle times for a given fixed processing time in Lemmas 5.1 and 5.2. We prove that they result in the minimum cycle time for the specified processing time region in Theorem 5.2. For the remaining processing time region, the worst case performance of these two cycles are determined in Lemma 5.3. Furthermore, we analyzed the 3-machine case and in Theorem 5.3, it is observed that the proposed pure cycles result in minimum cycle time for most of the processing time region except a small region. For the second problem, the processing times are considered as controllable processing times and the objective is to minimize the cycle time and total manufacturing cost simultaneously. The cycle times of the two proposed cycles are defined in Lemma 5.4. The cycle time lower bound is determined for controllable processing times in robot centered cells in Theorem 5.4. Consequently, the efficient set of these two proposed cycles are determined in Lemma 5.6 and compared to the lower bound of total manufacturing cost of pure cycles. These two pure cycles are proved to dominate the rest of pure cycles for the specified cycle time region in Theorem 5.5.

In this thesis, we study the problems with identical parts and the robot has a single gripper. As future research directions, the results of this study can be extended to multiple parts case or robotic cells with a dual gripper robot case. We considered that there is a single robot performing all of the required material handling activities in the cell, for the cells with a higher number of machines there can be more than one robot. Thus, multiple robot case is another future research direction. In addition, the dual gripper and the multiple robot cases are not extensively studied in the literature. For different robotic cell layouts, the same bicriteria optimization problem can be solved to determine the efficient pure cycles. Solving these problems will contribute to the literature extensively.

Bibliography

[1] E. Akçalı, K. Nemoto, and R. Uzsoy. Cycle-time improvements for photolithography process in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 14:48–56, 2001.

[2] T.P. Bagchi, J.N.D. Gupta, and C. Sriskandarajah. A review of TSP based approaches for flowshop scheduling. *European Journal of Operational Research*, 169:816–854, 2006.

[3] N. Brauner and G. Finke. Final results on the one-cycle conjecture in robotic cells. *Internal note, Laboratoire LEIBNIZ, Institut IMAG, Grenoble, France,* 1997.

[4] N. Brauner and G. Finke. Cycles and permutations in robotic cells. *Mathematical and Computer Modeling*, 34:565–591, 2001.

[5] T.C.E. Cheng, A. Janiak, M. Y. Kovalyov. Bicriterion single machine scheduling with resource dependent processing times. *SIAM Journal on Optimization*, 8:617–630, 1998.

[6] Y. Crama, J. van de Klundert. Cylic scheduling of identical parts in a robotic cell. *Operations Research*, 6:952–965, 1997.

[7] Y. Crama, J. van de Klundert. Cyclic schedulig in 3-machine robotic flowshops. *Journal of Scheduling*, 2:35–54, 1999.

[8] Y. Crama, V. Kats, J. van de Klundert, E. Levner. Cylic scheduling in robotic flowshops. *Annals of Operations Research: Mathematics of Industrial Systems*, 96:97–124, 2000.

[9] M. Dawande, H. N. Geismar, S. Sethi, C. Sriskandarajah. Sequencing and scheduling in robotic cells: recent developments. *Journal of Scheduling*, 8:387–426, 2005.

[10] M. Dawande, C. Sriskandarajah, and S. Sethi. On throughput maximization in constant travel-time robotic cells. *Manufacturing and Service Operations Management*, 4(4):296–312, 2002.

[11] P. Gilmore and R. Gomory. Sequencing a one-state variable machine: A solvable case of the travelling salesman problem. *Operations Research*, 12:675–679, 1964.

[12] H. Gultekin, O. E. Karasan, M. S. Akturk. Pure cycles in flexible robotic cells. *Computers& Operations Research*, in press, 2008a.

[13] H. Gultekin, M. S. Akturk, O. E. Karasan. Bicriteria robotic cell scheduling. *Journal of Scheduling*, in press, 2008b.

[14] H. Gultekin, M. S. Akturk, O. E. Karasan. Scheduling in robotic cells: process flexibility and cell layout. *International Journal of Production Research*, 46(8):2105–2121, 2008c.

[15] S. Gurel and M. S. Akturk. Considering manufacturing cost and scheduling performance on a CNC turning machine. *European Journal of Operational Research*, 177:325–343, 2007.

[16] S. Gurel and M. S. Akturk. Scheduling parallel CNC machines with time/cost trade-off considerations. *Computers& Operations Research*, 34:2774–2789, 2007.

[17] N. G. Hall, H. Kamoun, and C. Sriskandarajah. Scheduling in robotic cells: Classification, two- and three-machine cells. *Operations Research*, 45:421–439, 1997.

[18] N. G. Hall, H. Kamoun, and C. Sriskandarajah. Scheduling in robotic cells: Complexity and steady state analysis. *European Journal of Operational Research*, 109:43–63, 1998.

[19] 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.

[20] H. Hoogeven. Multicriteria scheduling. *European Journal of Operational Research*, 167:592–623, 2005.

[21] R. K. Kayan, M. S. Akturk. A new bounding mechanism for the CNC machine scheduling problems with controllable processing times. *European Journal of Operational Research*, 167:624–643, 2005.

[22] V. Kats, E. Levner. Cycle scheduling in a robotic production line. *Journal of Scheduling*, 5:23–41, 2002.

[23] S. Kumar, N. Ramanan, and C. Sriskandarajah. Minimizing cycle time in large robotic cells. *IIE Transactions*, 37(2):123–136, 2005.

[24] L. Lei and T.J. Wang. Determining optimal cyclic hoist schedules in a single-hoist electroplating line. *IIE Transactions*, 26(2):25–33, 1994.

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

[26] R.K. Miller. Robots in industry: applications for the electronics industry. *SEAI Institute, New York*, 1984.

[27] T. Perkinson, P. McLarty, R. Gyurcsik, and R. Cavin. Single-wafer cluster tool performance: an analysis of throughput. *IEEE Transactions on Semiconductor Manufacturing*, 7:369–373, 1994.

[28] T. Perkinson, R. Gyurcsik, P. McLarty. Single-wafer cluster tool performance: an analysis of the effects of redundant chambers and revisitation sequences on throughput. *IEEE Transactions on Semiconductor Manufacturing*, 9:384–400, 1996.

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

[30] D. Shabtay, G. Steiner. A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155:1643–1666, 2007.

[31] C. Sriskandarajah, I. G. Drobouchevitch. S. Sethi and R. Chandrasekaran. Scheduling multiple parts in a robotic cell served by a dual gripper robot. *Operations Research*, 52:65–82, 2004.

[32] V. T'kindt, J.-C. Billaut. Multicriteria scheduling: theory, models and algorithms second edition. *Springer, Berlin*, 2006.

[33] J.A. Tompkins, J.A. White, Y.A. Bozer, E.H. Frazelle, J.M.A. Tanchoco, J. Trevino. Facilities Planning. *John Wiley and Sons, New York*, 1996.

[34] R. G. Vickson. Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine. *Operations Research*, 28:1155–1167, 1980.

[35] S. Wood. Simple performance models for integrated processing tools. *IEEE Transactions on Semiconductor Manufacturing*, 9:320–328, 1996.