ENERGY CONSCIOUS SCHEDULING IN A TWO-MACHINE ROBOTIC CELL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

VAHID EGHBAL AKHLAGHI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2017

Approval of the thesis:

ENERGY CONSCIOUS SCHEDULING IN A TWO-MACHINE ROBOTIC CELL

submitted by **VAHID EGHBAL AKHLAGHI** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. M. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Murat Köksalan
Head of Department, **Industrial Engineering** ______________

Assoc. Prof. Dr. Sinan Gürel
Supervisor, **Industrial Eng. Department, METU** ______________

Assoc. Prof. Dr. Hakan Gültekin
Co-supervisor, **Industrial Eng. Department, TOBB ETU** ______________

Examining Committee Members:

Prof. Dr. Ömer Kırca
Industrial Engineering Department, METU ______________

Assoc. Prof. Dr. Sinan Gürel
Industrial Engineering Department, METU ______________

Assoc. Prof. Dr. Hakan Gültekin
Industrial Engineering Department, TOBB ETU ______________

Assist. Prof. Dr. Mustafa Kemal Tural
Industrial Engineering Department, METU ______________

Assist. Prof. Dr. Salih Tekin
Industrial Engineering Department, TOBB ETU ______________

Date: ***February 1, 2017***

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: VAHID EGHBAL AKHLAGHI

Signature :

ABSTRACT

ENERGY CONSCIOUS SCHEDULING IN A TWO-MACHINE ROBOTIC CELL

Akhlaghi, Vahid Eghbal
M.S., Department of Industrial Engineering
Supervisor : Assoc. Prof. Dr. Sinan Gürel
Co-Supervisor : Assoc. Prof. Dr. Hakan Gültekin

February 2017, 103 pages

Robotic cells usually operate under time pressure to minimize time related objectives such as cycle time, makespan, and tardiness. Robot moves constitute a significant portion of cycle time. Also, robots consume significant amount of energy determined by the speeds and distances of their moves. We study the tradeoff between the cycle time and energy consumption of a robot in a two machine flexible robotic cell which processes identical parts. In this system, the loading and unloading of machines are made by a robot. Each machine performs a different operation on each part and the robot moves linearly along a track. There are alternative cyclic schedules for such a cell and each cycle involves a number of different robot moves. Energy consumption of a robot can be formulated as a convex function of its speed, since the energy consumption increases with the speed. Given a cycle time, we find optimal speeds for different robot moves in robotic cell cycles. We determine the best cyclic schedule and the optimal robot speeds that minimize the total energy consumption. Furthermore, we generate efficient frontier for energy consumption and cycle time objectives for a number of examples.

Keywords: Scheduling, Flexible manufacturing, Robotic cells, Robot speed, Nonlinear optimization

ÖZ

İKİ MAKİNELİ ROBOTİK ÜRETİM HÜCRELERİNDE ENERJİ TABANLI ÇİZELGELEME

Akhlaghi, Vahid Eghbal
Yüksek Lisans, Endüstri Mühendisliği Bölümü

Tez Yöneticisi	: Doç. Dr. Sinan Gürel
Ortak Tez Yöneticisi	: Doç. Dr. Hakan Gültekin

Şubat 2017 , 103 sayfa

Robotik hücreler; çevrim zamanı, toplam tamamlanma zamanı ve gecikme ile ilgili amaç fonksiyonlarını en küçüklemek için zaman kısıtı altında çalışırlar. Robot hareketleri, çevrim zamanının önemli bir kısmını oluşturur. Ayrıca, robotlar kendi hareketleri arasındaki uzaklık ve hızla belirlenen yüksek miktarlarda enerji tüketirler. Bu sistemlerde, makinelerin yüklenmesi ve boşaltılması robotlar tarafından yapılır. Her makine, her parça üzerinde farklı bir operasyon yapar ve robot doğrusal bir hat üzerinde hareket eder. Bu tip bir hücrede alternatif çevrimsel çizgeler vardır ve her çevrim farklı robot hareketlerini içerir. Bir robotun enerji tüketimi, robotun hızı ile arttığı için, enerji tüketimi hızın dışbükey bir fonksiyonu olarak modellenebilir. Robot üretim hücrelerindeki farklı robot hareketleri ve verilen bir çevrim zamanı için en iyi hızları bulduk. En iyi çevrimsel çizelgeyi ve toplam enerji tüketimini en küçükleyen en iyi robot hareketlerini belirledik. Ayrıca, enerji tüketimi ve çevrim zamanı amaçlarını etkin çözümler türettik.

Anahtar Kelimeler: Çizelgeleme, Esnek imalat sistemleri, Robotik hücreler, Robot hızı, Doğrusal olmayan optimizasyon

To Baran and Bardia, who brought light to my family's life...

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to Assoc. Prof. Sinan Gürel, my supervisor. I had an enormous benefit from the numerous discussions we had, and I thank him for his patience. He provided me with unfailing guidance. His contribution is essential to the accomplishment of this work.

I am greatly indebted to my co-advisor, Assoc. Prof. Hakan Gültekin, for his constant support, guidance, encouragement and trust over the two and a half years. He has been ready to provide help, support and advice not only in academic issues but in all aspects of life, whenever I need. Without his supervision, I would not be able to manage all those. I will always need his advice throughout my entire life.

I also had the chance to benefit from the great experience of my bachelor advisor, Assoc. Prof. Mahdi Zarghami whom I would like to thank very much for the incredible insight he offered me. I have made significant academic progress with his invaluable guidance, remarks, and recommendations.

I would like to express my gratitude to my high school physics teacher, Mr. Mahmoud Pourakbari, who initiated me to the beauty of science, and his gracious wife, Neda Mosayebpour. Without their moral support, I would not be able to bear all. I am greatly indebted to them for giving me the push in the right direction when I was struggling to decide about my future.

My deepest gratitude goes to my whole family, particularly my dear parents for their endless love and unconditional support throughout my life. Special thanks are due to my elder brother who has a great impact on my personal development, my eldest brother and his wife who always strengthened my morale by standing by me in all situations.

I also would like to thank Arsham Atashi Khoei and Mohammad Saleh Farham

for their friendship, technical assistance and scientific help during my M.S. studies.

I particularly like to express my profound gratitude to my friend Raha Shabani who I see as my sister and who never leaves me alone in tougher times. Nothing gives me pleasure as much as her precious friendship.

I am also thankful to Ata Mehrad, Omid Malekpour and Shahrzad Nikghadam for their friendship. Without their help I would not be able to get admission in METU.

I would like to express my appreciation to all my friends who have contributed directly or indirectly to this thesis. I am grateful to Mohammad Salimian, Soheyl Zehtabiyan, Sanam Azadi Amin, Saeid Sattari, Reza Shabani, Siamak Gharahjeh, Majid Biazaran, Aysan Shadmand, Pooya Navid, Vahid Naseri, Mohammad Daraei, Hossein Akbaripour, and Sima Ghadirzadeh. I always feel lucky and happy to have such beautiful friends.

I am also thankful to my Turkish friends Gülten Gökayaz, İpek Eroğlu, Avni Mete Baydar, Neslihan Ayyıldız, Ali Samet Arslan, Altan Akdoğan, Mehmet Tezcan, Ali Emre Varol, and all others who helped make my stay in Turkey a very pleasant one.

I am also grateful to grants form the Scientific and Technological Research Council of Turkey (TUBITAK), grant number 215M845, for providing financial support of this research and study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

CHAPTER 1

INTRODUCTION

Industrial firms attempt to achieve the schedules of good quality, in terms of operational efficiency and custom-related objectives. Some of the most prominent objectives especially in mass production systems, are the cycle-time which is defined as the long-run average time to produce one part, the makespan which is defined as the completion time of all the parts, and the tardiness which is defined as the difference between the completion time and the due date. All of these terms allude to the amount of time it takes to get from a distinct starting point to a distinct ending point in a schedule.

Looking at the first-level supervisors, optimization in scheduling has mostly been done focusing on the system's cycle time, as mentioned by Panek et al. [1] and Kobetski and Fabian [2]. Whereas, in recent years, the energy consumption optimization became a major area of interest in industrial automation as part of the global trend, where the society is putting a growing effort for omitting the greenhouse gases and reaching an environmentally sustainable future [3, 4]. In the UK, the manufacturing industry's energy consumption accounts for 16% of the total amount. This is equal to 194 million metric tons of CO_2. In other words, it corresponds to greenhouse gas emissions from 451 million barrels of oil [5]. Generally, the industrial firms contribute to 36% of global CO_2 emissions [6]. This awareness moves high-level supervisors' attention from the optimization of the time-related objectives to the identification of a trade-off between the time-related and the energy consumption related objectives. The manufacturing energy consumption in the industrial sector has been reducing since 1998. For instance, in the US the manufacturing energy consumption decreased by 20% from 2002 to 2005 [3] and 17% from 2002 to 2010 [7]. However, an unnecessary use of energy equal to 20-40% may still be found [8]. Also in the EU, energy sustainability for industrial firms has become a major area of research and a concern for many manufacturing companies [4].

Several number of these industrial firms use automated material handling equipments in their manufacturing processes. This research focuses on a particular kind of manufacturing system called a robotic cell that utilizes a robot as a material handling equipment. The cell consists of an input device, a series of (m) processing stages (M_1, M_2, ..., M_m), an output device, and a material handling robot that can hold only one part at a time. Similarly, mechatronic industries usually apply the industrial robots as the leading component in their systems, especially for handling and processing. The repeatability, precision, speed, adaptability, as well as the efficiency of the robots are the main purposes for that decision [9]. By employing industrial robots, an industry is able to decrease the costs and increase the productivity. Recently, due to the current policy guidelines concerning greenhouse gases emissions and due to the rise of the energy prices, reducing the energy consumption of industrial robots is highly desired [10–13]. This is important due to the fact that about 8% of the total electrical energy consumption in manufacturing systems belongs to the industrial robots [14]. Hence, diminishing the energy consumption of industrial robots will automatically reduce operating costs. On the other hand, a huge potential for energy saving is within robotic cells [15–17]. Robotic cells are highly complex systems, resulting from a deep interaction between industrial, mechanical, control, and software engineering [18, 19]. Thus, an effective energy optimization method should be used to connectively exploit all of these fields. The performance of such manufacturing systems, including energy consumption rates, result from the integration of all different research fields entailed [19].

In this study, we consider a two-machine robotic cell as illustrated in Figure 1.1. The robot moves linearly along a track (linear layout) and the system follows the flow shop assumption which means that each part goes through the same sequence of machines ($M_1 - M_2$). However, the sequence of robot activities may be different. One of the decisions is to find the sequence of these robot activities. In this problem, a cycle is specified by a repeatable sequence of robot moves that consists of loading/unloading of the machines and part transfers between the machines. In a cyclic production approach, the robot starts from an initial position with a given state of the machines (loaded or unloaded) and follows a sequence of moves. At the end, the position of the robot as well as the machine states returns to the initial status. This fixed sequence of robot activities is called a cycle. Since the initial and the final states are the same, a cycle can be repeated indefinitely. If k parts are produced in a single repetition of a cycle, then it is called a k-unit cycle. 1-unit cycles derived special attention in the literature since they are simple, practical, easy to control and implement [20]. Furthermore, 1-unit cycles are optimal, i.e. give the minimum cycle time, for two-machine and three-machine robotic cells [20, 21] producing identical parts. They perform efficiently in larger robotic cells [22]. Moreover, the cyclic production provides a variety of advantages, such as shorter change-over times, higher line speeds, reliable production, reliable deliveries, etc. [23].

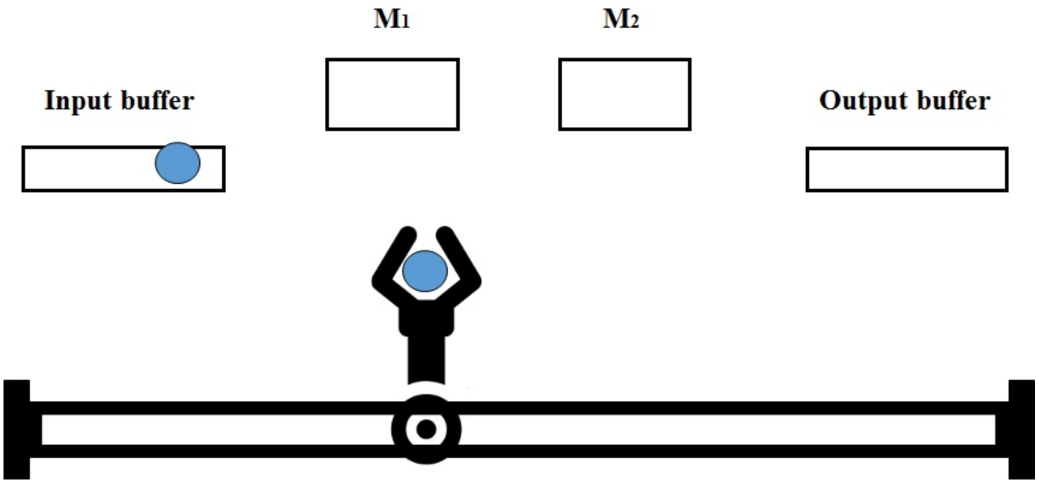


Figure 1.1: A single gripper robotic cell with two machines

In theory and practice of robotic cell scheduling problems, it is often assumed that a robot performs the operations at its maximal speed in order to achieve the maximum throughput (minimum cycle time). However, this may not be the most efficient energy consumption policy; at its maximum speed, the energy consumption is also at its maximum. On the other hand, the robot may need to wait in front of some machines to unload them because of reaching them earlier than necessary (before the processing is completed). Accordingly, there is a considerable potential for energy saving [24]. Reducing these losses can be accomplished by developing energy-efficient alternative equipments or introducing new energy saving technologies. However, energy consumption can also be reduced by just rethinking how energy is used to perform the operations, which is the focus of this thesis.

In many manufacturing systems, e.g. robotic cells, the consumption can be classified as "process energy" or "auxiliary energy" [25]. The process energy, is the energy directly used in the manufacturing, such as casting, fabrication, and assembly (welding, soldering or other fastening methods). The auxiliary energy is the energy required by the operations that allow the accomplishment of the process, such as the energy consumption of the robot moves. This paper considers the minimization of auxiliary energy consumption which deserves to be studied [8]. In our cell, the robot does not perform any industrial process, whereas the transportation of the parts and loading/unloading of the parts are performed by the robot. Barili et al. [26] described the concept of controlling the speeds to save energy for a mobile robot. Our study includes studying the permutation of robot activities in robotic cells as well as a set of optimal speeds in the schedule. However, the former work considered the path plan of robots that should travel from a source to one or more destinations and did not discuss the relationship between path planning and speed control. To take into account both energy and business objectives (e.g. cycle time, makespan, etc.) at the same time, a multi-objective optimization approach might be helpful.

The present work handles a bicriteria flow shop scheduling problem for optimizing the cycle time and energy consumption of the robot at the same time. We compare the energy consumption of different cycles with each other to determine the best in terms of both objectives. The subject is to schedule industrial robot moves, i.e. analysis of energy consumption corresponding to the speeds of different robot moves within a given cycle time upper bound. The rest of the thesis is organized as follows. Chapter 2 reviews the relevant literature. Chapter 3 develops the mathematical model and presents the proposed solution procedure. The experimental study followed by the discussion of the results are presented in Chapter 4. For given ranges of cycle times, we plot the pareto frontier with the set of non-dominated solutions for different cases and present the average percentage of energy consumption reduction through our computational study. Finally, Chapter 5 concludes the work and identifies future studies.

CHAPTER 2

LITERATURE REVIEW

As we described in the previous chapter, our robotic cell has a single robot that moves linearly and can hold only one part at a time. Also, we have only one machine per stage and no buffer for intermediate storage among them. However, there is a variety of different cell configurations or characteristics for machine environments. Therefore, before reviewing the problems considered in the literature, we need to briefly introduce all different classifications for robotic cell environments.

Robotic Cell Environments

Robotic cells can be classified in to three categories depending on the number of machines, number of robots and their types, pickup criterion and travel-time metric of the corresponding manufacturing system, as well as the number of part-types. In the following subsections, we describe all different robotic cell classifications that are suggested by Dawande et al. [27].

Number of Machines

If each processing stage has only one machine, the robotic cell is called a simple robotic cell. Such a cell contrasts with a robotic cell including parallel machines, in which at least one of the workstations has two or more identical machines. The scheduling of flow shops including parallel machines referred to as the hybrid flow shop as well. In our problem, each stage has only one machine and as it is shown in Figure 1.1, the input and output devices are on the opposite sides of the cell. Whereas, in some implementations, the input and output devices are at the same place.

Number of Robots

Manufacturers may utilize additional robots to increase the throughput rate of the cell by increasing the material handling capacity. Cells with a single robot are called single-robotic cells, while a cell with more than one robot is called a multiple-robotic cell. In this case, the robot moves must be selected in order to avoid a collision. In this study, we focus on a single-robotic cell environment.

Types of Robots

A single-gripper robot can hold only one part at a time. In contrast, a dual-gripper robot can hold two parts simultaneously. On the other hand, a self-buffered robot has a single gripper but can utilize its own buffer space to transport multiple parts. In our case, in which a bufferless single-gripper robot is employed, the robot cannot unload a part from machine M_i, $\forall i = 0, ..., m$ (where m is the total number of machines. The input and output buffers are denoted by 0 and $m + 1$ respectively), unless the next machine M_{i+1} is empty. There is another type of robot called a dual arm robot that has two arms in opposite directions moving simultaneously and a single gripper on both arms.

Pickup Criterion

As mentioned before, we focus on bufferless robotic cells. For such cells, all parts must be either on the input/output buffer, on one of the machines, or with the robot. Based on the pickup criterion, robotic cells are classified into three types: free pickup (blocking), no-wait, and interval. In this study, we consider the free pickup criterion, in which after the processing on a machine is completed, the part can wait on the machine indefinitely. Whereas in no-wait cells, a part must be removed from a machine as soon as the machine completes processing that part. This type of manufacturing is typical of steel manufacturing or plastic molding, where the materials should stay in a certain temperature. No-wait cells are also typical of food canning to assure freshness [28–33]. For interval robotic cells, there is a processing time window for each machine on which a part can be processed. Namely, the robot has a specific interval of time to unload a part after the machine finishes processing.

Travel-Time Metric

The robot's travel time between machines greatly influences a cell's performance. One common model often applies when the machines are arranged in numeric order in a line (as shown in Figure 1.1). The robot's travel time between any two machines M_i and M_j, $0 \leq i < j \leq m+1$, denoted $d(M_i, M_j)$, equals $d(M_i, M_{i+1}) + d(M_{i+1}, M_{i+2}) + ... + d(M_{j-1}, M_j)$. This type of robot travel time is called the additive travel time. However, if the travel time between all adjacent machines M_{i-1} and M_i are identical and equals δ, the travel time between any two machines M_i, M_j is $d(M_i, M_j) = |i-j|\delta$ which is called the identical distance case. Another type of travel time type considers a constant (δ) time value between any pair of machines. Finally, the general type, which is considered in this thesis, is neither additive nor constant. In other words, the robot travel times between any two machines M_i and M_j is equal to specific value δ_{ij}. For this case, it can be assumed that the triangular equality holds. That is $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$ $\forall k \in \{0, 1, ..., m+1\}$. In the most general case, this assumption is also relaxed. In this thesis, our numerical studies consist of both of the cases, with or without this assumption.

Number of Part-Types

A cell producing identical parts is referred to as a single-part-type cell. In this case, there is no part sequencing problem and the only decision is the sequence of robot moves. In contrast, a multiple-part-type cell processes lots that contain different types of parts. Generally, these different parts require different processing times on a given machine. In multiple-part production under cyclic scheduling assumption, the parts are produced as a Minimal Part Set schedule (MPS). An MPS is the smallest possible set of production parts that jointly match, in proportion, the total product sales mix [34]. The cell under consideration of this work processes identical parts.

Robotic Cell Scheduling Literature

The majority of studies on robotic cells have been performed since 1990. Some of the most beneficial and comprehensive surveys are done by Crama et al. [35], Hall and Sriskindarajah [31], and Dawande et al. [36].

In one of the earliest studies, Baumann et al. [37] derived models to analyze resource utilization for an application in which machines are served by a robot. Multiple robotic cells subject is studied by Medeiros et al. [38] and Nof and Hannah [39]. A large number of early studies applied simulation approaches to handle time-related objectives. By the use of simulation, Kondoleon [40] studied the effects of different layout configurations on the cycle time. Claybourne [41] analyzed the effects that sequencing robot activities has on throughput.

In the next section, we introduce the two possible 1-unit cycles (S_1 and S_2) in a two-machine cell and explain them in details. These cycles are constructed by Blazewicz et al. [42]. They presented an analytical approach to derive cycle time formulas for robotic cells. Later, Sethi et al. [20] provided analytical solutions to robot move sequencing concerning two or three-machine cells that produce identical parts. Logendran and Sriskandarajah [43] generalized this work to three different robotic cell layouts: the robot-centered cell, the mobile-robot cell, and the in-line robot cell. In the robot-centered cell, the machines are set in the arc of a circle and the robot will be incorporated at the center of the circle. In the mobile-robot cell, a transportation mean, such as floor-mounted or overhead rail system, is provided to help the robot move along a linear track. In an in-line robot cell, the robot is placed along an in-line material transport system. Brauner and Finke [44–47] performed several studies to compare different 1-unit cycles in robotic cells. Crama and van de Klundert [21] studied the cycles in an additive travel-time single-gripper robotic cell and proved that the robot move sequencing problem in a flowshop, which produces identical parts, is NP-hard. Dawande et al. [22] and Brauner et al. [48] proved the same for constant travel-time and general travel-time robotic cells, respectively. However, in all of these studies, they only consider the identical part-type cell in which all of the parts are same and the focus is on the robot move sequence. Hall et al. [49, 50] and

Sriskandarajah et al. [51] studied part sequencing problems in multiple part-type robotic cells. The majority of these works focused on blocking cells. They analyzed the complexity of the problem and provided exact (if possible) and heuristic approaches.

Akturk et al. [52] studied the two-machine blocking robotic cell scheduling problem in which the processing times are decision variables. They determined the parameter regions and corresponding robot move cycles that minimize the cycle time. Gultekin et al. [53] considered the same problem with tooling constraints in which some operations can only be executed on the first machine while some others only on the second machine. Consequently, the problem was assigning the remaining operations and finding the optimal robot move cycle. They proved that the optimal solution is either a 1-unit or a 2-unit robot move cycle. Levner et al. [33] proposed an algorithm to find the optimal cycle-time in a no-wait identical part-type robotic cell. Agnetis [28] worked on no-wait cells with two or three machines. For a three machine case, Agnetis and Pacciarelli [29] studied the complexity of part sequencing problem in a no-wait cell. They proved that two cycles out of the six feasible ones are unary NP-complete. Lei and Wang [54] developed a branch and bound algorithm for interval robotic cells. Chen et al. [55] and Chen et al. [56] applied branch and bound, linear programming, and bi-valued graphs to find the optimal 1-unit cycles, and multi-unit cycles, respectively. In short, for a two-machine robotic flowshop, the robot move sequencing problem is solvable in polynomial time [20]. For a three-machine cell producing identical parts, the problem is polynomial. However, for a three-machine case producing multiple parts, it is strongly NP-hard [49]. On the other hand, in an m-machine flowshop, where $m \geq 2$, part sequencing problems associated with exactly $2m-2$ of the $m!$ given robot cycles are polynomially solvable, while the remaining cycles are unary NP-hard [51]. In identical parts, 1-unit cycles are optimal for two and three-machine cells. Finding the best 1-unit cycles in an m-machine cell is polynomial-time solvable [20].

According to the literature, it is easy to see that robotic cells are typically designed under time pressure with the only optimization criterion to satisfy the production cycle time. However, while concentrating on minimizing the cycle time, they pay no attention to the speed of the robot. Often, they set the robots to run at their maximum speed which is undesirable since it tends to increase energy consumption. Improvement of this situation as well as the cycle time minimization are the main concerns of this work.

Robot Speed Energy Consumption Literature

In the last decade, caused by recognition of necessity in reducing energy consumption as a result of energy shortage and its cost rise, the number of researches on effective energy management and energy-efficient scheduling in manufacturing environments has increased.

The importance of the energy consumption reduction in robotic operations is accentuated by Rehman et. al. [57]. They handle the optimal placement of a path in the robot workspace so that the energy consumption is minimized coping with geometric, kinematic, and dynamic constraints. Bryan et. al. [58] defined a method to find the optimal robot speed and acceleration while moving from a starting point to a target point to reduce the energy consumption in the desired time. Likewise, the effect of robot movement parameters on energy consumption is presented, focusing on robot's optimal speed, acceleration, and jerk by Smetanova [59]. A more comprehensive study is done by Meike and Ribickis [60] with reference to the automobile industry, where the energy consumed by the robotic application is about 8%. In this approach, several strategies to save energy for industrial payload robots are presented. All of the mentioned studies in this paragraph focus on the path planning programming, usually a point-to-point movement, to decrease the energy consumption from a mechatronics point of view. However, in our study, we consider different robot move schedules and try to minimize the energy consumption by reducing the robot speeds through the predefined moves in each cyclic schedule.

Another energy saving study in motion planning is presented by Pellicciari et. al. [61] for pick-and-place robots. In this study, they concentrate on reducing the total energy consumption by means of constant time scaling, starting from pre-scheduled trajectories. Vergnano et. al. [62] presented an evolution for this paper by minimizing the energy consumption in cooperative multi-robot manufacturing systems. The system contains multiple robots at workstations that each of them performs a set of operations on the part. The robots are synchronized through the variation of the speeds as well as acceleration along a predefined path. The task execution sequence is dynamic while moving on the path, i.e. it changes according to the minimization criterion. likewise the previous paragraph, these two studies propose algorithms for the point-to-point path planning. Kobetski and Fabian [63], proposed two methods to reduce the speeds and accelerations of mobile robots in their operations in a multi-robot manufacturing system, similar to the previous paper, by eliminating the idle times in each schedule. The methods take advantage from an overall system perspective to reduce the components wear, while the schedule is derived without concerning the energy consumptions. In comparison to our study, they reduce the energy consumption in a given working schedule, without compromising cycle time optimality.

There is a large number of energy concerned studies, considering different manufacturing settings such as parallel machines [64–67], flow shop [68–70] and hybrid flow shop [71–74]. However, they all contemplate a single energy concerned objective, whereas the focus of this thesis is a bicriteria optimization problem considering both the energy concerned and time-related objectives at the same time. Moreover, none of those studies are in the robotic cell environment. Due to these differences, we do not go into details of their results and refer to two surveys by Gahm et al. [75] and Giret et al. [76]. In the next section, we concentrate more on the review of the works that are closer to our study, i.e., energy concerned studies considering the scheduling criteria at the same time.

Bicriteria Optimization Literature in Time and Energy Related Objectives

There are several studies in the literature that reflect both time and energy related objectives simultaneously. Fang and Lin [77] studied a parallel machine scheduling problem that minimizes tardiness penalties and power consumption costs. They determined the allocation of jobs to machines and the optimal frequency for each machine-to-job pairing. Mansouri et al. [68] studied a two-machine flow shop problem. They introduced the concept of green scheduling and developed a heuristic for an optimal trade-off between the makespan and energy consumption. For the hybrid flow shop, Du et al. [78] proposed an ant colony algorithm to minimize completion time and improve the energy efficiency at the same time. Dai et al. [79] developed a hybrid meta-heuristic approach combining genetic algorithm and simulated annealing algorithm to minimize the energy consumption within a jobshop. Subai et al. [80] investigated energy reduction in the Hoist Scheduling Problem (HSP) of the surface treatment processes. HSP is the scheduling problem of hoists which transport parts between tanks in automated electroplating lines. By keeping the system productivity at the same level, they minimized the hoist idle time. Zhang et al. [81] proposed a goal programming approach, which reduces the energy consumption and improves the scheduling efficiency at the same time in a flexible manufacturing system. Mouzon and Yildirim [82] considered a bi-objective single machine problem to minimize the total tardiness and total energy consumption. Later, Yildirim and Mouzon [83] focused on the same problem but minimizing the energy consumption and the makespan. Liu et al. [84] investigated a single machine scheduling problem to minimize the total makespan and the total carbon dioxide emission. Nonetheless, for the first time in the robotic cell scheduling literature, as a bicriteria problem, Gultekin et al. [85] minimized the manufacturing cost subject to a given cycle time where the processing times of the machines are controllable. They assumed the robot speeds are not controllable. However, in this thesis, we assume the speed of the robot can be controlled and the objective is to minimize the total energy consumption subject to a given cycle time. In another bicriteria problem, Gultekin et al. [86] considered the cycle time and manufacturing cost optimization in a two-machine robotic cell. They could minimize the manufacturing cost by optimizing the cycle time. Akturk and Ilhan [87] described a single CNC machine scheduling with controllable processing times to minimize the total tardiness, tooling and machining costs at the same time. Uruk et al. [88], considered a two-machine robotic cell and found the optimal operations assignment and the processing time for each operation, so that the total manufacturing cost and makespan are minimized.

To summarize, there is no study in the literature that considers maximization of throughput (minimization of cycle time) and minimization of energy consumption in terms of robot's speed at the same time in a robotic cell manufacturing system. In the next chapter we develop the mathematical model and present our solution procedure to deal with this bicriteria problem.

CHAPTER 3

PROBLEM DEFINITION AND SOLUTION PROCEDURE

In this chapter, we first define the problem and its preliminaries. Then, we develop mathematical programming formulations for the problem. Finally, we give the proposed solution procedure. There are two objectives to minimize in this problem. One is the cycle time and the other one is the robot's energy consumption.

The two objectives are conflicting; i.e. they negatively influence each other. In other words, improving one of them will sacrifice the other one and further achievement on cycle time (energy consumption) can only be accomplished at the expense of higher energy consumption (cycle time).

In order to handle this bicriteria problem, we will use the ε-constraint method, in which one of the objectives is written as a constraint with an upper bound on its value. By utilizing different upper bounds, different non-dominated solutions are generated. In this study, we consider the cycle time objective as a constraint. Therefore, the problem becomes the minimization of the total energy consumption subject to a given upper bound for cycle time.

In the following sections, the cycle time and the robot energy consumption functions will be described in detail. There are two 1-unit cycles (S_1 and S_2) in a two machine robotic cell. The cycle time of each cycle can be calculated separately. For each cycle, in some of the robot moves, the gripper of the robot is empty. Whereas, in other moves, the gripper is full and the robot is carrying a part to load it on a machine. Therefore, the total energy consumption during one cycle is formulated as the sum of the energy consumptions for every full and empty robot moves in the cycle multiplied by energy constants that correspond to the weight and friction forces for the robot moves per unit distance. While satisfying robot move sequence in each cycle, we want to decide the speed of robot for each move. Finally yet importantly, the upper bounds for both robot's practical speed and cycle time should be satisfied. Before introducing our functions and developing the mathematical model, we present the notation of the problem.

Notation

Sets and parameters:

$i, j \in \{0, 1, 2, 3\}$: Index for machines (M), where 0 and 3 refer to the input and output buffer respectively

$h \in \{e, f\}$: Index to indicate the state of the robot: empty (e) and full (f) robot moves

ε : Loading/unloading time of the machines

$d_{i,j,h}$: Traveled distance by the robot from M_i to M_j ($i \neq j$), in state $h \in \{e, f\}$ [1]

C_e : A constant for energy consumption function. It corresponds to the weight and friction forces for the empty robot moves per unit distance

C_f : A constant for energy consumption function. It corresponds to the weight and friction forces for the full robot moves per unit distance

P_i : Processing time of a part on machine $i \in \{1, 2\}$

w_i : Robot waiting time in front of machine $i \in \{1, 2\}$

$\overline{Ct}$: Upper bound for the cycle time

[1] It should be noted that the distance from M_i to M_j is always fixed and does not depend on the gripper's status (i.e., empty or full). Nevertheless, for making the equations easier to understand, we used the third index h for this parameter.

LB/UB : Lower/upper bound for robot speeds

Decision Variables:

$v_{i,j,h}$: Speed of robot while moving from M_i to M_j $(i \neq j)$, in state $h \in \{e, f\}$

Now, by the help of the proposed notation, we can define the problem functions and the mathematical models presented in the next sections.

Cycle Time Calculation in a Two-machine Robotic Cell

To begin with, let us see how we can instruct the robot to feed the machines, assuming that the cycle always starts from the input buffer. Since we consider cyclic scheduling, this assumption does not yield loss of generality. We consider 1-unit cycles in this study and there are two 1-unit cycles (S_1 and S_2) in a two-machine robotic cell (Sethi et. al [20]). Below, we give the time for each robot move in parenthesis.

***S*₁**: The robot picks up a part from the input buffer (ε), moves to M_1 $\left(\frac{d_{01f}}{v_{01f}}\right)$, loads the part on M_1 (ε), waits until the part has been processed (P_1), unloads the part from M_1 (ε), moves to M_2 $\left(\frac{d_{12f}}{v_{12f}}\right)$, loads the part on this machine (ε), waits in front of M_2 until the part is finished (P_2), unloads the part (ε), moves to the output buffer $\left(\frac{d_{23f}}{v_{23f}}\right)$, drops the part ($\varepsilon$) and moves back to the input buffer by an empty gripper $\left(\frac{d_{30e}}{v_{30e}}\right)$. Therefore, we have totally four robot travel times that are shown in Figure 3.1 according to their order in the cycle.

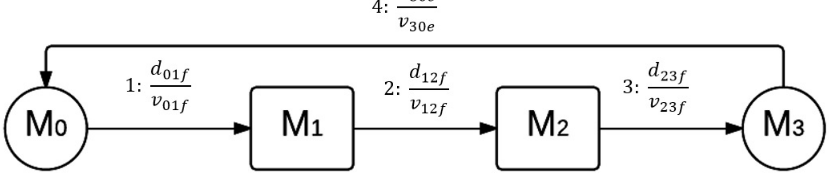

4: $\frac{d_{30e}}{v_{30e}}$
1: $\frac{d_{01f}}{v_{01f}}$
2: $\frac{d_{12f}}{v_{12f}}$
3: $\frac{d_{23f}}{v_{23f}}$
M0
M1
M2
M3

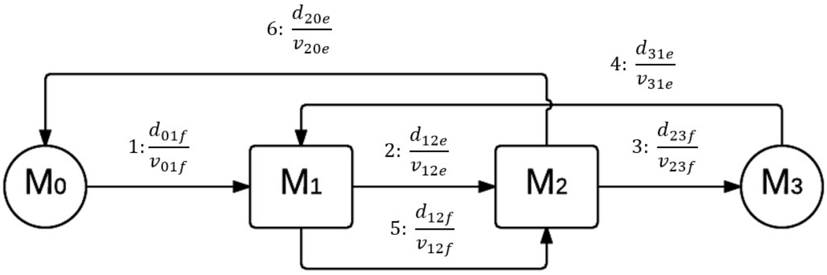
Figure 3.1: The sequence of S_1 cycle

As a result, the cycle time of S_1 cycle is

$$Ct^{S_1} = 6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} \tag{3.1}$$

It can be asserted that the set of distance indices for full and empty moves are $\mathcal{D}_1^f$= $\{(01f), (12f), (23f)\}$ and $\mathcal{D}_1^e$=$\{(30e)\}$ respectively. Therefore, the set of indices for all distances passed by the robot is $\mathcal{D}_1$=$\{(01f), (12f), (23f),(30e)\}$. Later on, we will use these sets in our formulations.

***S*₂**: In this cycle, the second machine (M_2) is full at the beginning of the cycle, i.e. at the end of each cycle, the second machine is occupied by a part. In this case, the robot picks up another part (say part 2) from M_0 (ε), moves to M_1 $\left(\frac{d_{01f}}{v_{01f}}\right)$, loads part 2 on machine M_1 (ε), moves (by empty gripper) to M_2 $\left(\frac{d_{12e}}{v_{12e}}\right)$, if necessary waits at M_2 until the previous part (part 1) is finished (w_2), unloads part 1 (ε), moves to the output buffer $\left(\frac{d_{23f}}{v_{23f}}\right)$, drops part 1 ($\varepsilon$), moves (by empty gripper) to M_1 $\left(\frac{d_{31e}}{v_{31e}}\right)$, if necessary waits at M_1 until part 2 has been processed (w_1), unloads part 2 (ε), moves to M_2 $\left(\frac{d_{12f}}{v_{12f}}\right)$, loads part 2 on M_2 (ε), moves (by empty gripper) to M_0 $\left(\frac{d_{20e}}{v_{20e}}\right)$ to pick up a new part (say part 3). Therefore, we have totally six robot travel times that are shown in Figure 3.1 according to their order in the cycle.



6: $\frac{d_{20e}}{v_{20e}}$
4: $\frac{d_{31e}}{v_{31e}}$
1: $\frac{d_{01f}}{v_{01f}}$
2: $\frac{d_{12e}}{v_{12e}}$
3: $\frac{d_{23f}}{v_{23f}}$
M0
M1
M2
M3
5: $\frac{d_{12f}}{v_{12f}}$

Figure 3.2: The sequence of S_2 cycle

Subsequently, the computation of cycle time is different from the S_1 cycle. Since, after loading each part, the robot waits in front of the machine until the end of its processing time, it can be observed that the robot has full waiting times in front of M_1 and M_2 in S_1 cycle which take P_1 and P_2 time units respectively.

However, in S_2 cycle, the robot leaves the machine after loading a part on it to perform some other operations. As a result, we can compute the cycle time for S_2 as follows:

$$Ct^{S_2} = 6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + w_2 + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + w_1 + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \tag{3.2}$$

where w_1 and w_2 are the robot waiting times in front of M_1 and M_2, respectively. It should be noticed that the processing time of the first machine (P_1) starts after robot travels d_{01f} and loads a part (ε) on M_1. If the processing time is not finished when the robot again reaches M_1 by d_{31e}, then robot waits in front of M_1. Otherwise, it is equal to zero. Therefore, w_1 is calculated as follows:

$$w_1 = \max\left\{0, P_1 - \frac{d_{12e}}{v_{12e}} - w_2 - \varepsilon - \frac{d_{23f}}{v_{23f}} - \varepsilon - \frac{d_{31e}}{v_{31e}}\right\} \tag{3.3}$$

Similarly, when the robot loads a part on M_2 by $d_{12f} + \varepsilon$, P_2 starts and we may have a waiting time if P_2 is not finished when the robot comes back at M_2 by d_{12e}. Then

$$w_2 = \max\left\{0, P_2 - \frac{d_{20e}}{v_{20e}} - \varepsilon - \frac{d_{01f}}{v_{01f}} - \varepsilon - \frac{d_{12e}}{v_{12e}}\right\} \tag{3.4}$$

It can be seen that the set of indices for full and empty robot moves are $\mathcal{D}_2^f$= {(01f), (12f), (23f)} and $\mathcal{D}_2^e$= {(12e), (31e), (20e)} respectively. Hence, the set of total indices for S_2 cycle can be declared as $\mathcal{D}_2$= {(01f), (12e), (23f), (31e), (12f), (20e)}.

In each cycle, the robot returns to the same position after completing all activities and the cell returns to its initial state as well. Therefore, it can be repeated indefinitely. Starting the cycles from a different activity does not change the cycle time. Therefore, for straightforwardness, we always assume that the robot starts the cycle by unloading a part from the input buffer and loading it on the first machine. To find out which of S_1 and S_2 cycles provides the optimal solution, the corresponding objective functions should be calculated and compared with each other. In our problem, since we have two objectives, we have to compare the cycles in terms of both objectives. For a given cycle time value, one cycle is better than the other one if it provides a smaller energy consumption value. The problem is to determine the robot speeds which minimizes total energy consumption.

Robot Energy Consumption Function

The energy consumption is evaluated according to the traveled distance by the robot. The robot travels from one machine to another along a linear track as shown in Figure 1.1. We assume that once determined, the robot travel speed between any pair of machines is constant (acceleration and deceleration are negligible). However, the speed between one pair of machines can be different from the speed between another pair of machines. The loading/unloading energy consumptions are negligible. The total robot energy consumption is the sum of energy needed for all robot moves. In some of these cases, the robot holds a part with its gripper, whereas in some others it is empty. Depending on the weight of the part, the energy consumption of empty and full moves can be different. In this study, we consider this general case. To clarify, the energy consumed at each move is calculated by the expression:

$$F(v) = Cdv^k \tag{3.5}$$

Where C is a constant corresponding the weight and friction forces. We will use C_e and C_f for empty and full movements, respectively. d is the traveled distance, v is the correlated speed and k is a constant that represents the relation between the speed of the robot and energy loss which depends on the type and the model of the robot. Namely, k can have a different value for different kinds of robots. By forcing the robot to move faster, it consumes more energy. Because it requires more energy to increase the speed. In Figure 3.3, the behavior of this function is shown. In this study, we consider $k \geq 1$. As shown in Figure 3.3, convexity holds for this function with respect to $k \geq 1$. Next, we give the math formulation for the problem and our approach is same for both linear ($k = 1$) and nonlinear objective ($k > 1$) functions.

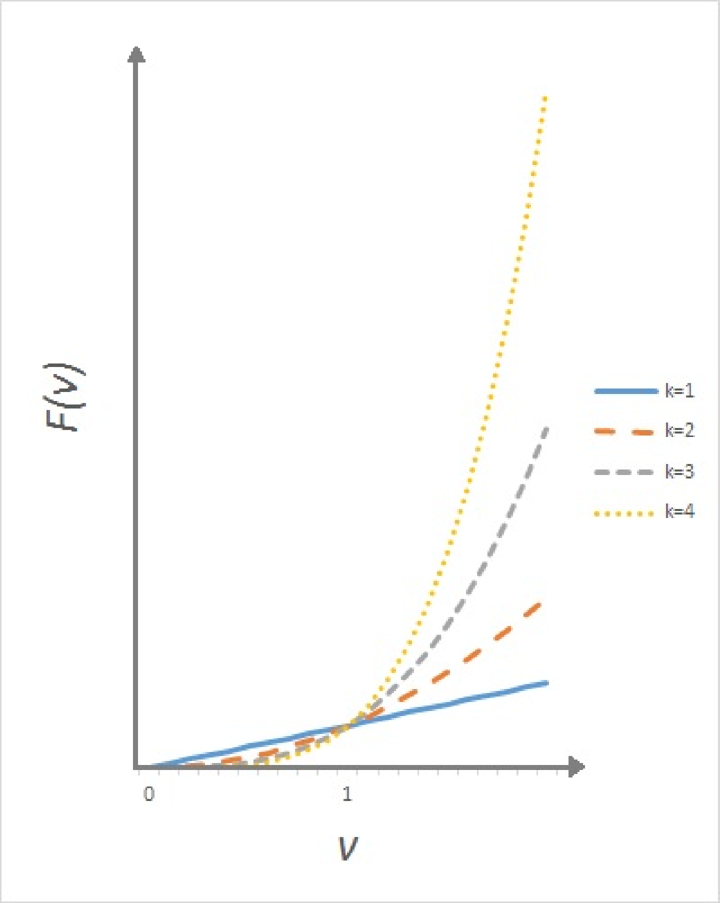


Figure 3.3: Robot energy consumption versus cycle time

Mathematical Model

The general model that we are dealing with is written as follows:

$$
\begin{aligned}
&\min \quad \underline{\text{EC (Robot Energy Consumption)}} \\
&\min \quad \underline{\text{Ct (Cycle Time)}} \\
&\text{Subject to} \\
&\qquad \text{Ct} = \min(\text{Ct}^{S_1}, \text{Ct}^{S_2}) \\
&\qquad \text{Ct}^{S_1} = 6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} \\
&\qquad \text{Ct}^{S_2} = 6\varepsilon + w_1 + w_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \\
&\qquad LB \leq v_{ijh} \leq UB \qquad\qquad\qquad \forall (ijh) \in \mathcal{D}_1 \cup \mathcal{D}_2
\end{aligned}
$$

The minimum and the maximum values for Ct^{S_1} and Ct^{S_2} occur when all of the speed values are equal to UB and LB respectively. Also, it can be seen that Ct (cycle time) is equal to the minimum of Ct^{S_1} and Ct^{S_2} values. Notice that if any of the speeds equals zero, the corresponding cycle time will be equal to infinity, which is not feasible. Consequently, we must have $LB > 0$. In order to solve this bi-criteria problem, we apply ε-constraint approach. This consists minimizing a one of the objectives and expressing the other objective in the constraint part of the model and in the form of inequality constraints. By parametrical modification in the right-hand side of the constrained objective function (cycle time upper bound), we can find the efficient solutions. Now by applying the ε-constraint approach, we can revise the general model for each cycle as follows:

S_1 Model:

$$\begin{aligned} &\min \quad \text{Robot Energy Consumption} \\ &\text{Subject to} \\ &\qquad 6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} \leq \overline{Ct} \\ &\qquad LB \leq v_{ijh} \leq UB \qquad\qquad \forall (ijh) \in \mathcal{D}_1 \end{aligned}$$

S_2 Model:

$$\begin{aligned} &\min \quad \text{Robot Energy Consumption} \\ &\text{Subject to} \\ &\qquad 6\varepsilon + w_1 + w_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct} \\ &\qquad LB \leq v_{ijh} \leq UB \qquad\qquad \forall (ijh) \in \mathcal{D}_2 \end{aligned}$$

Since the objectives are conflicting, it is clear that by increasing $\overline{Ct}$ we can decrease the energy consumption. On the other hand, we know that the speed of the robot is limited by its lower and upper bounds. Therefore the minimum and the maximum amount of energy consumption will occur in these two points and the other feasible solutions will exist between this range (Figure 3.4). However, they may be dominated by other solutions which can provide a smaller amount of energy consumption, satisfying the same cycle time. For instance, if the speed of the robot is set at its maximum value but there is a waiting time in front of a machine, we can reduce the speed of the robot to be in front of the machine later. As a result, we reduce the energy consumption that correspond to that move and consequently the total robot energy consumption. In this chapter, we will see how the robot can perform the operations by slower moves while satisfying the same cycle time in a number of problems.

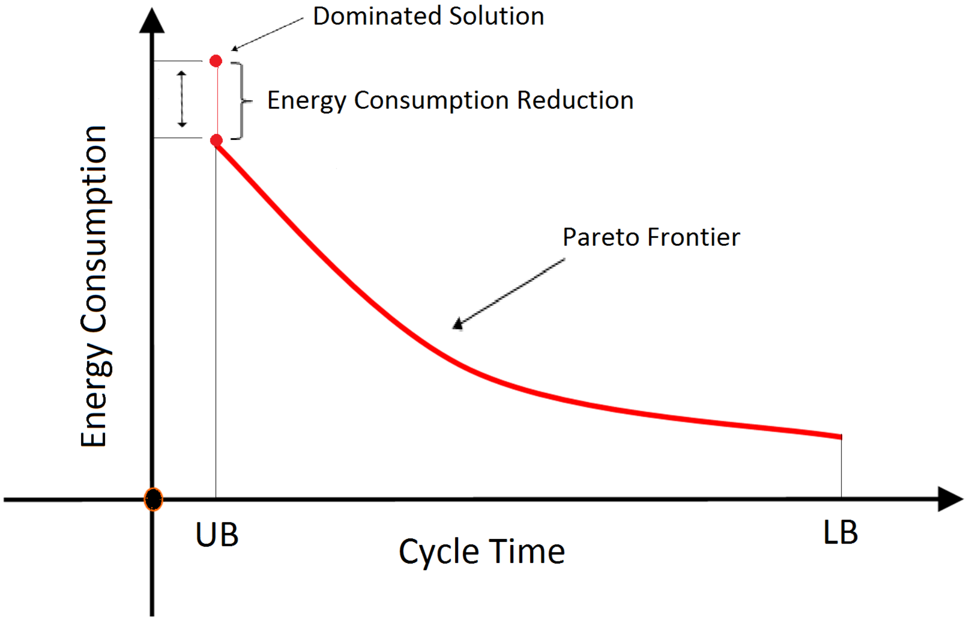


Figure 3.4: Robot energy consumption versus cycle time

As mentioned before, we have only two cycles (S_1 and S_2) in a two-machine robotic cell and it is proved by Sethi et. al. [20] that they are the only two possible cycles for this problem. Also, they proved that 1-unit cycles are optimal for two-machine cells producing identical parts. Since the set of distances in each cycle is different, two models are developed separately. In the following section, we will consider S_1 cycle.

Mathematical Model for the $\mathbf{S_1}$ Cycle:

As mentioned in the previous section, the cycle time of S_1 is calculated as $6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}}$. The formulation that minimizes the total energy consumption with respect to a given cycle time upper bound can be developed as follows:

Formulation 1 (S$_1$ Model)

$$\begin{aligned}
&\text{Minimize } C_f\left[d_{01f}v_{01f}^k + d_{12f}v_{12f}^k + d_{23f}v_{23f}^k\right] + C_e\left[d_{30e}v_{30e}^k\right] \\
&\text{Subject to} \\
&6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} \le \overline{Ct} && (3.6) \\
&LB \le v_{ijh} \le UB \qquad \forall (ijh) \in \mathcal{D}_1 && (3.7)
\end{aligned}$$

The objective function is derived from formula (3.5). The full and empty moves are separated due to their corresponding constants C_e and C_f. According to the ε-constraint approach, the calculated cycle time, should be less than or equal to the cycle time upper bound ($\overline{Ct}$) which is satisfied by the first constraint (3.6). Also, all of the speeds should be within their bounds (3.7). We assume speed bounds are the same for all moves.

Note that, if $\overline{Ct}$ value in constraint (3.6) is smaller than a threshold value depending on ε, P_1, and P_2, this constraint can never be satisfied. In other words, $\overline{Ct} > 6\varepsilon + P_1 + P_2$ must hold. Otherwise, there is no feasible solution in S_1 cycle. On the other hand, the lower and upper bounds of speeds depend on technological constraints. It is not possible to increase the speeds indefinitely. Meaning that there must be an upper bound. In this study, we assume $LB = 0$. Nevertheless, it should be noticed that if any of the speeds equals to zero, the cycle will not be completed at all, which is infeasible. As a consequence of the nonlinear objective and nonlinear constraints, this is a Nonlinear Programming Problem (NLP). In order to solve this, the Karush–Kuhn–Tucker (KKT) conditions are applied. We will first construct the Lagrangian function as follows:

$$\begin{aligned}
L(v_{ijh}, \mu) = C_f\left[d_{01f}v_{01f}^k + d_{12f}v_{12f}^k + d_{23f}v_{23f}^k\right] + C_e[d_{30e}v_{30e}^k] \\
+\mu_1\left[6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} - \overline{Ct}\right]
\end{aligned}$$

Here, we initially ignore the bounds of the speeds and consider them later in section 3.4. As a consequence of the KKT conditions, either constraint (3.6) is tight and the corresponding Lagrangian multiplier $\mu_1 \ge 0$, or it is loose and $\mu_1 = 0$. In the optimal solution, for the speed of every robot move, we must have $\dfrac{\partial L}{\partial v_{i,j,h}} = 0$. So:

$$\frac{\partial L}{\partial v_{01f}} = kC_f d_{01f} v_{01f}^{k-1} - \mu_1 \frac{d_{01f}}{v_{01f}^2} = 0 \tag{3.8}$$

$$\frac{\partial L}{\partial v_{12f}} = kC_f d_{12f} v_{12f}^{k-1} - \mu_1 \frac{d_{12f}}{v_{12f}^2} = 0 \tag{3.9}$$

$$\frac{\partial L}{\partial v_{23f}} = kC_f d_{23f} v_{23f}^{k-1} - \mu_1 \frac{d_{23f}}{v_{23f}^2} = 0 \tag{3.10}$$

$$\frac{\partial L}{\partial v_{30e}} = kC_e d_{30e} v_{30e}^{k-1} - \mu_1 \frac{d_{30e}}{v_{30e}^2} = 0 \tag{3.11}$$

The equations (3.8) to (3.11) represent the first set of KKT conditions called stationary condition. We also have the following KKT optimality conditions.

Primal feasibility:

$$6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} \leq \overline{Ct}$$

Dual feasibility:

$$\mu_1 \geq 0$$

Complementary slackness:

$$\mu_1 \left[6\varepsilon + P_1 + P_2 + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{30e}}{v_{30e}} - \overline{Ct} \right] = 0$$

We know that a differentiable function is convex on an interval if and only if its derivative is monotonically non-decreasing there. Also, a twice differentiable function is convex on an interval if and only if its second derivative is non-negative on that interval. Since in this problem we consider $k \geq 1$ and the variables are in the interval $(0, UB]$, the objective functions are convex. The reason is that its first derivative is non-decreasing and second derivative in non-negative in the interval $(0, UB]$. Also, the summation of a number of convex functions is a convex function as well. As a result, since the constraints are additively separable and each separable part ($\dfrac{d_{ijh}}{v_{ijh}}$, $\forall v_{ijh} \in (0, UB]$) defines a convex set, the constraints are also convex. Due to the convexity of the problem, the KKT conditions yield the global optimum solution.

As mentioned before, when all of the speed values are equal to UB, we will achieve the minimum possible cycle time value (Ct_{min}). Conversely, if they are equal to LB, we will have the maximum possible cycle time value (Ct_{max}) for the problem. By the help of this explanation, the following lemma proves that the first constraint of Formulation 1 is always tight.

Lemma 1. *If $Ct_{min} \leq \overline{Ct} \leq Ct_{max}$, i.e. Formulation 1 has a feasible solution, then the optimal cycle time is equal to $\overline{Ct}$ and constraint* (3.6) *is tight.*

Proof. If constraint (3.6) is not tight, then $\mu_1 = 0$ must hold. When $\mu_1 = 0$, equations (3.8)-(3.11) yield all speeds to be 0. However, this solution does not satisfy constraint (3.6), therefore it is infeasible. □

In reality, we mostly possess different energy constants for empty and full moves ($C_f > C_e$) of the robot. For different constants, different results may appear. It is important to consider both identical and different cases in optimal situations. In the following theorem, we prove that the only empty robot move of the cycle is faster than or as fast as the full moves when $C_f \geq C_e$.

Theorem 1. *If $C_f \geq C_e$, then in the optimal solution we have $v_{30e} \geq v_{01f} = v_{12f} = v_{23f}$ and the optimal speeds are*

$$v^*_{01f} = v^*_{12f} = v^*_{23f} = \frac{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_f}(\overline{Ct} - 6\varepsilon - P_1 - P_2)}$$

$$v^*_{30e} = \frac{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_e}(\overline{Ct} - 6\varepsilon - P_1 - P_2)}$$

Proof. By considering $\dfrac{\partial L}{\partial v_{01f}} = 0$ in (3.8) we have

$$kC_f d_{01} v_{01f}^{k-1} = \mu_1 \frac{d_{01}}{v_{01f}^2}, \text{ then}$$

$$v_{01f}^{k+1} = \frac{\mu_1}{kC_f}, \quad \text{or} \quad v_{01f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}}$$

By the same method, the value of all other speeds will be as follows:

$$v^*_{01f} = v^*_{12f} = v^*_{23f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}} \tag{3.12}$$

$$v^*_{30e} = \sqrt[k+1]{\frac{\mu_1}{kC_e}} \tag{3.13}$$

Since $C_f \geq C_e$, from equations (3.12) and (3.13) we can conclude that $v_{30e} \geq v_{01f} = v_{12f} = v_{23f}$.

In order to simplify finding the value of μ_1, we can define $z = \frac{1}{\sqrt[k+1]{\mu_1}}$. Then, by putting z in constraint (3.6), we have:

$$z\left(\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh}\ \sqrt[k+1]{kC_f}\right) + z\left(\sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh}\ \sqrt[k+1]{kC_e}\right) = \overline{Ct} - 6\varepsilon - P_1 - P_2, \text{ or}$$

$$z\left(\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh}\ \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh}\ \sqrt[k+1]{kC_e}\right) = \overline{Ct} - 6\varepsilon - P_1 - P_2 \text{ then}$$

$$z = \frac{\overline{Ct} - 6\varepsilon - P_1 - P_2}{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh}\ \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh}\ \sqrt[k+1]{kC_e}},$$

Finally μ_1 can be found as follows:

$$\mu_1 = \left(\frac{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh}\ \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh}\ \sqrt[k+1]{kC_e}}{\overline{Ct} - 6\varepsilon - P_1 - P_2}\right)^{k+1}$$

Now, by finding μ_1 and substituting its value in equations (3.12) and (3.13), the optimal speed values are achieved and this theorem is proven. Since our problem is convex and the solution satisfies KKT conditions, there is no doubt about the optimality of the ascertained speed values in this theorem. □

In some practical situations, the effect of the weight of the produced parts is negligible. Therefore, the following corollary considers $C_f = C_e$ case. However, the general case is always $C_f \geq C_e$, so we do not consider the other situation out of this condition (i.e., $C_f < C_e$). Also, the following corollary asserts that we do not need to find the value of μ_1, if $C_e = C_f$.

Corollary 1. *If $C_e = C_f$, at optimality, we have $v_{01f} = v_{12f} = v_{23f} = v_{30e}$. That is, the speed of the robot is the same through the cycle for all of the moves.*

Proof. In equations (3.12) and (3.13), if we consider $C_f = C_e = C$, we have

$$v^*_{01f} = v^*_{12f} = v^*_{23f} = v^*_{30e} = \sqrt[k+1]{\frac{\mu_1}{kC}} \tag{3.14}$$

In this situation, the speeds of all robot moves are equal. Since their energy constant coefficients are the same in the objective function and there is no restriction for moving of the robot from the beginning to the end of the cycle time. From equation (3.14), by putting $\sqrt[k+1]{\frac{\mu_1}{kC}}$ in constraint (3.6), we have

$$\frac{\sum_{(ijh)\in\mathcal{D}_1} d_{ijh}}{\sqrt[k+1]{\frac{\mu_1}{kC}}} = \overline{Ct} - 6\varepsilon - P_1 - P_2, \text{ therefore}$$

$$v^*_{01f} = v^*_{12f} = v^*_{23f} = v^*_{30e} = \sqrt[k+1]{\frac{\mu_1}{kC}} = \frac{\sum_{(ijh)\in\mathcal{D}_1} d_{ijh}}{\overline{Ct} - 6\varepsilon - P_1 - P_2} \tag{3.15}$$

Needless to calculate μ_1, equation (3.15) yields identical speed values for all robot moves. □

In this section, we achieved the optimal speed values for the S_1 cycle. In the next section, we start to study the S_2 cycle.

Mathematical Model for the $\mathbf{S_2}$ Cycle:

In this section, at first we develop a mathematical model for the S_2 cycle. Then, by utilizing the KKT conditions, we analyze different situations and determine the optimal speed values for each robot move in the S_2 cycle.

By utilizing (3.3) and (3.4) in (3.2), we obtain

$$\begin{aligned} 6\varepsilon &+ \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \\ &+ \max\left\{0, P_1 - \frac{d_{12e}}{v_{12e}} - \varepsilon - \frac{d_{23f}}{v_{23f}} - \varepsilon - \frac{d_{31e}}{v_{31e}},\right. \\ &\qquad \left. P_2 - \frac{d_{20e}}{v_{20e}} - \varepsilon - \frac{d_{01f}}{v_{01f}} - \varepsilon - \frac{d_{12e}}{v_{12e}}\right\} \le \overline{Ct} \end{aligned} \tag{3.16}$$

In the mathematical model, we will minimize the total energy consumption subject to the constraint that the cycle time given in equation (3.16) is less than or equal to the upper bound of the cycle time. However, in order to linearize the max term in this equation, we write three separate constraints as follows:

$$6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct}$$
$$P_1 + 4\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct}$$
$$P_2 + 4\varepsilon + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} \leq \overline{Ct}$$

Now, the mathematical model can be written as follows:

Formulation 2 ($\mathbf{S_2}$ Model):

$$\text{Minimize } C_f[d_{01f}v_{01f}^k + d_{23f}v_{23f}^k + d_{12f}v_{12f}^k] + C_e[d_{12e}v_{12e}^k + d_{31e}v_{31e}^k + d_{20e}v_{20e}^k]$$

Subject to

$$6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct} \tag{3.17}$$
$$P_1 + 4\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct} \tag{3.18}$$
$$P_2 + 4\varepsilon + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} \leq \overline{Ct} \tag{3.19}$$
$$LB \leq v_{ijh} \leq UB \forall (ijh) \in \mathcal{D}_2 \tag{3.20}$$

In the objective function, the energy consumption function of each move is multiplied by the corresponding energy constant. Constraints (3.17)-(3.19) are the linearized version of the cycle time upper bound and the bounds of the variables are considered in the last constraint (3.20). Not that $\overline{Ct} > \max\{6\varepsilon, P_1{+}4\varepsilon{+}P_2{+}4\varepsilon\}$ must hold. Otherwise, there is no feasible solution in S_2 cycle.In order to solve this NLP, we will again develop the Lagrangian function as follows:

$$\begin{aligned}
&L(v_{ijh}, \mu) = \\
&C_f\left[d_{01f}v_{01f}^k + d_{23f}v_{23f}^k + d_{12f}v_{12f}^k\right] + C_e\left[d_{12e}v_{12e}^k + d_{31e}v_{31e}^k + d_{20e}v_{20e}^k\right] \\
&\qquad +\mu_1\left[6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} - \overline{Ct}\right] \\
&\qquad +\mu_2\left[P_1 + 4\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} - \overline{Ct}\right] \\
&\qquad +\mu_3\left[P_2 + 4\varepsilon + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} - \overline{Ct}\right]
\end{aligned}$$

For each constraint, if it is tight then the corresponding μ value must be greater than or equal to zero, but if it is loose then the corresponding μ must be zero.

Since there are three constraints, we have a total of eight situations (2^3) for constraints being tight ($\mu \geq 0$) and loose ($\mu = 0$). Notice that for each constraint $i \in \{1, 2, 3\}$, $\mu_i \geq 0$ means the feasibility of dual variable. For our problem, where we have eight situations (subsequently we will prove that four of them are infeasible), each situation must be examined along with the feasibility of the dual variable as well as the primal. With equality of each constraint, values of primal variables and μ_i's (dual variables) are obtained.

In the optimal solution, for every variable we must have $\dfrac{\partial L}{\partial v_{i,j,h}} = 0$. So,

$$\frac{\partial L}{\partial v_{01f}} = kC_f d_{01f} v_{01f}^{k-1} - \mu_1 \frac{d_{01f}}{v_{01f}^2} - \mu_2 \frac{d_{01f}}{v_{01f}^2} = 0 \tag{3.21}$$

$$\frac{\partial L}{\partial v_{12e}} = kC_e d_{12e} v_{12e}^{k-1} - \mu_1 \frac{d_{12e}}{v_{12e}^2} = 0 \tag{3.22}$$

$$\frac{\partial L}{\partial v_{23f}} = kC_f d_{23f} v_{23f}^{k-1} - \mu_1 \frac{d_{23f}}{v_{23f}^2} - \mu_3 \frac{d_{23f}}{v_{23f}^2} = 0 \tag{3.23}$$

$$\frac{\partial L}{\partial v_{31e}} = kC_e d_{31e} v_{31e}^{k-1} - \mu_1 \frac{d_{31e}}{v_{31e}^2} - \mu_3 \frac{d_{31e}}{v_{31e}^2} = 0 \tag{3.24}$$

$$\frac{\partial L}{\partial v_{12f}} = kC_f d_{12f} v_{12f}^{k-1} - \mu_1 \frac{d_{12f}}{v_{12f}^2} - \mu_2 \frac{d_{12f}}{v_{12f}^2} - \mu_3 \frac{d_{12f}}{v_{12f}^2} = 0 \tag{3.25}$$

$$\frac{\partial L}{\partial v_{20e}} = kC_e d_{20e} v_{20e}^{k-1} - \mu_1 \frac{d_{20e}}{v_{20e}^2} - \mu_2 \frac{d_{20e}}{v_{20e}^2} = 0 \tag{3.26}$$

The equations (3.21) to (3.26) give the stationary condition. Other than these, we also have:

Primal feasibility:

$$6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct}$$

$$P_1 + 4\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} \leq \overline{Ct}$$

$$P_2 + 4\varepsilon + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} \leq \overline{Ct}$$

Dual feasibility:

$$\mu_i \geq 0 \qquad \forall i \in \{1, 2, 3\}$$

Complementary slackness:

$$\mu_1\left[6\varepsilon+\frac{d_{01f}}{v_{01f}}+\frac{d_{12e}}{v_{12e}}+\frac{d_{23f}}{v_{23f}}+\frac{d_{31e}}{v_{31e}}+\frac{d_{12f}}{v_{12f}}+\frac{d_{20e}}{v_{20e}}-\overline{Ct}\right]=0$$

$$\mu_2\left[P_1+4\varepsilon+\frac{d_{01f}}{v_{01f}}+\frac{d_{12f}}{v_{12f}}+\frac{d_{20e}}{v_{20e}}-\overline{Ct}\right]=0$$

$$\mu_3\left[P_2+4\varepsilon+\frac{d_{23f}}{v_{23f}}+\frac{d_{31e}}{v_{31e}}+\frac{d_{12f}}{v_{12f}}-\overline{Ct}\right]=0$$

The following lemma shows that the first constraint of Formulation 2 is always tight.

Lemma 2. *If there is a feasible solution to Formulation 2, then the optimal cycle time is equal to $\overline{Ct}$, i.e. constraint* (3.17) *is tight.*

Proof. If constraint (3.17) is not tight, then $\mu_1 = 0$ must hold. However, when $\mu_1 = 0$, equation (3.22) yields $v_{12e} = 0$, which is infeasible. Since otherwise the cycle time will be infinite. □

In order to find the optimal solution for the S_2 model, we need to consider the tightness and looseness of the constraints. As a consequence of Lemma 2, we know that constraint (3.17) is always tight. Consequently, all the four situations in which this constraint is not tight is diminished. Then, there remain 4 situations to be considered to determine the optimal solution. In subsequent sections of this chapter, we consider each of these situations one by one.

Situation 1: The second and the third constraints ((3.18) and (3.19)) are loose ($\mu_2 = \mu_3 = 0$)

In this section, we consider Situation 1 where only constraint (3.17) is binding. The following theorem determines the optimal speed values and also shows that the empty movements are faster than the full movements if $C_f \geq C_e$.

Theorem 2. *If $C_f \geq C_e$, in the optimal solution we have $v_{12e} = v_{31e} = v_{20e} \geq v_{01f} = v_{23f} = v_{12f}$ and the optimal speed values can be achieved by the following* *equations:*

$$v^*_{01f} = v^*_{12f} = v^*_{23f} = \frac{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_f}(\overline{Ct} - 6\varepsilon)}$$

$$v^*_{12e} = v^*_{31e} = v^*_{20e} = \frac{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_e}(\overline{Ct} - 6\varepsilon)}$$

Proof. By considering $\dfrac{\partial L}{\partial v_{01f}} = 0$ in equation (3.21) and setting $\mu_2 = \mu_3 = 0$ we have:

$$kC_f d_{01} v_{01f}^{k-1} = \mu_1 \frac{d_{01}}{v_{01f}^2}, \text{ then}$$

$$v_{01f}^{k+1} = \frac{\mu_1}{kC_f}, \text{ or} \qquad v_{01f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}}$$

By the same method, all of the speed values will be as follows:

$$v_{01f} = v_{23f} = v_{12f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}} \tag{3.27}$$

$$v_{12e} = v_{31e} = v_{20e} = \sqrt[k+1]{\frac{\mu_1}{kC_e}} \tag{3.28}$$

(3.27) and (3.28) result in the inverse relation between the speed of the robot and its energy constant. Since $C_f \geq C_e$, then $v_{12e} = v_{31e} = v_{20e} \geq v_{01f} = v_{23f} = v_{12f}$.

By substituting the achieved speed values in constraint (3.17) we have:

$$\frac{\sum_{(ijh)\in\mathcal{D}_2^f} d_{ijh}}{\sqrt[k+1]{\frac{\mu_1}{kC_f}}} + \frac{\sum_{(ijh)\in\mathcal{D}_2^e} d_{ijh}}{\sqrt[k+1]{\frac{\mu_1}{kC_e}}} = \overline{Ct} - 6\varepsilon, \text{therefore}$$

$$\sqrt[k+1]{\frac{\mu_1}{kC_f}} = \frac{\sqrt[k+1]{\frac{\mu_1}{kC_e}} \sum_{(ijh)\in\mathcal{D}_2^f} d_{ijh}}{\sqrt[k+1]{\frac{\mu_1}{kC_e}}(\overline{Ct} - 6\varepsilon) - \sum_{(ijh)\in\mathcal{D}_2^e} d_{ijh}} \tag{3.29}$$

Using (3.29), we obtain μ_1 value and the v_{ijh}'s as well. In order to simplify finding the value of μ_1, we can define $z = \dfrac{1}{\sqrt[k+1]{\mu_1}}$. Then, by putting z in constraint (3.17), we have:

$$z \sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + z \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e} = \overline{Ct} - 6\varepsilon, \text{ or}$$

$$z \left(\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e} \right) = \overline{Ct} - 6\varepsilon \text{ then}$$

$$z = \frac{\overline{Ct} - 6\varepsilon}{\sum_{(ijh)\in\mathcal{D}_1^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_1^e} d_{ijh} \sqrt[k+1]{kC_e}},$$

and

$$\mu_1 = \left(\frac{\sum_{(ijh)\in\mathcal{D}_2^f} d_{ijh} \sqrt[k+1]{kC_f} + \sum_{(ijh)\in\mathcal{D}_2^e} d_{ijh} \sqrt[k+1]{kC_e}}{\overline{Ct} - 6\varepsilon} \right)^{k+1}$$

Now by putting the value of μ_1 in equations (3.27) and (3.28), we get the speed values. □

Although Theorem 2 provides general formulas to find optimal speed values, the following corollary shows an easier calculation to find optimal speed values when $C_e = C_f$.

Corollary 2. *If $C_e = C_f$, at optimality, the speed of the robot is the same through the cycle for all of the moves as follows:*

$$v^*_{01f} = v^*_{23f} = v^*_{12f} = v^*_{12e} = v^*_{31e} = v^*_{20e} = \frac{\sum_{(ijh)\in\mathcal{D}_2} d_{ijh}}{\overline{Ct} - 6\varepsilon}.$$

Proof. In equations (3.27) and (3.28), if we consider $C_f = C_e = C$, we have

$$v_{01f} = v_{23f} = v_{12f} = v_{12e} = v_{31e} = v_{20e} = \sqrt[k+1]{\frac{\mu_1}{kC}}$$

By putting $\sqrt[k+1]{\frac{\mu_1}{kC}}$ in the first constraint (3.17), we have

$$\frac{\sum_{(ijh)\in\mathcal{D}_2} d_{ijh}}{\sqrt[k+1]{\frac{\mu_1}{kC}}} = \overline{Ct} - 6\varepsilon, \text{ therefore}$$

$$v^*_{01f} = v^*_{23f} = v^*_{12f} = v^*_{12e} = v^*_{31e} = v^*_{20e} = \sqrt[k+1]{\frac{\mu_1}{kC}} = \frac{\sum_{(ijh)\in\mathcal{D}_2} d_{ijh}}{\overline{Ct} - 6\varepsilon} \quad (3.30)$$

Needless to calculate μ_1, equation (3.30) yields identical speed values for all robot moves. □

As mentioned above, we can have different situations depending on P_i values, in which one or both of the constraints 2 and 3 (constraints (3.17) and (3.18)) in the Formulation 2 are tight along with the first constraint. These situations are discussed as follows:

Situation 2: The second constraint ((3.18)) is binding and the third constraint ((3.19)) is loose ($\mu_3 = 0$)

In this section, we consider the second situation, where only constraint (3.19) is not binding. In the following theorem, we find the optimal speed values for this situation.

Theorem 3. *If $C_f \geq C_e$, at optimality we have $v_{20e} \geq v_{01f} = v_{12f}$ and $v_{12e} = v_{31e} \geq v_{23f}$. In addition, the optimal speed values are as follows:*

$$v^*_{01f} = v^*_{12f} = \frac{(d_{01f} + d_{12f}) \sqrt[k+1]{kC_f} + d_{20e} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_f} \left(\overline{Ct} - 4\varepsilon - P_1\right)} \tag{3.31}$$

$$v^*_{20e} = \frac{(d_{01f} + d_{12f}) \sqrt[k+1]{kC_f} + d_{20e} \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_e} \left(\overline{Ct} - 4\varepsilon - P_1\right)} \tag{3.32}$$

$$v^*_{12e} = v^*_{31e} = \frac{P_1 - 2\varepsilon}{\sqrt[k+1]{kC_e} \left((d_{12e} + d_{31e}) \sqrt[k+1]{kC_e} + d_{23f} \sqrt[k+1]{kC_f}\right)} \tag{3.33}$$

$$v^*_{23f} = \frac{P_1 - 2\varepsilon}{\sqrt[k+1]{kC_f} \left((d_{12e} + d_{31e}) \sqrt[k+1]{kC_e} + d_{23f} \sqrt[k+1]{kC_f}\right)} \tag{3.34}$$

Proof. By setting $\mu_3 = 0$ in equations (3.21) to (3.26), we can achieve the following values:

$$v^*_{01f} = v^*_{12f} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC_f}} \tag{3.35}$$

$$v^*_{20e} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC_e}} \tag{3.36}$$

$$v^*_{12e} = v^*_{31e} = \sqrt[k+1]{\frac{\mu_1}{kC_e}} \tag{3.37}$$

$$v^*_{23f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}} \tag{3.38}$$

Since μ_1 and μ_2 are nonnegative and $C_f \geq C_e$, we can say $v_{01f} = v_{12f} \leq v_{20e}$ and similarly $v_{23f} \leq v_{12e} = v_{31e}$. Therefore, in this situation, we can always say that the robot, while passing d_{20e} (d_{12e} and d_{31e}) moves faster than or as fast as the times passing d_{01f} and d_{12f} (d_{23f}). μ_1 and μ_2 values can be calculated by putting the achieved values for the speeds from equations (3.35) through (3.38) in the first and second constraints (equations (3.17) and (3.18)).

First constraint (3.17):

$$\frac{d_{01f}+d_{12f}}{\sqrt[k+1]{\dfrac{\mu_1+\mu_2}{kC_f}}}+\frac{d_{20e}}{\sqrt[k+1]{\dfrac{\mu_1+\mu_2}{kC_e}}}+\frac{d_{12e}+d_{31e}}{\sqrt[k+1]{\dfrac{\mu_1}{kC_e}}}+\frac{d_{23f}}{\sqrt[k+1]{\dfrac{\mu_1}{kC_f}}}=\overline{Ct}-6\varepsilon \qquad (3.39)$$

Second constraint (3.18):

$$\frac{d_{01f}+d_{12f}}{\sqrt[k+1]{\dfrac{\mu_1+\mu_2}{kC_f}}}+\frac{d_{20e}}{\sqrt[k+1]{\dfrac{\mu_1+\mu_2}{kC_e}}}=\overline{Ct}-4\varepsilon-P_1 \qquad (3.40)$$

In order to simplify the equations (3.39) and (3.40), we set $z_1 = \dfrac{1}{\sqrt[k+1]{\mu_1+\mu_2}}$ and $z_2 = \dfrac{1}{\sqrt[k+1]{\mu_1}}$. By substituting z_1 and z_2 in equation (3.39), we have:

$$\begin{aligned}&(d_{01f}+d_{12f})\ \sqrt[k+1]{kC_f}z_1+d_{20e}\ \sqrt[k+1]{kC_e}z_1+(d_{12e}+d_{31e})\ \sqrt[k+1]{kC_e}z_2\\&+d_{23f}\ \sqrt[k+1]{kC_f}z_2=\overline{Ct}-6\varepsilon,\text{ or}\\&z_1\left((d_{01f}+d_{12f})\ \sqrt[k+1]{kC_f}+d_{20e}\ \sqrt[k+1]{kC_e}\right)\\&+z_2\left((d_{12e}+d_{31e})\ \sqrt[k+1]{kC_e}+d_{23f}\ \sqrt[k+1]{kC_f}\right)=\overline{Ct}-6\varepsilon \qquad (3.41)\end{aligned}$$

Also, by substituting z_1 and z_2 in equation (3.40), we have:

$$\begin{aligned}&z_1\left((d_{01f}+d_{12f})\ \sqrt[k+1]{kC_f}+d_{20e}\ \sqrt[k+1]{kC_e}\right)=\overline{Ct}-4\varepsilon-P_1,\text{ then}\\&z_1=\frac{\overline{Ct}-4\varepsilon-P_1}{(d_{01f}+d_{12f})\ \sqrt[k+1]{kC_f}+d_{20e}\ \sqrt[k+1]{kC_e}} \qquad (3.42)\end{aligned}$$

Now, we have two equations (3.41) and (3.42) with two unknowns z_1 and z_2. So, by substituting z_1 from equation (3.42) into equation (3.41) we can find z_2 as follows:

$$z_2=\frac{P_1-2\varepsilon}{\left((d_{12e}+d_{31e})\ \sqrt[k+1]{kC_e}+d_{23f}\ \sqrt[k+1]{kC_f}\right)}$$

By substituting z_1 and z_2 values into equations (3.35) to (3.38), we can find equations (3.31) to (3.34). □

Although Theorem 3 provides the optimal solution for the general case of Situation 2, the following corollary provides an easier and simpler formula to achieve the optimal solution when $C_f = C_e = C$ for the identical distance case.

Corollary 3. *If $C_f = C_e = C$ and $P_1 > 2\varepsilon$, at optimality we have $v_{01f} = v_{12f} = v_{20e} \geq v_{12e} = v_{23f} = v_{31e}$ and the optimal speed values for the robot moves are as follows:*

$$v^*_{01f} = v^*_{12f} = v^*_{20e} = \frac{d_{01f} + d_{12f} + d_{20e}}{\overline{Ct} - 4\varepsilon - P1}$$
$$v^*_{12e} = v^*_{23f} = v^*_{31e} = \frac{d_{12e} + d_{23f} + d_{31e}}{P_1 - 2\varepsilon}$$

Proof. By putting the parameter of C in equation (3.35) to (3.38) instead of C_f and C_e we have

$$v_{01f} = v_{12f} = v_{20e} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}} \tag{3.43}$$
$$v_{12e} = v_{23f} = v_{31e} = \sqrt[k+1]{\frac{\mu_1}{kC}} \tag{3.44}$$

Since $\mu_2 \geq 0$, it is obvious that $v_{01f} = v_{12f} = v_{20e} \geq v_{12e} = v_{23f} = v_{31e}$. For this situation, we can always say that the robot never moves slower while passing d_{01f}, d_{12f}, and d_{20e} than d_{12e}, d_{23f}, and d_{31e}.

By putting the values of $\sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}}$ and $\sqrt[k+1]{\frac{\mu_1}{kC}}$ (from equations (3.43) and (3.44)) in the first and second constraints (equations (3.17) and (3.18)), μ_1 and μ_2 can be calculated as follows:

Equation (3.17):

$$\frac{d_{01f} + d_{12f} + d_{20e}}{\sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}}} + \frac{d_{12e} + d_{23f} + d_{31e}}{\sqrt[k+1]{\frac{\mu_1}{kC}}} = \overline{Ct} - 6\varepsilon \tag{3.45}$$

Equation (3.18):

$$\frac{d_{01f} + d_{12f} + d_{20e}}{\sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}}} = \overline{Ct} - 4\varepsilon - P1,$$

by revising this equation we will have

$$\sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}} = \frac{d_{01f} + d_{12f} + d_{20e}}{\overline{Ct} - 4\varepsilon - P1} \tag{3.46}$$

Now, according to equation (3.46), we have

$$\mu_1 + \mu_2 = kC \left(\frac{d_{01f} + d_{12f} + d_{20e}}{\overline{Ct} - 4\varepsilon - P_1} \right)^{k+1} \tag{3.47}$$

According to equation (3.46), by putting the value of $\sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}}$ in the first constraint (3.45), we can issue

$$\overline{Ct} - 4\varepsilon - P_1 + \frac{d_{12e} + d_{23f} + d_{31e}}{\sqrt[k+1]{\frac{\mu_1}{kC}}} = \overline{Ct} - 6\varepsilon, \text{or}$$

$$\sqrt[k+1]{\frac{\mu_1}{kC}} = \frac{d_{12e} + d_{23f} + d_{31e}}{P_1 - 2\varepsilon} \tag{3.48}$$

Finally, from equation (3.48) we can find the value of μ_1. Consequently, μ_2 will be calculated from (3.47). However, from equations (3.46) and (3.48), it can be seen that in order to find the speed values we don't need to calculate μ_1 and μ_2 in this case. In other words, just by substituting the equations (3.46) and (3.48) in equations (3.44) and (3.43), the corollary is proven. □

The following proposition proves that if we are solving a problem where $C_e = C_f$, the distances between any pair of adjacent machines is identical and $P_1 \leq P_2$, then definitely the optimal solution cannot be Situation 2. In other words, we do not need to consider Situation 2. Alternatively, solving the other three situations will provide the global optimum for us.

Proposition 1. *If $C_e = C_f$ and the distances between any pair of adjacent machines are equal (additive distance case), at optimality, a KKT point can be found in situation 2 only if $P_1 \geq P_2$.*

Proof. The Situation 2 can be feasible whenever $P_1 \geq P_2$ and this is obvious, since for higher P_1 values, v_{12e}, v_{23f} and v_{31e} can be slower than the rest because the robot has more time for doing its other activities during P_1 and less time in the rest of the cycle, in which the first machine is idle.

As we concluded before (in Corollary 5), we have

$$v_{01f} = v_{12f} = v_{20e} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}} \geq v_{12e} = v_{23f} = v_{31e} = \sqrt[k+1]{\frac{\mu_1}{kC}}$$

On the other hand, from equations (3.46) and (3.48), we know that

$$\sqrt[k+1]{\frac{\mu_1+\mu_2}{kC}} = \frac{d_{01f}+d_{12f}+d_{20e}}{\overline{Ct}-4\varepsilon-P1}, \text{ and}$$

$$\sqrt[k+1]{\frac{\mu_1}{kC}} = \frac{d_{12e}+d_{23f}+d_{31e}}{P_1-2\varepsilon}$$

Since we are in the additive distance case, the nominators of these two equations are equal. Therefore, the only way to have

$$\frac{d_{01f}+d_{12f}+d_{20e}}{\overline{Ct}-4\varepsilon-P1} \geq \frac{d_{12e}+d_{23f}+d_{31e}}{P_1-2\varepsilon}$$

is that, by comparing their denominators, we should have

$$\overline{Ct}-4\varepsilon-P_1 \leq P_1-2\varepsilon$$

In other words

$$\overline{Ct} \leq 2P_1+2\varepsilon \Longrightarrow P_1 \geq \frac{\overline{Ct}-2\varepsilon}{2} \tag{3.49}$$

Additionally, in the third constraint (equation (3.19), we have

$$P_2+4\varepsilon+\frac{d_{23f}+d_{31e}}{\sqrt[k+1]{\frac{\mu_1}{kC}}}+\frac{d_{12f}}{\sqrt[k+1]{\frac{\mu_1+\mu_2}{kC}}}-\overline{Ct} \leq 0 \tag{3.50}$$

From equations (3.46) and (3.48) we can revise (3.50) as follows:

$$P_2+4\varepsilon+\frac{(P_1-2\varepsilon)\,(d_{23f}+d_{31e})}{d_{12e}+d_{23f}+d_{31e}}+\frac{\left(\overline{Ct}-4\varepsilon-P_1\right)\,(d_{12f})}{d_{01f}+d_{12f}+d_{20e}}-\overline{Ct} \leq 0 \tag{3.51}$$

By simplifying (3.51) we have

$$P_2 \leq \frac{\overline{Ct}-2\varepsilon}{2} \tag{3.52}$$

Finally, comparing (3.49) by (3.52), we are able to prove $P_1 \geq P_2$. □

Situation 3: The second constraint ((3.18)) is loose and the third constraint ((3.19)) is binding ($\mu_2 = 0$)

In this situation, instead of the second constraint, the third constraint is binding. So the most of the findings are similar to the Situation 2.

Theorem 4. *If $C_f \geq C_e$, at optimality we have $v_{20e} = v_{12e} \geq v_{01f}$ and $v_{23f} = v_{12f} \leq v_{31e}$. In addition the optimal speed values are as follows:*

$$\begin{aligned}
v^*_{01f} &= \frac{P_2 - 2\varepsilon}{\sqrt[k+1]{kC_f}\left((d_{12e} + d_{20e})\ \sqrt[k+1]{kC_e} + d_{01f}\ \sqrt[k+1]{kC_f}\right)}\\
v^*_{12e} = v^*_{20e} &= \frac{P_2 - 2\varepsilon}{\sqrt[k+1]{kC_e}\left((d_{12e} + d_{20e})\ \sqrt[k+1]{kC_e} + d_{01f}\ \sqrt[k+1]{kC_f}\right)}\\
v^*_{31e} &= \frac{(d_{23f} + d_{12f})\ \sqrt[k+1]{kC_f} + d_{20e}\ \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_e}\left(\overline{Ct} - 4\varepsilon - P_2\right)}\\
v^*_{12f} = v^*_{23f} &= \frac{(d_{23f} + d_{12f})\ \sqrt[k+1]{kC_f} + d_{20e}\ \sqrt[k+1]{kC_e}}{\sqrt[k+1]{kC_f}\left(\overline{Ct} - 4\varepsilon - P_2\right)}
\end{aligned}$$

Proof. By setting $\mu_2 = 0$ in equation (3.21) to (3.26), we can achieve the following values:

$$v^*_{01f} = \sqrt[k+1]{\frac{\mu_1}{kC_f}} \tag{3.53}$$

$$v^*_{12e} = v^*_{20e} = \sqrt[k+1]{\frac{\mu_1}{kC_e}} \tag{3.54}$$

$$v^*_{31e} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC_e}} \tag{3.55}$$

$$v^*_{12f} = v^*_{23f} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC_f}} \tag{3.56}$$

Since $C_f \geq C_e$ and $\mu_3 \geq 0$, the first statement of this theorem is proven. As in Situation 2 (Theorem 3), in order to find μ_1 and μ_3 values, we should put v_{ijh} values in binding constraints (equations (3.17) and (3.19)). Then, by considering $z_1 = \frac{1}{\sqrt[k+1]{\mu_1}}$ and $z_2 = \frac{1}{\sqrt[k+1]{\mu_1 + \mu_3}}$, we have:

$$\begin{aligned}
&z_1\left(d_{01f}\ \sqrt[k+1]{kC_f} + (d_{12e} + d_{20e})\ \sqrt[k+1]{kC_e}\right)\\
&+z_2\left((d_{12f} + d_{23f})\ \sqrt[k+1]{kC_f} + d_{31e}\ \sqrt[k+1]{kC_e}\right) = \overline{Ct} - 6\varepsilon
\end{aligned}$$

and

$$z_2\left((d_{23f} + d_{12f})\ \sqrt[k+1]{kC_f} + d_{31e}\ \sqrt[k+1]{kC_e}\right) = \overline{Ct} - 4\varepsilon - P_2$$

These two equations help us to find two unknowns z_1 and z_2, and consequently μ_1 and μ_3. Since $z_1 = \frac{1}{\sqrt[k+1]{\mu_1}}$ and $z_2 = \frac{1}{\sqrt[k+1]{\mu_1 + \mu_3}}$, after substituting their values in equations (3.53) to (3.56), this theorem is also proven. Due to the similarity of the computations with Theorem 3 we don't issue them here again. □

The next corollary develops an easier formula to compute the optimal speed values when the energy constants for the full and empty moves are identical.

Corollary 4. *If $C_f = C_e = C$ and $P_1 > 2\varepsilon$, at optimality we have $v_{01f} = v_{12e} = v_{20e} \leq v_{12f} = v_{23f} = v_{31e}$ and the optimal speed value for each move is as follows:*

$$v^*_{01f} = v^*_{12e} = v^*_{20e} = \frac{d_{01f} + d_{12f} + d_{20e}}{\overline{C}t - 4\varepsilon - P1} \tag{3.57}$$

$$v^*_{12f} = v^*_{23f} = v^*_{31e} = \frac{d_{12e} + d_{23f} + d_{31e}}{P_1 - 2\varepsilon} \tag{3.58}$$

Proof. By substituting the parameter C in equation (3.53) to (3.56) instead of C_f and C_e we get

$$v^*_{01f} = v^*_{12e} = v^*_{20e} = \sqrt[k+1]{\frac{\mu_1}{kC}}$$

$$v^*_{12f} = v^*_{23f} = v^*_{31e} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC}}$$

Since $\mu_3 \geq 0$, it is obvious that $v_{01f} = v_{12e} = v_{20e} \leq v_{12f} = v_{23f} = v_{31e}$.

Similar to Corollary 5, by putting the calculated speed values from (3.57) and (3.58) in the first and third constraints (equations (3.17) and (3.17)), this corollary can be proved as well. Like the second situation, by using the same procedure, μ_1 and μ_3 can be calculated. Nonetheless, it can be seen that we do not need to calculate μ_1 and μ_3 values for this case. □

The following proposition proves that if we are solving a problem where $C_e = C_f$, the distances between any pair of adjacent machines is identical and $P_1 \geq P_2$, then definitely the optimal solution cannot be Situation 3. In other words, we do not need to consider Situation 3. Alternatively, solving the other three situations will provide the global optimum for us.

Proposition 2. *If $C_e = C_f$ and the distances between any pair of the adjacent machine are equal (additive distance case), at optimality, a KKT point can be found in Situation 3 only if $P_1 \leq P_2$.*

Proof. By the same calculations as in the proof of Proposition 2, it can be shown that for feasibility of this situation we should have $P_2 \geq \dfrac{\overline{C}t - 2\varepsilon}{2}$ and

$P_1 \leq \dfrac{\overline{Ct} - 2\varepsilon}{2} \leq P_2$. Due to the similarity of the calculations with Proposition 4, they are not indicated here. □

Situation 4: All of the constraints (equation (3.17) to (3.19)) are binding

In this situation, all of the constraints are binding. Therefore, by revising equations (3.21) to (3.26), we achieve the following values for the speeds:

$$v^*_{01f} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC_f}} \tag{3.59}$$

$$v^*_{20e} = \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC_e}} \tag{3.60}$$

$$v^*_{12e} = \sqrt[k+1]{\frac{\mu_1}{kC_e}} \tag{3.61}$$

$$v^*_{23f} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC_f}} \tag{3.62}$$

$$v^*_{31e} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC_e}} \tag{3.63}$$

$$v^*_{12f} = \sqrt[k+1]{\frac{\mu_1 + \mu_2 + \mu_3}{kC_f}} \tag{3.64}$$

Before finding the value of Lagrangian multipliers, let us first consider the optimal solution for the case when $C_f = C_e = C$ in the following two corollaries.

Corollary 5. *If $C_f = C_e = C$, at optimality we have $v_{12f} \geq v_{01f} = v_{20e}, v_{23f} = v_{31e} \geq v_{12e}$.*

Proof. By putting the parameter C, instead of C_f and C_e in the equations (3.59) to (3.64), we have:

$$v^*_{01f} = v^*_{20e} \sqrt[k+1]{\frac{\mu_1 + \mu_2}{kC}} \tag{3.65}$$

$$v^*_{12e} = \sqrt[k+1]{\frac{\mu_1}{kC}} \tag{3.66}$$

$$v^*_{23f} = v^*_{31e} = \sqrt[k+1]{\frac{\mu_1 + \mu_3}{kC}} \tag{3.67}$$

$$v^*_{12f} = \sqrt[k+1]{\frac{\mu_1 + \mu_2 + \mu_3}{kC}} \tag{3.68}$$

Again, considering $\mu_1, \mu_2, \mu_3 \geq 0$, it is clear that the statement of the proposition is true. □

Corollary 6. *If we consider the additive distance case and $C_f = C_e = C$ besides $P_1 = P_2$, we have $v_{01f} \geq v_{01f} = v_{20e} = v_{23f} = v_{31e} \geq v_{12e}$.*

Proof. According to the equations (3.65) to (3.68), we can say that if $P_1 = P_2$, then $v_{01f}, v_{20e}, v_{23f}$ and v_{31e} have to be identical. The reason is that their corresponding distances (their nominators in equations (3.18) and (3.19)) are the same in the additive distance case. Therefore, the only remained unknowns (their speeds) should have the same value. This results in $\mu_2 = \mu_3$. Notice that, if $\mu_2 = \mu_3 = 0$, we have the same situation as Situation 1. □

Generally, if we consider the additive distance case and set $C_f = C_e = C$, the following statements can be concluded for the S_2 cycle:

- If $P_1 \geq P_2$, then $\mu_2 \geq \mu_3$. In this case, we are either in Situation 1 or 2, where $\mu_2 = \mu_3 = 0$ or $\mu_2 > \mu_3 = 0$ respectively (which depends on the value of P_1).

- If $P_2 \geq P_1$, then $\mu_3 \geq \mu_2$. So, we are either in Situation 1 or 3, where $\mu_2 = \mu_3 = 0$ or $\mu_3 > \mu_2 = 0$ respectively (which depends on the value of P_2 in this case).

- If $P_1 = P_2$, we are either in Situation 1 or 4, which results in $\mu_2 = \mu_3 = 0$ or $\mu_2 = \mu_3 > 0$ respectively. This case depends on the values of both P_1 and P_2 at the same time.

- Whenever $\mu_2 = \mu_3$, we have $v_{01f} = v_{20e} = v_{23f} = v_{31e}$ and whenever $\mu_2 = \mu_3 = 0$, all of the v_{ij}'s obtain the same value.

In Situation 4, μ_1, μ_2, and μ_3 values can be found by inserting equations (3.59)-(3.64) in the constraints (equations (3.17)-(3.19)). In order to simplify the calculations, we set

$$z_1 = \frac{1}{\sqrt[k+1]{\mu_1 + \mu_2}} \tag{3.69}$$

$$z_2 = \frac{1}{\sqrt[k+1]{\mu_1}} \tag{3.70}$$

$$z_3 = \frac{1}{\sqrt[k+1]{\mu_1 + \mu_3}} \tag{3.71}$$

$$z_4 = \frac{1}{\sqrt[k+1]{\mu_1 + \mu_2 + \mu_3}} \tag{3.72}$$

Then, we can rewrite the constraints as follows:
First constraint (3.17):

$$\begin{aligned} &6\varepsilon + \left(d_{01f}\,\sqrt[k+1]{kC_f} + d_{20e}\,\sqrt[k+1]{kC_e}\right) z_1 + d_{12e}\,\sqrt[k+1]{kC_e} z_2 \\ &+ \left(d_{23f}\,\sqrt[k+1]{kC_f} + d_{31e}\,\sqrt[k+1]{kC_e}\right) z_3 + d_{12f}\,\sqrt[k+1]{kC_f} z_4 = \overline{Ct} \end{aligned}$$

Second constraint (3.18):

$$P_1 + 4\varepsilon + \left(d_{01f}\,\sqrt[k+1]{kC_f} + d_{20e}\,\sqrt[k+1]{kC_e}\right) z_1 + d_{12f}\,\sqrt[k+1]{kC_f} z_4 = \overline{Ct}$$

Third constraint (3.19):

$$P_2 + 4\varepsilon + \left(d_{23f}\,\sqrt[k+1]{kC_f} + d_{31e}\,\sqrt[k+1]{kC_e}\right) z_3 + d_{12f}\,\sqrt[k+1]{kC_f} z_4 = \overline{Ct}$$

Now, we possess three nonlinear equations and four unknowns $(z_1, ..., z_4)$. Then, if we consider z_4 as a constant and try to find z_1, z_2, and z_3 values in terms of z_4, we will find the following equations:

$$z_1 = \frac{\overline{Ct} - P_1 - 4\varepsilon - d_{12f}\,\sqrt[k+1]{kC_f} \cdot z_4}{d_{01f}\,\sqrt[k+1]{kC_f} + d_{20e}\,\sqrt[k+1]{kC_e}} \tag{3.73}$$

$$z_2 = \frac{P_1 + P_2 + 2\varepsilon - \overline{Ct} + \left(d_{12f}\,\sqrt[k+1]{kC_f}\right) \cdot z_4}{d_{12e}\,\sqrt[k+1]{kC_e}} \tag{3.74}$$

$$z_3 = \frac{\overline{Ct} - P_2 - 4\varepsilon - d_{12f}\,\sqrt[k+1]{kC_f} \cdot z_4}{d_{23f}\,\sqrt[k+1]{kC_f} + d_{31e}\,\sqrt[k+1]{kC_e}} \tag{3.75}$$

On the other hand, according to the equations (3.69)-(3.72), we know that

$$\left(\frac{1}{z_1}\right)^{k+1} + \left(\frac{1}{z_3}\right)^{k+1} = \left(\frac{1}{z_2}\right)^{k+1} + \left(\frac{1}{z_4}\right)^{k+1} \tag{3.76}$$

Now, by putting the equations (3.73)-(3.75) into equation (3.76), we will have a nonlinear function (let us call it $h(z_4)$) with the power of $k + 1$, in which the root of this function is the value of z_4 where $h(z_4) = 0$.

$$h(z_4) = \left(\frac{1}{z_1}\right)^{k+1} + \left(\frac{1}{z_3}\right)^{k+1} - \left(\frac{1}{z_2}\right)^{k+1} - \left(\frac{1}{z_4}\right)^{k+1} \tag{3.77}$$

By substituting the value of z_4 in equations (3.73)-(3.75), we can find z_1, z_2, and z_3. Then, μ_i, i $\in$ {1,2,3} values can be calculated easily by equations (3.69)-(3.72). Subsequently, optimal v_{ijh} values will be calculated by equations (3.59)-(3.64).

However, finding the root of this function is not straightforward. The main reason is that according to the value of the parameters, the function can have a different behavior. Later by looking at the first derivative of our function ($h(z_4)$) in equation (3.78) we can see that whenever k is odd, the function can be monotonically increasing or decreasing with regard to the other parameters. On the other hand, by looking at the second derivative at (3.79), whenever k is even, the function can be convex, concave or none of them. So, there may be more than one root for this function. we can use a simple algorithm and utilize the well-known methods in the literature to find all of the roots for this function. If the number of roots that generate a feasible solution for Formulation 2 is more than one, we will calculate the objective function for each of them and the root that generates the smallest objective function value will be selected. Notice that we know the lower and upper bounds of v_{12f} (LB and UB). Then, according to equations (3.68) and (3.72), we can find the bounds of z_4 as well. Since it is equal to:

$$z_4 = \frac{1}{v_{12f}\ \sqrt[k+1]{kC_f}},$$

then

$$\frac{1}{UB\ \sqrt[k+1]{kC_f}} \leq z_4 \leq \frac{1}{LB\ \sqrt[k+1]{kC_f}},$$

Finally, by calculating a range for the values of z_4 we can utilize available methods in the literature, such as Bisection or Newton-Raphson (NR) methods in our algorithm to find the roots of the function. The algorithm of our proposed approach is given in Algorithm 1. In this algorithm we applied NR method. In the standard NR method, we start with a function (f), its derivative (f'), and an initial solution (x_0) as a guess for the root of the function. If the function satisfies the assumptions made in the derivation of the formula and the initial guess is close, then a better approximation x_1 is

$$x_1 = x_0 + \frac{f(x)}{f'(x)}$$

The process is repeated for consequent x_n and x_{n+1}, until a sufficiently accurate value is reached.

Algorithm 1: Finding the roots of function $h(z_4)$

```
Data: $\epsilon$: a very small number as a step-size, $LB_{z_4}$ and $UB_{z_4}$ (which are
      the bounds of $z_4$), $h(z_4) = (\frac{1}{z_1})^{k+1} + (\frac{1}{z_3})^{k+1} - (\frac{1}{z_2})^{k+1} - (\frac{1}{z_4})^{k+1}$
Result: roots of the function
$x \leftarrow LB_{z_4}$;
Calculate $h(x)$;
do
    $x_1 = x + \epsilon$;
    Calculate $h(x_1)$;
    if $h(x) \cdot h(x_1) < 0$ then
        Start NR algorithm for $[x, x_1]$ and display the root;
        $x \leftarrow x_1$;
        $h(x) \leftarrow h(x_1)$;
    else
        $x \leftarrow x_1$;
        $h(x) \leftarrow h(x_1)$;
    end
while $x_1 \leq UB_{z_4}$;
```

In order to apply NR method, we need to use both the function and its derivative. To calculate the derivative of equation (3.77), let us first define the constants a_1, a_2, a_3, b, c_1, c_2, and c_3 as follows:

$$a_1 = \overline{Ct} - P_1 - 4\varepsilon$$

$$a_2 = P_1 + P_2 + 2\varepsilon - \overline{Ct}$$

$$a_3 = \overline{Ct} - P_2 - 4\varepsilon$$

$$b = d_{12f} \sqrt[k+1]{kC_f}$$

$$c_1 = d_{01f} \sqrt[k+1]{kC_f} + d_{20e} \sqrt[k+1]{kC_e}$$

$$c_2 = d_{12e} \sqrt[k+1]{kC_e}$$

$$c_3 = d_{23f} \sqrt[k+1]{kC_f} + d_{31e} \sqrt[k+1]{kC_e}$$

Now we can rewrite the equations (3.73) through (3.75) and define the equation (3.77) as a function of z_4.

$$z_1 = \frac{a_1 - b \cdot z_4}{c_1}$$
$$z_2 = \frac{a_2 + b \cdot z_4}{c_2}$$
$$z_3 = \frac{a_3 - b \cdot z_4}{c_3}$$

$$h(z_4) = \left(\frac{c_1}{a_1 - b \cdot z_4}\right)^{k+1} + \left(\frac{c_3}{a_3 - b \cdot z_4}\right)^{k+1} - \left(\frac{c_2}{a_2 + b \cdot z_4}\right)^{k+1} - \left(\frac{1}{z_4}\right)^{k+1}$$

Consequently, the first derivative of this function is

$$h^{'}(z_4) = \frac{(k+1)b \cdot c_1^{k+1}}{(a_1 - b \cdot z_4)^{k+2}} + \frac{(k+1)b \cdot c_3^{k+1}}{(a_3 - b \cdot z_4)^{k+2}} + \frac{(k+1)b \cdot c_2^{k+1}}{(a_2 + b \cdot z_4)^{k+2}} + \frac{(k+1)}{z_4^{k+2}} \tag{3.78}$$

Also, the second derivative is

$$\begin{aligned} h^{''}(z_4) = & \frac{(k+2)(k+1)b^2 \cdot c_1^{k+1}}{(a_1 - b \cdot z_4)^{k+3}} + \frac{(k+2)(k+1)b^2 \cdot c_3^{k+1}}{(a_3 - b \cdot z_4)^{k+3}} \\ & - \frac{(k+2)(k+1)b^2 \cdot c_2^{k+1}}{(a_2 + b \cdot z_4)^{k+3}} - \frac{(k+2)(k+1)}{z_4^{k+3}} \end{aligned} \tag{3.79}$$

The whole proposed approach of this thesis is coded in C++ in a way that we can find the optimal solution for both S_1 and S_2 cycles. Also, the optimal situation of the S_2 cycle. In the following section, we show an example of the results of our algorithm.

Example 1. In this example, assume the set of parameters are given as follows: $d_{01f} = d_{12e} = d_{23f} = d_{12f} = 1$, $d_{31e} = d_{20e} = 2$, $k = 3$, $\varepsilon = 1$, $C_f = 4$, $C_e = 2$ and $\overline{Ct} = 40$, $P_1 = 13$ and $P_2 = 11$.

The results of this example after solving the problem is shown in Table 3.1.

As it is shown in the Table 3.1, the only feasible situation for the S_2 cycle is Situation 1. According to the KKT approach, only one of the four situations is feasible. Therefore, we can stop the algorithm after finding the first feasible situation. On the other hand, S_1 cycle provides a feasible solution as well.

Table 3.1: Results for Example 1

Cycle	Situation	Lagrange multipliers	results
S_1	-	μ_1= 1.116	$v_{01f} = 0.552$ $v_{12f} = 0.552$ $v_{23f} = 0.552$ $v_{30e} = 0.657$ objective function=3.721
S_2	1	μ_1= 0.024	$v_{01f} = 0.212$ $v_{12f} = 0.212$ $v_{23f} = 0.212$ $v_{12e} = 0.252$ $v_{31e} = 0.252$ $v_{20e} = 0.252$ objective function=0.274
S_2	2	μ_1= 0.126, μ_2= -0.118	Infeasible
S_2	3	μ_1= 0.282, μ_3= -0.276	Infeasible
S_2	4	No feasible root	Infeasible

Whenever both cycles provide a feasible solution, we should compare their solutions (objective functions) and choose the minimum of them, which is S_2 in this example. In another set of parameters, the optimal solution can be in another cycle or situation. For instance, if we keep all of the parameters same but reduce the cycle time upper bound by 14 seconds $\left(\overline{Ct} = 26\right)$, the results can completely change as shown in Table 3.2. In this example (Example 2), the second situation of S_2 is optimal.

Table 3.2: Results for Example 1 (cont.)

Cycle	Situation	Lagrange multipliers	results
S_1	-	-	Infeasible (constraint is not satisfied)
S_2	1	μ_1= 0.202	Infeasible (constraints are not satisfied)
S_2	2	$\mu_1 = 0.126, \mu_2 = 0.209$	$v_{01f} = 0.409$ $v_{12f} = 0.409$ $v_{23f} = 0.320$ $v_{12e} = 0.381$ $v_{31e} = 0.381$ $v_{20e} = 0.486$ objective function=1.471
S_2	3	μ_1= 0.282, μ_3= -0.131	Infeasible (μ_3 is negative)
S_2	4	μ_1= 0.155, μ_2= 0.191 μ_3= -0.036	Infeasible (μ_3 is negative)

So far, we applied KKT analysis and the optimal situations in each feasible state are interpreted analytically when the speeds are unbounded. A set of theorems are proposed that can help us to find the optimal solution to each specific situation for each cycle. In the next section, we will describe how to deal with bounded speed case by applying a heuristic algorithm. After next section, we will provide the complete algorithm of our proposed procedure in this thesis.

Dealing With Robot Speed Upper Bounds

In this section, we present a greedy heuristic algorithm to deal with the case where robot speeds are bounded from above. Our reason for not considering the speed upper bound in the first step of solving problems is to reduce the number of calculations and complexity of the problems. If we consider all of the speeds less than or equal to UB at the beginning of our solution procedure, we should consider six more constraints in our analysis. Namely, six new equations should be inspected in complementary slackness condition and six new Lagrangian multipliers in dual feasibility condition. As shown before, in the S_2 cycle, we already have three constraints corresponding to the cycle time upper bound. Since each constraint can be either tight or loose, by adding six new constraints the number of situations will be (2^9) which is drastically larger than (2^3). In this case, implementing KKT analysis might not be a suitable method for this problem[2]. Therefore, we solve the problem without considering the upper bound for the decision variables and after finding the optimal solution for the unconstrained speeds, we check whether they are within their bounds or not. If they are within their bounds, we are done. Otherwise, we solve the problem with a heuristic algorithm that is developed in this section.

Our algorithm is inspired by the greedy heuristic algorithm for solving the knapsack problem proposed by George Dantzig [89]. In this algorithm, the items are sorted in a non-increasing order of value per unit of weight ($\frac{v_i}{w_i}$). Each decision variable has a weight w_i in the constraint and a value v_i in the objective function, along with a maximum weight capacity W. Next, the algorithm greedily selects items in the achieved order. Provided that there is an infinite supply of each item, if m is the maximum value of items that fit into the sack, then the greedy algorithm is guaranteed to achieve at least a value of $m/2$. In this thesis, we developed an algorithm by applying the same logic.

In our algorithm, at first we set $v_{ijh} = UB$ (for all $(ijh) \in \mathcal{D}_1$ and $\mathcal{D}_2$). As shown before, in Formulation 1 we have only one constraint that corresponds to the cycle time upper bound and we proved that in the optimal solution, it is always tight. On the other hand, in Formulation 2, we have three constraints that correspond to the cycle time upper bound. Similarly, we proved that in the optimal solution, the first constraint is always tight (Lemma 1 and Lemma 2). Therefore, similar to the Dantzig's greedy heuristic algorithm, we start decreasing the speed values step by step until making these constraints tight. At each step, the decision variable that contributes the most to the objective function is selected (the speed which its value per unit of weight is greater than other speeds). Namely, we calculate the following equation for all of the

[2] Since we proved that half of the situations are infeasible, due to the tightness of the first constraint, then the numbers 2^9 and 2^3 will shrink to 2^8 and 2^2

variables:

$$\left|\frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}}\right| \qquad \forall (ijh) \in \mathcal{D}_1 \text{ and } \mathcal{D}_2 \tag{3.80}$$

where f_{ijh} is the part of the objective function that corresponds to the move (ijh) and g_{ijh} is the part of each constraint that corresponds to the move (ijh). It should be noticed that, in both Formulation 1 and Formulation 2, the objective functions and the constraints are additively separable and each separable part corresponds to a different decision variable. In the S_1 formulation, there is only one constraint. On the other hand, in the S_2 formulation, there are three constraints. But the coefficient of each decision variable is the same through all of the constraints. As a result, considering any constraint of the S_2 cycle, we will achieve the same solution from equation (3.80). After calculating equation (3.80) for all of the decision variables, the one with the maximum value is selected to be reduced. We apply this formula in each step and after selecting a variable, we reduce its value by a small step length (Δ). We continue this approach until reaching the stopping criteria, which is making the first constraint binding (for both S_1 and S_2). The whole procedure of our algorithm is presented below:

The procedure:

Step 1. Set $v_{ijh} = UB, \forall (ijh) \in \mathcal{D}_1$.

Step 2. If the first constraint of S_1 is tight, report the solution for S_1 cycle and Go to Step 6. Else, Go to Step 3.

Step 3. Choose speed $v^*_{ijh} = \text{argmax} \left|\frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}}\right|, \forall (ijh) \in \mathcal{D}_1$.

Step 4. Set $v^*_{ijh} = v^*_{ijh} - \Delta$.

Step 5. If the first constraint of S_1 is tight, report the solution for S_1 cycle and Go to Step 5. Else, Go to Step 3.

Step 6. Set $v_{ijh} = UB, \forall (ijh) \in \mathcal{D}_2$.

Step 7. If the first constraint of S_2 is tight, report the solution for the S_2 cycle and STOP; Else, Go to Step 8.

Step 8. Choose speed $v^*_{ijh} = \text{argmax} \left| \frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}} \right|, \forall (ijh) \in \mathcal{D}_2$.

Step 9. Set $v^*_{ijh} = v^*_{ijh} - \Delta$.

Step 10. If the first constraint of S_2 is tight, report the solution for the S_2 cycle and STOP; Else, Go to Step 11.

Step 11. If the second or the third constraint of S_2 is tight, set $v^*_{ijh} = \text{argmax} \left| \frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}} \right|$ such that (ijh) is not in the binding constraint and Go to Step 9. Else, Go to Step 8.

The algorithm is coded in C++ and its pseudo code is provided in Algorithm 2. Tables 3.3 and 3.4 shows the effect of different Δ values on the solution and CPU time for solving S_1 and S_2, respectively. The solutions are compared with General Algebraic Modeling System (GAMS) using CONOPT nonlinear solver.

It can be seen that after $\Delta = 0.0001$, the achieved solutions by the proposed algorithm is almost equal to the provided solution by CONOPT (GAMS) and the CPU time is negligible. Therefore, $\Delta = 0.0001$ is selected for conducting our numerical studies in Chapter 4.

It should be noticed that, if only one of the decision variables exceeds the upper bound value and the rest of the speeds be within the bound, we don't need to solve the problem by our heuristic algorithm; we can only substitute its value by UB and solve the problem again with a reduced number of decision variables with the presented exact approach in the previous section. If after solving this new problem, we achieve all of the speeds within the bound, we are done. Otherwise, we need to solve it by the proposed heuristic algorithm. In order to understand this specific situation better, we consider a parametric example.

Since in Situation 4 of the S_2 cycle all of the optimal speed values can be different from each other, we consider our example in this situation. Therefore, assume the optimal solution, after solving a problem, is in situation 4 and all of the optimal speed values are different from each other. At that point, we may see

Algorithm 2: Greedy heuristic algorithm for finding the optimal robot speed values

```
Data: $\varepsilon$, $P_1$, $P_2$, $\overline{Ct}$, $UB$, and the set of distances $d_{ijh} \forall (ijh) \in \mathcal{D}_1$ and $\mathcal{D}_2$
Result: The optimal speed values ($v^*_{ijh}$) and the optimal objective
        function (OPT) when the speeds are bounded
$v^*_{ijh} \leftarrow UB, \forall (ijh) \in \mathcal{D}_1$;
while equation (3.1) $< \overline{Ct}$ do
    Choose speed $v^*_{ijh} = \text{argmax} \left| \frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}} \right|, \forall (ijh) \in \mathcal{D}_1$ ;
    $v^*_{ijh} = v^*_{ijh} - \Delta$;
end
$v^*_{ijh} = v^*_{ijh} + \Delta$;
Report OPT and $v^*_{ijh}$ for the $S_1$ cycle;
$v^*_{ijh} \leftarrow UB, \forall (ijh) \in \mathcal{D}_2$;
while equation (3.17) $< \overline{Ct}$ do
    Choose speed $v^*_{ijh} = \text{argmax} \left| \frac{\frac{\partial f_{ijh}(v_{ijh})}{\partial v_{ijh}}}{\frac{\partial g_{ijh}(v_{ijh})}{\partial v_{ijh}}} \right|, \forall (ijh) \in \mathcal{D}_2$ ;
    $v^*_{ijh} = v^*_{ijh} - \Delta$;
    if equation (3.18) or (3.19) does not hold then
        $v^*_{ijh} = v^*_{ijh} + \Delta$;
        Remove the corresponding $(ijh)$ from the argmax and don't select
          it anymore;
    end
end
$v^*_{ijh} = v^*_{ijh} + \Delta$;
Report OPT and $v^*_{ijh}$ for the $S_2$ cycle;
```

Table 3.3: Evaluating the performance of the heuristic algorithm by different Δ values for solving S_1 with $UB = 3.2$

	Solution Approach							
	CONOPT	Greedy Approximation Algorithm						
Δ	-	0.3	0.1	0.01	0.001	0.0005	0.0001	0.00005
Obj. Func.	157.108	160.973	157.8	157.321	157.117	157.117	157.108	157.108
Time (s)	0	0	0.008	0.005	0.033	0.067	0.326	0.556

Table 3.4: Evaluating the performance of the heuristic algorithm by different Δ values for solving S_2 with $UB = 2$

	Solution Approach							
	CONOPT	Greedy Approximation Algorithm						
Δ	-	0.2	0.1	0.01	0.001	0.0005	0.0001	0.00005
Obj. Func.	53.703	54.64	54.04	53.717	53.706	53.706	53.704	53.704
Time (s)	1.07	0.002	0.004	0.02	0.17	0.353	1.7	3.059

that one of the speeds is greater than UB. Let us assume v_{12f} is the only decision variable that exceeds the upper bound while the rest are less than or equal to UB. Then, the Lagrangian function for the modified problem is as follows:

$$\begin{aligned}
&L(v_{ijh}, \mu) = \\
&\text{Minimize } C_f \left[d_{01f}v_{01f}^k + d_{23f}v_{23f}^k + d_{12f}v_{12f}^k\right] + C_e \left[d_{12e}v_{12e}^k + d_{31e}v_{31e}^k + d_{20e}v_{20e}^k\right] \\
&\qquad +\mu_1 \left[6\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12e}}{v_{12e}} + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} - \overline{Ct}\right] \\
&\qquad +\mu_2 \left[P_1 + 4\varepsilon + \frac{d_{01f}}{v_{01f}} + \frac{d_{12f}}{v_{12f}} + \frac{d_{20e}}{v_{20e}} - \overline{Ct}\right] \\
&\qquad +\mu_3 \left[P_2 + 4\varepsilon + \frac{d_{23f}}{v_{23f}} + \frac{d_{31e}}{v_{31e}} + \frac{d_{12f}}{v_{12f}} - \overline{Ct}\right] \\
&\qquad +\mu_4 \left[v_{12f} - UB\right]
\end{aligned}$$

Now, we want to solve this new problem. Earlier, we proved that we have four feasible situations for the S_2 cycle. Conversely, we will have eight situations concerning the new constraint that can be either tight ($\mu_4 \geq 0$) or loose ($\mu_4 = 0$) now. However, if the added constraint is not binding (i.e., $\mu_4 = 0$), the problem renovates to its previous format with three Lagrangian multipliers and we have to continue the same approach as we did before which yields again an infeasible speed for v_{12f}. Consequently, the variable that exceeds the upper bound has to be equal to UB. In this case, we set v_{12f} equal to UB. In other words, we act with it as a parameter and solve the problem with a reduced number of decision variables. After solving the adapted problem, we check the newly computed values for the speeds. If they are all less than or equal to UB, we are done. However, if we find another decision variable greater than UB, this approach is not helpful anymore and we cannot continue with fixing the value of the new infeasible speed to UB again. In addition, this approach is not applicable if the number of speeds that are higher than UB is greater than 1. The reason is that by reducing the value of the speed which is higher than UB, the value of a number of other speeds may decrease as well. We show this situation through an example:

Example 2. The set of parameters for this example are given as follows:

$d_{31e} = d_{12e} = d_{23f} = d_{12f} = 1$, $d_{01f} = d_{20e} = 2$, $k = 3$, $\varepsilon = 1$, $C_f = 4$, $C_e = 2$ and $\overline{Ct} = 30$, $P_1 = 22$ and $P_2 = 22$.

At first, the problem is solved without considering any speed upper bound. The results are shown in the first row of Table 3.5 as Problem 1. Now, assume that for this problem $UB = 1.3$. Therefore, v_{20e} exceeds the upper bound value. Then, we add the corresponding constraint ($v_{20e} \leq UB$) to the problem and solve the problem again. As it is proved before, in the optimal solution, we have $v_{20e} = UB$. However, we can see that by reducing the value of v_{20e}, two other speeds, v_{23f} and v_{31e} that are underlined in the table, decrease as well. In order to be sure that there is no alternative solution that can avoid reducing other decision variables as well, we restrict those two speeds to be greater than or equal to their values in the first problem (Problem 1). That is, the following constraints are added to the problem:

1- $v_{23f} \geq 0.569$

2- $v_{31e} \geq 0.716$

As the results show in the last row of Table 3.5 (Problem 3), after adding these constraints and resolving the problem, the objective function value is increased. So, there cannot be an alternative solution.

Table 3.5: Results for Example 2

Problem	v_{01f}	v_{12f}	v_{23f}	v_{31e}	v_{12e}	v_{20e}	Obj. Func.
1	1.137	1.183	$\underline{\mathbf{0.569}}$	$\underline{\mathbf{0.716}}$	0.059	1.433	26.481
2	1.206	1.245	$\underline{\mathbf{0.561}}$	$\underline{\mathbf{0.707}}$	0.06	1.3	26.862
3	1.219	1.219	0.569	0.716	0.059	1.3	26.911

By the help of this example, we understand that by decreasing one of the decision variables, another decision variable(s) can do likewise. So, for instance, if two of the speeds exceed the UB, we cannot set both of them equal to UB. Because by reducing only one of them, the other one may decrease as well and be less than UB. The problem will be more complicated if we have more variables greater than UB. Another situation that can happen is when after solving a problem, only one of the speeds, say v_1, be greater than UB, but after setting it equal to UB and solving the problem again, another speed, say v_2, exceeds the upper bound value. In this situation, we cannot decrease v_2 and set it equal to UB and keep the previous variable $v_1 = UB$ as well. The reason is that by decreasing v_2, v_1 may decrease as well and $v_1 = UB$ will not be optimal anymore. As a result, if the number of decision variables that exceed the upper bound is more than one, we cannot set all of them equal to UB. Instead, we will use our proposed heuristic algorithm.

As mentioned before, applying the greedy heuristic algorithm for this problem is much easier than solving it with considering all of the upper bounds at the starting point of the approach. Next, we give the steps of our general procedure for solving an energy-conscious scheduling problem in a two-machine robotic cell. The pseudo code of this algorithm (Algorithm 3), which is coded in C++, is shown in the Appendix.

Step 1. Start solving the S_1 cycle.

Step 2. Solve S_1 without considering speed upper bounds.

Step 3. If there is only one speed greater than UB, set it to UB, solve the problem again and Go to Step 4. If there are more than one speeds greater than UB, Go to step 5. Else, Go to Step 6.

Step 4. If there is another speed greater than UB, Go to Step 5. Else, Go to Step 6.

Step 5. Solve the problem by the developed greedy heuristic algorithm.

Step 6. Start solving the S_2 cycle.

Step 7. Solve S_2 without considering speed upper bounds.

Step 8. If there is only one speed greater than UB, set it to UB, solve the problem again and Go to Step 9. If there are more than one speeds greater than UB, Go to step 10. Else, Go to Step 11.

Step 9. If there is another speed greater than UB, Go to Step 10. Else, Go to Step 11.

Step 10. Solve the problem by the developed greedy heuristic algorithm.

Step 11. Compare the objective function of S_1 with S_2 and choose the best one.

CHAPTER 4

NUMERICAL RESULTS AND ANALYSIS

In this section, we study the numerical results derived by comparing the controlled and uncontrolled speed systems; in an uncontrolled system, we can not modify the robot's speed and its value is fixed at UB, while the speeds in the controlled system are adjustable and can be achieved according to the approaches proposed in Chapter 3. 288 instances are designed to evaluate the benefits of using the controlled speed systems. Later, we will explain that, while comparing controlled and uncontrolled systems, there is no energy saving in S_1 cycle, in which we cannot have any improvement in energy consumption reduction. We show via numerical results that our proposed approach can significantly reduce the robot energy consumption inside a robotic cell manufacturing system without increasing the cycle time value. Also, we analyze the effect of different levels of parameters on the behavior of the optimal solution.

Comparison of Controlled and Uncontrolled Speed Systems

As described in Chapter 2 (Section 2.1), there are different cases for the robot move distances. Therefore, four levels of distance cases are determined for each problem: additive-identical, additive-general, constant, and general. To have a credible comparison among the distance cases, the summation of all distances that the robot should travel in a cycle is considered the same for all instances. To clarify, in our instances for the additive-identical distance case, the values of distances are $d_{01f} = d_{12f} = d_{23f} = d_{12e} = 1.5$, $d_{31e} = d_{20e} = 3$, and for the constant case we have $d_{01f} = d_{12f} = d_{23f} = d_{12e} = d_{31e} = d_{20e} = 2$. It can be seen that the distance value between any pair of machines is different in each case, whereas the summation of them is equal to 12 in both cases. In addition, five different sets of values for distances in the general case and another five different sets of values for the additive-general case are generated randomly.

As shown in Table 4.1, three levels (small, medium, and large) of processing time values are considered. Furthermore, we have two levels for energy constants. In one of them the constants corresponding to the empty and full moves are identical and in the other case, they are different. But the summation of them is considered the same for each case. As mentioned before, the speed upper bound of the robot is diverse according to the type and the model of the robot. However, by looking at some of the papers in the literature [59, 90, 91], we can distinguish a range for the robot speed from 0.0 $5\frac{m}{s}$ to 2.2 $\frac{m}{s}$. As a result, we determined two levels for UB equal to 1 and 2. Similarly, an even value besides an odd value are defined for the value of k. All of the levels are shown in Table 4.1. It can be seen that 2 levels are defined for each of the k, UB, and C_h parameters. Also, 3 levels for P_m and 4 cases for distances are considered. As a result, we have $(2^3) \times 3 \times 4$ or 96 different sets of parameters, which the corresponding results are shown in 4 different tables (Tables 4.3 to 4.6).

Table 4.1: The level of parameters

Parameters	Levels		
k	2	3	
UB	1	2	
C_h	$C_e = C_f = 3$	$C_e = 2$, $C_f = 4$	
P_i	$P_1 = 1$, $P_2 = 3$	$P_1 = P_2 = 10$	$P_1 = 22$, $P_2 = 19$
Distances	Constant General	Additive-general	Additive-identical

The algorithm is coded and compiled with Microsoft Visual Studio C++. Since the algorithm solution time is less than a second, the effect of time is not shown in the computational results.

The objective function or the energy consumption values are calculated for both of the controlled and uncontrolled systems. To investigate the benefits of using the speed controlled system instead of setting all of the speeds at their upper bound, at first we solve each problem by considering all of the speeds equal to UB and achieve the corresponding cycle time and energy consumption values. Next, we set $\overline{Ct}$ equal to the achieved cycle time and then, by the help of our proposed algorithm, we solve the problem again through controlling the robot speeds (speed controlled system). If we can satisfy the cycle time upper bound with a lower energy consumption, it means that in at least one of the moves the robot can move slower than its upper bound value. Consequently, a better objective function can be reached.

The defined distance sets are shown in Table 4.2. Notice that, as mentioned before, the summation of distances in all defined sets is equal to 12. In Table 4.3, the robot traveled distances are in the additive identical case and Table 4.4 is dedicated to the constant case. As shown in Table 4.2, for each of these cases, only one set of parameters is defined. The last columns of Tables 4.3 and 4.4 show the percentage of the deviation (improvement) between the objective function values and represents the percentage of the energy saving amount for each instance.

Tables 4.5 and 4.6 show the results for the general and additive-general distance cases, respectively. For each of these cases, we have generated 5 sets of distances randomly (table 4.2). The cycle time, waiting times and energy consumption columns represent the average of these 5 replications. The last two columns, show the average and the maximum deviation (improvement) (energy consumption reduction) value derived from those five sets.

As shown in Tables 4.3 through 4.6, we have two columns w_1 and w_2 that show waiting times of the robot in front of M_1 and M_2, respectively. The waiting times are related to the uncontrolled system that the robot moves at its maximum speed (UB). It can be seen that for the cases that the summation of w_1 and w_2 is equal to zero, there is no energy saving. The reason is that either the optimum cycle is S_1, or there is no waiting time in S_2 cycle. We should notice that if the

Table 4.2: The sets of distance values (given in meters) for each distance case

Distance	**Set**	d_{01f}	d_{12e}	d_{23f}	d_{31e}	d_{12f}	d_{20e}
Constant	1	2	2	2	2	2	2
Additive-identical	1	1.5	1.5	1.5	3	1.5	3
Additive-general	1	2	1	2	3	1	3
	2	1	2	1	3	2	3
	3	1	1	3	4	1	2
	4	3	1	1	2	1	4
	5	0.5	2.5	0.5	3	2.5	3
General	1	1	1	3	1	5	1
	2	2	2	2	3	2	1
	3	4	1	2	1	2	2
	4	0.5	2	0.5	2	0.5	6.5
	5	1	3	1.5	3.5	2.5	0.5

optimal solution is in S_1 cycle, we can never achieve the same cycle time by reducing the speed values. Since the cycle time is equal to the summation of the full processing times plus the travel time of the robot which is done at its maximum speed. Therefore, by reducing the speed of the robot, it is impossible to reach the same cycle time. On the other hand, for S_2 cycle, we can reduce the robot speed if there is a waiting time. In other words, the robot does not need to travel at its maximum speed in order to reach a machine and wait there until finishing the part processing time. This yields excessive energy consumption. However, if the waiting times are equal to zero and the robot never waits in front of any machine, the achieved cycle time cannot be satisfied by a set of smaller robot speeds.

Table 4.3: Comparison of controlled and uncontrolled systems in additive identical distance case

P_1	P_2	C_e	C_f	UB	k	Ct^{S_1}	Ct^{S_2}	w_1	w_2	US*	CS**	Improv. (%)
10	10	3	3	1	2	38	20	0	2	36	32.3	10.2
					3	38	20	0	2	36	31.9	11.5
				2	2	32	17	0	5	144	126.3	12.3
					3	32	17	0	5	288	252.1	12.5
		2	4	1	2	38	20	0	2	33	30.6	7.4
					3	38	20	0	2	33	30.2	8.4
				2	2	32	17	0	5	132	120.2	8.9
					3	32	17	0	5	264	240.1	9.1
1	3	3	3	1	2	22	18	0	0	36	36.0	0.0
					3	22	18	0	0	36	36.0	0.0
				2	2	16	12	0	0	144	144.0	0.0
					3	16	12	0	0	288	288.0	0.0
		2	4	1	2	22	18	0	0	33	33.0	0.0
					3	22	18	0	0	33	33.0	0.0
				2	2	16	12	0	0	132	132.0	0.0
					3	16	12	0	0	264	264.0	0.0
22	19	3	3	1	2	59	32	3	11	36	22.9	36.3
					3	59	32	3	11	36	20.9	41.9
				2	2	53	29	3	14	144	82.0	43.1
					3	53	29	3	14	288	152.5	47.0
		2	4	1	2	59	32	3	11	33	22.2	32.7
					3	59	32	3	11	33	20.5	37.9
				2	2	53	29	3	14	132	80.5	39.0
					3	53	29	3	14	264	151.2	42.7
									Avg:	120.75	99.3	16.7

*Uncontrolled System (US) energy consumption

**Controlled System (CS) energy consumption

Table 4.4: Comparison of controlled and uncontrolled systems in constant distance case

P_1	P_2	C_e	C_f	UB	k	Ct^{S_1}	Ct^{S_2}	w_1	w_2	US*	CS**	Improv. (%)
10	10	3	3	1	2	38	20	0	2	36	31.5	12.5
					3	38	20	0	2	36	30.8	14.6
				2	2	32	17	0	5	144	120.7	16.2
					3	32	17	0	5	288	240.2	16.6
		2	4	1	2	38	20	0	2	36	33.0	8.3
					3	38	20	0	2	36	32.5	9.7
				2	2	32	17	0	5	144	128.4	10.8
					3	32	17	0	5	288	256.1	11.1
1	3	3	3	1	2	22	18	0	0	36	36.0	0.0
					3	22	18	0	0	36	36.0	0.0
				2	2	16	12	0	0	144	144.0	0.0
					3	16	12	0	0	288	288.0	0.0
		2	4	1	2	22	18	0	0	36	36.0	0.0
					3	22	18	0	0	36	36.0	0.0
				2	2	16	12	0	0	144	144.0	0.0
					3	16	12	0	0	288	288.0	0.0
22	19	3	3	1	2	59	32	3	11	36	22.1	38.7
					3	59	32	3	11	36	20.3	43.7
				2	2	53	29	3	14	144	79.8	44.6
					3	53	29	3	14	288	150.2	47.9
		2	4	1	2	59	32	3	11	36	23.9	33.7
					3	59	32	3	11	36	22.2	38.5
				2	2	53	29	3	14	144	87.5	39.3
					3	53	29	3	14	288	165.9	42.4
									Avg:	126	102.2	17.9

*Uncontrolled System (US) energy consumption

**Controlled System (CS) energy consumption

Table 4.5: Comparison of controlled and uncontrolled systems in general distance case

P_1	P_2	C_e	C_f	UB	k	Ct^{S_2}	w_1	w_2	US*	CS**	Avg.*	Max.*
10	10	3	3	1	2	21.8	1.3	3.2	36	27.2	24.4	25.6
					3	21.8	1.3	3.2	36	25.6	29.0	31.5
				2	2	17.9	0.75	6.3	144	101.3	29.7	34.5
					3	17.9	0.75	6.3	288	194.5	32.5	36.7
		2	4	1	2	21.8	1.3	3.2	35.8	28.0	22.1	25.5
					3	21.8	1.3	3.2	35.8	26.6	26.2	30.6
				2	2	17.9	0.75	6.3	143.2	106.7	26.3	33.6
					3	17.9	0.75	6.3	286.4	205.8	29.0	35.9
1	3	3	3	1	2	18	0	0	36	36.0	0.0	0.0
					3	18	0	0	36	36.0	0.0	0.0
				2	2	12	0	0	144	144.0	0.0	0.0
					3	12	0	0	288	288.0	0.0	0.0
		2	4	1	2	18	0	0	35.8	35.8	0.0	0.0
					3	18	0	0	35.8	35.8	0.0	0.0
				2	2	12	0	0	143.2	143.2	0.0	0.0
					3	12	0	0	286.4	286.4	0.0	0.0
22	19	3	3	1	2	32.4	3.1	13.8	36	25.6	29.0	35.5
					3	32.4	3.1	13.8	36	24.5	32.0	37.0
				2	2	29.2	3	17.1	144	87.8	39.0	47.9
					3	29.2	3	17.1	288	165.2	42.7	54.3
		2	4	1	2	32.4	3.1	13.8	35.8	27.0	25.4	35.1
					3	32.4	3.1	13.8	35.8	25.9	28.4	36.6
				2	2	29.2	3	17.1	143.2	94.4	34.5	39.4
					3	29.2	3	17.1	286.4	179.0	37.8	45.0
								Avg:	125.7	97.9	20.3	24.4

*The average of the improvement in the five distance sets

**The maximum of the improvement in the five distance sets

Table 4.6: Comparison of controlled and uncontrolled systems in general additive distance case

P_1	P_2	C_e	C_f	UB	k	Ct^{S_2}	w_1	w_2	US*	CS**	Avg.*	Max.*
10	10	3	3	1	2	20.8	0.8	2.0	36.0	29.9	16.9	25.0
					3	20.8	0.8	2.0	36.0	28.9	19.6	29.2
				2	2	17.4	0.4	5.0	144.0	114.7	20.3	28.6
					3	17.4	0.4	5.0	288.0	225.6	21.7	31.4
		2	4	1	2	20.8	0.8	2.0	33.0	28.3	14.1	22.3
					3	20.8	0.8	2.0	33.0	27.6	16.3	25.9
				2	2	17.4	0.4	5.0	132.0	109.8	16.8	25.2
					3	17.4	0.4	5.0	264.0	216.4	17.9	27.6
1	3	3	3	1	2	18.0	0.0	0.0	36.0	36.0	0.0	0.0
					3	18.0	0.0	0.0	36.0	36.0	0.0	0.0
				2	2	12.0	0.0	0.0	144.0	144.0	0.0	0.0
					3	12.0	0.0	0.0	288.0	288.0	0.0	0.0
		2	4	1	2	18.0	0.0	0.0	33.0	33.0	0.0	0.0
					3	18.0	0.0	0.0	33.0	33.0	0.0	0.0
				2	2	12.0	0.0	0.0	132.0	132.0	0.0	0.0
					3	12.0	0.0	0.0	264.0	264.0	0.0	0.0
22	19	3	3	1	2	32.2	3.2	11.0	36.0	24.2	32.7	40.8
					3	32.2	3.2	11.0	36.0	22.7	36.9	45.3
				2	2	29.0	3.0	14.0	144.0	87.8	39.0	45.9
					3	29.0	3.0	14.0	288.0	164.3	42.9	48.5
		2	4	1	2	32.2	3.2	11.0	33.0	23.6	28.6	34.1
					3	32.2	3.2	11.0	33.0	22.4	32.4	38.1
				2	2	29.0	3.0	14.0	132.0	86.0	35.0	38.9
					3	29.0	3.0	14.0	264.0	162.2	38.6	43.4
								Avg:	120.8	97.5	17.9	22.9

*The average of the improvement in the five distance sets

**The maximum of the improvement in the five distance sets

The average percentage of energy saving which is calculated by the average of the last column numbers is equal to 18.73%. Also, the maximum percentage of energy saving is equal to 54.30%. However, to look at the effect of the different levels for each parameter, we show the results separately in a distinct table (Table 4.7). It can be seen that for small processing times the energy saving is nil, but for larger processing times, the percentage of energy saving becomes larger. By looking at Tables 4.3-4.6, it can be seen that the reason is related to the waiting times. Namely, for bigger processing times, the probability of having waiting times is higher than the instances with smaller processing times. Consequently, when we have larger waiting times, we can have slower robot speeds.

Further, the energy consumption reduction is smaller when the energy constants are not identical for full and empty moves ($C_e = 2$ and $C_f = 4$). The reason is that in half of the instances, we consider an additive distance case (either additive-identical or additive-general), in which the summation of empty moves is larger than the summation of full moves. In addition, in the constant case, the summations are equal to each other. Therefore, in 75% of the instances, the summation of distances for empty moves are greater than or equal to the full moves. As a result, one unit of increase in the energy constant of empty moves ($C_e = 2$ to $C_e = 3$) has a greater influence on energy saving in comparison with a unit of decrease in the energy constant of full moves ($C_f = 4$ to $C_e = 3$). Similarly, by the same logic, it can be explained why in the general distance case, the percentage of energy saving is higher than other cases. Since this is the only case that the summation of full moves can be higher than empty moves and we always have $C_f \geq C_e$, the amount of energy saving can be higher for this situation and the numerical studies provide us the expected result.

On the other hand, it is obvious that for a robot that its maximum speed is very large, the policy of applying it at its maximum speed is not economical. Consequently, if we use the controlled speed system approach for a robot that has the capability to move at higher speeds, we can save a more significant amount of energy.

Finally yet importantly, having a bigger value for k does not mean a bigger objective function value. The reason was related to the value of the speeds that can be either less than, equal or bigger than 1. In any of these cases, we can have a different result by using a bigger k value. However, when k is bigger, there is more improvement in energy saving, because energy consumption becomes more sensitive to the reduced amount of speeds. In Section 4.2, we analyze the effect of different levels of k on the optimal solution in more detail.

.

Table 4.7: Numerical results

Levels	Energy saving (%)
Small processing times	0.00
Medium processing times	20.65
Large processing times	35.53
$C_e = C_f = 3$	20.08
$C_e = 2$, $C_f = 4$	17.38
$UB = 1$	16.97
$UB = 2$	20.49
$k = 2$	17.71
$k = 3$	19.75
Constant	18.86
Additive-identical	16.71
Additive-general	17.91
General	20.13

In this section, we understood that we can have a different amount of energy saving percentages according to our problems and their parameters. Evidently, the minimum amount of energy saving is equal to zero. But this question may arise that how much is the maximum amount of energy saving in a two-machine robotic cell scheduling problem. In the following, we show the maximum possible amount of energy saving for this problem.

Proposition 3. *For a two-machine robotic cell scheduling problem, the maximum possible amount of energy saving can be calculated by the following equation:*

$$\textit{Maximum energy saving percentage} = \frac{UB^k - LB^k}{UB^k} \times 100 \tag{4.1}$$

Proof. In order to compute the energy saving percentage, we have to calculate the difference between the objective function of the uncontrolled and controlled speed systems divided by the objective function of the uncontrolled system. In the uncontrolled system, we substitute all of the speeds by UB. Therefore, the objective function is equal to

$$\begin{aligned}\text{Obj. func.} &= C_f \left[\sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} \cdot UB^k \right] + C_e \left[\sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \cdot UB^k \right] \\ &= UB^k \left[C_f \cdot \sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} + C_e \cdot \sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \right] \end{aligned} \tag{4.2}$$

Now, assume that by applying our proposed approach in this thesis, we can find the same cycle time by substituting all of the speeds at their lower bounds. Since the robot cannot move slower than LB, we know that this is the best possible case that can happen. Consequently, the objective function will be similar to equation 4.2, but we have LB instead of UB inside of the equation.

$$\begin{aligned}\text{Obj. func.} &= C_f \left[\sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} \cdot LB^k \right] + C_e \left[\sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \cdot LB^k \right] \\ &= LB^k \left[C_f \cdot \sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} + C_e \cdot \sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \right] \end{aligned} \tag{4.3}$$

Therefore, by subtracting the two objective functions and dividing it by the objective function of the uncontrolled system (equation 4.2), we achieve

$$\begin{aligned}&\text{Maximum energy saving percentage} = \\ &\frac{(UB^k - LB^k) \left[C_f \cdot \sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} + C_e \cdot \sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \right]}{UB^k \left[C_f \cdot \sum_{(ijh)\in D_1^f or D_2^f} d_{ijh} + C_e \cdot \sum_{(ijh)\in D_1^e or D_2^e} d_{ijh} \right]} \end{aligned} \tag{4.4}$$

Since we are comparing the uncontrolled and controlled speed objective functions of the same problem, the values inside of the brackets are equal and we can omit them from the fraction. Then, we get the same equation as 4.1. □

Sensitivity Analysis on Parameters

Our optimization models developed in the previous chapter have mutual and different parameters. The mutual ones are $\overline{Ct}$, ε, k, P_1, and P_2. On the other hand, the sets of the robot moves ($\mathcal{D}_1$ and $\mathcal{D}_2$) are different for S_1 and S_2 cycle. We provided a number of theorems, lemmas, propositions and corollaries in chapter 3 and achieved several formulas that show the optimal speed values for different situations. According to the provided formulas and equations, we can easily analyze the effect of some of the parameters on energy consumption and the optimal solution behavior. For instance, any modification to the value of UB is known for us. If there are a number of speeds that their values are equal to UB, by decreasing the UB those speed values will be smaller as well and the other speeds will be larger. Therefore, the objective function will increase. In addition, if the loading/unloading time (ε) plus the distances become larger, the robot should move faster to finish the cycle within the predefined cycle time upper bound. As a result, performing more analysis on those parameters does not bring any new observation for us. Therefore, in this section, we only focus on the analysis of other parameters ($\overline{Ct}$, k and P_i) that can provide new observations and novel results to this work.

Numerical Results for Different Levels of $\overline{Ct}$

In order to assess the effect of different levels of $\overline{Ct}$ on the optimal solution, we consider the rest of the parameters constant, which their values are as follows:

$d_{01f} = 1d_{12e} = d_{23f} = d_{12f} = 1$, $d_{31e} = d_{20e} = 2$, $k = 3$, $\varepsilon = 1$, $C_f = 4$, $C_e = 2$, $P_1 = 13$ and $P_2 = 11$.

S_1 **cycle:**

We consider a sample feasible range for cycle time upper bound ($\overline{Ct}$=[38,48]). The achieved objective function and the optimal speed values for each $\overline{Ct}$ value are given in Table 4.8 and finally, the set of non-dominated solutions (Pareto frontier) that corresponds to the selected $\overline{Ct}$ values is shown in Figure 4.1. It can be seen that all of the speeds are decreasing while the cycle time upper bound is increasing.

Table 4.8: Numerical results of $\overline{Ct}$ on S_1 cycle

$\overline{Ct}$	$Obj.func.$	v_{01f}	v_{12f}	v_{23f}	v_{30e}
38	7.27	0.69	0.69	0.69	0.82
39	5.10	0.61	0.61	0.61	0.73
40	3.72	0.55	0.55	0.55	0.66
41	2.80	0.50	0.50	0.50	0.60
42	2.15	0.46	0.46	0.46	0.55
43	1.69	0.43	0.43	0.43	0.51
44	1.36	0.39	0.39	0.39	0.47
45	1.10	0.37	0.37	0.37	0.44
46	0.91	0.35	0.35	0.35	0.41
47	0.76	0.33	0.33	0.33	0.39
48	0.64	0.31	0.31	0.31	0.37

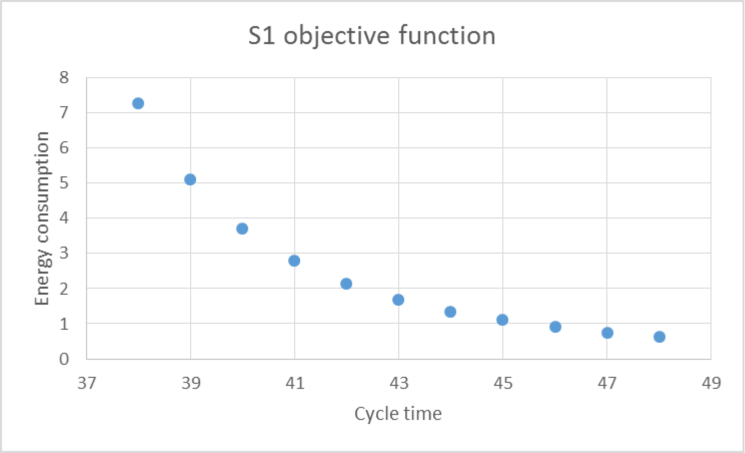


Figure 4.1: The Set of Non-dominated solutions for S_1 cycle corresponding $\overline{Ct} = [38, 48]$

It is clear that by increasing the cycle time the energy consumption will decrease. Namely, the robot can move slower and consume less amount of energy to satisfy the cycle time upper bound. But the set of non-dominated solutions will provide additional insights for the decision makers that can be used to select the best-preferred solution.

S_2 cycle:

Most of the results for S_2 cycle is similar to S_1 cycle. The achieved objective function and the optimal speed values for $\overline{Ct}$=[20,30] are given in Table 4.9 and the set of non-dominated solutions that corresponds to the selected $\overline{Ct}$ values is shown in Figure 4.2.

Table 4.9: Numerical results of $\overline{Ct}$ on S_2 cycle

$\overline{Ct}$	$Obj.func.$	v_{01f}	v_{12e}	v_{23f}	v_{31e}	v_{12f}	v_{20e}	$Situation$
20	30.04	1.22	0.15	0.64	0.76	1.22	1.45	4
21	13.23	0.91	0.17	0.54	0.65	0.94	1.09	
22	7.03	0.73	0.19	0.47	0.56	0.76	0.87	
23	4.21	0.61	0.22	0.42	0.50	0.64	0.72	
24	2.75	0.52	0.26	0.37	0.45	0.55	0.62	
25	1.93	0.46	0.32	0.34	0.40	0.48	0.54	
26	1.47	0.41	0.38	0.32	0.38	0.41	0.49	2
27	1.20	0.37	0.38	0.32	0.38	0.37	0.44	
28	1.02	0.34	0.38	0.32	0.38	0.36	0.40	
29	0.89	0.31	0.37	0.31	0.37	0.31	0.37	1
30	0.78	0.30	0.36	0.30	0.36	0.30	0.36	

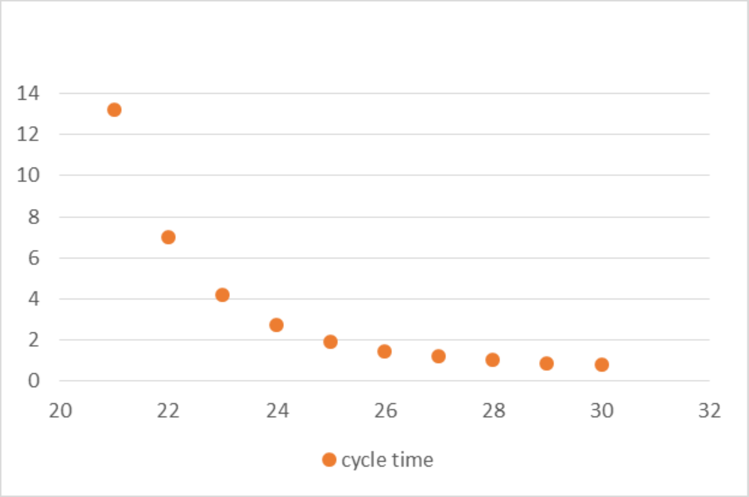


Figure 4.2: The Set of Non-dominated solutions for S_2 cycle

If we look at Table 4.9 we can see that when the cycle time upper bound is smaller, the optimal solution is in Situation 4 and by increasing $\overline{Ct}$, we will move to Situation 2 and finally to Situation 1. The reason that we never meet Situation 3 is that $P_1 > P_2$ (it was proved in Section 3). If P_2 was greater than P_1, instead of being in Situation 2, we would have Situation 3 besides situations 1 and 4.

Table 4.10: Comparison of S_1 and S_2 Cycles

$\overline{Ct}$	S_1 obj. func.	S_2 obj. func.
84	0.024	0.023
85	0.022	0.021
86	0.021	0.021
87	**0.02**	**0.02**
88	0.019	0.02
89	0.018	0.019
90	0.017	0.018

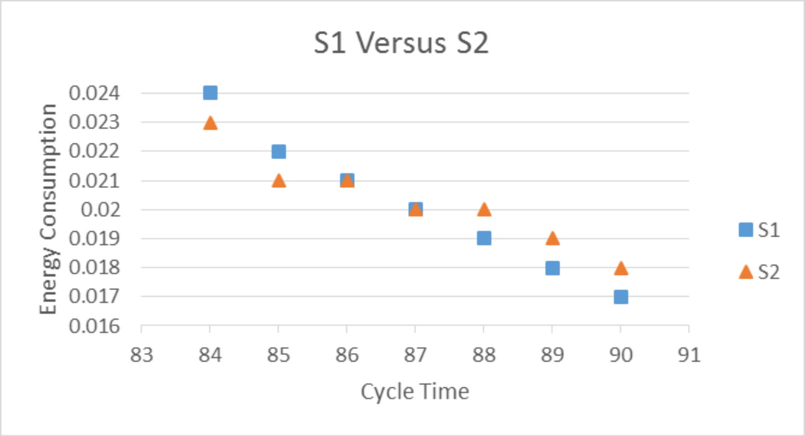


Figure 4.3: The effect of the $\overline{Ct}$ on the optimal cycle

An important question that may arise is that which cycle is optimal according to the different values of $\overline{Ct}$. In Table 4.10 and Figure 4.3, we show that after a certain value the optimal cycle will change from S_2 to S_1. The reason is related to the number of moves in each cycle. In S_1 cycle, we have only 4 moves (3 full moves and one empty move). Whereas, in S_2 cycle, there are 6 moves (3 full moves and 3 empty moves). When the cycle time upper bound is very large, the speeds are very small and the robot can satisfy the cycle time upper bound with lower speed values. Consequently, the cycle with a smaller number of moves requires a smaller amount of energy to finish a cycle.

As it is shown, after $\overline{Ct}$=87, S_2 is not optimal anymore. This critical point can happen earlier when P_1 and P_2 are smaller. Then $\overline{Ct}$ can be smaller as well to meet this critical point when S_1 can provide a better solution instead of S_2. As an example, if we keep all of the parameters with the same values and only change $P_1 = 1$ and $P_2 = 2$, this can happen at $\overline{Ct}$=15, as shown in Figure 4.4.

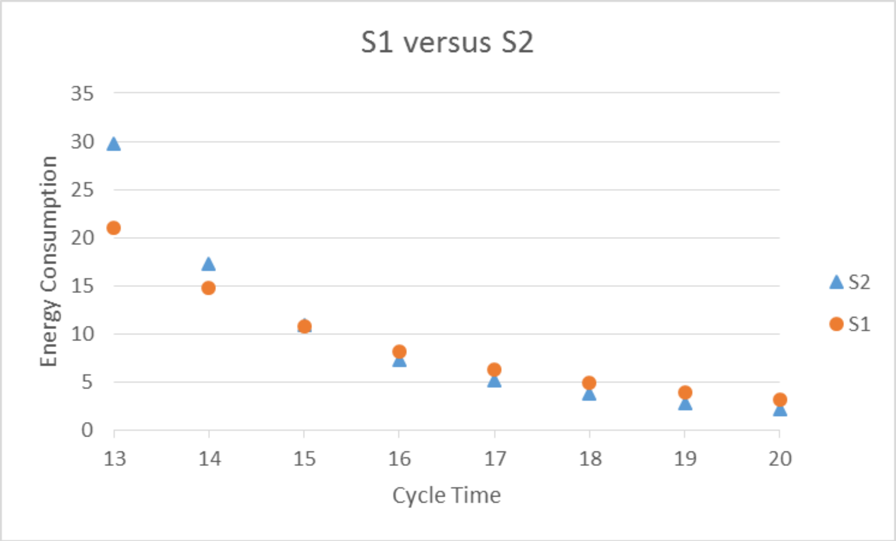


Figure 4.4: The effect of the $\overline{Ct}$ on the optimal cycle

In the following section, we will study the effects of P_1 and P_2 in more details.

Numerical Results for Different Processing Time Levels

In this section, we consider all of the parameters as same as the previous section. In S_1 cycle, we have only one constraint containing both P_1 and P_2. Then, by increasing either P_1 or P_2, it is clear that the right-hand side of the constraint $(\overline{Ct} - 6\varepsilon - P_1 - P_2)$ will be smaller and all of the speeds should be increased to keep the constraint feasible. As an instance, three levels for processing time values $(P_1 + P_2 = 6, P_1 + P_2 = 8$, and $P_1 + P_2 = 10)$ are selected and the results are shown in Figure 4.5. Three different sets of non-dominated solutions are achieved according to the different P_1 and P_2 values.

On the other hand, in S_2 cycle, P_1 and P_2 are in different constraints and dealing with different variables; we have no processing time in the first constraint, but P_1 and P_2 appear in the second and third constraints, respectively. Therefore, we can have different speed behaviors in this cycle. At first, we plot the effect of the processing times on the achieved pareto frontier or set of non-dominated solutions for S_2 cycle that correspond to the selected $\overline{Ct}$ values (Figure 4.6). Again three different levels for processing time values are selected and it can be seen that for smaller P_m values, energy consumption values are smaller as well.

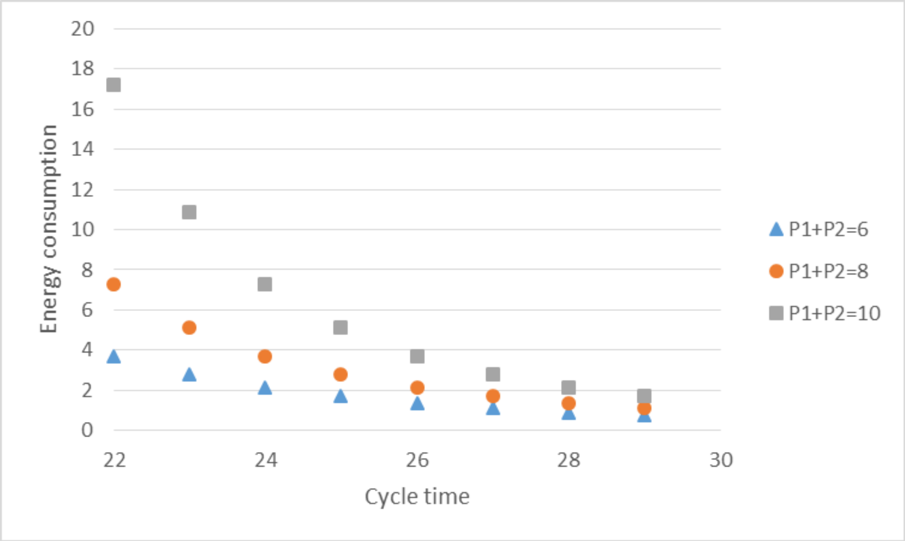


Figure 4.5: The effect of the processing time values on the energy consumption pareto frontier in S_1 cycle

Furthermore, we consider the effect of any change in any of the processing times. For instance, assume $P_2 = 12$. By increasing P_1 from 1 to 25 we can achieve the objective function and the optimal speed values as shown in Table 4.11.

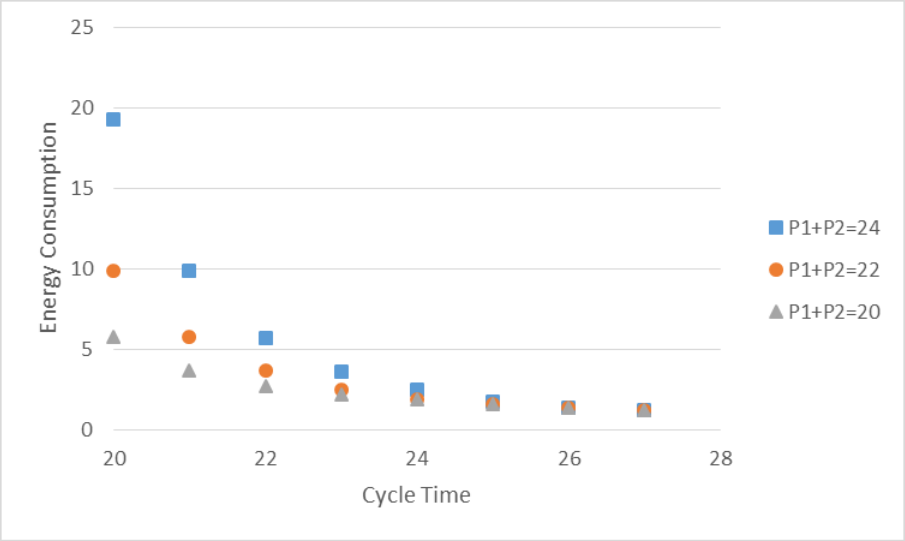


Figure 4.6: The effect of the processing time values on the energy consumption pareto frontier in S_2 cycle

It can be seen that for small P_1 values, there is no change in objective function, since the minimum speed values that keep the first constraint tight satisfy the other two constraints and by increasing P_1, still they are loose. However, when P_1 reaches a value that makes the second constraint tight as well as the first constraint (Situation 2), the variables that are in the second constraint start to

Table 4.11: Numerical results of $\overline{Ct}$ on S_2 cycle

P_1	Obj. func	v_{01f}	v_{12e}	v_{23f}	v_{31e}	v_{12f}	v_{20e}	Situation
1	0.27	0.21	0.25	0.21	0.25	0.21	0.25	1
2	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
3	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
4	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
5	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
6	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
7	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
8	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
9	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
10	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
11	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
12	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
13	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
14	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
15	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
16	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
17	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
18	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
19	0.27	0.22	0.25	0.21	0.25	0.22	0.26	2
20	0.29	0.23	0.23	0.20	0.23	0.23	0.27	
21	0.31	0.25	0.22	0.19	0.22	0.25	0.29	
22	0.34	0.26	0.21	0.18	0.21	0.26	0.31	
23	0.40	0.28	0.20	0.17	0.20	0.28	0.34	
24	0.48	0.31	0.19	0.16	0.19	0.31	0.36	
25	0.60	0.33	0.18	0.15	0.18	0.33	0.40	

be increased to keep the constraint feasible. Consequently, the other speeds can be smaller to compensate the added cost. Nevertheless, the energy consumption still increases by increasing P_1 in Situation 2. The behavior of the variables is shown in Figure 4.7.

Same results will occur if we keep P_1 fixed and modify P_2. The only difference is that, by increasing P_2, after Situation 1, we will meet the optimal solution in Situation 3 instead of Situation 2. Due to similarity, we do not consider this case. However, we are interested to see what will happen if we increase both P_1 and P_2 simultaneously. A similar instance is shown in Table 4.12. In this case,

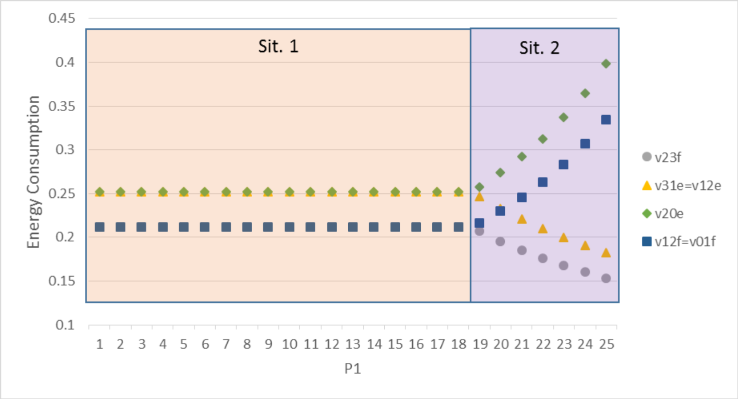


Figure 4.7: The effect of the P_1 on the optimal speed values for S_2 cycle

we increase both P_1 and P_2 from 1 to 25 seconds.

As we proved in Section 3, if we are in additive identical distance case and P_1=P_2, the optimal solution is either in Situation 1 or 4. When P_1 and P_2 are very small numbers, we are in Situation 1 and by increasing their values we will shift to Situation 4. Through the Situation 1, there is no change in speeds and objective function values. Since the achieved optimal speeds are necessary to keep the first constraint tight and we cannot have smaller values for them. However, after a point, we meet the optimal solution at Situation 4 that all of the constraints are binding. Therefore, in all of the moves correlated to P_1 and P_2 in the second and third constraints the robot has to be faster in order to satisfy the cycle time upper bound. Whereas, the v_{12e} which is the only variable that is not dealing with P_1 and P_2 in second and third constraints can be reduced to compensate a partial cost contributed from the rise of the other variables.

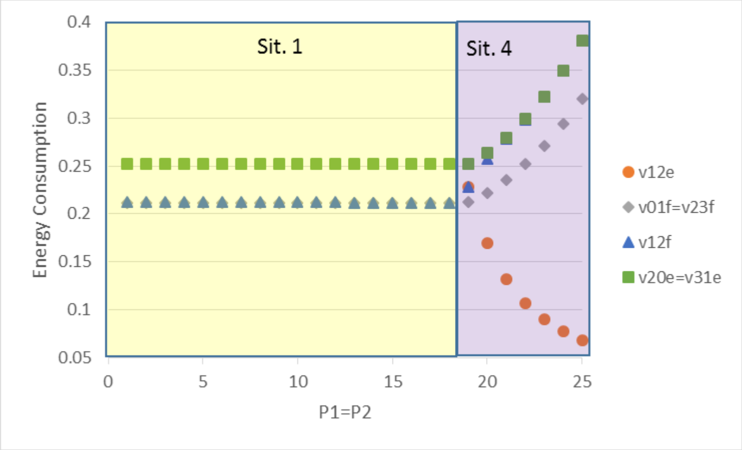


Figure 4.8: The effect of the P_1 and P_2 on the optimal speed values for S_2 cycle

Table 4.12: Numerical results of $\overline{Ct}$ on S_2 cycle

P_1 & P_2	Obj. func.	v_{01f}	v_{12e}	v_{23f}	v_{31e}	v_{12f}	v_{20e}	Situation
1	0.27	0.21	0.25	0.21	0.25	0.21	0.25	1
2	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
3	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
4	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
5	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
6	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
7	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
8	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
9	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
10	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
11	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
12	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
13	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
14	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
15	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
16	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
17	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
18	0.27	0.21	0.25	0.21	0.25	0.21	0.25	
19	0.28	0.21	0.23	0.21	0.25	0.23	0.25	4
20	0.31	0.22	0.17	0.22	0.26	0.26	0.26	
21	0.37	0.24	0.13	0.24	0.28	0.28	0.28	
22	0.45	0.25	0.11	0.25	0.30	0.30	0.30	
23	0.56	0.27	0.09	0.27	0.32	0.32	0.32	
24	0.71	0.29	0.08	0.29	0.35	0.35	0.35	
25	0.93	0.32	0.07	0.32	0.38	0.38	0.38	

Similar to the previous section (analysis of $\overline{Ct}$), we can see that in smaller P_1 and P_2 values, S_1 is optimal but for greater P_1 and P_2 values, S_2 provides the optimal solution (Figure 4.9).

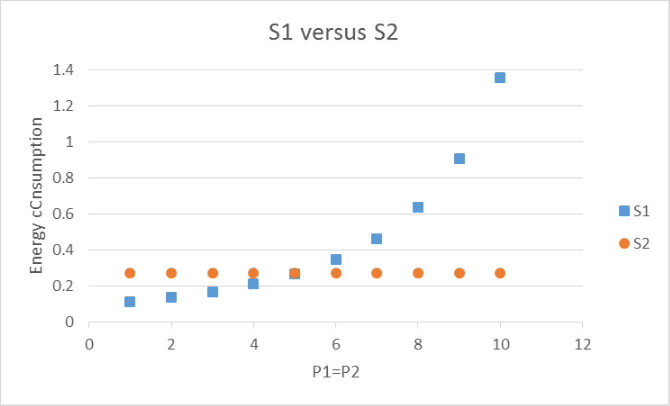


Figure 4.9: The effect of the P_1 and P_2 on the optimal cycle

Numerical Results for Different k Levels

At the first glance, we may assert that increasing the value of k will increase the energy consumption as well. However, the numerical studies show that any modification on k could have a different effect on the objective function. In the following example, we increase the value of k from 2 through 6. For S_1 cycle, we set $P_1 = P_2 = 3$. But three different levels are selected for the cycle time upper bound to show that the effect of any change in the value of k is not consistent; the consequent change in energy consumption can be monotonically increasing, decreasing or constant. The rest of the parameters are same with the mentioned values at the beginning of the Section 4.1.1. The change in the amount of energy consumption in S_1 cycle is shown in Table 4.13 and Figure 4.10. It can be seen that by considering three different levels for $\overline{Ct}$ equal to 17, 18, and 19, the behavior of the diagram can be completely dissimilar.

A more important finding is that as k increases, energy consumption function becomes more sensitive to the value of $\overline{Ct}$. In Table 4.13, for $k = 2$, it can be seen that when $\overline{Ct} = 17$, the energy consumption value is about two times greater than the energy consumption value when $\overline{Ct} = 19$. Whereas, for $k = 6$, when $\overline{Ct} = 17$, it is about 7.5 times greater than the energy consumption value when $\overline{Ct} = 19$.

We can see the same results for S_2 cycle in Table 4.14 and Figure 4.11.

The reason for having different behaviors for different cycle times is that for

Table 4.13: Numerical results of k on S_1 cycle under different policies for $\overline{Ct}$

	Energy Consumption		
k	$\overline{Ct} = 17$	$\overline{Ct} = 18$	$\overline{Ct} = 19$
2	24.93	17.31	12.72
3	29.77	17.23	10.85
4	35.62	17.18	9.27
5	42.65	17.14	7.93
6	51.11	17.12	6.79

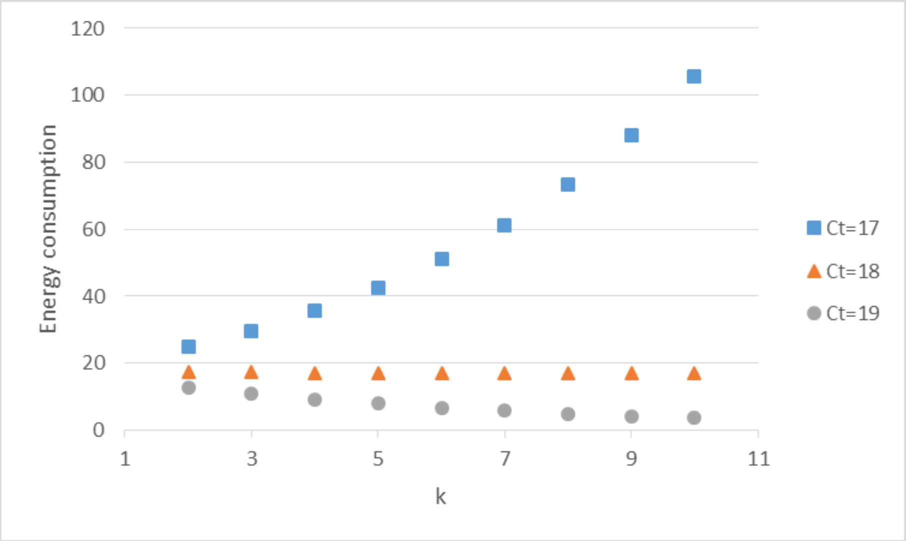


Figure 4.10: The effect of k on the energy consumption pareto frontier under different values for $\overline{Ct}$ in S_1 cycle

Table 4.14: Analytical results of k on S_2 cycle under different policies for $\overline{Ct}$

	Energy Consumption		
k	$\overline{Ct} = 17$	$\overline{Ct} = 18$	$\overline{Ct} = 19$
2	34.40	19.38	12.44
3	45.63	19.26	9.87
4	60.69	19.21	7.87
5	80.80	19.17	6.28
6	107.61	19.15	5.02

smaller cycle times, the robot should go faster and consequently the speeds are greater. If the values of the speeds are greater than 1, then the power k will increase their values drastically. On the other hand, if their values are very

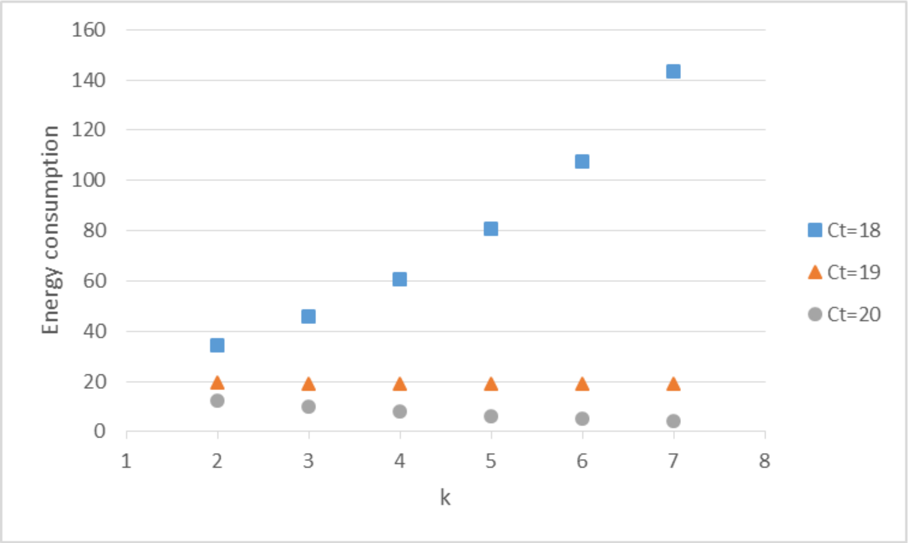


Figure 4.11: The effect of k on the energy consumption pareto frontier under different values for $\overline{Ct}$ in S_2 cycle

close to 1, there is no significant change in their values if we multiply them k times. Finally, by increasing the value of k, the energy consumption will decrease when the speed values are less than 1. In other words, the value of k can have an inverse effect on the energy consumption if the speed values are very small (< 1).

CHAPTER 5

SUMMARY, CONCLUSIONS AND FUTURE WORKS

The thesis addresses a flow shop robotic cell scheduling problem consisting two machines, each of which performs a different operation on each part, plus an input and output buffer, and one robot that moves linearly along a track to transport the parts through the machines. We deal with a bicriteria flow shop scheduling problem for optimizing the cycle time and energy consumption of the robot at the same time. We compare the energy consumption of different cycles (at different speeds). The objective function is formulated as the sum of the energy consumptions for every full and empty robot move in the system. The decision variables are the speeds of every robot activity. The approach is to derive the schedule from an energy consumption criterion according to the desired cycle time. To this effect, we look at the robot energy consumption optimization by optimizing the speed of the robot in each movement. Leveraging the variable speed of robot activities leading to different energy consumption levels under a considered upper bound for cycle time, we explore the potential for energy saving in two-machine robotic cells. In order to do this, we applied KKT analysis and the optimal situations in each feasible state are interpreted analytically for the unbounded case.

We developed the mathematical models for both S_1 and S_2 cycles. By the help of a controlled speed system, in which the speed of the robot is controllable, the models aim to minimize the energy consumption and cycle time which are two conflicting objectives. A set of theorems, lemmas, propositions and corollaries are proposed that can help us to find the optimal solution (speed values) to each specific situation for each cycle. Second, we presented an approximation algorithm to deal with robot speed upper bounds. Third, we combined both of the approaches corresponding to the bounded and unbounded cases and developed an algorithm that runs within a second and provides the optimal speed values in a two-machine robotic cell. Moreover, a number of pareto frontiers are offered in several instances for which correspond to the selected range of cycle time upper bound. Pareto frontiers can help the production planners to make a trade-off between the cycle time and energy consumption criteria. Additionally, the key observations with regard to the changes in parameters that might affect the behavior of the decision variables and the optimal solution are explained.

By means of the proposed approach of this thesis, which applies a controlled speed system instead of an uncontrolled system, we can get not only an economic return but also an environmental benefit through reducing carbon emissions by decreasing the need for electric power across the manufacturing sector.

This thesis has provided an overview about the green scheduling in a two-machine single gripper robotic cell producing identical parts. Here, we list the main areas that are still open.

- Determine multiple part-type case instead of producing identical parts.

- Compare the energy consumption between single and dual-gripper robotic cells.

- Reflect robot move acceleration.

- Reflect the energy consumption of the machines.

- Determine three-machine and m-machine cases.

REFERENCES

[1] Sebastian Panek, Olaf Stursberg, and Sebastian Engell. Optimization of timed automata models using mixed-integer programming. In *Formal Modeling And Analysis of Timed Systems*, pages 73–87. Springer, 2003.

[2] Avenir Kobetski and Martin Fabian. Time-optimal coordination of flexible manufacturing systems using deterministic finite automata and mixed integer linear programming. *Discrete Event Dynamic Systems*, 19(3):287–315, 2009.

[3] U.S. Department of Commerce. 2005 annual survey of manufacturers. https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?src=bkmk, Visited on Dec. 2016.

[4] Manufuture—strategic research agenda. *ManuFuture Platform*, 2006.

[5] Iain MacLeay. *Digest of United Kingdom energy statistics 2010.* The Stationery Office, 2010.

[6] Energy Efficiency. Tracking industrial energy efficiency and co2 emissions. *International Energy Agency*, 34(2):1–12, 2007.

[7] http://www.eia.gov/consumption/manufacturing. visited on Dec. 2014.

[8] A Fysikopoulos, D Anagnostakis, K Salonitis, and G Chryssolouris. An empirical study of the energy consumption in automotive assembly. *Procedia CIRP*, 3:477–482, 2012.

[9] T Chettibi, HE Lehtihet, M Haddad, and S Hanchi. Minimum cost trajectory planning for industrial robots. *European Journal of Mechanics-A/Solids*, 23(4):703–715, 2004.

[10] J Franke, S Kreitlein, F Risch, and S Guenther. Energy-efficient production strategies and technologies for electric drives. In *Industrial Technology*

(ICIT), 2013 IEEE International Conference on, pages 1898–1903. IEEE, 2013.

[11] Sebastian Thiede. *Energy efficiency in manufacturing systems.* Springer Science & Business Media, 2012.

[12] M Bornschlegl, M Drechsel, S Kreitlein, and J Franke. Holistic approach to reducing co2 emissions along the energy-chain (e-chain). In *Sustainable Automotive Technologies 2013*, pages 227–234. Springer, 2014.

[13] Fridtjof Unander. Decomposition of manufacturing energy-use in iea countries: How do recent developments compare with historical long-term trends? *Applied Energy*, 84(7):771–780, 2007.

[14] Jörg Engelmann. *Methoden und Werkzeuge zur Planung und Gestaltung energieeffizienter Fabriken.* IBF, 2009.

[15] Davis Meike and Leonids Ribickis. Energy efficient use of robotics in the automobile industry. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 507–511. IEEE, 2011.

[16] Anton Rassolkin, Hardi Hoimoja, and Raivo Teemets. Energy saving possibilities in the industrial robot irb 1600 control. In *2011 7th International Conference-Workshop Compatibility and Power Electronics (CPE)*, 2011.

[17] Zdenek Kolíbal and Anna Smetanova. Experimental implementation of energy optimization by robot movement. In *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, 2010.

[18] Friedrich Pfeiffer. Rolf isermann: Mechatronic systems—fundamentals. *Meccanica*, 43(4):459–460, 2008.

[19] Marcello Pellicciari, Angelo O Andrisano, Francesco Leali, and Alberto Vergnano. Engineering method for adaptive manufacturing systems design. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 3(2):81–91, 2009.

[20] Suresh P Sethi, Chelliah Sriskandarajah, Gerhard Sorger, Jacek Blazewicz, and Wieslaw Kubiak. Sequencing of parts and robot moves in a robotic

cell. *International Journal of Flexible Manufacturing Systems*, 4(3-4):331–358, 1992.

[21] Yves Crama and Joris Van De Klundert. Cyclic scheduling of identical parts in a robotic cell. *Operations Research*, 45(6):952–965, 1997.

[22] Milind Dawande, Chelliah Sriskandarajah, and Suresh Sethi. On throughput maximization in constant travel-time robotic cells. *Manufacturing & Service Operations Management*, 4(4):296–312, 2002.

[23] Leon van Os. The power of cyclic production. https://www.involvation.nl/en/articles/power-cyclic-production-2/, Visited on Sep. 2016.

[24] IG Drobouchevitch, S Sethi, J Sidney, and Chelliah Sriskandarajah. A note on scheduling multiple parts in two-machine dual gripper robotic cell: Heuristic algorithm and performance guarantee. *International Journal of Operations and Quantitative Management*, 10(4):297–314, 2004.

[25] R Neugebauer, M Putz, J Böhme, M Todtermuschke, and M Pfeifer. New aspects of energy consumption analysis in assembly processes and equipment. In *Sustainable Manufacturing*, pages 197–201. Springer, 2012.

[26] A Barili, M Ceresa, and C Parisi. Energy-saving motion control for an autonomous mobile robot. In *Industrial Electronics, 1995. ISIE'95., Proceedings of the IEEE International Symposium on*, volume 2, pages 674–676. IEEE, 1995.

[27] Milind W Dawande, H Neil Geismar, Suresh P Sethi, and Chelliah Sriskandarajah. *Throughput optimization in robotic cells*, volume 101. Springer Science & Business Media, 2007.

[28] Alessandro Agnetis. Scheduling no-wait robotic cells with two and three machines. *European Journal of Operational Research*, 123(2):303–314, 2000.

[29] Alessandro Agnetis and Dario Pacciarelli. Part sequencing in three-machine no-wait robotic cells. *Operations Research Letters*, 27(4):185–192, 2000.

[30] Ada Che, Chengbin Chu, and Eugene Levner. A polynomial algorithm for 2-degree cyclic robot scheduling. *European Journal of Operational Research*, 145(1):31–44, 2003.

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

[32] Vladimir Kats and Eugene Levner. Cyclic scheduling in a robotic production line. *Journal of Scheduling*, 5(1):23–41, 2002.

[33] Eugene Levner, Vladimir Kats, and Vadim E Levit. An improved algorithm for cyclic flowshop scheduling in a robotic cell. *European Journal of Operational Research*, 97(3):500–508, 1997.

[34] William D Hitz and Cecil R Stewart. Oxygen and carbon dioxide effects on the pool size of some photosynthetic and photorespiratory intermediates in soybean (glycine max [l.] merr.). *Plant physiology*, 65(3):442–446, 1980.

[35] Yves Crama, Vladimir Kats, J Van de Klundert, and Eugene Levner. Cyclic scheduling in robotic flowshops. *Annals of operations Research*, 96(1-4):97–124, 2000.

[36] Milind Dawande, H Neil Geismar, Suresh P Sethi, and Chelliah Sriskandarajah. Sequencing and scheduling in robotic cells: Recent developments. *Journal of Scheduling*, 8(5):387–426, 2005.

[37] W Baumann, R Birner, J Haeusler, and R-P Hartmann. Operating and idle times for cyclic multi-machine servicing. *Industrial Robot: An International Journal*, 8(1):44–49, 1981.

[38] DJ Medeiros, EF Watson, JS Carson, and MS Manivannan. Operational modeling & simulation in semiconductor manufacturing. In *Proceedings of the 1998 Winter Simulation Conference*.

[39] SY Nof and D Hanna. Operational characteristics of multi-robot systems with cooperation. *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH*, 27(3):477–492, 1989.

[40] Anthony S Kondoleon. *Cycle time analysis of robot assembly systems.* Society of Manufacturing Engineers, 1979.

[41] BH Claybourn. Scheduling robots in flexible manufacturing cells. *Chartered Mechanical Engineer*, 30:36–40, 1983.

[42] Jacek Błażewicz, Suresh P Sethi, and Chelliah Sriskandarajah. *Scheduling of robot moves and parts in a robotic cell.* École des hautes études commerciales, 1989.

[43] 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.

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

[45] Nadia Brauner and Gerd Finke. On a conjecture about robotic cells: new simplified proof for the three-machine case. *Infor-Information Systems and Operational Research*, 37(1):20–36, 1999.

[46] N Brauner and G Finke. Optimal moves of the material handling system in a robotic flow-shop. In *Proceedings IEPM*, volume 99, pages 409–417, 1999.

[47] Nadia Brauner and Gerd Finke. Cycles and permutations in robotic cells. *Mathematical and Computer Modelling*, 34(5):565–591, 2001.

[48] Nadia Brauner, Gerd Finke, and Wieslaw Kubiak. Complexity of one-cycle robotic flow-shops. *Journal of Scheduling*, 6(4):355–372, 2003.

[49] Nicholas G Hall, Hichem Kamoun, and Chelliah Sriskandarajah. Scheduling in robotic cells: Classification, two and three machine cells. *Operations Research*, 45(3):421–439, 1997.

[50] Nicholas G Hall, Hichem Kamoun, and Chelliah Sriskandarajah. Scheduling in robotic cells: Complexity and steady state analysis. *European Journal of Operational Research*, 109(1):43–65, 1998.

[51] Chelliah Sriskandarajah, Nicholas G Hall, and Hichem Kamoun. Scheduling large robotic cells without buffers. *Annals of Operations Research*, 76:287–321, 1998.

[52] M Selim Akturk, Hakan Gultekin, and Oya Ekin Karasan. Robotic cell scheduling with operational flexibility. *Discrete Applied Mathematics*, 145 (3):334–348, 2005.

[53] Hakan Gultekin, M Selim Akturk, and Oya Ekin Karasan. Cyclic scheduling of a 2-machine robotic cell with tooling constraints. *European Journal of Operational Research*, 174(2):777–796, 2006.

[54] Lei Lei and Tzyh-Jong Wang. Determining optimal cyclic hoist schedules in a single-hoist electroplating line. *IIE transactions*, 26(2):25–33, 1994.

[55] Haoxun Chen, Chengbin Chu, and Jean-Marie Proth. Cyclic scheduling of a hoist with time window constraints. *Robotics and Automation, IEEE Transactions on*, 14(1):144–152, 1998.

[56] Ada Che, Chengbin Chu, and Feng Chu. Multicyclic hoist scheduling with constant processing times. *Robotics and Automation, IEEE Transactions on*, 18(1):69–80, 2002.

[57] Raza Ur-Rehman, Stéphane Caro, Damien Chablat, and Philippe Wenger. Path placement optimization of manipulators based on energy consumption: application to the orthoglide 3-axis. *arXiv preprint arXiv:0910.4000*, 2009.

[58] Cory Bryan, Mitch Grenwalt, and Adam Stienecker. Energy consumption reduction in industrial robots. In *Proceedings ASEE North Central Sectional Conference*, 2010.

[59] A Smetanová. Optimization of energy by robot movement. *Modern Machinery Science Journal*, 3(1):172–176, 2010.

[60] Davis Meike and Leonids Ribickis. Analysis of the energy efficient usage methods of medium and high payload industrial robots in the automobile industry. In *10th International Symposium „Topical Problems in the Field of Electrical and Power Engineering "Pärnu, Estonia*, 2011.

[61] M Pellicciari, G Berselli, F Leali, and A Vergnano. A method for reducing the energy consumption of pick-and-place industrial robots. *Mechatronics*, 23(3):326–334, 2013.

[62] Alberto Vergnano, Carl Thorstensson, Bengt Lennartson, Petter Falkman, Marcello Pellicciari, Francesco Leali, and Stephan Biller. Modeling and optimization of energy consumption in cooperative multi-robot systems. *Automation Science and Engineering, IEEE Transactions on*, 9(2):423–428, 2012.

[63] Avenir Kobetski and Martin Fabian. Velocity balancing in flexible manufacturing systems. In *Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on*, pages 358–363. IEEE, 2008.

[64] Kai Li, Xun Zhang, Joseph Y-T Leung, and Shan-Lin Yang. Parallel machine scheduling problems in green manufacturing industry. *Journal of Manufacturing Systems*, 38:98–106, 2016.

[65] Yi-Chi Wang, Ming-Jun Wang, and Sung-Chi Lin. Selection of cutting conditions for power constrained parallel machine scheduling. *Robotics and Computer-Integrated Manufacturing*, 2015.

[66] Min Ji, Jen-Ya Wang, and Wen-Chiung Lee. Minimizing resource consumption on uniform parallel machines with a bound on makespan. *Computers & Operations Research*, 40(12):2970–2974, 2013.

[67] Dvir Shabtay and Moshe Kaspi. Parallel machine scheduling with a convex resource consumption function. *European Journal of Operational Research*, 173(1):92–107, 2006.

[68] S Afshin Mansouri, Emel Aktas, and Umut Besikci. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. *European Journal of Operational Research*, 248(3):772–788, 2016.

[69] Jian-Ya Ding, Shiji Song, and Cheng Wu. Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research*, 248(3):758–771, 2016.

[70] Wenwen Lin, DY Yu, Chaoyong Zhang, Xun Liu, Sanqiang Zhang, Yuhui Tian, Shengqiang Liu, and Zhanpeng Xie. A multi-objective teaching-learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint. *Journal of Cleaner Production*, 101:337–347, 2015.

[71] Dunbing Tang, Min Dai, Miguel A Salido, and Adriana Giret. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization. *Computers in Industry*, 81:82–95, 2016.

[72] Min Dai, Dunbing Tang, Adriana Giret, Miguel A Salido, and Wei Dong Li. Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing*, 29(5):418–429, 2013.

[73] Hao Luo, Bing Du, George Q Huang, Huaping Chen, and Xiaolin Li. Hybrid flow shop scheduling considering machine electricity consumption cost. *International Journal of Production Economics*, 146(2):423–439, 2013.

[74] AAG Bruzzone, D Anghinolfi, M Paolucci, and F Tonelli. Energy-aware scheduling for improving manufacturing process sustainability: a mathematical model for flexible flow shops. *CIRP Annals-Manufacturing Technology*, 61(1):459–462, 2012.

[75] Christian Gahm, Florian Denz, Martin Dirr, and Axel Tuma. Energy-efficient scheduling in manufacturing companies: a review and research framework. *European Journal of Operational Research*, 248(3):744–757, 2016.

[76] Adriana Giret, Damien Trentesaux, and Vittal Prabhu. Sustainability in manufacturing operations scheduling: a state of the art review. *Journal of Manufacturing Systems*, 37:126–140, 2015.

[77] Kuei-Tang Fang and Bertrand MT Lin. Parallel-machine scheduling to minimize tardiness penalty and power cost. *Computers & Industrial Engineering*, 64(1):224–234, 2013.

[78] Bing Du, Huaping Chen, George Q Huang, and HD Yang. Preference vector ant colony system for minimising make-span and energy consumption in a hybrid flow shop. In *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pages 279–304. Springer, 2011.

[79] Min Dai, Dunbing Tang, Yuchun Xu, and Weidong Li. Energy-aware integrated process planning and scheduling for job shops. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 229(1 suppl):13–26, 2015.

[80] Corinne Subai, Pierre Baptiste, and Eric Niel. Scheduling issues for environmentally responsible manufacturing: The case of hoist scheduling in an electroplating line. *International Journal of Production Economics*, 99(1): 74–87, 2006.

[81] Liping Zhang, Xinyu Li, Liang Gao, Guohui Zhang, and Xiaoyu Wen. Dynamic scheduling model in fms by considering energy consumption and schedule efficiency. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on*, pages 719–724. IEEE, 2012.

[82] Gilles Mouzon and Mehmet B Yildirim. A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, 1(2):105–116, 2008.

[83] Mehmet Bayram Yildirim and Gilles Mouzon. Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm. *IEEE transactions on engineering management*, 59(4):585–597, 2012.

[84] ChenGuang Liu, Jing Yang, Jie Lian, WenJuan Li, Steve Evans, and Yong Yin. Sustainable performance oriented operational decision-making of single machine systems with deterministic product arrival time. *Journal of Cleaner Production*, 85:318–330, 2014.

[85] Hakan Gultekin, M Selim Akturk, and Oya Ekin Karasan. Bicriteria robotic cell scheduling. *Journal of Scheduling*, 11(6):457–473, 2008.

[86] Hakan Gultekin, M Selim Akturk, and Oya Ekin Karasan. Bicriteria robotic operation allocation in a flexible manufacturing cell. *Computers & operations research*, 37(4):779–789, 2010.

[87] M Selim Akturk and Taylan Ilhan. Single cnc machine scheduling with controllable processing times to minimize total weighted tardiness. *Computers & Operations Research*, 38(4):771–781, 2011.

[88] Zeynep Uruk, Hakan Gultekin, and M Selim Akturk. Two-machine flowshop scheduling with flexible operations and controllable processing times. *Computers & Operations Research*, 40(2):639–653, 2013.

[89] George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

[90] Yongguo Mei, Yung-Hsiang Lu, Y Charlie Hu, and CS George Lee. Energy-efficient motion planning for mobile robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4344–4349. IEEE, 2004.

[91] John A Broderick. *Energy and Mobility Management of a Ground Robot to Increase Operational Capacity.* PhD thesis, University of Michigan, 2015.

APPENDIX

Algorithm 3: Finding the optimal robot speed values in a two-machine robotic cell

```
Data: $\varepsilon$, $k$, $P_1$, $P_2$, $\overline{Ct}$, $LB$, $UB$, $C_e$, $C_f$, and the set of distances
      $d_{ijh} \forall (ijh) \in \mathcal{D}_1$ and $\mathcal{D}_2$
Result: The optimal speed values ($v^*_{ijh}$) and the optimal objective
        function (OPT)
OPT $\leftarrow$ A big value (e.g. 1,000,000,000);
Check the feasibility of $S_1$ constraint by substituting all of the variables
 by $UB$;
if If equation (3.6) holds then
    Goto Algorithm 4
else
    $S_1$ cycle is inf.;
end
Check the feasibility of $S_2$ constraints by substituting all of the variables
 by $UB$;
if equations (3.17), (3.18), and (3.19) hold then
    Find $\mu_1$ for Sit. 1;
    if $\mu_1 > 0$ then
        Goto Algorithm 5;
    else
        Sit. 2 is inf.;
    end
    Find $\mu_1$ and $\mu_2$ for the Sit. 2;
    if $\mu_1 > 0$ and $\mu_2 \geq 0$ then
        Goto Algorithm 6;
    else
        Sit. 2 is inf.;
    end
    Find $\mu_1$ and $\mu_3$ for the Sit. 3;
    if $\mu_1 > 0$ and $\mu_3 \geq 0$ then
        Goto Algorithm 7;
    else
        Sit. 3 is inf.;
    end
    Calculate $LB_{z_4}$, $UB_{z_4}$ and $h(z_4)$;
    Goto Algorithm 1;
    Find $\mu_1$, $\mu_2$, and $\mu_3$ for the Sit. 4;
    if $\mu_1 > 0$ and $\mu_2$, $\mu_3 \geq 0$ then
        Goto Algorithm 8;
    else
        Sit. 4 is inf.;
    end
else
    $S_2$ cycle is inf.;
end
```

Algorithm 4: Finding the optimal robot speed values in S_1 cycle

```
Find $\mu_1$;
if $\mu_1 > 0$ then
    Find $v_{ijh} \forall (ijh) \in \mathcal{D}_1$;
    if any of the speeds is greater than $UB$ then
        Consider the bounded speed case for $S_1$ and solve the problem;
        Find $\mu_1$;
        if $\mu_1 > 0$ then
            Find $v_{ijh} \forall (ijh) \in \mathcal{D}_1$;
            $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_1$;
            Calculate the obj. func.;
            OPT $\leftarrow$ obj. func.;
        else
            $S_1$ cycle is inf.;
        end
    else
        $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_1$;
        Calculate the obj. func.;
        OPT $\leftarrow$ obj. func.;
    end
else
    $S_1$ cycle is inf.;
end
```

Algorithm 5: Finding the optimal robot speed values in Situation 1 of S_2 cycle

```
Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
if any of the speeds is greater than $UB$ then
    Consider the bounded speed case and solve the problem;
    Find $\mu_1$;
    if $\mu_1 > 0$ then
        Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        Calculate the obj. func.;
        if obj. func. $< OPT$ then
            $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
            OPT $\leftarrow$ obj. func.;
            Stop Algorithms 5 and 3;
        else
            Stop Algorithms 5 and 3;
        end
    else
        Sit. 2 is inf.;
    end
else
    $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
    Calculate the obj. func.;
    if obj. func. $< OPT$ then
        $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        OPT $\leftarrow$ obj. func.;
        Stop Algorithms 5 and 3;
    else
        Stop Algorithms 5 and 3;
    end
end
```

Algorithm 6: Finding the optimal robot speed values in Situation 2 of S_2 cycle

```
Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
if any of the speeds is greater than $UB$ then
    Consider the bounded speed case and solve the problem;
    Find $\mu_1$ and $\mu_2$;
    if $\mu_1 > 0$ and $\mu_2 \geq 0$ then
        Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        Calculate the obj. func.;
        if obj. func. $<$ OPT then
            $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
            OPT $\leftarrow$ obj. func.;
            Stop Algorithms 6 and 3;
        else
            Stop Algorithms 6 and 3;
        end
    else
        Sit. 2 is inf.;
    end
else
    $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
    Calculate the obj. func.;
    if obj. func. $<$ OPT then
        $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        OPT $\leftarrow$ obj. func.;
        Stop Algorithms 6 and 3;
    else
        Stop Algorithms 6 and 3;
    end
end
```

Algorithm 7: Finding the optimal robot speed values in Situation 3 of S_2 cycle

```
Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
if any of the speeds is greater than $UB$ then
    Consider the bounded speed case and solve the problem;
    Find $\mu_1$ and $\mu_3$;
    if $\mu_1 > 0$ and $\mu_3 \geq 0$ then
        Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        Calculate the obj. func.;
        if obj. func. $<$ OPT then
            $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
            OPT $\leftarrow$ obj. func.;
            Stop Algorithms 7 and 3;
        else
            Stop Algorithms 7 and 3;
        end
    else
        Sit. 3 is inf.;
    end
else
    $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
    Calculate the obj. func.;
    if obj. func. $<$ OPT then
        $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        OPT $\leftarrow$ obj. func.;
        Stop Algorithms 7 and 3;
    else
        Stop Algorithms 7 and 3;
    end
end
```

Algorithm 8: Finding the optimal robot speed values in Situation 4 of S_2 cycle

```
Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
if any of the speeds is greater than $UB$ then
    Consider the bounded speed case and solve the problem;
    Find $\mu_1$, $\mu_2$,and $\mu_3$;
    if $\mu_1 > 0$ and $\mu_2$, $\mu_3 \geq 0$ then
        Find $v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        Calculate the obj. func.;
        if obj. func. $<$ OPT then
            $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
            OPT $\leftarrow$ obj. func.;
            Stop Algorithms 8 and 3;
        else
            Stop Algorithms 8 and 3;
        end
    else
        Sit. 4 is inf.;
    end
else
    $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
    Calculate the obj. func.;
    if obj. func. $<$ OPT then
        $v^*_{ijh} \leftarrow v_{ijh} \forall (ijh) \in \mathcal{D}_2$;
        OPT $\leftarrow$ obj. func.;
        Stop Algorithms 8 and 3;
    else
        Stop Algorithms 8 and 3;
    end
end
```