SEQUENCING AND SCHEDULING OPTIMIZATION IN LOW-VOLUME LOW-VARIETY PRODUCTION SYSTEMS

By

Arya Russell, P.Eng

Master of Engineering, Mechanical and Industrial Engineering, Ryerson University, 2017

Bachelor of Engineering, Industrial Engineering, Ryerson University, 2014

A Dissertation

Presented to Ryerson University

In Partial Fulfillment of the

Requirements for the degree of

Doctor of Philosophy

In the program of

Mechanical and Industrial Engineering

Toronto, Ontario, Canada

AUTHOR’S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Sequencing and Scheduling Optimization in Low-Volume Low-Variety Production Systems

Doctor of Philosophy
Mechanical and Industrial Engineering
Ryerson University

Abstract

Despite prominent scholarly advancements in scheduling optimization approaches for a wide range of production systems, limited research has been reported on sequencing and scheduling optimization strategies in Low-Volume Low-Variety production systems. This dissertation fills the gap in the current literature through the formulation and the proposal of a suite of mathematical programming models and heuristic algorithms, capturing the unique characteristics and constraints inherent in such production systems. In the first section of this dissertation a suite of mixed-integer multi-objective linear mathematical programming models are proposed for solving discrete-time single work center scheduling problems, distinguished by a key decision criterion of permitting or prohibiting the traveling of incomplete activities. It was evident through personal observations however, that there exist scenarios where resources are shared between parallel work centers, which yielded to further research in the use of shared resource pools in multi-parallel work center scheduling problems. A novel suite of mathematical programming models are proposed for solving single and multi-parallel work center scheduling problems with shared or dedicated resources. The mathematical programming models formulated in this section are modular, signifying that constraints can be added or removed without jeopardizing the integrity of the mathematical models. The proposed optimization models were validated and verified through a real-world case study where significant cost savings in form of resource requirements are realized through the integration of shared resource pools. It is often the case however, that activity processing times and planning horizon are not discrete. To tackle continuous-time work center scheduling problems a novel suite of mathematical programming models is formulated and proposed in the final section of this dissertation, as well as two new genetic algorithms for solving large-scale scheduling problems. The proposed mathematical programming models and metaheuristics are aimed at optimizing the production schedule as well as activity execution sequence to minimize overall cost and resource requirements. The optimization models proposed through this dissertation are validated and verified through a real-world case study of the final assembly line of a narrow body private aircraft, where the problems were solved to optimality.

Acknowledgments

First and foremost, I would like to thank my supervisor, Dr. Sharareh Taghipour for opening the door for me to this great program. I must recognize the inspiration, wisdom and professionalism that Dr. Sharareh Taghipour has shown me. She set a high academic standard but offered guidance and support whenever requested. I have enjoyed our countless hours of fruitful discussions and I have learned an incredible amount from her during my time at Ryerson University. Without Dr. Taghipour's support, encouragement, and constructive criticism, this thisis would not have been possible.

I would also like to extend my gratitude to my supervisory committee members, Dr. Saeed Zolfaghari and Dr. Saman Amin for their continuous support, valuable insight and important contributions in the betterment of this work.

And last but not least, I wish to express my sincere gratitude to my family, my friends, and particularly my parents for their endless and unconditional love, support and encouragement. None of this would have been possible without them, and I am eternally grateful.

Table of Contents

List of Tables

Introduction

Discrete-Time Single Work Center

Discrete-Time Parallel Work Centers

Continuous-Time Single Work Center

List of Figures

Introduction

Discrete-Time Single Work Center

Discrete-Time Parallel Work Centers

Continuous-Time Single Work Center

CHAPTER 1

INTRODUCTION

Sequencing and scheduling optimization problems have been subject to extensive research since the early days of operations research [1,2,3]. The primary motivation of such research is the replacement of the traditional decision-making process in the scheduling of activities and orders with a systematic approach, often solved using mathematical programming models, constraint programing, or heuristic algorithms [4,5,6,7]. Scheduling optimization problems can be classified by their time representations as either discrete or continuous [8,9,10], as well as material balances, where products can be processed individually or in batches [11,12,13,14]. Furthermore, depending on the type of the problem, the organizational objectives, and the priorities and the preferences of the decision-makers, scheduling optimization problems may aim to minimize makespan [15,16,17,18], earliness [19,20,21,22], tardiness [23,24,25,26], on-hand inventory [27,28,29], or cost [30,31,32,33,34,35]. As such, it is crucial to understand any scheduling problem at hand with full alignment with stakeholders prior to formulating or adopting a previously established optimization model. It is important to note, however, that while optimization models can be classified based on their time representation, material balance, and objectives, the reported scheduling optimization models are *not* interchangeable between the different types of production systems. For instance, continuous-time scheduling optimization models with the objective of minimizing cost for job-shop scheduling problems cannot be adapted and employed in modeling and solving continuous-time scheduling problems in an oil refinery [36,37,38,39,40]. It is thus critical to understand not only the defining parameters for scheduling problems but to also comprehend and accurately reflect the behavior and characteristics inherent in a specific production system. Production systems can be distinguished and classified based on the volume of produced goods as well as the degree of variety between products. For the purpose of this dissertation, production systems are classified into three main classifications, namely High-Volume Low-Variety Production Systems (HVLVPS), Low-Volume High-Variety Production Systems (LVHVPS), and Low-Volume Low-Variety Production systems (LVLVPS).

1.1. Background & Literature Review

Low-Volume Low-Variety Production Systems (LVLVPS) are classified as a hybrid form of High-Volume Low-Variety Production Systems (HVLVPS) and Low-Volume High-Variety Production Systems (LVHVPS) [41]. Products assembled in LVLVPS are subject to minimal variation in product configuration and exhibit high unit-costs and long lead-times [42]. Examples of which include the final assembly of aircraft, heavy aero-structures, empennages, cockpits, wings, emergency response vehicles, and heavy mining and military equipment. Products assembled in such production systems follow a pre-defined processing order through a series of unique manufacturing cells, referred to as work centers, responsible to complete the pre-defined statement of work with the budgeted resources over the span of the imposed takt-time. Takt time, in the context of manufacturing, is referred to as the drumbeat of the assembly line or the available processing time at each work center, equivalent to the inter-departure time between two consecutive products [43,44]. Similar product flow, to that of LVLVPS, is exhibited in HVLVPS, commonly referred to as *Job Shops* [45,46,47,48] and *Flow-Shops* [49,50], differentiated by their degree of variety in product configuration. Products assembled in HVLVPS, exhibit lower unit-costs, and shorter lead-times, following a pre-defined processing order through a series of machines, where each machine is capable of performing a task or series of tasks. The job is completed once all tasks are successfully executed, and the completion time of a job is equivalent to the completion time of the final task [51]. Reported mathematical programming models and algorithms developed for solving scheduling problems in HVLVPS are aimed at minimizing lateness, tardiness, or makespan in completion of orders, referred to as jobs, through the optimum allocation of activities to machines, where each job is subject to a pre-specified due date [52,53,54,55,56,57,58,59,60,61,62,63,64,65]. Despite the similarities exhibited between the two production systems, the mathematical programming models and heuristics reported for HVLVPS cannot be directly adopted in solving scheduling problems in LVLVPS. This is primarily due to differences in resource capabilities and profiles, and the incompatibility of the reported literature in the modeling of characteristics and constraints unique to LVLVPS. Contrary to the allocation of a single machine to a workstation, programmed to execute a task, as is the case of HVLVPS, multiple classifications of multi-skilled human resources are assigned to a work center in LVLVPS, responsible to complete a pre-defined statement of work, comprised of a set of multi-resource and interdependent activities. Furthermore, while the mathematical programming models in solving HVLVPS are aimed at attaining the optimum allocation of products to machines, the optimization models for LVLVPS are formulated to optimize the allocation of resources to the activities assigned to a work center. In scheduling optimization of LVLVPS with the mathematical programming models developed for solving Job Shop scheduling problems with parallel machines [66,67], each human resource may be considered as a machine, capable of performing a wide range of tasks and activities. The problem then becomes a single job scheduling problem with a due date equivalent to the takt-time of the assembly line. The problem can be further extended to a multi-job problem through the integration of parallel work centers where due dates are staggered, offset by a function of takt-time and the number of parallel work centers. However, the mathematical programming models developed for solving Job Shop scheduling problems fail to consider key constraints and characteristics exhibited in LVLVPS. Activities assigned to work centers are highly interdependent and may be executed in one of the available modes. This characteristic cannot be effectively modeled in job shop scheduling problems, as the addition of resources in crashing or fast-tracking of an activity cannot be accommodated in machine shops. Furthermore, human resources assigned to a work center in LVLVPS are simultaneously utilized for the duration of the takt, where resources are classified as multi-skilled, signifying their competency in performing multiple types of activities. For instance, in the case of an aerospace assembly line, a mechanical assembler is capable of drilling holes, deburring, tightening fasteners, swaging pipes, performing bonding checks, and executing reworks. In the case of HVLVPS however, jobs are transferred between machines once an activity is completed. This is in addition to the models' inability in capturing the simultaneous use of all resources on the same job, resulting in deficiencies in capturing constraints such as zonal densities, concurrencies, and non-concurrencies. Furthermore, Job Shop scheduling problems are considered as resource-constrained scheduling problems, whereas in the mathematical programming models proposed in this paper, deviation to the aspiration criterion for resources is permitted.

Resource profiles and task assignments, similar to that of LVLVPS, are most common in LVHVPS, where products are held in position while varying quantities and classifications of resources are continuously deployed onto the product to complete a pre-defined set of interdependent single and multi-mode activities [68,69,70,71]. LVHVPS are commonly referred to as projects, exhibiting long lead-times and high unit-costs, in assembly or construction of unique products [72,73,74,75]. Examples of which include the assembly of ships, submarines, and construction of architectural structures. Project scheduling problems are aimed at optimizing the activity execution sequence through the optimum allocation of resources to activities. Project scheduling problems are further classified as either *Resource-Constrained Project Scheduling Problems* (RCPSP) with the objective of minimizing makespan [76,77,78,79,80,81,82,83,84,85,86,87] or *Time-Constrained Project Scheduling Problems* (TCPSP) aimed at minimizing resource requirements [88,89,90,91,92]. Extensive literature has been reported on mathematical programming approaches and metaheuristics such as evolutionary algorithms and simulated annealing on modeling and solving project scheduling problems [93,94,95]. The GA developed for solving such problems has adapted a unique chromosome representation to capture precedence and are found effective and efficient in solving single and multi-mode RCPSP with [96]. However, despite the similarities exhibited between the two production systems, the optimization models developed for solving RCPSP and TCPSP are found deficient in modeling and solving scheduling problems in LVLVPS. In modeling of scheduling problems in LVLVPS, using optimization models developed for LVHVPS, each work center may be considered as a project, with a mathematical programming model or heuristic aimed at optimizing the activity execution sequence and resource allocation to minimize makespan or resource requirements. This can be demonstrated through a Start/End Event-Based RCPSP approach, developed by Kone et.

al (2011) [85] and adopted by Borreguero et. al (2015) [97] in solving work center scheduling problems in the final assembly line of aircraft. The objective of this mixed-integer linear programming model is to minimize the makespan for completion of all activities, subject to precedence, resource, and non-concurrency constraints. The Start/End Event-Based formulation, proposed by Zapata et al. (2008), defines a series of events corresponding to the start or the end of activities. The proposed optimization model is based on the assumption that for RCPSP, there exists an optimal schedule, such that activities start at either $t = 0$, or upon the completion of another activity. An alternative approach to modeling of RCPSP is the discrete-time mixed-integer linear mathematical programming model proposed by Pritsker et al. (1969) in solving multi-project scheduling problems with shared resources [84]. While this mathematical model has not been employed in solving scheduling problems in LVLVPS, its capability in capturing multi-mode and multi-resource activities, non-concurrencies, and job splitting can be exploited. The proposed "Zero-One Programming Model" uses a binary decision variable to solve for the optimum start-time of activities with the objective of minimizing makespan, lateness or total throughput time for all projects. Whilst the mathematical programming model proposed by Borreguero et al. (2015) was solved to optimality on multiple instances, we argue that the resultant solution is not practical due to deficiencies in capturing key characteristics inherent in LVLVPS. This is primarily due to a lack of an upper bound on time, and the proposed objective function, aimed at minimizing the makespan. Work centers in LVLVPS are responsible to complete the pre-defined statement of work over the span of the imposed takt time, and the product must move downstream once the takt-time has elapsed. Negative or positive deviation of makespan from the imposed takt-time will result in excess labor costs in the form of resource idle times or assembly line stoppage, respectively. The lack of an upper bound and an aspiration criterion for time, in the mathematical models driven from RCPSP, results in schedules found unrealistic from a practical standpoint. Similarly, the employment of TCPSP approaches, as proposed by Möhring et al. (1984), in scheduling optimization of activities in LVLVPS may result in naïve solutions, where the required number of resources may exceed the budgeted or the available number of human resources. It can thus be concluded that while there exist similarities in constraints and characteristics between projects and LVLPS, the adaptation of project scheduling approaches in solving work center scheduling problems will result in unrealistic schedules. Furthermore, contrary to multi-project scheduling problems with shared resources

[98],multi-parallel scheduling problems with shared resources in LVLVPS have not received much scholarly attention. Figures 1.01 and 1.02 depicted below, illustrate the production layout of the final assembly line of a narrow-body aircraft and the work center layout respectively, demonstrating the movement of products between work centers and the deployment of human resources onto the aircraft.

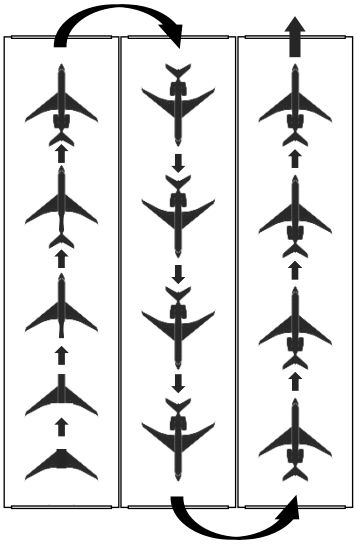

Figure 1.02 – Production Layout

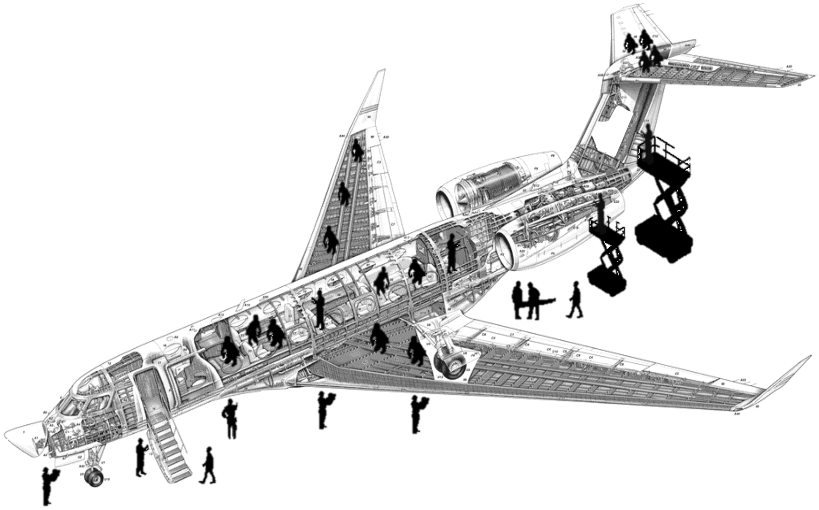

Figure 1.01 – Work Center Layout

1.2. Research Contributions

Despite the notable scholarly advancements in the optimization of scheduling problems for a wide range of production systems through mathematical programming models, constraints programming and metaheuristics, limited research has been reported on scheduling optimization approaches for LVLVPS. This dissertation fills the gap in the current literature through the formulation of a suite of mathematical programming models and a novel Genetic Algorithm (GA) proposed for solving large-scale scheduling problems in LVLVPS. In Chapter 3, three new discrete-time multi-objective mixed-integer linear mathematical programming models are formulated and proposed for solving single work center scheduling problems, differentiated by their objectives, underlying assumptions, and constraints. The initial model adopts a pre-emptive goal-programming approach in minimizing the number of resources required in the completion of the maximum number of activities, where travel work is permitted. The second mathematical programming model is an extension to the first, and is aimed at solving scheduling problems in scenarios prohibiting travel work, with the objective of minimizing the positive deviation of makespan and resource requires to their corresponding aspiration criteria, in the completion of all activities assigned to the work center. Furthermore, a third mixed-integer mathematical programming model is proposed for evaluating a work center's maximum capacity through the complete saturation of its resources. In Chapter 4 the implications involved in the integration of multi-parallel work centers are investigated and explored and a series of discrete-time multi-objective mixed-integer linear mathematical programming models are formulated and proposed for solving multi-parallel scheduling problems with shared or dedicated resource pools. The first mathematical programming model adopts the lexicographic method and is proposed to be used in scenarios permitting the traveling of incomplete activities, aimed at maximizing the number of completed activities. In modeling of scenarios prohibiting travel work, an alternative approach is proposed, employing a pre-emptive goal-programming model, aimed at minimizing the positive deviation to the aspiration criteria for time and resources. To tackle scheduling problems with continuous-time (non-integer) processing times and planning horizons, a new set of mathematical programming models are formulated in addition to a novel GA in Chapter 5. The proposed continuous-time mathematical programming models adopt a pre-emptive goal programming approach in the formulation of a suite of priority-based multi-objective optimization models, capturing all characteristics and constraints inherent in LVLVPS. Similar to Chapter 3, two new mathematical programming models are formulated and are proposed to be used in scenarios permitting or prohibiting the traveling of incomplete activities, with the objectives of minimizing the number of incomplete activities, and minimizing the positive deviation to the aspiration criteria to time and resources respectively. Two metaheuristics employing GA have also been developed to tackle large-scale scheduling problems in LVLVPS. The proposed mathematical programming models and metaheuristics are validated and verified through a real-world case study of a work center in the final assembly line of a narrow-body, dual-jet business aircraft. Through these case studies, it can be demonstrated that the proposed optimization models are effective in modeling and solving large-scale scheduling problems in LVLVPS, capturing the constraints and characteristics inherent in such production systems. The contributions made through this dissertation are as follows:

[1] Discrete-time multi-objective mixed-integer programming model for single work center scheduling problems, permitting the traveling of incomplete activities with strict time and resource constraints, aimed at minimizing the number of incomplete activities on-time with the budgeted number of resources.

[2] Discrete-time multi-objective mixed-integer programming model for single work center scheduling problems, mandating the completion of the imposed statement of work, aimed at minimizing the positive deviation to the aspiration criteria to time and resources while minimizing the overall resource requirements.

[3] Discrete-time multi-objective mixed-integer programming model for single work center scheduling problems, aimed at minimizing the overall makespan and resource requirements where the former takes precedence. This mathematical programming model is proposed to be used in evaluating a work center's capability in satisfying the foreseen takt-times, given the complete saturation of its resources.

[4] Discrete-time multi-objective mixed-integer programming model for single and multi-parallel work center scheduling problems with dedicated resources aimed at minimizing the number of incomplete activities, on-time with the budgeted number of resources, while minimizing resource requirements if possible. This mathematical programming model is proposed to be used in scenarios where the strict enforcement of time and resource constraints may lead to the traveling of incomplete activities.

[5] Discrete-time multi-objective mixed-integer programming model for single and multi-parallel work center scheduling problems with shared resources aimed at minimizing the number of incomplete activities, on-time, and on-budget, while minimizing overall resource requirements. The resource pool in these scenarios is assumed to be shared between the parallel work centers, thus improving overall resource utilizing and further minimizing the number of incomplete activities.

[6] Discrete-time multi-objective mixed-integer programming model for single and multi-parallel work center scheduling problems with dedicated resources, mandating the completion of the imposed statement of work in its entirety. The proposed optimization model is aimed at minimizing the positive deviation to the aspiration criteria to time and resources, while maximizing the negative deviation to the aspiration criterion to resources, thus minimizing the overall resource requirements. In the case of dedicated resource pools, contrary to the shared resource problem, each work center is budgeted with a dedicated set of resources, responsible to complete the predefined work package.

[7] Discrete-time multi-objective mixed-integer programming model for single and multi-parallel work center scheduling problems with shared resources, mandating the completion of the imposed statement of work. The proposed mathematical programming model is aimed at minimizing the positive deviation to the aspiration criteria to time and resources while maximizing the negative deviation to the aspiration criterion to resource budgets.

[8] Continuous-time multi-objective mathematical programming model for single work center scheduling problems, aimed at minimizing the number of incomplete activities in scenarios permitting travel work as a consequence to the strict enforcement of time and resource constraints. The proposed optimization model is aimed at minimizing the number of resources required in the completion of the maximum number of activities, where the latter takes precedence.

[9] Genetic Algorithm for single work center scheduling problems with continuous-time planning horizons and processing times, aimed at minimizing the number of resources required in the completion of the maximum number of activities, where the latter takes precedence. Genetic Algorithm is employed as an alternative to mathematical programming models to enable efficient modeling and solving of the complex large-scale industrial problems.

[10] Continuous-time multi-objective mathematical programming model for single work center scheduling problems, aimed at minimizing the positive deviation to the aspiration criteria to time and resources while maximizing the negative deviation to the aspiration criterion to resources in an effort to minimize overall resource requirements. The proposed mathematical programming model is applicable to scheduling problems mandating the completion of the statement of work in its entirety through allowances to previously established time and resource budgets.

[11] Genetic Algorithm for single work center scheduling problems with continuous-time planning horizons and processing times, aimed at minimizing the positive deviation to the aspiration criteria to time and resources, while maximizing the negative deviation to the aspiration criterion to resource budgets. The proposed evolutionary algorithm is proposed to be used in scenarios mandating the completion of the imposed statement of work in its entirety and is adapted to tackle large-scale real-world industrial problems.

The problem under study is classified as NP-Hard in the strong sense, and is considered as one of the most intractable combinatorial optimization problems. According to the complexity theory, an optimization problem is NP-Hard in the strong sense if its decision version is NP-Complete in the strong sense [105]. Work center scheduling problems in LVLVPS with shared resources exhibits all characteristics of RCPSP, as such the decision variant of this problem, similar to that of an RCPSP is NP-Complete in the strong sense as proven by Johnson et al. (1975) through reduction from the 3-partition problem [106]. Further research has concluded the NP-Hardness of RCPSP [107]. It is thus concluded that scheduling problems in LVLVPS are classified as NP-Hard in the strong sense, as they exhibit all characteristics and constraints inherent in RCPSP and more.

Table 1.01 provides a summary of characteristics and constraints inherent in the proposed optimization models with a comparison to the state-of-the-art to demonstrate similarities to previously established methodologies and to highlight deficiencies of the reported literature in modeling and solving scheduling problems in LVLVPS.

Reported Literature	Resource Constraint	Time Constraint	Multiple Parallel Work Centers	Single & Multi-Mode Activities	Precedence Constraints	Multi-Skilled Resources	Multi-Resource Activities	Multiple Types Of Resource	Travel Work Flexibility	Earliest & Latest Start	Earliest & Latest Finish	Zonal Constraints	Non-Concurrency	Concurrent Start & Finish	Discrete-Time Scheduling	Continuous-Time Scheduling
02. Russell & Taghipour, 2019 *(Accepted)*	✓	✓		✓	✓	✓	✓	✓	✓			✓	✓		✓	
01. Russell & Taghipour, 2020 *(Accepted)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
03. Russell & Taghipour, 2020 *(Under Review)*	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
04. Guldemond, Hurink, & Paulus, 2005		✓		✓	✓	✓	✓	✓								
05. Hurink, Kok, Paulus, & Schutten, 2011		✓	✓	✓	✓	✓	✓	✓								
06. Borreguero, Mas, Menéndez, 2015	✓			✓	✓	✓	✓	✓								
07. Deckro & Hebert, 1989	✓				✓	✓	✓	✓		✓						
08. Christofides, 1987	✓				✓	✓	✓	✓								
09. Pritsker, Waiters, & Wolfe, 1969	✓		✓	✓	✓	✓	✓	✓				✓	✓			
10. Möhring, 1984		✓			✓	✓	✓	✓								
11. Kone, Artigues, Lopez, 2011	✓		✓	✓	✓	✓	✓	✓								
12. Drezet & Billaut, 2008	✓			✓	✓	✓	✓	✓								
13. Browning & Yassine, 2010	✓		✓		✓		✓									
14. Naber & Kolisch, 2014	✓				✓	✓	✓	✓								
15. Grigoriev, Kutin, & Turkin, 2013	✓				✓					✓	✓					
16. Beşikci, Bilge, & Ulusoy, 2015	✓		✓	✓	✓	✓	✓	✓								
17. Borreguero, García, & Ortega, N.D.	✓			✓	✓	✓	✓	✓								

Table 1.01 - Comparison of Contributions to the State-Of-The-Art

The structure of this dissertation is as follows: Chapter 2 is dedicated to the formulation of a novel mixed-integer linear multi-objective mathematical programming model proposed for solving large-scale discrete-time single-work center scheduling problems in LVLVPS. Three distinct mathematical models are proposed in this chapter, all of which are validated and verified through a real-world case study with a global leader in the aerospace industry. In Chapter 3, a new set of mixed-integer linear multi-objective mathematical programming models are formulated for modeling and solving discrete-time multi-parallel work center scheduling problems with dedicated or shared resource pools in LVLVPS. The proposed mathematical programming models are similarly validated and verified through a real-world case study, where the benefits of the integration of parallel work centers with shared resources are highlighted. A new suit of mixed-integer mathematical programming models are formulated and proposed in Chapter 4 to tackle scheduling problems with continuous-time planning horizons and processing times, accompanied by a case study of the final assembly line of a narrow body, dual-engine private aircraft assembly line.

1.3. Publications

This dissertation is based on a series of published journal articles listed below:

[1] A. Russell, S. Taghipour, Multi-Objective Optimization of Complex Scheduling Problems in Low-Volume Low-Variety Production Systems, *International Journal of Production Economics, 208* (2019), 1-16.

[2] A. Russell, S. Taghipour, Multi-Parallel Work Centers Scheduling Optimization with Shared or Dedicated Resources in Low-Volume Low-Variety Production Systems, *Applied Mathematical Modelling, 80* (2020), 472-505

[3] A. Russell, S. Taghipour, Mathematical Programming and Metaheuristics for Solving Continuous-Time Scheduling Optimization Problems in Low-Volume Low-Variety Production Systems, *Computer and Industrial Engineering, Under Review*

CHAPTER 2

DISCRETE-TIME SINGLE WORK CENTER

In this chapter, a suit of mixed-integer linear multi-objective mathematical programming models is formulated and proposed for solving discrete-time single work center scheduling problems in LVLVPS. Products assembled in LVLVPS, follow a pre-defined processing order through a series of unique work centers, each budgeted with multiple classifications of multi-skilled human resources, responsible to complete the pre-defined statement of work, over the span of the imposed takt-time. The statement of work assigned to each work center consists of a set of multi-resource and multi-mode activities, where activities may require one or more resources of distinct classifications and may be executed in one of the available modes. In the context of scheduling, multi-mode activities refer to a subset of activities that can be crashed or fast-tracked, where each mode of an activity corresponds to the activity's expected processing time that is negatively correlated to the assigned number of resources, suggesting that the activity can be crashed in duration through the assignment of additional resources. Moreover, activities are highly interdependent and may be imposed to lead or lag times, where lag time refers to an imposed delay between activity completion and the starting time of its successor(s) while lead time provides an allowance for an activity to start prior to the completion of its predecessor(s). Each activity is assigned to a specific zone, representing the physical location of work, where each zone is subject to maximum allowable capacity. There may also exist non-concurrency constraints between two or more zones and activities, restricting their simultaneous progression, an example of which includes the simultaneous progression of activities in the main landing gear bay of an aircraft, while performing the functional testing of the main landing gears which involves the swinging of the gears into the main landing gear bay area. Furthermore, activities are assumed to be non-preemptive, suggesting that once an activity has started, it cannot be paused or interrupted, and must progress to completion. Figure 2.01 illustrates the sequence of products processed in a single work center, signifying that a positive deviation of the makespan to the takt-time cannot be accommodated, while a negative deviation will neither increase throughput nor decrease cost. Figure 2.02 demonstrates an example of a standardized scheduled to be executed in a work center, the optimization of which is the objective of this chapter.

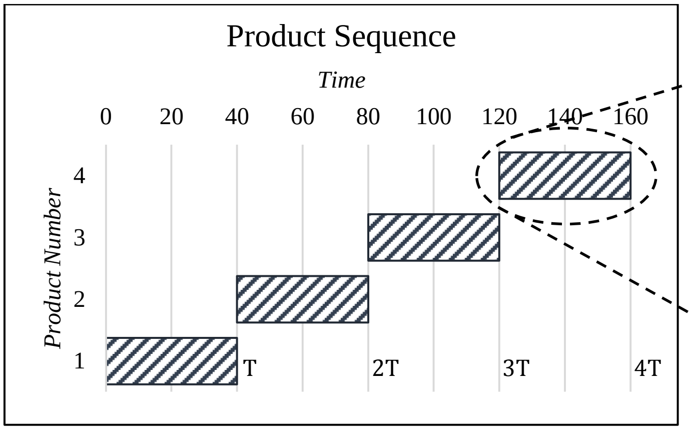

Figure 2.01 - Product Sequence through Work Centers

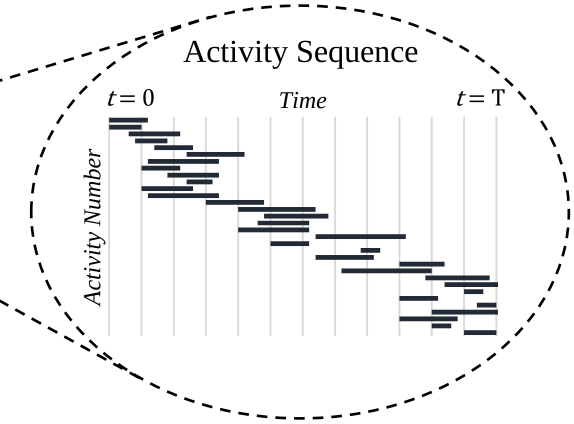

Figure 2.02 - Standardized Activity Schedule

While the aspiration criteria for this problem is the completion of all activities on-time and on-budget, the imposed constraints may result in scenarios where only a portion of the pre-defined statement of work is feasible to be completed with the budgeted resources over the span of the takt-time. Incomplete activities are scheduled in downstream work centers, as their completion in the originating work center will result in a positive deviation to the aspiration criteria for time and/or resources. Travel work can only include non-critical activities, where critical activities are defined as a subset of activities that must be completed in-station, examples of which include activities requiring specialized tooling and equipment stationed at the work center. Table 2.01

provides a summary of activity attributes and generic assumptions, as well as constraints shared between the models proposed in this chapter.

Activity Attributes	*Assumptions*	*Constraints*
• Processing Time • Interdependencies • Lead & Lag Times • Non-Concurrencies • Multi-Mode Activities • Resource Classification • Resource Quantity • Zonal Assignment • Criticality	• Discrete-Time Processing Time • Discrete-Time Planning Horizon • Pre-emption Not Allowed • Equal Resource load per Shift • Equal Efficiency for all Resources	• Limited Resources • Limited Planning Horizon • Precedence Constraints • Lead & Lag Time Constraints • Zonal Capacity Constraints • Completion of Critical Activities

Table 2.01 - Assumptions for Discrete-Time Single Work Center Scheduling Problems

2.1. Mathematical Programming Models

Three mixed-integer linear multi-objective mathematical programming models are proposed for solving discrete-time single work center scheduling problems in LVLVPS. The first model, presented in Section 2.1.1 employs a pre-emptive goal programming approach in the optimization of a dual-objective problem, aimed at minimizing the number of resources required in the completion of the maximum number of activities. The pre-emptive goal programming approach is a step-wise methodology found effective in solving multi-objective priority-based optimization problems, in which the total deviation of the objective functions to their aspiration criteria or goals is minimized [100,101,102,103]. This model is formulated specifically for scenarios permitting travel work, where the strict enforcement of time and resource constraints may result in traveling of incomplete activities. In Section 2.1.2, an alternative mathematical programming model is proposed, where the pre-emptive goal programming approach is adopted in the modeling of a multi-objective function with four levels of priority, aimed at minimizing the positive deviation of makespan and resource requirement to their corresponding aspiration criteria while minimizing their objective value. The optimization model formulated in Section 2.1.2 is proposed for scenarios where travel work is prohibited, and deviation from the aspiration criteria for time and resources may be required in the completion of the statement of work in its entirety. Furthermore, the mathematical programming model, proposed in Section 2.1.3, can be applied to solve work center capacity problems, where through increased resource saturation, the makespan for completion of the assigned statement of work is minimized, demonstrating a work center's maximum capacity. In the formulation of this dual-objective priority-based function, the lexicographic method is adopted, where the objective functions are arranged based on their relative importance and are solved iteratively following a lexicographical order [104,105,106]. Table 2.02 provides an overview of the proposed models, model-specific assumptions, adopted methodologies, and objectives, based on their order of priority.

Sec.	*Assumptions & Constraints*	*Approach*	*Objective*
2.1.1	▪ Travel work permitted ▪ Hard time & resource constraints	Pre-emptive Goal Programming	1. Minimize number of incomplete activities 2. Minimize resource requirements
2.1.2	▪ Travel work prohibited ▪ Completion of all activities ▪ Allowable deviation to time and resource constraints	Pre-emptive Goal Programming	1. Minimize positive deviation from takt-time 2. Minimize positive deviation from budgets 3. Minimize resource requirements 4. Minimize makespan
2.1.3	▪ Completion of all activities ▪ No resource constraints	Lexicographic Method	1. Minimize makespan 2. Minimize resource requirements

Table 2.02 - Proposed Mathematical Programming Models for Discrete-Time Single Work Center Scheduling Problems

In mathematical programming of the proposed models, a set of activities $j \in \{1, \dots, N\}$ are defined to be completed by a budgeted number of resources B, over the span of a pre-defined takt-time T. Rational variable J_{min} represents the minimum percentage of activities that must be completed in-station, enabling the traveling of non-critical activities $j \in \mu$ to downstream work centers, where applicable. The key decision variable in the formulation of the proposed mathematical models is the binary variable x_{jt}, representing the completion time of activity j, where $x_{jt} = 1$ if activity j is completed at time t, and $x_{jt} = 0$ if otherwise. Activities can only be scheduled once, and are classified as either critical $j \in \lambda$ or non-critical $j \in \mu$, and may retain a single $j \in \alpha$ or multiple modes $j \in \beta$. Alternative modes of multi-mode activities are introduced as new dummy activities, with an identical set of attributes including interdependencies $P_{jj'}$, zonal assignments y_{ji}, and non-concurrencies, but are differentiated by their unique processing times p_j that is negatively proportional to the assigned number of resources. Binary parameter $M_{jj'}$ establishes the association between alternative modes of the multi-mode activity, where $M_{jj'} = 1$ if activity j' is an alternative mode of activity j, and $M_{jj'} = 0$ if otherwise. Interdependencies are similarly represented through the binary parameter $P_{jj'}$, where $P_{jj'} = 1$ if activity j is a predecessor to activity j', and $P_{jj'} = 0$ if otherwise. Lead and lag times between interdependent activities are represented by the integer parameter $L_{jj'}$, where lag times are imposed if $L_{jj'} \geq 0$, lead times are permitted if $L_{jj'} \leq 0$, and $L_{jj'} = 0$ if no interdependencies or lead or lag times are captured between activities j and j'. Moreover, non-concurrency constraints may exist in a variety of forms, between two activities, two zones, or between activities and zones, and are denoted by binary parameters $C_{jj'}$, $C_{ii'}$, and C_{ji} respectively, where the parameters assume a value of 1, if a non-concurrency constraint exists, and is equal to zero otherwise. Furthermore, activities may require multiple resources w_{jl} of distinct classifications $l \in \{1, \dots, L\}$, and are assigned to a zone $i \in \{1, \dots, I\}$ through binary parameter y_{ji}, where $y_{ji} = 1$ if activity j is assigned to zone i, and $y_{ji} = 0$ if otherwise, where each zone i is subject to a maximum allowable capacity Z_i. In addition to an overall work center resource budget B, available resources from each classification W_l^{max} cannot be exceeded in solving for the decision variable W_l, representing the overall number of resources required from each resource pool l. Table 2.03 provides an overview and description of variables, parameters, and sets used in the mathematical programming of the proposed optimization models.

Model Component	***Notation***		***Description***
Sets	J	$j \in \{1, \dots, N\}$	Activity Number
	J', J''	$j', j'' \equiv j$	Equivalent of Activity j
	L	$l \in \{1, \dots, L\}$	Resource Classification
	I	$i \in \{1, \dots, I\}$	Zone Classification
	I'	$i' \equiv i$	Equivalent of Zone i
	T	$t \in [1, T]$	Discrete-Time Planning-Horizon
	U	$u = [t-1, t]$	Single Time Interval
	Α	$\alpha \in j \setminus \beta$	Single-Mode Activities
	Β	$\beta \in j \setminus \alpha$	Multi-Mode Activities
	Γ	$\lambda \in j \setminus \mu$	Critical Activities
	Μ	$\mu \in j \setminus \lambda$	Non-Critical Activities
Parameters	$M_{jj'}$	$M_{jj'} \in \{0,1\}$	Multi-Mode Matrix of Activities j and j' $M_{jj'} = 1$ if j' is an alternative mode of j, 0 otherwise
	p_j	$p_j \geq 0$	Duration of Activity j
	$P_{jj'}$	$P_{jj'} \in \{0,1\}$	Precedence Matrix of Activities j and j' $P_{jj'} = 1$ if j is a predecessor to j', 0 otherwise
	$C_{jj'}$	$C_{jj'} \in \{0,1\}$	Non-Concurrency Matrix of Activities j and j' $C_{jj'} = 1$ if j and j' are non-concurrent, 0 otherwise
	$C_{ii'}$	$C_{ii'} \in \{0,1\}$	Non-Concurrency Matrix of Zones i and i' $C_{ii'} = 1$ if i and i' are non-concurrent, 0 otherwise

Model Component	*Notation*		*Description*
	C_{ji}	$C_{ji} \in \{0,1\}$	Non-Concurrency Matrix of Activity j with Zones i $C_{ji} = 1$ if j and i are non-concurrent, 0 otherwise
	$L_{jj'}$	$L_{jj'} \in [-T, +T]$	Lead/Lag Time Between Activities j and j' $L_{jj'} = 0$ if no Precedence or Lead or Lag Times $L_{jj'} > 0$ if j' is Lagged After Completion of j $L_{jj'} < 0$ if j' has a Lead Prior to Completion of j
	y_{ji}	$y_{ji} \in \{0,1\}$	Designation of Zone i to Activity j $y_{ji} = 1$ if j is in zone i, 0 otherwise
	Z_i	$Z_i > 0$	Maximum Capacity for Zone i
	B	$B > 0$	Resource Budget
	W_l^{max}	$W_l^{max} \geq 0$	Available Resource from Pool l
	J_{min}	$J_{min} \in [0,1]$	Minimum Percentage of Completed Activities
	ES_j	$ES_j \in [1, T]$	Earliest Start Time of Activity j
	w_{jl}	$w_{jl} \geq 0$	Required Resources from Pool l for Activity j
Variables	x_{jt}	$x_{jt} \in \{0,1\}$	Completion of Activity j at Time t $x_{jt} = 1$ if j is completed at t, 0 otherwise
	π_j	$\pi_j \in \{0,1\}$	Measures Scheduling of Activity j $\pi_j = 0$ if j is scheduled, 1 otherwise
	W_l	$W_l \geq 0$	Total Required Resources from Pool l

Table 2.03 - Sets, Parameters, and Variables for Discrete-Time Single Work Center Scheduling Problems

2.1.1. Work Center Scheduling Problem – Travel Work Permitted

The proposed multi-objective mixed-integer optimization model employs a pre-emptive goal programming approach in the formulation of a priority-based objective function, subject to scarce time and resources. To tackle the potential infeasibility in the completion of all activities on-time, and on-budget, travel work has been permitted, allowing the traveling of non-critical activities. The proposed priority-based objective function has two levels of priority, the higher level priority objective $P1[\delta_1^-]$, is aimed at maximizing the number of completed activities, through minimizing the negative deviation of scheduled activities $\sum_j \sum_t x_{jt}$ from their aspiration criterion $\sum_j \sum_t x_{jt} = N$. The lower level priority objective $P2[\delta_2^+]$ however, minimizes the required number of resources, through minimizing the positive deviation between resource requirements $\sum_l W_l$ and its corresponding aspiration criterion $\sum_l W_l = 0$. The resultant schedule obtained from solving this objective function yields the minimum number of resources required in the completion of the maximum number of activities where the latter takes precedence. Note that the two objectives $P1[\delta_1^-]$ and $P2[\delta_2^+]$ are conflicting, where achieving the optimal value of one objective requires a compromise to the subsequent objective. The required number of resources is found to be positively correlated to the number of completed activities, where an increase in the number of resources results in an increase in the number of completed activities. It can thus be concluded that in achieving the optimal objective value for the higher priority objective, the lower priority level objective is compromised through positive deviation to its aspiration criterion. The proposed mathematical programming model is as follows.

Objective Function:

$$Min\,Z \;\; = P1[\delta_1^-] + P2[\delta_2^+] \tag{2.01}$$

Subject To:

$$\sum_j \sum_t x_{jt} + \delta_1^- = N \tag{2.02}$$

$$\frac{1}{N} \sum_j \sum_t x_{jt} \geq J_{min} \tag{2.03}$$

$$\sum_t x_{jt} = 1 \qquad \forall \;\; j \in \alpha \cap \lambda \tag{2.04}$$

$$\sum_t x_{jt} + \sum_t x_{j't} = 1 \qquad \forall \;\; j, j' \in \{\, \beta \cap \lambda \;\; : \; M_{jj'} = 1 \,\} \tag{2.05}$$

$$\sum_t x_{jt} \leq 1 \qquad \forall \;\; j \in \alpha \cap \mu \tag{2.06}$$

$$\sum_t x_{jt} + \sum_t x_{j't} \leq 1 \qquad \forall \;\; j, j' \in \{\, \beta \cap \mu \;\; : \; M_{jj'} = 1 \,\} \tag{2.07}$$

$$\sum_l W_l - \delta_2^+ = 0 \tag{2.08}$$

$$\sum_j w_{jl} \sum_{u=t}^{t+p_j-1} x_{ju} \leq W_l \qquad \begin{array}{l} \forall \;\; l = 1, \ldots, L \\ \forall \;\; t = 1, \ldots, T \end{array} \tag{2.09}$$

$$W_l \leq W_l^{max} \qquad \forall \;\; l = 1, \ldots, L \tag{2.10}$$

$$\sum_t x_{jt} + \pi_j = 1 \qquad \forall \;\; j = 1, \ldots, N \tag{2.11}$$

$$P_{jj'}\left[\sum_t t x_{jt} + L_{jj'}\right] \le \sum_t (t - p_{j'}) x_{j't} + M\pi_j \qquad \forall \;\; j,j' \in \{\, \alpha \cup \beta \;\; : \; P_{jj'} = 1 \,\} \tag{2.12}$$

$$\sum_t x_{jt} + \sum_t x_{j't} \ge \sum_t x_{j'',t} \qquad \forall \quad j,j' \in \beta,\; j'' = \alpha \cup \beta \;\; : \; P_{jj''} = 1,\; M_{jj'} = 1 \tag{2.13}$$

$$\sum_t x_{jt} \ge \sum_t x_{j't} \qquad \forall \quad j,j' \in \{\, \alpha \quad : \; P_{jj'} = 1 \,\} \tag{2.14}$$

$$C_{jj'}\left[\sum_{u=t}^{t+p_j-1} x_{ju} + \sum_{u=t}^{t+p_{j'}-1} x_{j'u}\right] \le \; 1 \qquad \begin{array}{l} \forall \;\; t = 1, \dots, T \\ \forall \;\; j,j' \in \{\alpha \quad : \; C_{jj'} = 1\,\} \end{array} \tag{2.15}$$

$$C_{ii'}\left[\sum_j y_{ji} \sum_{u=t}^{t+p_j-1} x_{ju} + \sum_j y_{ji'} \sum_{u=t}^{t+p_j-1} x_{ju}\right] \le \; 1 \qquad \begin{array}{l} \forall \;\; t = 1, \dots, T \\ \forall \;\; i,i' \quad : \; C_{ii'} = 1 \end{array} \tag{2.16}$$

$$C_{ji}\left[\sum_{u=t}^{t+p_j-1} x_{ju} + \sum_j y_{ji} \sum_{u=t}^{t+p_j-1} x_{ju}\right] \le \; 1 \qquad \begin{array}{l} \forall \;\; t = 1, \dots, T \\ \forall \quad j,i \; : \; C_{ji} = 1 \end{array} \tag{2.17}$$

$$\sum_l \sum_j w_{jl}\, y_{ji} \sum_{u=t}^{t+p_j-1} x_{ju} \le Z_i \qquad \begin{array}{l} \forall \quad i = 1, \dots, I \\ \forall \quad t = 1, \dots, T \end{array} \tag{2.18}$$

$$\sum_t x_{jt}(t - p_j) \ge \; ES_j \sum_t x_{jt} \qquad \forall\, j = 1, \dots, N \tag{2.19}$$

Constraint (2.02) is set forth to establish the value of δ_1^-, equivalent to the negative deviation between the number of completed activities $\sum_j \sum_t x_{jt}$, and the number of activities assigned to the work center N. Through Constraint (2.03), the minimum number of activities that must be completed in-station is enforced, $(N \times J_{min})$. Constraint set (2.04) ensures the scheduling of all activities classified as single-mode and critical $j \in \{\alpha \cap \lambda\}$, where activities can only be scheduled once. Similarly, Constraint set (2.05) enforces the scheduling of all critical and multi-mode activities $j \in \beta \cap \lambda$, where only a single mode of a multi-mode activity can be scheduled. Constraint sets (2.06) and (2.07) resemble those of (2.04) and (2.05) and are formulated for non-critical single-mode and multi-mode activities, respectively. Similar to Constraint (2.02), Constraint (2.08) is enforced to establish an association between the required number of resources, and the aspiration criterion for resources, through a positive deviation variable δ_2^+. Constraint set (2.09) states that the total number of resources from pool l occupied at time interval $u = [t-1, t]$ must not exceed the decision variable W_l, representing the total number of resources required from pool l, and is aimed at establishing the value of this decision variable. Constraint set (2.10) is applied to enforce resource constraints, where an upper bound on resource availability per classification has been imposed. The value of π_j is established through Constraint set (2.11), further to be used in Constraint set (2.12) to ensure flexibility in the modeling of multi-mode activities. Constraint set (2.12) ensures that precedence constraints and the corresponding lead and lag times $L_{jj'}$ are successfully satisfied, where the start time of a successor activity $\sum_t (t - p_{j'}) x_{j't}$, must exceed the completion time of its predecessor(s) $\sum_t t x_{jt}$. The Big-M Method has been adopted in the formulation of this constraint set, in conjunction with the decision variable π_j to allow for scheduling of activities with multi-mode predecessors. Constraint sets (2.13) and (2.14) are precedence constraints for multi-mode and single-mode activities respectively, enforcing the completion of all predecessor activities prior to the start time of their successor(s). Constraint sets (2.15), (2.16) and (2.17) impose non-concurrencies between two activities, two zones, and between an activity and a zone respectively, restricting their simultaneous progression at any time interval $u = [t-1, t]$. Zonal constraints are enforced through Constraint set (2.18), through an imposed upper bound on density for each zone i, representing the maximum allowable capacity for that zone Z_i, at any time interval $u = [t-1, t]$. Lastly, Constraint set (2.19) incorporates earliest start times of activities and is most effective in scenarios where the start time of an activity is paced by factors external to the system, and is to be relaxed if such factors do not exist. This model is validated and verified in Section 2.2 through a case study and is proved to result in the optimum schedule for scenarios permitting travel work. Moreover, this model yields to the scheduling of all activities with the minimum number of resources in scenarios where there exists a feasible solution for the completion of the entire statement of work on-time and on-budget.

2.1.2. Work Center Scheduling Problem – Travel Work Prohibited

An alternative mathematical programming model is proposed, tailored for scenarios prohibiting travel work through mandating the completion of all activities in-station. The optimum solution in such scenarios is thus defined as the schedule that minimizes the number of resources required to complete all activities with minimum positive deviation from the takt-time. To address the potential infeasibility of this problem, positive deviation to time and resource constraints have been permitted. In the modeling of this problem, a new activity $j = N + 1$ is introduced, representing the finishing node of the activity on node (AON) network diagram. This dummy activity is a successor to all activities without successors, with a processing time of zero $p_{N+1} = 0$. Figure 2.03 is a sample AON network diagram with $N = 15$ activities, demonstrating a statement of work with three finishing nodes $j = \{13, 14, 15\}$. To ensure the structural integrity of the proposed optimization model, the highlighted dummy activity $j = N + 1 = 16$ is added, with the purpose of succeeding all activities without successors $j = \{13, 14, 15\}$, the completion time of which represents the makespan for completion of all activities assigned to a work center.

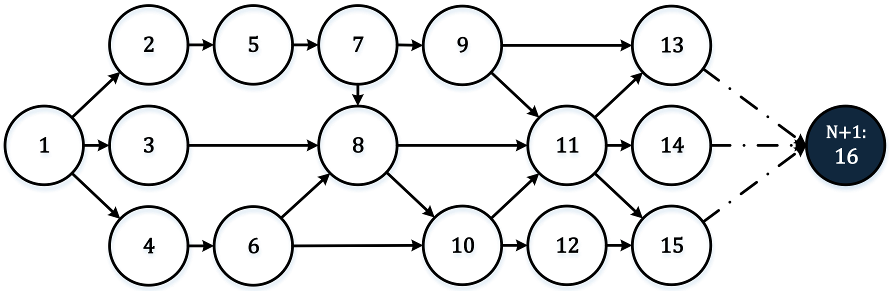

Figure 2.03 – Sample Activity-On-Node Network Diagram

In formulating this multi-objective function, a pre-emptive goal programming approach is employed. This priority-based approach uses a set of deviation variables δ_k^- and δ_k^+, to allow for flexibility in the utilization of additional time and resources in completion of all activities. This objective function is aimed at minimizing the positive deviation of makespan to the takt-time, and the positive deviation of the required number of resources from the budget, while minimizing resource requirements and makespan. Note that the two latter objectives ensure that the model yields to an optimum schedule even in scenarios where the completion of the statement of work does not require a positive deviation to time and/or resource constraints, thus improving model applicability and flexibility. Objective function (2.20) is a priority-based function, with four levels of priority, where $P1, P2, P3$, and $P4$ represent the priority levels of each of the objectives. $P1[\delta_3^+]$ represents the positive deviation of makespan from the takt-time, $P2\left[\sum_l \delta_{4,l}^+\right]$ represents the overutilization of resources beyond the budget, $P3[\sum_l W_l]$ is aimed at minimizing the required number of resources, and $P4\left[\sum_t x_{N+1,t}\right]$ represents the completion time of the dummy finishing node, equivalent to the overall makespan for completion of the statement of work in its entirety.

$$Min\,Z \;= P1[\delta_3^+] + P2\left[\sum_l \delta_{4,l}^+\right] + P3\left[\sum_l W_l\right] + P4\left[\sum_t x_{N+1,t}\right] \tag{2.20}$$

Objective function (2.20) can also be represented through objective function (2.21), relying solely on deviation variables, where $+P3[\sum_l W_l]$ is replaced with $-P3\left[\sum_l \delta_{4,l}^-\right]$, representing the negative deviation from the imposed resource budgets. Similarly, the fourth level priority objective $+P4\left[\sum_t x_{N+1,t}\right]$ can be replaced with $-P4[\delta_3^-]$, representing the negative deviation of makespan from the takt-time.

$$Min\,Z \;= P1[\delta_3^+] + P2\left[\sum_l \delta_{4,l}^+\right] - P3\left[\sum_l \delta_{4,l}^-\right] - P4[\delta_3^-] \tag{2.21}$$

Subject to constraint sets (2.08), (2.09), (2.11), (2.12), and (2.15) – (2.19) and the following four constraint sets.

$$\sum_t x_{N+1,t} + \delta_3^- - \delta_3^+ = T \tag{2.22}$$

$$W_l + \delta_{4,l}^- - \delta_{4,l}^+ = \; W_l^{max} \qquad \forall \quad l = 1, \dots, L \tag{2.23}$$

$$\sum_t x_{jt} = 1 \qquad \forall \quad j \in \alpha \tag{2.24}$$

$$M_{jj'}\left[\sum_t x_{jt} + \sum_t x_{j't}\right] = 1 \qquad \forall \quad j, j' \in \{\beta \;\; : \; M_{jj'} = 1\} \tag{2.25}$$

In addition to the previously formulated constraints, Constraint (2.22) is introduced, ensuring that the completion time of activity $j = N + 1$, plus and minus the negative and positive deviation variables (δ_3^-, δ_3^+), is equal to its aspiration criterion, equivalent to the takt-time T. Constraint set (2.23) is similarly imposed to establish the relationship between the required number of resources W_l and its corresponding aspiration criterion W_l^{max}, through deviation variables $\left(\delta_{4,l}^-, \delta_{4,l}^+\right)$. Constraint sets (2.24) and (2.25) replace Constraint sets (2.04) – (2.07) presented in Section 2.1.1, to ensure the scheduling and completion of all activities in-station. This model is validated and verified in Section 2.2 and is found to be an effective scheduling optimization approach in scenarios where travel work is prohibited.

2.1.3. Capacity Study

Another characteristic of LVLVPS is the rapid ramp-ups and cool downs of production rates, where the takt-times are suddenly dropped or increased in satisfying customer demands. The mathematical programming model proposed in this section is an effective approach for capacity analysis and early detection of bottlenecks for desired takt-times, through the complete saturation of resources. In the context of sequencing and scheduling, resource saturation refers to the maximum number of resources that can be utilized in a work center. The objective of this model is to solve for the minimum number of resources required in achieving the minimum makespan for completion of the assigned statement of work. The lexicographic method is adopted in the formulation of this multi-objective priority-based function (2.26), where the objectives are arranged and solved iteratively following a lexicographical order. The higher level priority objective $P1\left[\sum_t tx_{N+1,t}\right]$ is aimed at minimizing the makespan for completion of the pre-defined statement of work, equivalent to the completion time of the dummy finishing node $j = N + 1$. The second level priority objective $P2[\sum_l W_l]$ minimizes resource requirements in the completion of all activities over the span of the prescribed makespan $\sum_t tx_{N+1,t}$.

$$Min \;\; Z = \quad P1\left[\sum_t tx_{N+1,t}\right] + P2\left[\sum_l W_l\right] \tag{2.26}$$

In the modeling of this problem, a new dummy activity $j = N + 1$ is introduced, similar to that of Section 2.1.2, with a processing time of $p_{N+1} = 0$, succeeding all activities without successors. This is in addition to minor adjustments to the previously formulated models, where the upper bound on resources has been relaxed, and the planning horizon has been extended to the sum of all processing times $T = \sum_j p_j$, to ensure feasibility. This problem is subject to constraints sets (2.09), (2.10), (2.11), (2.12), and (2.15) – (2.19) of Section 2.1.1, and constraint sets (2.24) and (2.25) of Section 2.1.2, in assuring the completion of all activities, while satisfying all zonal, precedence, non-concurrencies, and earliest start constraints. The results obtained from solving this model will provide strategic insight into a work center's capacity. The resultant makespan $\sum_t tx_{N+1,t}$ is an indication of a work center's capability in satisfying foreseen takt-times and is used in evaluating the need for alternative shift patterns or parallel work centers in an attempt to fulfill projected demand rates. Moreover, the required number of resources $\sum_l W_l$ obtained from solving this problem is the maximum number of value-added resources that can be assigned to the work center, yielding to the complete saturation of resources. It can be demonstrated that additional resources beyond the obtained saturation level will not improve makespan or add value to the build. This model, in addition to the models proposed in Section 2.1.1 and 2.1.2 is validated and verified in the following section through a real-world case study.

2.2. Case Study

The mathematical programming models proposed in Section 2.1 are validated and verified, through a real-world case study with a global leader in the aerospace industry. The production system under this study is the final assembly line of a light-body single-aisle green aircraft. Assemblies follow a pre-defined processing order through a series of unique work centers, each budgeted with multiple classifications of resources (structural mechanics, avionics specialists, aero-structure assemblers, aerodynamic sealers, etc.). The final assembly lines of aircraft are classified as LVLVPS, and thus can be optimized through the proposed mathematical programming models. The objective of this case study is to evaluate the proposed models' capability in the modeling of characteristics, constraints, and objectives inherent in such production systems and to solve for the optimum schedule. For this purpose, a work center on the final assembly line has been selected, and the mathematical programming models proposed in Sections 2.1.1, 2.1.2 and 2.1.3 are applied and evaluated in Sections 2.2.1, 2.2.2, and 2.2.3, respectively. The computational experiments performed in this section were executed on 64-bit Windows with an Intel 6th generation i7 processor, running at 2.6 GHz with a 16.0 GB RAM.

The structural assembly of the wings to the center fuselage is comprised of $N = 147$ activities, each attributed with a discrete processing time p_j, ranging from 1 to 4 hours. Activities are highly interdependent, with 474 unique interdependencies as illustrated in the AON network diagram depicted in Figure 2.04, to be completed in $T = 124$ hours, equivalent to a two-shift operation, over 8 consecutive days, where each shift accounts for 7.75 effective production hours. It can be demonstrated through Figure 2.04, that there exists a single finishing node $j = 73$ in the statement of work assigned to this work center, the completion time of which represents the makespan for completion of all activities, eliminating the need for the introduction of a new dummy finishing node as prescribed in Section 2.1.2. This work center is budgeted with $B = 9$ resources of $L = 2$ classifications, $W_1^{max} = 8$ of which are classified as type 1, and the remaining as type 2 resources $W_2^{max} = 1$. While the proposed mathematical programming models are capable of modeling production systems with activities that require multiple classifications of resources, activities assigned to this particular work center are completed by either type 1 or type 2 resources. There exists a total of $I = 6$ active zones in this structural assembly, representing the physical location of work, where each zone is subject to a maximum capacity ranging between 2 to 4 resources.

2.2.1. Travel Work Permitted

In this section, the mathematical programming model proposed in Section 2.1.1 is applied to the work center under study. The proposition being tested is the proposed model's capability in solving for the optimum solution in scenarios permitting travel work. This problem allows the traveling of up to 10% of the assigned statement of work, thus mandating the completion of a minimum of $J_{min} = 90\%$ of activities. The mathematical programming model proposed in Section 2.1.1 was programmed into IBM ILOG CPLEX, where the problem was solved to optimality in two phases after 5,989,022 iterations in 568.33 seconds, yielding an the objective value of $\delta_1^- + \delta_2^+ = 6$. This scheduling problem is subject to 44,234 constraints, with 61,448 decision variables, and 2,883,872 non-zero coefficients. Following the pre-emptive goal programming approach, the problem was initially solved for the higher level priority objective $P1[\delta_1^-] = N - \sum_j \sum_t x_{jt}$, representing the number of incomplete activities, resulting in objective value of $\delta_1^- = 0$, signifying the existence of a feasible schedule for completion of all activities $\left(\sum_j \sum_t x_{jt} = N\right)$ on-time and on-budget. In the second phase, the problem was solved for the lower level priority objective $P2[\delta_2^+] = \sum_l W_l$, representing the required number of resources, yielding to an objective value of $\delta_2^+ = 6$, indicating the need for a minimum of $W_1 = 5$ units of type 1, and $W_2 = 1$ unit of type 2 resources, in the completion of all activities over the span of the takt-time. Figure 2.05 is the resultant Gantt chart, demonstrating the optimum schedule for the work center under this study.

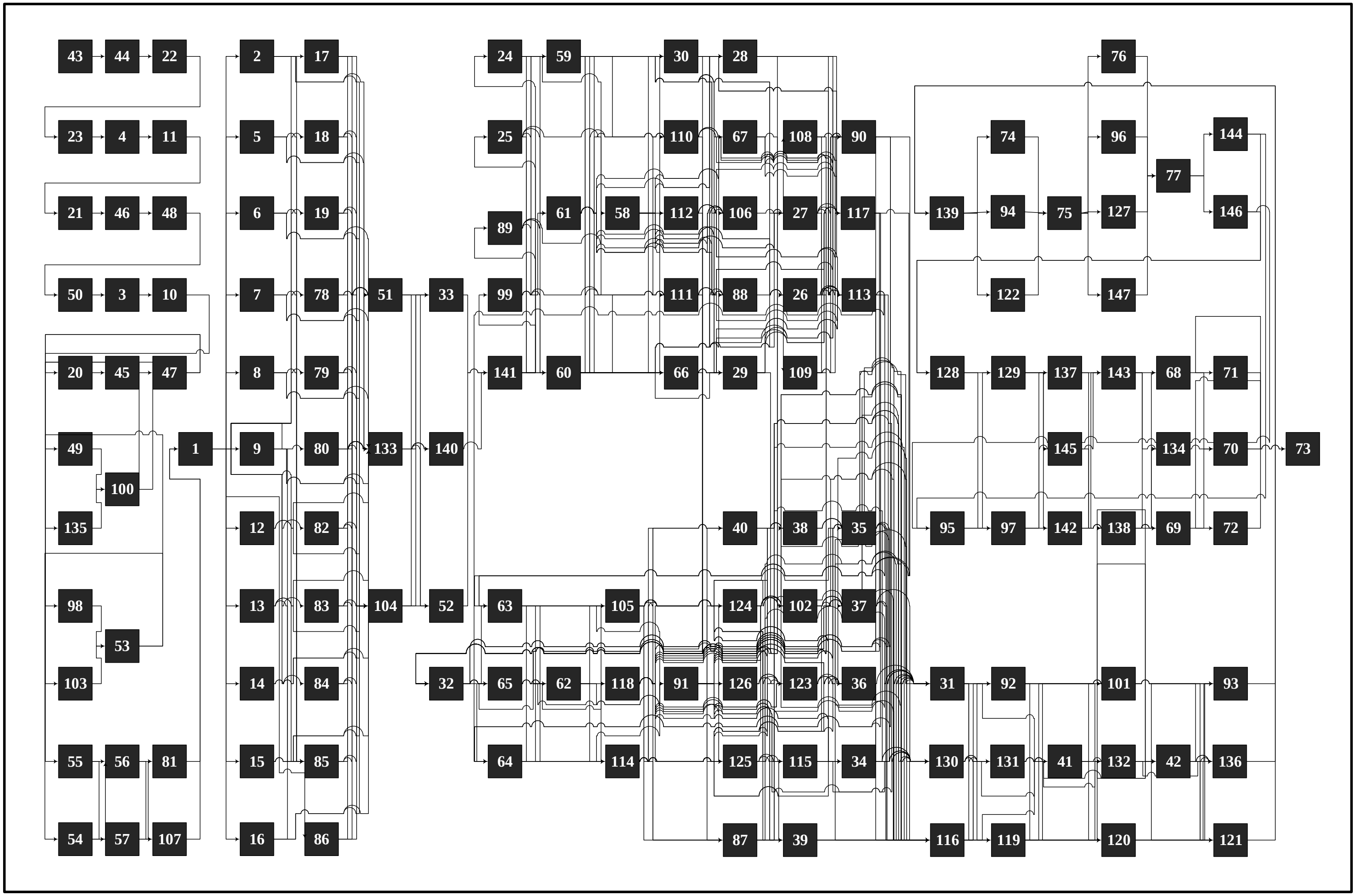

Figure 2.04 - Case Study 1 - Activity-On-Node Network Diagram

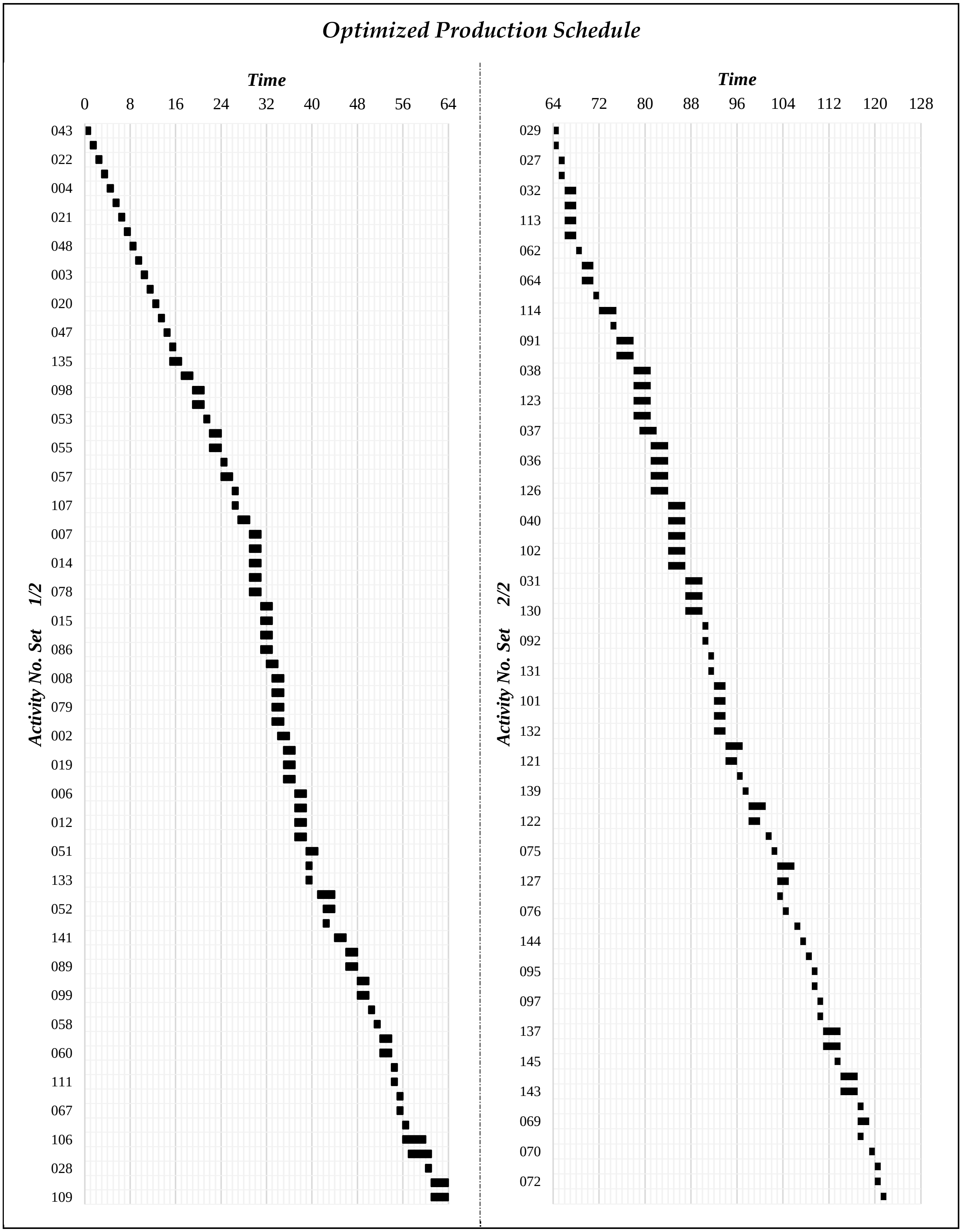

Figure 2.05 - Gantt Chart for the Optimized Work Center Schedule

Figure 2.06 illustrates the free float between interdependent activities, representing the amount of time between the completion of a predecessor and the start time of its successor. It is demonstrated through this figure that a positive free float is obtained for each of the interdependencies, indicating the successful fulfillment of all precedence constraints, where successor activities start upon or after the completion of their predecessor(s). Further analysis of free float values suggests that on average, successor activities start 3.06 hours after the completion of their predecessors. This extensive lag time is primarily due to the secondary objective, aimed at minimizing the required number of resources, resulting in an increased free float between interdependent activities. Note that the objective of this model is not to minimize makespan but rather to minimize the number of resources required in the completion of the maximum number of activities over the span of the takt-time. Figure 2.07 illustrates the number of active activities at each time interval, yielding an average of 2.18 concurrent activities. Figure 2.08 and 2.09 highlight the number of type 1 and type 2 resources utilized at each time interval $[t-1, t]$, demonstrating the need for $W_1 = 5$ units of type 1, and $W_2 = 1$ unit of type 2 resources. Note that while this work center is budgeted with $W_1^{max} = 8$ and $W_2^{max} = 1$ units of type 1 and 2 resources, the optimized schedule requires only 66.67% of the budget, resulting in labor savings of 33.34%, equivalent to 3 full-time resources. Figures 2.10 through 2.15 illustrate the density and capacity of each zone, demonstrating the successful fulfillment of all zonal constraints, where no zone is oversaturated at any point in time. Oversaturation refers to the assignment of resources to a particular zone, surpassing its capacity. It can thus be concluded through this study, that the resultant schedule satisfies all applicable constraints in the completion of the pre-defined statement of work over the span of the takt-time while utilizing only 66.67% of the budgeted number of resources. The proposed mathematical programming model has thus proven to be an effective and efficient method for solving large-scale scheduling problems in LVLVPS. Note, that while the problem under this study was solved to optimality through the scheduling of all activities on-time and on-budget, the infeasibility in the completion of all activities would result in a solution that maximizes the number of scheduled activities while satisfying the imposed time and resource constraints. However, there exist scenarios and scheduling strategies that prohibit the traveling of incomplete activities, where the work center is ultimately responsible to complete the statement of work in its entirety. Such problems would require the use of the mathematical programming model proposed in Section 2.1.2 with flexibility in aspiration criteria to constraints.

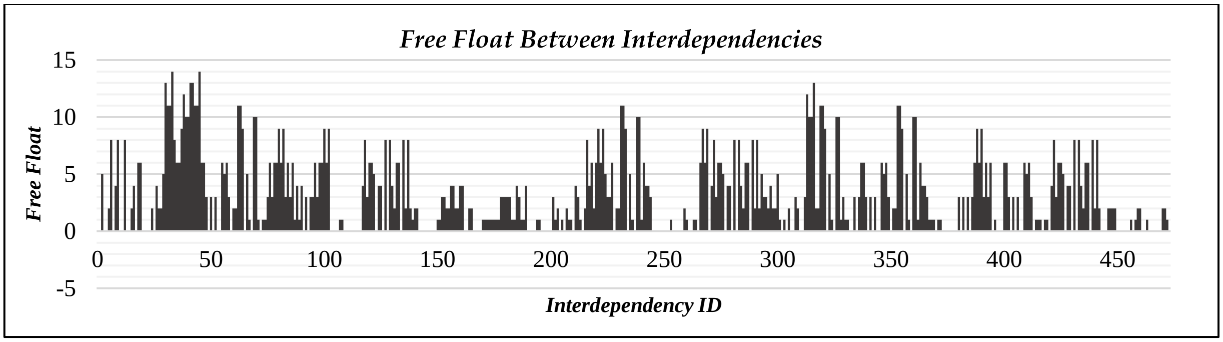

Figure 2.06 - Free Float Between Interdependent Activities

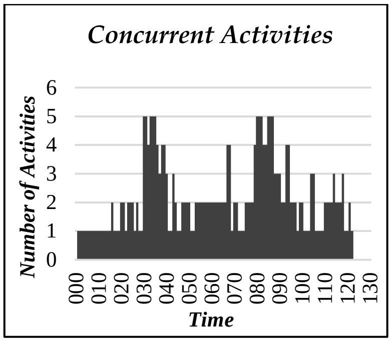

Figure 2.07 - Concurrent Activities

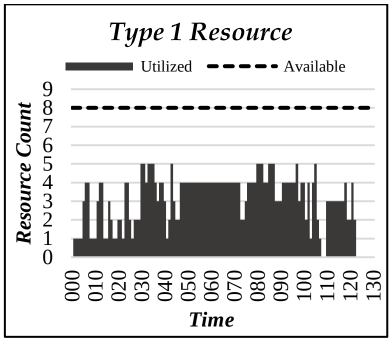

Figure 2.08 - Resource 1 Utilization

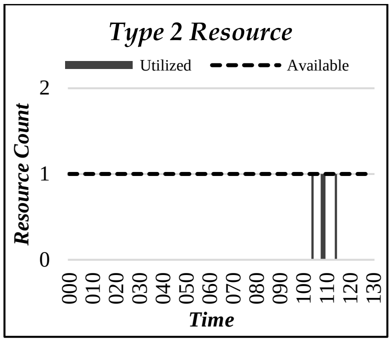

Figure 2.09 - Resource 2 Utilization

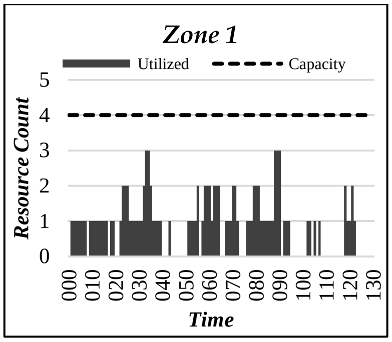

Figure 2.10 - Zone 1 Utilization

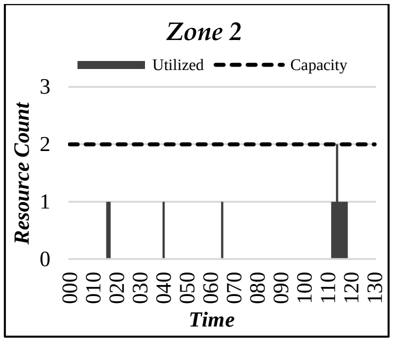

Figure 2.11 - Zone 2 Utilization

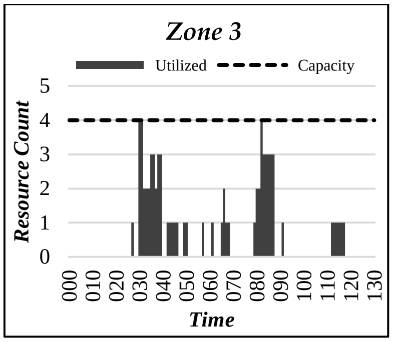

Figure 2.12 - Zone 3 Utilization

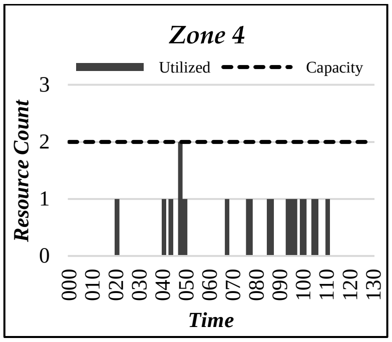

Figure 2.13 - Zone 4 Utilization

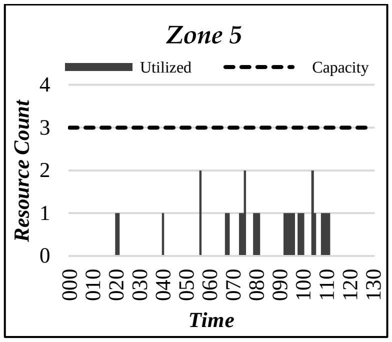

Figure 2.14 - Zone 5 Utilization

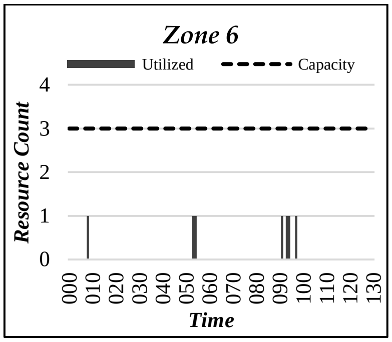

Figure 2.15 - Zone 6 Utilization

2.2.2. *Travel Work Prohibited*

While the mathematical programming model proposed in Section 2.1.1 has yielded to the optimum schedule for the completion of all activities, in an effort to verify and validate the model proposed in Section 2.1.2, some of the input parameters disclosed in Section 2.2 are slightly modified. This is to ensure infeasibility in the completion of all activities on-time and on-budget, accomplished by reducing the number of budgeted resources from $B = 9$ to $B = 6$, where type 1 and 2 resource budgets are set to $W_1^{max} = 5$ and $W_2^{max} = 1$ resources per shift, respectively. Furthermore, three identical activities are added to the statement of work $j \in \{1, \dots, 147\} \cup \{148, 149, 150\}$, each of which is a successor to the original finishing activity $j = 73$ (Refer to Figure 2.04). Each of the new activities has a processing time of $p_j = 10$ hours, and require $w_{j1} = 2$ units of type 1 resources, and $w_{j1} = 0$ units of type 2 resources. Moreover, a dummy finishing node $j = N + 1$ is introduced in addition to a new set of deviation variables $\delta_3^{\pm},\ \delta_{4,l}^{\pm}$ to the aspiration criteria for time and resources, equivalent to $\sum_t x_{151,t} + \delta_3^- - \delta_3^+ = 124$ hours, with a budget comprised of $W_1 + \delta_{4,1}^- - \delta_{4,1}^+ =$ 5 units of type 1, and $W_2 + \delta_{4,2}^- - \delta_{4,2}^+ =$ 1 unit of type 2 resources.

With four levels of priority as prescribed in objective function (2.20) in Section 2.1.2, the proposed pre-emptive goal programming model is solved in four phases. The problem is initially solved for the first level priority objective $P1[\delta_3^+]$, representing the positive deviation of makespan from its aspiration criterion $T = 124$, yielding to an objective value of $\delta_3^+ = 0$, indicating the feasibility in the completion of all activities with a makespan that is less than or is equal to $T = 124$ hours. Following this pre-emptive approach, the obtained objective value $\delta_3^+ = 0$ is added as a new constraint prior to solving for lower priority objectives. The model is then solved for the second level priority objective $\delta_{4,1}^+ + \delta_{4,2}^+$, representing the positive deviation of resource requirements per classification from their corresponding aspiration criteria. The resultants obtained from solving this phase of the problem is comprised of $\delta_{4,1}^+ = 1$ and $\delta_{3,2}^+ = 0$, yielding to an objective value of $\delta_{4,1}^+ + \delta_{4,2}^+ = 1$. It can thus be concluded that there exists a feasible schedule for completion of all activities on-time, if and only if an additional unit of type 1 resource $\delta_{4,1}^+ = 1$ is added to the work center, beyond the imposed budgets $\sum_l W_l^{max}$. Similar to the previous phase, $\delta_{4,1}^+ = 1$ and $\delta_{3,2}^+ = 0$ are added as new constraints prior to solving for the third level priority objective $P3\left[\sum_l \delta_{4,l}^-\right]$, aimed at maximizing the negative deviation of resource requirements from the budget $\sum_l W_l^{max}$. This objective is most effective in scenarios where positive deviation from resource requirement to their aspiration criteria is not encountered i.e. $\delta_{4,1}^{+} + \delta_{4,2}^{+} = 0$. The resultant objective value from solving the third level priority objective is $\delta_{4,1}^{-} + \delta_{4,2}^{-} = 0$, suggesting that a negative deviation to the resource budgets does not allow for timely completion of all activities. The objective value obtained in this phase is then added as a new constraint, prior to solving for the fourth level priority objective $P4[\delta_{3}^{-}]$, aimed at minimizing makespan through maximizing the negative deviation between makespan and its aspiration criterion. The objective function was solved to optimality, yielding to an objective value of $\delta_{3}^{-} = 3$, suggesting that the minimum makespan for completion of all activities with $\sum_{l} W_{l} = 7$ resources are $124 - 3 = 121$ hours. The problem was solved iteratively in CPLEX, where an optimum solution was reached after 4,243,142 iterations in 605.58 seconds. Figure 2.16 is the Gantt chart of the resultant schedule, demonstrating activity start and finish times. Similar to Section 2.2.1, constraints, their corresponding aspiration criteria and utilization are analyzed in Figures 2.17 through 2.26. It is established through Figure 2.17 that all the interdependencies exhibit positive free float values, with an average of 2.27 hours, a 25.8% reduction compared to the solution obtained in Section 2.2.1, suggesting that due to the addition of the new activities, the schedule became denser. It can be illustrated from Figure 2.18 that a minimum of 1 and a maximum of 6 activities are in progress at any point in time, with an average of 2.44 concurrent activities. Figure 2.19 highlights that although the aspiration criteria for type 1 resources is set to 5 units, the goal has been surpassed by one unit i.e. $W_{1} = W_{1}^{max} + 1$, in an effort to complete all activities over the span of the pre-defined scheduling horizon. Similarly, it can be established through Figure 2.20 that the type 2 resource constraint has been successfully satisfied with no deviation to its aspiration criterion i.e. $W_{2} \leq W_{2}^{max}$. Figures 2.21 through 2.26 represent zonal density and capacity over time, verifying that all zonal constraints have been successfully satisfied, where at least one of the zones are completely saturated for 11.6% of active production hours, yielding to an average increase of 103% in zonal saturation compared to the results obtained in Section 2.2.1. It can thus be concluded that the resultant schedule has satisfied all applicable constraints and that the mathematical programming model proposed in Section 2.1.2 is an effective method in scheduling optimization of activities in LVLVPS in scenarios where travel work is prohibited.

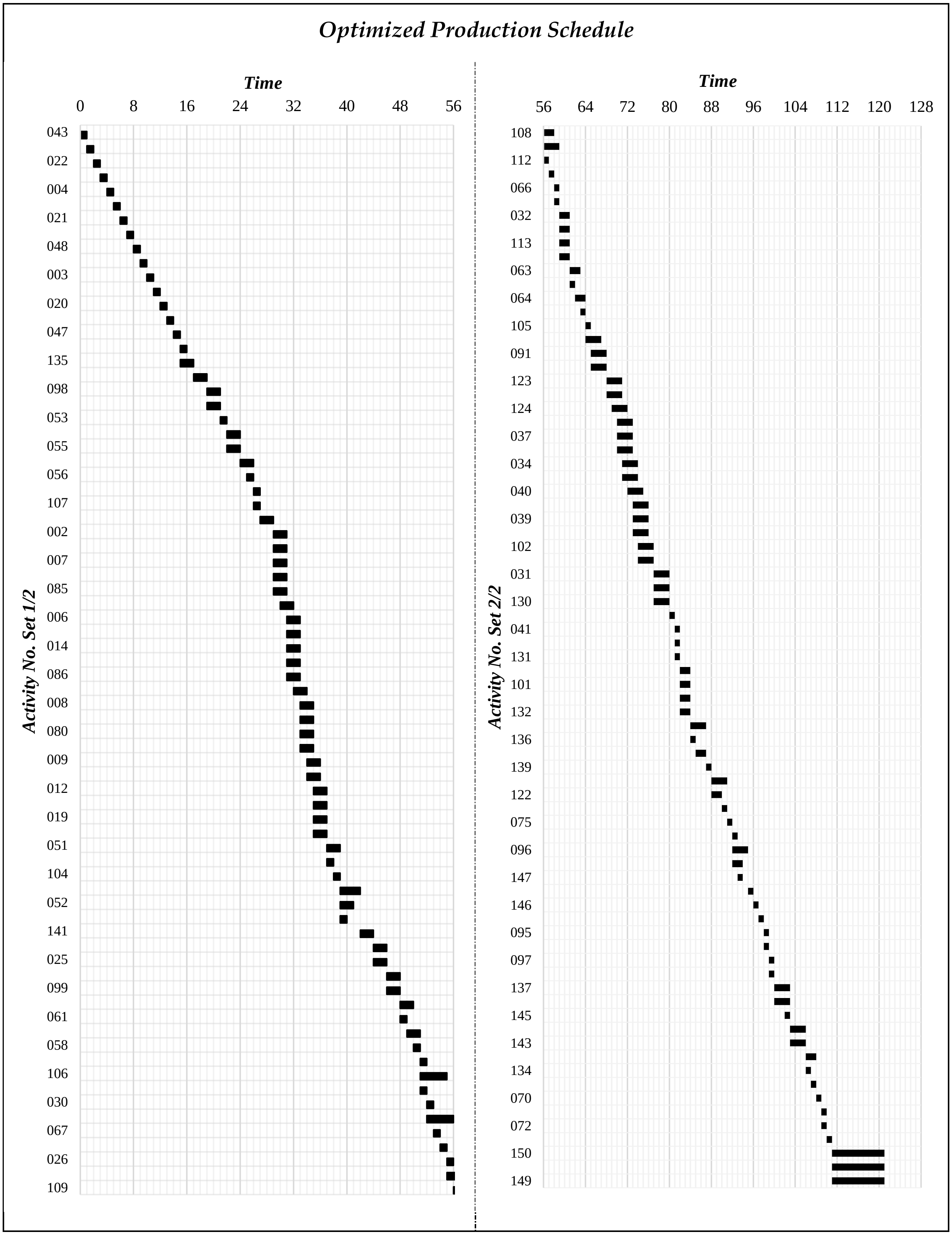

Figure 2.16 - Gantt Chart for the Optimized Work Center Schedule

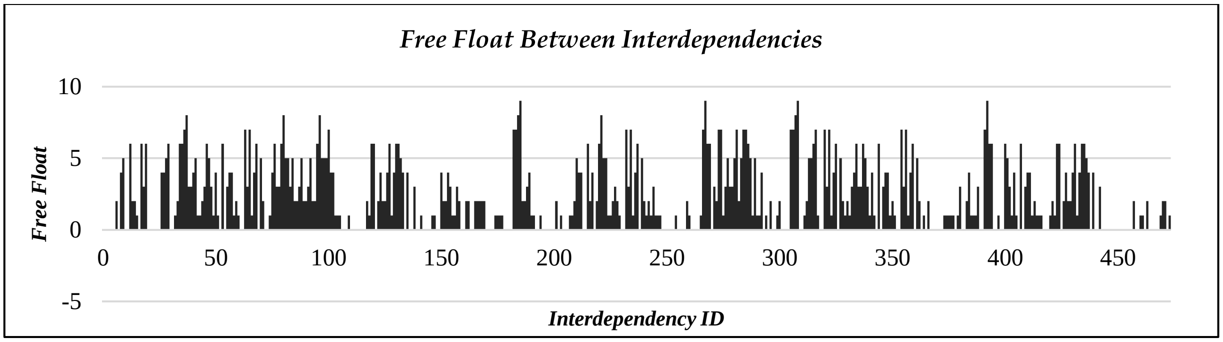

Figure 2.17 - Free Float Between Interdependent Activities

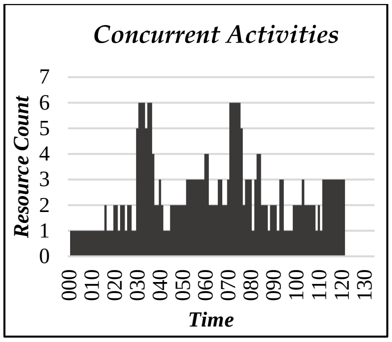

Figure 2.18 - Concurrent Activities

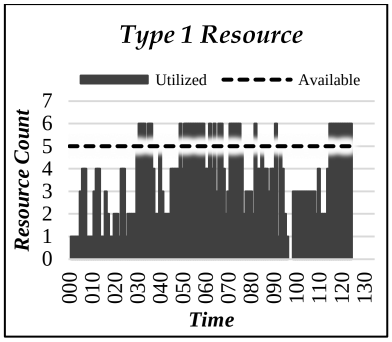

Figure 2.19 - Resource 1 Utilization

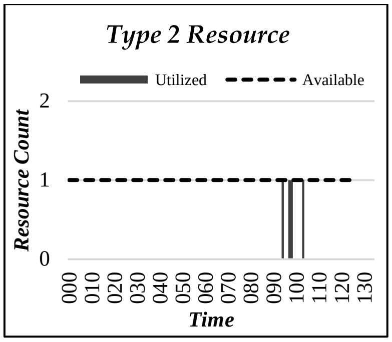

Figure 2.20 - Resource 2 Utilization

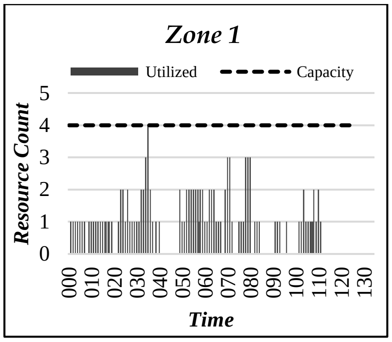

Figure 2.21 - Zone 1 Utilization

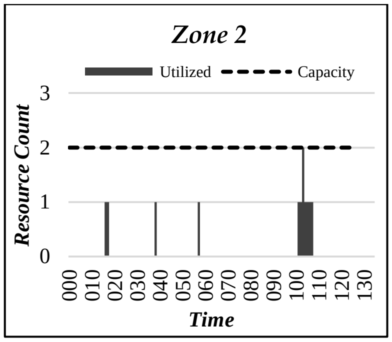

Figure 2.22 - Zone 2 Utilization

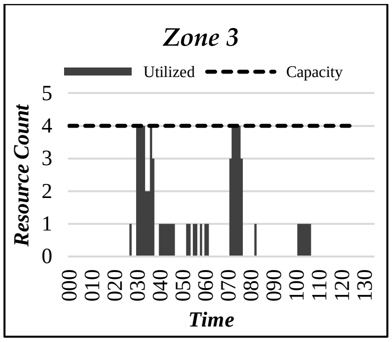

Figure 2.23 - Zone 3 Utilization

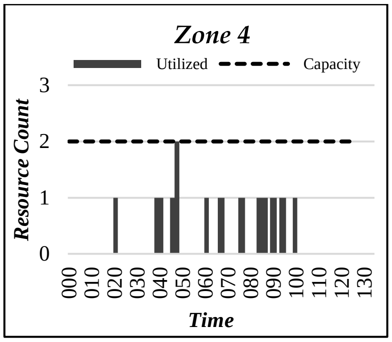

Figure 2.24 - Zone 4 Utilization

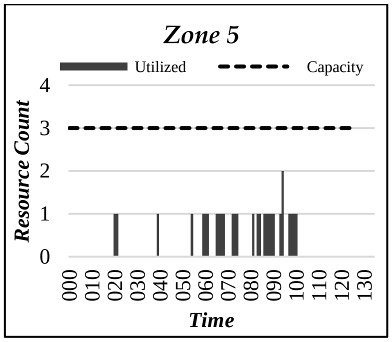

Figure 2.25 - Zone 5 Utilization

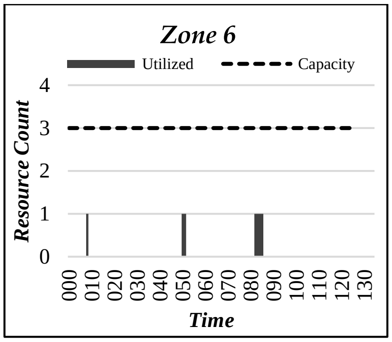

Figure 2.26 - Zone 6 Utilization

2.2.3. Capacity Study

In this section, the mathematical programming model proposed in Section 2.1.3 is applied in evaluating this work center's capacity, given the complete saturation of resources. This dual-objective problem is solved in two phases, following the lexicographical order prescribed in objective function (2.26), where the problem is initially solved for the higher priority level objective $P1\left[\sum_t tx_{N+1,t}\right]$, aimed at minimizing the makespan. Solving this phase of the problem yields to an objective value of $\sum_t tx_{N+1,t} = 96$ hours, signifying that the minimum feasible makespan for the completion of all activities. The problem is then solved for the lower level priority objective, aimed at minimizing the required number of resources $P2[\sum_l W_l]$, resulting in an overall resource requirement of $\sum_l W_l = 16$ units, comprised of $W_1 = 14$ units of type 1, $W_2 = 2$ units of type 2 resources. It is thus concluded that 16 resources are required to complete the pre-defined statement of work in 96 hours. Figure 2.27 is the Gantt chart of the resultant saturated schedule, demonstrating activity start and finish times. This methodology is most effective in performing capacity studies and early detection of bottlenecks for desired takt-times. For instance, through this case study, the lowest feasible makespan for completion of all activities in-station is found to be 96 hours, equivalent to 12 8-hour shifts. If the desired takt-time is 6 days, then a 2-shift operation can be implemented, in a scenario mandating a takt-time of 4 days, a 3-shift operation will be required. However, scenarios mandating a takt-time that is less than 4 days would either require the incorporation of parallel work centers, or the permanent reallocation of activities to other work centers. Similar to the verification process in Section 2.2.1 and 2.2.2, the results obtained from solving this model are verified through the analysis of constraints. It is demonstrated through Figure 2.28 that all interdependencies have been successfully satisfied, with an average free float of 0.96 hours, representing a 31.4% decrease in slack time compared to the results obtained in Section 2.2.1. Analysis of Figure 2.29 highlights that a maximum of 14 activities is in progress at each time interval, with an average of 2.77 concurrent activities, 27% greater than the results obtained in Section 2.2.1. Figures 2.30 and 2.31 demonstrate a maximum utilization of 14 units of type 1 and 2 units of type 2 resources respectively, where upper bound on resources have been relaxed. Zonal density and capacity are depicted in Figures 2.32 to 2.37, through which it is concluded that all zonal constraints are successfully satisfied, where at least one of the zones are completely saturated for 16.7% of active production hours, representing a 280% increase in zonal saturation compared to the results obtained in Section 2.2.1.

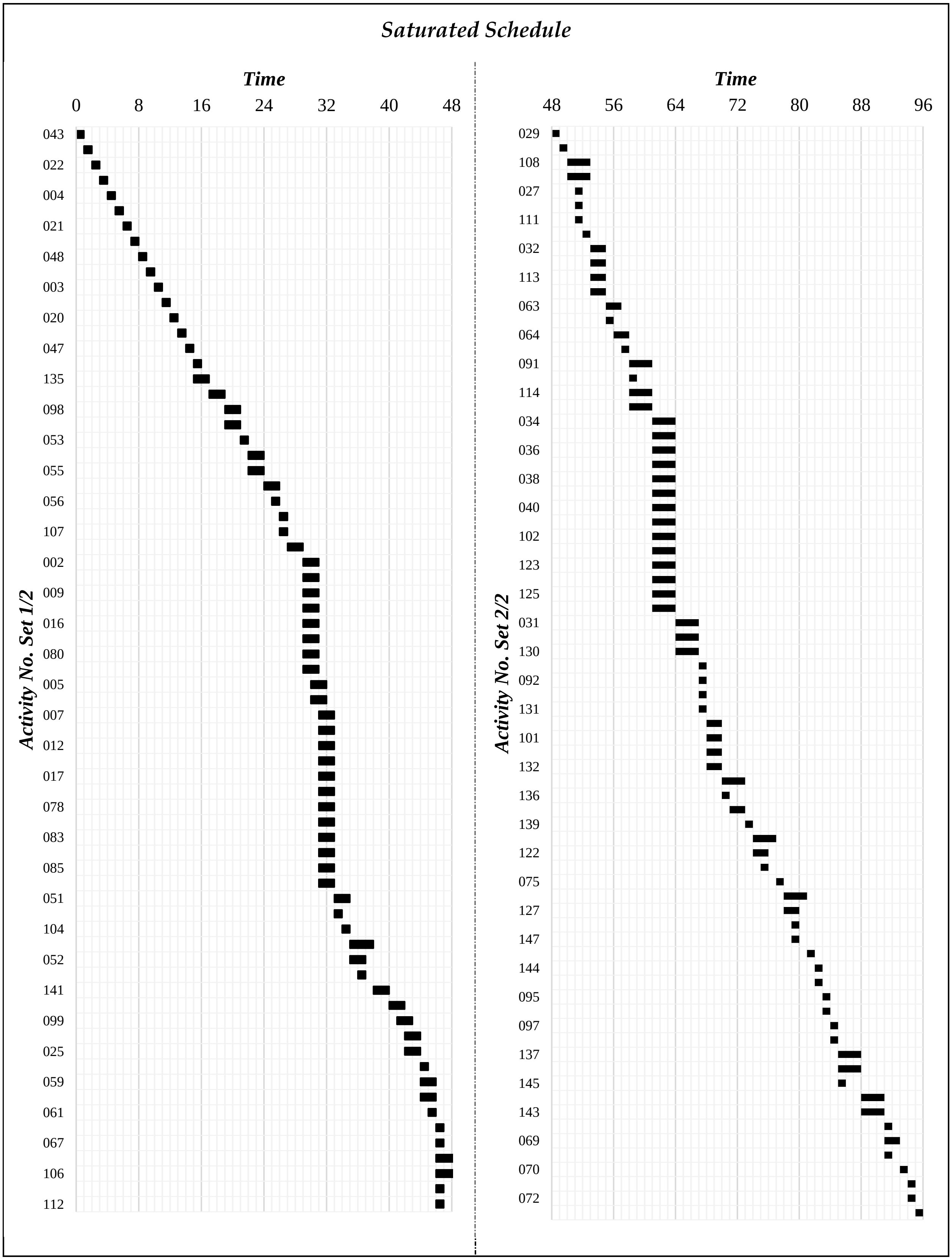

Figure 2.27 - Gantt Chart for the Saturated Schedule

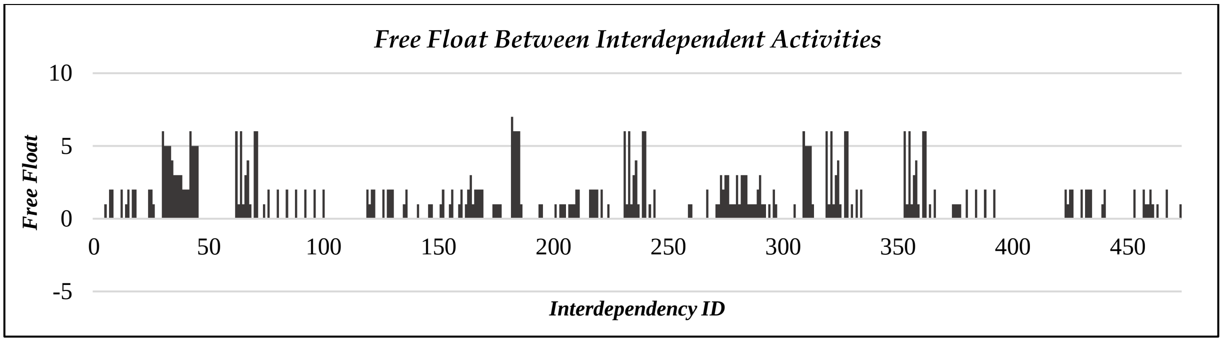

Figure 2.28 - Free Float Between Interdependent Activities

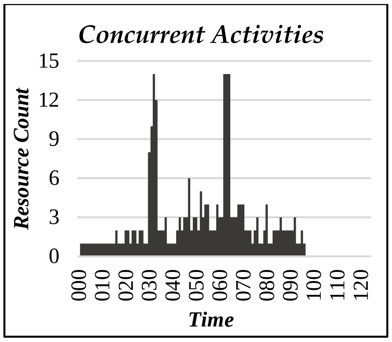

Figure 2.29 - Concurrent Activities

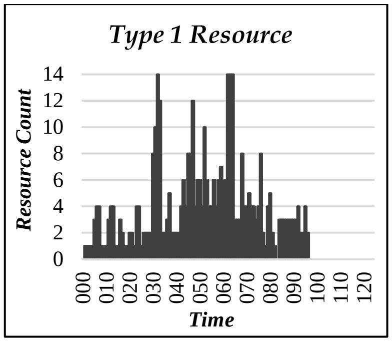

Figure 2.30 - Resource 1 Utilization

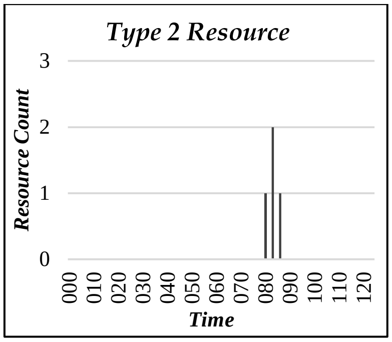

Figure 2.31 - Resource 2 Utilization

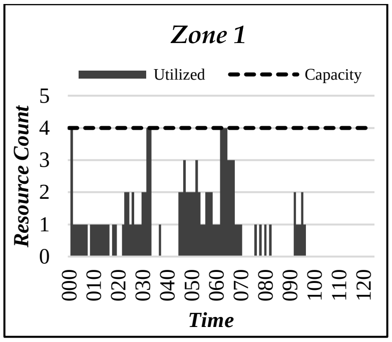

Figure 2.32 - Zone 1 Utilization

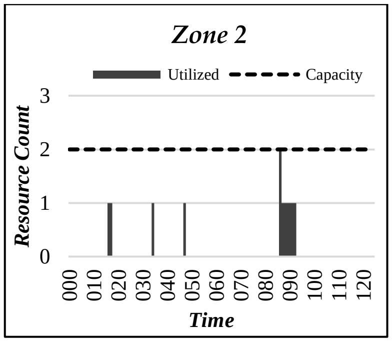

Figure 2.33 - Zone 2 Utilization

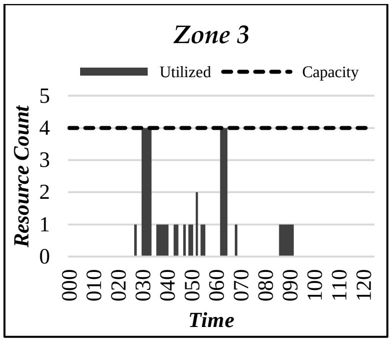

Figure 2.34 - Zone 3 Utilization

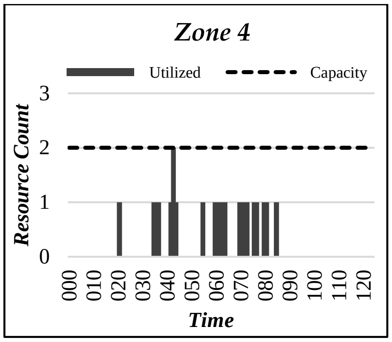

Figure 2.35 - Zone 4 Utilization

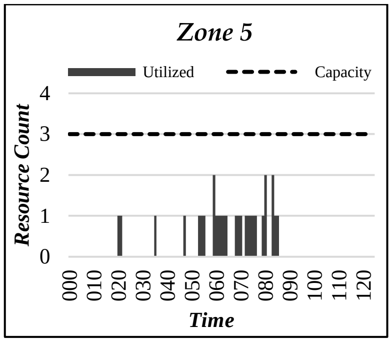

Figure 2.36 - Zone 5 Utilization

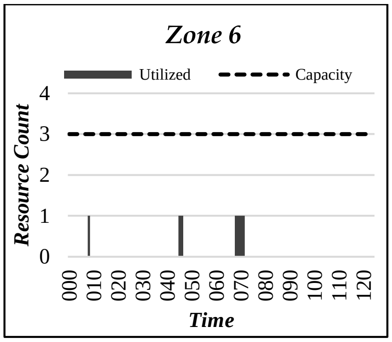

Figure 2.37 - Zone 6 Utilization

2.3. Chapter Summary

In this chapter, a series of discrete-time multi-objective mixed-integer linear mathematical programming models have been formulated and proposed for modeling and solving of scheduling problems in LVLV PS. The initial model proposed in Section 2.1.1 adopts a pre-emptive goal programming approach in the formulation of a priority-based multi-objective function, aimed at minimizing the number of resources required in the completion of the maximum number of activities, over the span of the imposed takt times. This model is recommended to be used in scenarios where the strict enforcement of the time and resource constraints may result in traveling of incomplete activities and is applicable to scenarios where travel work is permitted. Similarly, a pre-emptive goal programming approach is employed in the formulation of an alternative mathematical model in Section 2.1.2, aimed at minimizing the positive deviation of makespan and resource requirements to their respective aspiration criteria while minimizing the overall resource requirements. This mathematical programming model is applicable to scenarios where travel work is prohibited and the work center is responsible to complete all activities in-station. Lastly, in Section 2.1.3 a lexicographic method is adapted in the formulation of a multi-objective function, aimed at minimizing resource requirements and makespan, where the latter takes precedence. This mathematical model is recommended to be used to provide managerial insight into a work center's capabilities, given the complete saturation of resources. The proposed mathematical programming models capture critical constraints and characteristics omitted in previously reported literature, accomplished through the incorporation of time and resource constraints. Additionally, through the proposal of two distinct mathematical models, decision-makers' priorities and preferences can be accurately reflected to ensure the practicality of the resultant schedule. To ensure accuracy and reliability, the proposed mathematical models were validated and verified with a numerical experiment through a real-world case study with a global leader in the aerospace industry. It is concluded through this study, that the proposed mathematical programming models are effective methods for the optimization of large-scale industrial scheduling problems in LVLVPS.

CHAPTER 3

DISCRETE-TIME PARALLEL WORK CENTERS

The integration of parallel work centers is a common and practical approach in shifting of bottlenecks and balancing of assembly lines [107]. This manufacturing practice often results in improved production capacity and fulfillment of takt-times otherwise found infeasible. Through this chapter, we examine the implications of the integration of parallel work centers in LVLVPS and formulate a set of mathematical programming models for optimization of single and multi-parallel work centers scheduling problems with shared or dedicated resources. The mathematical programming models presented in this chapter were published in the Journal of Applied Mathematical Modelling [108]. As described in Chapter 1, products assembled in LVLVPS are processed through a series of work centers, where a budgeted number of resources of various classifications are deployed onto the product. Note that the quantity and classifications of resources assigned to a work center are in-line with the allocated statement of work and that resources of a particular classification cannot execute activities requiring other skill types. An example of which is a mechanical assembler's inability to swage fuel pipes, install and test electrical harnesses, or apply aerodynamic sealing to an aircraft. Work centers are responsible for completing a pre-assigned statement of work, comprised of a set of interdependent activities, with the budgeted resources, through the span of the available time. Activities are assumed to be none pre-emptive, signifying that activities cannot be interrupted and are to be completed once started. Furthermore, activities may be single or multi-mode, where alternative modes can be crashed through the allocation of additional resources. In addition to time, resource and interdependency constraints, there exist non-concurrencies, constraining the concurrent progression of two or more activities, activities in two or more zones, or a combination of activities and zones. Activities are unique to specific zones, defined as the physical location of work on the product, subject to pre-defined allowable capacity. There may also exist constraints enforcing simultaneous start or finish of two or more activities, or mandating an earliest or latest start or finish times on activities.

Whilst the objective of work centers is the completion of the pre-defined set of activities, the scarcity of time and resources may result in failure to fulfill this objective. Such infeasibilities yield to the traveling of incomplete activities, as their completion in the originating work center, results in the stoppage of the assembly line or the need for additional resources. A practical alternative to increasing a work center's capacity is the integration of identical parallel work centers, yielding to an increase in the available planning horizon by a factor equivalent to the number of parallel work centers. Resources in parallel work centers may be either dedicated to a single work center or may be shared between multiple parallel work centers, which may potentially lead to lower labor costs and reduced resource requirements. Furthermore, to ensure schedule standardization between parallel work centers, an identical schedule must be executed by all parallel work centers. Table 3.01 summarizes activity attributes, assumptions, and constraints in the mathematical programming models proposed in this chapter.

Activity Attributes	*Assumptions*	*Constraints*
• Processing Time • Interdependencies • Lead & Lag Times • Non-Concurrencies • Concurrencies • Single or Multi-Mode • Resource Classification • Resource Quantity • Zonal Assignment • Earliest Start & Finish Times • Latest Start & Finish Times	• Discrete Processing Time • Discrete Planning Horizon • Pre-emption Not Allowed • Equal Resource Load per Shift • Equal Resource Efficiency	• Resource Constraints • Time Constraints • Precedence Constraints • Lead & Lag Time Constraints • Non-Concurrency Constraints • Concurrency Constraints • Zonal Saturation Constraints • Identical Work Center Schedules • Earliest Start & Finish Times • Latest Start & Finish Times •

Table 3.01 - Assumptions for Discrete-Time Multi-Parallel Work Center Scheduling Problems

3.1. Mathematical Programming Models

The mixed-integer mathematical programming models formulated in this section are capable of modeling characteristics and constraints inherent in LVLVPS, and are compatible with single or multi-parallel work center scheduling problems with shared or dedicated resources. In Section 3.1.1, sets, variables, and parameters used in formulating the proposed mathematical models are defined, and a set of preprocessing procedures are established. Two objective functions have been formulated in Section 3.1.2 to improve flexibility and to allow for decision-makers to accurately reflect their preferences and priorities, differentiated by a deciding criterion of permitting or prohibiting travel work. To minimize resource requirements and to maximize the number of completed activities, in scenarios permitting the traveling of incomplete activities, a multi-objective function is formulated in Section 3.1.2.1, employing the lexicographic method. An additional objective function is formulated in Section 3.1.2.2, adapting a pre-emptive goal programming approach aimed at minimizing the positive deviation to the aspiration criteria for time and resources, where the former takes precedence in scenarios prohibiting travel work. Applicable constraints are formulated in Section 3.1.3, a combination of which may be used to tailor the proposed mathematical model to be reflective of the specific scheduling problem at hand.

3.1.1. Variables, Parameters, and Pre-Processing Procedures

3.1.1.1. Sets, Parameters, and Variable Definition

Sets, parameters, and variables defined in Table 3.02 are used in the formulation of the mixed-integer programming model proposed in this chapter. Note that all sets and parameters are to be collected in their entirety and are to be presented in the form of matrices or integer where applicable.

Component	***Notation***		***Description***
Sets	Σ	$\sigma \in \{ 1, \dots, \Phi \}$	Work Center Number
	Γ	$\gamma \in \{ 1, \dots, \Upsilon \}$	Product Number
	S_Γ	$S_\gamma \in t$	Release Time of Product γ
	J	$j \in \{ 1, \dots, N \}$	Activity Number
	J', J''	$j', j'' \equiv j$	The Equivalent Set of Activity j
	L	$l \in \{ 1, \dots, L \}$	Resource Classification
	I	$i \in \{ 1, \dots, I \}$	Zone Classification

Component	*Notation*		*Description*
Sets	I	$i' \equiv i$	The Equivalent Set of Zone i
	T	$T \in t$	Desired Takt-Time
	D_{Φ}	$D_{\Phi} = \Phi T$	Available Time Per Work Center
	T	$t \in \left[1, \frac{3\Phi-2}{\Phi} D_{\Phi}\right]$	Discrete-Time Planning-Horizon
	U	$u = [\, t-1, t\,]$	Single Time Interval
	A	$\alpha \in j \setminus \beta$	Single-Mode Activities
	B	$\beta \in j \setminus \alpha$	Multi-Mode Activities
Parameters	$M_{jj'}$	$M_{jj'} \in \{\, 0,1 \,\}$	Multi-Mode of Activities j and j'
	p_j	$p_j \geq 0$	Processing Time of Activity j
	$P_{jj'}$	$P_{jj'} \in \{\, 0,1 \,\}$	Precedence Between Activities j and j'
	$NC_{jj'}$	$NC_{jj'} \in \{\, 0,1 \,\}$	Non-Concurrency Between Activities j and j'
	$NC_{ii'}$	$NC_{ii'} \in \{\, 0,1 \,\}$	Non-Concurrency Between Zones i and i'
	NC_{ji}	$NC_{ji} \in \{\, 0,1 \,\}$	Non-Concurrency Between Activity j with Zones i
	$CS_{jj'}$	$CS_{ji} \in \{\, 0,1 \,\}$	Concurrent Start Between Activities j and j'
	$CF_{jj'}$	$CF_{ji} \in \{\, 0,1 \,\}$	Concurrent Finish Between Activities j and j'
	$L_{jj'}$	$L_{jj'} \in [\, -\mathrm{D}_{\Phi}, +\mathrm{D}_{\Phi} \,]$	Lead/Lag Time Between Activities j and j'
	$y_{\sigma ji}$	$y_{\sigma ji} \in \{\, 0,1 \,\}$	Allocation of Activity j to Zone i of Work Center σ
	Z_i	$Z_i > 0$	Capacity of Zone i
	B	$B > 0$	Resource Budget
	W_l^{max}	$W_l^{max} \geq 0$	Resource Availability of Pool l
	w_{jl}	$w_{jl} \geq 0$	Resource Requirement of Activity j from Pool l
	ES_j	$ES_j \in [\, 1, T \,]$	Earliest Start Time of Activity j
	LS_j	$LS_j \in [\, 1, T \,]$	Latest Start Time of Activity j
	EF_j	$EF_j \in [\, 1, T \,]$	Earliest Finish Time of Activity j
	LF_j	$LF_j \in [\, 1, T \,]$	Latest Finish Time of Activity j
Variables	W_l	$W_l \geq 0$	Resource Requirement from Pool l
	π_j	$\pi_j \in \{\, 0,1 \,\}$	Scheduling of Activity j
	$x_{\gamma jt}$	$x_{\gamma jt} \in \{\, 0,1 \,\}$	Completion of Activity j of product γ at Time t

Table 3.02 - Sets, Parameters, and Variables for Discrete-Time Multi-Parallel Work Center Scheduling Problems

3.1.1.2. Required Number of Products

A key parameter in solving multi-parallel work center scheduling problems is the required number of products Υ to be processed through the system. It is essential to calculate this parameter as a part of the pre-processing procedure, as it is critical in calculating the minimum number of products to be processed through the system, such that all work centers are simultaneously active.

$$\Upsilon = 2\Phi - 1 \tag{3.01}$$

The linear function illustrates in Figure 3.01, demonstrates the required number of products as a function of the number of parallel work centers as presented in Equation (3.01). It is critical to calculate this parameter as a part of the pre-processing procedure to ensure accurate modeling of multi-parallel work center scheduling problems. The required number of products Υ is equivalent to the number of products that ensure that there exists a time interval $[t-1, t]$ that all work centers are simultaneously active. Note that in solving single work center scheduling problems or multi-parallel work center scheduling problems with dedicated resources the number of parallel work centers is set to $\Phi = 1$, concluding that only a single product $\Upsilon = 1$ is required in the modeling of the scheduling problem.

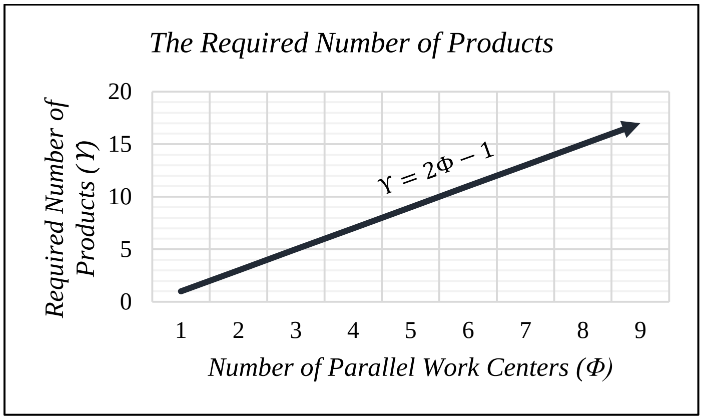

Figure 3.01 - Linear Function of the Required Number of Products

3.1.1.3. Available Processing Time, Planning Horizon, and Release Times

The available processing time at each work center, denoted by D_{Φ}, represents the imposed time constraint of an assembly line with Φ parallel work centers, operating at a takt-time of T. D_{Φ} is calculated through Equation (3.02), and is substituted into Equation (3.03) and (3.04) in calculating the overall planning horizon and product release times, respectively.

$$D_{\Phi} = \Phi T \tag{3.02}$$

$$t \in \left[1, \frac{\Upsilon - 1 + \Phi}{\Phi} D_{\Phi}\right] = \left[1, \frac{3\Phi - 2}{\Phi} D_{\Phi}\right] \tag{3.03}$$

$$S_\gamma = \frac{\gamma - 1}{\Phi} D_\Phi \tag{3.04}$$

Note that in case of single or multi-parallel work center scheduling problems with dedicated resources, the planning horizon reduces to $t \in [1, T]$, and the product is released at $S_\gamma = 0$. Figure 3.02 illustrates the release S_γ and the completion time C_γ of product γ, where the shaded area represents the overlapping of work between the products.

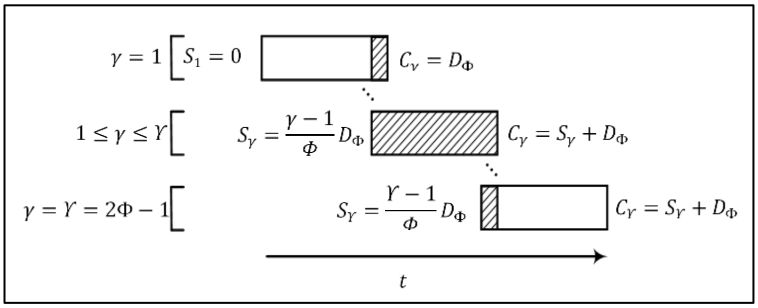
$\gamma = 1$
$S_1 = 0$
$C_\gamma = D_\Phi$
$1 \leq \gamma \leq \Upsilon$
$S_\gamma = \frac{\gamma - 1}{\Phi} D_\Phi$
$C_\gamma = S_\gamma + D_\Phi$
$\gamma = \Upsilon = 2\Phi - 1$
$S_\Upsilon = \frac{\Upsilon - 1}{\Phi} D_\Phi$
$C_\Upsilon = S_\Upsilon + D_\Phi$
t

Figure 3.02 - Product Release and Completion Times

Note that the processing and release times of each product and the overall planning horizon are independent of the layout of the assembly line. The proposed mixed-integer mathematical models are compatible with a wide range of assembly line layouts, represented in Figure 3.03.

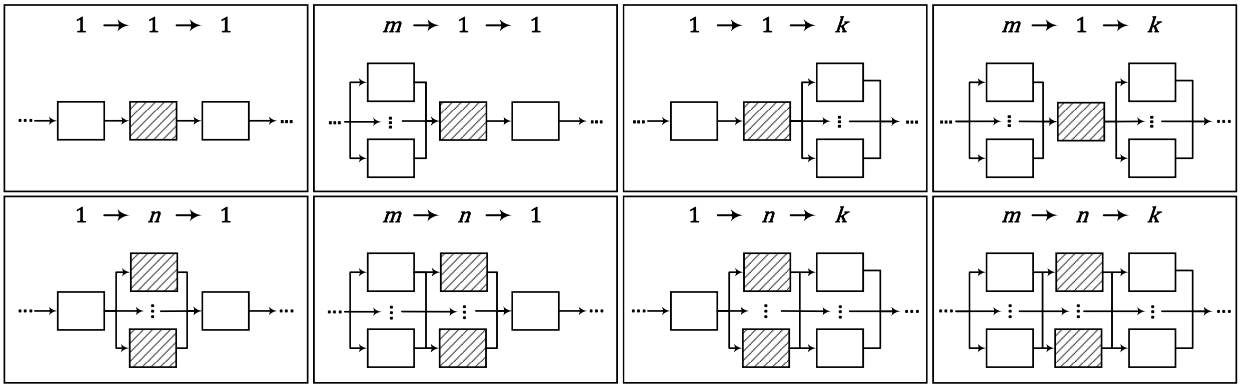

Figure 3.03 - Compatible Assembly Line Configurations

Figure 3.04 demonstrates the start time, duration and the overall planning horizon in a sample scheduling problem with three parallel work centers and a takt-time of 40 hours. Following Equation (3.01), five products are required to be processed through the system, with an available processing time of 120 hours per work center and an overall planning horizon of 280 hours, calculated through Equations (3.02) and (3.03) respectively. The shaded area in Figure 3.04 and 3.05 represents the overlapping of work, validating the need for the processing of five products in solving this sample scheduling problem. Additionally, Figure 3.05 is intended to illustrate the assignment of product γ to work center σ, to be further discussed in Section 3.1.1.4.

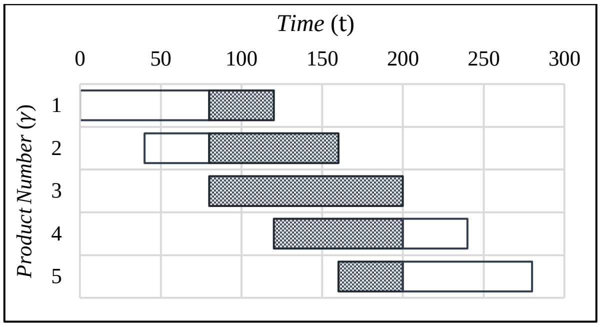

Figure 3.04 - Available Planning Horizon per Product

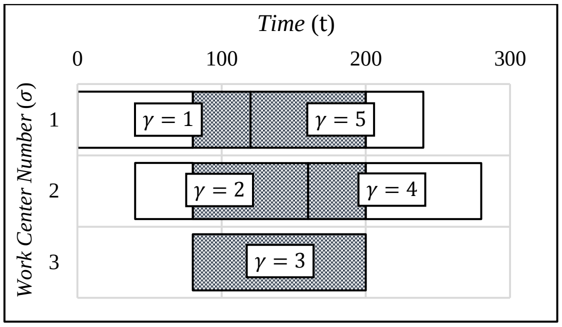

Figure 3.05 - Assignment of Products to Work Centers

3.1.1.4. Assignment of Products to Work Centers

Products must be allocated to the first available work center following a First-In-First-Out (FIFO) methodology to ensure compatibility. The first product released $\gamma = 1$ is thus allocated to work center $\sigma = 1$ and subsequent products are assigned to the next available work centers. This rule eliminates work center idle times and ensures that an optimum schedule is obtained from solving the proposed mathematical programming models.

3.1.2. Objective Function

Scheduling problems in LVLVPS can be optimized by minimizing the number of incomplete activities, resource requirements, makespan, or deviation to a set of aspiration criteria. The selection of an objective function is dependent on the preferences and priorities of the decision-maker and may differ depending on the characteristics and features inherent in the production system under study. In this section, objective functions most applicable to scheduling optimization of single or multi-parallel work centers with shared or dedicated resources are formulated and proposed.

3.1.2.1. Travel Work Permitted

A common feature of scheduling problems in LVLVPS is the flexibility to travel work due to limited resources and/or time, to be completed in work centers with excess capacities. The objective function formulated in this section is recommended to be used in scenarios where the flexibility in traveling of work is permitted. In the formulation of the objective function represented in Equation (3.05), the lexicographic method is adopted, where the problem is solved iteratively following the imposed lexicographical order [104]. The proposed multi-objective function minimizes the quantity of the required resources $P_2[W_l]$ while minimizing the number of travelled activities $P_1\left[N - \sum_j \sum_t x_{\gamma=1,j,t}\right]$, where the latter takes precedence.

$$Min\ Z \ = P_1 \left[N - \sum_j \sum_t x_{\gamma=1,j,t} \right] + P_2\ [W_l] \tag{3.05}$$

3.1.2.2. Travel Work Prohibited

The completion of the assigned statement of work may be mandated by the organization in scenarios where incomplete activities may impede the successful execution of work assigned to subsequent work centers. To calculate the makespan of the assigned work package, an artificial activity $j = N + 1$ is presented with a duration of $p_{N+1} = 0$, representing the end node of the activity on the node network diagram. This artificial activity is a successor to activities without successors and its completion time is equivalent to the makespan of the assigned work package. The pre-emptive goal programming model is employed in the formulation of the multi-objective functions presented in Equations (3.06) and (3.07), with two and three levels of priorities, respectively. Goal programming is an effective approach to solving large-scale multi-objective optimization problems. Through this methodology, the problem is solved in multiple iterations, in order of the priority sequence, with the objective of minimizing the deviation of decision variables to their respective aspiration criterion [109,110].

$$Min\ Z \ = P_1[\delta_1^+] - P_2[\delta_2^-] \tag{3.06}$$

$$Min\ Z \ = P_1[\delta_1^+] + P_2 \left[\sum_l \delta_{3l}^+ \right] - P_3[\delta_2^-] \tag{3.07}$$

These objective functions are employed in solving single or multi-parallel work center scheduling problems with shared or dedicated resources. Objective function (3.06) is proposed for scenarios in which an overall time and resource budget is imposed onto the work center, and is solved in two iterations, where the first priority objective P_1 is intended to minimize the positive deviation of makespan from the imposed aspiration criterion to time δ_1^+, while the second priority objective P_2 maximizes the negative deviation to the budgeted number of resources δ_2^-. Deviation variables δ_1^+, δ_2^-, and δ_{3l}^+ later used in objective function (3.07) are evaluated through Constraints

(3.28), (3.25) and (3.26) respectively, and represent the deviation of the decision variables from the corresponding aspiration criterion. In objective function (3.07), a new objective $P_2[\sum_l \delta_{3l}^+]$ is added to the previously formulated objective function (3.06), with a priority level two. This objective is intended to minimize the positive deviation to the available number of resources per classification, to be used in scenarios with limited access to specialized human resources, as in the case of aerospace.

3.1.3. Constraints

The proposed mixed-integer programming models are compatible with single or multi-parallel work center scheduling problems with shared or dedicated resources and may permit or prohibit the traveling of activities. In this section, a series of constraints applicable to be used for various scheduling problems in LVLVPS are formulated. Table 3.03 is a summary of formulated constraints, and their applicability to the selected objective function and problem classification. Note that while the applicability of some constraints is considered mandatory or restricted, the employment of other constraints is at the discretion of the decision-maker. Discretionary constraints are intended to enforce characteristics which may be encountered in some problems but are not generic across all scheduling problems in LVLVPS.

✓ Mandatory O Discretionary × Restricted

Constraint No.	***3.1.2.1. Travel Work Permitted***			***3.1.2.2. Travel Work Prohibited***		
	Single Work Center	*Multi-Parallel Work Centers*		*Single Work Center*	*Multi-Parallel Work Centers*	
	Dedicated Resources	*Dedicated Resources*	*Shared Resources*	*Dedicated Resources*	*Dedicated Resources*	*Shared Resources*
3.08	×	×	✓	×	×	✓
3.09	×	×	✓	×	×	✓
3.10	✓	✓	✓	×	×	×
3.11	✓	✓	✓	×	×	×
3.12	×	×	×	✓	✓	✓
3.13	×	×	×	✓	✓	✓
3.14	✓	✓	✓	✓	✓	✓
3.15	✓	✓	✓	✓	✓	✓
3.16	✓	✓	✓	×	×	×
3.17	✓	✓	✓	×	×	×
3.18	O	O	O	O	O	O
3.19	O	O	O	O	O	O
3.20	O	O	O	O	O	O
3.21	O	O	O	O	O	O

✓ Mandatory O Discretionary × Restricted

Constraint No.	***3.1.2.1. Travel Work Permitted***			***3.1.2.2. Travel Work Prohibited***		
	Single Work Center	*Multi-Parallel Work Centers*		*Single Work Center*	*Multi-Parallel Work Centers*	
	Dedicated Resources	*Dedicated Resources*	*Shared Resources*	*Dedicated Resources*	*Dedicated Resources*	*Shared Resources*
3.22	✓	✓	✓	✓	✓	✓
3.23	✓	✓	✓	×	×	×
3.24	✓	✓	✓	×	×	×
3.25	×	×	×	✓	✓	✓
3.26	×	×	×	✓	✓	✓
3.27	✓	✓	✓	×	×	×
3.28	×	×	×	✓	✓	✓
3.29	O	O	O	O	O	O
3.30	O	O	O	O	O	O
3.31	O	O	O	O	O	O
3.32	O	O	O	O	O	O
3.33	O	O	O	O	O	O
3.34	O	O	O	O	O	O

Table 3.03 - Application of Constraints to Various Types of Scheduling Problems

In the scheduling of activities for multi-parallel work centers, it is imperative to create a standard schedule and a sequence of work that allows for repeatability. A standardized schedule is defined as one that ensures all products start the pre-defined activities at the same time, offset by the duration of their release time S_γ into work centers. For instance, an activity that starts at $t = 0$ and finishes at $t = 10$ for the first product in line will start at $t = S_\gamma$ and finishes at $t = S_\gamma + 10$ for all subsequent products. This is accomplished through constraint sets (3.08) and (3.09), enforcing the fixed start and finish times and the scheduling of an identical set of activities for all products. Constraint Set (3.08) enforces the completion of an identical set of activities on all products, where $\sum_t x_{\gamma jt} = 1$ if activity j of product γ is scheduled to be completed, and $\sum_t x_{\gamma jt} = 0$ if activity j is to be traveled. This also results in the scheduling of identical modes of multi-mode activities where applicable, thus ensuring standardization in the selection of an optimum mode for all multi-mode activities. Constraint Set (3.09) is similarly formulated to standardize the time in which activities are scheduled to be completed for products processed in each work center, where $\sum_t tx_{\gamma jt}$ represents the completion time of activity j of product γ. This constraint set will not only ensure standardization and repeatability in the resultant schedule, but will also prove to improve the efficiency in formulating the remainder of constraints. Subsequent constraints presented in this section will rely on constraint sets (3.08) and (3.09) in the optimization of scheduling problems with multi-parallel work centers, while directly imposing constraints only on the first product in line $\gamma = 1$.

$$\sum_t x_{\gamma jt} = \sum_t x_{\gamma+1,j,t} \qquad \begin{array}{l} \forall \;\; j = 1, \dots, N \\ \forall \;\; \gamma = 1, \dots, \Upsilon - 1 \end{array} \tag{3.08}$$

$$\sum_t tx_{\gamma+1,j,t} = \sum_t tx_{\gamma jt} + S_\gamma \sum_t x_{\gamma jt} \qquad \begin{array}{l} \forall \;\; j = 1, \dots, N \\ \forall \;\; \gamma = 1, \dots, \Upsilon - 1 \end{array} \tag{3.09}$$

The scarcity of time and resources in LVLVPS may result in scenarios where incomplete activities are traveled to downstream work centers. Constraint Sets (3.10) are formulated to ensure that activities can only be scheduled once. Furthermore, constraint set (3.11) is formulated to capture multi-mode activities $j \in \beta$, signifying that only a single mode can be scheduled. Binary variable $M_{jj'}$ represents the relationship between multiple modes of multi-mode activities, where $M_{jj'} = 1$ if j' is a secondary mode of activity j, and $M_{jj'} = 0$ if otherwise.

$$\sum_t x_{\gamma=1,j,t} \leq 1 \qquad \forall \;\; j \in \alpha \tag{3.10}$$

$$\sum_t x_{\gamma=1,j,t} + \sum_t x_{\gamma=1,j',t} \leq 1 \qquad \forall \;\; j, j' \in \{ \beta \;\; : M_{jj'} = 1 \} \tag{3.11}$$

Subsequently, constraint sets (3.12) and (3.13) are alternative forms of constraint sets (3.10) and (3.11), respectively, to be used in conjunction with objective function (3.06) and (3.07), where travel work is prohibited. Constraint Set (3.12) enforces the completion of single-mode activities $j \in \alpha$, and constraint set (3.13) enforces that one mode of multi-mode β activities must be scheduled.

$$\sum_t x_{\gamma=1,j,t} = 1 \qquad \forall \;\; j \in \alpha \tag{3.12}$$

$$\sum_t x_{\gamma=1,j,t} + \sum_t x_{\gamma=1,j',t} = 1 \qquad \forall \;\; j, j' \in \{ \beta \;\; : M_{jj'} = 1 \} \tag{3.13}$$

Precedence constraints are imposed when the starting of an activity j' requires the completion of its predecessor activity j. The completion time of activity j of product γ and the start time of activity j' of product γ are calculated through $\sum_t tx_{\gamma jt}$ and $\sum_t (t - p_{j'}) x_{\gamma j' t}$, respectively. Activities in LVLVPS are often constrained with multiple interdependencies. For simplicity and to minimize the time required in the programming of this constraint using the Optimization

Programming Language (OPL)™ , a binary precedence matrix $P_{jj'}$ is introduced as an input to the model, where $P_{jj'} = 1$ if j is a predecessor of j' and $P_{jj'} = 0$ otherwise. Lead and lag times are represented through variable $L_{jj'}$, where $L_{jj'} > 0$ if activity j'can start only after an imposed lag time has elapsed from the completion of activity j, and $L_{jj'} < 0$ is scenarios where a successor activity requires a pre-defined progression of its predecessor(s) prior to its commissioning. This variable assumes a value of zero in scenarios where lead or lag time between interdependent activities are not observed. Note that neither the precedence $P_{jj'}$ nor the lead or lag time matrices $L_{jj'}$ are not product dependent. This is due to uniformity in the work packages assigned to work centers. Precedence constraints are represented through constraints sets (3.14) through (3.17). Constraint set (3.14) is formulated to establish the value of the binary variable π_j, subsequently used in constraint set (3.15) to enforce precedence constraints, where $\pi_j = 0$ if activity j has been scheduled, and $\pi_j = 1$ otherwise. Constraint set (3.15) enforces the successful fulfillment of precedence and lead and lag time constraints, where applicable. This is accomplished through ensuring that the start time of an activity $\sum_t (t - p_{j'}) x_{\gamma j' t}$ is greater than or is equal to the finish time of its predecessor(s) $\sum_t t x_{\gamma jt}$. The Big-M methodology has been adopted in the formulation of this constraint, in conjunction with the decision variable π_j, where M refers to a large number, used to ensure that the inequality preserves its integrity for multi-mode and traveled activities. Constraint sets (3.16) and (3.17) ensure that only a subset of activities that have not had their predecessors completed are able to travel, and are formulated for multi-mode and single-mode activities, respectively.

$$\sum_t x_{\gamma=1,j,t} + \pi_j = 1 \qquad \forall \;\; j = 1, \dots, N \qquad (3.14)$$

$$\sum_t (t - p_j) x_{\gamma=1,j,t} + M\pi_j \geq \sum_t t x_{\gamma=1,j',t} + L_{jj'} \qquad \forall \;\; j, j' \in \{ \alpha \cup \beta \;\; : \; P_{jj'} = 1 \} \qquad (3.15)$$

$$\sum_t x_{\gamma=1,j,t} + \sum_t x_{\gamma=1,j',t} \geq \sum_t x_{\gamma=1,j'',t} \qquad \forall \;\; j, j' \in \beta, \;\; j'' = \alpha \cup \beta \;\; : \; P_{jj''} = 1, \;\; M_{jj'} = 1 \qquad (3.16)$$

$$\sum_t x_{\gamma=1,j,t} \geq \sum_t x_{\gamma=1,j',t} \qquad \forall \;\; j, j' \in \{ \alpha \;\; : \; P_{jj'} = 1 \} \qquad (3.17)$$

To ensure successful fulfillment of restrictions on the earliest and the latest start and finish times, constraint sets (3.18) through (3.21) are formulated. Constraint sets (3.18) and (3.19) enforce the activities to start after the imposed earliest start time ES_j and prior to the latest start time S_j , respectively. Restrictions on finish times are formulated through constraint sets (3.20) and (3.21), to ensure that activities finish after the mandated earliest finish times EF_j and prior to the latest finish times LF_j, respectively.

$$\textstyle\sum_t (t - p_j) x_{\gamma=1,j,t} \geq ES_j \textstyle\sum_t x_{\gamma=1,j,t} \qquad \forall \quad j = 1, \dots, N \tag{3.18}$$

$$\sum_t (t - p_j) x_{\gamma=1,j,t} \leq LS_j \sum_t x_{\gamma=1,j,t} \qquad \forall \quad j = 1, \dots, N \tag{3.19}$$

$$\sum_t t x_{\gamma=1,j,t} \geq EF_j \sum_t x_{\gamma=1,j,t} \qquad \forall \quad j = 1, \dots, N \tag{3.20}$$

$$\sum_t t x_{\gamma=1,j,t} \leq LF_j \sum_t x_{\gamma=1,j,t} \qquad \forall \quad j = 1, \dots, N \tag{3.21}$$

To ensure that the resultant schedule does not exceed the imposed resource budget B, a decision variable W_l is defined, representing the maximum quantity of the required resources from classification l. The value of this non-basic decision variable is obtained through constraint set (3.22), where w_{jl} represent the required number of resources, of classification l for activity j, and $\sum_{u=t}^{t+p_j-1} x_{\gamma ju}$ is a binary variable indicating whether activity j of product γ is in progress at time interval $u = [t-1, t]$. This constraint is imposed to determine the required number of resources per classification l at each time interval $u = [t-1, t]$.

$$\sum_\gamma \sum_j w_{jl} \sum_{u=t}^{t+p_j-1} x_{\gamma ju} \leq W_l \qquad \begin{array}{l} \forall \quad l = 1, \dots, L \\ \forall \quad t = S_{\Upsilon/2+1}, \dots, (S_{\Upsilon/2+1} + D_\Phi) \end{array} \tag{3.22}$$

Once the required number of resources per classification W_l have been established, a resource constraint is imposed to ensure that the required number of resources at time t does not surpass the budgeted quantity of resources. Constraint (3.23) is recommended to be used in scenarios where the number of resources utilized $\sum_l W_l$ cannot exceed the budget B, as in the case of objective function (3.06), allowing versatility for the work center to optimize the required number of resources per classification l. On the contrary, constraint set (3.24) is applied where distinct budgets W_l^{max} are imposed for each resource classification l.

$$\sum_l W_l \leq B \tag{3.23}$$

$$W_l \leq W_l^{max} \qquad \forall \quad l = 1, \dots, L \tag{3.24}$$

Constraints (3.23) and (3.24) are explicitly formulated for modeling of resource constraints in scenarios where travel work is permitted. However, in scenarios where travel work is not allowed, constraints (3.23) and (3.24) are replaced with constraints (3.25) and (3.26), respectively. This modification is due to the inclusion of deviation variables $\delta_k^{\pm}$, following the goal programming model, where δ_k^{+} and δ_k^{-} represent the positive and negative deviation to the imposed aspiration criterion.

$$\sum_{l} W_l + \delta_2^{-} - \delta_2^{+} = B \tag{3.25}$$

$$W_l + \delta_{3l}^{-} - \delta_{3l}^{+} = W_l^{max} \qquad \forall \quad l = 1, \dots, L \tag{3.26}$$

Similar to resource constraints, to impose an upper bound on the available processing times, two new constraints are formulated, differentiated by their capability to permit or prohibit the traveling of work. The strict enforcement of time constraints in scenarios permitting the traveling of incomplete activities is imposed through constraints (3.27), solely used in conjunction with objective function (3.05). Constraint (3.28) is similarly employed, in conjunction with objective functions (3.06) and (3.07), where travel work is prohibited. This is accomplished through the use of deviation variables $\delta_1^{\pm}$, to allow for deviation to the aspiration criterion for time.

$$t \leq \frac{3\Phi - 2}{\Phi} D_\Phi \tag{3.27}$$

$$\sum_{t} x_{\gamma=1,N+1,t} + \delta_1^{-} - \delta_1^{+} = \mathrm{D}_\Phi \tag{3.28}$$

Zones are defined as the physical location of work and are captured as a mean of quantifying the number of resources assigned to each area at time t, subject to a saturated capacity Z_i. Constraint set (3.29) is applied to avoid the oversaturation of zones, where zonal utilization is established through a binary variable $y_{\sigma ji}$, such that $y_{\sigma ji} = 1$, activity j is assigned to zone i of work center σ, and $y_{\sigma ji} = 0$, otherwise.

$$\sum_{l} \sum_{j} w_{jl}\, y_{\sigma=1,j,i} \sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u} \leq Z_i \qquad \begin{array}{l} \forall \quad i = 1, \dots, I \\ \forall \quad t = S_{\Upsilon/2+1}, \dots, (S_{\Upsilon/2+1} + D_\Phi) \end{array} \tag{3.29}$$

In modeling of scenarios prohibiting the simultaneous progression of two or more activities, constraint set (3.30) is imposed. This inequality ensures that such non-concurrencies are respected, where binary variable $NC_{jj'} = 1$ if activities j and j' are non-concurrent.

$$NC_{jj'}\left[\sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u} + \sum_{u=t}^{t+p_{j'}-1} x_{\gamma=1,j',u}\right] \leq 1 \qquad \begin{array}{l}\forall \;\; j,j' \in \{\, \alpha \cap \beta \; : NC_{jj'} = 1\,\} \\ \forall \;\;\; t = S_{\Upsilon/2+1}, \dots, (S_{\Upsilon/2+1} + D_{\Phi})\end{array} \tag{3.30}$$

Additionally, non-concurrencies between two or more zones may be encountered. Constraint set (3.31) is formulated as such, to prevent the simultaneous progression of activities in two or more non-concurrent zones, where binary variable $NC_{ii'} = 1$ if zones i and i' are non-concurrent, and $NC_{ii'} = 0$, otherwise.

$$\begin{aligned} NC_{ii'}\Bigg[\sum_j y_{\sigma=1,j,i} \sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u} \\ + \sum_j y_{\sigma=1,j,i'} \sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u}\Bigg] \leq 1 \end{aligned} \qquad \begin{array}{l}\forall \;\;\; i,i' = 1,\dots, I \;: NC_{ii'} = 1 \\ \forall \;\;\; t = S_{\Upsilon/2+1}, \dots, (S_{\Upsilon/2+1} + D_{\Phi})\end{array} \tag{3.31}$$

Non-concurrencies may also exist between an activity and a set of activities assigned to a zone. Such non-concurrencies are enforced through constraint set (3.32), where binary variable $NC_{ji} = 1$ if activity j and zone i are non-concurrent, and $NC_{ji} = 0$, otherwise.

$$NC_{ji}\left[\sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u} + \sum_j y_{\sigma=1,j,i} \sum_{u=t}^{t+p_j-1} x_{\gamma=1,j,u}\right] \leq 1 \qquad \begin{array}{l}\forall \;\;\; j,i \;:\; NC_{ji} = 1 \\ \forall \;\;\; t = S_{\Upsilon/2+1}, \dots, (S_{\Upsilon/2+1} + D_{\Phi})\end{array} \tag{3.33}$$

Similar to the imposed non-concurrency constraints, concurrency constraints are imposed through constraint sets (3.34) and (3.35). In formulating these inequalities, binary variables $CS_{jj'}$ and $CF_{jj'}$ are used, representing the concurrent start and finish between activities j and j' respectively, where $CS_{jj'} = 1$ if activities j and j' must start concurrently and $CS_{jj'} = 0$, wise.

$$CS_{jj'}\left[\left[\sum_t (t - p_j) x_{\gamma=1,j,t}\right] - \left[\sum_t (t - p_{j'}) x_{\gamma=1,j',t}\right]\right] = 0 \quad \forall \;\;\; j,j' \in \{\alpha \cap \beta \; : CS_{jj'} = 1\,\} \tag{3.34}$$

$$CF_{jj'}\left[\sum_t t x_{\gamma=1,j,t} - \sum_t t x_{\gamma=1,j',t}\right] = 0 \qquad \forall \;\;\; j,j' \in \{\, \alpha \cap \beta \; : CF_{jj'} = 1\,\} \tag{3.35}$$

3.2. Case Study

A real-world study of the assembly operation of narrow-body business jets is conducted to verify and validate the mixed-integer programming models developed for solving complex industrial problems in LVLVPS. The assembly of aircraft is categorized as LVLVPS, where aircraft are processed through a series of work centers, where a diverse set of resources is deployed to complete the pre-defined work package over the span of the available processing time. For the purpose of this study, the scope has been limited to two parallel work centers, dedicated to the assembly cockpit and the empennage to the fuselage. The work center is assigned $N = 52$ activities, five of which are classified as multi-mode activities, and the remaining as single-mode activities. Currently, work centers are budgeted with a dedicated set of resources of three distinct classifications with the objective of completing the pre-defined work package over $t = 48$ hours. Through this study, the prescribed scheduling problem is solved using shared and dedicated resources. Section 3.2.1 provides an overview of this scheduling problem, in addition to quantifying the underlying boundaries and parameters. The problem is solved in Sections 3.2.2 and 3.2.3 using each of the objective functions proposed in Section 3.1.2, differentiated by permitting or prohibiting travel work. The Optimization models outlined in this section were executed on 64-bit Windows with an Intel 6th generation i7 processor, operating at 2.6GHz with a 16.0GB RAM.

3.2.1. Variables and Parameters

The objective of this study is the optimization of a schedule for a statement of work consisting of $N = 52$ activities in $\Phi = 2$ identical and parallel work centers, each budgeted with $L = 3$ classification of resources. Resource profiles are broken down into $W_1^{max} = 4$ type 1 resources, $W_2^{max} = 4$ type 2 resources, and $W_3^{max} = 4$ type 3 resources per work center per shift. Activities are classified as either single-mode α or multi-mode β, and are located in $I = 3$ distinct zones, subject to a maximum capacity of $Z_1 = 6$, $Z_2 = 4$ and $Z_3 = 6$, for zones 1, 2 and 3 respectively. Following Equation (3.01) of Section 3.1.1.2, $\Upsilon = 3$ products are required to be processed in modeling of this scheduling problem. To satisfy the imposed takt-time of $T = 24$ hours, based on Equation (3.02), each work center σ has an available processing time of $D_{\Phi} = 48$ hours, with an overall planning horizon of 96 hours for completion of all products γ with an inter-arrival time of $S_{\gamma} = 24$ hours, as calculated through Equation (3.04) of Section 3.1.1.3. Activities have a pre-defined set of attributes, comprised of independent processing time p_j, resource requirement w_{jl}, zonal assignment $y_{\sigma ji}$, concurrent start and finish $CS_{jj'}, CF_{jj'}$ and non-concurrency $NC_{jj'}$ relationships, in addition to precedence $P_{jj'}$ and lead and lag times $L_{jj'}$. The activity-on-node (AON) network diagram depicted in Figure 3.06, demonstrates the interdependencies between activities, where the lightly shaded blocks represent the secondary modes of multi-mode activities and are positioned directly below their parent block.

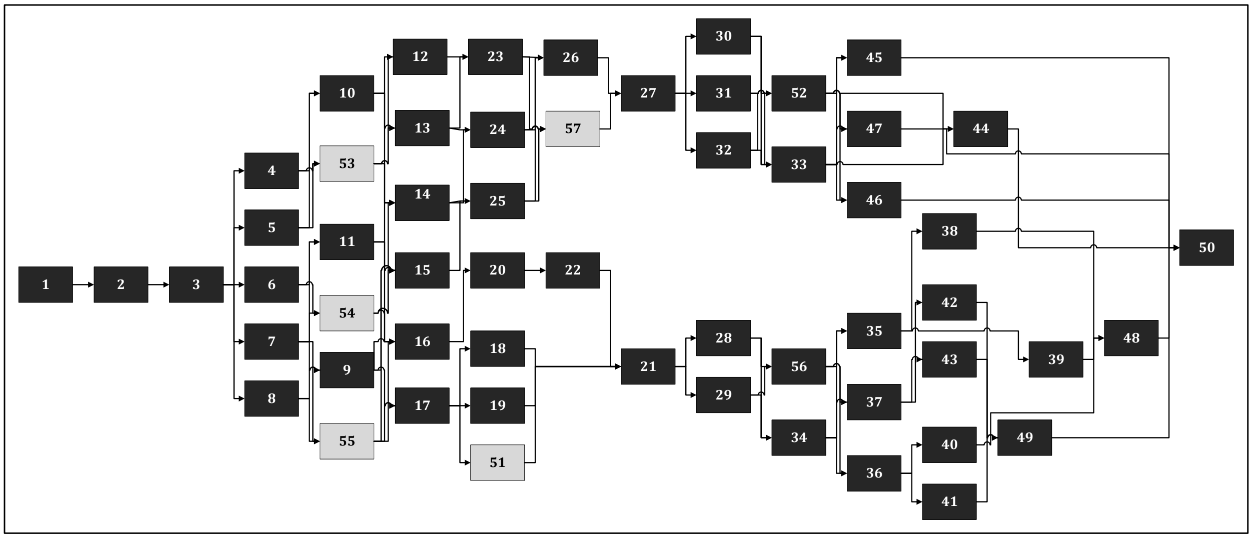

Figure 3.06 - Case Study 2 - Activity-On-Node Network Diagram

3.2.2. Travel Work Permitted

The scheduling problem discussed in Section 3.1 is solved in this section, using objective function (3.05), proposed in Section 3.1.2.1, with the objective of minimizing resource requirements and the number of activities traveled. The objective of this section is to optimize the scheduling problem for the given work center with shared and dedicated resources in Section 3.2.2.1 and 3.2.2.2, respectively. In Section 3.2.2.3, the results obtained from each of the models are compared, and the optimum solution, yielding the minimum number of incomplete activities is highlighted.

3.2.2.1. Shared Resource Problem

In optimizing of this scheduling problem, objective function (3.06) is adopted, along with constraint sets (3.08), (3.09), (3.10), (3.11), (3.14), (3.15), (3.16), (3.17), (3.22), (3.23), (3.24), (3.27), (3.29), (3.30), (3.33), and (3.34) to ensure that precedence, lead time and lag times, resource, time, zonal, and concurrency constraints are successfully satisfied. This problem was programmed in

IBM ILOG CPLEX, and solved in 2,708 seconds, after 19,385,232 iterations resulting in the objective value of $Z = 8$. The objective value represents the traveling of 2 incomplete activities and use of 6 resources. This scheduling problem is subject to 17,662 constraints with 32,895 decision variables.

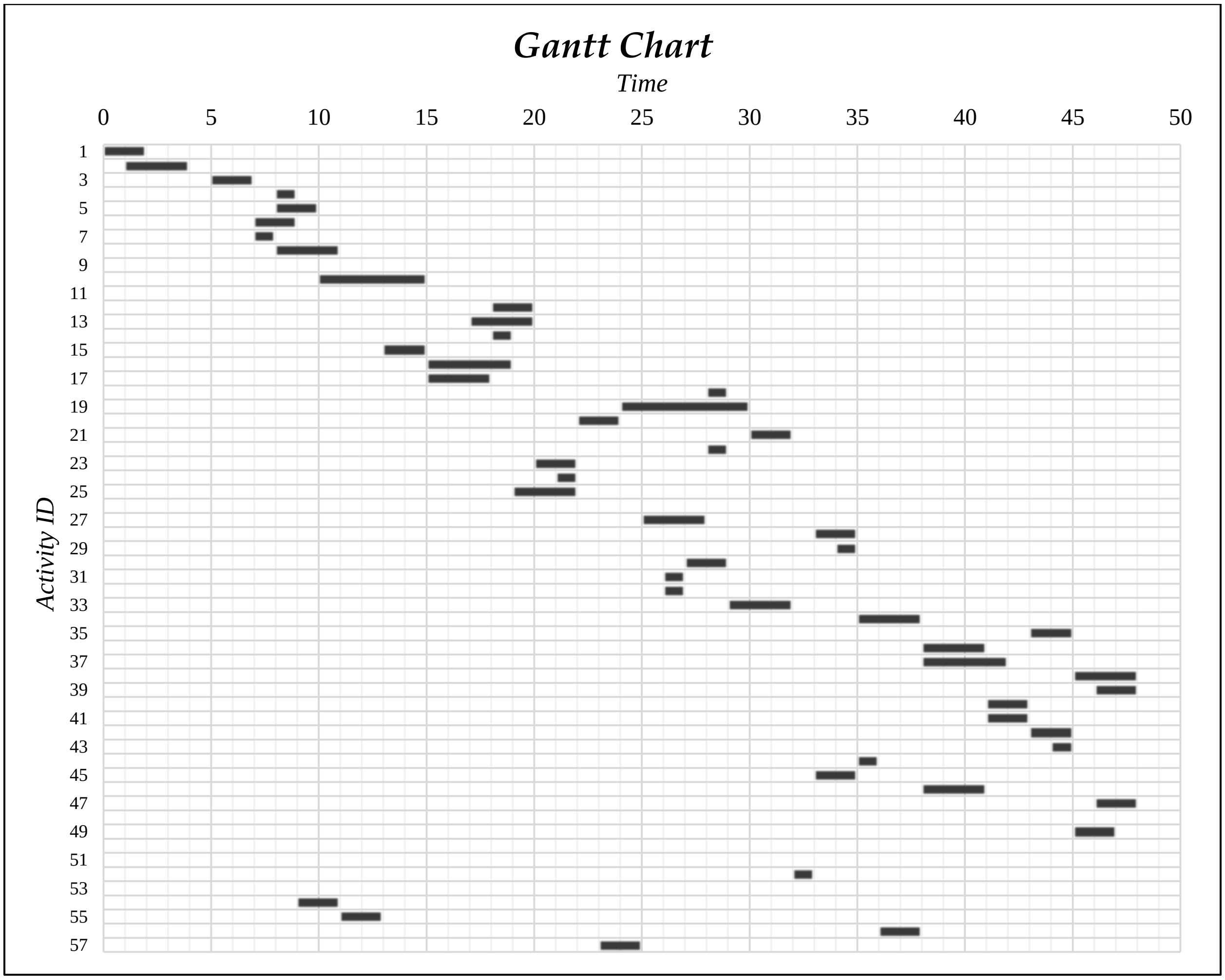

Figure 3.07 - Gantt Chart for the Optimized Work Center Schedule

Gantt chart depicted in Figure 3.07 represents the optimum work center schedule for this problem with shared resources, illustrating the start, duration and finish time of each of the activities. It can be demonstrated through the analysis of activity start and finish times, that all concurrency, non-concurrency and lead and lag time constraints have been successfully satisfied. The resultant schedule yields to an average concurrent progression of 2.29 activities at any time t, as illustrated in Figure 3.08. Figures 3.09 through 3.16 verify that the imposed constraints have also been satisfied. Figure 3.09 quantifies the slack between interdependent activities. Note that slack times that are equal or greater than zero are an indication of the successful fulfillment of predecessor constraints. Negative slacks are observed at four instances on interdependencies 0, 52, 53, and 54, due to lead time allowances between activities 1 and 2, 27 and 30, 27 and 31, and 27 and 32. Figures 3.10 through 3.13 illustrates resource utilization and availability through time for the sum of all classifications, type 1 resources, type 2 resources, and type 3 resources, respectively. It is concluded through these figures that all applicable resource constraints are satisfied, where no positive deviation to resource constraints is incurred in the resultant schedule. Similarly, Figures 3.14 through 3.16 validate that all zonal constraints pertaining to zones 1, 2 and 3 have been satisfied, and oversaturated zones are not observed at any time t.

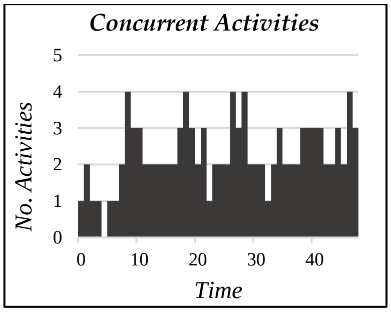

Figure 3.08 - Concurrent Activities

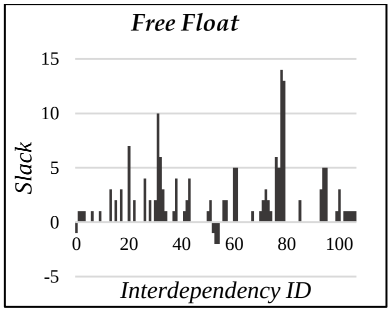

Figure 3.09 - Free Float

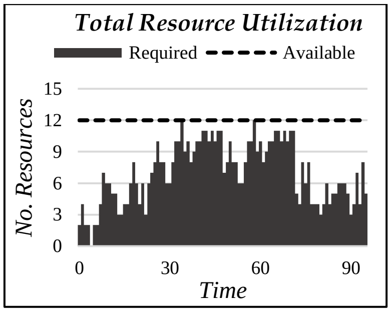

Figure 3.10 - Resource Utilization

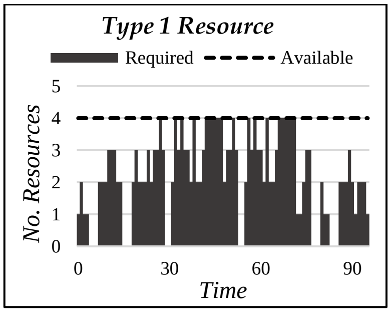

Figure 3.11 - Resource 1 Utilization

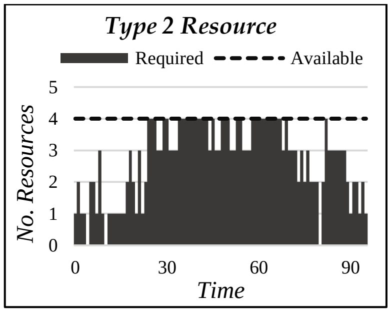

Figure 3.12 - Resource 2 Utilization

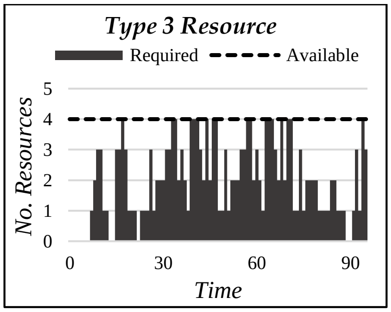

Figure 3.13 - Resource 3 Utilization

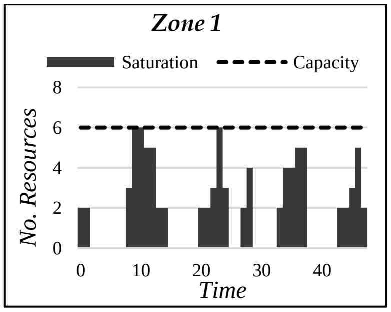

Figure 3.14 - Zone 1 Utilization

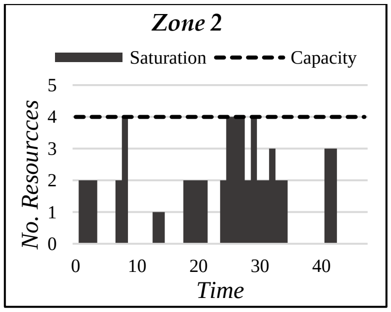

Figure 3.15 - Zone 2 Utilization

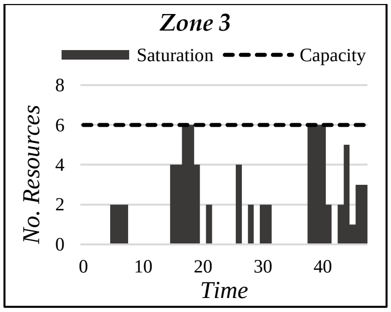

Figure 3.16 - Zone 3 Utilization

Further analysis of the utilization of resources yields to an average utilization of 55.5% for type 1 resources, 68.8% utilization for type 2 resources, and 50.8% utilization for type 3 resources, with an overall average utilization of 58.33% for all classifications, details of which are depicted in Table 3.04.

No. Resources	*Type 1 Utilization*	*Type 2 Utilization*	*Type 3 Utilization*
0	16.7%	3.1%	12.5%
1	9.4%	16.7%	28.1%
2	30.2%	16.7%	21.9%
3	22.9%	29.2%	18.8%
4	20.8%	34.4%	18.8%
Average Utilization	55.5%	68.8%	50.8%

Table 3.04 - Resource Utilization – Shared Resources Permitting Travel Work

It can thus be concluded through this section that the developed mixed-integer programming model is effective in optimizing parallel work center scheduling problems with shared resources subject to limited time and resources.

3.2.2.2. Dedicated Resource Problem

The proposed mixed-integer programming models are applied to solve an identical scheduling problem but with dedicated resources, aimed at examining the models' capability and reliability in modeling single work center scheduling problems or multi-parallel scheduling problems with dedicated resources. This is accomplished through a modification to the variables and parameters set in Section 3.2.1 by altering the total number of work centers to one $\Phi = 1$ and processing a single product $\Upsilon = 1$. The upper bound on time and jobs is then re-calculated through Equation (3.03) with $D_{\Phi} = 48$ and $t \in [1,48]$. The available number of resources of each classification has also been divided by two, to provide each of the two work centers with a dedicated set of resources, yielding to $W_1^{max} = 2$, $W_2^{max} = 2$, and $W_3^{max} = 2$, type 1, 2, and 3 resources per work center per shift, respectively. Similar to the previous model, this problem is solved using objective function (3.06) to minimize the required resources and the number of traveled activities, subject to constraint sets (3.10), (3.11), (3.14), (3.15), (3.16), (3.17), (3.22), (3.23), (3.24), (3.27), (3.29), (3.30), (3.33), and (3.34). The differentiation between this model and that of Section 3.2.2.1 is the omission of constraints sets (3.08) and (3.09), as this problem only requires a single product to be processed, and as such, the standardization between parallel work centers is no longer applicable. This problem was modeled and solved in CPLEX in 15.3 seconds, after

91,549 iterations, resulting in an objective value of $Z = 29$, yielding to the traveling of 23 incomplete activities and the use of 6 resources. This problem is subject to 3,559 constraints with 5,535 variables. The optimized schedule obtained from solving this problem is depicted in the Gantt chart presented in Figure 3.17, demonstrating the successful fulfillment of precedence, lead and lag times, and concurrency constraints.

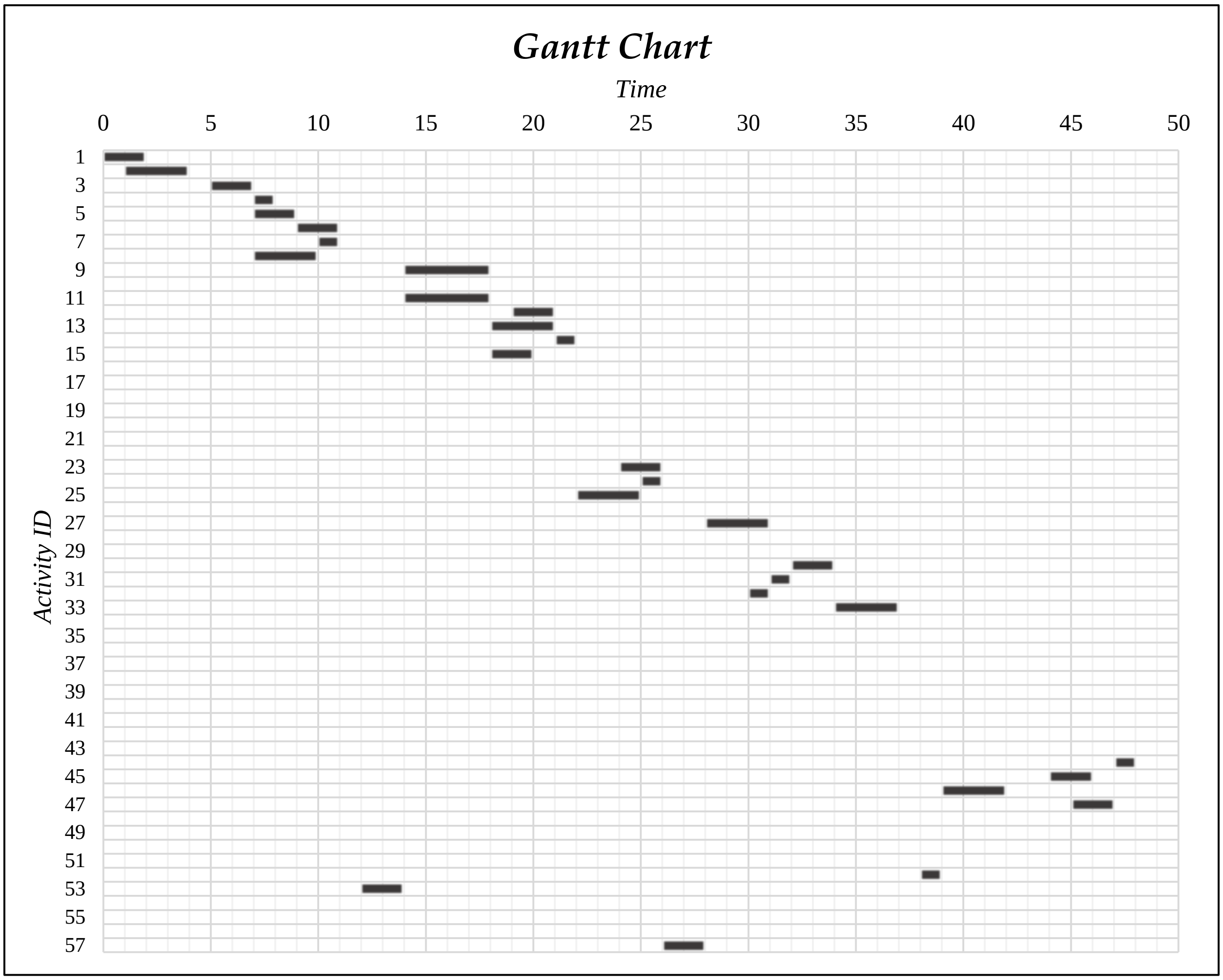

Figure 3.17 - Gantt Chart for the Optimized Work Center Schedule

An identical set of figures as the one presented in Section 3.2.2.1 has been generated to evaluate the resultant's schedule's feasibility. Figures 3.18 through 3.26 demonstrate the number of concurrent activities and slack times between activities, in addition to resource utilization and zonal saturation obtained from the resultant schedule. Figure 3.18 illustrates the number of concurrent activities, yielding an average of 1.27 concurrent activities. It can be established through Figure 3.19 that precedence, and lead and lag constraints are satisfied through an analysis of the slack times between interdependent activities. Figures 3.20 through 3.23 illustrate resource availability and utilization throughout the planning horizon for the sum of all resources, and type 1, type 2, and type 3 resources, respectively, confirming the successful fulfillment of the imposed resource constraints. Figures 3.24 through 3.26 are similarly generated, demonstrating that the resultant zonal utilization levels are less than or are equal to the imposed zonal capacities, thus satisfying all applicable zonal constraints.

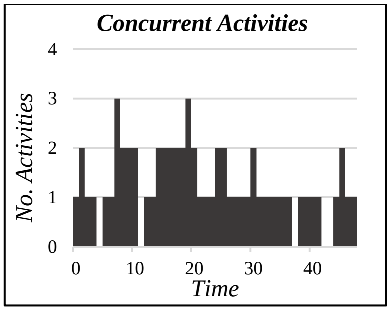

Figure 3.18 - Concurrent Activities

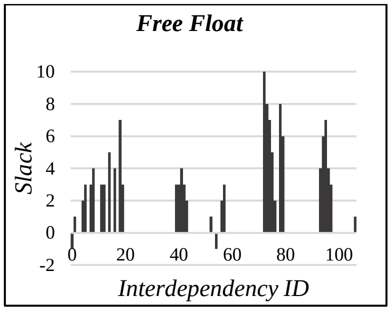

Figure 3.19 - Free Float

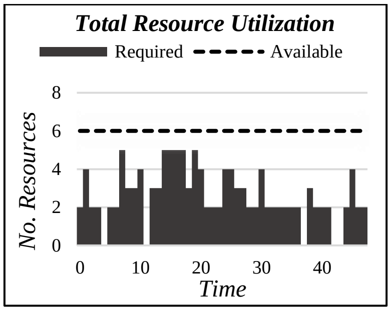

Figure 3.20 - Total Resource Usage

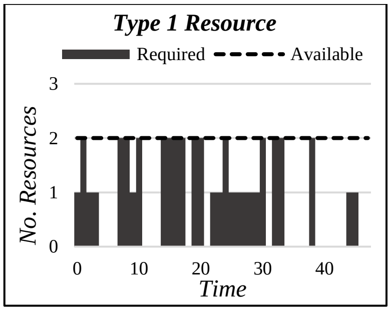

Figure 3.21 - Resource 1 Utilization

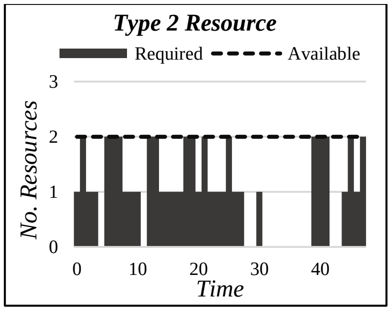

Figure 3.22 - Resource 2 Utilization

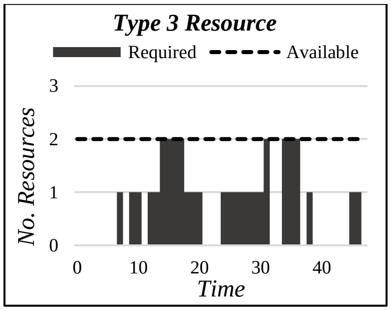

Figure 3.23 - Resource 3 Utilization

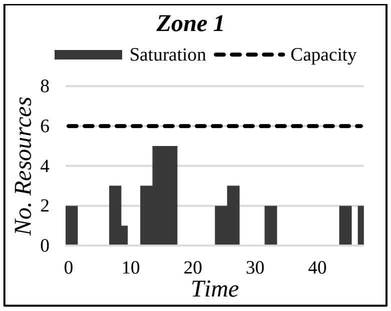

Figure 3.24 - Zone 1 Utilization

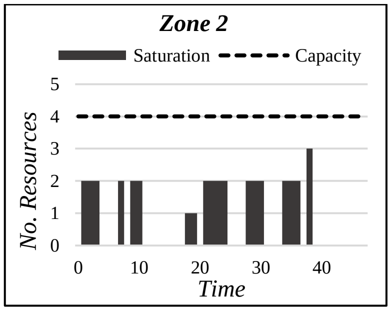

Figure 3.25 - Zone 2 Utilization

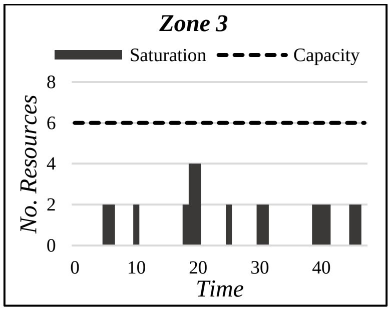

Figure 3.26 - Zone 3 Utilization

The analysis of resource usage through time yields to an average utilization of 44.79%, 51.04%, and 35.42% for type 1, 2 and 3 resources, respectively, with an overall average utilization of 43.8% for all classifications of resources, details of which are summarized in Table 3.05.

No. Resources	*Type 1 Utilization*	*Type 2 Utilization*	*Type 3 Utilization*
0	41.7%	29.2%	45.8%
1	27.1%	39.6%	37.5%
2	31.2%	31.2%	16.7%
Average Utilization	44.8%	51.0%	35.4%

Table 3.05 - Resource Utilization – Dedicated Resources Permitting Travel Work

It can thus be concluded that the proposed mixed-integer programming model is effective in optimizing single or parallel work center scheduling problems with dedicated resources.

3.2.2.3. Comparison of Results and Managerial Insights

The results from Sections 3.2.2.1 and 3.2.2.2 are compared and analyzed in an effort to select the optimum resource allocation profile. It can be concluded from the data summarized in Table 3.06, that while the shared resource problem is significantly larger than the dedicated resource problem with 5.94 times more variables, taking 211.75 times more time, it yields to a schedule that completes 72.4% more activities while increasing the overall resource utilization by additional 33.3%.

Evaluation Metrics	*Sec. 3.2.2.1* *Shared Resources*	*Sec. 3.2.2.2* *Dedicated Resources*	*3.2.2.1 Vs. 3.2.2.2* *Improvement*
Number of Variables	32,895	5,535	- 594%
Number of Constraints	17,662	3,559	- 496%
Number of Iterations	19,385,232	91,549	- 21175%
Processing Time (CPLEX)	2,708	15	- 17699%
Number of Complete Activities	50	29	+ 72.4%
Percentage of Activities Completed	96%	56%	
Type 1 Resource Availability (2 Work Centers)	4	4	-
Type 2 Resource Availability (2 Work Centers)	4	4	-
Type 3 Resource Availability (2 Work Centers)	4	4	-
Average Concurrent Activities	2.29	1.27	+ 80.3%
Average Slack Between Interdependent Activities	1.88	2.78	+ 47.0%
Type 1 Resource Utilization	55.5%	44.8%	+ 23.9%
Type 2 Resource Utilization	68.8%	51.0%	+ 34.8%
Type 3 Resource Utilization	50.8%	35.4%	+ 43.4%
Overall Resource Utilization	58.3%	43.8%	+ 33.3%

Table 3.06 - Travel Work Permitted - Comparison of Results

3.2.3. Travel Work Prohibited

The problem discussed in Section 3.2.1 is solved in this section, with objective function (3.07), enforcing the in-station completion of the assigned work package, through a priority-based goal programming model. This objective of this problem is to complete all activities while minimizing the positive deviation to the aspiring time and resource criteria. The proposed multi-objective goal programming model is employed in solving the prescribed scheduling problem, to ensure its compatibility and reliability in modeling large-scale industrial problems with shared or dedicated resources.

3.2.3.1. Shared Resource Problem

In mathematical modeling of this problem, objective function (33) is used in conjunction with constraint sets (3.08), (3.09), (3.12), (3.13), (3.14), (3.15), (3.16), (3.17), (3.22), (3.25), (3.26), (3.27), (3.28), (3.29), (3.33), and (3.34) that ensure time and resource constraints, in addition to precedence, lead and lag times, zonal, and concurrency constraints, while enforcing the scheduling of all activities. In solving this problem using a goal programming approach, constraint sets (3.25) (3.26) and (3.28) are used in place of constraint sets (3.23), (3.24), and (3.27), in the modeling of resource and time constraints, respectively. This problem was programmed and solved in CPLEX in 1,203 seconds, after 3,544,787 iterations, and is subject to 17,663 constraints with 32,895 variables, yielding to the objective value of $Z = 1$, equivalent to the use of 13 resources. The optimum schedule results in no positive deviation to time constraints, signifying that activities assigned to the two parallel work centers with shared resources can be completed on time, with one additional type 2 resource. The optimum schedule for this problem is presented in Figure 3.27 in the form of a Gantt chart.

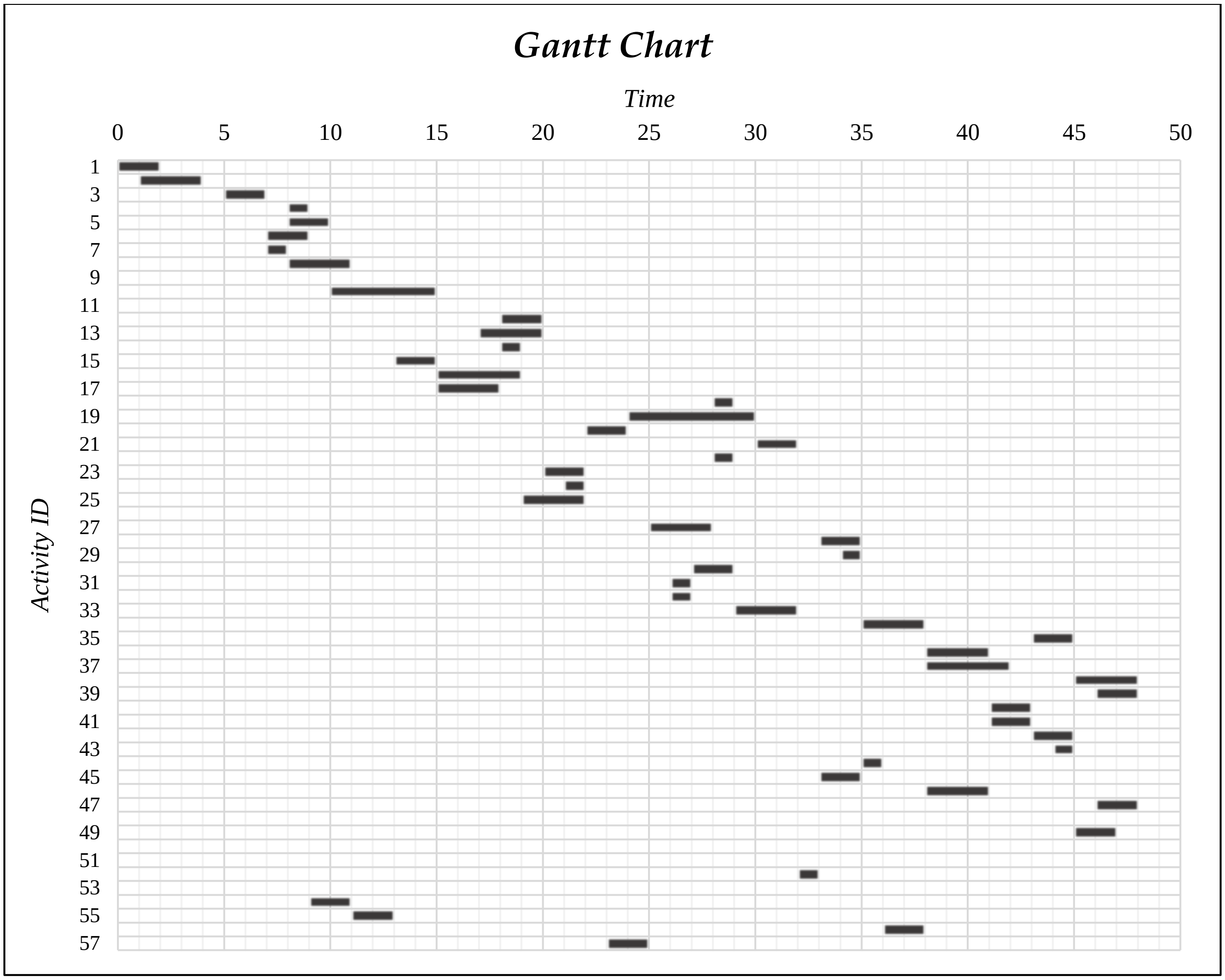

Figure 3.27 - Gantt Chart for the Optimized Work Center Schedule

Figures 3.28 through 3.36 illustrate the successful satisfaction of all underlying constraints, where Figure 3.28 illustrates an average of 2.21 concurrent activities are active. Through Figure 3.29 it can be established that all interdependencies are exhibiting a slack time that are equal or are greater than zero, concluding that the imposed precedence constraints have been satisfied, with the exception of where lead times are allowed. Figure 3.30 through 3.33 show total resource in addition to type 1, type 2 and type 3 resource requirements through time, demonstrating that while type 1 and type 3 resource constraints have been satisfied, a positive deviation of $\delta^{+}_{3,2} = 1$ is incurred for type 2 resources. Moreover, Figures 3.34 through 3.36 verify that all zonal constraints for zones 1, 2 and 3 have been satisfied, where no zone is over-saturated at any time t.

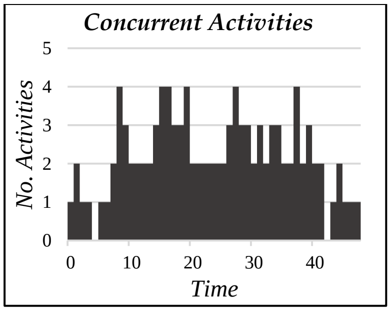

Figure 3.28 - Concurrent Activities

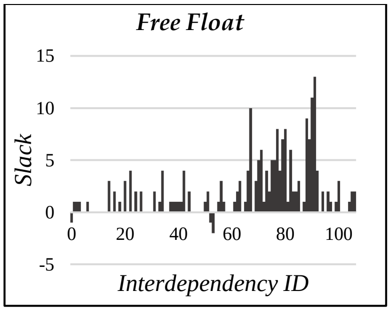

Figure 3.29 - Free Float

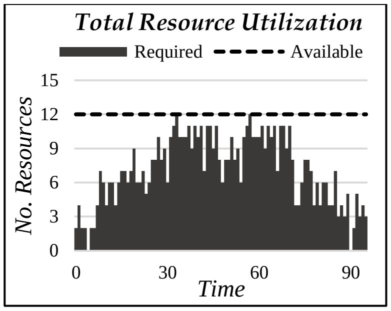

Figure 3.30 - Resource Utilization

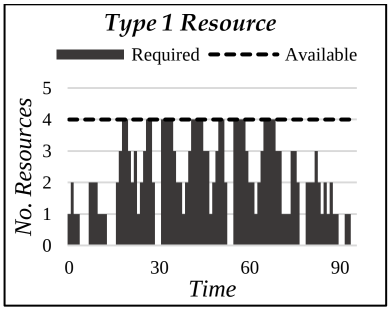

Figure 3.31 - Resource 1 Utilization

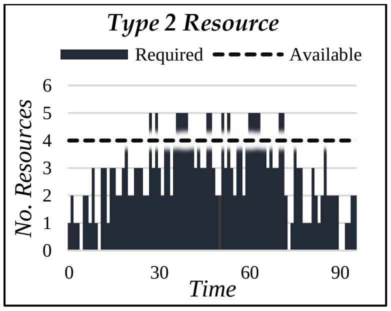

Figure 3.32 - Resource 2 Utilization

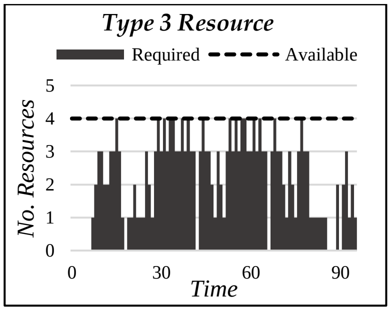

Figure 3.33 - Resource 3 Utilization

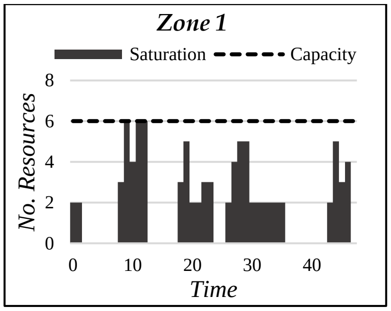

Figure 3.34 - Zone 1 Utilization

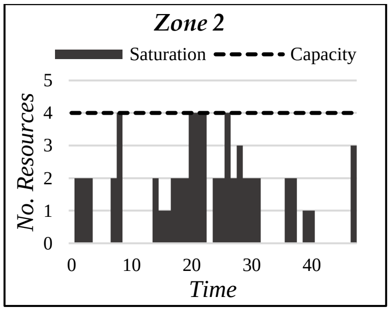

Figure 3.35 - Zone 2 Utilization

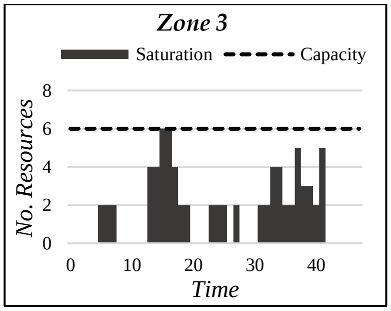

Figure 3.36 - Zone 3 Utilization

Table 3.07 summarizes utilization levels for each classification of resources obtained from the resultant schedule, yielding to an average utilization of 52.3%, 56.3%, and 54.7% for type 1, type 2 and type 3 resources respectively, with an overall average utilization of 59.1% for all resource classifications.

No. Resources	*Type 1 Utilization*	*Type 2 Utilization*	*Type 3 Utilization*
0	16.7%	5.2%	14.6%
1	19.8%	13.5%	20.8%
2	24.0%	25.0%	12.5%
3	16.7%	24.0%	35.4%
4	22.9%	15.6%	16.7%
5	-	16.7%	-
Average Utilization	52.3%	56.3%	54.7%

Table 3.07 - Resource Utilization – Shared Resources Prohibiting Travel Work

Through an analysis of the results, it is established that overall resource utilization has decreased by 3.7% through a positive deviation to the budgeted type 2 resources yielding to the in-station completion of allocated activities. It is thus concluded that the proposed mixed-integer programming model is effective for solving multi-parallel work center scheduling problems with shared resources.

3.2.3.2. Dedicated Resource Problem

In solving single or multi-parallel work center scheduling problems with dedicated resources, a goal programming methodology, similar to that of Section 3.2.3.1 is employed. In formulation of this problem, constraint sets (3.12) (3.13) (3.14) (3.15) (3.16) (3.17) (3.22) (3.25) (3.26) (3.28) (3.29) (3.30) (3.33) and (3.34) are used, enforcing identical constraints as those outlined in Section 3.2.3.1 but differentiated through the omission of constraint sets (3.08) and (3.09), aimed at generating a standardized schedule. This problem was solved in CPLEX in 9.6 seconds, after 275,943 iterations, and is subject to 3,559 constraints, with 5,535 variables, resulting in an objective value of $Z = 2$, equivalent to the use of 8 resources, satisfying the imposed time and resource budgets. The optimum solution for this problem is presented in Figure 3.37 through a Gantt chart.

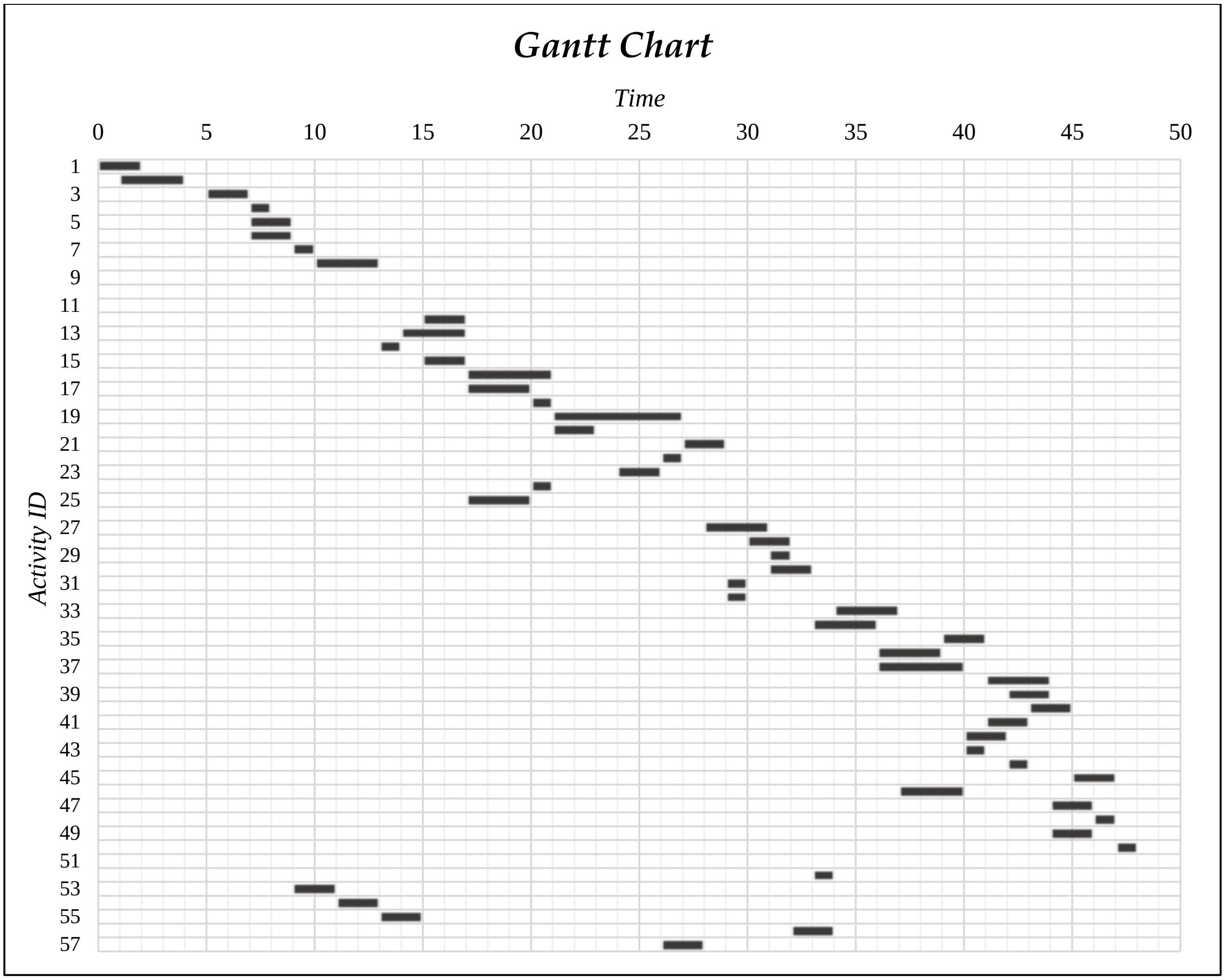

Figure 3.37 - Gantt Chart for the Optimized Work Center Schedule

Figures 3.38 through 3.46 are used to evaluate the feasibility of the resultant schedule. Figure 3.38 demonstrates the quantity of simultaneously progressed activities for each unit of time, with an average of 2.27 activities. Figure 3.39 demonstrates the slack times between interdependent activities, indicating that precedence constraints have been successfully satisfied as no unanticipated positive deviations are observed. Figures 3.40 through 3.43 illustrate resource utilization and availability for all classifications and type 1, type 2, and type 3 resources, respectively, where a positive deviation to resource availability is observed for type 2 and type 3 resources, signifying that an additional unit for each of these resource classifications are required in timely completion of the work package. Figures 3.44 through 3.46 validate the successful fulfillment of the imposed zonal constraints, verifying that no zone is over-saturated at time *t*.

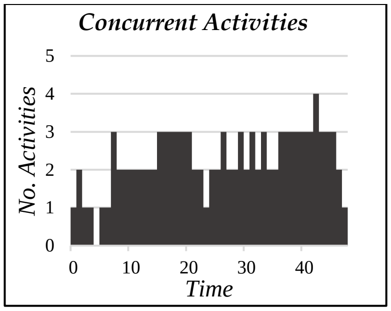

Figure 3.38 - Concurrent Activities

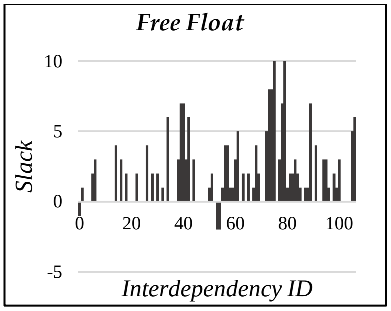

Figure 3.39 - Free Float

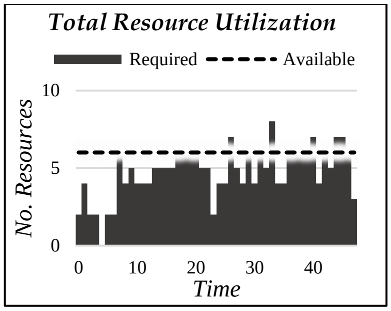

Figure 3.40 - Resource Utilization

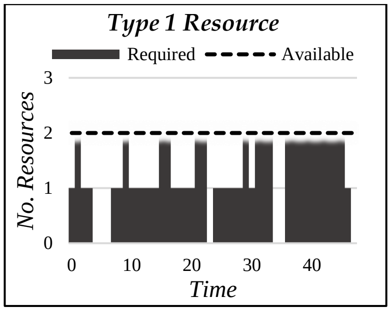

Figure 3.41 - Resource 1 Utilization

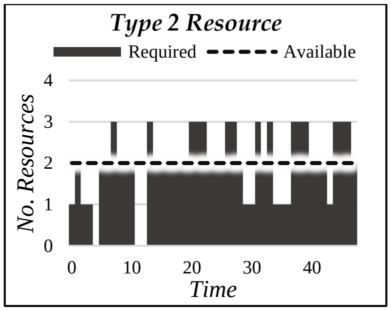

Figure 3.42 - Resource 2 Utilization

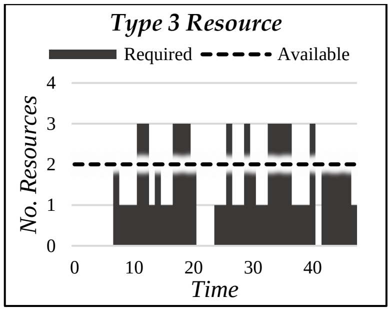

Figure 3.43 - Resource 3 Utilization

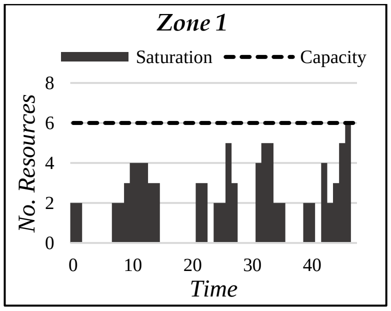

Figure 3.44 - Zone 1 Utilization

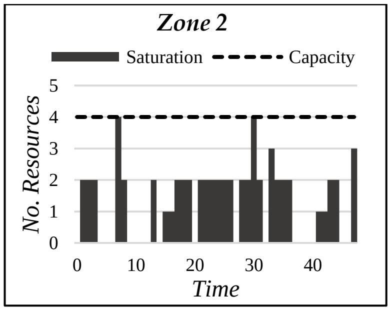

Figure 3.45 - Zone 2 Utilization

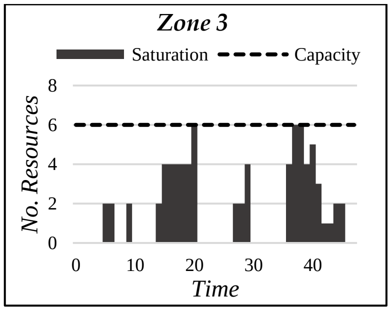

Figure 3.46 - Zone 3 Utilization

The analysis of the resultant optimum solution yields an overall average utilization of 59.1% for all classifications of resources, where type 1, type 2, and type 3 resources each have an average utilization of 63.5%, 53.1%, and 35.4% respectively, details of which are depicted in Table 3.08.

No. Resources	***Type 1 Utilization***	***Type 2 Utilization***	***Type 3 Utilization***
0	14.6%	6.3%	22.9%
1	43.8%	18.8%	33.3%
2	41.7%	43.8%	18.8%
3	-	31.3%	25.0%
Average Utilization	63.5%	53.1%	35.4%

Table 3.08 - Resource Utilization – Dedicated Resources Prohibiting Travel Work

3.2.3.3. Comparison of Results and Managerial Insights

Through Section 3.2.3.1 and 3.2.3.2 of this case study, it is established that the proposed goal programming model is a novel approach for solving multi-parallel work center scheduling problems with shared or dedicated resources, yielding to the optimum solution, where travel work is prohibited. The results obtained in Sections 3.2.3.1 and 3.2.3.2 are analyzed and compared to provide managerial insight into a decision regarding the use of shared or dedicated resources. Evaluation criteria and their corresponding metrics are summarized in Table 3.09 differentiated by the use of dedicated or shared resources. It can be concluded that although solving this problem with dedicated resources takes over 125 times less time, with 5.94 times fewer variables and 4.96 times fewer constraints, the results obtained from the use of shared resources are far more favorable. In completion of all activities assigned to the two parallel work centers with dedicated resources, 16 resources are required for both work centers, whereas only 13 resources are required if resources are shared between the two work centers, resulting in a cost-saving of 18.75% in labor, equivalent to 3 full-time resources per shift.

Evaluation Metrics	***Sec. 3.2.3.1*** ***Shared Resources***	***Sec. 3.2.3.2*** ***Dedicated Resources***	***3.2.3.1 Vs. 3.2.3.2*** ***Improvement***
Number of Variables	32,895	5,535	- 594%
Number of Constraints	17,663	3,559	- 496%
Number of Iterations	3,554,787	275,943	- 1288%
Processing Time (Seconds)	1,203	10	- 12531%
Percentage of Activities Completed	100%	100%	-
Type 1 Resource Requirement (2 Work Centers)	4	4	0%
Type 2 Resource Requirement (2 Work Centers)	5	6	+ 16.7%
Type 3 Resource Requirement (2 Work Centers)	4	6	+ 33.3%
Average Concurrent Activities	2.21	2.27	- 2.6%
Average Slack Between Interdependent Activities	2.27	2.19	+ 47.0%
Type 1 Resource Utilization	52.3%	63.50%	- 17.6%
Type 2 Resource Utilization	56.3%	53.10%	+ 6.0%
Type 3 Resource Utilization	54.7%	35.40%	+ 54.5%
Overall Resource Utilization	54.6%	59.1%	- 7.6%

Table 3.09 - Travel Work Prohibited - Comparison of Results

3.3. Chapter Summary

Despite the prominent scholarly advancements in scheduling optimization methodologies, limited research is reported on mixed-integer programming models for solving scheduling problems in LVLVPS, with a gap in capturing multi-parallel work center scheduling problems with shared resources. The integration of identical parallel work centers is an effective and common practice in the shifting of bottlenecks and fulfillment of demand rates otherwise found infeasible. In the assignment of resources to multi-parallel work centers, two distinct resource profiles can be imposed. Commonly, a dedicated set of resources are assigned to each work center; however, there exist scenarios where a shared pool of resources is assigned to multiple parallel work centers. Through this chapter, the implications involved in the integration of identical parallel work centers are examined, and a mixed-integer programming model is developed and proposed, adaptable to single or multi-parallel work center scheduling problems with shared or dedicated resources. The optimum schedule and execution sequence for such scheduling problems is obtained through minimizing resource requirements and the number of incomplete activities in scenarios where travel work is permitted. As demonstrated through the case study of an aircraft assembly line, it is concluded that while the multi-parallel work center scheduling problems with shared resources is significantly more complex than single or multi-parallel work center scheduling problems with dedicated resources, it can result in substantial labor savings, thus, directly improving an organization's bottom line. The proposed optimization model is thus recommended to be used in large-scale in place of manually created activity schedules or those created by trivial scheduling software incapable of capturing many of the characteristics and constraints inherent in LVLVPS. For this purpose, a new series of mixed-integer programming models are proposed, tailored to single and multi-parallel work center scheduling problems with shared or dedicated resources. The formulated multi-objective mixed-integer linear mathematical programming models capture characteristics and features inherent in LVLVPS, and result in an optimum solution, yielding the minimum number of resources while minimizing deviation to the imposed aspiration criteria. The proposed mixed-integer programming models are validated through a detailed real-world study of a narrow-body aircraft assembly line, demonstrating the applicability of the proposed models for solving complex and large-scale industrial problems.

CHAPTER 4

CONTINUOUS-TIME SINGLE WORK CENTER

This chapter is dedicated to the formulation of a novel suite of multi-objective mathematical programming models for modeling and solving continuous-time single work center scheduling problems in LVLVPS. The mathematical models and metaheuristics proposed in this chapter are submitted and under review with the Journal of Computers & Industrial Engineering. As highlighted in Chapters 3 and 4, work centers in LVLVPS are budgeted with L classifications of multi-skilled human resources, where distinct W_l^{Max} budgets for each resource classification l is imposed. The work center is responsible to complete the assigned work package, comprised of N single-mode α, and M multi-mode β activities over the span of the imposed takt time T_{Max}. Secondary modes j' of multi-mode activities $j \in \beta$ are represented as new dummy activities with identical attributes to that of their origin and are identified through the binary parameter $M_{jj'}$, where $M_{jj'} = 1$ if $j'$is a secondary mode of activity j, and is equal to zero otherwise. Activities have continuous (non-integer) processing times p_j, are assumed to be non-preemptive and may require multiple resources of distinct classifications, where resource requirements for activity j

are denoted by w_{jl}. Due to the complex nature of the products assembled in LVLVPS, there exist complex interdependencies between predecessor j and successor activities j', with lead and lag times $L_{jj'}$. Lead time in the context of scheduling is applicable in scenarios where a successor activity j' can start prior to the completion of its predecessors j', while lag time refers to an imposed delay between successor and predecessor(s). To ensure accurate modeling of characteristics inherent in LVLVPS, a zonal constraint is imposed, where zones $i \in \{1, \dots, I\}$ represents the physical location of work, subject to a maximum capacity Z_i, representing the maximum number of resources $\sum_j \sum_l w_{jl}$ that can work in a zone concurrently. Activities are assigned to zones through the binary parameter y_{ij}, where $y_{ij} = 1$ if activity j is assigned to zone i, and $y_{ij} = 0$ if otherwise. There also exists concurrency and non-concurrency constraints $NC_{jj'}$, between two or more activities, restricting or mandating their simultaneous progression. Concurrency constraints are classified as either concurrent start $CS_{jj'} = 1$ or concurrent finish $CF_{jj'} = 1$. Each activity j may also be subject to earliest start ES_j or latest start time LS_j, as well as earliest or latest finish times, denoted by EF_j and LF_j, respectively. Table 4.01 provides a summary of activity attributes and constraints inherent in LVLVPS, as well as assumptions made in formulating the proposed heuristics and mathematical programming models.

Activity Attributes	*Assumptions*	*Constraints*
• Processing Time • Interdependencies • Lead & Lag Times • Single or Multi-Mode • Resource Classification • Resource Quantity • Zonal Assignment • Earliest Start & Finish • Latest Start & Finish	• Continuous Processing Time • Continuous Planning Horizon • Pre-emption is Prohibited • Equal Resource load per Shift • Equal Resource Efficiency • Activities May Travel[1]	• Resource Constraints • Time Constraints • Precedence Constraints • Lead & Lag Time Constraints • Non-Concurrency Constraints • Concurrency Constraints • Earliest Start & Finish Times • Latest Start & Finish Times • Scheduling of All Jobs[2]

Table 4.01 - Assumptions for Continuous-Time Work Center Scheduling Problems

The high-level production layout depicted in Figure 4.01 demonstrates the product flow of an aircraft through work centers in an assembly line. The final assembly of aircraft is considered as an LVLVPS, with strict enforcement of time and resource constraints. The scarcity of certified skilled resources imposes an upper bound on the available number of resources, while the

[1] This assumption is only valid in scenarios permitting the traveling of incomplete activities.
[2] This constraint is imposed in scenarios where travel work is prohibited.

moving nature of the assembly line requires an upper bound on time. It is thus crucial to formulate an optimization model that satisfies the strict enforcement of time and resource constraints, which may only be possible at the cost of travel work. Travel work refers to the activities that had traveled to downstream work centers as a result of not being completed on-time. On the contrary, there exist scenarios where a milestone must be met within a work center to ensure the safe movement of the product. An example of which (Figure 4.01) is the assembly of the wings in the first work center, the final assembly of the wing assembly to the center fuselage in the second work center, and the installation of the cockpit and the empennage in the third work center. Such milestones ensure the structural integrity of the product as it flows through the work centers. It is thus critical to formulate an additional optimization model that enforces the successful completion of the work package in its entirety, which may only be possible through deviation to the aspiration criteria to time and resources.

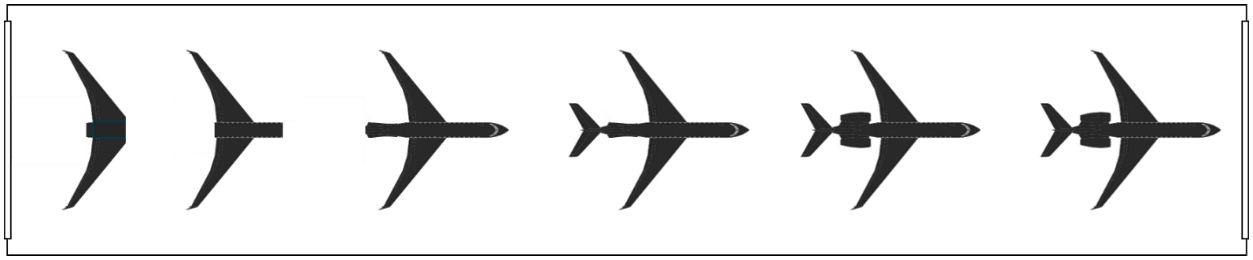

Figure 4.01 - Sample production layout

4.1. Mathematical Programming Models

4.1.1. Variables, Parameters, and Sets

The sets, parameters, and variables defined for formulating the proposed optimization models are appended in Table 4.02, where the primary decision variable is the starting time S_j of activity j. In scenarios where travel work is permitted, there exists a hard constraint on the available planning horizon. To identify the incomplete activities, a binary variable x_j is introduced, representing the on-time completion of activities, where $x_j = 1$ if activity j was completed $[S_j + p_j]$ within the span of the planning horizon $[S_j + p_j] \leq T_{Max}$, and $x_j = 0$ if activity j cannot be completed with the budgeted resources over the span of the takt time. In Section 4.1.2, two new objective functions are formulated and proposed for solving scheduling problems in LVLVPS, in-line with the priorities and aspirations in such production systems. Section 4.1.3 is dedicated to the formulation of constraints and provides a detailed description and use-cases for each constraint.

Component	*Notation*		*Description*
Sets	J	$j \in \{ 1, \dots , N + M \}$	Activity Number
	J', J''	$j', j'' \equiv j$	The Equivalent Set of Activity j
	A	$\alpha \ \in \{1, \dots , N\} \setminus \beta$	Single-Mode Activities
	B	$\beta \ \in \{N+1, \dots , N+M\} \setminus \alpha$	Multi-Mode Activities
	L	$l \in \{ 1, \dots , L \}$	Resource Classification
	I	$i \in \{ 1, \dots , I \}$	Zone Classification
	T	$t \in \{ 1, \dots , T_{Max} \}$	Available Planning Horizon
Parameters	p_j	$p_j \geq 0$	Processing Time of Activity j
	$P_{jj'}$	$P_{jj'} \in \{ 0{,}1 \}$	Precedence Between Activities j and j'
	$L_{jj'}$	$L_{jj'} \in [-T, T]$	Lead/Lag Time Between Activities j and j'
	T_{Max}	$T_{Max} \geq 0$	Imposed Takt Time or Planning Horizon
	$M_{jj'}$	$M_{jj'} \in \{ 0{,}1 \}$	Multi-Mode of Activities j and j'
	w_{jl}	$w_{jl} \geq 0$	Resource Requirement of Activity j from Pool l
	W_l^{Max}	$W_l^{Max} \geq 0$	Resource Availability of Pool l
	y_{ij}	$y_{ij} \in \{ 0{,}1 \}$	Allocation of Activity j to Zone i
	Z_i	$Z_i > 0$	Capacity of Zone i
	$NC_{jj'}$	$NC_{jj'} \in \{ 0{,}1 \}$	Non-Concurrency Between Activities j and j'
	$CS_{jj'}$	$CS_{ji} \in \{ 0{,}1 \}$	Concurrent Start Between Activities j and j'
	$CF_{jj'}$	$CF_{ji} \in \{ 0{,}1 \}$	Concurrent Finish Between Activities j and j'
	ES_j	$ES_j \in [1, T_{Max}]$	Earliest Start Time of Activity j
	LS_j	$LS_j \in [1, T_{Max}]$	Latest Start Time of Activity j
	EF_j	$EF_j \in [1, T_{Max}]$	Earliest Finish Time of Activity j
	LF_j	$LF_j \in [1, T_{Max}]$	Latest Finish Time of Activity j
Variables	x_j	$x_j \in \{ 0{,}1 \}$	Scheduling of Activity j
	π_j	$\pi_j \in \{ 0{,}1 \}$	The Inverse of Scheduling of Activity j
	W_l	$W_l \geq 0$	Resource Requirement from Pool l
	$h_{jj'}$	$h_{jj'} \in \{ 0{,}1 \}$	Binary Variable to be used in Conjunction with Non-Concurrency Constraint
Deviation Variables	$\delta_{wl}^{\pm}$	$\delta_{wl}^{\pm} \geq 0$	Deviation to Type l Resources
	$\delta_T^{\pm}$	$\delta_T^{\pm} \geq 0$	Deviation to Time Constraint
	$\delta_\alpha^{\pm}$	$\delta_\alpha^{\pm} \geq 0$	Deviation to Scheduled Single-Mode Activities
	$\delta_\beta^{\pm}$	$\delta_\beta^{\pm} \geq 0$	Deviation to Scheduled Multi-Mode Activities
	S_j	$S_j \geq 0$	Start Time of Activity j

Table 4.02 - Sets, Parameters, and Variables for Continuous-time Work Center Scheduling Problems

4.1.2. Objective Function

The scarcity of time and resources in addition to precedence, zonal, and other temporal constraints inherent in LVLVPS may result in scenarios where there does not exist a feasible solution for scheduling of all activities. To tackle the potential infeasibility of this problem, two new objective functions are formulated and proposed. Objective function (4.01) adopts a priority-based pre-emptive goal programming approach and aims to minimize the required number of resources $\sum_l W_l$ in completion of the maximum number of activities, subject to an upper bound on time and resources. Deviation variables are used in the mathematical modeling of this objective function, where the first priority objective $P_1[\delta_\alpha^+ + \delta_\beta^+]$ represents the positive deviation to the aspiration criteria for the quantity of the completed single-mode $J \in \alpha$ and multi-mode $J \in \beta$ activities. Second priority objective aims to maximize the negative deviation to the aspiration criterion to the summation of resource budgets $P_2[\sum_l \delta_{wl}^-]$. The aspiration criteria for single-mode, multi-mode, and resource requirements are set to N, M, and W_l^{Max}, respectively, through constraints 4.24, 4.26, and 4.06.

$$Min\ Z \ = P_1[\delta_\alpha^+ + \delta_\beta^+] - P_2\left[\sum_l \delta_{wl}^-\right] \qquad (4.01)$$

There also exist scenarios where travel work is prohibited and the work center must complete the pre-defined statement of work. Objective function (4.02) is formulated and is proposed to be used in these scenarios, where time and resources are considered as soft constraints. The priority-based pre-emptive goal programming approach is similarly adopted in formulating this objective function, aimed at minimizing the positive deviation to the aspiration criteria for time δ_T^+ and resources δ_{wl}^+, while maximizing the negative deviation to the aspiration criterion for resources δ_{wl}^-. The deviation variables for time and resources are derived from constraints 4.08 and 4.07, respectively, where the aspiration criterion for time is set to the takt time T_{Max}.

$$Min\ Z \ = P_1[\delta_T^+] + P_2\left[\sum_l \delta_{wl}^+\right] - P_3\left[\sum_l \delta_{wl}^-\right] \qquad (4.02)$$

4.1.3. Constraints

Scheduling problems in LVLVPS are considered as heavily constrained problems, with precedence, resource, and time constraints, in addition to zonal, concurrency, non-concurrency, earliest and latest start and finish constraints. In this section, a new set of constraints is formulated for modeling of characteristics and constraints in LVLVPS. As discussed in Section 4.1.2, two distinct objective functions have been proposed through this chapter, permitting or prohibiting the traveling of incomplete activities. As such, constraints formulated in this section are considered as mandatory, discretionary, or restricted, depending on the selected objective function. Table 4.03 summarizes the applicability of constraints for the proposed objective functions.

✓ Mandatory O Discretionary × Restricted

Constraint No.	*Objective Function (61)* *Travel Work Permitted*	*Objective Function (62)* *Travel Work Prohibited*
63	✓	✓
64	✓	✓
65	✓	×
66	×	✓
67	✓	×
68	×	✓
69	O	O
70	O	O
71	O	O
72	O	O
73	O	O
74	O	O
75	O	O
76	O	O
77	O	O
78	✓	✓
79	✓	×
80	✓	✓
81	×	✓
82	×	✓
83	✓	×
84	✓	×
85	✓	×
86	✓	×
87	✓	✓
88	✓	✓

Table 4.03 - Application of Constraints to Various Types of Scheduling Problems

Interdependencies between successor and predecessor activities is a key constraint inherent in LVLVPS. The interdependencies between activities are modeled through the binary parameter $P_{jj'}$, where $P_{jj'} = 1$ if activity j is a predecessor to activity j', and $P_{jj'} = 0$ if otherwise. Constraint set (4.03) enforces the start time of successor activity $S_{j'}$ to be greater than or equal to the finish time of its predecessors $[S_j + p_j]$, plus or minus the lag and the lead times $L_{jj'}$. To ensure the compatibility of the proposed mathematical programming model with the scheduling of multi-mode activities, a new binary variable π_j is introduced and calculated through constraint set 80, where $\pi_j + x_j = 1$.

$$S_{j'} + M\pi_j \geq [S_j + p_j] + L_{jj'} \qquad \forall \quad j, j' \in \{ \alpha \cup \beta \quad : P_{jj'} = 1,\, x_j = 1 \} \qquad (4.03)$$

The resource constraints are imposed through constraint sets (4.04), (4.05), and (4.06). Constraint set (4.04) measures the total number of the required resources W_l from each classification l at the start time S_j of each activity j, where w_{jl} represents resource requirements for activity j of classification l, and x_j represents the successful scheduling of an activity. The upper bound on resources W_l^{Max} for each classification l are imposed through constraint set (4.05), applicable only in scenarios where travel work is permitted and deviation to the aspiration criterion to resource availability and budgets are prohibited. Constraint set (4.06) however, allows deviation to the aspiration criterion through deviation variables $[\delta_{wl}^{+}, \delta_{wl}^{-}]$, representing the positive and negative deviation of resource requirements to resource availabilities, respectively.

$$x_j w_{jl} + \sum_{j'} x_{j'} w_{j'l} \leq W_l \qquad \forall \quad j, j' \in \{ \alpha \cup \beta \quad : S_{j'} \leq S_j \leq S_{j'} + p_{j'} \} \qquad (4.04)$$

$$W_l \leq W_l^{Max} \qquad \forall \quad l = 1, \dots, L \qquad (4.05)$$

$$W_l + \delta_{wl}^{-} - \delta_{wl}^{+} = W_l^{Max} \qquad \forall \quad l = 1, \dots, L \qquad (4.06)$$

To ensure the on-time completion of all activities or to enforce an upper bound on the available planning horizon, constraints (4.07) and (4.08) are imposed. Constraint (4.07) is formulated and is proposed to be used in conjunction with objective function (4.01) where travel work is permitted. This constraint will thus imply the strict enforcement of the time constraint, to which deviation is not allowed. In scenarios where travel work is prohibited, there may exist scenarios where a feasible schedule can only be obtained through deviation to the aspiration criterion to time. Deviation variables $[\delta_T^{+}, \delta_T^{-}]$ are thus introduced and are quantified through constraint (4.08), representing the positive and negative deviation to the aspiration criterion for time T_{Max}, respectively.

$$T \leq T_{Max} \tag{4.07}$$

$$T + \delta_T^- - \delta_T^+ \leq T_{Max} \tag{4.08}$$

A common characteristic of LVLVPS is the deployment of resources onto the product. To ensure that the maximum allowable capacity Z_i of zone i has not exceeded, zonal constraints are enforced. An example of which includes the assignment of 3 resources in the main landing gear bay of an aircraft, where only two people can work safely in that area. Constraint set (4.09) is enforced at the start time S_j of every activity j, and ensures that the total number of resources of all classifications $\sum_l w_{jl}$ assigned to each zone i, is less than or equal to the allowable capacity Z_i for that zone. This constraint is imposed for all zones and must be satisfied to ensure the resultant schedule if feasible from a practical standpoint.

$$\sum_l x_j y_{ij} w_{jl} + \sum_l \sum_{j'} x_{j'} y_{ij'} w_{j'l} \leq Z_i \qquad \begin{array}{l} \forall \quad j,j' \in \{\, \alpha \cup \beta \quad : S_{j'} \leq S_j \leq S_{j'} + p_{j'} \,\} \\ \forall \quad i = 1, \dots, I \end{array} \tag{4.09}$$

There exist scenarios where the simultaneous progression of two or more activities are prohibited due to the nature of work or factors affecting resource or product safety. Such non-concurrency constraints are imposed through constraint sets (4.10) and (4.11). Two activities j and j' are identified as non-concurrent if $NC_{jj} = 1$. Constraint set (4.10) ensures that the completion time $[S_j + p_j]$ of activity j is less than or equal to the start time $S_{j'}$ of activity j', while constraint set (4.11) is imposed to confirm that the start time S_j of activity j is greater than or equal to the completion time $S_{j'} + p_{j'}$ of activity j. In formulating a robust mathematical model, compatible with single and multi-mode activities, a new binary variable $h_{jj'}$ is employed in conjunction with Big M, representing a large number. Through the use of binary variable $h_{jj'}$, only one of the following conditions have to hold, while both constraints will be satisfied, where an activity j must be completed prior to starting of its non-concurrent activity j', or activity j must start after the completion of its non-concurrent activity j'.

$$S_j + p_j \leq S_j' + M h_{jj'} \qquad \forall \quad j,j' \in \{\, \alpha \cup \beta \quad : NC_{jj'} = 1 \,\} \tag{4.10}$$

$$S_j + M[1 - h_{jj'}] \geq S_{j'} + p_{j'} \qquad \forall \quad j,j' \in \{\, \alpha \cup \beta \quad : NC_{jj'} = 1 \,\} \tag{4.11}$$

Contrary to non-concurrency constraints, there may exist a mandate to start or complete two or more activities concurrently. Concurrent start between activity j and j' is imposed through constraint set (4.12), enforcing the identical start times for the two activities. Concurrent finish constraint is similarly imposed through constraint set (4.13), where the completion time $[S_j + p_j]$ of activities j and j' are set to be equal if $CF_{jj'} = 1$, signifying the two activities must finish concurrently.

$$CS_{jj'}[S_j - S_{j'}] = 0 \qquad \forall \quad j, j' \in \{ \alpha \cup \beta \quad : CS_{jj'} = 1 \} \qquad (4.12)$$

$$CF_{jj'}\left[[S_j + p_j] - [S_{j'} + p_{j'}]\right] = 0 \qquad \forall \quad j, j' \in \{ \alpha \cup \beta \quad : CF_{jj'} = 1 \} \qquad (4.13)$$

The start time and/or completion time of activities may also be influenced by factors external to the manufacturing process, and thus there may exist mandates for the earliest or latest start or finish times. Earliest start is enforced through constraint set (4.14), where the start time S_j of activity j must be greater than or equal to the earliest start time ES_j of that activity. Note that to ensure flexibility in capturing single and multi-mode activities as well as travel work, the binary variable x_j is used. The latest start constraint is similarly imposed through constraint set (4.15) where the start time S_j of activity, j is set to be less than or equal to the latest start LS_j of that activity. Constraint sets (4.16) and (4.17) are formulated to capture the latest LF_j and earliest completion times EF_j of activity j, respectively. Through these constraints, activity j must be completed $[S_j + p_j]$ prior to the imposed latest completion time LF_j or after the mandated earliest completion time EF_j. Note that constraint sets (4.14), (4.15), (4.16), and (4.17) are categorized as discretionary in Table 4.03, signifying that these constraint sets may be omitted without impacting the functionality of the proposed mathematical programming models, in scenarios where the problem at hand is not subject to such constraints.

$$x_j S_j \geq x_j ES_j \qquad \forall \quad j = 1, \dots, N + M \qquad (4.14)$$

$$x_j S_j \leq x_j LS_j \qquad \forall \quad j = 1, \dots, N + M \qquad (4.15)$$

$$x_j[S_j + p_j] \leq x_j LF_j \qquad \forall \quad j = 1, \dots, N + M \qquad (4.16)$$

$$x_j[S_j + p_j] \geq x_j EF_j \qquad \forall \quad j = 1, \dots, N + M \qquad (4.17)$$

Binary variable x_j is used in constraint sets (4.18) and (4.19) to distinguish between traveled activities and activities that were completed over the span of the imposed planning horizon T_{Max}, where $x_j = 1$ if activity j was completed prior to the end of the takt $[S_j + p_j] \leq T_{Max}$, and $x_j = 0$ if the activity j was completed after the imposed takt time $[S_j + p_j] > T_{Max}$.

$$x_j = 1 \qquad \forall \quad j \in \{\, \alpha \cap \beta \quad : \; S_j + p_j \leq T_{Max} \,\} \qquad (4.18)$$

$$x_j = 0 \qquad \forall \quad j \in \{\, \alpha \cap \beta \quad : \; S_j + p_j > T_{Max} \,\} \qquad (4.19)$$

Binary variable π_j is a function of x_j, where $\pi_j = [1 - x_j]$ as formulated through constraint set (4.20). This variable is used in conjunction with the Big M in the precedence constraint set (4.03) and represents the inverse of the on-time completion of activity j.

$$\pi_j + x_j = 1 \qquad \forall \quad j = 1, \dots, N + M \qquad (4.20)$$

Constraint sets (4.21) and (4.22) are formulated to ensure the on-time completion of all activities in scenarios where travel work is prohibited. Constraint set (4.21) ensures that all single-mode activities $j \in \alpha$ are scheduled to be completed prior to the imposed time constraint, where each activity j can only be scheduled once. Constraint set (4.22) on the other hand, is formulated to ensure the on-time completion of multi-mode activities $j \in \beta$, where only a single mode of a multi-mode activity must be scheduled. Multi-mode activities are identified through the binary parameter $M_{jj'}$, where $M_{jj'} = 1$ if activity j is a secondary mode of activity j' or vise versa.

$$x_j = 1 \qquad \forall \quad j \in \alpha \qquad (4.21)$$

$$x_j + x_{j'} = 1 \qquad \forall \quad j, j' \in \{\, \beta \quad : \; M_{jj'} = 1 \,\} \qquad (4.22)$$

Contrary to constraint sets (4.21) and (4.22), constraint sets (4.23) – (4.26) are formulated to be used in conjunction with objective function (4.01), where travel work is permitted. Constraint set (4.23) ensures that single-mode activities $j \in \alpha$ cannot be scheduled to be completed more than once, where $x_j = 1$ if activity j was completed on-time. The aspiration criterion for the number of single-mode activities $j \in \alpha$ is set to N, equivalent to the total number of single-mode activities, through constraint (4.24). Additionally the deviation variables for the number of completed single-mode activities $\delta_{\alpha}^{\pm}$ are defined through this constraint set.

$$x_j \leq 1 \qquad \forall \quad j \in \alpha \tag{4.23}$$

$$\sum_{j \in \alpha} x_j + \delta_{\alpha}^{-} - \delta_{\alpha}^{+} = N \tag{4.24}$$

Constraint sets (4.25) and (4.26) are similar to that of (4.23) and (4.24), aimed at multi-mode activities $j \in \beta$. Constraint set (4.25) enforces that only a single mode of a multi-mode activity can be scheduled, where activities cannot be scheduled more than once. Through constraint (4.26), the aspiration criterion for the aspired number of completed multi-mode activities is defined to be M, equivalent to the total number of multi-mode activities. Deviation variables for multi-mode activities $\delta_{\beta}^{\pm}$ are also defined through this constraint, used in conjunction with deviation variable for single-mode activities $\delta_{\alpha}^{\pm}$, formulated through constraint set (4.24) in objective function (4.01) aimed at minimizing the number of incomplete activities.

$$x_j + x_{j'} \leq 1 \qquad \forall \quad j, j' \in \{\beta \quad : \; M_{jj'} = 1\} \tag{4.25}$$

$$\sum_{j \in \beta} x_j + \sum_{j' \in \beta} x_{j'} + \delta_{\beta}^{-} - \delta_{\beta}^{+} = M \tag{4.26}$$

To ensure that the proposed mathematical programming model is compatible with single-mode as well as multi-mode activities, and permits the traveling of incomplete activities, constraint sets (4.27) and (4.28) are formulated and proposed. These constraint sets are complementary to precedence constraints (4.03) and are imposed to ensure that a predecessor activity must be scheduled if a successor activity is planned to be completed. This condition is imposed through constraint set (4.27) for single-mode activities $j \in \alpha$ and through constraint set (88) for multi-mode activities $j \in \beta$.

$$x_j \geq x_{j'} \qquad \forall \quad j, j' \in \{\alpha \quad : P_{jj'} = 1\} \tag{4.27}$$

$$x_j + x_{j'} \geq x_{j''} \qquad \forall \quad j, j' \in \beta,\ j'' = \alpha \cup \beta\ :\ P_{jj''} = 1,\ M_{jj'} = 1 \qquad (4.28)$$

4.2. Genetic Algorithm

Exact optimization algorithms are not the most effective in tackling large-scale optimization problems with a high-dimensional search space. In such problems, the search space grows exponentially with the problem size, thus an exhaustive search algorithm is not practical [111]. Metaheuristics on the other hand, are effective in modeling and solving large-scale continuous optimization problems yielding to near-optimal solutions in reasonable computational time. Metaheuristics are a set of intelligent strategies to enhance the efficiency of heuristic procedures, through an iterative generation process by guiding a subordinate heuristic by combining intelligently different concepts for exploring the search space [112]. Metaheuristics can be classified based on a set of selected characteristics, and most notably include nature-inspired against non-nature inspired, population-based against single point search, dynamic against static objective function, single neighborhood against various neighborhood structures, and memory usage against memory-less methods [113]. The majority of metaheuristics are nature-inspired algorithms such as Ant Colony Optimization, Particle Swarm Optimization, Simulated Annealing, and Genetic Algorithm. The Ant Colony Optimization (ACO) is a naturally inspired, population-based, implicit metaheuristic and is most properly used for graph-based optimization problems. This metaheuristic offers a scalable, robust and flexible algorithm in dynamic environments. However, it is not easily coded and uses a trial and error procedure in initializing the parameters. The original ACO algorithm is designed for discrete search space [114,115]. The Particle Swarm Optimization (PSO) on the other hand, is a swarm-based meta-heuristic algorithm which simulates the motion of birds and insects in order to find the near-optimum solution. It is easy to implement with few parameters to adjust and has an efficient global search approach. However, although many extended PSO algorithms have been reported, their performance enhancement is an open problem due to its simple structure. Furthermore, PSO Suffers from trapping into local optima and slow convergence speed [116,117]. On the contrary, Simulated Annealing (SA) is a local search metaheuristic, inspired by the energy transfer as demonstrated through thermodynamics. SA has a low computation time with convergent properties and is easily implemented, however the solution quality is highly dependent on the maximum number of iterations in the inner loop and the initial temperature [118,119].

Genetic Algorithm (GA) was initially introduced in by Holland [120], and later extended and described in greater detail by Goldberg [121]. This metaheuristic is the most popular form of Evolutionary Algorithms, employing the principles of biological evolution through a structured yet randomized search strategy in solving complex optimization problems. GA is characterized as a naturally inspired, population-based, implicit metaheuristic, which uses a defined set of procedures and techniques to combine existing solutions in creating a new generation of solutions, in search for the global optimum solution [122]. GA is proven effective in solving a wide range of optimization problems, particularly in solving large-scale scheduling problems. It is designed for real and binary search space and results in the global optimum solutions in many cases. However, it has a complex encoding scheme, and high dependency on the crossover and mutation rates on the stability and convergence, and the results may be unpredictable and sub-optimal [123,124]. Despite its disadvantages and drawbacks, GA has been proven to be highly effective and efficient in solving continuous large-scale optimization problems. As such, it has been selected as the preferred metaheuristic in modeling and solving continuous-time large-scale scheduling problems in LVLVPS. The survival-of-the-fittest strategy leads to a selection of potential solutions with a bias towards reinforcing chromosomes with the highest fitness, thus resulting in successively better solutions. Prior to executing the GA, a suitable representation of the scheduling problem must be devised, incorporating the characteristics and activity attributes inherent in LVLVPS. Fitness functions must also be formulated to assign fitness values to potential chromosomes, in addition to a strategy for genetic operators in crossovers and mutations, taking advantage of the information representation in the chromosomes. Despite the extensive application of GA in solving a wide range of scheduling problems, the majority are restricted to job-shops, flow-shops, and RCPSP, with limited literature reported on the application of GA in modeling and solving scheduling problems in LVLVPS. Through this section, a detailed procedure for initializing and solving large-scale scheduling problems using GA in LVLVPS is proposed.

4.2.1. Pre-Processing Procedure

In modeling the scheduling problem at hand using GA, a scheduling order is devised that prioritizes the scheduling of activities based on the cumulative number of their predecessors. This methodology was initially proposed by Tavarez et. al (1990) [125], and later adopted in solving multi-mode RCPSP [96], through which, the activities with the lowest scheduling order, exhibiting the least number of cumulative predecessors, are prioritized to be scheduled first. The mathematical formulation for calculating the scheduling order of activities are represented in Equations (4.29) and (4.30). Through Equation (4.29), the scheduling order of 1 is assigned to all activities without predecessors, if and only if $\sum_{j=1}^{N+M} P_{jj'} = 0$. For all succeeding activities, the scheduling order number will be the maximum scheduling order of their predecessor(s) plus one, as demonstrated through Equation (4.30). The scheduling order is calculated for each node, and is used by the algorithm in sequencing and scheduling of the activities. The algorithm starts with scheduling of activities with a scheduling order of 1, and then proceeds to schedule the activities with the lowest calculated scheduling order, representing the cumulative number of predecessors. This enables the algorithm to create an initial population that satisfies the imposed precedence constraints.

$$f_{j'} = 1 \qquad \Leftrightarrow \qquad \sum_{j=1}^{N+M} P_{jj'} = 0 \tag{4.29}$$

$$f_{j'} = m + 1 \qquad \Leftrightarrow \qquad \max_{j\,:\,\left[P_{jj'}=1\right]} f_j = m \tag{4.30}$$

4.2.2. Chromosome Representation

In the direct representation of the problem, the resultant schedule is regarded as a chromosome, as such no decoding procedure is required. The chromosome, inclusive of all activities includes the information required by the GA in search of the optimum solution. The scheme of the direct representation is depicted in Figure 4.02, where each cell, representing a gene, is comprised of the activity number, the selected mode for that activity, the scheduling order, processing time, zonal assignment, and the scheduled start and finish times. The corresponding modes of activities are encoded such that mode α denotes that the activity is a single-mode, and as such has only one mode to be scheduled. Mode β_1 represents the primary mode of multi-mode activities, and β_2 represents the secondary mode.

$Activity$: 1 $Mode$: α $Order$: 1 $Duration$: p_1 $Zone$: i $[Start, Finish]$	$Activity$: 2 $Mode$: β_1 $Order$: 2 $Duration$: p_2 $Zone$: i $[Start, Finish]$	…	$Activity$: $N + M$ $Mode$: β_2 $Order$: Q $Duration$: p_{N+M} $Zone$: i $[Start, Finish]$

Figure 4.02 - Chromosome Representation

For instance, the first gene in Figure 4.02, exemplifies that activity 1 is a single-mode activity, with a scheduling order of 1, duration of p_1, and is assigned to zone i with the corresponding scheduled start and finish times. The objective of the algorithm is to construct a feasible schedule that satisfies the imposed constraints, aimed at minimizing a pre-determined evaluation function, represented through a fitness function, discussed in Section 4.2.4.

4.2.3. Initialization

The GA starts by generating the initial population. The scheduling order numbers are initially calculated through Equations (4.29) and (4.30). Thereafter, activities are scheduled based on their schedule order numbers, where activities with identical scheduling order numbers are randomly scheduled, such that the precedence constraints are satisfied. The modes of multi-mode activities are selected randomly for the initial population, where only a single mode of a multi-mode activity can be scheduled. The start and finish times of each activity are then obtained in accordance with the successful satisfaction of the resource, lead and lag times, zonal, concurrencies, and non-concurrency constraints. The corresponding makespan for completion of all activities and the number of the activities completed on-time is then calculated for each schedule or chromosome. This procedure is repeated in generating the initial population, where the random assignment of modes and scheduling of activities with identical scheduling order numbers will result in a diverse, yet feasible set of solutions. The initial population is generated such that all precedence, resource, and zonal constraints, as well as earliest and latest start and finish times are successfully satisfied, where each activity can only be scheduled once. The initial population is thus comprised of a N_p feasible solutions, where N_p represents the population size.

4.2.4. Evaluation

Upon the complete generation of the initial population, the chromosomes representing feasible schedules are evaluated through a fitness function. The fitness value is equivalent to the objective value corresponding to Equations (4.01) and (4.02). In Equation (4.01) the objective is to minimize the required number of resources in completion of the maximum number of activities, while Equation (4.02) aims to minimize the positive deviation to the aspiration criteria to time and resource constraints. In convention, the higher fitness values are desired, however since this problem aims at minimizing a series of evaluation criteria, solutions yielding lower objective values are superior to those of higher objective values. As such, a chromosome resulting in a lower number of travelled work and/or resource requirement will have a higher probability of selection, discussed in the following section.

4.2.5. Selection Strategy

Selection embodies the principle of the survival of the fittest and is based on the fitness value of chromosomes. The strategy in selecting prospective parents is carried out stochastically where schedules with higher fitness have a greater probability of being selected for the reproduction of the next generation. There exist four popular selection strategies, namely the Rank Selection, Tournament Selection, and the Roulette Wheel Selection. In the rank selection, all chromosomes are ranked and are sorted from the best fitness value to the worst, where the worst chromosome is ranked as 1. The purpose of the rank selection is to prevent quick convergences, however it requires the ranking and sorting of the population for each iteration [126]. In tournament selection, various tour of a few individuals are selected at random from the population, and the best individuals are selected as parents. The individual with the highest fitness value is the winner of the tournament and is selected for generating the next population. The tournament selection does not require the scaling and sorting of the solutions, but does not guarantee the reproduction of best solutions [127,128]. The roulette wheel strategy uses a combination of the rank selection strategy and a stochastic selection model, where each individual is provided with a slice of the roulette wheel, with size reflective of its fitness value. The roulette wheel provides a probability for each individual to be selected for reproduction and is found to be effective and efficient in avoiding local optima [129]. As such, the roulette wheel strategy is adopted as the selection strategy where segments of the roulette wheel are allocated to the schedules in proportion to their relative fitness, and individuals with greater fitness have a higher probability of being selected [130]. The selection of the parents is executed through generating random numbers in the interval of [0,1] and selecting the corresponding individuals as parents, where two parents undergo the reproductive operators. The probability of an individual $P(\mu)$ being selected is calculated through Equation (4.31), where μ represents the chromosome number, and N_P denotes the size of the population [121].

$$P(\mu) = \frac{f(\mu)}{\sum_{\mu=1}^{N_P} f(\mu)} \tag{4.31}$$

4.2.6. Crossover

The crossover operation is applied to each pair of parents, in the reproduction of two offspring, inhering part of their features. There exist various crossover strategies, differentiated by the problem type and chromosome structure. Some of the more notable and popular crossover techniques include the single-point crossover, 2-point crossover, multi-point crossover, variable to variable crossover, and uniform crossover. The single point crossover is the simplest approach, where paired chromosomes are each cut at a randomly chosen crossover points [131]. In the 2-point crossover, chromosomes are cut at two randomly selected crossover points. The exchange is executed through the swapping of either the inner portion of the crossover point or the outer portion. The 2-point crossover gives a higher chance to chromosomes in exchanging the foremost genes [132]. The multi-point crossover involves more than two cuts points, providing a more dispersed exchange of design characteristics. The implementation of the multi-point crossover is similar to that of 2-point crossover point, where multiple randomly selected crossover points are selected and the inner portions or the outer portions are swapped [133]. In the variable-to-variable crossover technique, the paired individuals are decomposed into their substrings, and a single point crossover is carried out on all substrings. Each design variable amongst the chromosomes is activated to accomplish the design exchange separately [134]. Uniform crossover requires a randomly created crossover mask. The genes of offspring are copied from parents according to this mask. At positions where there is a 1 in the mask, genes are carried from parent 1, and at positions where there is a 0 in the mask, and genes are carried from parent 2. The second offspring is created either by using the complementary of the original mask, or by forming a new mask and repeating the same process [135].

The one-point crossover is adopted as the reproductive strategy, as it is the most appropriate given the imposed precedence constraints. The one-point crossover allows the generation of new offspring without significant disruption to the precedence constraints. In one-point crossover a random integer in the interval of $[1, N + M]$ is generated, representing the position number of the activity in the representation scheme. The portion to the left of the selected activity of the first parent is passed onto the first offspring, followed by the activities on the right portion of the selected activity of the second parent. Duplicate activities are omitted, and activities scheduled in the left portion of the second parent that are absent in the left portion of the first parent are scheduled immediately after the randomly generated position. Figure 4.03 demonstrates the crossover strategy adapted in the proposed GA, where a simplified pair of chromosomes comprised of 10 activities are selected in the reproduction of two offspring. The crossover operation is executed $\lceil N_P \times P_C \rceil$ times, where P_C represents the probability of crossover. This crossover strategy leads to the reproduction of $2\lceil N_P \times P_C \rceil$ offspring, as each crossover results in the generation of two new offspring. Note that while a one-point crossover strategy is adapted, the genes within a chromosome are rearranged following a crossover operation to ensure feasibility of the resultant offspring. Through this rearrangement, the algorithm will select the first portion of the first parent, and the second portion of the second parent. The algorithm will then move the minimum number of genes within the chromosome to ensure the successful satisfaction of the imposed precedence, lead and lag times, concurrency and non-concurrency constraints, as well as zonal and resource constraints. As such, the resultant offspring of the crossover operation will always be a feasible schedule.

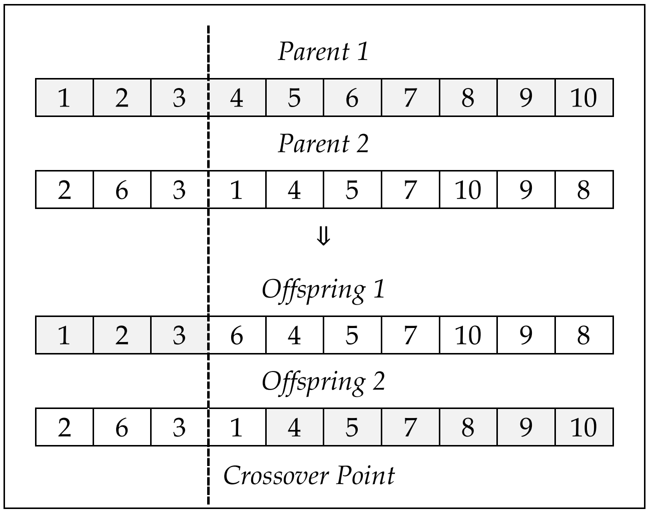

Figure 4.03 - Crossover strategy example

4.2.7. Mutation

Mutation is a unary operator that reinforces lost genetic material into the population that alters information represented in the chromosome. Mutation operator requires only one parent in generating an offspring with a probability of P_M, where the mutation is executed $\lceil N_P \times P_M \rceil$ times, in generating $\lceil N_P \times P_M \rceil$ offspring. The selection of the parents is carried out through the Roulette Wheel procedure. The mutation strategy devised for solving scheduling problems in LVLVPS is comprised of two types, selected by a random number in the interval of [0,1]. If the generated random number is less than or equal to 0.5, the first mutation strategy is applied, where the modes of η activities are swapped. η are random numbers in the interval of $[1, M]$. On the contrary, if the random number produced is greater than 0.5, a new solution is generated applying the initialization methodology proposed in Section 4.2.3. This procedure ensures that the algorithm avoids falling into local optimums and accelerate the search for a solution with improved fitness. Similar to the crossover operation, the resultant schedule generated from a mutation operator must be feasible, satisfying the imposed time, resource, zonal, concurrency and non-concurrency constraints, as well as precedence and lead and lag-time constraints. For instance, a new chromosome generated as a result of a gene medication through a mode-swap as is the case with the mutation operator, may ultimately require additional resources and thus impede the imposed resource budget. The algorithm is designed to rearrange and recalculate a new start and finish time for each activity to ensure feasibility, while minimizing slack. This will ensure that all chromosomes in the solution space are feasible, whether transferred over from the previous generation, or created as a part of a genetic operation, thus eliminating the need for a feasibility check in the selection of the parents.

4.2.8. New Generation

Upon the successful completion of the genetic operators, three pools of solutions are obtained to be evaluated. The first pool is the original solution pool from the initialization phase or the previous iterations with a size of N_P. The second and the third solution sets are obtained through the crossover and the mutation operators, where $2\lceil N_P \times P_C \rceil$, and $\lceil N_P \times P_M \rceil$ new solutions are generated, respectively. The three pools are then combined and are sorted in an ascending order based on their fitness value, obtained through Equation (4.31). The survival-of-the-fittest is then employed in the selection of the top N_P non-duplicated solutions with the highest fitness. The algorithm will then move onto the next iteration executing procedures outlined in Sections 4.2.5 through 4.2.8. Once the pre-determined termination criterion is met, set to the maximum number of iterations, the algorithm will be terminated and the chromosome with the greatest fitness will be selected as the optimum or the best-reached solution.

4.3. Case Study

To validate and verify the proposed optimization models, a real-world case study of the final assembly line of a dual-engine narrow-body aircraft is conducted. The assembly line of this aircraft comprises 15 work centers, each responsible to complete a pre-defined statement of work over the span of the takt time. The takt time represents the drumbeat of the assembly line and is calculated and is imposed in response to the market demand. The takt time or the available planning horizon is set to $T_{Max} = 48$ hours, equivalent to 6 shifts, or 2 days operating on three shifts per day. The work center under study is responsible for the final assembly of the cockpit and the empennage to the center fuselage. This statement of work is comprised of $N + M = 48$ activities, $N = 43$ of which are classified as single-mode $J \in \alpha$ activities, and the remaining $M = 5$ activities are classified as multi-mode $J \in \beta$ activities. Multi-mode activities are highlighted in a lighter shade of grey in Figure 89 and are a subset of activities assigned to the work center $\beta = \{7,8,9,17,24\}$. The processing time of activities ranges from 30 minutes to 5.6 hours, where activities may require multiple resources of $L = 3$ distinct resource types. Human resources assigned to this work center are segregated by their skills, where mechanical assemblers, electrical technicians, and aerodynamic sealers are considered as type 1,2, and 3 resources, respectively, with a budget of $W_1^{max} = 2$, $W_2^{max} = 2$, and $W_3^{max} = 2$. Figure 4.04 is the activity-on-node network diagram for the statement of work assigned to this work center. It can be demonstrated through this that activities are highly interdependent, where there exist 101 interdependency relationships between $N + M = 48$ activities. For the purpose of this study, interdependencies $P_{jj'}$ between activities j and j' are represented through a 48×48 matrix. Concurrency and non-concurrency activities are similarly represented through matrices in the development of a generic metaheuristic, compatible to solve different work center scheduling problems in LVLVPS. Activities are assigned to three distinct zones $I = 3$, where the maximum allowable capacity for zone 1, 2, and 3 are set to $Z_1 = 6$, $Z_2 = 4$, and $Z_3 = 6$, respectively.

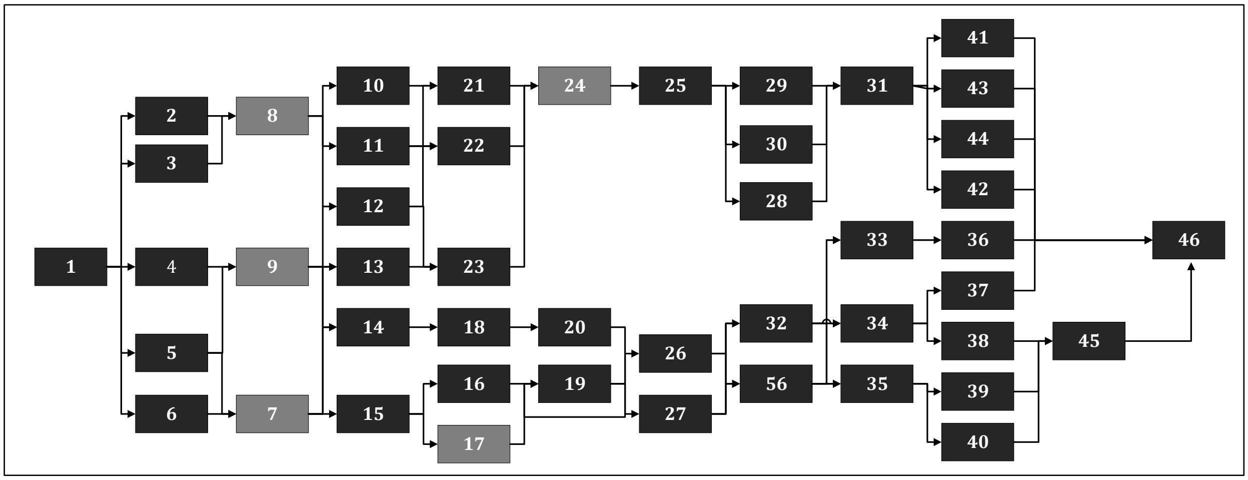

Figure 4.04 – Case Study 3 - Activity-On-Node Network Diagram

This case study is solved using the two distinct approaches outlined in Section 4.1.2, wherein Section 4.3.1, the objective is to minimize the number of incomplete activities through the strict enforcement of time and resource constraints. The proposed algorithm will also aim to minimize the required number of resources if a feasible solution for on-time completion of all activities can be obtained. The problem is solved again in Section 4.3.2, assuming that travel work is prohibited, where the work center is responsible for completing the pre-defined work package. This case study was modeled using GA and solved in MATLAB on a 64-bit Windows operating system with an Intel 6th generation i7 processor, operating at 2.6GHz with a 16.0GB RAM.

4.3.1. Travel Work Permitted

The metaheuristic prescribed in Section 4.2 was employed in developing a GA for solving time and resource-constrained scheduling problems in LVLVPS. The objective of this algorithm is to minimize the required number of resources in the completion of the maximum number of activities within the pre-defined planning horizon while utilizing a maximum of the imposed resource quantities. The algorithm was programmed into MATLAB, and solved in 7.6 seconds, after 1,000 iterations, where the termination criterion was set to a thousand iterations. The algorithm yielded to an objective value of 10, representing the traveling of 10 incomplete activities $\delta_{\alpha}^{+} + \delta_{\beta}^{+} = 10$, while a reduction in resources was not possible $\sum_{l} \delta_{wl}^{-} = 0$. The Gantt chart depicted in Figure 4.05 represents the optimum work center schedule for the problem at hand. It can be demonstrated through an analysis of this schedule that all interdependencies, lead and lag times, concurrencies, and non-concurrencies were successfully satisfied.

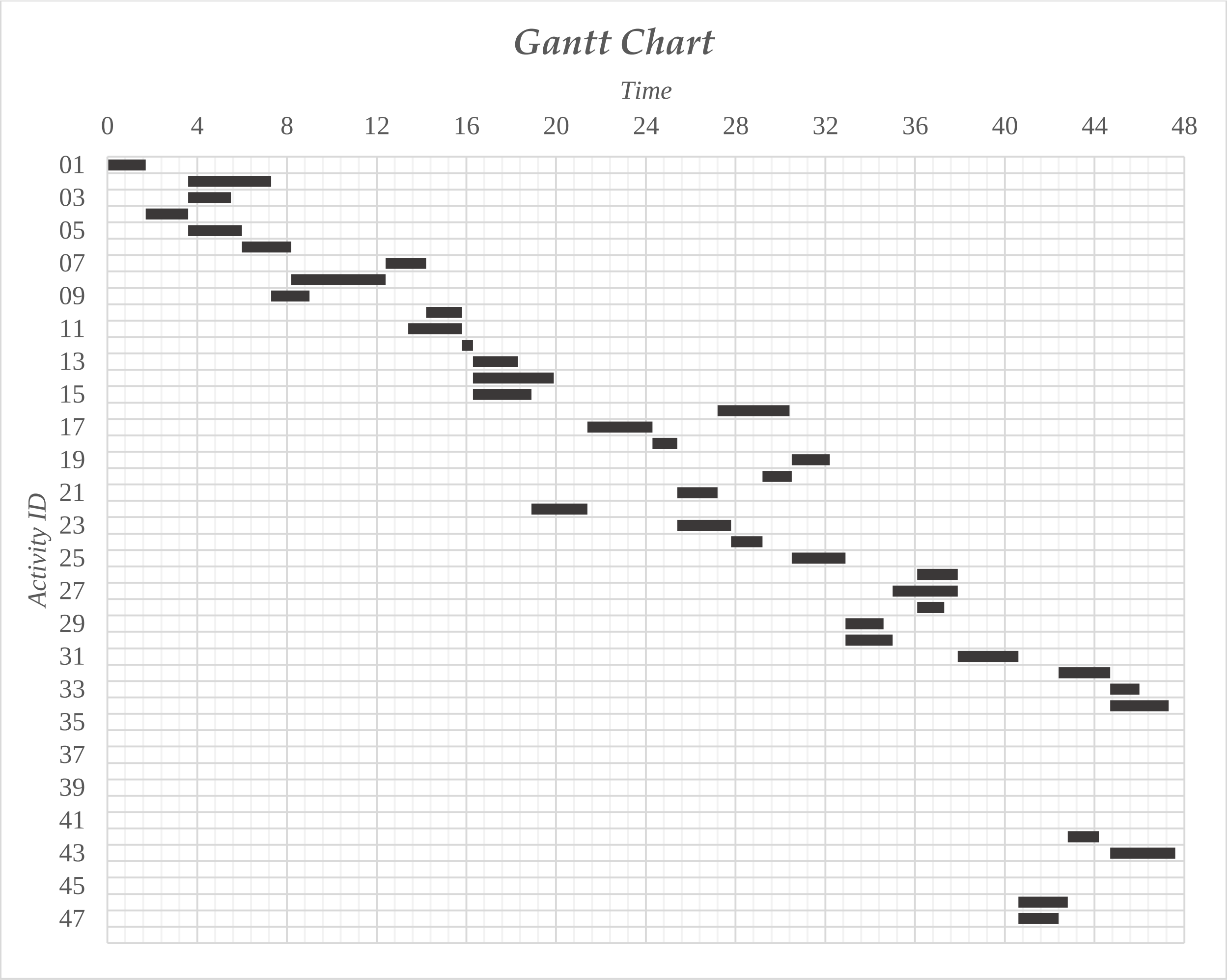

Figure 4.05 – Gantt Chart for the Optimized Work Center Schedule

Figures 4.05 through 4.13 demonstrate that the imposed precedence, resource, and zonal constraints were successfully satisfied. Figure 4.06 illustrates the successful satisfaction of the interdependency constraints, where the x-axis represents the interdependency identifier, and the y-axis represents the free float time in hours. The free float times that are equal to or are greater than zero are evidence that a successor activity has started upon or after the completion of a predecessor activity. Figures 4.07, 4.08, and 4.09 demonstrated resource usage and utilization for type 1, type 2, and type 3 resources, respectively, representing mechanical assemblers, electrical technicians, and aerodynamic sealers. The dashed horizontal line denotes resource availability for each classification, and the bars represent the utilized number of resources. Figure 4.10 highlights the overall resource availability and utilization. It is demonstrated through Figures 4.07 through 4.10 that all resource constraints are successfully satisfied.

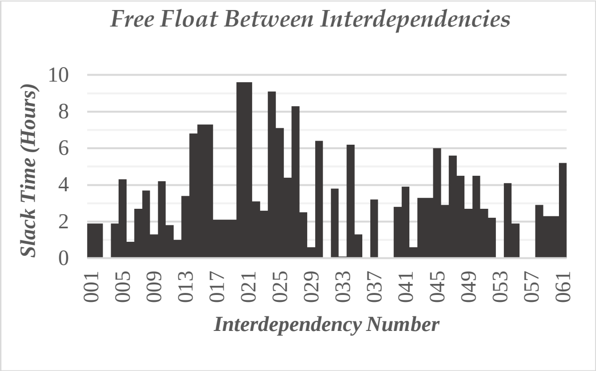

Figure 4.06 - Free Float

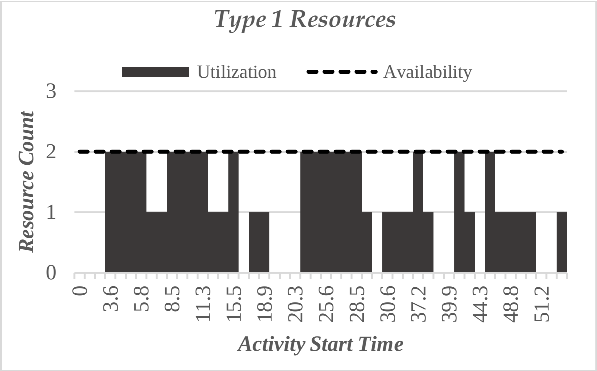

Figure 4.07 - Resource 1 Utilization

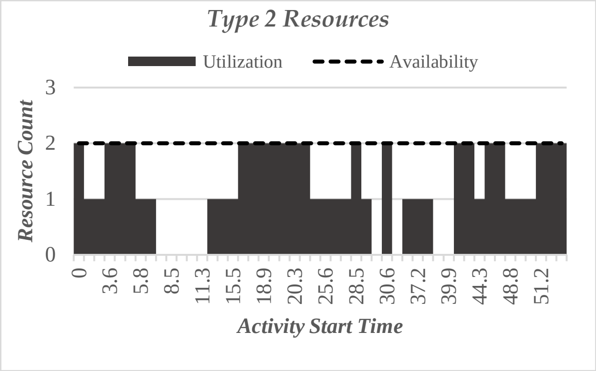

Figure 4.08 - Resource 2 Utilization

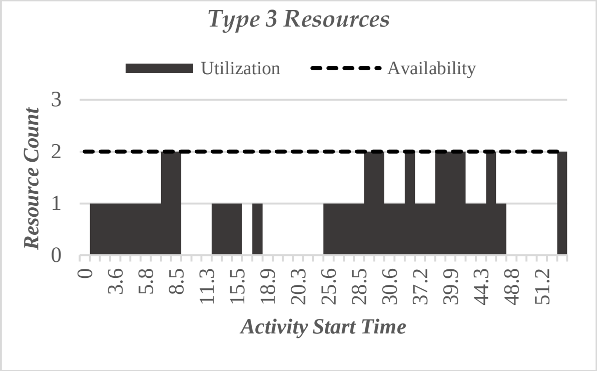

Figure 4.09 - Resource 3 Utilization

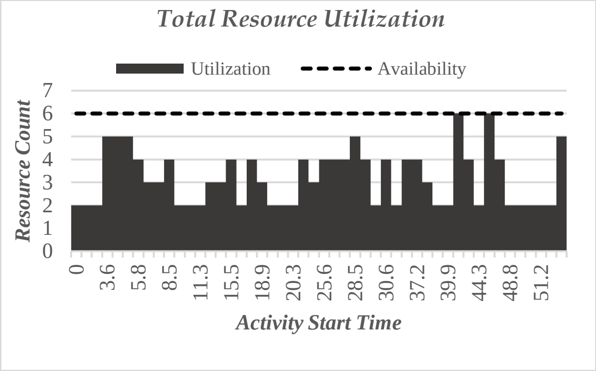

Figure 4.10 - Overall Resource Utilization

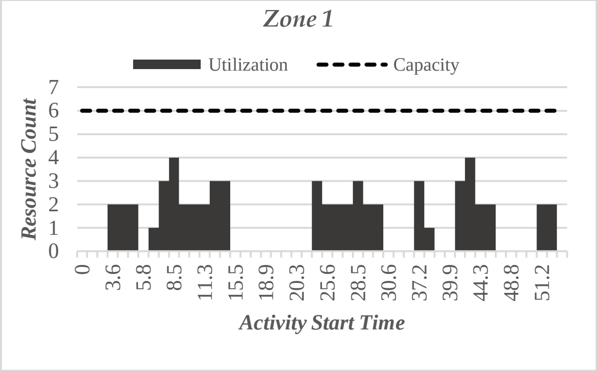

Figure 4.11 - Zone 1 Utilization

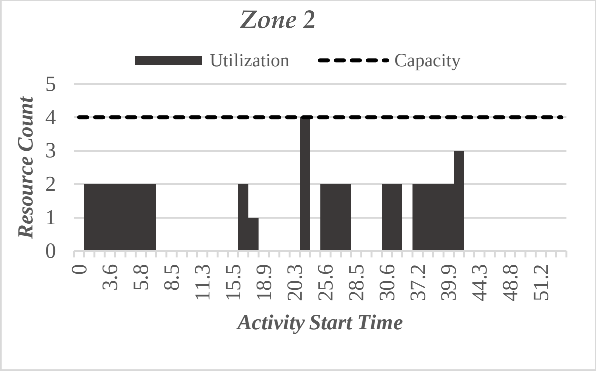

Figure 4.12 - Zone 2 Utilization

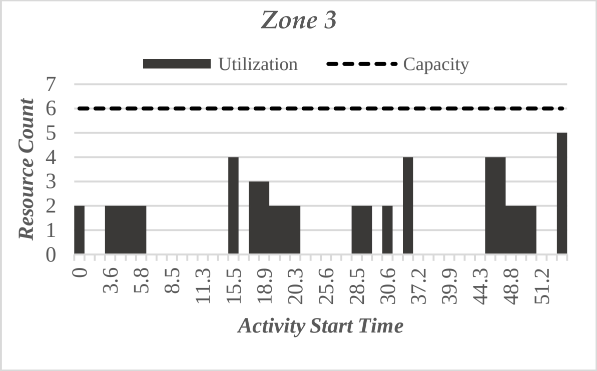

Figure 4.13 - Zone 3 Utilization

Figures 4.11, 4.12, and 4.13 are similarly appended, representing zonal utilization for zones 1, 2 and 3 respectively, where the dashed line represents the zonal capacity or the maximum number of resources that can be simultaneously assigned to each zone. The bars represent the actual number of resources that are assigned to each of the zones at the starting time of each activity, validating that all zonal constraints were successfully satisfied. It can thus be concluded that the proposed optimization model is effective in modeling and solving complex scheduling problems in LVLVPS, where the strict enforcement of time and resources are in effect, resulting in potential traveling of incomplete activities.

4.3.2. Travel Work Prohibited

To verify and validate the proposed metaheuristic for solving large-scale continuous-time scheduling problems, mandating the completion of the work package in its entirety, the case study presented in Section 4.3 is solved, tailored to incorporate objective function (4.02). The developed GA is aimed at minimizing the positive deviation to the aspiration criteria to time and resources while maximizing the negative deviation to resource requirements. This problem is solved iteratively, where minimizing the positive deviation to the aspiration criterion to time has the highest priority, followed by minimizing the excess number of resources. The GA was programmed into MATLAB and was solved in 6.2 seconds, after 1000 iterations, yielding to the objective value of 10.8. The resultant schedule required a positive deviation of $\delta_T^+ = 8.8$ hours to the aspiration criterion to time, and a positive deviation of $\sum_l \delta_{wl}^+ = 2$ units to the aspiration criterion to resources. To ensure the completion of all activities within 56.8 hours, there is a need for an additional type 2 resource $\delta_{w2}^+ = 1$, and an additional type 3 resource $\delta_{w3}^+ = 1$. It was also found that a feasible schedule with negative deviation to the imposed resource constraint could not be obtained $\sum_l \delta_{wl}^- = 0$. The Gantt chart depicted in Figure 4.14 is the optimum production schedule, an analysis of which demonstrates the successful satisfaction of the imposed precedence, concurrency, and non-concurrency constraints. It can also be concluded through this figure that the pre-defined statement of work is planned to be completed in 56.8 hours, representing the critical path with consideration to the imposed constraints and assumptions.

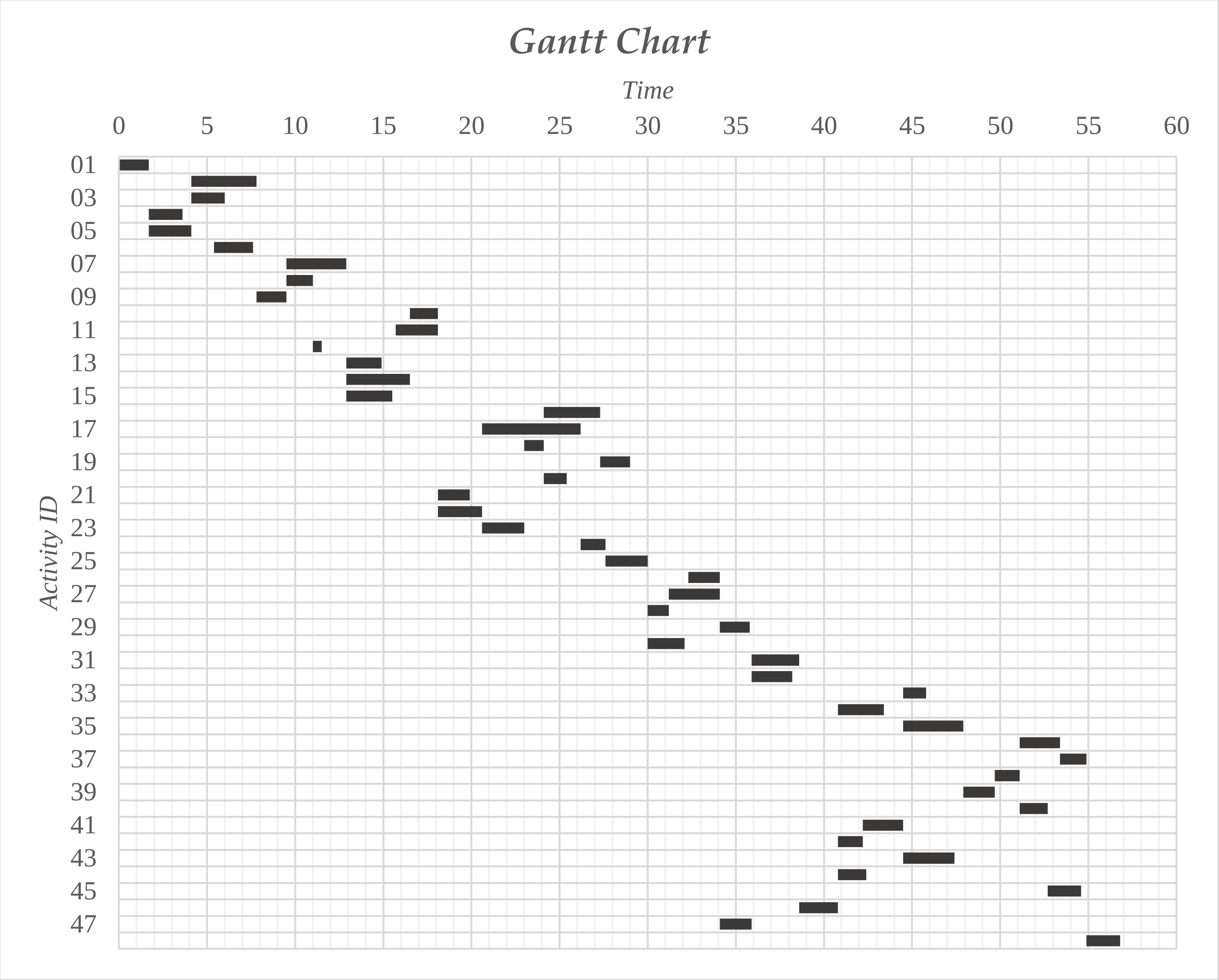

Figure 4.14 – Gantt Chart for the Optimized Work Center Schedule

Similar to the analysis conducted in Section 4.3.1, an identical set of figures is appended to illustrate the successful fulfillment of the imposed interdependency, resource, and zonal constraints. It is demonstrated through Figure 4.15 that all interdependency constraints have been successfully satisfied as the free-float value for each unique interdependency relationship is greater than or equal to zero. Figures 4.16 illustrates that a positive deviation to the aspiration criterion to type 1 resources is not required in the successful completion of all activities in 56.8 hours. On the contrary, through Figures 4.17 and 4.18, it can be demonstrated an extra type 2, and type 3 resources are required. Figures 4.20 through 4.22 have been generated, similar to that of Section 4.3.1, to ensure that the imposed zonal constraints are successfully satisfied.

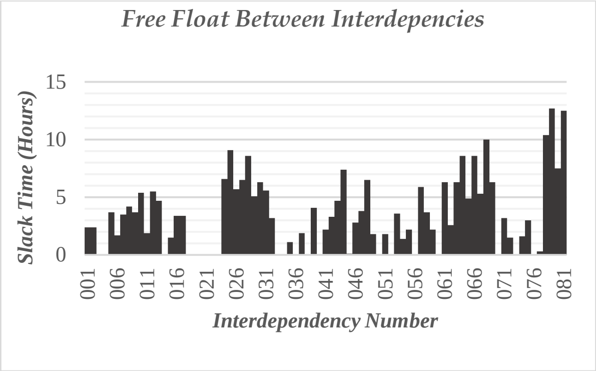

Figure 4.15 - Free Float

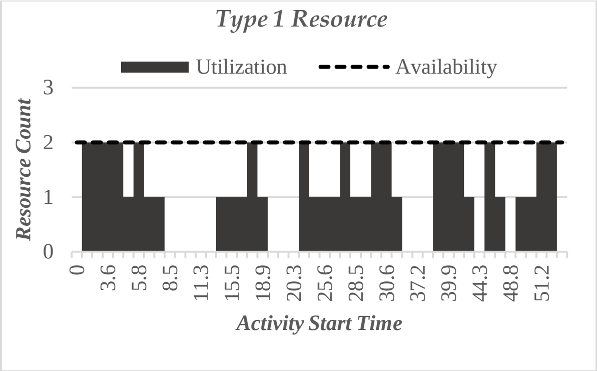

Figure 4.16 - Resource 1 Utilization

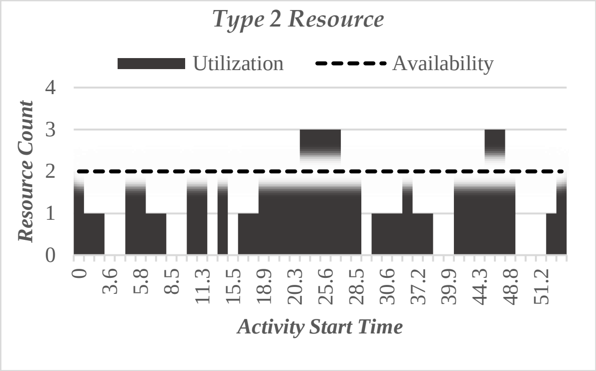

Figure 4.17 - Resource 2 Utilization

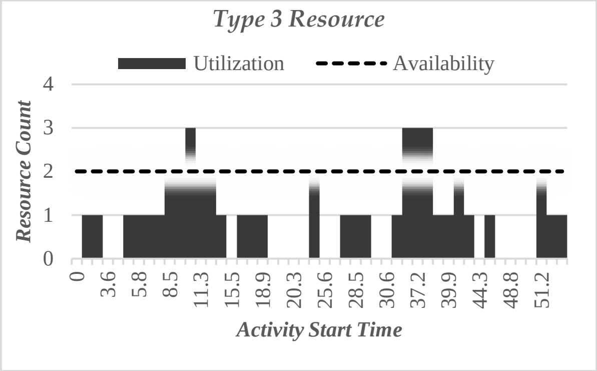

Figure 4.18 - Resource 3 Utilization

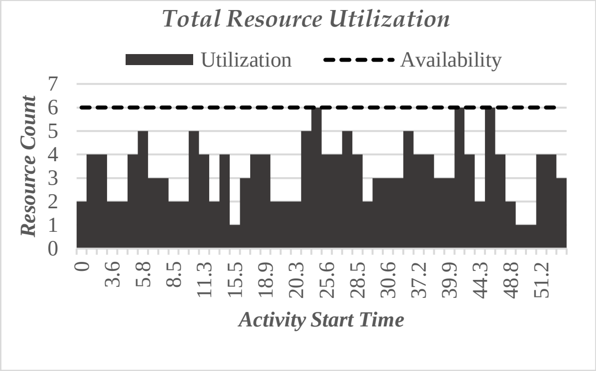

Figure 4.19 - Overall Resource Utilization

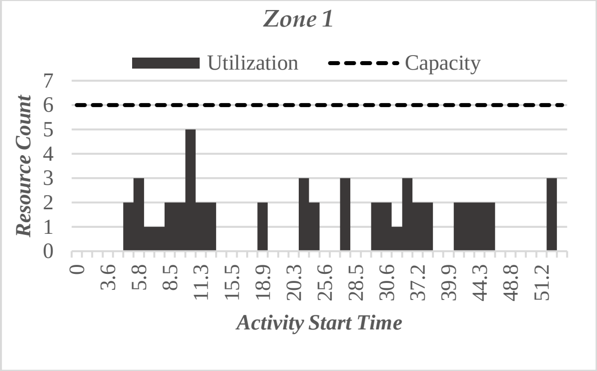

Figure 4.20 - Zone 1 Utilization

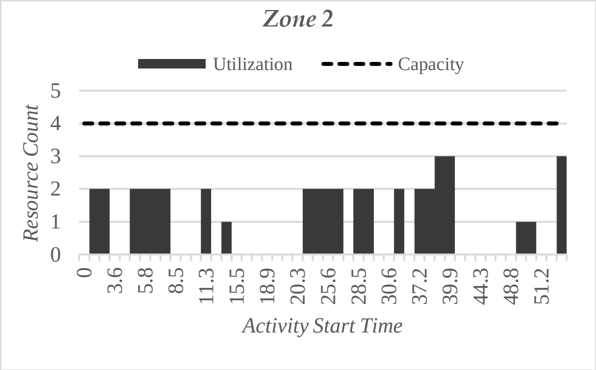

Figure 4.21 - Zone 2 Utilization

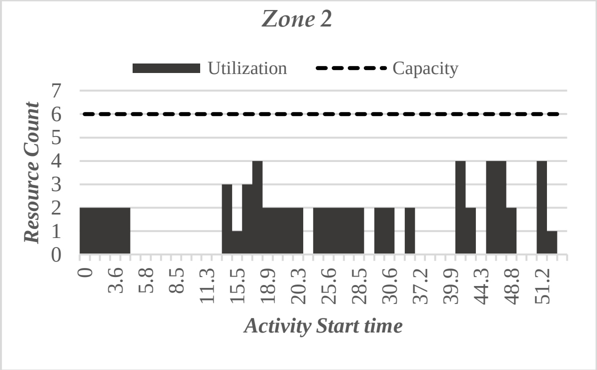

Figure 4.22 - Zone 3 Utilization

4.4. Chapter Summary

Through this chapter, a novel approach for modeling and solving large-scale scheduling problems in LVLVPS is proposed. Despite the scholarly advancements in sequencing and scheduling optimization methodologies and heuristic for a wide range of production systems, limited research has been reported on mathematical programming and heuristic approaches for modeling and solving scheduling problems in LVLVPS. The proposed multi-objective continuous-time mathematical programming models and the GA are developed to model the characteristics and constraints inherent in such production systems. To validate and verify the proposed metaheuristics, a case study of a work center in the final assembly line of a narrow body dual-engine aircraft was conducted. This case study concludes that the proposed optimization models are effective in the modeling of complex and large-scale scheduling problems in LVLVPS. The two problem types prescribed in this paper are validated through this case study, where travel work, referring to the omission of incomplete activities, may be permitted or prohibited depending on the nature of work assigned to the work center. In scenarios where through the strict enforcement of time and resources, travel work is permitted, the algorithm searches for the optimum sequence that minimizes the number of resources required in the completion of the maximum number of resources. On the contrary, in scenarios prohibiting travel work, the proposed mathematical programming model and the GA searches the solution space for the optimum activity execution sequence that minimizes the positive deviation of the work center completion time to the imposed takt time, while minimizing the positive deviation to resource budgets, in completion of the pre-defined statement of work.

CHAPTER 5

CONCLUSION

This dissertation aims on modeling and solving a wide variety of large-scale and complex work center scheduling problems in LVLVPS. Despite the notable scholarly advancements in sequencing and scheduling optimization of a wide range of scheduling problems, limited research has been reported on mathematical programming and heuristic approaches for solving scheduling problems in LVLVPS. Furthermore, while LVLVPS is a hybrid form of HVLVPS and LVHVPS, exhibiting characteristics and features inherent in both production systems, the methodologies cannot be effectively adopted in solving scheduling problems in LVLVPS. This is primarily due to deficiencies in capturing key characteristics and constraints inherent in such production systems.

5.1. Summary of Contributions & Managerial Insight

In Chapter 2, a suite of mixed-integer linear multi-objective mathematical programming models is proposed for modeling of discrete-time single work center scheduling problems. The initial model proposed in this chapter is aimed at minimizing the number of resources required in completion of the maximum number of activities, yielding to a minimum amount of travel work, in scenarios where the strict enforcement of time and resource constraints may lead to the infeasibility in completion of all activities on-time and on-budget. The second mathematical programming model formulated in this chapter is aimed at minimizing the positive deviation to the aspiration criteria to time and resource budgets in the completion of the entire statement of work, in scenarios where a work center *must* complete the pre-defined statement on work in-station. As prescribed throughout the dissertation, such constraints may be imposed to work centers where a major structural assembly must be executed prior to moving of the assembly to downstream work center, an example of which includes the final assembly of the wing-box to the center fuselage or the final assembly of the main landing gear or the nose landing gear to the cabin. It can be demonstrated through these examples that travel work may compromise the structural integrity of the product. The third and the final mathematical programming model proposed is this chapter is intended for analyzing a work center's capacity and evaluating its capabilities, given a pre-defined statement of work through minimizing the makespan for completion of the work package through the complete saturation of its resources. This mathematical model is intended to be used to justify the need for multi-parallel work centers in scenarios where the minimum makespan obtained through the resource saturation exceeds the desired takt times, which leads to the optimization models proposed in the following chapter. In scenarios where the incorporation of parallel work centers is not feasible, this mathematical programming model can be adapted and employed to raise a flag, indicating an early risk through a robust mathematical model. The three mathematical programming models formulated and proposed in this chapter are validated and verified through a real-world case study of the final assembly line of a business aircraft. Each of the mathematical models were applied on a work station, and the results demonstrated that the proposed optimization models are effective in modeling and solving complex and large-scale industrial problems.

While the mathematical programming models proposed in Chapter 2 were found effective in resulting in optimal solutions, the problem of scheduling problems in LVLVPS extends far beyond discrete-time single work center scheduling problems. In Chapter 3, a novel set of mixed-integer linear multi-objective mathematical programming models are formulated and proposed for single and multi-parallel work center scheduling problems with shared or dedicated resources. In case of dedicated resources, the acquisitioned resource pool is dedicated to a work center, whereas a common resource pool is shared between parallel work centers in the case of the shared resource problem. Similar to Chapter 2, a suite of mathematical programming models for discrete-time work center scheduling problems are formulated and proposed. The initial model is proposed for scenarios where travel work is permitted, where the on-time completion of activities with the budgeted resources may not be feasible, and deviation to the time and resource constraints are not permitted. To tackle such problem, the proposed optimization models are aimed at minimizing the required number of resources in completion of the maximum number of activities, where the latter takes precedence. Incomplete activities are then scheduled to be completed in downstream work centers. As such, in adapting this optimization model, work centers must be optimized starting from the first work center moving downstream, to ensure that travelled activities are scheduled to completed prior to their predecessors (if any) in subsequent work stations. The second mathematical model is proposed for scenarios where travel work is prohibited and the work center is ultimately responsible to complete the pre-defined statement of work, aimed at minimizing the positive deviation to the aspiration criteria to time and resource budgets, while maximizing the negative deviation to the aspiration criterion to resource. As such the mathematical programming model solves for a schedule that yields to the minimum makespan for completion of all activities that is greater than or equal to the takt-time, while minimizing overall resource requirements. The proposed optimization models are applicable to single and multi-parallel work center scheduling problems with shared or dedicated resource pools. In the case of shared resource pools, the proposed mathematical models aim to minimize the overall resource requirements for the department, comprised of a set of identical parallel work centers. Furthermore, the proposed optimization models are modular, where constraints can be added or removed without jeopardizing the integrity of the mathematical models. A real-world case study of a work center in the final assembly line of a global leader in the aerospace industry was conducted. The scheduling problems were solved to optimality, and the results obtained through the integration of a shared resource pool were compared against the use of a dedicated resource pool per work center. It was found through this case study that not only the proposed optimization models are effective and efficient in solving large-scale industrial problems, and that the integration of shared resource pools in the case of multi-parallel work centers can lead to substantial cost savings in the form of reduced labor, directly impacting the organization's bottom line.

The mathematical programming models proposed in Chapters 2 and 3 were found to be effective in modeling and solving large-scale scheduling problems in LVLVPS. However, their incompatibility in capturing continuous-time processing times and planning horizon suggested a potential deficiency in their application in the industry. Most often activity processing times are continuous, as the nature of the work does not always lend itself to discrete-time processing times. To tackle this problem, a suite of continuous-time multi-objective mathematical programming models and metaheuristics are proposed in Chapter 4. The initial model, similar to that of Chapter 2, is aimed at minimizing the required number of resources in completion of the maximum number of activities, where the latter takes precedence. This mathematical programming model is proposed to be used in scenarios where the strict enforcement of time and resource constraints may lead to infeasibility in completion of all activities on-time and on-budget. However, to ensure efficiency in modeling and solving the prescribed continuous-time work center scheduling problem a novel Genetic Algorithm is formulated and proposed to tackle large-scale industrial problems. The proposed GA starts with an initial random population and searches through the solution space through a single-point crossover strategy, resulting in incrementally better results. The proposed GA was formulated in MATLAB and was employed in solving a real-world case study of a work center scheduling problem on the final assembly line of a narrow-body green aircraft. While this model was solved to optimality, the proposed mathematical programming model and GA could not model and solve scheduling problems mandating the completion of the imposed statement of work in-station. As such, a second mathematical programming model was formulated and proposed, accompanied with a novel GA, aimed at minimizing the positive deviation to the aspiration criteria to time and resource budgets while maximizing the negative deviation to the aspiration criterion to resources, thus minimizing overall resource requirements. The proposed GA was found effective and efficient in solving large-scale work center scheduling problem mandating the completion of the imposed statement of work in-station. It is thus concluded through this dissertation, that the proposed optimization models are effective and efficient in modeling and solving a wide range of scheduling problems with discrete-time or continuous-time planning horizons and processing times, in departments comprised of single or multi-parallel work centers with shared or dedicated resource pool. However, while the foundation for such scheduling problems are set through this dissertation, further research and contributions to the state-of-the-art can be made in future work as described in the following section.

5.2. Future Research

The proposed mathematical programming models and metaheuristics assume equal shift loads, signifying that an equal number of resources per classification is employed in each shift. However, to further minimize overall resource requirements an unequal distribution of resources to shifts and an optimized shift pattern can be obtained. For instance, the optimum shift pattern may require the utilization of a single resource of a specific classification in only one shift, an example of which includes a work center that has been assigned with 18-hour worth of wing-tank sealing, for which a single resource can be assigned to the midnight shift. This can be represented in discrete-time as well as continuous-time and will yield to significant cost-savings and improvements compared to the mathematical programming models proposed in this dissertation.

The incorporation of operator efficiencies can also prove to improve the accuracy of the proposed optimization models. The mathematical programming models and metaheuristics proposed in this dissertation assumes equal operator efficiencies which is a known deficiency in any scheduling problem. Now that the foundation for a continuous-time single work center scheduling problem has been established, operator efficiencies can be effectively implemented. However, the mathematical programming models will have to be revised and redefined, as activities will not only be scheduled at a particular point in time but will also have to be assigned to a specific operator. This enables the operations management team to provide work packages to each employee, while being provided with a realistic schedule, developed based on the strengths and weaknesses of their resource pool.

The problem of the discrete-time multi-parallel work center scheduling problems with shared resources can further be extended and enhanced to incorporate continuous-time planning horizons and activity processing times to widen the application of the mathematical programming models. Furthermore, while shared resource pool was explored between parallel work centers, the problem can further be extended to optimize the work center schedule for adjacent work centers or work centers within the same business unit. Business units are defined as a group of work centers whose costs are rolled up to a business unit, often comprised of three to five work centers. The problem will then aim to not only minimize the resource requirements in a specific work center but to minimize the overall operating cost within a business unit. An example is the wing-tank sealer responsible to complete an 18-hour worth of work within a work center. In the previous section we discussed the potential improvement in cost given unequal shift patterns, optimized based on the assigned statement of work, which would require a single resource to be assigned to one shift in completion of the in-tank sealing work. However, with a takt-time of 6 days, working an eight-hour shift, will leave the resource idle for of the 30 hours out of the 48 available hours. This time can be spread between adjacent work centers or work centers within the same business unit to maximize resource utilization, thus significantly reducing the overall resource requirements.

Finally, a line balancing methodology can be formulated that will aim to minimize overall resource requirements in completion of the product within the specific cycle-time. This line balancing methodology must differ from the conventional line balancing exercises and algorithms, as the imposed constraints within each work center must be satisfied, in assigning activities to work centers. The formulation of this continuous-time line balancing heuristic will certainly result in the maximum gains in cost reduction relating to resource requirements, utilization and idle times. Coupled with an optimized shift pattern for each work center and resource sharing between parallel and adjacent work centers, the optimization model will have a direct on the bottom line. However, as all the activities must be included, the proposed algorithm must be much more efficient that the conventional metaheuristics. An example of which is the final assembly of an aircraft which is comprised of a network of over 3,000 unique activities.

References

[1] H.M. Wagner, An integer linear-programming model for machine scheduling, Nav. Res. Logist. Q. 6 (1959) 131–140. https://doi.org/10.1002/nav.3800060205.

[2] E.H. Bowman, The Schedule-Sequencing Problem, JSTOR. 7 (1959) 621–624.

[3] L.G. Mitten, Sequencing n Jobs on Two Machines with Arbitrary Time Lags , Manage. Sci. 5 (1959) 293–298. https://doi.org/10.1287/mnsc.5.3.293.

[4] R.J. Freeman, A Generalized Network Approach to Project Activity Sequencing, IRE Trans. Eng. Manag. EM-7 (1960) 103–107. https://doi.org/10.1109/IRET-EM.1960.5007550.

[5] G.B. Dantzig, A Machine-Job Scheduling Model, Manage. Sci. 6 (1960) 191–196. https://doi.org/10.1287/mnsc.6.2.191.

[6] A.S. Manne, On the Job-Shop Scheduling Problem, JSTOR. 8 (1960) 219–223.

[7] E.H. Bowman, Assembly-Line Balancing by Linear Programming, Oper. Res. 8 (1960) 385–389. https://doi.org/10.1287/opre.8.3.385.

[8] P. Ghoddousi, E. Eshtehardian, S. Jooybanpour, A. Javanmardi, Multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using non-dominated sorting genetic algorithm, Autom. Constr. 30 (2013) 216–227. https://doi.org/10.1016/j.autcon.2012.11.014.

[9] Y. Nie, L.T. Biegler, C.M. Villa, J.M. Wassick, Discrete time formulation for the integration of scheduling and dynamic optimization, Ind. Eng. Chem. Res. 54 (2015) 4303–4315. https://doi.org/10.1021/ie502960p.

[10] X. Lin, C.A. Floudas, S. Modi, N.M. Juhasz, Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant, Ind. Eng. Chem. Res. 41 (2002) 3884–3906. https://doi.org/10.1021/ie011002a.

[11] C.A. Méndez, J. Cerdá, I.E. Grossmann, I. Harjunkoski, M. Fahl, State-of-the-art review of optimization methods for short-term scheduling of batch processes, Comput. Chem. Eng.

30 (2006) 913–946. https://doi.org/10.1016/j.compchemeng.2006.02.008.

[12] K.J. Shaw, A.L. Nortcliffe, M. Thompson, J. Love, P.J. Fleming, C.M. Fonseca, Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem, in: Proc. 1999 Congr. Evol. Comput. CEC 1999, IEEE Computer Society, 1999: pp. 37–45. https://doi.org/10.1109/CEC.1999.781905.

[13] T. Bhatia, L.T. Biegler, Dynamic optimization in the design and scheduling of multiproduct batch plants, Ind. Eng. Chem. Res. 35 (1996) 2234–2246. https://doi.org/10.1021/ie950701i.

[14] I.H. Kuo, S.J. Horng, T.W. Kao, T.L. Lin, C.L. Lee, T. Terano, Y. Pan, An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model, Expert Syst. Appl. 36 (2009) 7027–7032. https://doi.org/10.1016/j.eswa.2008.08.054.

[15] Z. Lian, X. Gu, B. Jiao, A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan, Chaos, Solitons and Fractals. 35 (2008) 851–861. https://doi.org/10.1016/j.chaos.2006.05.082.

[16] M. Ji, Y. He, T.C.E. Cheng, Single-machine scheduling with periodic maintenance to minimize makespan, Comput. Oper. Res. 34 (2007) 1764–1770. https://doi.org/10.1016/j.cor.2005.05.034.

[17] M.A. Salido, J. Escamilla, A. Giret, F. Barber, A genetic algorithm for energy-efficiency in job-shop scheduling, Int. J. Adv. Manuf. Technol. 85 (2016) 1303–1314. https://doi.org/10.1007/s00170-015-7987-0.

[18] D. Lei, Z. Wu, Crowding-measure-based multiobjective evolutionary algorithm for job shop scheduling, Int. J. Adv. Manuf. Technol. 30 (2006) 112–117. https://doi.org/10.1007/s00170-005-0029-6.

[19] J. Behnamian, M. Zandieh, A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties, Expert Syst. Appl. 38 (2011) 14490–14498. https://doi.org/10.1016/j.eswa.2011.04.241.

[20] Y. Yin, S.R. Cheng, C.C. Wu, Scheduling problems with two agents and a linear non-increasing deterioration to minimize earliness penalties, Inf. Sci. (Ny). 189 (2012) 282–292.

https://doi.org/10.1016/j.ins.2011.11.035.

[21] Z. Zhu, R.B. Heady, Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach, Comput. Ind. Eng. 38 (2000) 297–305. https://doi.org/10.1016/S0360-8352(00)00048-6.

[22] D. Shabtay, Scheduling and due date assignment to minimize earliness, tardiness, holding, due date assignment and batch delivery costs, Int. J. Prod. Econ. 123 (2010) 235–242. https://doi.org/10.1016/j.ijpe.2009.08.012.

[23] H. Zhou, W. Cheung, L.C. Leung, Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm, Eur. J. Oper. Res. 194 (2009) 637–649. https://doi.org/10.1016/j.ejor.2007.10.063.

[24] D. Cao, M. Chen, G. Wan, Parallel machine selection and job scheduling to minimize machine cost and job tardiness, Comput. Oper. Res. 32 (2005) 1995–2012. https://doi.org/10.1016/j.cor.2004.01.001.

[25] A. Bauer, R.F. Hartl, C. Strauss, B. Bullnheimer, Minimizing Total Tardiness on a Single Machine Using Ant Colony Optimization, n.d. https://www.researchgate.net/publication/260423657 (accessed March 4, 2020).

[26] V.A. Armentano, D.P. Ronconi, Tabu search for total tardiness minimization in flowshop scheduling problems, Comput. Oper. Res. 26 (1999) 219–235. https://doi.org/10.1016/S0305-0548(98)00060-4.

[27] X. Wang, T.C.E. Cheng, Logistics scheduling to minimize inventory and transport costs, Int. J. Prod. Econ. 121 (2009) 266–273. https://doi.org/10.1016/j.ijpe.2009.05.007.

[28] G. Dobson, C. Arai Yano, Cyclic scheduling to minimize inventory in a batch flow line, Eur. J. Oper. Res. 75 (1994) 441–461. https://doi.org/10.1016/0377-2217(94)90087-6.

[29] M. LAGUNA, A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times, IIE Trans. 31 (1999) 125–134. https://doi.org/10.1023/A:1007560417495.

[30] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez, M. Ortega-Mier, Optimizing the

production scheduling of a single machine to minimize total energy consumption costs, J. Clean. Prod. 67 (2014) 197–207. https://doi.org/10.1016/j.jclepro.2013.12.024.

[31] D.S. Yamashita, V.A. Armentano, M. Laguna, Robust optimization models for project scheduling with resource availability cost, J. Sched. 10 (2007) 67–76. https://doi.org/10.1007/s10951-006-0326-4.

[32] Z. Laslo, Project portfolio management: An integrated method for resource planning and scheduling to minimize planning/scheduling-dependent expenses, Int. J. Proj. Manag. 28 (2010) 609–618. https://doi.org/10.1016/j.ijproman.2009.10.001.

[33] K. El-Rayes, D.H. Jun, Optimizing resource leveling in construction projects, J. Constr. Eng. Manag. 135 (2009) 1172–1180. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000097.

[34] M.S. El-Abbasy, A. Elazouni, T. Zayed, Generic Scheduling Optimization Model for Multiple Construction Projects, J. Comput. Civ. Eng. 31 (2017). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000659.

[35] S.M. Easa, Resource leveling in construction by optimization, J. Constr. Eng. Manag. 115 (1989) 302–316. https://doi.org/10.1061/(ASCE)0733-9364(1989)115:2(302).

[36] M. Göthe-Lundgren, J. T. Lundgren, J. A. Persson, An optimization model for refinery production scheduling, Int. J. Prod. Econ. 78 (2002) 255–270. https://doi.org/10.1016/S0925-5273(00)00162-6.

[37] C.A. Méndez, I.E. Grossmann, I. Harjunkoski, P. Kaboré, A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations, Comput. Chem. Eng. 30 (2006) 614–634. https://doi.org/10.1016/j.compchemeng.2005.11.004.

[38] J.M. Pinto, M. Joly, L.F.L. Moro, Planning and scheduling models for refinery operations, Comput. Chem. Eng. 24 (2000) 2259–2276. https://doi.org/10.1016/S0098-1354(00)00571-8.

[39] R. Karuppiah, K.C. Furman, I.E. Grossmann, Global optimization for scheduling refinery crude oil operations, Comput. Chem. Eng. 32 (2008) 2745–2766. https://doi.org/10.1016/j.compchemeng.2007.11.008.

[40] L.M. Simao, D.M. Dias, M.A.C. Pacheco, Refinery scheduling optimization using genetic algorithms and cooperative coevolution, in: Proc. 2007 IEEE Symp. Comput. Intell. Sched. CI-Sched 2007, 2007: pp. 151–158. https://doi.org/10.1109/SCIS.2007.367683.

[41] H.A. El Maraghy, Flexible and reconfigurable manufacturing systems paradigms, in: Flex. Serv. Manuf. J., Springer New York LLC, 2006: pp. 261–276. https://doi.org/10.1007/s10696-006-9028-7.

[42] G. Heike, M. Ramulu, E. Sorenson, P. Shanahan, K. Moinzadeh, Mixed model assembly alternatives for low-volume manufacturing: The case of the aerospace industry, Int. J. Prod. Econ. 72 (2001) 103–120. https://doi.org/10.1016/S0925-5273(00)00089-X.

[43] J. Linck, D.S. Cochran, The importance of takt time in manufacturing system design, SAE Tech. Pap. (1999). https://doi.org/10.4271/1999-01-1635.

[44] J. Crabill, E. Harmon, N. Grumman, D. Meadows, L. Martin, R.M. Textron, C. Miller, N. Grumman, Production Operations Level Transition To Lean Roadmap, 2000.

[45] R. Cheng, M. Gen, Y. Tsujimura, A tutorial survey of job-shop scheduling problems using genetic algorithms - I. Representation, Comput. Ind. Eng. 30 (1996) 983–997. https://doi.org/10.1016/0360-8352(96)00047-2.

[46] J.H. Blackstone, D.T. Phillips, G.L. Hogg, A state-of-the-art survey of dispatching rules for manufacturing job shop operations, Int. J. Prod. Res. 20 (1982) 27–45. https://doi.org/10.1080/00207548208947745.

[47] R. Ramasesh, Dynamic job shop scheduling: A survey of simulation research, Omega. 18 (1990) 43–57. https://doi.org/10.1016/0305-0483(90)90017-4.

[48] A. Allahverdi, C.T. Ng, T.C.E. Cheng, M.Y. Kovalyov, A survey of scheduling problems with setup times or costs, Eur. J. Oper. Res. 187 (2008) 985–1032. https://doi.org/10.1016/j.ejor.2006.06.060.

[49] Y.B. Park, C.D. Pegden, E.E. Enscore, A survey and evaluation of static flowshop scheduling heuristics, Int. J. Prod. Res. 22 (1984) 127–141. https://doi.org/10.1080/00207548408942436.

[50] S. Reza Hejazi, S. Saghafian, Flowshop-scheduling problems with makespan criterion: A review, Int. J. Prod. Res. 43 (2005) 2895–2929. https://doi.org/10.1080/0020754050056417.

[51] M. Rasti-barzoki, S. Reza, M. Mahdavi, A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs, Appl. Math. Model. 37 (2013) 4924–4937. https://doi.org/10.1016/j.apm.2012.10.001.

[52] K. Tamssaouet, S. Dauzère-Pérès, C. Yugma, Metaheuristics for the job-shop scheduling problem with machine availability constraints, Comput. Ind. Eng. 125 (2018) 1–8. https://doi.org/10.1016/j.cie.2018.08.008.

[53] M. Yazdani, A. Aleti, S.M. Khalili, F. Jolai, Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem, Comput. Ind. Eng. 107 (2017) 12–24. https://doi.org/10.1016/j.cie.2017.02.019.

[54] J.M. Novas, Production scheduling and lot streaming at flexible job-shops environments using constraint programming, Comput. Ind. Eng. 136 (2019) 252–264. https://doi.org/10.1016/j.cie.2019.07.011.

[55] R.H. Ã, Int . J . Production Economics Multi-objective job-shop scheduling with lot-splitting production, Intern. J. Prod. Econ. 124 (2010) 206–213. https://doi.org/10.1016/j.ijpe.2009.10.026.

[56] G. Moslehi, M. Mahnam, Int . J . Production Economics A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, Intern. J. Prod. Econ. 129 (2011) 14–22. https://doi.org/10.1016/j.ijpe.2010.08.004.

[57] C. Zhao, H. Tang, Int . J . Production Economics Two-machine flow shop scheduling with deteriorating jobs and chain precedence constraints, Intern. J. Prod. Econ. 136 (2012) 131–136. https://doi.org/10.1016/j.ijpe.2011.09.023.

[58] Y. Demir, Evaluation of mathematical models for flexible job-shop scheduling problems, Appl. Math. Model. 37 (2013) 977–988. https://doi.org/10.1016/j.apm.2012.03.020.

[59] A. Ham, Flexible job shop scheduling problem for parallel batch processing machine with

compatible job families, Appl. Math. Model. 45 (2017) 551–562. https://doi.org/10.1016/j.apm.2016.12.034.

[60] P. Fattahi, F. Jolai, J. Arkat, Flexible job shop scheduling with overlapping in operations, Appl. Math. Model. 33 (2009) 3076–3087. https://doi.org/10.1016/j.apm.2008.10.029.

[61] D. Shabtay, K. Arviv, Optimal robot scheduling to minimize the makespan in a three-machine flow-shop environment with job-independent processing times, Appl. Math. Model. 40 (2016) 4231–4247. https://doi.org/10.1016/j.apm.2015.11.021.

[62] M. Bank, S.M.T.F. Ghomi, F. Jolai, J. Behnamian, Two-machine flow shop total tardiness scheduling problem with deteriorating jobs, Appl. Math. Model. 36 (2012) 5418–5426. https://doi.org/10.1016/j.apm.2011.12.010.

[63] P.H. Randolph, Job Shop Scheduling A Case Study 1, 4 (1976) 463–477.

[64] M. Nawaz, E.E. Enscore, I. Ham, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, Omega. 11 (1983) 91–95. https://doi.org/10.1016/0305-0483(83)90088-9.

[65] Z. Yan, G. Hanyu, X. Yugeng, Modified bottleneck-based heuristic for large-scale job-shop scheduling problems with a single bottleneck, J. Syst. Eng. Electron. 18 (2007) 556–565. https://doi.org/http://dx.doi.org/10.1016/S1004-4132(07)60129-7.

[66] X. Huang, M. Wang, Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties, Appl. Math. Model. 35 (2011) 1349–1353. https://doi.org/10.1016/j.apm.2010.09.013.

[67] B. Cheng, S. Yang, X. Hu, B. Chen, Minimizing makespan and total completion time for parallel batch processing machines with non-identical job sizes, Appl. Math. Model. 36 (2012) 3161–3167. https://doi.org/10.1016/j.apm.2011.09.061.

[68] J. Slomp, J.A.C. Bokhorst, R. Germs, A lean production control system for high-variety/low-volume environments: A case study implementation, Prod. Plan. Control. 20 (2009) 586–595. https://doi.org/10.1080/09537280903086164.

[69] M.M. Srinivasan, S.J. Ebbing, A.T. Swearingen, Woodward Aircraft Engine Systems Sets

Work-in-Process Levels for High-Variety, Low-Volume Products, Interfaces (Providence). 33 (2003) 61-69+89. https://doi.org/10.1287/inte.33.4.61.16377.

[70] Y. Zhang, H. Chen, Knowledge-based dynamic job-scheduling in low-volume/high-variety manufacturing, Artif. Intell. Eng. 13 (1999) 241–249. https://doi.org/10.1016/S0954-1810(98)00014-4.

[71] I. Nasri, G. Habchi, R. Boukezzoula, Scheduling and Control Modelling of HVLV Systems Using Max-Plus Algebra, 2011. http://hal.univ-smb.fr/hal-00627509 (accessed March 4, 2020).

[72] R. Kolisch, R. Padman, An integrated survey of deterministic project scheduling, 29 (2001) 249–272.

[73] W. Herroelen, B. De Reyck, E. Demeulemeester, Resource-constrained project scheduling: A survey of recent developments, Comput. Oper. Res. 25 (1998) 279–302. https://doi.org/10.1016/S0305-0548(97)00055-5.

[74] L. Özdamar, G. Ulusoy, A survey on the resource-constrained project scheduling problem, IIE Trans. (Institute Ind. Eng. 27 (1995) 574–586. https://doi.org/10.1080/07408179508936773.

[75] J. Lancaster, M. Ozbayrak, Evolutionary algorithms applied to project scheduling problems - A survey of the state-of-the-art, Int. J. Prod. Res. 45 (2007) 425–450. https://doi.org/10.1080/00207540600800326.

[76] M. Ranjbar, S. Hosseinabadi, F. Abasian, Minimizing total weighted late work in the resource-constrained project scheduling problem, Appl. Math. Model. 37 (2013) 9776–9785. https://doi.org/10.1016/j.apm.2013.05.013.

[77] B. Afshar-nadjafi, Multi-mode resource availability cost problem with recruitment and release dates for resources, Appl. Math. Model. 38 (2014) 5347–5355. https://doi.org/10.1016/j.apm.2014.04.009.

[78] R. Deckro, J. Hebert, Resource Constrained Project Crashing, Omega. 17 (1989) 69–79.

[79] D. Hermel, H. Hasheminia, N. Adler, M.J. Fry, A solution framework for the multi-mode

resource-constrained cross-dock scheduling problem $, Omega. 59 (2016) 157–170. https://doi.org/10.1016/j.omega.2015.06.002.

[80] M. Tritschler, A. Naber, R. Kolisch, A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles, Eur. J. Oper. Res. 262 (2017) 262–273. https://doi.org/10.1016/j.ejor.2017.03.006.

[81] C. Kellenbrink, S. Helber, Quality- and profit-oriented scheduling of resource-constrained projects with flexible project structure via a genetic algorithm, Eur. J. Ind. Eng. 10 (2016) 574. https://doi.org/10.1504/EJIE.2016.078807.

[82] J. Li, Y. Huang, X. Niu, A branch population genetic algorithm for dual-resource constrained job shop scheduling problem, Comput. Ind. Eng. 102 (2016) 113–131. https://doi.org/10.1016/j.cie.2016.10.012.

[83] R. Zamani, An effective mirror-based genetic algorithm for scheduling multi-mode resource constrained projects, Comput. Ind. Eng. 127 (2019) 914–924. https://doi.org/10.1016/j.cie.2018.11.031.

[84] A.A.B. Pritsker, L.J. Waiters, P.M. Wolfe, Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach, Manage. Sci. 16 (1969) 93–108. https://doi.org/10.1287/mnsc.16.1.93.

[85] O. Kone, C. Artigues, P. Lopez, M. Mongeau, Event-based MILP models for resource-constrained project scheduling problems, Comput. Oper. Res. 38 (2011) 3–13. https://doi.org/10.1016/j.cor.2009.12.011.

[86] T.R. Browning, A.A. Yassine, Int . J . Production Economics Resource-constrained multi-project scheduling : Priority rule performance revisited, Intern. J. Prod. Econ. 126 (2010) 212–228. https://doi.org/10.1016/j.ijpe.2010.03.009.

[87] L. Drezet, J. Billaut, A project scheduling problem with labour constraints and time-dependent activities requirements, 112 (2008) 217–225. https://doi.org/10.1016/j.ijpe.2006.08.021.

[88] R.H. Möhring, Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time, Oper. Res. 32 (1984) 89–120.

https://doi.org/10.1287/opre.32.1.89.

[89] V. Tiwari, J.H. Patterson, V.A. Mabert, Scheduling projects with heterogeneous resources to meet time and quality objectives, Eur. J. Oper. Res. 193 (2009) 780–790. https://doi.org/10.1016/j.ejor.2007.11.005.

[90] J.L. Hurink, A.L. Kok, J.J. Paulus, J.M.J. Schutten, Time-constrained project scheduling with adjacent resources, Comput. Oper. Res. 38 (2011) 310–319. https://doi.org/10.1016/j.cor.2010.05.006.

[91] F. Mohammadipour, S.J. Sadjadi, Project cost-quality-risk tradeoff analysis in a time-constrained problem, Comput. Ind. Eng. 95 (2016) 111–121. https://doi.org/10.1016/j.cie.2016.02.025.

[92] T.A. Guldemond, J.L. Hurink, J.J. Paulus, Time-Constrained Project Scheduling, (2005) 1–16.

[93] F.F. Boctor, Resource-constrained project scheduling by simulated annealing, Int. J. Prod. Res. 34 (1996) 2335–2351. https://doi.org/10.1080/00207549608905028.

[94] K.W. Kim, M. Gen, G. Yamazaki, Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling, Appl. Soft Comput. J. 2 (2003) 174–188. https://doi.org/10.1016/S1568-4946(02)00065-0.

[95] R.L. Kadri, F.F. Boctor, An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case, Eur. J. Oper. Res. 265 (2018) 454–462. https://doi.org/10.1016/j.ejor.2017.07.027.

[96] M. Mori, C.C. Tseng, A genetic algorithm for multi-mode resource constrained project scheduling problem, Eur. J. Oper. Res. 100 (1997) 134–141. https://doi.org/10.1016/S0377-2217(96)00180-4.

[97] T. Borreguero, F. Mas, J.L. Menéndez, M.A. Barreda, Enhanced Assembly Line Balancing and Scheduling Methodology for the Aeronautical Industry, Procedia Eng. 132 (2015) 990–997. https://doi.org/10.1016/j.proeng.2015.12.587.

[98] E.W. Hans, W. Herroelen, R. Leus, G. Wullink, A hierarchical approach to multi-project

planning under uncertainty, Omega. 35 (2007) 563–577. https://doi.org/10.1016/j.omega.2005.10.004.

[99] A. Russell, S. Taghipour, Multi-objective optimization of complex scheduling problems in low-volume low-variety production systems, Int. J. Prod. Econ. 208 (2019) 1–16. https://doi.org/10.1016/j.ijpe.2018.11.005.

[100] A. Charnes, W.W. Cooper, Goal programming and multiple objective optimizations, Eur. J. Oper. Res. 1 (1977) 39–54. https://doi.org/10.1016/S0377-2217(77)81007-2.

[101] A. Charnes, W.W. Cooper, Goal programming and multiple objective optimizations: Part 1, Eur. J. Oper. Res. 1 (1977) 39–54. https://doi.org/10.1016/S0377-2217(77)81007-2.

[102] J. Spronk, J. Spronk, Interactive Multiple Goal Programming, in: Interact. Mult. Goal Program., Springer Netherlands, 1981: pp. 129–156. https://doi.org/10.1007/978-94-009-8165-2_6.

[103] C. Ter Chang, Multi-choice goal programming, Omega. 35 (2007) 389–396. https://doi.org/10.1016/j.omega.2005.07.009.

[104] M.J. Rentmeesters, W.K. Tsai, K. Lin, A Theory of Lexicographic Multi-Criteria Optimization, in: IEEE, 1996: pp. 76–79.

[105] H. Isermann, Linear lexicographic optimization, OR Spektrum. 4 (1982) 223–228. https://doi.org/10.1007/BF01782758.

[106] P.C. Fishburn, Exceptional Paper—Lexicographic Orders, Utilities and Decision Rules: A Survey, Manage. Sci. 20 (1974) 1442–1471. https://doi.org/10.1287/mnsc.20.11.1442.

[107] R.H. Arpaci, A.C. Dusseau, A.M. Vahdat, L.T. Liu, T.E. Anderson, D.A. Patterson, The interaction of parallel and sequential workloads on a network of workstations, ACM SIGMETRICS Perform. Eval. Rev. 23 (1995) 267–278. https://doi.org/10.1145/223586.223618.

[108] A. Russell, S. Taghipour, Multi-parallel work centers scheduling optimization with shared or dedicated resources in low-volume low-variety production systems, Applied Math. Model. 80 (2020) 472–505. https://doi.org/10.1016/j.rser.2016.01.089.

[109] J.S.H. Kornbluth, A Survey of Goal Programming, Omega. I (1973) 193–205.

[110] W.L. Thomas, A Survey of Goal Programming Applications, Omega. 8 (1980) 115–117.

[111] Z. Beheshti, S. Mariyam Shamsuddin, S. Mariyam Hj Shamsuddin, A review of population-based meta-heuristic algorithm Web Caching View project identification View project A Review of Population-based Meta-Heuristic Algorithm, 2013. www.i-csrs.org (accessed April 9, 2020).

[112] G. Laporte, I.H. Osman, Routing problems: A bibliography, Ann. Oper. Res. 61 (1995) 227–262. https://doi.org/10.1007/BF02098290.

[113] R.J.M. Vaessens, E.H.L. Aarts, J.K. Lenstra, A local search template, Comput. Oper. Res. 25 (1998) 969–979. https://doi.org/10.1016/S0305-0548(97)00093-2.

[114] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, Eur. J. Oper. Res. 185 (2008) 1155–1173. https://doi.org/10.1016/j.ejor.2006.06.046.

[115] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization artificial ants as a computational intelligence technique, IEEE Comput. Intell. Mag. 1 (2006) 28–39. https://doi.org/10.1109/CI-M.2006.248054.

[116] F.T.S. Chan, M.K. Tiwari, Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, I-Tech Education and Publishing, 2012. https://doi.org/10.5772/48.

[117] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (2006) 281–295. https://doi.org/10.1109/TEVC.2005.857610.

[118] F. Romeo, A. Sangiovanni-Vincentelli, A theoretical framework for simulated annealing, Algorithmica. 6 (1991) 302–345. https://doi.org/10.1007/BF01759049.

[119] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science (80-.). 220 (1983) 671–680. https://doi.org/10.1126/science.220.4598.671.

[120] J.H. (John H. Holland, Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence, MIT Press, 1992.

[121] D.E. Goldberg, J.H. Holland, Genetic Algorithms in Search, Optimization, and Machine

Learning David E. Goldberg The University of Alabama T, Mach. Learn. 3 (1979) 95–99. https://doi.org/10.1007/BF01920603.

[122] M. Srinivas, L.M. Patnaik, Genetic Algorithms: A Survey, Computer (Long. Beach. Calif). 27 (1994) 17–26. https://doi.org/10.1109/2.294849.

[123] O. Hrstka, A. Kucerova, Improvements of real coded genetic algorithms based on differential operators preventing premature convergence, Adv. Eng. Softw. 35 (2009) 237–246. https://doi.org/10.1016/S0965-9978(03)00113-3.

[124] M.M. Azevedo, J.A. Crispim, J.P. de Sousa, Flexible and reconfigurable layouts in complex manufacturing systems, in: IFIP Adv. Inf. Commun. Technol., Springer New York LLC, 2013: pp. 484–493. https://doi.org/10.1007/978-3-642-40352-1_61.

[125] L.V. Tavares, A multi-stage non-deterministic model for project scheduling under resources constraints, 1990.

[126] L.M. Schmitt, Theory of genetic algorithms, Theor. Comput. Sci. 259 (2001) 1–61. https://doi.org/10.1016/S0304-3975(00)00406-0.

[127] A. Shukla, H.M. Pandey, D. Mehrotra, Comparative review of selection techniques in genetic algorithm, in: 2015 1st Int. Conf. Futur. Trends Comput. Anal. Knowl. Manag. ABLAZE 2015, Institute of Electrical and Electronics Engineers Inc., 2015: pp. 515–519. https://doi.org/10.1109/ABLAZE.2015.7154916.

[128] J. Yang, C.K. Soh, Structural Optimization by Genetic Algorithms with Tournament Selection, J. Comput. Civ. Eng. 11 (1997) 195–200. https://doi.org/10.1061/(ASCE)0887-3801(1997)11:3(195).

[129] P. Sharma, A. University, H.A.P. Komal, Analysis of Selection Schemes for Solving an Optimization Problem in Genetic Algorithm Amit Wadhwa, 2014. www.ijcaonline.org (accessed April 9, 2020).

[130] K.-L. Du, M.N.S. Swamy, Search and Optimization by Metaheuristics, 2016. https://doi.org/10.1007/978-3-319-41192-7.

[131] I.A. Chaudhry, A.M. Khan, A.A. Khan, A genetic algorithm for flexible job shop

scheduling, in: Lect. Notes Eng. Comput. Sci., 2013: pp. 703–708. https://doi.org/10.1109/robot.1999.772512.

[132] H. Ouerfelli, A. Dammak, The genetic algorithm with two point crossover to solve the resource-constrained project scheduling problems, in: 2013 5th Int. Conf. Model. Simul. Appl. Optim. ICMSAO 2013, 2013. https://doi.org/10.1109/ICMSAO.2013.6552686.

[133] F.T.S. Chan, S.H. Chung, P.L.Y. Chan, An adaptive genetic algorithm with dominated genes for distributed scheduling problems, Expert Syst. Appl. 29 (2005) 364–371. https://doi.org/10.1016/j.eswa.2005.04.009.

[134] O. Hasançebi, F. Erbatur, Evaluation of crossover techniques in genetic algorithm based optimum structural design, Comput. Struct. 78 (2000) 435–448. https://doi.org/10.1016/S0045-7949(00)00089-4.

[135] E. Semenkin, M. Semenkina, Self-configuring genetic algorithm with modified uniform crossover operator, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer, Berlin, Heidelberg, 2012: pp. 414–421. https://doi.org/10.1007/978-3-642-30976-2_50.