Mathematical Programming and Metaheuristics for Solving Continuous-Time Scheduling Optimization Problems in Low-Volume Low-Variety Production Systems

Arya Russel[1], Ph.D.
Sharareh Taghipour [1,], Ph.D., Associate Professor and Canada Research Chair in Physical Asset Management, Ph.D.*
** Corresponding author: sharareh@torontomu.ca*
Mani Sharifi[1], Ph.D., Postdoctoral Fellow

[1] The Reliability, Risk, and Maintenance Research Laboratory (RRMR Lab), Mechanical and Industrial Engineering Department, Toronto Metropolitan University, Toronto, Ontario, Canada

Abstract

Despite prominent scholarly advancements in operations research, limited literature has been reported on mathematical and heuristic approaches for scheduling low-volume low-variety production systems. This paper proposes a new approach for modeling and solving large-scale sequencing and scheduling problems in Low-Volume, Low-variety production systems. The proposed non-linear mathematical programming models and genetic algorithms are subject to time and resource constraints, aimed at maximizing the number of activities completed in-station or intended to minimize the positive deviation to the aspiring time and resources budgets in scenarios where the allocated work package must be completed in-station. The proposed algorithms are compatible with discrete and continuous-time scheduling problems and are found to be effective in modeling characteristics and constraints inherent in Low-Volume, Low-Variety production systems. To validate the proposed models, a real-world case study of a work center in the final assembly line of a private jet aircraft is conducted.

Keywords

Scheduling Optimization; Goal Programming; Genetic Algorithms; Time and Resource Constrained; Aerospace Scheduling Problem.

1. Introduction

Low-Volume Low-Variety Production Systems (LVLVPS) are classified as a hybrid of High-Volume Low-Variety Production Systems (HVLVPS) and Low-Volume High-Variety Production Systems (LVHVPS). Sequencing and scheduling optimization models in LVLVPS aim to minimize the cost or resource requirements in the on-time completion of the assigned work package. Products assembled in LVLVPS follow identical processing orders through a series of manufacturing cells responsible for completing the assigned work package with the budgeted resources over the span of the takt time, examples of which include the final assembly of aircraft and heavy aerostructures. The flow of products in LVLVPS is similar to that of HVLVPS, commonly referred to as Flow-Shops and Job Shops. Products assembled in HVLVPS flow through a series of machines, where each can perform a task or a series of tasks. Reported mathematical programming models and heuristics for solving scheduling problems in HVLVPS are aimed at minimizing lateness, tardiness, or makespan in the completion of all orders referred to as jobs [1–5]. Project Scheduling problems can further be classified as Resource-Constrained Project Scheduling Problems (RCPSP) or Time-Constrained Project Scheduling Problems (TCPSP) to minimize makespan [6,7] and cost through reduced resource requirements, respectively [8,9]. However, despite identical resource profiles and capabilities, optimization models reported for modeling scheduling problems in LVHVPS cannot be directly adopted to solve scheduling problems in LVLVPS.

Since limited research has been reported on scheduling optimization approaches for LVLVPS, this paper fills the gap in the current literature through the formulation of a new set of mathematical programming models and a novel Genetic Algorithm (GA) proposed for solving large-scale scheduling problems. The proposed continuous-time non-linear mathematical programming models adopt a lexicographic goal programming approach in formulating a priority-based multi-objective optimization model that captures characteristics inherent in LVLVPS. Two new mathematical programming models are proposed in this paper. The first model is aimed at the on-time completion of the maximum number of activities in scenarios where the stringent enforcement of time and resource constraints may result in travel work. In the context of cellular manufacturing, traveled work refers to a subset of activities that have been assigned to the work center, which are found infeasible to be completed in-takt with the budgeted number of resources. While work centers can complete activities traveled from upstream work centers, the task is often achieved through additional costs due to setup and additional labor requirements. To tackle scenarios mandating the completion of all activities, a second optimization model is formulated and proposed aimed at minimizing the positive deviations to the aspiration criteria to time and resources. Two GA metaheuristics have also been developed to tackle large-scale scheduling problems in LVLVPS. At the same time, discrete-time models and methods have been investigated by many researchers [10–14]. The objective and novelty of this manuscript lie in formulating a multi-objective continuous-time linear mathematical model, subject to resource, time, precedence, zonal, concurrency, and non-concurrency constraints. For this purpose, we draw on previous contributions to state-of-the-art highlighting the efficiency and effectiveness of GA in solving scheduling problems [15–20]. To ensure compatibility with real-world scheduling problems, the proposed optimization models are validated and verified through a real-world case study of an aerospace assembly line.

The structure of this paper is as follows: In Section 2, the related literature is reviewed, highlighting their contributions and deficiencies in modeling and solving scheduling problems in LVLVPS. The characteristics and constraints inherent in LVLVPS and activity attributes and assumptions made in the mathematical modeling of the proposed optimization models are explored in Section 3. Section 4 summarizes the solution approach with a detailed procedure for applying the proposed GA. The proposed optimization models are then validated through a case study of a scheduling problem in an assembly line of a narrow-body dual-jet aircraft in Section 5. This paper is concluded in Section 6, with a review of the contributions to the state of the art and the next steps.

2. Literature Review

Scheduling optimization problems have been subject to extensive research since the early work of [21] with a focus on minimizing makespan, tardiness, and resource requirements. However, limited research has been reported on mathematical programming approaches to large-scale scheduling problems in LVLVPS. The optimization models developed for modeling and solving the job shop scheduling problem cannot be directly incorporated into solving scheduling problems in LVLVPS. This is primarily due to differences in the nature of work and resource capabilities. Project scheduling problems are generally classified as either time or resource-constrained scheduling problems to minimize resource requirements or makespan, respectively [22–25]. 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 [15,26]. The GA developed to solve such problems has adapted a unique chromosome representation to capture precedence and is found effective and efficient in solving single and multi-mode scheduling problems with resource constraints [27]. However, despite the similarities exhibited between the two production systems, the optimization models developed for solving RCPSP and TCPSP are deficient in modeling and solving scheduling problems in LVLVPS. This is primarily due to the strict enforcement of both time and resource constraints in the case of LVLVPS. This can be demonstrated through a start/end event-based RCPSP approach, developed by [28] and adopted by [29] in solving scheduling problems in the final assembly line of aircraft. While this model was solved optimally, the results are not practical due to deficiencies in capturing the imposed time constraints. The omission of time constraints may result in early completion of work, leaving the resources idle for a period of time equivalent to the difference between the obtained makespan and the imposed takt time. A series of discrete-time mixed-integer linear mathematical programming models were developed in our previous work, aimed at minimizing the number of incomplete activities in scenarios where the strict enforcement of time and resources are in effect [30,31]. While the mathematical programming models were found to be effective in modeling large-scale scheduling problems in LVLVPS, the incompatibility to modeling continuous-time problems has led to further research. This paper proposes a new set of non-linear continuous-time mathematical programming models, in addition to a novel GA, for solving scheduling problems in LVLVPS.

3. Problem Description & Mathematical Modeling

The characteristics, constraints, and assumptions inherent in LVLVPS are investigated and explored throughout this section. As briefly discussed, products assembled in LVLVPS exhibit minimal variation in product configuration. Each work center is 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 for completing the assigned work package, comprised of N single-mode and M multi-mode activities over the span of the imposed takt time T_{Max}, where multi-mode activities are a subset of activities that are permitted to be crashed or fast-tracked by allocating additional resources. Note that In defining the problem, we must ensure that the indices for all single-mode activities, as well as the initial mode of multi-mode activities, are defined over the set $\{1,\ldots,N\}$. The secondary modes of multi-mode activities are defined over the set $\{N+1,\ldots,M\}$. 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-time processing times p_j, and are assumed to be non-preemptive, signifying that once an activity has started, it cannot be paused or interrupted and must be progressed to completion. Activities may require multiple resources of distinct classifications, where resource requirements for activity j are denoted by $\sum_l w_{jl}$. 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 before the completion of its predecessors j', while lag time refers to an imposed delay between the successor and predecessor(s). To ensure accurate modeling of characteristics inherent in LVLVPS, a zonal constraint is imposed, where zones $i \in \{1,\ldots,I\}$ represent the physical location of work, subject to a maximum capacity Z_i, representing the maximum number of resources $\sum_j \sum_l w_{jl} y_{ij}$ 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 the 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. Figure 1 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. Note that this model will only assign activities to the work center that can be completed within the available planning horizon, $x_j = 1$. The remainder of the activities, which include activities that can be started but cannot be completed within the takt-time, and activities that cannot be started within the takt-time, must be scheduled for completion in downstream work centers.

Insert Fig. 1 here

The high-level production layout depicted in Figure 2 demonstrates the product flow of an aircraft through work centers in an assembly line. The final assembly of aircraft is considered 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 moving nature of the assembly line requires an upper bound on time. Thus, it is crucial to formulate an optimization model that satisfies the strict enforcement of time and resource constraints, which may only be feasible through traveling incomplete activities. On the contrary, there exist scenarios where a milestone must be met within a work center to ensure the safe movement of the product.

Insert Fig. 2 here

The sets, parameters, and variables defined for formulating the proposed optimization models are appended as follows, where the primary decision variable is the starting time S_j of activity j. In scenarios where travel work is permitted, a hard constraint exists 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 $\left[S_j + p_j\right]$ within the span of the planning horizon $\left[S_j + p_j\right] \leq T_{Max}$, and $x_j = 0$ if activity j cannot be completed with the budgeted resources over the span of the takt time. Note that all sets and parameters must be collected in their entirety to ensure practical solutions to real-world problems and are to be presented in the form of matrices, integers, or continuous values where applicable. In Section 3.1, 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 3.2 is dedicated to formulating constraints and provides a detailed description and use cases for each constraint.

- Indices

Notation	***Description***
j, j', j'', k	Activity index
α	Index for single-mode activity
β	Index for multi-mode activity
l	Resource index
i	Zonal index

- Sets

Notation	***Description***
$j \in \{1, \dots, N+M\}$	Set of activities
$j', j'', k \equiv j$	The equivalent set of activity j
$\alpha \in \{1, \dots, N\}$	Set of single-mode activities
$\beta \in \{N+1, \dots, N+M\}$	Set of multi-mode activities
$l \in \{1, \dots, L\}$	Set of resource classes
$i \in \{1, \dots, I\}$	Set of zonal classes

- Parameters

Notation		Description
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 ofaActivity 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	$Z_i > 0$	The capacity of zone i
NC_{jk}	$NC_{jk} \in \{0,1\}$	Non-Concurrency between activities j and k
CS_{jk}	$CS_{jk} \in \{0,1\}$	Concurrent start between activities j and k
CF_{jk}	$CF_{jk} \in \{0,1\}$	Concurrent finish between activities j and k
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

Notation		Description
x_j	$x_j \in \{0,1\}$	Completion of activity j within the planning horizon
π_j	$\pi_j \in \{0,1\}$	Non-completion of activity j within the planning horizon
W_l	$W_l \geq 0$	Resource requirement from Pool l
T	$0 \leq T \leq T_{Max}$	Time & T_{Max} represents The planning horizon
$h_{jj'}$	$h_{jj'} \in \{0,1\}$	Used in conjunction with non-concurrency constraint, where $h_{jj'} = 1$ if activity j is started prior to the completion time of activity j', and $h_{jj'} = 0$ otherwise.

- Decision Variables

Notation		Description
$\delta_{wl}^{\pm}$	$\delta_{wl}^{\pm} \geq 0$	Deviation to type l resources
$\delta_T^{\pm}$	$\delta_T^{\pm} \geq 0$	Deviation to time constraint
δ_α^{-}	$\delta_\alpha^{\pm} \geq 0$	Deviation to xcheduled single-mode activities
δ_β^{-}	$\delta_\beta^{\pm} \geq 0$	Deviation to scheduled multi-mode activities
S_j	$S_j \geq 0$	Start time of activity j

3.1. Objective Functions

The main difference between the resource-constrained shortest path problem (RCSP) and the investigated problem in this paper is that the developed model is a time and resources constrained problem, making it hard to handle. Since in a RCSP, there are no restrictions for time, in any circumstances, we have a feasible reason. However, in time and resource-constrained problems, there is no certainty of having a feasible region. To put it better, the scarcity of time and resources and precedence, zonal, and other temporal constraints inherent in LVLVPS may result in scenarios where there does not exist a feasible solution for scheduling all activities. To tackle the potential infeasibility of this problem, two new objective functions are formulated and proposed. The objective function (1) adopts a priority-based lexicographic goal programming approach and aims to minimize the required number of resources $\sum_l W_l$ in completing the maximum number of activities, subject to an upper limit on time and resources. Deviation variables are used in the mathematical modeling of this objective function, where $\left[\delta_\alpha^+ + \delta_\beta^+\right]$ 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. The first priority objective, represented by P_1 is thus aimed at minimizing the number of incomplete activities, while the second priority objective, represented by P_2, maximizes the negative deviation to the aspiration criterion to the summation of resource budgets $\sum_l \delta_{wl}^-$. The aspiration criteria for single-mode, multi-mode, and resource requirements are set to N, M, and W_l^{Max}, respectively, through constraints (24), (26), and (6). In solving this objective, following the lexicographic goal programming approach, the problem is first solved for the first priority objective P_1. The resultant objective value is then added as a constraint prior to solving the problem for the second-priority objective P_2.

$$Lex\ Min Z = \left[P_1\left(\delta_\alpha^- + \delta_\beta^-\right), P_2\left(-\sum_l \delta_{wl}^-\right)\right] \tag{1}$$

There also exist scenarios where travel work is prohibited, and the work center must complete the pre-defined statement of work. The objective function (2) is formulated and proposed to be used in these scenarios, where time and resources are considered soft constraints. The priority-based lexicographic 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 (8) and (7), respectively, where the aspiration criterion for time is set to the takt time T_{Max}.

$$Lex\ Min Z = \left[P_1\left(\delta_T^+\right), P_2\left(\sum_l \delta_{wl}^+\right), P_3\left(-\sum_l \delta_{wl}^-\right)\right] \tag{2}$$

3.2. Constraints

Interdependencies between successor and predecessor activities are 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 (3) enforces the start time of the successor activity $S_{j'}$ to be greater than or equal to the finish time of its predecessors $\left[S_j+p_j\right]$, 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 (20) in conjunction with Big M, represented as B, where $\pi_j+x_j=1$.

$$S_{j'}+B\pi_j \geq \left(S_j+p_j\right)+L_{jj'} \qquad \forall \quad \left\{ j, j'=1,\ldots,N+M \quad : P_{jj'}=1,\ x_j=1 \right\} \tag{3}$$

The resource constraints are imposed through constraint sets (4) - (6). Constraint set (4) 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 (5), applicable only in scenarios where travel work is permitted and deviation to the aspiration criterion to resource availability and budgets are prohibited. Constraint set (6), however, allows deviation to the aspiration criterion through deviation variables $\left[\delta_{wl}^{+},\delta_{wl}^{-}\right]$, representing the positive and negative deviation of resource requirements to resource availabilities, respectively.

$$\sum_{l} x_j w_{jl} + \sum_{j'} x_{j'} w_{j'l} \leq W_l \qquad \forall \quad \left\{ j, j'=1,\ldots,N+M \quad : S_{j'} \leq S_j \leq S_{j'}+p_{j'} \right\} \quad \forall \quad l=1,\ldots,L \tag{4}$$

$$W_l \leq W_l^{Max} \qquad \forall \quad l=1,\ldots,L \tag{5}$$

$$W_l+\delta_{wl}^{-}-\delta_{wl}^{+}=W_l^{Max} \qquad \forall \quad l=1,\ldots,L \tag{6}$$

To ensure the on-time completion of all activities or enforce an upper bound on the available planning horizon, constraints (7) and (8) are imposed. Constraint (7) is formulated and is proposed to be used in conjunction with objective function (1), where travel work is permitted. Thus, this constraint will 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 from the aspiration criterion to time. Deviation variables $\left[\delta_T^{+},\delta_T^{-}\right]$ are thus introduced and are quantified through constraint (8), representing the positive and negative deviation to the aspiration criterion for time T_{Max}, respectively.

$$T \le T_{Max} \tag{7}$$

$$T + \delta_T^- - \delta_T^+ = T_{Max} \tag{8}$$

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 includes the assignment of 3 resources in an aircraft's main landing gear bay, where only two people can work safely in that area. Constraint set (9) 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 is feasible from a practical standpoint.

$$\sum_l x_j y_{ij} w_{jl} + \sum_l \sum_{j'} x_{j'} y_{ij'} w_{j'l} \le Z_i \qquad \forall \; \left\{ j, j' = 1, \ldots, N+M \quad : S_{j'} \le S_j \le S_{j'} + p_{j'} \right\} \quad \forall \; i = 1, \ldots, I \tag{9}$$

There exist scenarios where the simultaneous progression of two or more activities is prohibited due to the nature of work or factors affecting resources or product safety. Such non-concurrency constraints are imposed through constraint sets (10) and (11). Two activities j and k are identified as non-concurrent if $NC_{jj'} = 1$. Constraint set (10) ensures that the completion time $\left[S_j + p_j \right]$ of activity, j is less than or equal to the start time S_k of activity k, while constraint set (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 and denoted by B. Through the use of binary variable $h_{jj'}$, only one of the following conditions has to hold, while both constraints will be satisfied, where an activity j must be completed prior to its non-concurrent activity k, or activity j must start after the completion of its non-concurrent activity k.

$$S_j + p_j \le S_{j'} + Bh_{jj'} \qquad \forall \; \left\{ j, j' = 1, \ldots, N+M \; : NC_{jk} = 1 \right\} \tag{10}$$

$$S_j + B\left(1 - h_{jj'}\right) \ge S_{j'} + p_{j'} \qquad \forall \; \left\{ j, j' = 1, \ldots, N+M \; : NC_{jk} = 1 \right\} \tag{11}$$

Contrary to non-concurrency constraints, a mandate may exist to start or complete two or more activities concurrently. The concurrent start between activity j and k is imposed through constraint set (12), enforcing the identical start times for the two activities. Concurrent finish constraint is similarly imposed through constraint set (13), where the completion time

$\left[S_j + p_j\right]$ of activities j and k are set to be equal if $CF_{jj'} = 1$, signifying that the two activities must finish concurrently.

$$CS_{jk}\left(S_j - S_{j'}\right) = 0 \qquad \forall \ \left\{j, j' = 1, \ldots, N+M \ : CS_{jk} = 1\right\} \quad (12)$$

$$CF_{jk}\left[\left(S_j + p_j\right) - \left(S_{j'} + p_{j'}\right)\right] = 0 \qquad \forall \ \left\{j, j' = 1, \ldots, N+M \ : CF_{jk} = 1\right\} \quad (13)$$

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. The earliest start is enforced through constraint set (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 (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 (16) and (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 $\left[S_j + p_j\right]$ prior to the imposed latest completion time LF_j or after the mandated earliest completion time EF_j. Note that constraint sets (14) – (17) are categorized as optional, 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. Note that constraints (14) through (17) are categorized as optional, where they may exist scenarios in which the earliest and latest start and finish times for activities are not mandated or previously defined. These constraints may be omitted in such cases, resulting in a linear mathematical model.

$$x_j S_j \geq x_j ES_j \qquad \forall \quad j = 1, \ldots, N+M \quad (14)$$

$$x_j S_j \leq LS_j \qquad \forall \quad j = 1, \ldots, N+M \quad (15)$$

$$x_j\left(S_j + p_j\right) \leq LF_j \qquad \forall \quad j = 1, \ldots, N+M \quad (16)$$

$$x_j\left(S_j + p_j\right) \geq x_j EF_j \qquad \forall \quad j = 1, \ldots, N+M \quad (17)$$

Binary variable x_j is used in constraint sets (18) and (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 task $\left[S_j + p_j\right] \leq T_{Max}$, and $x_j = 0$ if the activity j was completed after the imposed takt time $\left[S_j + p_j\right] > T_{Max}$.

$$x_j = 1 \qquad \forall \quad \{ j = 1, \ldots, N+M \quad : S_j + p_j \le T_{Max} \} \tag{18}$$

$$x_j = 0 \qquad \forall \quad \{ j = 1, \ldots, N+M \quad : S_j + p_j > T_{Max} \} \tag{19}$$

Binary variable π_j is a function of x_j, where $\pi_j = [1 - x_j]$ as formulated through constraint set (2). This variable is used in conjunction with the Big M in the precedence constraint set (3) and represents the inverse of the on-time completion of activity j.

$$\pi_j + x_j = 1 \qquad \forall \quad j = 1, \ldots, N+M \tag{20}$$

Constraint sets (21) and (22) are formulated to ensure the on-time completion of all activities in scenarios where travel work is prohibited. Constraint set (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 (22), on the other hand, is formulated to ensure the on-time completion of multi-mode activities $j \in \beta$, where only a single model 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 vice versa.

$$x_j = 1 \qquad \forall \quad j \in \alpha \tag{21}$$

$$x_j + x_{j'} = 1 \qquad \forall \quad \{ j, j' \in \beta \quad : M_{jj'} = 1 \} \tag{22}$$

Contrary to constraint sets (21) and (22), constraint sets (23) – (26) are formulated to be used in conjunction with objective function (1), where travel work is permitted. Constraint set (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 (24). Additionally, the deviation variables for the number of completed single-mode activities $\delta_\alpha^\pm$ are defined through this constraint set.

$$x_j \le 1 \qquad \forall \quad j \in \alpha \tag{23}$$

$$\sum_{j \in \alpha} x_j + \delta_\alpha^- = N \tag{24}$$

Constraint sets (25) and (26) are similar to that of (23) and (24), aimed at multi-mode activities $j \in \beta$. Constraint set (25) enforces that only a single mode of a multi-mode activity can be scheduled, whereas activities cannot be scheduled more than once. Through constraint

(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 (24) in the objective function (1) aimed at minimizing the number of incomplete activities.

$$x_j + x_{j'} \le 1 \qquad \forall \ \left\{ j, j' \in \beta \quad : M_{jj'} = 1 \right\} \tag{25}$$

$$\sum_{j \in \beta} x_j + \sum_{j' \in \beta} x_{j'} + \delta_{\beta}^{-} = M \tag{26}$$

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 (27) and (28) are formulated and proposed. These constraint sets are complementary to precedence constraint (3) 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 (27) for single-mode activities $j \in \alpha$ and constraint set (28) for multi-mode activities $j \in \beta$.

$$x_j \ge x_{j'} \qquad \forall \ \left\{ j, j' \in \alpha \quad : P_{jj'} = 1 \right\} \tag{27}$$

$$x_j + x_{j'} \ge x_{j''} \qquad \forall \ \left\{ j, j' \in \beta, \ \ j'' = \alpha \cup \beta \quad : P_{jj''} = 1, \ \ M_{jj'} = 1 \right\} \tag{28}$$

4. Genetic Algorithm

The GA was initially introduced in [32] and later extended and described in greater detail [33]. Despite the extensive application of GA in solving a wide range of scheduling problems, most are restricted to job-shop, flow-shop, and RCPSP. Moreover, the survey study conducted by Pellerin et al. [34] shows the superiority of GA in solving RCPSP. So, this algorithm has been widely used for RCPSP (i.e., [35–40]). This section proposes a detailed procedure for initializing and solving large-scale scheduling problems using GA in LVLVPS.

4.1. Pre-processing Procedure

The mathematical formulation for calculating the scheduling order of activities is represented in equations (29) and (30). Through equation 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 (30).

$$f_{j'} = 1 \qquad \Leftrightarrow \quad \sum_{j=1}^{N+M} P_{jj'} = 0 \tag{29}$$

$$f_{j'} = m+1 \qquad � \qquad \max_{j \,:\left[P_{jj'}=1\right]} f_j = m \tag{30}$$

4.2. Chromosome Representation

The scheme of the direct representation ag the proposed chromosome is depicted in Figure 3, where each cell representing a gene comprises the activity index, 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.

Insert Fig. 3 here

4.2. Initialization

The GA starts by generating the initial population. The scheduling order numbers are initially calculated through equations (29) and (30). Thereafter, activities are scheduled based on their order numbers, whereas activities with identical scheduling order numbers are randomly scheduled to satisfy the precedence constraints.

4.4. Evaluation

Upon the complete generation of the initial population, the chromosomes representing feasible schedules are evaluated through a fitness function. The fitness value is a function of the objective function and is represented through equation (31), where λ is the selection pressure. The fitness function will result in a fitness value in the interval of $[0,1]$ for each chromosome, so the schedules with higher fitness will be superior to those of lower fitness values. This modular fitness function is compatible to be used in conjunction with objective functions (1) and (2), aimed at minimizing the required number of resources in the completion of the maximum number of activities.

$$f = e^{-\lambda \times Obj\ Funct \big/ \max[Obj\ Funct]} \tag{31}$$

4.3. Selection Strategy

The roulette wheel strategy is adopted as the selection strategy. The parents' selection is executed by 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 (32), where μ represents the chromosome number, and N_P denotes the size of the population [33].

$$P(\mu) = \frac{f(\mu)}{\sum_{\mu=1}^{N_P} f(\mu)} \tag{32}$$

4.4. Crossover

Figure 4 demonstrates the crossover strategy adapted in the proposed GA, where a simplified pair of chromosomes comprised of 10 activities is selected to reproduce two offspring. The crossover operation is executed $N_P \times P_C$ times, where P_C represents the probability of crossover. This crossover strategy leads to the reproduction of $2N_P \times P_C$ offspring, as each crossover results in the generation of two new offspring.

Insert Fig. 4 here

4.5. Mutation

The mutation strategy devised for solving scheduling problems in LVLVPS comprises 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 more significant than 0.5, a new solution is generated by applying the initialization methodology proposed in Section 4.2. This procedure ensures that the algorithm avoids falling into local optimums and accelerates the search for a solution with improved fitness.

4.6. New Generation

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 on to the next iteration executing procedures outlined in Sections 4.3 through 4.6. Once the pre-determined termination criterion is met and 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.

5. 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 for completing 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 imposed in response to the market demand. The takt time for this case study is set to $T_{Max} = 48$ hours, equivalent to 6 shifts, or two days operating on three daily shifts. The work center understudy is responsible for the assembly of the cockpit and the empennage to the center fuselage, a statement of work 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 5 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 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 5 is the activity-on-node network diagram for the statement of work assigned to this work center. It can be demonstrated through this figure 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 developing a generic metaheuristic compatible with solving 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.

Insert Fig. 5 here

This case study is solved using a GA, coded in MATLAB on a 64-bit Windows operating system with an Intel 6th generation i7 processor, operating at 2.6GHz with 16.0GB RAM. We used the response surface methodology to tune the algorithm's parameters, which are the number of iterations (NOI), size of the initial population ($npop$), crossover probability (p_c), and mutation probability (p_m). The ranges of these parameters and their optimal values are presented in Table 1.

Insert Table 1 here

As is presented in Table 1, the optimal parameters of the GA are $Npop=70, p_c=0.7$ and $p_m=0.3$. To have a fair evaluation between all combinations of the parameters in this stage, we set the GA to check 100,000 individuals. So, the optimal value of NOI is calculated as follows: $NOI=100'000/npop=100'000/70\approx 1400$. However, when we started solving the main instance, the algorithm converged to its optimal value after approximately 800 interactions (in different algorithm runs). So, we reduced the range of NOI to 1000 to reduce the computational time of the algorithm.

Two scenarios are considered for solving the model, one with Travel Work Permitted and the second with Travel Work Prohibited. For each scenario, the GA is run 30 times, and the best case, the worst case, and the range of outputs in the objective function and the computational times are reported in figure 6.

Insert Fig. 6 here

As presented in Figure 6a, the number of traveled jobs in scenario 1 is between 9 to 12 in all iterations of GA. Figure 6b illustrates that the jobs' completion time of 30 iterations of GA is between 51.1 and 56.8 hours. Moreover, Figure 6c shows no significant difference between the GA elapsed time in 30 iterations of GA for both scenarios. The results of the best case for both scenarios are reported in Table 2.

Insert Table 2 here

Figures 7 and 8 illustrate the Gant chart for both scenarios.

Insert Figures 7 and 8 here

Figures 9 through 16 are the comparative illustration of both scenarios. Figure 9 illustrates the successful satisfaction of the interdependency constraints, where the x-axis represents the interdependency identifier, and the y-axis represents the slack time in hours. The slack times equal to or greater than zero are evidence that a successor activity has started upon or after completing a predecessor activity.

Insert Fig. 9 here

Figures 10, 11, and 12 demonstrated resource usage and utilization for type 1, type 2, and type 3 resources, 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.

Insert Figures 10 to 12 here

Note that resource requirements are calculated at the starting time of each activity, represented through the x-axis. Figure 13 highlights the overall resource availability and utilization.

Insert Fig. 13 here

.

It is demonstrated through Figures 9 through 13 that all resource constraints are successfully satisfied as the usage bars do not exceed the dashed line, representing resource constraints at any time interval. Figures 14, 15, and 16 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 assigned to each zone at the starting time of each activity, validating that all zonal constraints were successfully satisfied. Thus, 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 the potential traveling of incomplete activities.

Insert Figures 14 to 16 here

6. Conclusion

This paper proposes a novel approach for modeling and solving large-scale scheduling problems in LVLVPS. Despite the scholarly advancements in sequencing and scheduling optimization methodologies and heuristics 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 non-linear multi-objective continuous-time mathematical programming models and the GA are developed to accurately model the characteristics and constraints inherent in such production systems. A case study of a work center in the final assembly line of a narrow body dual-engine aircraft was conducted to validate and verify the proposed metaheuristics. This case study concludes that the proposed optimization models effectively model 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 travel work is permitted through the strict enforcement of time and resources, the algorithm searches for the optimum sequence that minimizes the number of resources required to complete 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. This research can be further extended to incorporate efficiency factors, shared resource pools between multiple parallel work centers, as well as shared resource pools between adjacent work centers. We hereby hope to have advanced and motivated further research in scheduling optimization strategies in LVLVPS.

Data Availability Statement

Due to the nature of this research, supporting data is not available.

Acknowledgments

Will be announced in case of acceptance.

References

1. Zhou, H., Cheung, W., and Leung, L. C., "Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm", *Eur J Oper Res*, **194**(3), pp. 637–649 (2009).

2. Tamssaouet, K., Dauzère-Pérès, S., and Yugma, C., "Metaheuristics for the job-shop scheduling problem with machine availability constraints", *Comput Ind Eng*, **125**, pp. 1–8 (2018).

3. Yazdani, M., Aleti, A., Khalili, S. M., et al., "Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem", *Comput Ind Eng*, **107**, pp. 12–24 (2017).

4. Novas, J. M., "Production scheduling and lot streaming at flexible job-shops environments using constraint programming", *Comput Ind Eng*, **136**, pp. 252–264 (2019).

5. Hajibabaei, M. and Behnamian, J., "Reducing noise pollution by flexible job-shop scheduling with worker flexibility: Multi-subpopulation evolutionary algorithm", *Scientia Iranica*, **0**(0), pp. 0–0 (2022).

6. Li, J., Huang, Y., and Niu, X., "A branch population genetic algorithm for dual-resource constrained job shop scheduling problem", *Comput Ind Eng*, **102**, pp. 113–131 (2016).

7. Zamani, R., "An effective mirror-based genetic algorithm for scheduling multi-mode resource constrained projects", *Comput Ind Eng*, **127**, pp. 914–924 (2019).

8. Mohammadipour, F. and Sadjadi, S. J., "Project cost–quality–risk tradeoff analysis in a time-constrained problem", *Comput Ind Eng*, **95**, pp. 111–121 (2016).

9. Guldemond, T. A., Hurink, J. L., Paulus, J. J., et al., "Time-constrained project scheduling", *Journal of Scheduling*, **11**(2), pp. 137–148 (2008).

10. Sharifi, M. and Taghipour, S., "Optimizing a redundancy allocation problem with open-circuit and short-circuit failure modes at the component and subsystem levels", *Engineering Optimization* (2020).

11. Bazargan-Lari, M. R., Taghipour, S., Zaretalab, A., et al., "Planning and scheduling of a parallel-machine production system subject to disruptions and physical distancing", *IMA Journal of Management Mathematics* (2022).

12. Bazargan-Lari, M. R., Taghipour, S., Zaretalab, A., et al., "Production scheduling optimization for a parallel machine subject to physical distancing due to COVID-19", *Operations Management Research* (2022).

13. Sharifi, M. and Taghipour, S., "Redundancy allocation problem of a Multi-State system with Binary-State continuous performance level components", *Expert Syst Appl*, **200**, p. 117161 (2022).

14. Zaretalab, A., Sharifi, M., Guilani, P. P., et al., "A multi-objective model for optimizing the redundancy allocation, component supplier selection, and reliable activities for multi-state systems", *Reliab Eng Syst Saf*, **222**, p. 108394 (2022).

15. Kadri, R. L. and Boctor, F. F., "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case", *Eur J Oper Res*, **265**(2), pp. 454–462 (2018).

16. Bjarnason, E. T. S. and Taghipour, S., "Optimizing simultaneously inspection interval and inventory levels (s, S) for a k-out-of-n system", *Proc Annu Rel. Maintainability Symp*, pp. 1-6, Jan. (2014).

17. Beşikci, U., Bilge, Ü., and Ulusoy, G., "Multi-mode resource constrained multi-project scheduling and resource portfolio problem", *Eur J Oper Res*, **240**(1), pp. 22–31 (2015).

18. El-Abbasy, M. S., Elazouni, A., and Zayed, T., "Generic Scheduling Optimization Model for Multiple Construction Projects", *Journal of Computing in Civil Engineering*, **31**(4) (2017).

19. Ghaleb, M., Taghipour, S., and Zolfagharinia, H., "Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance", *J Manuf Syst*, **61**, pp. 423–449 (2021).

20. Sharifi, M. and Taghipour, S., "Optimal production and maintenance scheduling for a degrading multi-failure modes single-machine production environment", *Appl Soft Comput*, **106** (2021).

21. Pritsker, A. A. B., Waiters, L. J., and Wolfe, P. M., "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", *Manage Sci*, **16**(1), pp. 93–108 (1969).

22. Hurink, J. L., Kok, A. L., Paulus, J. J., et al., "Time-constrained project scheduling with adjacent resources", *Comput Oper Res*, **38**(1), pp. 310–319 (2011).

23. Taghipour, S., Azimpoor, S., "Joint optimization of jobs sequence and inspection policy for a single system with two-stage failure process", IEEE Trans Reliab, 67, pp. 156-169 (2018).

24. Elloumi, S., Fortemps, P., and Loukil, T., "Multi-objective algorithms to multi-mode resource-constrained projects under mode change disruption", *Comput Ind Eng*, **106**, pp. 161–173 (2017).

25. Yassine, A. A., Mostafa, O., and Browning, T. R., "Scheduling multiple, resource-constrained, iterative, product development projects with genetic algorithms", *Comput Ind Eng,* **107**, pp. 39–56 (2017).

26. Kim, K. W., Gen, M., and Yamazaki, G., "Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling", *Appl Soft Comput*, **2**(3), pp. 174–188 (2003).

27. Mori, M. and Tseng, C. C., "A genetic algorithm for multi-mode resource constrained project scheduling problem", *Eur J Oper Res*, **100**(1), pp. 134–141 (1997).

28. Koné, O., Artigues, C., Lopez, P., et al., "Event-based MILP models for resource-constrained project scheduling problems", *Comput Oper Res*, **38**(1), pp. 3–13 (2011).

29. Borreguero, T., Mas, F., Menéndez, J. L., et al., "Enhanced Assembly Line Balancing and Scheduling Methodology for the Aeronautical Industry", *Procedia Eng*, **132**, pp. 990–997 (2015).

30. Russell, A. and Taghipour, S., "Multi-objective optimization of complex scheduling problems in low-volume low-variety production systems", *Int J Prod Econ*, **208**, pp. 1–16 (2019).

31. Russell, A. and Taghipour, S., "Multi-parallel work centers scheduling optimization with shared or dedicated resources in low-volume low-variety production systems", *Appl Math Model*, **80**, pp. 472–505 (2020).

32. Holland, J., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology*, The MIT Press, Cambridge, Massachusetts (1975).

33. Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Dorling Kindersley Pvt Ltd (1989).

34. Pellerin, R., Perrier, N., and Berthaut, F., "A survey of hybrid metaheuristics for the resource-constrained project scheduling problem", *Eur J Oper Res*, **280**(2), pp. 395–416 (2020).

35. Sharifi, M., Saadvandi, M., and Shahriari, M., "Presenting a Series-Parallel Redundancy Allocation Problem with Multi-State Components Using Recursive Algorithm and Meta-Heuristic", *Scientia Iranica*, **0**(0), pp. 0–0 (2018).

36. Liu, J., Liu, Y., Shi, Y., et al., "Solving Resource-Constrained Project Scheduling Problem via Genetic Algorithm", *Journal of Computing in Civil Engineering*, **34**(2) (2020).

37. Lin, J., Zhu, L., and Gao, K., "A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem", *Expert Syst Appl*, **140**, p. 112915 (2020).

38. Zhu, L., Lin, J., Li, Y.-Y., et al., "A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem", *Knowl Based Syst*, **225**, p. 107099 (2021).

39. Fathallahi, F. and Najafi, A. A., "A Hybrid Genetic Algorithm to Maximize Net Present Value of Project Cash Flows in Resource Constrained Project Scheduling Problem with fuzzy parameters", *Scientia Iranica*, **23**(4), pp. 1893–1903 (2016).

40. Birjandi, A., Mousavi, S. M., and Vahdani, B., "Designing a resource-constrained project scheduling model considering multiple routes for flexible project activities: meta-heuristic algorithms", *Scientia Iranica*, **0**(0), pp. 0–0 (2019).

Toronto Metropolitan University. As a Certified Six Sigma Black Belt, a Project Management Professional, and a member of the Professional Engineers of Ontario, he has demonstrated a pragmatic yet systematic approach to applying core industrial engineering principles and methodologies in the manufacturing sector. He has published in the Journal of Production Economics, Journal of Applied Mathematical Modelling, and the Journal of Computer and Industrial Engineering.

Dr. Sharareh Taghipour is Associate Professor and Canada Research Chair in Physical Asset Management at the Department of Mechanical and Industrial Engineering at Toronto Metropolitan University (TMU). She obtained her PhD in Industrial Engineering from the University of Toronto and received her BSc in

Mathematics and Computer Science and her MASc in Industrial Engineering, both from Sharif University of Technology, Iran. The focus of her research has been on stochastic modeling and optimization with applications in physical asset management, such as reliability modeling, maintenance optimization, and production scheduling.

Mani Sharifi is a Visiting Assistant Professor at Miami University. He was a post-doctoral research fellow at the Mechanical and Industrial Engineering Department at Toronto Metropolitan University from 2018 to 2022. He holds a BSc degree from Qazvin Islamic Azad University, an MSc degree from the South Tehran branch of Islamic Azad University, and a PhD from Tehran Research and Science Islamic Azad University in Industrial Engineering. He was the Managerial Editor of the Journal of Optimization in Industrial Engineering. His area of interest includes Reliability Engineering, Combinatorial Optimization, Statistical Optimization, as well as Production Scheduling.

List of Tables:

List of Figures:

Fig. 12: Type 3 Resource Utilization for both scenarios.

Fig. 13: Total Resource Utilization for both scenarios.

Fig. 14: Zone 1 Capacity for both scenarios.

Fig. 15: Zone 2 Capacity for both scenarios.

Fig. 16: Zone 3 Capacity for both scenarios.

Table 1.
Ranges of the GA's parameters and their tuned optimal values.

	Lower level	Upper level	Optimal value
$npop$	50	200	70
p_c	0.7	0.9	0.7
p_m	0.1	0.3	0.3

Table 2.
The output of GA for the best solution of both scenarios.

Parameter	**Value for Scenario 1**	**Value for Scenario 2**
Number of completed jobs	0	48
Number of incompleted jobs (traveling jobs, $\delta_{\alpha}^{+} + \delta_{\beta}^{+}$)	9	39
Reduction in resources $\left(\sum_{l} \delta_{wl}^{-}\right)$	0	NA
Resource positive deviation $\left(\sum_{l} \delta_{wl}^{+}\right)$	2	0
Jobs total completion time (hours)	60.3	51.1

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

Fig. 2: Activity attributes, assumptions, and constraints.

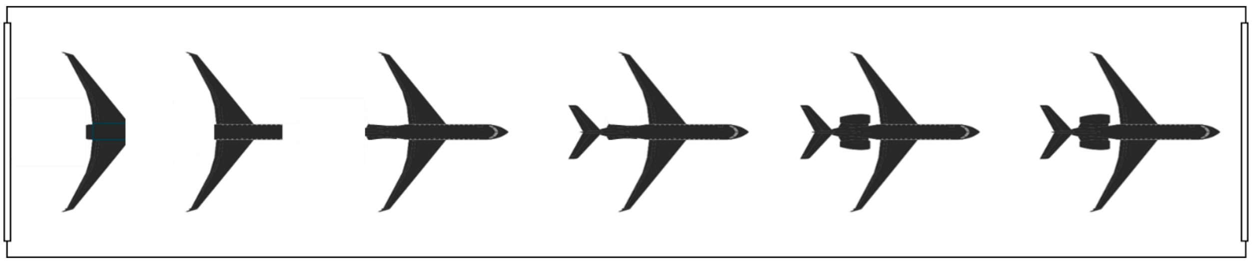


Fig. 2: Sample production layout.

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

Fig. 3: Chromosome Representation.

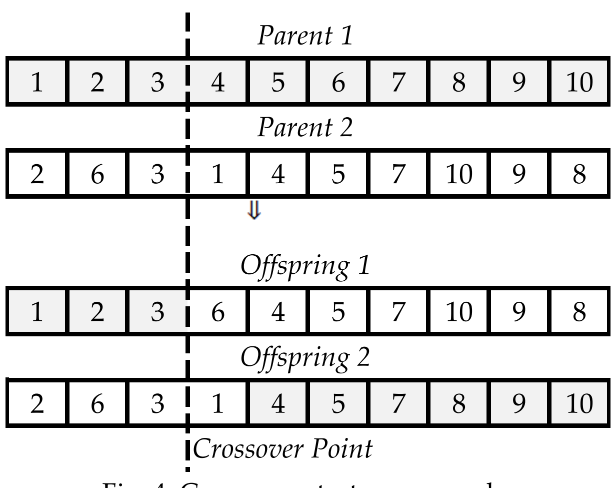


Fig. 4: Crossover strategy example.

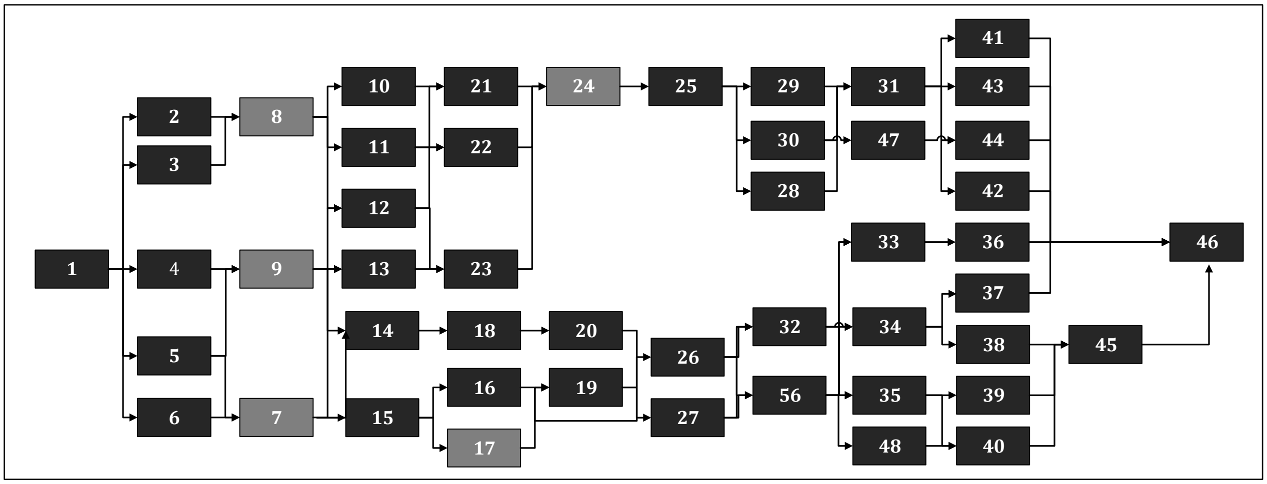
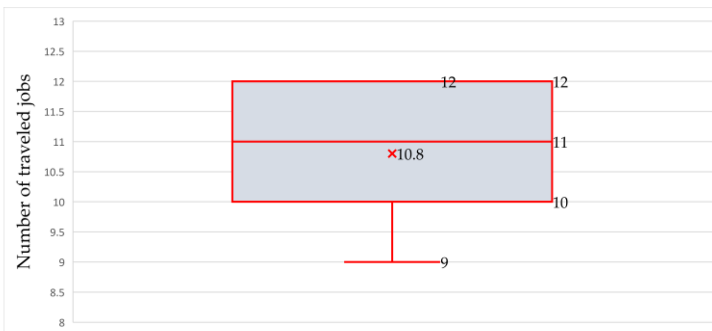
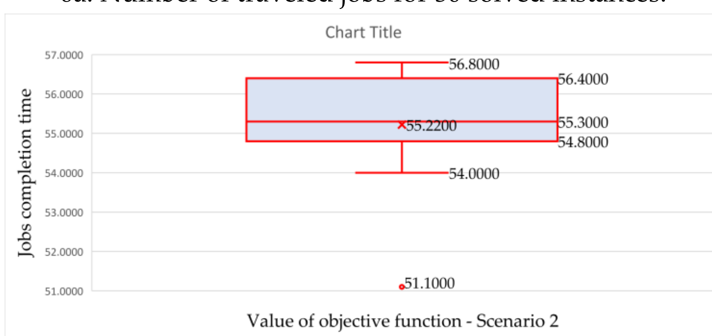


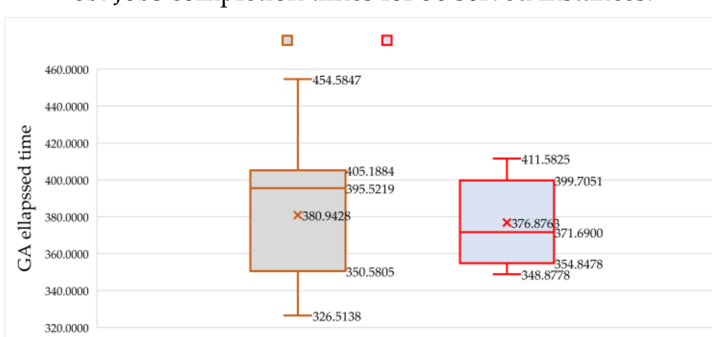
Fig. 5: Activity-on-node network diagram.



6a: Number of traveled jobs for 30 solved instances.



6b: Jobs completion times for 30 solved instances.



6c GA elapsed time for 30 solve instances – scenario 1 and scenario 2.

Fig 6: Output of 30 iterations of GA.

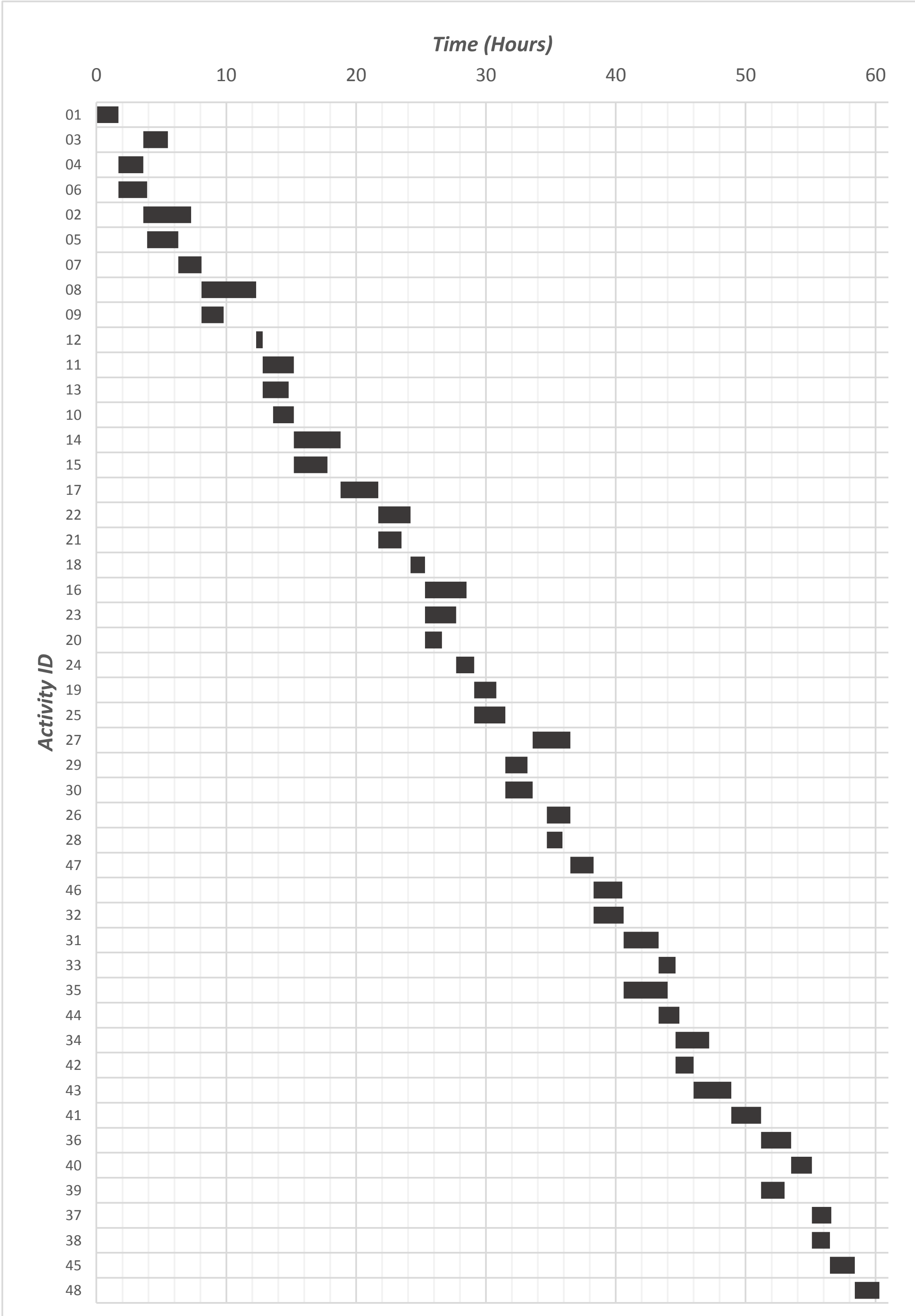


Fig. 7: Gantt Chart (Travelled work permitted).

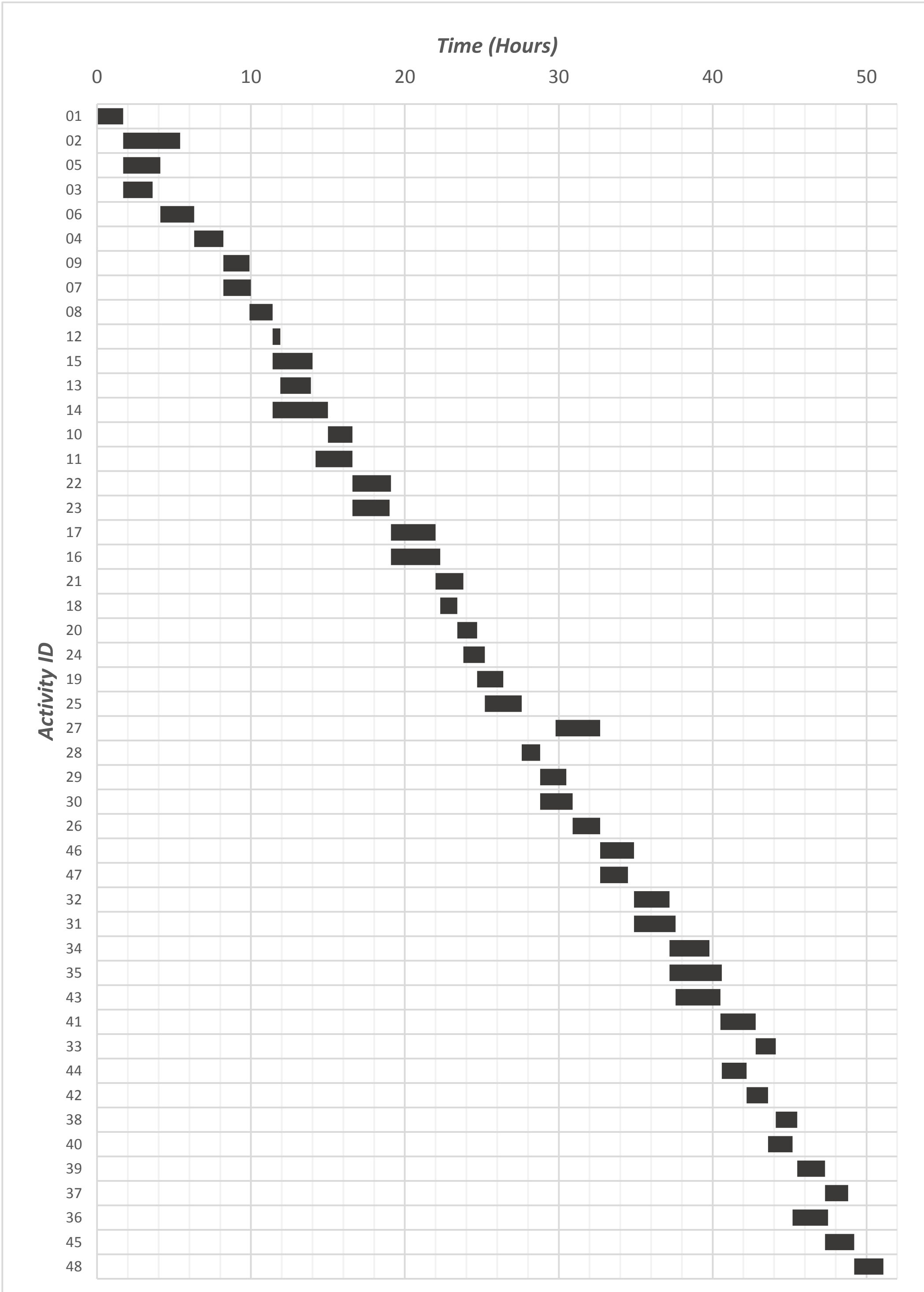


Fig. 8: Gantt Chart (Travel Work Prohibited).

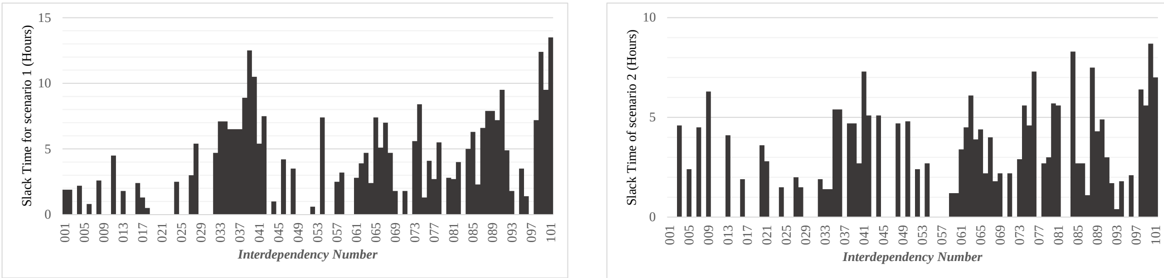


Fig. 9: Slack Time for both scenarios.

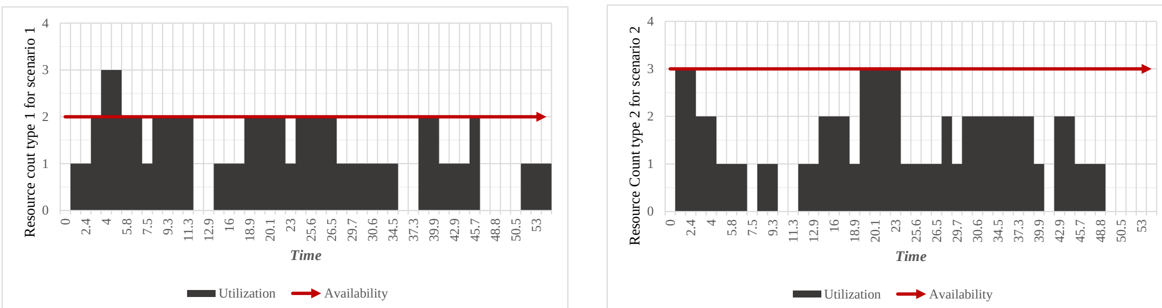


Fig. 10: Type 1 Resource Utilization for both scenarios.

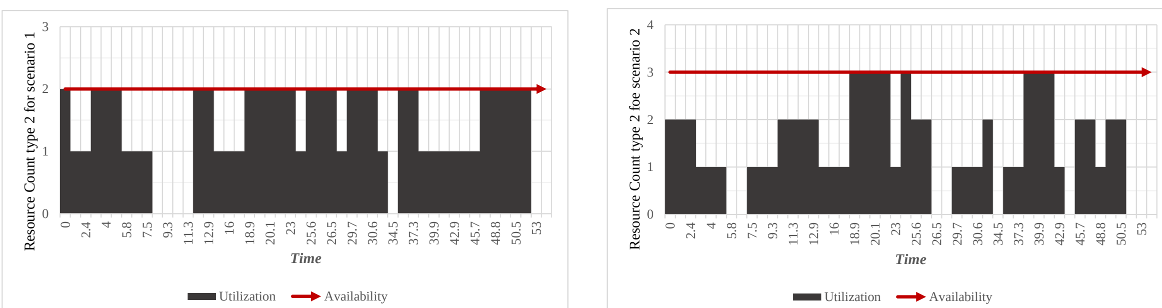


Fig. 11: Type 2 Resource Utilization for both scenarios.

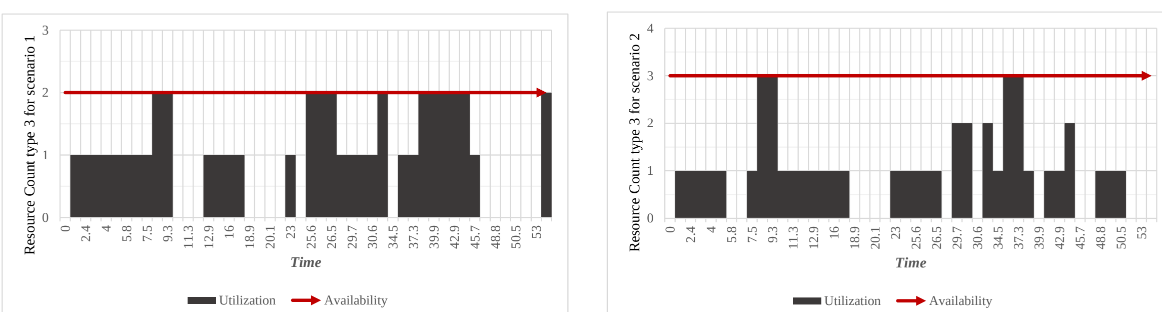


Fig. 12: Type 3 Resource Utilization for both scenarios.

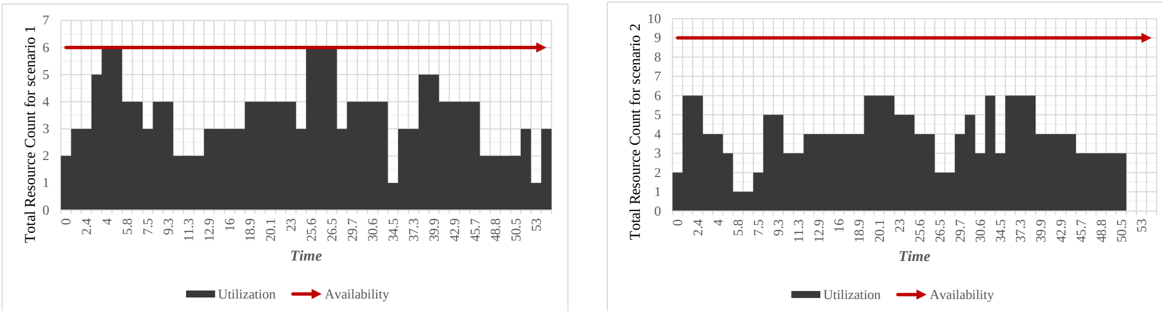


Fig. 13: Total Resource Utilization for both scenarios.

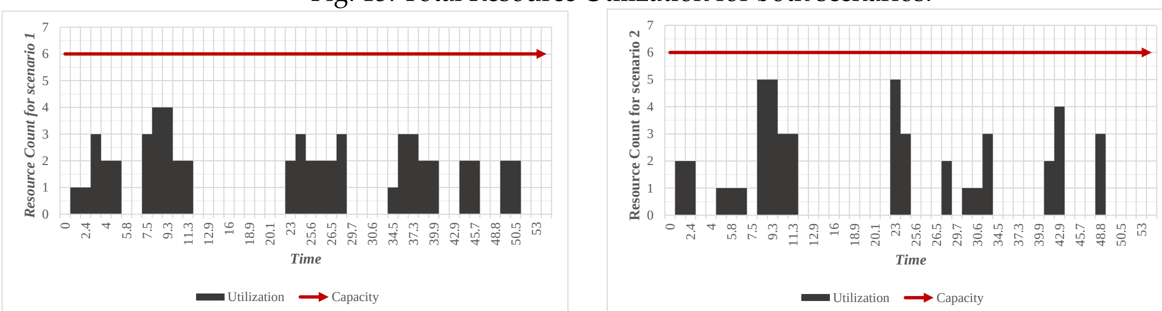


Fig. 14: Zone 1 Capacity for both scenarios.

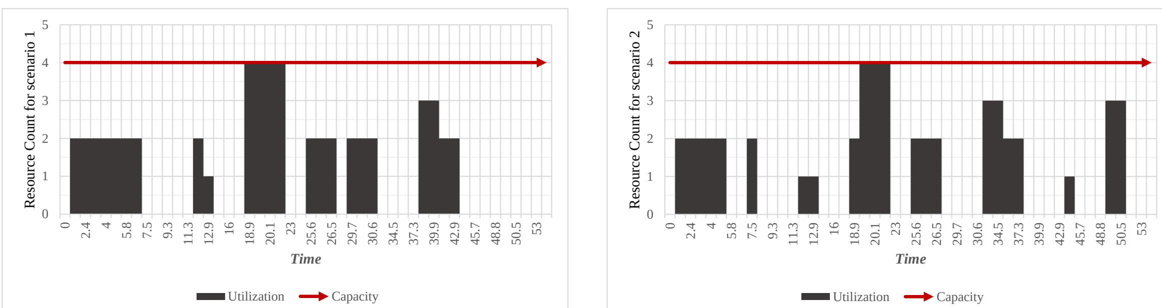


Fig. 15: Zone 2 Capacity for both scenarios.

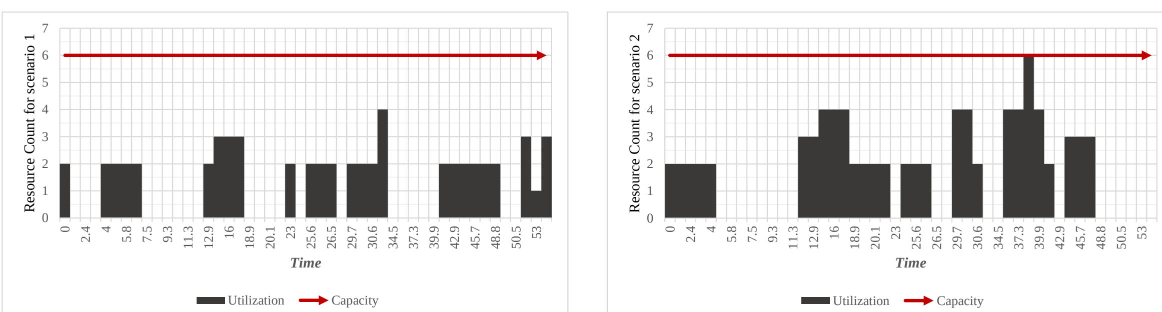


Fig. 16: Zone 3 Capacity for both scenarios.