
Doctoral Dissertations

Student Theses and Dissertations

Summer 2020

Balancing labor requirements in a manufacturing environment

Patrick Bernard Dwyer

Follow this and additional works at: https://scholarsmine.mst.edu/doctoral_dissertations



Part of the [Operations Research, Systems Engineering and Industrial Engineering Commons](#)

Department: Engineering Management and Systems Engineering

Recommended Citation

Dwyer, Patrick Bernard, "Balancing labor requirements in a manufacturing environment" (2020). *Doctoral Dissertations*. 3035.

https://scholarsmine.mst.edu/doctoral_dissertations/3035

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

BALANCING LABOR REQUIREMENTS IN A MANUFACTURING
ENVIRONMENT

by

PATRICK DWYER

A DISSERTATION

Presented to the Graduate Faculty of the
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

ENGINEERING MANAGEMENT

2020

Approved by:

Ruwen Qin, Advisor
Suzanna Long
Cihan H Dagli
Zeyi Sun
Vincent Yu
Dincer Konur
Helene Simon

© 2020

Patrick Dwyer

All Rights Reserved

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following three articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, found on pages 9–24, has been published in the proceedings of the 2018 Annual IISE Conference in Orlando, FL, in May 2018.

Paper II, found on pages 25–38, has been accepted in the proceedings of the 2020 Annual IISE Conference in New Orleans, LA, in May 2020.

Paper III, found on pages 39–95, has been submitted to European Journal of Operational Research.

ABSTRACT

This research examines construction environments within manufacturing facilities, specifically semiconductor manufacturing facilities, and develops a new optimization method that is scalable for large construction projects with multiple execution modes and resource constraints. The model is developed to represent real-world conditions in which project activities do not have a fixed, prespecified duration but rather a total amount of work that is directly impacted by the level of resources assigned. To expand on the concept of resource driven project durations, this research aims to mimic manufacturing construction environments by allowing a non-continuous resource allocation to project tasks. This concept allows for resources to shift between projects in order to achieve the optimal result for the project manager. Our model generates a novel multi-objective resource constrained project scheduling problem. Specifically, two objectives are studied; the minimization of the total direct labor cost and the minimization of the resource leveling. This research will utilize multiple techniques to achieve resource leveling and discuss the advantage each one provides to the project team, as well as a comparison of the Pareto Fronts between the given resource leveling and cost minimization objective functions. Finally, a heuristic is developed utilizing partial linear relaxation to scale the optimization model for large scale projects. The computation results from multiple randomly generated case studies show that the new heuristic method is capable of generating high quality solutions at significantly less computational time.

ACKNOWLEDGMENTS

This work would not have been possible without the financial support of Intel Corporation. Intel's construction organization's willingness to promote continued education and to be an advocate for my research will not be forgotten. A special thanks goes out to Dr. Dincer Konur, my research advisor. He has been with me through the entire process and has taught me not only how to properly conduct research, but how to look at problems in new and creative ways that will serve me throughout my career.

I am grateful to all of the professors and classmates that I have had the pleasure of working with as I pursued the completion of my projects. Conducting research in a different time zone could cause potential issues, but everyone involved was always willing to accommodate me along the journey. I am grateful for the members of my dissertation committee as they have provided me with guidance and taught me a great deal about scientific research that will help me as I continue my career. I would especially like to thank Dr. Ruwen Qin, the chair of my committee. She stepped up to provide guidance as I completed my research and helped push me to achieve my goals.

I would like to express my deepest thanks to my family in the pursuit of this project. I wish to thank my loving and supportive wife, Julia, and my beautiful daughter, Emilia, who provide unending inspiration.

TABLE OF CONTENTS

	Page
PUBLICATION DISSERTATION OPTION	iii
ABSTRACT.....	iv
ACKNOWLEDGMENTS	v
LIST OF ILLUSTRATIONS.....	ix
LIST OF TABLES	x
NOMENCLATURE	xi
 SECTION	
1. INTRODUCTION.....	1
1.1. BACKGROUND	1
1.1.1. Manufacturing Construction.....	1
1.1.2. Construction Scheduling Platforms.....	2
1.2. PROBLEM STATEMENT.....	3
1.3. CHALLENGES AND TECHICAL NEEDS	5
1.4. EXPECTED CONTRIBUTIONS.....	6
 PAPER	
I. SCHEDULE OPTIMIZATION FOR CONSTRUCTION IN MANUFACTURING ENVIRONMENTS	9
ABSTRACT	9
1. INTRODUCTION AND LITERATURE REVIEW	10
2. PROBLEM FORMULATION.....	12
3. REFORMULATION AND LABOR LEVELING.....	18

3.1. LINEAR REFORMULATION.....	18
3.2. LABOR LEVELING	20
4. FURTHER DISCUSSION AND CONCLUSIONS	22
REFERENCES	23
II. SCHEDULE OPTIMIZATION FOR CONSTRUCTION: COMPARISON OF RESOURCE LEVELING TECHNIQUES	25
ABSTRACT	25
1. INTRODUCTION.....	26
2. RESEARCH METHODOLOGY	29
2.1. RESOURCE LEVELING OBJECTIVE FUNCTIONS	30
3. ADAPTIVE ε -CONSTRAINT.....	31
4. CASE STUDY	33
4.1. IMPLEMENTATION OF THE MODEL.....	33
5. FURTHER DISCUSSION AND CONCLUSIONS	36
REFERENCES	37
III. INTEGRATED PROJECT SCHEDULING AND WORKER ASSIGNMENT WITH COST AND LABOR CONSIDERATIONS FOR A SEMICONDUCTOR MANUFACTURING CONSTRUCTION PROBLEM.....	39
ABSTRACT	39
1. INTRODUCTION AND LITERATURE REVIEW	40
2. PROBLEM SETTINGS AND FORMULATION	46
3. SOLUTION METHOD	56
3.1. CLASSICAL ε -CONSTRAINT METHOD	58
3.2. PARTIAL LINEAR RELAXATION BASED APPROXIMATING METHOD	61

4. NUMERICAL STUDIES.....	69
4.1. PROBLEM INSTANCES AND COMPUTATIONAL SETTING.....	70
4.2. COMPARISON OF THE SOLUTION METHODS	75
5. CONCLUDING REMARKS	87
REFERENCES.....	89
SECTION	
2. CONCLUSIONS AND RECOMMENDATIONS.....	96
2.1. CONCLUSIONS	96
2.2. RECOMMENDATIONS.....	97
BIBLIOGRAPHY.....	99
VITA.....	101

LIST OF ILLUSTRATIONS

SECTION	Page
Figure 1.1 Example of crossover and mutation operations.....	6
 PAPER II	
Figure 1. Procedure flowchart.....	32
Figure 2. Sample schedule network diagram and schedule inputs.	33
Figure 3. Resource histogram before resource leveling.....	34
Figure 4. Pareto Charts for three different optimization methods.	35
 PAPER III	
Figure 1. Illustration of the average extreme and density points for Instance 10 of I = 20.....	77
Figure 2. Illustration of the average extreme and density points for Instance 10 of I = 40	77
Figure 3. Illustration of average extreme and density points.....	81
Figure 4. Illustration of average extreme.....	86

LIST OF TABLES

PAPER I	Page
Table 1. Notation.	15
PAPER II	
Table 1. Objective functions for resource leveling.	30
PAPER III	
Table 1. Notation.	57
Table 2. Chain lengths and averages of T and A for problem sets with varying I	72
Table 3. Quantitative comparison of Algorithms 1 and 2 for CSP.....	76
Table 4. 2-mode results.....	79
Table 5. 3-mode results.....	80
Table 6. Comparison of PF and \widehat{PF} based results on hypervolume and unary measures.....	85

NOMENCLATURE

Symbol	Description
$i, j \in P$	Set of projects (installation or demolition of a tool)
$k \in K$	Set of modes
$r, t \in T$	Set of weeks
z_{ikt}	1 if project i is performed on mode k during week t , 0 otherwise
x_{it}	# of workers working on project i during week t
h_k	# of hours a worker works on mode k
s_{it}	1 if project i 's activity j starts at the beginning of week t , and 0 otherwise
f_{it}	1 if project i 's activity j finishes at the end of week t , and 0 otherwise
h_k	# of hours a worker works on mode k
H	# of hours required by project i
W	# of workers that can work during a week
c_k	the hourly wage of a worker working on mode k
lw_i	Minimum number of workers allowed to work during a week on project i
uw_i	Maximum number of workers allowed to work during a week on project i
sd_i	the earliest start time (beginning of week) of project i
dd_i	the due date (end of a week) of project i
a_{ij}	The precedence relationship between projects i and j

1. INTRODUCTION

1.1. BACKGROUND

Construction is a labor driven industry. As such, the ability to accurately forecast and manage the trade workforce is critical in any construction project. From the planning to the implementation phase, the project management team must be aware of market conditions, as the amount of total available resources directly correlates with number of construction activities that can occur at one time. Poorly managed labor can result either in a schedule push when certain tasks cannot be staffed or in the project going over budget as resources are paid for and not utilized efficiently. While construction scheduling is usually generalized into one category, there exists multiple subsets, each of which possess unique constraints that can greatly alter the model and the subsequent optimal schedules and resource allocation.

1.1.1. Manufacturing Construction. Construction in a manufacturing or operations facility differs vastly from construction in the singular project model. Manufacturing facilities involve multiple individual tools or pieces of equipment where construction activities can be divided into three categories: installation, demolition or conversion. While each piece of equipment may be part of a larger production line, the schedule for each piece of equipment is typically independent from the other tools. A good example of manufacturing construction can be found in the semiconductor industry. Semiconductor facilities operate 24 hours a day, 7 days a week. When a new technology node is introduced thousands of tools have to be install, demolished, or converted, all the while not impacting the remainder of the

facility. Construction focused on a singular model is comprised of individual tasks interratted to achieve one result. A new building is a good axample of this type of construction. A project is kicked off with groundbreaking and excavation activities, followed by pouring a foundation, setting steal, installing mechanical and electrical equipment, and finally architectural finishes. While each task is important, the project is not complete until all systems are complete and ready to be turned over.

1.1.2. Construction Scheduling Platforms. Commercial scheduling platforms are a valuable tool for managing construction projects. Software platforms are widely used as they are able to provide project planning as it relates to scheduling, resource allocation, and cost management. These platforms are vital in that they allow the project management team to track the status of the project in real time. A key feature in scheduling function allows project managers to link multiple tasks in finish-to-start, finish-to-finish, or start-to-start precedence relationships with any desired lag associated with the operations. Each task can have resources and a cost allocated to them which allows the project management team insight to any potential risk resulting from resource constraints or cost overrun. While these platforms provide valuable information once a schedule is generated, they lack the ability to generate optimum schedules based on a series of inputs. As noted by Mellentien and Trautmann [1], there exist a considerable performace gap between the scheduling platforms and state-of-the-art scheduling algorithms.

1.2. PROBLEM STATEMENT

Through a thorough literature analysis, we have discovered that multiple gaps exist in the current research regarding multi-objective resource constrained scheduling problems (MORCSP) specific to manufacturing construction. As manufacturing facilities have hundreds of independent tools or machines, the model must be able to treat each project independently. Project float is defined as the amount of time a project can be delayed before it impacts the deadline the project. Resource leveling can be achieved by creating a critical path and shifting schedules along the project float. The presence of project float indicates that there is an initial task and a finish task that all the projects tie into, but this is not the case in manufacturing. Another aspect that is unique to manufacturing construction is flexible resource allocation. In industry, a construction manager or superintendent can shift shared workforce across multiple projects, adjusting the number of workers allocated on a daily or weekly basis. Current models do not allow for this type of flexibility. A project is assigned a number of resources and that number is static until the task is complete. Recently, research has been conducted on an approach that allows for resources to vary throughout the lifespan of a project. This method is called flexible resource profile (FRP). While this field of study is promising, it does have its limitations. FRP models do allow the duration of a project to be independent of the resource profile. However, the profiles are still pre-determined which limits flexibility. Also, a key constraint in FRP is that once a project starts it must be continuously worked. Our research will challenge this assumption and account for resource splitting, which allows a project to start and stop multiple times before completion.

The objectives that are optimized in academic research do not represent the ultimate goals of manufacturing construction. The four main aspects studied in literature include resource level, cost, makespan, and quality, as these constitute the pillars of any construction project. However, there has been little research thus far that focuses on the interaction between resource leveling and cost. Much of the existing research on multi-objective resource constrained scheduling problems has focused on the total project makespan [2-4]. As previously discussed, in manufacturing and operations with multiple independent projects, the total project makespan is not a vital success criterion as each project has an equally important completion date.

Our research will explore schedule generation schemes where the main objective is to reduce resource leveling while providing the lowest total labor cost. In response to this problem, our study proposes to generate a multi-objective model that is based on academic research but is useful within the construction industry. We will focus our study on two of the largest risks to project success: total cost and resource availability. From the model that we will develop, the success criteria will be tested against existing industry construction schedules.

While it is straightforward to minimize cost and makespan, this is less so the case with resource leveling. Previous research has varied in how resource leveling is calculated, from measuring the absolute difference in resources between periods [5] to calculating the difference between the actual and desired headcount [6-8]. While each method can provide valuable information, the difficulty lies in reconciling the results of the different methods. Our research will utilize a model developed for manufacturing

construction and perform the multi-objective optimization using multiple resource leveling techniques.

1.3. CHALLENGES AND TECHICAL NEEDS

The main challenge of this research is to develop a model that can be easily scaled for use in real world scenarios. Multi-objective resource constrained problems generally are NP hard problems (non-deterministic polynomial-time hardness) [9], which already complicate scaling due to the size and complexity of the problem. Our model aims to provide greater flexibility to construction scheduling solutions, as it is our goal to simulate the choices that management teams face every day. There have been numerous studies on various heuristic methods for larger multi-objective resource constrained problems. Two common heuristic approaches are genetic algorithms [4,10-11] and the manipulation of activities float in the schedule [12-13]. Unfortunately, neither approach will suffice due to the conditions established in our problem statement. As each project is independent, there is no project float. Also, a key aspect of our study is that the number of resources drives the length of a project's duration. For example, a given project requires 10 resources to be completed. The work can be defined as 2 resources a day for 5 days, 5 resources a day for 2 days or any combination to achieve the desired resource usage. While this offers increased scheduling flexibility, it also creates a scenario where the same activity on two schedules may have different duration. Because of this, we will not be able to utilize traditional genetic algorithms as we will not be able to ensure feasible schedules during a crossover or mutation operation, see Figure 1.1.

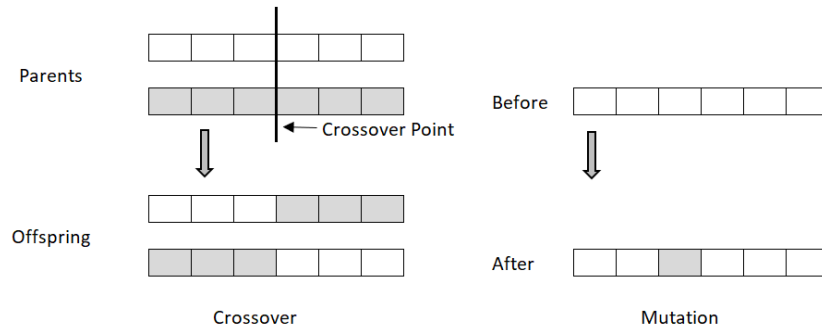


Figure 1.1. Example of crossover and mutation operations.

1.4. EXPECTED CONTRIBUTIONS

The expected contribution of our research is the creation of a novel model that is for schedule generation in manufacturing construction projects. While there are multiple studies and industrial software packages that address the need for schedule generation solutions, we believe that there is a gap in current methods in that they do not allow project management teams to take full advantage of all options available during the construction phase. From an industry standpoint, the largest contribution will come during both the project planning and execution phases of the project. Unlike software packages, the novel model will be able to provide the project management team with choices regarding the level of risk in resource availability and total cost early on in the planning phase. This will allow projects to be accurately budgeted at their onset. Furthermore, as the model will have the ability to generate a new resource profile for each period per project, the project's construction manager or superintendent will have insight on how to schedule and micro-manage the short-term schedule to optimize the workforce.

In academia, we are expected to expand on the latest research regarding FRP projects that base the duration of a project on the number of allocated resources. While it is our assumption that the novel model will produce non-dominated outcomes as it relates to similar situations, the ability to scale our system to handle large problem sets is what will make it unique. We plan to develop a novel heuristic approach to allow for near optimum schedule generation in scenarios in which a project's duration may not be identical among the various schedules.

While this research contributes a new method for approaching multi-objective construction schedule problems, there are situations within the construction industry that would not benefit from the novel model and approach. The first scenario involves scheduling work with an in-house labor workforce. There exist multiple commercial software platforms that are built to coordinate and schedule work for a set number of employees. These platforms act more as a central repository of information and are useful for establishing a standard collection of inputs that are used to create schedules or make adjustments in real time [14]. Typically, these platforms track current market cost, productivity rates and updated worker availability. These platforms are not required to create buy off charts between resource leveling and cost minimization due to the fact that the total labor force is a constant number and the goal is to properly allocate that constant labor force.

The second scenario that would not benefit from our model is the traditional construction project. Traditional construction projects can consist of thousands of interrelated tasks to achieve one project schedule. A project has a defined start and finish that each task contributes to. Our model would not be able to take advantage of key

features of these schedules, such as project and total float. While proven to be less efficient than models used in research, current software such as Primavera are built to handle large single project construction activities [1]. These tools are used to plan, schedule, and control large-scale individual projects will provide the ability to visualize project performance versus planned schedule and budget.

PAPER

I. SCHEDULE OPTIMIZATION FOR CONSTRUCTION IN MANUFACTURING ENVIRONMENTS

ABSTRACT

Construction in a semi-conductor manufacturing facility involves the constant construction, demolition and modification of thousands of machines that enable raw silicon to be transformed to a wafer with over 1.4 billion transistors. While maintenance is always required to operate such large facilities, a majority of the construction follows a cyclical pattern of a two-year cycle. The ability to handle the construction loads of over 3,000 machines in a 6-12 month period requires a specialized construction workforce that is able to meet the strict quality requirements of working in a class 1 cleanroom environment. Within a given construction ramp, the trade headcount can rise from hundreds to low thousands during the peak periods. Baseline schedules are usually created around the technology demands without added cost or inefficiencies to the construction contractors. These inefficiencies range from slipped schedules to added cost due to overtime or rework requirements. The model we propose treats each construction activity as an individual project and aims to minimize the total cost of labor during the technology ramp while also minimizing the amount of labor resources that are hired and fired. Labor resources represent the largest risk to the program from a cost, schedule and quality focus.

1. INTRODUCTION AND LITERATURE REVIEW

Manufacturing of semiconductor microchips consists of hundreds of individual machines that change silicon into highly developed transistors. Advances in semiconductor technology have traditionally followed a new technology cadence of every two years and the number of transistors in an integrated circuit will double as well as the transistor costs will [1]. This pattern was predicted by Gorgon Moore in 1965 and has mostly held steady for the last 50 years. The manufacturing process consists of hundreds of machines that work in an assembly line process. Each machine is constructed specifically for the current technological node and requires a unique electrical and chemical infrastructure. The demolition and installation construction of the semiconductor manufacturing machines drive the cadence to maintain Moore's law and mass produce product every two years. Due to the cyclical cycle of Moore's law, there are periods of peak construction and valleys, in which a large amount of labor resources must be hired and fired over a short period of time. Manufacturing enabled schedules are traditionally created to hit key technology milestone without considering the effect of construction or labor resources available. This paper models the system as a resource-constrained project scheduling problem with labor requirements and discusses labor leveling approaches.

Schedule development is a major aspect of managing a construction project. The critical chain method (CCM) is a common technique that is based on the estimated time durations of activities on the critical path as discovered by calculating the early and late start and finish dates of the activities [2]. CCM is an accurate technique for repetitive

and predictable tasks or activities with predictable durations; however, construction projects are unique in nature [3], which causes issues in estimating time and resource durations when developing an accurate schedule. Construction projects are traditionally late and over budget due to the challenges related to the unique or custom conditions involved. There are many surveys that identify the major causes for delay in construction projects (see, e.g., [4]).

Classical resource-constrained project scheduling problems involve a set of activities with a fixed duration where each activity needs a certain amount of resources to accomplish the task in the given time duration. The total resources available is also constrained [5]. The objective for these systems can range, given the project's overall goal, from minimizing cost, makespan to resource fluctuations. Construction projects generally deal with multi-objective resource constrained project scheduling problems (MORCPSP) as a project manager must know the impact of schedule on costs or quality. Brucker et al. [6] review the notation and characteristics associated with MORCPSP problems. A survey of scheduling constrained projects that deal with the classification of multiple methods is reviewed by Blazewicz et al. [7]. In this study, we concentrate on multi-mode resource constrained problems.

Construction projects in a semiconductor manufacturing company consist of thousands of independent projects with shared labor resources, with shifting manufacturing need dates throughout the life of the program. A major concentration within the semiconductor construction industry is therefore to minimize the resource fluctuations by moving non-critical activities in the project schedule. This is important considering the fact that hiring and firing large amounts of labor resources from one time

period to the next is undesired. To avoid the risks of not having the proper labor to accomplish all activities in a given period, many projects have to pay for standing or non-value added labor time. Techniques to minimize resource fluctuations can be broken down into sum of squares method [8], minimum moment arm method [9], absolute difference between resource consumption in consecutive time periods [10-11], and no predefined levelling pattern [9]. These models optimize the release and re-hire across multiple time periods. This study extends the El-Rayes [12] that aims to minimize resource fluctuations with no predefined pattern by investigating the effects of resource leveling across multiple projects.

The rest of the paper is organized as follows. Next section presents the basic optimization model for the multi-mode scheduling problem with labor requirements. The resulting model is a non-linear integer programming problem. In Section 3, we provide a linear formulation and formulate different approaches to include labor resource leveling in the scheduling model. The last section briefly discusses a solution framework and future steps of the study.

2. PROBLEM FORMULATION

The problem of interest in this study is scheduling of construction of special tools/machines that transform raw silicon to integrated semiconductor circuits. We refer to construction of a tool/machine as a project. Let these projects be indexed by $i \in I = \{1, 2, \dots, n\}$. The construction involved on a tool consists of either the demolition or installation of certain activities/utilities performed by trade partners. Let the trade

partners (activities) be indexed by $j \in J = \{1, 2, \dots, m\}$. While the formulation presented for an arbitrary number of trade partners, we note that, typically, there are three main trades such that $j = 1$ defines the process trade (activity), $j = 2$ defines the mechanical trade (activity), and $j = 3$ defines electrical trade (activity).

Construction of each tool is an independent project with trade resources (labor) shared amongst multiple projects. Each project requires a given amount of construction hours from each trade for completion. Let H_{ij} denote the time required to complete project i 's activity j (by trade j). The following assumptions define the working conditions:

- The maximum number of workers each trade partner can provide is fixed throughout the whole schedule. Let W_j be the maximum number of workers from trade j that can work for construction during any period (week).
- Each worker from any trade can work on different modes during a week. Let the working modes be indexed by $k \in K = \{1, 2, \dots, l\}$ and let h_k denote the number of hours a worker in mode k works per week. Again, even the formulation considers an arbitrary number of models, there are three different modes considered for a worker during a week such that $h_1 = 40 \text{ hours/week}$, $h_2 = 50 \text{ hours/week}$, and $h_3 = 60 \text{ hours/week}$.
- There is not a linear relation between the number of hours worked and the hourly rate of a worker. Furthermore, each trade has different rates. Therefore, we define c_{jk} as the hourly wage of a trade j worker working on mode k (one would guess that as h_k increases, c_{jk} increases as well due to overtime).
- A worker will not switch projects and will not change modes during a week.

- Workers performing activity j on a project during a week will have the same mode throughout the week as trade workers form a crew and follow the same work plan throughout the week.
- A project is completed when all of its activities are completed.

In addition to the above assumptions, the following restrictions apply. Each project cannot start prior to a specific date (this is because, for an installation project, parts are being delivered or, for a demolition project, the current work of the tool should be completed) and each project should be completed before a due date. Let sd_i and dd_i denote the earliest start time (beginning of week) and due date (end of a week) of project i , respectively. Based on the working conditions, we focus on scheduling project activities on a weekly basis, i.e., each period is one week and let the weeks be indexed by $t \in T = \{1, 2, \dots, \tau\}$. Note that one can define $\tau = \max_{i \in I} \{dd_i\}$. Ultimately, the decisions to be made are how many workers and in what mode they will work on each project's each activity during each week. Let $z_{ijkt} = 1$ if project i 's activity j is performed on mode k during week t , 0 otherwise. Note that $\sum_{k \in K} z_{ijkt} \leq 1 \quad \forall i \in I, \forall j \in J, \forall t \in T$ as at most one mode can be selected (when no mode is selected, there is no-one working on that project's that activity during that week). Next, let x_{ijt} number of workers (from trade j) working on project i 's activity j during week t . One can notice that if $z_{ijkt} = 0$, then $x_{ijt} = 0$. Due to the start and finish time restrictions and availability of the workers, it might be possible that there is no activity going on a project during an intermediate week after the project starts. For instance, during weeks 2 and 3, a project's activity 1 can be worked on by trade 1 workers, and then, there is no work on the project during week 4, and trade 1 or another trade continues its work on the project in week 5. To define

project, start and finish times, we define additional variables as follows. Let $s_{ijt} = 1$ if project i 's activity j starts at the beginning of week t , and 0 otherwise, $f_{ijt} = 1$ if project i 's activity j finishes at the end of week t , and 0 otherwise.

Remark that $\sum_{t \in T} s_{ijt} = 1 \forall i \in I, \forall j \in J$ and $\sum_{t \in T} f_{ijt} = 1 \forall i \in I, \forall j \in J$ since each project's each activity will start and finish during a week. Furthermore, one can note that project i 's start time will be $S_i = \min_{j \in J} \{\sum_{t \in T} t s_{ijt}\}$ and project i 's finish time will be $F_i = \max_{j \in J} \{\sum_{t \in T} t f_{ijt}\}$. Table 1 summarizes the notation used. Additional notation will be defined as needed.

Table 1. Notation.

Type:	Notation:	Explanation:
Indices	$i \in I = \{1, 2, \dots, n\}$	Set of projects (installation or demolition of a tool)
	$j \in J = \{1, 2, \dots, m\}$	Set of trade partners/activities
	$k \in K = \{1, 2, \dots, l\}$	Set of modes
	$t \in T = \{1, 2, \dots, \tau\}$	Set of weeks
Variables	$z_{ijkt} \in \{0, 1\}$	1 if project i 's activity j is performed on mode k during week t , 0 otherwise
	$x_{ijt} \in \{0, 1, 2, \dots\}$	# of workers (from trade j) working on project i 's activity j during week t
	$s_{ijt} \in \{0, 1\}$	1 if project i 's activity j starts at the beginning of week t , and 0 otherwise
	$f_{ijt} \in \{0, 1\}$	1 if project i 's activity j finishes at the end of week t , and 0 otherwise
Parameters	h_k	# of hours a worker works on mode k
	H_{ij}	# of hours required by project i 's activity j
	W_j	# of workers from trade j that can work during a week
	c_{jk}	the hourly wage of a trade j worker working on mode k
	sd_i	the earliest start time (beginning of week) of project i
	dd_i	the due date (end of a week) of project i

Next, we present the mathematical formulation for the scheduling problem of interest. The objective is to minimize the total labor cost of the construction schedule. One can note that the total cost amounts to $\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} z_{ijkt} x_{ijt}$. Then, the mathematical formulation reads as follows:

$$(\mathbf{P}) \quad \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} z_{ijkt} x_{ijt}$$

$$\text{Subject} \quad \sum_{k \in K} z_{ijkt} \leq 1 \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (1)$$

to

$$\sum_{t \in T} s_{ijt} = 1 \quad \forall i \in I, \forall j \in J \quad (2)$$

$$\sum_{t \in T} f_{ijt} = 1 \quad \forall i \in I, \forall j \in J \quad (3)$$

$$\min_{j \in J} \left\{ \sum_{t \in T} t s_{ijt} \right\} \geq sd_i \quad \forall i \in I \quad (4)$$

$$\max_{j \in J} \left\{ \sum_{t \in T} t f_{ijt} \right\} \leq dd_i \quad \forall i \in I \quad (5)$$

$$\sum_{k \in K} \sum_{t \in T} h_k z_{ijkt} x_{ijt} \geq H_{ij} \quad \forall i \in I, \forall j \in J \quad (6)$$

$$\sum_{i \in I} x_{ijt} \leq W_j \quad \forall j \in J, \forall t \in T \quad (7)$$

$$x_{ijt} \leq M \sum_{k \in K} z_{ijkt} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (8)$$

$$\sum_{k \in K} z_{ijkt} \leq \sum_{r=1}^t s_{ijr} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (9)$$

$$\sum_{k \in K} z_{ijkt} \leq \sum_{r=t}^{\tau} f_{ijr} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (10)$$

$$x_{ijt} \in \{0, 1, 2, \dots\}, \quad s_{ijt} \in \{0, 1\}, \quad f_{ijt} \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (11)$$

$$z_{ijkt} \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall k \in K, \forall t \in T \quad (12)$$

$\in T$

The objective function minimizes the total labor cost. Constraints (1) assure that at most one mode is selected for each project's each activity during any week. Constraints (2) and (3) define that a project's activity starts (at the beginning) and finish (at the end) at a single week, respectively. Constraints (4) and (5) enforce that a project cannot start before the earliest start time and it should be completed before the due date, respectively. Constraints (6) ensure that the number of hours performed on a project's activity is at least how much needed to complete that activity for that project. Constraints (7) restrict the total number of workers from each trade working during any week to be less than or equal to the available number (maximum) of workers from that trade, while constraints (8) guarantee that there will be no worker from a trade on a project during a week if there is no mode selection for the workers to perform the corresponding activity on that project during that week. Constraints (9) and (10) assure that there is no mode selection for a trade (hence, no workers performing the corresponding activity considering constraints (8)) on a project's activity before the project's activity starts and after the project's activity ends, respectively. Finally, constraints (11) and (12) are integer and binary definitions of the variables.

One can note that (P) is a non-linear integer programming model. In particular, the non-linearity follows from the objective function and constraints (4), (5), and (6). In the next section, we present a linear reformulation. Furthermore, (P) is a single-objective model with sole cost minimization objective. However, as noted in the introduction, resource leveling is an important factor that should be considered in designing work schedules for the trades. It is possible that the number of workers needed from one trade can change significantly from one week to the next over the production cycle. This

situation is not desired as it creates issues such as the lack of the ability to retain key talent or extra payments made to retain labor that is not being utilized. Market labor resources are not always able to keep up with the manufacturing demand, which can result undesired pushes in the project schedule. Therefore, while reformulating (P), we also discuss additional objectives to overcome the fluctuations in labor and present multi-objective schedule optimization model in the next section.

3. REFORMULATION AND LABOR LEVELING

3.1. LINEAR REFORMULATION

Recall that the non-linearity of (P) is due to the objective function and constraints (4), (5), and (6). Specifically, the objective function and constraints (6) are non-linear as they include multiplications of the variables z_{ijkt} and x_{ijt} . To overcome these, we introduce a new variable to replace x_{ijt} . Particularly, let x_{ijkt} = number of workers from trade j working on project i on mode k during week t . With the introduction of x_{ijkt} , the objective function can be rewritten as $\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} x_{ijkt}$, which is linear, and constraints (6) can be rewritten as $\sum_{k \in K} \sum_{t \in T} h_k x_{ijkt} \geq H_{ij} \forall i \in I, \forall j \in J$, which are also linear. Note that we will still have z_{ijkt} variables and we will need to modify constraints (7) and (8) of (P). Introduction of x_{ijkt} increases the number of variables. To linearize constraints (4) and (5), we can simply replace them with $\sum_{t \in T} t s_{ijt} \geq s d_i \forall i \in I, \forall j \in K$ and $\sum_{t \in T} t f_{ijt} \leq d d_i \forall i \in I, \forall j \in J$. These reformulations increase the number of constraints.

With the above changes, the non-linearity in (P) is eliminated in an expense of increased number of variables and constraints. Furthermore, we note that constraints (1) can be eliminated from (P) due to constraints (2) and (9). The reformulation of (P) with these changes, denoted by (P'), is presented below.

$$(\mathbf{P}') \quad \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} x_{ijkt}$$

$$\text{Subject} \quad (2), (3), (9), (10)$$

to

$$\sum_{t \in T} t s_{ijt} \geq s d_i \quad \forall i \in I, \forall j \in J \quad (13)$$

$$\sum_{t \in T} t f_{ijt} \leq d d_i \quad \forall i \in I, \forall j \in J \quad (14)$$

$$\sum_{k \in K} \sum_{t \in T} h_k x_{ijkt} \geq H_{ij} \quad \forall i \in I, \forall j \in J \quad (15)$$

$$\sum_{i \in I} \sum_{k \in K} x_{ijkt} \leq W_j \quad \forall j \in J, \forall t \in T \quad (16)$$

$$x_{ijkt} \leq M z_{ijkt} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (17)$$

$$s_{ijt} \in \{0,1\}, f_{ijt} \in \{0,1\} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (18)$$

$$z_{ijkt} \in \{0,1\}, x_{ijkt} \in \{0,1,2, \dots\} \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (19)$$

$$\in K \quad \forall t$$

$$\in T$$

The objective function and the constraints are defined similar to (P).

3.2. LABOR LEVELING

In this section, we discuss approaches for resource leveling and show how to modify (P') with these approaches. First approach is minimization of the sum of the absolute values of the difference of the number of workers in consecutive time periods from each trade (see also [12]). Specifically, the difference in the number of workers used from a trade in two consecutive weeks (if negative, represents fires; and, if positive, represents hires) is preferred to be low. Since both firing and hiring additional workers is not preferred throughout the whole schedule, one can minimize the sum of the absolute values of the differences for each trade individually or over all trades. To do so, one can minimize $\sum_{i \in I} \sum_{j \in J} \sum_{t \in T} |\sum_{k \in K} x_{ijkt} - \sum_{k \in K} x_{ijk(t-1)}|$ or minimize $\sum_{i \in I} \sum_{t \in T} |\sum_{k \in K} x_{ijkt} - \sum_{k \in K} x_{ijk(t-1)}| \quad \forall j \in J$. Note that, in either case, the additional objective function(s) would be non-linear due to the absolute value function. Since the objective is to minimize, the model can be made linear by introducing U_{ijt} and enforce constraints such that $U_{ijt} \geq \sum_{k \in K} x_{ijkt} - \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T$ and $U_{ijt} \geq -\sum_{k \in K} x_{ijkt} + \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T$. Then, the multi-objective schedule optimization model with the first approach for leveling would read:

$$(P') \quad \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} x_{ijkt}$$

1)

$$\begin{aligned} & \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{t \in T} U_{ijt} && \text{(or, for each trade} \\ & && \text{separately, Minimize} \\ & && \sum_{i \in I} \sum_{t \in T} U_{ijt} \quad \forall j \in J) \end{aligned}$$

Subject to (2), (3), (9), (10), (13)-(19)

$$U_{ijt} \geq \sum_{k \in K} x_{ijk t} - \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (20)$$

$$U_{ijt} \geq -\sum_{k \in K} x_{ijk t} + \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (21)$$

The first approach, when it minimizes the sum of the differences over all trades, can disfavor a trade, on the other hand, if it minimizes the sum of the differences for each trade separately, the number of objectives will be large. Given that the problem on hand is already complex, this increased in the number of objectives would make the model even more challenging. The next approach discussed can effectively overcome these issues. The second approach is minimization of the maximum difference (see also [13]), rather than sum of the differences over all trades or for each trade separately. The second approach is to minimize $\max_{i \in I, j \in J, t \in T} \{U_{ijt}\}$. Again, this additional objective function is non-linear but the resulting model can be modified easily to be linear by using a single variable, say U , due to the minimization objective. In particular, the multi-objective schedule optimization model with the second approach for leveling would read:

$$(\mathbf{P}'\text{-}2) \quad \text{Minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{t \in T} h_k c_{jk} x_{ijk t}$$

$$\text{Minimize} \quad U$$

$$\text{Subject to} \quad (2), (3), (9), (10), (13)\text{--}(19)$$

$$U \geq \sum_{k \in K} x_{ijk t} - \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (22)$$

$$U \geq -\sum_{k \in K} x_{ijk t} + \sum_{k \in K} x_{ijk(t-1)} \quad \forall i \in I, \forall j \in J, \forall t \in T \quad (23)$$

4. FURTHER DISCUSSION AND CONCLUSIONS

In the above discussions, we presented formulations for the multi-mode schedule optimization with labor considerations for semiconductor manufacturing projects. We note that, without resource leveling considerations, even if (P') is linear, it is still a large-scale integer programming model. Furthermore, once one aims to include labor leveling approaches, the problem becomes a large-scale multi-objective integer programming model, which would be more challenging than its single-objective version. Therefore, due to these complexities of the models, we will focus on developing genetic algorithms, which are successfully used for multi-objective discrete optimization models. The genetic algorithms will focus on generating a set of Pareto efficient solutions, which then can be used to finalize schedules by comparing their costs and labor fluctuations. We plan to develop multi-stage genetic algorithms varying in their chromosome representations and stage definitions as done in [14] and use separations to improve computational performance as done in [15]. The ultimate goal is to compare various resource leveling approaches and decide which approach results in the best labor leveling with the minimum cost increase. To provide a proof of concept, a sample scenario will be developed to represent a real period of time of a construction ramp.

REFERENCES

1. Schaller, Robert R., 1997, "Moore's law: past, present and future." *IEEE spectrum* 34.6, 52-59.
2. Gray, C. F., & Larson, E. W, 2006, *Project management: The managerial process*, McGraw-Hill/Irwin, 2006.
3. Zhou, Jingyang, et al., 2013, "A review of methods and algorithms for optimizing construction scheduling." *Journal of the Operational Research Society* 64.8, 1091-1105.
4. Assaf, Sadi A., and Sadiq Al-Hejji, 2006, "Causes of delay in large construction projects." *International journal of project management* 24.4, 349-357.
5. Ballestín, Francisco, and Rosa Blanco, 2011, "Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems." *Computers & Operations Research* 38.1, 51-62.
6. Brucker P, Drexl A, Mohring R, Neumann K, Pesch E., 1999, "Resource-constrained project scheduling: notation, classification, models and methods." *European Journal of Operational Research*, 112:3–41.
7. Blazewicz, Jacek, Jan Karel Lenstra, and AHG Rinnooy Kan, 1983, "Scheduling subject to resource constraints: classification and complexity." *Discrete Applied Mathematics* 5.1, 11-24.
8. Son, Jaeho, and Mirosław J. Skibniewski, 1999, "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *Journal of Construction Engineering and Management* 125.1, 23-31.
9. Harris, R. I, 1978, *Precedence and arrow networking techniques for construction*, Wiley, New York
10. Easa, Said M, 1989, "Resource leveling in construction by optimization." *Journal of construction engineering and management* 115.2, 302-316.
11. Son, Jaeho, and Kris G. Mattila, 2004, "Binary resource leveling model: Activity splitting allowed." *Journal of construction engineering and management* 130.6, 887-894.
12. El-Rayes, Khaled, and Dho Heon Jun, 2009, "Optimizing resource leveling in construction projects." *Journal of Construction Engineering and Management* 135.11, 1172-1180.

13. Damci, A., Arditi, D. and Polat, G., 2016. "Impacts of different objective functions on resource leveling in Line-of-Balance scheduling." *KSCE Journal of Civil Engineering*, 20(1), 58.
14. Konur, Dincer, and Mihalıs M. Golias, 2017. "Loading Time Flexibility in Cross-docking Systems." *Procedia Computer Science* 114, 491-498.
15. Konur, Dinçer, Hadi Farhangi, and Cihan H. Daglı, 2016. "A multi-objective military system of systems architecting problem with inflexible and flexible systems: formulation and solution methods." *OR spectrum* 38.4, 967-1006.

II. SCHEDULE OPTIMIZATION FOR CONSTRUCTION: COMPARISON OF RESOURCE LEVELING TECHNIQUES

Patrick B Dwyer

Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409

Dr. Dincer Konur

Department of Computer Information Systems and Quantitative Methods, Texas State University, San Marcos, TX 78666

ABSTRACT

One of the largest challenges and risks to a successful project is to be able to allocate the available labor resource in a way to maintain schedule while also achieving the project budget. This study investigates the impacts of using different resource leveling objective functions in multi-objective multi-mode resource constrained project scheduling problems within the construction field. Specifically, the two objectives are studied: the minimization of the total direct labor cost and the minimization of the resource leveling. Three alternative formulations for defining the resource leveling objective function are used to formulate three alternative bi-objective construction scheduling models. These models enable the project durations to be adjusted based on the number of resources (workers) assigned as well as the mode selected for the assigned resources in each period. An exact methodology based on the adaptive ε -constraint method is used to solve the resulting bi-objective integer linear programming models. Using a case study, the different resource leveling objective functions are tested by

comparing the three Pareto Fronts, each corresponding to an alternative resource leveling objective function. The results from three objective functions allow the project management team to evaluate different options with respect to risk tolerance and confidence about future market conditions.

1. INTRODUCTION

The success of a construction project relies on the ability of the management team to manage the available resources [1]. Resource management involves both the generation of a scheduling plan to assign the available resource and the utilization of those resource to not only complete the project on time, but also at the lowest cost possible. Construction projects within manufacturing and operational settings can involve many repetitive activities or projects, often unique in nature, where a shared labor pool is utilized amongst all the projects. In the research reported in this paper, different resource leveling objective functions are investigated for multi-objective resource constrained project scheduling problems. Our research builds off a bi-objective construction scheduling model proposed by Dwyer and Konur [2] in order to identify the different approaches for resource leveling and their impacts when scheduling a large construction program.

Resource management can be broken up into two main categories during a construction project [7]. The first category involves a market with limited resources and not obtaining proper number of resources will result in an extension of the project duration. In such cases, the project manager or scheduler must decide which projects not

located on the critical path to adjust the start or finish dates. This category falls within the resource allocation category where the resources drive the schedules. The typical objective of the problems in this category is to reduce the total project makespan using the available resources [4]. The second category is known as resource leveling or smoothing and involves a fixed duration on project tasks. In this case, resource leveling is used to ensure efficient use of the labor resources. The goal of this category is to minimize the fluctuations of the resources used by shifting activity start dates and resource allocations. Traditionally, the most important challenge to a construction project is to achieve resource leveling within a fixed duration. This study focuses on a combination of the above categories such that we aim to minimize the resource fluctuations within a resource constrained environment.

The main idea of resource leveling is to shift the start times of non-critical activities along their available float [5]. One of the earliest research to solve resource leveling was presented by Burgess and Killebrew [6]. The goal of their research was to create a uniform resource histogram by minimizing the sum of the squares of the resources allocated. Harris [7] expanded on the work of Burgess and Killebrew [6] and utilized the minimum moment method for resource leveling. Minimum moment method states that the moment of the periodic resource demands about the horizontal axis of a project's resource histogram is a good measure of the resource utilization and the optimal resource allocation exists when the total moment is at a minimum, i.e., when the resource histogram is of a rectangular shape [8]. The goal of this method is to minimize resource buildups by considering the advantages of shifting non-critical activities. Because it is assumed that the duration of a project is fixed, the critical activities are not altered. This

method assumes that, once a project has started, it cannot be altered, and resources are uniformly distributed across the duration of the projects.

Further studies in resource leveling allow for the methods to be broken down into four main categories: sum of squares method [6,9], minimum moment arm method [7-8], absolute difference between resource consumption in consecutive time periods [10], and no predefined levelling pattern [11]. Damci and Potal [3] researched the effects of multiple resource leveling objective functions on construction projects. In their study, the durations were assumed to be fixed and cost of the projects were not taken into account. Damci et al. [12] expanded on that research to measure the effect of multiple resource leveling objective functions on line-of-balance scheduling, in which the same activities are repeated in a linear fashion.

Although there exist many studies on resource allocation, very few of those studies allowed for resources to be split between periods. Resource splitting varies from traditional modelling methods in that resources do not have to be uniformly distributed across the duration of the projects. Karaa and Nasr [13] emphasized that one of the major weaknesses of Critical Path Method (CPM) is the fact that activities cannot be intermittent. Hariga and El-Sayegh [14] presented an optimization model for resource leveling, in which activity splitting is allowed by moving non-critical activities within their float. Our study expands on the research from Hariga and El-Sayegh [14] as we also allow construction resources work under different overtime modes. Allowing for resource splitting and operating under multiple working modes, our research allows for more opportunities to reduce the total cost of the project while achieving the desired resource leveling as there are more options in which to allocate each individual trade.

The rest of the paper is organized as follows. Next section presents the research methodology and outlines the resource leveling objective functions that are studied and compared. Section 3 summarizes the implementation of the ε -constraint method for generating the exact Pareto front for the bi-objective models with alternative resource leveling objectives. Section 4 presents a case study, where the Pareto fronts corresponding to alternative resource leveling objectives are compared. Concluding remarks and future research directions are noted in Section 5.

2. RESEARCH METHODOLOGY

A review of literature indicates that there have been several studies focusing on multi-objective resource constrained project scheduling problems (MORCPSP). However, the impact of using different resource leveling objective functions for such problems, where resources are shared amongst multiple projects and splitting is allowed, has not been investigated. The main objective of this research is to investigate the impacts of different resource leveling techniques and discuss the advantages and shortfalls of each technique. To do so, we first review the literature on resource leveling to identify the objective functions used for resource leveling in resource constrained project scheduling problems. After that, we use a construction scheduling optimization model from Dwyer and Konur [2] and create alternative bi-objective optimization models which differ in their resource leveling objective functions. For solving these models, we use the adaptive ε -constraint method algorithm, which is described in Section 3. Finally, a case study is presented to compare different resource leveling approaches

2.1. RESOURCE LEVELING OBJECTIVE FUNCTIONS

Our study focuses on the multi-objective scheduling problems. The two objectives that we chose to study are cost minimization and workforce resource leveling. Construction cost can be broken up into two main categories: labor and materials. While material cost can be minimized through value engineering in the design phases, construction labor can be minimized by allocating resources efficiently through scheduling techniques. The objective function that we used as our first objective function (1) is located in Table 1. Literature research indicates that there are multiple objective functions that can be used to level a resource histogram (Table 1). While it is difficult to maintain a uniform usage rate for labor resources, the multiple objective functions goal is to make the usage rate as uniform as possible or to make any non-uniform rate fit the owner's requirements

Table 1. Objective functions for resource leveling.

Objective Function Number	Optimization Criteria	References
1	Minimize the direct construction labor cost	2
2	Minimization of the sum of the absolute deviations in resource usage for a determined time interval (day, week, etc.)	10,17
3	Minimization of the maximum resource usage for a determined time interval (day, week, etc.)	17
4	Minimization of the maximum deviation in resource usage for a determined time interval (day, week, etc.)	17

3. ADAPTIVE ε -CONSTRAINT

The ε constrained method is one of the most common exact methods, it is solved by optimizing one of the objective functions using the remaining objective functions as constraints [15]. By varying the right-hand side of the constrained objective functions, the efficient solutions of the problem are obtained. Laumanns et al. [15] proposed adaptive ε -constraint method, which is modification of the ε -constraint method for bi-objective integer programming models. In the adaptive ε -constraint method, similar to the classic ε -constraint method, one of the objective functions is moved to the constraints with an upper bound. In the traditional ε -constraint method, this upper bound on the objective function moved to the constraints is decreased by ε at each iteration until the model with the upper bound constraint becomes infeasible. In our research, we utilize the adaptive ε -constraint method because the bi-objective scheduling model of [2] is an integer programming model. Specifically, in the adaptive ε -constraint method, the upper bound value is defined by subtracting ε from the objective function value of the final solution selected in the last iteration. This way, it is guaranteed that a different solution is returned at each iteration. The solution at the end of each iteration is a Pareto efficient solution.

The adaptive ε -constraint method is coded in Matlab and the flowchart in Figure 1 summarizes the overall procedure. The first step of the procedure is to establish model parameters. These inputs include schedule precedence relationships and early start and late finish dates for each project. The parameters also include all the resource constraints, from individual construction tasks to overall projects/programs. The final inputs for the

model include the different overtime working modes and the hourly costs associated with each mode as well as the step values for the ϵ -constraint. Our research set the step value to 1 because the resource leveling can change by at most 1 unit due to integrality of the number of workers.

Step 3 of the procedure conducts the optimization operations for the cost and resource leveling objectives. The first step is to minimize the total cost objective function using the initial upper bound value of the resource leveling constraint. The goal of the adaptive ϵ -constraint method is to create a Pareto Front or a non-dominated front in which none of the objective functions can be improved in value without degrading some of the other objective values [16]. The schedule generated from the cost minimization objective function cannot guarantee that the resource level is not dominated. To alleviate the possibility of a dominated solution, the next step calculates the minimum resource level objective function using the total cost calculated in the previous step as a constraint. The additional calculation guarantees that the solution lands on the Pareto Front.

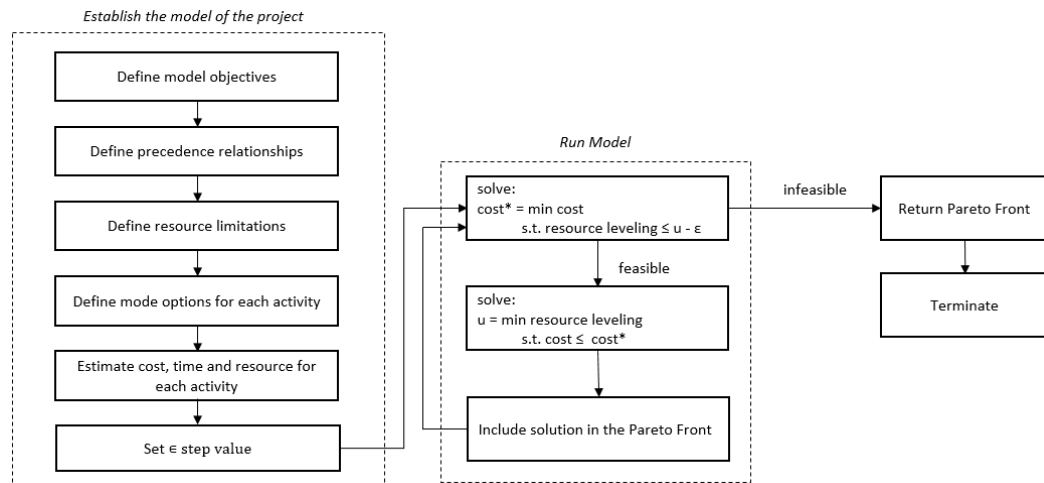


Figure 1. Procedure flowchart.

4. CASE STUDY

The impacts of using different objective function in MORCPSP involving resource leveling can best be demonstrated utilizing an example project. A project with eight activities over a 19-period duration is depicted in Figure 2. The network diagram in Figure 2 also shows the precedence relationships, the activities required hours to complete, early start and late finish times, and the minimum and maximum allowable workers on each project. We assumed that the work can be scheduled utilizing three different modes or overtime rates: 40 per week at \$75/hour, 50 hours per week at \$82/hour and 60 hours per week at \$90/hour.

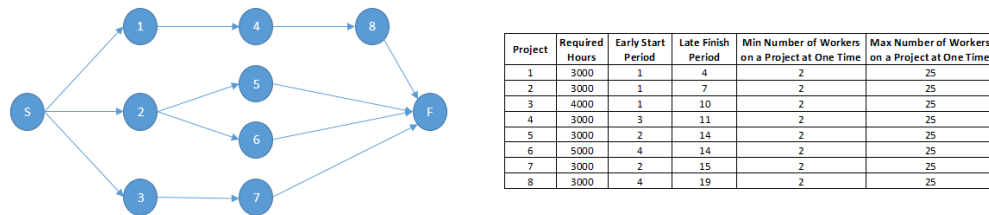


Figure 2. Sample schedule network diagram and schedule inputs.

4.1. IMPLEMENTATION OF THE MODEL

Prior to running the model of the case study, the resource histogram in Figure 3 was plotted before leveling to reflect the conditions in the initial schedule prior to any optimization operations. The schedule was created utilizing the critical path method in Microsoft Projects. The total number of resources required to complete a project was calculated by dividing the minimum duration, in hours, by 40. The value of 40 represents one trade working in a 40 hour week. The total number of resources per project was then

uniformly distributed across the projects total duration. This method does not account for resource constraints.

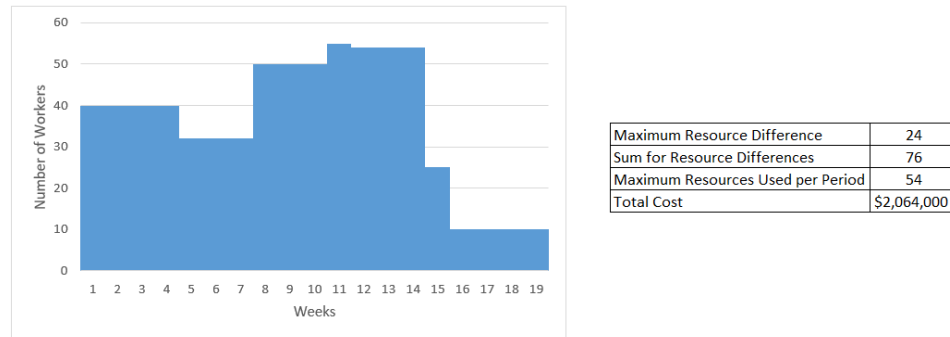


Figure 3. Resource histogram before resource leveling.

The adaptive ε -constraint method discussed above was then implanted on the same model shown in Figure 3 utilizing the three different objective functions (2-4) for resource leveling. Figure 4 shows the results of the bi-objective optimization models using the three different resource leveling object functions. The top row of Figure 4 is a Pareto Front between the total cost of the project and the resource leveling objective function. However, a simple comparison of the three Pareto Fronts does not provide the scheduling team much insight into the advantages of each method. In order to compare the three methods, we normalized all the schedules generated using the three objective functions against a singular objective function. The second row of Figure 4 depicts the maximum resource difference between periods of the schedules that were generated by the objective functions (2-4) in the top row. The results highlight three different risk scenarios for the project manager and scheduler to consider when developing the project schedule. While objective function (4) provides the model with the least total cost

options per max labor difference between periods, it does not take into account how many periods reach that maximum value. Objective function (2) allows the team to hedge that risk by knowing the total amount of resource deviation between projects. In comparing the results of the two Pareto Fronts, a summation of the projects maximum difference between tasks plateaus at 25 laborers while the sum of all the deviations between periods reduces as 25 laborers at as total cost of \$2,130,000. This indicates only one period in the projects presents a deviation from a steady state resource allocation plan. That security can be attractive to risk adverse management teams. While objective function (3) presents the least desirable results in terms of both total cost and max difference between periods it does provide valuable information for the project team if there is uncertainty in total labor availability. Our sample schedule created an initial constraint of total resource availability of 55 labor trades per period. Objective function (3) provides the management team insight into potential cost impacts if that total labor availability changes from initial assumption and how that will impact total project cost.

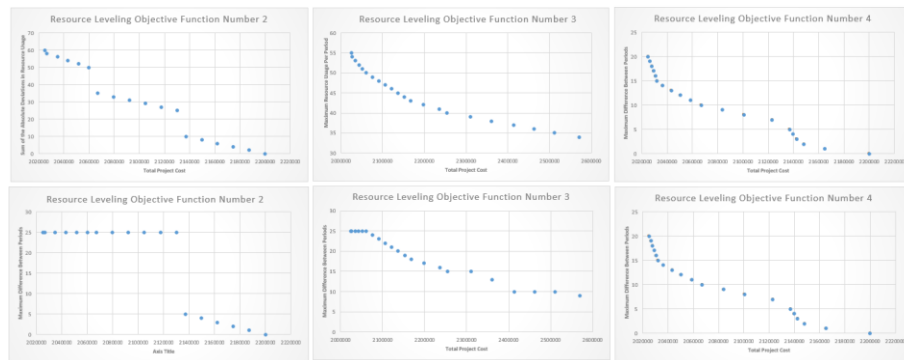


Figure 4. Pareto Charts for three different optimization methods.

5. FURTHER DISCUSSION AND CONCLUSIONS

The impacts of using different objective functions in leveling resources in MORCPSP were investigated in this study. Three different objective functions were identified after a review of prior studies focused on resource leveling utilizing linear methods. The objective functions were used in an adaptive ϵ -constraints method with the total direct labor cost for the project to create Pareto Fronts for each objective function. A simple test case of a project involving eight activities was utilized to compare the three different objective functions. Comparing the Pareto Fronts between three different resource leveling models help highlight the advantages and disadvantages of each methodology. The practical implementation of the study show that while the concept of resource leveling is simple to understand the means and methods to accomplish the task can great affect the final schedule. The goal of the study is not to provide the program manager or scheduler with the best scheduling techniques but to identify the different approaches when scheduling a large constructing program. Our study focused on comparing the three different resource leveling on a small case study utilizing an exact solving methodology. A direction for future research can be to study the effect of the three objective functions on larger models, solving with heuristic or evolutionally algorithms. Our research focused on linear resource leveling objective functions, future work can also expand that to non-linear techniques.

REFERENCES

1. Benjaoran, Vacharapoom, Wisitsak Tabyang, and Nart Sooksil. "Precedence relationship options for the resource levelling problem using a genetic algorithm." *Construction management and economics* 33.9 (2015): 711-723.
2. Dwyer, Patrick, and Konur, Dincer. "Schedule Optimization for Construction in Manufacturing Environments." *IISE Annual Conference*, 28 May 2018, Orland, FL, (2018): 1153-1158.
3. Damci, Atilla, and Gul Polat. "Impacts of different objective functions on resource leveling in construction projects: a case study." *Journal of civil engineering and management* 20.4 (2014): 537-547.
4. Senouci, Ahmed B., and Hojjat Adeli. "Resource scheduling using neural dynamics model of Adeli and Park." *Journal of construction engineering and management* 127.1 (2001): 28-34.
5. Lu, Ming, and Heng Li. "Resource-activity critical-path method for construction planning." *Journal of construction engineering and management* 129.4 (2003): 412-420.
6. Burgess, A. R., and James B. Killebrew. "Variation in activity level on a cyclical arrow diagram." *Journal of Industrial Engineering* 13.2 (1962): 76-83.
7. Harris, Robert B. "Packing method for resource leveling (PACK)." *Journal of Construction Engineering and Management* 116.2 (1990): 331-350.
8. Hiyassat, Mohammed A. Salem. "Applying modified minimum moment method to multiple resource leveling." *Journal of Construction Engineering and Management* 127.3 (2001): 192-198.
9. Son, Jaeho, and Mirosław J. Skibniewski. "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *Journal of construction engineering and management* 125.1 (1999): 23-31.
10. Son, Jaeho, and Kris G. Mattila. "Binary resource leveling model: Activity splitting allowed." *Journal of construction engineering and management* 130.6 (2004): 887-894.
11. El-Rayes, Khaled, and Dho Heon Jun. "Optimizing resource leveling in construction projects." *Journal of Construction Engineering and Management* 135.11 (2009): 1172-1180.

12. Karaa, Fadi A., and Anas Y. Nasr. "Resource management in construction." *Journal of construction engineering and management* 112.3 (1986): 346-357.
13. Hariga, Moncer, and Sameh M. El-Sayegh. "Cost optimization model for the multiresource leveling problem with allowed activity splitting." *Journal of construction engineering and management* 137.1 (2010): 56-64.
14. Laumanns, Marco, Lothar Thiele, and Eckart Zitzler. "An efficient, adaptive parameter variation scheme for metaheuristics based on the ϵ -constraint method." *European Journal of Operational Research* 169.3 (2006): 932-942.
15. Chen, Yu, Xiufen Zou, and Weicheng Xie. "Convergence of multi-objective evolutionary algorithms to a uniformly distributed representation of the Pareto front." *Information Sciences* 181.16 (2011): 3336-3355.
16. Wagner, Harvey M., Richard J. Giglio, and R. George Glaser. "Preventive maintenance scheduling by mathematical programming." *Management Science* 10.2 (1964): 316-334.
17. Mavrotas, George, and Kostas Florios. "An improved version of the augmented ϵ -constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems." *Applied Mathematics and Computation* 219.18 (2013): 9652-9669.

III. INTEGRATED PROJECT SCHEDULING AND WORKER ASSIGNMENT WITH COST AND LABOR CONSIDERATIONS FOR A SEMICONDUCTOR MANUFACTURING CONSTRUCTION PROBLEM

Dinçer Konur (corresponding author)

Assistant Professor, Department of Computer Information Systems and Quantitative
Methods, McCoy College of Business Administration, Texas State University

Patrick B Dwyer

Department of Engineering Management and Systems Engineering, Missouri University
of Science and Technology, Rolla, MO 65409

ABSTRACT

Preparing the manufacturing facility to produce new products is a very important process in competitive semiconductor industry. This preparation requires construction, demolition, and modification of high-tech machines/tools in a working environment. In this study, we present a project scheduling problem integrated with worker assignments for the problem of preparing a semi-conductor manufacturing facility. The project scheduling problem studied is a bi-objective model with flexible resource profiles where preemption is allowed. For the model, we first discuss the implementation of the well-known ε -constraint method for generating the exact Pareto front of the model. After that, we propose an approximation approach based on partial linear relaxation. Based on a set of numerical analyses, it is demonstrated that the approximation approach is computationally efficient, and it can find solutions within the proximity of the Pareto efficient solutions.

Keywords: Scheduling, Worker assignment, Bi-objective, Semiconductor

1. INTRODUCTION AND LITERATURE REVIEW

Semiconductor industry has been steadily growing since late 1980s and semiconductor industry sales are expected to reach over \$430 billion by the end of 2020. Short life cycles of the products, rapidly changing designs due to technological advancements, and increasing demand pressure semiconductor manufacturers to compete in several dimensions such as price, quality, innovation, and lead time. Effective management and strategic, tactical, and operational decision making are therefore crucial in every stage of the supply chains in semiconductor industry. For detailed review of different studies related to semiconductor supply chains, we refer the reader to a recent series of three reviews: Mönch et al. (2018a) review strategic supply chain decisions (part I of the review series), Uzsoy et al. (2018) review demand and capacity planning and inventory management, and Mönch et al. (2018b) review master and production planning and demand fulfillment (part III of the review series). Also, Mönch et al. (2012), Fowler and Mönch (2017), and Mönch et al. (2018c) are other comprehensive resources about research studies in semiconductor manufacturing and supply chains. Especially, production planning and job scheduling in semiconductor manufacturing have been intensively studied in the literature (see, e.g., other reviews by Uzsoy et al. (1992; 1994), Gupta and Sivakumar (2006), Mönch et al. (2011)).

Semiconductor industry has been following the candace of Moore's law, which suggested that the number of transistors on an integrated circuit would double regularly (Schaller, 1997). As a result, as noted by Mönch et al. (2018a) as well, one crucial competitive advantage for the semiconductor manufacturers is the ability to effectively

introduce new products into the market. This equates that the manufacturing facilities and production lines need to be periodically updated for the next production ramp-up of the new product. This process is referred to as equipment installation and qualification (Cheng et al. (2015)) and constitutes the major part of the lead time of a semiconductor supply chain (see, e.g., Cheng et al. (2012)). It is therefore important for manufacturers to efficiently complete this process for gaining competitive advantage.

Particularly, a semiconductor facility ramp-up corresponds to a facility construction planning problem, which includes various activities/tasks such as installation of new tools/machines and demolition or modification of some of the existing tools/machines in an active manufacturing environment. This problem corresponds to a variant of project scheduling problem. In this study, we analyze a multi-mode flexible resource profile project scheduling problem with two objectives: minimization of total labor cost and minimization of the maximum (peak) labor use. Accordingly, in what follows, we first review related project scheduling literature. After that, we discuss the studies that focus on scheduling of semiconductor facility ramp-up.

Put simply, project scheduling problem aims at scheduling projects (or project activities), which typically have precedence relations and/or resource constraints (Herroelen, 2005). Project scheduling problem is one of the most studied optimization problems. The books by Demeulemeester and Herroelen (2002) and Schwindt and Zimmerman (2015a; 2015b) are among many great resources to review various project scheduling concepts, problems, formulations, and solution approaches. The simple case of single-mode makespan minimization without resource constraints is polynomially solvable (see, e.g., Kolisch and Padman (2001)); therefore, a majority of the academic

research focuses on resource-constrained project scheduling problem (RCPSP), which is an NP-hard problem (see, e.g., Blacewicz et al. (1983)). We refer the reader to Özdamar and Ulusoy (1995), Herroelen et al. (1998), Brucker et al. (1999), and Hartmann and Briskorn (2010) for reviews of the studies on RCPSP.

In the model we study, we consider a single renewable resource: the labor required for completing the activities. The workers can work under different modes in a period, i.e., we have multi-mode resource assignment. One may refer to Weglarz et al. (2011) and Mika et al. (2015) for reviews of multi-mode project scheduling problems and to Van Peteghem and Vanhoucke (2014) for a study comparing various metaheuristic approaches for multi-mode RCPSP. In our model, while individual activities have a limit on the amount of the resource they can utilize, we do not have a resource constraint. It is noted by Herroelen (2005) that, resource constraint is not the main concern in project scheduling for practitioners. Instead of a resource constraint, as noted above, one of the objectives of our model is to minimize the maximum (peak) resource usage throughout the project duration. To this end, our model corresponds to a multi-objective project scheduling problem. As noted by Ballestín and Blanco (2011), multi-objective project scheduling problem, compared to single-objective project scheduling problem, is less studied in the literature. We refer the reader to Viana and Sousa (2000) and Ballestín and Blanco (2011; 2015) for basics of multi-objective RCPSP.

In most of the project scheduling research, preemption is not allowed (see, Ballestín et al. (2008)). That is, once an activity is started, it is not interrupted until completion. Specifically, when objectives other than makespan minimization are considered, preemption can be beneficial. In this study, neither of the objectives is

makespan minimization and we allow preemption. One may refer to Balestin et al. (2008), Quintanilla et al. (2015), and Schwindt and Paetz (2015) for overviews of preemption in project scheduling. Here, we determine the number and mode of workers to assign to each activity within each period throughout the project schedule; and, the work on an activity can be interrupted. For instance, it is allowed to assign 5 workers in mode 1, 10 workers in mode 2, no workers, and 5 workers in mode 1 to an activity during 4 consecutive periods within the project schedule. This indicates that, in addition to preemption, the project scheduling problem under investigation in this study allows flexible resource profiles.

Typically, in multi-mode project scheduling problems, the time to complete an activity and the amount of resource used for completion under each mode are given. On the other hand, in flexible resource profile project scheduling, each activity requires a specified amount of a resource (referred to as work-content by Fundeling and Trautmann (2010) for labor requirements and as resource requirement by Naber and Kolish (2014) for generic resources) and the amount of resource(s) allocated to the activities in each period (i.e., work-profiles or resource-profiles) are determined in addition to project start and finish times. RCPSP with flexible profiles (FRCPSP) has been relatively recently studied with discrete or continuous resources as well as under discrete or continuous times. Particularly, Fundeling and Trautmann (2010), Ranjbar and Kianjar (2010), Baumann and Trautmann (2014), Baumann et al. (2015), and Zimmermann (2016) analyze FRCPSP with discrete resources under discrete time. Fundeling and Trautmann (2010) propose priority-rule based heuristic, Ranjbar and Kianjar (2010) develop a genetic algorithm, and Zimmerman (2016) discuss a mixed-integer-programming based

heuristic approach, while the others present formulations and test instances with commercial solvers. On the other hand, Naber and Kolisch (2014) and Tritscheler et al. (2014; 2017) focus on FRCPSP with continuous resources under discrete time. Naber and Kolisch (2014) formulate different models for the problem and propose a priority-rule based heuristic approach. Tritscheler et al. (2014) discuss a genetic algorithm while Tritscheler et al. (2017) develop a hybrid metaheuristic method. Finally, Naber (2017) studies FRCPSP with continuous resources under continuous time and develops a branch-and-cut method for the problem.

In this study, we consider discrete time and a single discrete resource, which is renewable and unconstrained. The above studies on FRCPSP consider a single mode, do not allow preemption, and aim at minimizing project makespan. Different than these studies, as previously noted, preemption is allowed and there are multiple modes. Furthermore, rather than makespan minimization, we consider two objectives: cost minimization and maximum (peak) resource minimization. We will discuss the related project scheduling studies based on the objective functions considered while presenting the model in Section 2, especially related to resource leveling. The main motivation for our model is the need for planning the construction of the semiconductor manufacturing facility for the new production run. In what follows, we review the project scheduling studies that focus on scheduling activities for semiconductor facility ramp-up.

Cheng et al. (2012) study a multi-mode resource-constrained project scheduling for ramping up a semiconductor manufacturing facility. They consider multiple constrained resources and assume that an activity should be completed in a single mode (the model is basically the classical multi-mode RCPSP without preemption). The authors

propose a simulated annealing method integrated with simulation for makespan minimization and present a case study. Later, in Cheng et al. (2015), the authors analyze a similar multi-mode resource constraint project scheduling problem under three formulations: with preemption, with non-preemptive activity splitting (i.e., preemption is allowed only if resources are not sufficient at a period), and without activity splitting. They propose a modified branch-and-bound algorithm as the exact method and develop priority-rule based heuristic method. In our study, different than Cheng et al. (2012; 2015), we consider a single unconstrained resource with flexible profiles. In addition to project scheduling decisions, we also consider worker assignment decisions such that a worker can work in different modes. In an earlier work (Dwyer and Konur (2018)), we presented a similar model with multiple constrained resources and only discussed how to linearize the formulation in order to incorporate a resource leveling objective. Later in Dwyer and Konur (2020), we compared different resource leveling functions using a case study for the model presented in Dwyer and Konur (2018) with a single resource. The current study uses linearization from Dwyer and Konur (2018); and based on our preliminary analysis, we adopt to minimize the maximum resource use in addition to cost minimization; therefore, we do not have resource constraint in the current study. As mentioned above, we will review the resource leveling problems in Section 2.

In summary, our model is a bi-objective multi-mode flexible resource profile project scheduling problem with a single unconstrained renewable discrete resource under discrete time. For this model, we first discuss the implementation of the well-known ϵ -constraint method (see, e.g., (Özlen and Azizoğlu (2009))) to generate the exact Pareto front. After that, we propose an approximation method to generate near Pareto

efficient solutions. The approximation method first solves partial linear relaxation of the subproblems that are required to be solved within the ε -constraint method; then, a rounding approach is utilized to convert the non-integer solutions to integer ones. Finally, an improvement heuristic is used to enhance the converted solutions. Based on a set of numerical studies, we conclude that the approximation method can generate solutions within proximity of the exact Pareto front with significantly less computational time.

The contributions of this study are as follows: a novel model for a project scheduling problem motivated by semiconductor manufacturing facility construction is presented and a simple and computationally efficient approximation method is developed for finding near Pareto efficient solutions for the resulting bi-objective model. In addition, we discuss the details of problem instance generation and post the data and solutions of the problem instances generated for interested researchers. The rest of the paper is organized as follows. Section 2 discusses the details of the problem settings and presents the mathematical formulation of the model. In Section 3, we explain the implementation of the ε -constraint method and develop our approximation method. The setup and the results of the numerical studies are detailed in Section 4. Concluding remarks and possible future research directions are noted in Section 5.

2. PROBLEM SETTINGS AND FORMULATION

The problem of interest in this study is scheduling of construction of special tools/machines that transform raw silicon into integrated semiconductor circuits. The construction of a tool/machine can be the demolition of an existing tool/machine because

its technology is outdated, modification of an existing tool/machine so that its technology is updated, or installation of a completely new tool/machine. We refer to construction of a tool/machine as an activity and let the activities be indexed by i and j such that $i, j \in I = \{1, 2, \dots, |I|\}$. While it is possible that there exist stand-alone construction activities, a demolition project may be required to be completed before a specific installation activity (mainly due to cleanroom requirements) and some installation activities should be completed before others due to required connections along the production line. Such requirements necessitate precedence relations and, accordingly, we define $a_{ij} = 1$ if activity i is to be completed before activity j can start, and $a_{ij} = 0$ otherwise. Let A denote the set of (i, j) pairs such that $a_{ij} = 1$, that is, A defines the set of activity pairs that have precedence relations.

All of the activities should be finished before the targeted start date of the complete production line. The length of the scheduling horizon is $|T|$ periods and let the periods be indexed by r and t such that $r, t \in T = \{1, 2, \dots, |T|\}$. That is, the activities should be completed by the end of period $|T|$. We note that project scheduling problems typically aim at minimizing the makespan of the project (see, e.g., Ballestín and Blanco (2011)). Here, since there is a target date for finishing the project, i.e., all activities should be completed by this target date, makespan is not of concern. It is also worthwhile to note that there are studies that have upper bound limits on the project makespan (see, e.g., Neumann and Zimmermann (1999) and the deadline problem Brucker et al. (1999)).

Based on the working conditions considering the scheduling practice, we focus on scheduling project activities on a weekly basis, i.e., each period is one week (one can easily modify the modeling and solution methods discussed for other period definitions

such as days for shorter horizons or months for longer horizons). An activity cannot start prior to a specific date; this is typically because, for an installation task, parts should be delivered; and, for a demolition/modification task, the current work assigned to the tool/machine should be completed. Also, individual activities might have completion due dates earlier than the completion of the whole production line (project); this is typically because, for a demolition/modification task, the tool/machine or its parts may be needed to be transferred to other facilities; and, for an installation task, the new tool/machine may be required to be up and running to enable output from the production line by the targeted start date. Accordingly, let $e_i \in T$ and $d_i \in T$ denote the earliest start time (beginning of a week) and due date (end of a week) for activity i , respectively, such that $e_i \geq d_i \forall i \in I$. Similar time windows for individual activities are commonly defined for project scheduling problems in general (see, e.g., Hartmann and Briskorn (2010)) as well as for the flexible resource profile project scheduling problems (see, e.g., Naber and Kolisch (2014), Tritscheler et al. (2017), Naber (2017)). Furthermore, Cheng et al. (2012; 2015) also define ready time and due dates for the activities in semiconductor manufacturing facility ramp-up problem. Without loss of generality, we consider that $\min_{i \in I} \{e_i\} = 1$ and $\max_{i \in I} \{d_i\} = |T|$.

Each activity requires a given amount of construction (labor) hours, denoted by $H_i > 0$ for activity i , to be completed. This is referred to as work-content in FRCPS research (see, e.g., Fündeling and Trautmann (2010), Zimmermann (2016)). We need to assign workforce to the activities throughout the scheduling horizon. Worker assignments are based on a weekly schedule such that a worker will work on the same activity during a week and he/she will have the same working mode throughout the week. Specifically,

each worker can be assigned to work on one of the different modes during a week and will not change modes within a week. Let the working modes be indexed by k such that $k \in K = \{1, 2, \dots, |K|\}$. Furthermore, let $h_k > 0$ and $c_k > 0$ denote the number of hours a worker in mode k works per week and the hourly cost of a worker in mode k . As it is practical that the workers form weekly teams such that each team is given the same guidelines and work on the same activity as a team, all the workers assigned to the same activity during a week will have the same mode. This is also consistent with the safety requirement for a minimum number of workers that should simultaneously work on an activity.

It is considered that there is an ample amount of workforce that can be utilized each week; however, there are limits on the number of workers that can be utilized for individual activities. Due to safety and functional requirements (e.g., multiple workers are needed for minimizing the impact of possible accidents or for building the physical capacity required by a specific activity), there is a lower limit on the number of workers who can simultaneously work on an activity during a week. Furthermore, due to safety and functional requirements (e.g., having more workers than needed decreases safety), space limitations (e.g., since the construction takes place in an active manufacturing environment, too many workers might impact the ongoing production line), and congestion considerations (e.g., after a threshold number, having additional workers on an activity at the same time does not add value), there is also an upper limit on the number of workers who can simultaneously work on an activity during a week. Accordingly, let $l_i \geq 1$ and $u_i \geq l_i$ be the minimum and the maximum number of workers who should and can simultaneously work on activity i during a week,

respectively. Similar bounds on resources that can be simultaneously assigned to an activity are defined in FRCPSPP studies (see, e.g., Naber and Kolisch (2014), Tritscheler et al. (2017), Naber (2017)).

It is important to note that an activity is not required to be worked on continuously from its start to its completion. That is, it is allowed that a team of workers work on an activity for several weeks on a specific mode, then no workers work on the activity for several weeks, and then another team of workers continue working on the activity on another mode. That is, we allow preemption while scheduling the project activities with flexible worker profiles. This indicates that the scheduling problem has two main set of decisions: worker assignments and project schedules.

The worker assignment decisions include: the number of workers assigned to each activity each week, and the mode for the team of workers working on each activity each week. To avoid nonlinearities in model formulation, rather than defining number of workers assigned to each activity each week, we define the number of workers assigned to each activity on each mode during each week (see, e.g., Dwyer and Konur (2018)). Let $x_{ikt} \geq 0$ denote the (integer) number of workers assigned to activity i on mode k during week t and \mathbf{X} be the integer $|I| \times |K| \times |T|$ -array of x_{ikt} worker-assignment variables. As there are minimum and maximum limits on x_{ikt} and the workers on the same activity should work on the same mode through the week, we define $z_{ikt} = 1$ if the workers on activity i are working on mode k during week t , and $z_{ikt} = 0$ otherwise, and let \mathbf{Z} be the binary $|I| \times |K| \times |T|$ -array of z_{ikt} mode-selection variables.

The project schedule decisions include: the start and finish times for the project activities. These are needed to be determined to ensure the precedence relations in the model. In particular, let $s_{it} = 1$ if the work on activity i starts at the beginning of week t , and $s_{it} = 0$ otherwise; and \mathbf{S} be the binary $|I| \times |T|$ -matrix of s_{it} activity-start-time variables. Similarly, let $f_{it} = 1$ if the work on activity i finishes at the end of week t , and $f_{it} = 0$ otherwise; and \mathbf{F} be the binary $|I| \times |T|$ -matrix of f_{it} activity-finish-time variables. Note that $\sum_{t \in T} ts_{it}$ and $\sum_{t \in T} tf_{it}$ define activity i 's start and finish weeks, respectively. To avoid notational confusion, we define $\tau_t = t$ as the parameter defining week t (because t is used as an index) and let $\sum_{t \in T} \tau_t s_{it}$ and $\sum_{t \in T} \tau_t f_{it}$ define activity i 's start and finish week, respectively. Recall that an activity is not required to be continuously worked on; therefore, it is possible that $\sum_{k \in K} x_{ikt} = 0$ for activity i for some t such that $\sum_{t \in T} \tau_t s_{it} < t < \sum_{t \in T} \tau_t f_{it}$.

Cost minimization is an important criterion regarded by many companies. In project scheduling literature, even though time-based objectives, especially makespan minimization, are the most commonly used ones (see, e.g., Hartman and Briskorn (2010), Ballestin and Blanco (2011)), cost related objectives are also used in so-called time-cost trade-off problems (see, e.g., Brucker et al. (1999)) as well as time-resource trade-off problems (see, e.g., Weglarz et al. (2011)). We refer the reader to the survey papers cited in Section 1 for overviews of various project scheduling problems with cost related objectives. One particularly relevant project scheduling problem with cost related objective is the resource availability cost problem, which aims to minimize the cost of the resources used in order to finish the project before a deadline (Rodrigues and Yamashita (2010; 2015), Kreter et al. (2018)). Kreter et al. (2018) provide a detailed review of

resource availability cost problems and the solution approaches discussed. In the semiconductor facility construction scheduling problem of interest in this study, the variable resource costs incurred are the payments made to the workers. The total variable cost depends on the number of workers working on each mode during each week. It then follows that the total cost of the construction plan is $\sum_{i \in I} \sum_{k \in K} \sum_{t \in T} c_k h_k x_{ikt}$.

During the scheduling horizon, minimizing the total cost might result in worker assignments with significant variations in the total number of workers utilized per week. These changes in the number of workers utilized is especially not favored by the worker trades. Therefore, having balanced worker assignments throughout the planning horizon is as important as the cost of the construction schedule. In project scheduling literature, balanced resource profiles are typically sought in so-called resource leveling problems (see, e.g., Rieck and Zimmermann (2015)). Different objective functions are defined and used for resource leveling problems such as minimizing the maximum difference in the amount of resource used, minimizing the deviations from a desired resource profile, and minimizing the sum of squared resource usages. One can refer to Neumann et al. (2003) for various leveling objectives (also, Damci et al. (2016) list 10 different leveling objectives). It is important to note that, several cost-related objective functions are defined for resource leveling such as total adjustment cost (see, e.g., Kreter et al. (2014)), resource overload (see, e.g., Neumann et al. (2003), Rieck et al. (2012)), and release and rehire cost (see, e.g., Atan and Eren (2018)). Moreover, resource availability/investment cost problem mentioned above (see, e.g., Neumann and Zimmermann (1999), Rodrigues and Yamashita (2010; 2015), Kreter et al. (2018), Coughlan et al. (2015)) is also considered as resource leveling problem. In this study, as noted above, we aim to

minimize total resource (labor) cost; hence, for resource leveling, we consider minimization of the maximum (peak) resource needed.

As remarked by Takamoto et al. (1995) as well, minimizing the maximum resource use helps level resource profile. Similarly, Atan and Eren (2018) note that minimizing maximum resource usage is a resource leveling metric. Furthermore, Caramia and Dell’Olmo (2003) discuss that minimizing the peak resource level can be desired even if the resource usage is constant throughout the project duration. Caramia and Dell’Olmo (2003) study a single-mode project scheduling problem (without constrained resources) and propose heuristic approaches for the problem with makespan and peak resource use minimization objectives. Given that we already consider cost minimization, as noted above, we choose to minimize the maximum resource needed as the resource leveling objective. Also, it is worthwhile to remark that resource leveling is especially important during preplanning phase in project scheduling (Neumann et al. (2003)). For our problem, since we deal with worker assignment in addition to scheduling, our focus is more on the preplanning phase of the project scheduling for the semiconductor manufacturing facility construction problem. This preplanning includes contracting with worker trade to plan the labor requirements for the project. Minimizing the maximum number of workers within a week therefore provides a level of robustness for construction by minimizing the impact the week with the maximum number of workers can have on the schedule in case of unforeseeable disruptions in the work force. Furthermore, doing so reduces the pressure on the worker trade. The maximum number of workers used in a week is equal to $\max_{t \in T} \{\sum_{i \in I} \sum_{k \in K} x_{ikt}\}$, which is not a linear function

of \mathbf{X} . To overcome this, we introduce X^{max} as the auxiliary variable defining the maximum number of workers used in a week.

Based on the above discussion, construction scheduling problem (**CSP**) with total cost and maximum number of workers minimization objectives can be formulated as follows.

CSP:

$$\text{Minimize } \sum_{i \in I} \sum_{k \in K} \sum_{t \in T} c_k h_k x_{ikt}$$

$$\text{Minimize } X^{max}$$

$$\text{subject to } \sum_{k \in K} \sum_{t \in T} h_k x_{ikt} \geq H_i \quad \forall i \in I \quad (1)$$

$$\sum_{t \in T} s_{it} = 1 \quad \forall i \in I \quad (2)$$

$$\sum_{t \in T} f_{it} = 1 \quad \forall i \in I \quad (3)$$

$$\sum_{t \in T} \tau_t s_{it} \geq e_i \quad \forall i \in I \quad (4)$$

$$\sum_{t \in T} \tau_t f_{it} \leq d_i \quad \forall i \in I \quad (5)$$

$$\sum_{t \in T} \tau_t f_{it} \leq \sum_{t \in T} \tau_t s_{jt} - 1 \quad \forall (i, j) \in A \quad (6)$$

$$\sum_{k \in K} z_{ikt} \leq \sum_{r=1}^t s_{ir} \quad \forall i \in I, \forall t \in T \quad (7)$$

$$\sum_{k \in K} z_{ikt} \leq \sum_{r=t}^{|T|} f_{ir} \quad \forall i \in I, \forall t \in T \quad (8)$$

$$\sum_{k \in K} x_{ikt} \geq s_{it} \quad \forall i \in I, \forall t \in T \quad (9)$$

$$\sum_{k \in K} x_{ikt} \geq f_{it} \quad \forall i \in I, \forall t \in T \quad (10)$$

$$x_{ikt} \geq l_i z_{ikt} \quad \forall i \in I, \forall k \in K, \forall t \in T \quad (11)$$

$$x_{ikt} \leq u_i z_{ikt} \quad \forall i \in I, \forall k \in K, \forall t \in T \quad (12)$$

$$X^{max} \geq \sum_{i \in I} \sum_{k \in K} x_{ikt} \quad \forall t \in T \quad (13)$$

$$x_{ikt} \in \{0,1,2, \dots\} \quad \forall i \in I, \forall k \in K, \forall t \in T \quad (14)$$

$$z_{ikt} \in \{0,1\} \quad \forall i \in I, \forall k \in K, \forall t \in T \quad (15)$$

$$s_{it} \in \{0,1\} \quad \forall i \in I, \forall t \in T \quad (16)$$

$$f_{it} \in \{0,1\} \quad \forall i \in I, \forall t \in T \quad (17)$$

$$X^{max} \in \{0,1,2, \dots\}. \quad (18)$$

In **CSP**, total cost and maximum number of workers utilized are minimized.

Constraints in (1) assure that each activity gets the workhours needed for its completion.

Constraints (2) and (3) restrict that a single week is designated as the start and finish week for an activity, respectively; and, constraints (4) and (5) ensure an activity is worked on only after its earliest start time and before its due date, respectively.

Constraints (6) enforce the precedence relations between each pair of activities, which have a precedence relation. Constraints (7) and (8), together with constraints (2) and (3), guarantee that at most one mode is selected for the workers on an activity during the time between the activity's start and finish weeks, and no mode is selected for the weeks before the start and after the finish of the activity. Constraints (9) and (10) are introduced to eliminate symmetric solutions and they guarantee that there is at least one worker in the weeks an activity starts and ends, respectively. These avoid considering feasible solutions where an activity's start week is earlier than the first week the activity is worked on and/or where an activity's finish week is later than the last week the activity is worked on. Constraints (11) and (12) define the lower and upper limits on the number of workers to be assigned to an activity during a week, respectively, if a team of workers is assigned to the activity. Because X^{max} is minimized, constraints (13) assure that X^{max} is indeed the maximum number of workers utilized in a week throughout the project

horizon. Finally, constraints (14)-(18) state the integer and/or binary definitions for the worker assignment (x_{ikt} and z_{ikt}), project schedule (s_{it} and f_{it}), and maximum number of workers utilized (X^{max}) decision variables, respectively.

Table 1 summarizes the notation used and additional notation will be defined as needed. A construction plan is defined by $\langle \mathbf{X}, \mathbf{Z}, \mathbf{S}, \mathbf{F} \rangle$. **CSP** is a bi-objective integer linear programming model with $2|I||T||K| + 4|I||T| + 5|I| + |T| + |A|$ constraints (excluding binary/integer definitions) and $2(|I||T||K| + |I||T|) + 1$ variables. Furthermore, single-objective **CSP** with a single project without lower and upper bounds on the number of workers is a knapsack problem (particularly, due to constraints (1)); hence, even single-objective **CSP** is NP-hard. Therefore, in what follows, we develop a heuristic method for solving **CSP**.

3. SOLUTION METHOD

Two common approaches to solve multi-objective models are (i) reducing the problem into a single-objective model and finding the optimum solution for the resulting single-objective model and (ii) generating Pareto efficient solutions for the multi-objective model. In this study, we adopt approach (ii) for **CSP** as this approach gives the decision maker a set of alternative solutions, which can then be evaluated and compared using the objectives as well as other measures.

Note that, once \mathbf{X} is known, the other variables (i.e., \mathbf{Z} , \mathbf{S} , \mathbf{F} , and X^{max}) can be easily determined. Therefore, for notational simplicity, we use \mathbf{X} to represent a solution of **CSP** and let χ denote the sets of feasible solutions (\mathbf{X} 's) of **CSP**. Furthermore, we let

$TC(\mathbf{X})$ and $MW(\mathbf{X})$ denote the total cost (i.e., $TC(\mathbf{X}) = \sum_{i \in I} \sum_{k \in K} \sum_{t \in T} c_k h_k x_{ikt}$) and the maximum number of workers utilized (i.e., $MW(\mathbf{X}) = X^{max} = \max_{t \in T} \{\sum_{i \in I} \sum_{k \in K} x_{ikt}\}$) for solution \mathbf{X} , respectively. Then, **CSP** is to minimize $TC(\mathbf{X})$ and $MW(\mathbf{X})$ such that $\mathbf{X} \in \chi$.

Table 1. Notation.

Sets and Indices:	
$i, j \in I$	Indices used for and the set of activities
$r, t \in T$	Indices used for and the set of periods (weeks)
$k \in K$	Index used for and the set of modes
$(i, j) \in A$	Representation and the set of precedence relations
Parameters:	
a_{ij}	Binary indicator for precedence relation between activities $i \in I$ and $j \in I$
τ_t	Time indicator for period $t \in T$
e_i, d_i	Earliest start time and due date (i.e., time-window) for activity $i \in I$
l_i, u_i	Minimum and the maximum number of workers who should and can simultaneously work on activity $i \in I$ during a week, respectively
H_i	Amount of construction (labor) hours needed by activity $i \in I$ to be completed
h_k, c_k	Number of hours per week a worker works and the hourly cost of a worker in mode $k \in K$, respectively
Decision variables:	
x_{ikt}, \mathbf{X}	Integer number of workers assigned to activity $i \in I$ in mode $k \in K$ during period $t \in T$ and the array of x_{ikt} variables, respectively
z_{ikt}, \mathbf{Z}	Binary indicator for workers assigned to activity $i \in I$ in mode $k \in K$ during period $t \in T$ and the array of z_{ikt} variables, respectively
s_{it}, \mathbf{S}	Binary indicator for activity $i \in I$ starting at the beginning of period $t \in T$ and the matrix of s_{it} variables, respectively
f_{it}, \mathbf{F}	Binary indicator for activity $i \in I$ finishing at the end of period $t \in T$ and the matrix of f_{it} variables, respectively
X^{max}	Maximum number of workers utilized throughout the project horizon

A solution $\mathbf{X}' \in \chi$ is Pareto efficient for **CSP** if and only if there does not exist another solution $\mathbf{X}'' \in \chi$ such that $TC(\mathbf{X}'') \leq TC(\mathbf{X}')$, $MW(\mathbf{X}'') \leq MW(\mathbf{X}')$, and $[TC(\mathbf{X}''), MW(\mathbf{X}'')] \neq [TC(\mathbf{X}'), MW(\mathbf{X}')]$. Given $\mathbf{X}' \in \chi$ is Pareto efficient for **CSP**, the

point $[TC(\mathbf{X}'), MW(\mathbf{X}')] is a non-dominated point for \mathbf{CSP} and let PF and PE denote the set of non-dominated points (i.e., Pareto front) and Pareto efficient solutions of \mathbf{CSP} , respectively. In what follows, we first discuss implementation of the well-known classical ε -constraint method to generate PF . After that, we present a method based on partial linear relaxation to generate approximated PF , denoted by \widehat{PF} .$

3.1. CLASSICAL ε -CONSTRAINT METHOD

One of the most used methods for solving bi-objective optimization models is the ε -constraint method. In the ε -constraint method, one of the objective functions is optimized while the other objective function is incorporated as a constraint with an upper bound on its value. This upper bound is iteratively reduced by ε until the constrained sub-problem becomes infeasible. While ε -constraint method would approximate the continuous Pareto front for continuous optimization models, it generates the exact finite Pareto front for bi-objective integer optimization models. As \mathbf{CSP} is a bi-objective integer optimization model with a finite Pareto front, we can generate PF using ε -constraint method.

Note that $MW(\mathbf{X})$ is an integer-valued function by definition and, without loss of generality, one can consider that $TC(\mathbf{X})$ is also an integer-valued function (by simply multiplying $c_k h_k$ values with a sufficiently large number so that the products are integers). Therefore, we can implement the so-called classical ε -constraint method (Özlen and Azizoglu (2009)), which iteratively solves constrained weighted single-objective integer programming (CWSOIP) models to optimality $|PF|$ times. The CWSOIP for \mathbf{CSP} , simply referred to as sub-problem (\mathbf{SP}), reads as follows.

SP:

$$\text{Minimize } TC(\mathbf{X}) + \phi MW(\mathbf{X})$$

$$\text{subject to } MW(\mathbf{X}) \leq \Delta$$

$$\mathbf{X} \in \chi.$$

The optimal solution of **SP** corresponds to a point on PF under two conditions: (i) ϕ is sufficiently small such that $\phi(MW(\mathbf{X}') - MW(\mathbf{X}'')) < 1 \forall \mathbf{X}', \mathbf{X}'' \in \chi$ and (ii) $\Delta \in [\min\{MW(\mathbf{X}): \mathbf{X} \in \chi\}, \min\{MW(\mathbf{X}): TC(\mathbf{X}) = \min\{TC(\mathbf{X}): \mathbf{X} \in \chi\}, \mathbf{X} \in \chi\}]$.

Condition (i) is necessary and it assures that the optimum solution of **SP** is the one which minimizes $TC(\mathbf{X})$ and minimizes $MW(\mathbf{X})$ over all the alternative solutions with the minimum cost value for **SP**. As noted in Özlen and Azizoğlu (2009), one can define $\phi = \frac{1}{MW_{max} - MW_{min} + 1}$, where $MW_{max} \geq \max\{MW(\mathbf{X}): \mathbf{X} \in \chi\}$ and $MW_{min} \leq \min\{MW(\mathbf{X}): \mathbf{X} \in \chi\}$.

Remark 1. Let $\phi = \frac{1}{\max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} + 1}$. Then $\phi(MW(\mathbf{X}') - MW(\mathbf{X}'')) < 1 \forall \mathbf{X}', \mathbf{X}'' \in \chi$.

Proof. First, it can be noted from constraints (4)-(5), (7)-(8), and (11)-(12) that

$\sum_{i \in I} \sum_{k \in K} x_{ikt} \leq \sum_{i \in I: e_i \leq t \leq d_i} u_i \forall t \in T$ (i.e., the maximum number of workers that can be utilized in a given period is less than or equal to the sum of the maximum number of workers that can be utilized for a project over the projects that can be worked on during that period). This implies that $X^{max} \leq \max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} \forall \mathbf{X} \in \chi$, which means we have $\max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} \geq \max\{MW(\mathbf{X}): \mathbf{X} \in \chi\}$. Also, by definition, we have $X^{max} \geq$

$0 \forall \mathbf{X} \in \chi$, i.e., $0 \leq \min\{MW(\mathbf{X}): \mathbf{X} \in \chi\}$. Therefore, $\frac{1}{\max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} - 0 + 1} (MW(\mathbf{X}') - MW(\mathbf{X}'')) < 1 \forall \mathbf{X}', \mathbf{X}'' \in \chi$. ■

Based on Remark 1, we use $\phi = \frac{1}{\max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} + 1}$ while solving **SP**.

Condition (ii) defines the range of $MW(\mathbf{X})$ values of PF . Particularly, we have $MW(\mathbf{X}) \geq \min\{MW(\mathbf{X}): \mathbf{X} \in \chi\}$ by definition, and any point with $MW(\mathbf{X}) > \min\{MW(\mathbf{X}): TC(\mathbf{X}) = \min\{TC(\mathbf{X}): \mathbf{X} \in \chi\}, \mathbf{X} \in \chi\}$ is dominated by $[\min\{TC(\mathbf{X}): \mathbf{X} \in \chi\}, \min\{MW(\mathbf{X}): TC(\mathbf{X}) = \min\{TC(\mathbf{X}): \mathbf{X} \in \chi\}, \mathbf{X} \in \chi\}]$, therefore, is not on PF .

In the ε -constraint method, Δ is iteratively reduced while solving **SP**. As the objective functions are integer valued in **CSP**, we set $\varepsilon = 1$. Let \mathbf{X}^Δ be the solution of **SP** for a given Δ . Algorithm 1 gives the description of the classical ε -constraint method for generating PF .

Algorithm 1: Classical ε -constraint method for CSP	
Step 0.	Let $\phi = \frac{1}{\max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\} + 1}$, $\Delta = \max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\}$, $PF = \emptyset$ and $PE = \emptyset$. Go to Step 1.
Step 1.	Solve SP given ϕ and Δ . i. If feasible, go to Step 2. ii. Else, go to Step 3.
Step 2.	Set $PF := PF \cup \{[TC(\mathbf{X}^\Delta), MW(\mathbf{X}^\Delta)]\}$, $PE := PE \cup \{\mathbf{X}^\Delta\}$, and $\Delta = MW(\mathbf{X}^\Delta) - 1$; then, go to Step 1.
Step 3.	Stop and return PF and PE .

Note that, in Step 0 of Algorithm 1, $\Delta = \max_{t \in T} \{\sum_{i \in I: e_i \leq t \leq d_i} u_i\}$ and this is sufficiently large such that $MW(\mathbf{X}) \leq \Delta$ is redundant when **SP** is solved for the first time (see, e.g., proof of Remark 1). Also, it is important to note that while PF is the exact Pareto front for **CSP**, it is possible that PE is not because, even if unlikely, there might be

different solutions corresponding to the same non-dominated point. As solutions corresponding to the same non-dominated point is equally preferable for the decision maker, we consider that it is sufficient to generate one solution for each non-dominated point in PF .

One can note that **SP** is an integer programming mode and it is also NP-hard. Given that **SP** is to be solved $|PF|$ times, Algorithm 1 becomes computationally burdensome as the problem size grows. Therefore, it is important to develop computationally efficient heuristic methods that will generate near-Pareto efficient solutions. Next, we discuss the details of the heuristic method we propose to approximate PF .

3.2. PARTIAL LINEAR RELAXATION BASED APPROXIMATING METHOD

Our approximation method is based on partial linear relaxation of **SP**, which is iteratively solved in Algorithm 1. Specifically, to reduce the number of integer variables in **SP**, we focus on solving its relaxed version and generate a set of possibly non-integer solutions. After that, we first use a rounding procedure to convert such solutions to integer solutions; then use an improvement procedure to improve the rounded solutions; and finally determine the Pareto efficient solutions within the set of integer solutions generated. Accordingly, our heuristic method to approximate PF , i.e., generate \widehat{PF} , consists of four main phases:

(i) *Generating relaxed efficient solutions*: In this phase, we execute Algorithm 1 such that, instead of solving **SP** in Step 1, we solve its partial linear relaxation, denoted by **RSP**, which allows x_{ikt} variables to be continuous while all other variables are binary

or integer. We refer to the solutions generated by using a partial linear relaxation of **SP** in Algorithm 1 as *relaxed efficient solutions* and let \widetilde{PE} be the set of relaxed efficient solutions, \widetilde{PF} be the corresponding set of points, and $\widetilde{X} \in \widetilde{PE}$ denote an arbitrary relaxed efficient solution.

(ii) *Rounding non-integer relaxed efficient solutions*: In this phase, each relaxed efficient solution $\widetilde{X} \in \widetilde{PE}$ goes through a rounding process, which assures that the returned solution, denoted by \widehat{X} , respects project schedules and the resulting x_{ikt} variables are integer. We note that rounding approaches have been used in assignment and staffing problems as well as for generic integer programming models (see, e.g., Vohra (1988), Saltzman and Hillier (1992), Miller and Franz (1996)). Here, rounding is done by executing Procedure 1, which is detailed below. Because of the randomness in Procedure 1, it is possible to generate different integer solutions from the same relaxed efficient solution. Therefore, to create alternative rounded solutions, we apply Procedure 1 on each $\widetilde{X} \in \widetilde{PE}$ for a pre-specified number of times, denoted by N .

(iii) *Improving rounded solutions*: In this phase, each rounded solution goes through an improvement process, Procedure 2, which aims to decrease the maximum number of workers utilized. We use Procedure 2 on \widehat{X} and generate \overline{X} .

(iv) *Determining Pareto efficient improved rounded solutions*: At the end of phase (iii), we have a set of alternative integer solutions (a set of \widehat{X} s and \overline{X} s). In this phase, we determine the Pareto efficient solutions within this set of alternative integer solutions and the corresponding non-dominated points using Procedure 3.

Algorithm 2 gives the description of the heuristic approximation method for generating \widehat{PF} . In the description of Algorithm 2 as well as the procedures detailed next,

we let $U(\ell)$ denote the ℓ^{th} element of set U . Steps 1, 2, 3, and 4 of Algorithm 1 correspond to phases (i), (ii), (iii), and (iv), respectively, and the details of these phases are explained next.

Algorithm 2: Partial linear relaxation based rounding heuristic for CSP	
Step 0.	Given N , go to Step 1.
Step 1.	Execute Algorithm 1 by solving RSP in Step 1 and let \widetilde{PE} be the set of returned relaxed efficient solutions. Set $P = \emptyset$ and go to Step 2.
Step 2.	For $\ell = 1: \widetilde{PE} $ Let $\widetilde{X} = \widetilde{PE}(\ell)$. For $n = 1: N$ Execute Procedure 1 with \widetilde{X} , generate \widehat{X} , and set $P := P \cup \{\widehat{X}\}$. End End Set $\widehat{P} = P$ and go to Step 3.
Step 3.	For $\ell = 1: P $ Let $\widehat{X} = P(\ell)$, execute Procedure 2 with \widehat{X} , generate \overline{X} , and set $\widehat{P} := \widehat{P} \cup \{\overline{X}\}$. End Go to Step 4.
Step 4.	Execute Procedure 4 with \widehat{P} and return $\widehat{PF} = PF(\widehat{P})$ and $\widehat{PE} = PE(\widehat{P})$.

(i) Generating relaxed efficient solutions: The partial linear relaxation of **SP**, i.e., **RSP**, replaces constraints (14) in definition of χ with $x_{ikt} \geq 0 \forall i \in I, \forall k \in K, \forall t \in T$. Relaxing only x_{ikt} variables makes converting a potentially non-integer solution to a feasible integer solution easy because one does not need to consider mode selection and scheduling restrictions for feasibility (i.e., variables \mathbf{Z} , \mathbf{S} , and \mathbf{F} do not need to be modified). Also note that, when **RSP** solved, the maximum numbers of workers utilized in a week can be non-integer in the resulting solution because of non-integer x_{ikt} values. Nevertheless, the resulting solution's $MW(\mathbf{X})$ will be integer because we still restrict variable X^{max} be to integer (i.e., constraint (18) is valid) in **RSP**. Therefore, using $\varepsilon = 1$

in Step 2 of Algorithm 1 basically updates Δ to be an integer, which corresponds to the rounded-down value of the non-integer maximum number of workers utilized in a week. Each relaxed efficient solution $\tilde{\mathbf{X}} \in \widetilde{PE}$ is converted to N alternative integer solutions using a rounding approach, which is detailed next.

(ii) Rounding non-integer relaxed efficient solutions: In **CSP**, the projects are related to each other mainly through precedence relations, i.e., constraints (6). Furthermore, definition of X^{max} , i.e., constraints (13), are not required to be satisfied to have a functioning schedule. Therefore, if we do not change $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{F}}$ variables (i.e., projects start and finish time variables corresponding to $\tilde{\mathbf{X}}$) while rounding $\tilde{\mathbf{X}}$, we can work on each project separately. To this end, our rounding approach focuses on rounding only non-integer \tilde{x}_{ikt} variables (note that $\tilde{x}_{ikt} = 0$, i.e., integer, for $t < \sum_{t \in T} \tau_t \tilde{s}_{it}$ and $t > \sum_{t \in T} \tau_t \tilde{f}_{it} \forall i \in I$ and $\forall k \in K$). While rounding, we also need to be mindful of the increases in $TC(\mathbf{X})$ as well as $MW(\mathbf{X})$. For instance, simply rounding-up all non-integer \tilde{x}_{ikt} variables in $\tilde{\mathbf{X}}$ will produce a feasible integer schedule; however, the total cost and the maximum of the number of workers utilized can significantly increase. Our rounding approach allows both rounding-down and -up of a non-integer \tilde{x}_{ikt} value by decreasing or increasing the total number of hours provided by \tilde{x}_{ikt} . To do so, two non-zero \tilde{x}_{ikt} values from the same project in $\tilde{\mathbf{X}}$ are selected; and hours transferred between these two values so that at least one of them becomes integer. If there is only one non-zero \tilde{x}_{ikt} value for a project, it is simply rounded-up. Below, we give the description of the rounding approach and then discuss its details.

Procedure 1: Rounding approach for a relaxed efficient solution $\tilde{\mathbf{X}}$	
Step 0.	Given $\tilde{\mathbf{X}}$, set $\hat{\mathbf{X}} = \tilde{\mathbf{X}}$, $I' = I$. Go to Step 1.
Step 1.	<ul style="list-style-type: none"> i. If $I' = \emptyset$, go to Step 4. ii. Else, let $i = I'(1)$, define $U^i = \{\tilde{x}_{ikt} : \tilde{x}_{ikt} > 0\}$ such that its elements are randomly ordered, and then go to Step 2.
Step 2.	<ul style="list-style-type: none"> i. If $U^i = 1$, let $\hat{x}_{ikt} = \lceil U^i(1) \rceil$, $I' := I' \setminus \{i\}$, and go to Step 1. ii. Else, let $\tilde{x}_{ik_1t_1} = U^i(1)$ and $\tilde{x}_{ik_2t_2} = U^i(2)$, and go to Step 2.
Step 3.	Calculate $\psi_1 = (\tilde{x}_{ik_1t_1} - \lfloor \tilde{x}_{ik_1t_1} \rfloor)h_{k_1}$ and $\psi_2 = (\lfloor \tilde{x}_{ik_2t_2} \rfloor - \tilde{x}_{ik_2t_2})h_{k_2}$: <ul style="list-style-type: none"> i. If $\psi_1 \leq \psi_2$, set $\hat{x}_{ik_1t_1} = \lfloor \tilde{x}_{ik_1t_1} \rfloor$, $U^i(2) = \tilde{x}_{ik_2t_2} + \frac{\psi_1}{h_{k_2}}$, and $U^i := U^i \setminus \{U^i(1)\}$. ii. Else, set $U^i(1) = \tilde{x}_{ik_1t_1} - \frac{\psi_2}{h_{k_1}}$, $\hat{x}_{ik_2t_2} = \lfloor \tilde{x}_{ik_2t_2} \rfloor$, and $U^i := U^i \setminus \{U^i(2)\}$. Go to Step 2.
Step 4.	Stop and return $\hat{\mathbf{X}}$.

Procedure 1 applies the rounding method to each project individually.

Particularly, for $\tilde{\mathbf{X}}$, given $i \in I$, Step 1 first determines the non-zero \tilde{x}_{ikt} values. If there is only one such value for project i , it is rounded-up (see Step 2.i); otherwise, two such \tilde{x}_{ikt} values, $\tilde{x}_{ik_1t_1}$ and $\tilde{x}_{ik_2t_2}$, are randomly selected (see Step 1.ii and Step 2.ii) and Step 3 is executed. In Step 3, first $\psi_1 = (\tilde{x}_{ik_1t_1} - \lfloor \tilde{x}_{ik_1t_1} \rfloor)h_{k_1}$ and $\psi_2 = (\lfloor \tilde{x}_{ik_2t_2} \rfloor - \tilde{x}_{ik_2t_2})h_{k_2}$ are calculated. Note that ψ_1 defines the number of hours to subtract from $h_{k_1}\tilde{x}_{ik_1t_1}$ so that $(h_{k_1}\tilde{x}_{ik_1t_1} - \psi_1)/h_{k_1}$ is an integer and ψ_2 defines the number of hours to add to $h_{k_2}\tilde{x}_{ik_2t_2}$ so that $(h_{k_2}\tilde{x}_{ik_2t_2} + \psi_2)/h_{k_2}$ is an integer. After that, either $\tilde{x}_{ik_1t_1}$ is rounded-down or $\tilde{x}_{ik_2t_2}$ is rounded-up. Specifically, if $\psi_1 \leq \psi_2$, we transfer ψ_1 hours from $h_{k_1}\tilde{x}_{ik_1t_1}$ to $h_{k_2}\tilde{x}_{ik_2t_2}$ so that we have $(h_{k_1}\tilde{x}_{ik_1t_1} - \psi_1)/h_{k_1} = \lfloor \tilde{x}_{ik_1t_1} \rfloor$, i.e., it becomes an integer. Also note that, since $\psi_1 \leq \psi_2$, we will have $\tilde{x}_{ik_2t_2} + \psi_1/h_{k_2} \leq \lfloor \tilde{x}_{ik_2t_2} \rfloor$. On the other hand, if $\psi_1 > \psi_2$, we transfer ψ_1 hours from $h_{k_1}\tilde{x}_{ik_1t_1}$ to $h_{k_2}\tilde{x}_{ik_2t_2}$ so that $(h_{k_2}\tilde{x}_{ik_2t_2} + \psi_2)/h_{k_2} = \lfloor \tilde{x}_{ik_2t_2} \rfloor$, i.e., it becomes an integer. Also note that, since

$\psi_1 > \psi_2$, we will have $\tilde{x}_{ik_1t_1} - \psi_2/h_{k_1} \geq \lfloor \tilde{x}_{ik_1t_1} \rfloor$. These then imply that either $\hat{x}_{ik_1t_1} = \lfloor \tilde{x}_{ik_1t_1} \rfloor$ or $\hat{x}_{ik_2t_2} = \lfloor \tilde{x}_{ik_2t_2} \rfloor$ at the end of Step 3. Noting that $l_i \leq \tilde{x}_{ikt} \leq u_i$ when $\tilde{x}_{ikt} > 0$, we have $l_i \leq \lfloor \tilde{x}_{ikt} \rfloor \leq \lceil \tilde{x}_{ikt} \rceil \leq u_i$, which means that $l_i \leq \hat{x}_{ikt} \leq u_i$. Also, one can note that the total number of hours allocated to project i by $\tilde{\mathbf{X}}$ does not change when Step 3 is executed. Rounding for project i will be completed once Step 2.i is executed for project i and this is when the total number of hours allocated to project i changes as noted in the following remark.

Remark 2. Let $\hat{\mathbf{X}}$ be returned by Procedure 1 for a given $\tilde{\mathbf{X}}$. Then, $\hat{\mathbf{X}}$ is integer such that $\sum_{k \in K} \sum_{t \in T} h_k \tilde{x}_{ikt} \leq \sum_{k \in K} \sum_{t \in T} h_k \hat{x}_{ikt} < \sum_{k \in K} \sum_{t \in T} h_k \tilde{x}_{ikt} + \max_{k \in K} \{h_k\} \forall i \in I$.

Proof. First, note that U^i defined in Step 1.ii will never be \emptyset because $H_i > 0 \forall i \in I$; therefore, Step 2.i is executed exactly once in Procedure 1 for project i . Furthermore, Step 3 is executed until $|U^i| = 1$; and total number of hours allocated to project i does not change with an execution of Step 3. That is, total number of hours allocated to project i changes only when Step 1.ii is applied on a single \tilde{x}_{ikt} value and no modification takes place for project i after Step 1.ii is executed. Furthermore, since $h_k(\lceil \tilde{x}_{ikt} \rceil - \tilde{x}_{ikt}) < \max_{k \in K} \{h_k\}$ for any \tilde{x}_{ikt} , it then follows that $\sum_{k \in K} \sum_{t \in T} h_k \hat{x}_{ikt} < \sum_{k \in K} \sum_{t \in T} h_k \tilde{x}_{ikt} + \max_{k \in K} \{h_k\} \forall i \in I$. Finally, since $h_k \tilde{x}_{ikt} \leq h_k \lfloor \tilde{x}_{ikt} \rfloor$, we have $\sum_{k \in K} \sum_{t \in T} h_k \tilde{x}_{ikt} \leq \sum_{k \in K} \sum_{t \in T} h_k \hat{x}_{ikt}$. ■

Remark 2 suggests that the total number of hours allocated to a project under a relaxed efficient solution increases when Procedure 1 is applied, which means that $TC(\tilde{\mathbf{X}})$ is also likely to increase. However, this increase is expected to be relatively less, especially when $H_i \gg h_k$. Also, $MW(\tilde{\mathbf{X}})$ will change because of the rounding operations.

Therefore, once the relaxed efficient solution $\tilde{\mathbf{X}}$ is converted to an integer solution $\hat{\mathbf{X}}$, we try to improve this rounded solution $\hat{\mathbf{X}}$ as detailed next.

(iii) Improving the rounded solutions: Suppose that $\hat{\mathbf{X}}$ is given such that \hat{x}_{ikt} is integer $\forall i \in I, \forall k \in K, \forall t \in T$. One can then calculate \hat{X}^{max} for $\hat{\mathbf{X}}$. Here, we present a simple procedure, Procedure 2, which aims at decreasing \hat{X}^{max} . To reduce \hat{X}^{max} , we focus on the periods where the total number of workers utilized is equal to \hat{X}^{max} . As it is possible that there are multiple periods with \hat{X}^{max} workers, one should reduce the number of workers utilized in each of such periods. To reduce the number of workers in one such period, we attempt to change the mode of the workers allocated to a project in that period in the cost-minimum way possible. Below, we give the description of Procedure 2 and then discuss its details.

In Procedure 2, Step 0 first determines the periods with the maximum number of workers and randomly orders them. Then, one of these periods (period t) is selected and the projects with some workers allocated in the selected period (projects in I^t) are determined in Step 1.i. After that, one of these projects (i) is (projects in I^t) are determined in Step 1.ii. After that, one of these projects (i) is randomly selected and we determine the possible feasible mode changes that reduce the number of workers assigned to project i in period t (i.e., set K^{ti}) in Step 2.ii. Specifically, a mode change from mode k' to k is feasible and reduces the number of workers assigned to project i in period t (i.e., $\bar{x}_{ik't}$) as long as $l_i \leq [\bar{x}_{ik't} h_{k'}/h_k] < \bar{x}_{ik't}$ (note that, we already have $\bar{x}_{ik't} \leq u_i$ because $\hat{\mathbf{X}}$ is feasible; therefore, $[\bar{x}_{ik't} h_{k'}/h_k] < u_i$). If there is not any feasible mode change that can reduce $\bar{x}_{ik't}$, then we try another project (see Step 3.i) if there is one to

Procedure 2: Reducing \hat{X}^{max} of a rounded solution \hat{X}	
Step 0.	Given \hat{X} , set $\bar{X} = \hat{X}$ and determine $T' = \{t: \sum_{i \in I} \sum_{k \in K} \bar{x}_{ikt} = \hat{X}^{max}\}$ such that its elements are randomly ordered. Go to Step 1.
Step 1.	<ul style="list-style-type: none"> i. If $T' = \emptyset$, go to Step 4. ii. Else, let $t = T'(1)$, define $I^t = \{i: \sum_{k \in K} \bar{x}_{ikt} > 0\}$ such that its elements are randomly ordered, and then go to Step 2.
Step 2.	<ul style="list-style-type: none"> i. If $I^t = \emptyset$, go to Step 4. ii. Else, let $i = I^t(1)$ and $k' = \{k: \bar{x}_{ik't} > 0\}$, define $K^{ti} = \left\{k: l_i \leq \left\lceil \frac{\bar{x}_{ik't} h_{k'}}{h_k} \right\rceil < \bar{x}_{ik't}\right\}$, and go to Step 3.
Step 3.	<ul style="list-style-type: none"> i. If $K^{ti} = \emptyset$, set $I^t := I^t \setminus \{I^t(1)\}$ and go to Step 2. ii. Else, let $k'' = \arg \min_{k \in K^{ti}} \left\{ \left\lceil \frac{\bar{x}_{ik't} h_{k'}}{h_k} \right\rceil c_k \right\}$, set $\bar{x}_{ik't} := 0$, $\bar{x}_{ik''t} := \left\lceil \frac{\bar{x}_{ik't} h_{k'}}{h_{k''}} \right\rceil$, and $T' := T' \setminus \{T'(1)\}$, and go to Step 1.
Step 4.	Stop and return \bar{X} .

try; and if there is not any other project that can be used to reduce the total number of workers in period t , it means that \hat{X}^{max} could not be reduced and we terminate our attempt in Step 2.i. On the other hand, if there is at least one such feasible mode change, then we select the one which has the minimum cost implication as noted in Step 3.ii; and update the worker assignments for project i in period t . And in this case, we are able to reduce the total number of workers in period t , therefore, we repeat the process for another period, if any remains, that has the maximum number of workers (i.e., we go back to Step 1 after Step 3.ii). It is worthwhile to note that, given \hat{X} , the solution returned by Procedure 2, i.e., \bar{X} guarantees that $\bar{X}^{max} \leq \hat{X}^{max}$; therefore, we have $MW(\bar{X}) \leq MW(\hat{X})$. On the other hand, it is both possible that $TC(\bar{X}) < TC(\hat{X})$ and $TC(\bar{X}) \geq TC(\hat{X})$. That is, it is possible that one solution Pareto-dominates the other. Therefore, in phase (iv), we assure that we compare \bar{X} and \hat{X} for Pareto dominance.

(iv) Determining Pareto efficient improved rounded solutions: At the end of Step 3 of Algorithm 2, we have a set of integer solutions generated from relaxed efficient

solutions $\tilde{\mathbf{X}} \in \tilde{PE}$. Particularly, for each $\tilde{\mathbf{X}} \in \tilde{PE}$, we generate $\hat{\mathbf{X}}$ and $\bar{\mathbf{X}}$ N times; therefore, the size of the set of integer solutions generated at the end of phase (iii) is at most $2N$, i.e. $\hat{P} \leq 2N$. In this phase, we determine the set of Pareto efficient solutions among these integer solutions generated. To do so, Procedure 3, which is detailed below, is used.

We note that procedures similar to Procedure 3 exist in literature (see, e.g., Schaefer and Konur (2015), Konur and Schaefer (2016), Konur et al. (2017)). Next section presents the results of our numerical studies.

Procedure 3: Determining Pareto efficient solutions within a given set of solutions \hat{P}	
Step 0.	Given \hat{P} , update \hat{P} such that it has unique solutions and then sort the elements in \hat{P} such that $TC(\hat{P}(\ell)) \leq TC(\hat{P}(\ell + 1))$ and $MW(\hat{P}(\ell)) \leq MW(\hat{P}(\ell + 1))$ when $TC(\hat{P}(\ell)) = TC(\hat{P}(\ell + 1))$ for $1 \leq \ell < \hat{P} $. Go to Step 1.
Step 1.	Set $PF(\hat{P}) = [TC(\hat{P}(1)), MW(\hat{P}(1))]$, $PE(\hat{P}) = \hat{P}(1)$, and go to Step 2.
Step 2.	For $\ell = 2: \hat{P} $ If $MW(\hat{P}(\ell)) < \min_{1 \leq L \leq \ell-1} \{MW(\hat{P}(L))\}$, $PE(\hat{P}) := PE(\hat{P}) \cup \{\hat{P}(\ell)\}$ and $PF(\hat{P}) := PF(\hat{P}) \cup \{[TC(\hat{P}(\ell)), MW(\hat{P}(\ell))]\}$. End Go to Step 3.
Step 3.	Stop and return $PE(\hat{P})$ and $PF(\hat{P})$.

4. NUMERICAL STUDIES

In Section 3, we presented two solution methods for **CSP**: Algorithm 1 is the implementation of the classical ε -constraint method and it generates the exact Pareto front PF and Algorithm 2 is a heuristic method based on rounding and improving

solutions from Algorithm 1 when partial linear relaxations of the subproblems are solved and it generates an approximated Pareto front \widehat{PF} . This section quantitatively and qualitatively compares Algorithms 1 and 2. Prior to the comparison results, we first discuss the problem instance generation process and the computational settings for solving the problem instances generated.

4.1. PROBLEM INSTANCES AND COMPUTATIONAL SETTING

To the best of our knowledge, there is no data set for the problem under investigation in this study; therefore, we generate new problem instances. In the generation process, we take similar approaches with Coughlan et al. (2015; 2010) as detailed below.

We consider 10 problem sets, where each problem set is defined by $|I|$ such that $|I| \in \{10, 12, 14, 16, 18, 20, 25, 30, 35, 40\}$. For each problem set, we randomly generate 10 feasible problem instances. Each problem instance has $l_i = 2$ (minimum number of workers that should simultaneously work on a project in a week) and $u_i = 10$ (maximum number of workers that can simultaneously work on a project in a week) $\forall i \in I$. We consider two cases for each instance: 2-mode and 3-mode cases. In 2-mode case, $|K| = 2$ such that $[h_1, h_2] = [40, 50]$ and $[c_1, c_2] = [70, 75]$. In 3-mode case, $|K| = 3$ such that $[h_1, h_2, h_3] = [40, 50, 60]$ and $[c_1, c_2, c_3] = [70, 75, 80]$. These numbers are parallel with the practical settings we observed for the semiconductor manufacturing construction problem. Indeed, one can also note that $h_k > h_{k+1}$ and $c_k > c_{k+1}$, i.e., the hourly rate increases as the number of hours worked in a week increases, which is true for many practical settings considering overtime hours.

A problem instance is defined by its $|T|$ (length of the planning horizon), A (set of precedence relations), e_i and d_i values (earliest start time and due dates for projects), and H_i values (construction hours required for projects). Specifically, we first generate H_i values randomly such that $H_i \sim 50 \times U[20,40]$, where $U[\lambda_l, \lambda_u]$ denotes a discrete uniform distribution between λ_l and λ_u . That is, H_i values are randomly generated as multiples of 50 between 1,000 and 2,000 hours. Then, we generate A and T as follows. First, we define so-called project durations such that project i 's duration, D_i , is defined as $\lceil H_i / (50 \times 6) \rceil$, where $\lceil \rho \rceil$ rounds ρ to the nearest integer. Here, 6 is the average number of workers (i.e., average of the minimum, 2, and maximum, 10, number of workers on a project) and 50 is the average number of hours by a worker in 3-mode case. Therefore, D_i defines how many weeks it would take to complete project i when 6 workers assigned each week such that each worker works 50 hours. After that, chains of projects with varying lengths is generated by randomly generating chain lengths, denoted by B , such that the sum of the chain lengths is equal to $|I|$. Table 2 gives the chain lengths considered for each problem set. A chain of length $B = \beta$ has β projects such that a project precedes the next and, without loss of generality, we have project i preceding project $i + 1$, which is denoted by $i \rightarrow i + 1$. For instance, for a problem instance with $|I| = 10$, chain lengths of 3, 4, and 3 define the following chains: $1 \rightarrow 2 \rightarrow 3$, $4 \rightarrow 5 \rightarrow 6 \rightarrow 7$, and $8 \rightarrow 9 \rightarrow 10$.

After the chains are created, we calculate the chain duration as the sum of the durations of the activities in the chain; and then, we set T equal to the average of the chain durations. The chains created readily define a set of precedence relations. We randomly generate additional precedence relations $i \rightarrow j$ such that $i < j$, i is not the last

activity in a chain, and j is not the first activity in the other chain. This guarantees that there does not exist any circles in the precedence network. This process is repeated at most 100 times to generate precedence relations in addition to the ones already created within the chains so that the total number of precedence relations is between $|I|$ and $2|I|$. Table 2 further gives the average $|T|$ and $|A|$ values over the 10 problem instances generated within each problem set.

Following the creation of the precedence network, we determine e_i and d_i values for the projects as follows. We first add a dummy project, project 0, with 0 duration that precedes all of the first projects in the initial chains created. The arcs representing the

Table 2. Chain lengths and averages of T and $|A|$ for problem sets with varying $|I|$.

$ I $	10	12	14	16	18	20	25	30	35	40
B~	$U[2,4]$	$U[2,4]$	$U[3,5]$	$U[3,5]$	$U[4,6]$	$U[4,6]$	$U[5,7]$	$U[6,8]$	$U[7,9]$	$U[8,10]$
Avg. $ T $	14.2	14.7	21.8	20.4	22.6	23.8	31.6	37.8	44.4	45.9
Avg. $ A $	14.1	16.9	19.3	22.9	23.7	24.9	30.7	36.9	38.4	43.9

precedence relations have processor project's duration as its length; that is, arc $i \rightarrow j$ has a length of D_i . To assign e_i , we find the shortest path from project 0 to project i : if this length is less than T , we set it as e_i ; otherwise, we subtract D_i from the shortest path until e_i is less than $|T|$. The first projects in the initial chains created will have $e_i = 0$; and therefore, we set their $e_i = 1$. To assign d_i , we find the longest path from project 0 to project i and add D_i to the longest path: if the longest path plus D_i is less than $|T|$, we set it as d_i ; otherwise, we set $d_i = |T|$. Finally, we check if $e_i \geq d_i$; if not, we either increase d_i or decrease e_i by 1 until $e_i \geq d_i$.

Once T , A , and H_i , e_i , and d_i values are generated as discussed above, we check the feasibility of the corresponding instance by solving the following schedule feasibility problem, **SFP**, such that there is only one mode $K = \{2\}$ with $h_2 = 50$ (a common mode for 2-mode and 3-mode cases):

SFP:

$$\text{Minimize} \quad \sum_{i \in I} \sum_{t \in T} x_{ikt}$$

$$\text{subject to} \quad (1) - (6), (9) - (10), (16) - (17)$$

$$\sum_{k \in K} x_{ikt} \leq u_i \sum_{r=1}^t s_{ir} \quad \forall i \in I, \forall t \in T \quad (19)$$

$$\sum_{k \in K} x_{ikt} \leq u_i \sum_{r=t}^{|T|} f_{ir} \quad \forall i \in I, \forall t \in T \quad (20)$$

$$x_{ikt} \geq 0 \quad \forall i \in I, \forall k \in K, \forall t \in T. \quad (21)$$

In **SFP**, the objective is minimization of the total number of workers assigned, which is an arbitrary objective function. Recall that constraints (1)-(6) are the scheduling and precedence restrictions, constraints (9)-(10) assure at least one worker in start and finish weeks of a project, and constraints (16)-(17) are binary definitions of project-start and -finish times variables, i.e., \mathbf{S} and \mathbf{F} . As there is only one mode, z_{ikt} variables are not used in **SFP**. Therefore, instead of constraints (7), (8), and (12), we use constraints (19) and (20), which assure that at most u_i workers are used on a project between its start and finish periods. On the other hand, **SFP** ignores $x_{ikt} \geq l_i$ constraints (which would require z_{ikt} type of variables). Finally, as noted in (21), the worker assignments variables are continuous in **SFP**.

Remark 3. Given a problem instance, if **SFP** is feasible, then **CSP** is feasible for both 2-mode and 3-mode cases.

Proof. Let x_{i2t}^* , s_{it}^* , and f_{it}^* be the optimum solution of **SFP** for a given problem instance. Now, consider the following solution. Let $\hat{s}_{it} = s_{it}^*$ and $\hat{f}_{it} = f_{it}^* \forall i \in I \forall t \in T$; $\hat{x}_{i1t} = 0$ and $\hat{x}_{i2t} = \max\{[x_{i2t}^*], l_i\}$ when $x_{i2t}^* > 0$ and $\hat{x}_{i2t} = 0$ when $x_{i2t}^* = 0 \forall i \in I \forall t \in T$; and $\hat{z}_{i1t} = 0$ and $\hat{z}_{i2t} = \min\{1, \hat{x}_{i2t}\} \forall i \in I, \forall t \in T$. Since s_{it}^* and f_{it}^* values are optimal for **SFP**, it then follows that \hat{s}_{it} and \hat{f}_{it} satisfy constraints (2)-(6). Furthermore, since $x_{i2t}^* \leq u_i$, we have $\max\{[x_{i2t}^*], l_i\} \leq u_i$ given that $l_i \leq u_i$; and therefore, either $l_i \leq \hat{x}_{i2t} \leq u_i$ or $\hat{x}_{i2t} = 0$. Along with definition of \hat{z}_{ikt} and x_{i2t}^* , s_{it}^* , and f_{it}^* satisfying (19) and (20), it follows that \hat{z}_{ikt} and \hat{x}_{ikt} values satisfy constraints (7)-(12). Noting that $\sum_{k \in K} \sum_{t \in T} h_k \hat{x}_{ikt} \geq \sum_{t \in T} h_2 x_{i2t}^* \geq H_i$ by definition, \hat{x}_{ikt} values satisfy constraints (1). Finally, by definition, \hat{x}_{ikt} values are integer and \hat{z}_{ikt} values are binary. It then follows that $\langle \hat{X}, \hat{Z}, \hat{S}, \hat{F} \rangle$ is feasible for **CSP** under 2-mode case. Similarly, one can construct a solution that is feasible for **CSP** under 3-mode case (in addition to the above construction, one just needs to define $\hat{x}_{i3t} = 0$ and $\hat{z}_{i3t} = 0 \forall i \in I \forall t \in T$). ■

For each problem set, we generate instances and solve **SFP** until 10 problem instances, which are feasible for both 2- and 3-mode cases, are generated. The data for the problem instances are available at <http://dx.doi.org/10.17632/ngh6cvyfr7.1>. In all of the feasible problem instances generated, we have $|A| \geq |I|$, $\min_{i \in I} \{e_i\} = 1$, and $\max_{i \in I} \{d_i\} = T$

Feasibility of a problem instance implies that $|PF| \geq 1$ and $|\widehat{PF}| \geq 1$. We solve each problem instance with 2-mode and 3-mode cases using both Algorithms 1 and 2. The 2-mode and 3-mode solutions for each problem instance are posted at <http://dx.doi.org/10.17632/ngh6cvyfr7.1>. In Algorithm 2, we set $N = |I|$. Both

Algorithms 1 and 2 are coded in Matlab 2019a. We use Gurobi 9.0.1 for solving subproblems **SP** and **RSP**. Time limit is set to 1,800 seconds for solving any subproblem. All problem instances are solved on Inter Core i5-7600 at 3.5 GHz with 4 cores and 16 GB of RAM under 64-bit Windows 10 operating system.

4.2. COMPARISON OF THE SOLUTION METHODS

In this section, we compare Algorithms 1 and 2 for **CSP** under 2- and 3-mode cases. Our comparison is two-fold: (i) quantitative comparison and (ii) qualitative comparison. The details follow below.

(i) Quantitative comparison: Quantitative comparison focuses on computational time and the number of solutions returned by each algorithm. The computational times of Algorithms 1 and 2 are denoted by cpu_1 and cpu_2 , respectively, and given in terms of seconds. Table 3 gives the averages over 10 problem instances solved within each problem set (i.e., $|I|$) under both 2- and 3-mode cases for the number of solutions returned ($|PF|$ and $|\widehat{PF}|$), percentage of problem instances when one algorithm returned more solutions than the other ($\% |PF| > |\widehat{PF}|$ and $\% |PF| < |\widehat{PF}|$) and computational times (cpu_1 and cpu_2) under each algorithm used for solving **CSP**. The last row is the average of the averages, i.e., average of these statistics over the 100 problem instances. We have the following observations.

Based on Table 3, we can conclude that Algorithm 2 is significantly more efficient than Algorithm 1 for solving **CSP** in terms of computational time for all problem sets. The overall average times for Algorithms 1 and 2 are around 270 and 12 seconds for 2-mode case, respectively (Algorithm 2 is more than 20 times faster than

Algorithm 1); and, 968 and 56 seconds for 3-mode case, respectively (Algorithm 2 is almost 20 times faster than Algorithm 1). Indeed, in all of the problem instances solved

Table 3. Quantitative comparison of Algorithms 1 and 2 for **CSP**.

$ I $	2-mode Average Results						3-mode Average Results					
	$ PF $	cpu_1	$ \widehat{PF} $	cpu_2	$\% PF > \widehat{PF} $	$\% PF < \widehat{PF} $	$ PF $	cpu_1	$ \widehat{PF} $	cpu_2	$\% PF > \widehat{PF} $	$\% PF < \widehat{PF} $
10	7.3	9.2	6.9	0.8	40%	40%	10.9	376.9	9.9	2.1	60%	10%
12	6.8	8.3	6.8	0.9	20%	20%	12.0	28.4	11.9	3.0	20%	20%
14	7.2	206.1	6.8	2.2	40%	20%	10.5	327.1	9.5	7.4	70%	0%
16	6.9	14.7	7.2	2.3	20%	50%	10.8	49.3	10.4	8.7	30%	20%
18	6.9	20.6	6.9	3.4	10%	20%	10.5	418.6	10.4	14.9	30%	30%
20	7.3	203.8	7.4	3.3	20%	30%	11.2	790.4	11.7	11.3	10%	60%
25	7.6	414.3	7.5	6.7	30%	30%	11.7	1271.4	11.5	29.9	20%	10%
30	7.6	546.2	7.6	26.4	30%	30%	12.4	1816.7	11.8	134.1	50%	10%
35	7.5	586.8	7.5	46.2	10%	10%	11.6	2407.5	11.9	219.6	20%	30%
40	8.0	692.1	8.3	29.7	0%	30%	12.4	2193.6	12.8	126.1	20%	40%
Avg.	7.3	270.2	7.3	12.2	22%	28%	11.4	968.0	11.2	55.7	33%	23%

under both 2- and 3-mode cases, Algorithm 2 was faster. Furthermore, in terms of the number of solutions returned, Algorithms 1 and 2 are very close for all problem sets and overall average: overall averages are both 7.3 under 2-mode case and 11.4 vs. 11.2 under 3-mode cases, with Algorithm 1 returning slightly more solutions under both cases on overall average. Specifically, one can note that, both algorithms returned the same number of solutions for 50% and 44% of all problem instances solved under 2- and 3-mode cases, respectively; and, the percentages of the number of instances when Algorithm 1 returned more solutions (22% and 33% under 2- and 3-mode cases, respectively) are relatively close to the percentages of the number of instances when Algorithm 2 returned more solutions (%25 and 23% under 2- and 3-mode cases, respectively). Therefore, we conclude that, Algorithm 2 returns similar number of solutions with significantly less computation time. In particular, Algorithm 1 takes

around 37 seconds and 85 seconds per solution under 2- and 3-mode cases, respectively, whereas, Algorithm 2 takes around 1.7 seconds and 5 seconds per solution under 2- and 3-mode cases, respectively.

Figures 1 and 2 demonstrate PF and \widehat{PF} for two different problem instances under both 2- and 3-mode cases. In these figures, two extreme points (denoted by EP^1 and EP^2 for PF and \widehat{EP}^1 and \widehat{EP}^2 for \widehat{PF}) and the density points (denoted by DP and \widehat{DP} for PF and \widehat{PF} , respectively) of PF and \widehat{PF} are illustrated, which are used in our qualitative comparison and detailed below. We also posted these figures for all of the problem instances at <http://dx.doi.org/10.17632/ngb6cvyfr7.1> for the interested reader. It can be seen from the figures that PF and \widehat{PF} are parallel and close to each other. In what follows, we systematically compare PF and \widehat{PF} ; and, given that PF is the exact Pareto front, our aim is to assess the quality of \widehat{PF} .

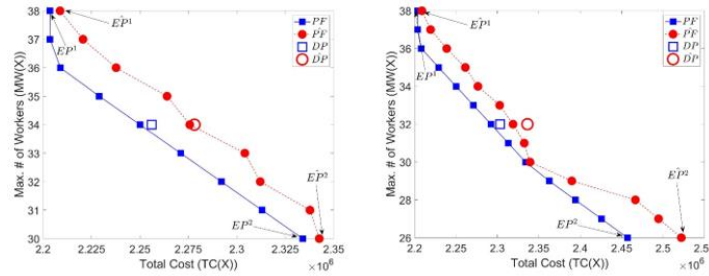


Figure 1. Illustration of the average extreme and density points for Instance 10 of $|I| = 20$.

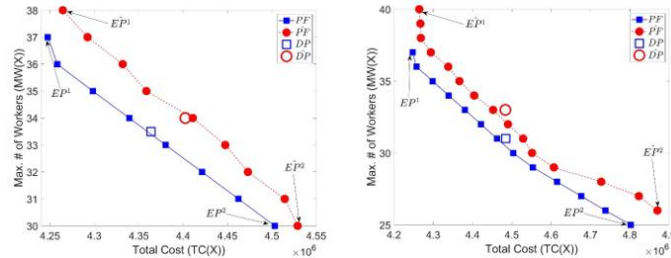


Figure 2. Illustration of the average extreme and density points for Instance 10 of $|I| = 40$.

(ii) Qualitative comparison: Qualitative comparison compares PF and \widehat{PF} using several qualitative measures. In what follows, we discuss how PF and \widehat{PF} compare based on each measure.

The first set of measures we use include the extreme and the density points of the Pareto fronts. Each Pareto front has two extreme points: cost-minimum, denoted by EP^1 and \widehat{EP}^1 for PF and \widehat{PF} , respectively, and maximum workers-minimum, denoted by EP^2 and \widehat{EP}^2 for PF and \widehat{PF} , respectively. One can note that $EP^1 =$

$$\left[\min_{X \in PE} \{TC(X)\}, \max_{X \in PE} \{MW(X)\} \right] \text{ and } EP^2 = \left[\max_{X \in PE} \{TC(X)\}, \min_{X \in PE} \{MW(X)\} \right]; \text{ and } \widehat{EP}^1 =$$

$$\left[\min_{X \in \widehat{PE}} \{TC(X)\}, \max_{X \in \widehat{PE}} \{MW(X)\} \right] \text{ and } \widehat{EP}^2 = \left[\max_{X \in \widehat{PE}} \{TC(X)\}, \min_{X \in \widehat{PE}} \{MW(X)\} \right].$$

A density point, denoted by DP and \widehat{DP} for PF and \widehat{PF} , respectively, defines the averages of

$TC(X)$ and $MW(X)$ over the solutions within a set of solutions. That is, $DP =$

$$\left[\frac{1}{|PF|} \sum_{X \in PE} TC(X), \frac{1}{|PF|} \sum_{X \in PE} MW(X) \right] \text{ and } \widehat{DP} =$$

$$\left[\frac{1}{|\widehat{PF}|} \sum_{X \in \widehat{PE}} TC(X), \frac{1}{|\widehat{PF}|} \sum_{X \in \widehat{PE}} MW(X) \right].$$

Tables 4 and 5 summarize the averages of the extreme and density points (EP^1 , EP^2 , DP and \widehat{EP}^1 , \widehat{EP}^2 , \widehat{DP}) over the 10 problem instances solved within each problem set under 2- and 3-mode cases, respectively, and document the overall overages of these density points. Figures 3 illustrates EP^1 vs. \widehat{EP}^1 (see Figures 3.a and 3.b), DP vs. \widehat{DP} (see Figures 3.c and 3.d), and EP^2 vs. \widehat{EP}^2 (see Figures 3.e and 3.f) for 2- and 3-mode cases.

We have the following observations from Tables 4 and 5, and Figure 3.

- Based on cost-minimum extreme points (i.e., EP^1 vs. \widehat{EP}^1), we can see that the cost-minimum solutions returned by Algorithm 2 are close to the cost-minimum solutions returned by Algorithm 1 in terms of not only the total cost but also the maximum number of workers utilized. Particularly, overall average of $TC(\mathbf{X})$ and $MW(\mathbf{X})$ of the cost-minimum solutions returned by Algorithms 1 and 2 are 2342640 vs. 2349958 and 31.6 vs. 32.3 for 2-mode cases; and, 2342560 vs. 2350058 and 31.6 vs. 32.2 for 3-mode cases. The average increases in $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values of \widehat{EP}^1 compared to EP^1 are around 0.3% (0.30% on average for 2-mode cases, 0.32% on average for 3-mode cases, the maximum was %0.64 under a 3-mode case) and %2 (2.03% on average for 2-mode cases and 1.97% on average for 3-mode cases). We note that, while $TC(\mathbf{X})$ of EP^1 is guaranteed to be less than or equal to $TC(\mathbf{X})$ of \widehat{EP}^1 , that is not necessarily the case for $MW(\mathbf{X})$ values. Indeed, even though, $MW(\mathbf{X})$ of EP^1 tends to be and on average is less than $MW(\mathbf{X})$ of \widehat{EP}^1 , in 28% and 30% of problem instances under 2-mode and 3-mode cases, respectively, $MW(\mathbf{X})$ of \widehat{EP}^1 was slightly less than or equal to the $MW(\mathbf{X})$ of EP^1 .

Table 4. 2-mode results.

	Average Points for Algorithm 1			Average Points for Algorithm 2		
	EP^1	DP	EP^2	\widehat{EP}^1	\widehat{DP}	\widehat{EP}^2
10	[1096895, 29.6]	[1119285, 26.0]	[1155300, 22.7]	[1100650, 29.8]	[1121977, 26.5]	[1152990, 23.4]
12	[1255595, 33.2]	[1277171, 30.3]	[1308330, 27.4]	[1259160, 33.7]	[1283445, 30.8]	[1316450, 27.8]
14	[1522580, 27.9]	[1549352, 24.4]	[1594545, 21.2]	[1526140, 27.7]	[1554366, 24.6]	[1594915, 21.6]
16	[1728695, 31.0]	[1763600, 27.8]	[1813020, 24.8]	[1732980, 31.9]	[1770227, 28.4]	[1814920, 25.2]
18	[1877535, 30.5]	[1915748, 27.4]	[1970500, 24.4]	[1882650, 31.1]	[1921909, 28.1]	[1976120, 25.0]
20	[2115245, 32.3]	[2162146, 29.2]	[2228425, 26.0]	[2122665, 33.2]	[2171253, 29.9]	[2226825, 26.5]
25	[2716610, 32.4]	[2770209, 29.1]	[2854740, 25.8]	[2724690, 33.1]	[2780332, 29.9]	[2857995, 26.6]
30	[3149305, 32.7]	[3204275, 29.4]	[3284575, 26.1]	[3158985, 33.4]	[3217769, 30.1]	[3294500, 26.8]
35	[3757640, 32.3]	[3833437, 29.0]	[3940630, 25.7]	[3769655, 33.1]	[3845148, 29.9]	[3949515, 26.6]
40	[4206300, 34.6]	[4296445, 31.0]	[4422865, 27.5]	[4222005, 35.5]	[4318770, 31.9]	[4437610, 28.2]
Avg.	[2342640, 31.6]	[2389167, 28.3]	[2457293, 25.2]	[2349958, 32.3]	[2398520, 29.0]	[2462184, 25.8]

Table 5. 3-mode results.

$ I $	Average Points for Algorithm 1			Average Points for Algorithm 2		
	EP^1	DP	EP^2	\widehat{EP}^1	\widehat{DP}	\widehat{EP}^2
10	[1096895, 29.6]	[1146072, 24.2]	[1227255, 19.1]	[1100595, 29.5]	[1151967, 24.6]	[1239705, 20.0]
12	[1255595, 32.9]	[1307928, 27.4]	[1390190, 21.9]	[1259720, 33.5]	[1318411, 27.9]	[1418970, 22.4]
14	[1522580, 27.9]	[1581956, 22.7]	[1686185, 17.9]	[1526515, 27.5]	[1590534, 23.1]	[1713910, 18.7]
16	[1728490, 31.1]	[1807581, 25.8]	[1927290, 20.8]	[1733635, 31.5]	[1814398, 26.5]	[1954320, 21.8]
18	[1877230, 30.5]	[1956228, 25.6]	[2077300, 20.8]	[1882540, 31.1]	[1964034, 26.4]	[2111555, 21.6]
20	[2114945, 32.3]	[2213119, 27.2]	[2359200, 22.1]	[2122375, 33.4]	[2230908, 28.0]	[2417400, 22.6]
25	[2716610, 32.4]	[2834489, 27.1]	[3023910, 21.7]	[2724875, 33.0]	[2852952, 27.8]	[3077715, 22.5]
30	[3149305, 32.7]	[3278119, 27.0]	[3488995, 21.3]	[3158705, 33.4]	[3288326, 28.0]	[3528570, 22.5]
35	[3757645, 32.2]	[3915766, 26.9]	[4163565, 21.6]	[3769665, 33.4]	[3932128, 27.9]	[4231660, 22.4]
40	[4206305, 34.4]	[4400226, 28.7]	[4698790, 23.0]	[4221950, 35.7]	[4424250, 29.8]	[4777905, 23.9]
Avg.	[2342560, 31.6]	[2444148, 26.2]	[2604268, 21.0]	[2350058, 32.2]	[2456791, 27.0]	[2647171, 21.8]

- Based on density points (i.e., DP vs. \widehat{DP}), we can see that the averages of the solutions within PF and \widehat{PF} are close in terms of both $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values. Particularly, average of $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values over all of the solutions returned by Algorithms 1 and 2 are 2389167 vs. 2398520 and 28.3 vs. 29.0 for 2-mode cases; and, 2444148 vs. 2456791 and 26.2 vs. 27.0 for 3-mode cases. The average increases in $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values of \widehat{DP} compared to DP are less than 0.6% (0.38% on average for 2-mode cases, 0.54% on average for 3-mode cases, and the maximum was 1.8% for a 3-mode case) and less than 3% (2.31% on average for 2-mode cases and 2.88% on average for 3-mode cases).
- Based on maximum workers-minimum extreme points (i.e., EP^2 vs. \widehat{EP}^2), we can see that the solutions with minimum X^{max} values returned by Algorithms 1 and 2 are close in terms of both total cost and X^{max} values. Particularly, overall average of $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values of EP^2 vs. \widehat{EP}^2 points are 2457293 vs. 2462184 and 25.2 vs. 25.8 for 2-mode cases; and, 2604268 vs. 2647171 and 21.0 vs. 21.8

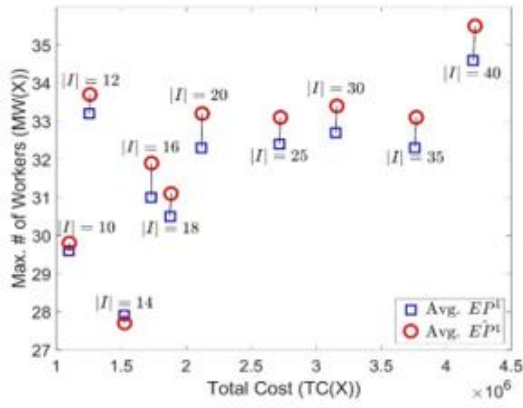
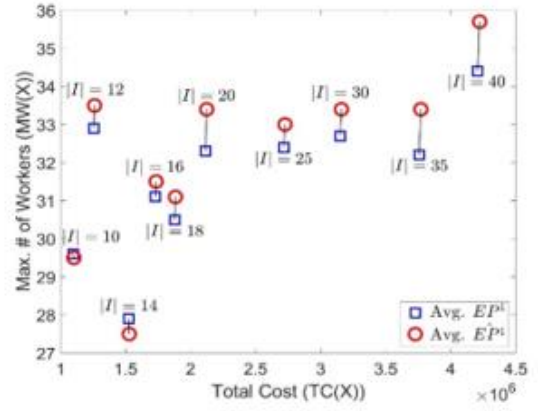
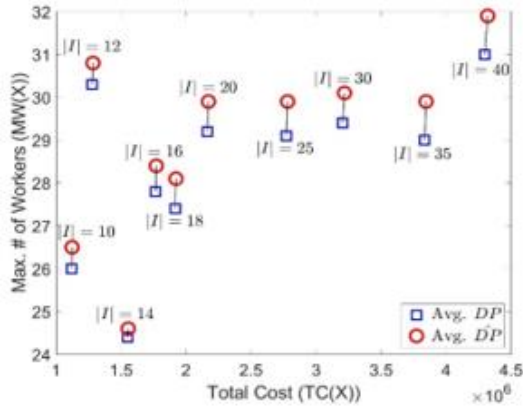
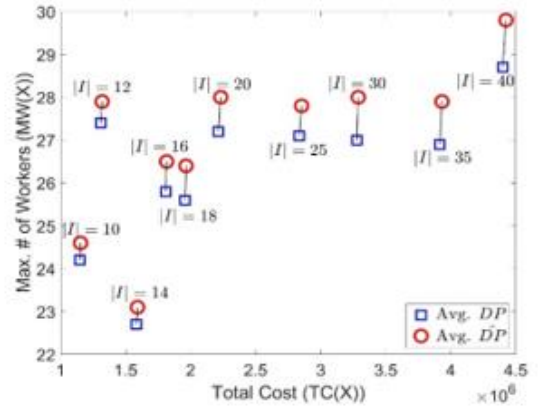
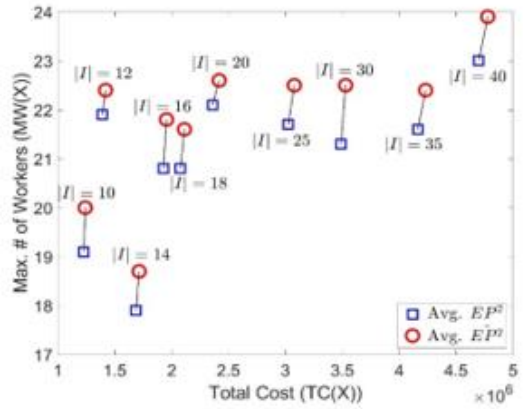
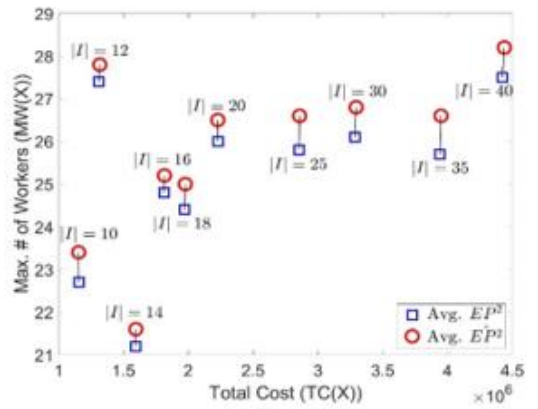
a) EP^1 vs. \widehat{EP}^1 under 2-mode casesb) EP^1 vs. \widehat{EP}^1 under 3-mode casesc) DP vs. \widehat{DP} under 2-mode casesd) DP vs. \widehat{DP} under 3-mode casese) EP^2 vs. \widehat{EP}^2 under 2-mode casesf) EP^2 vs. \widehat{EP}^2 under 3-mode cases

Figure 3. Illustration of average extreme and density points.

for 3-mode cases. Note that, by definition, $MW(\mathbf{X})$ of EP^2 is less than or equal to $MW(\mathbf{X})$ of \widehat{EP}^2 ; whereas, $TC(\mathbf{X})$ of EP^2 can be less than or greater than or equal to $TC(\mathbf{X})$ of \widehat{EP}^2 . For 2-mode cases, the average increases in $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values of \widehat{EP}^2 compared to EP^2 are 0.18% and 2.53%, respectively, and the maximum increase in $MW(\mathbf{X})$ value was 8%. Furthermore, for 40% of the instances under 2-mode case, Algorithm 2 was able to determine the minimum X^{max} value (i.e., EP^2 vs. \widehat{EP}^2 had the same $MW(\mathbf{X})$ values); and, even though $TC(\mathbf{X})$ of EP^2 tends to be and on average is less than $TC(\mathbf{X})$ of \widehat{EP}^2 , in 35% of problem instances under 2-mode cases, $TC(\mathbf{X})$ of \widehat{EP}^2 was slightly less than or equal to the $TC(\mathbf{X})$ of EP^2 . For 3-mode cases, the average increases in $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values of \widehat{EP}^2 compared to EP^2 are 1.66% and 4.05%, respectively, and the maximum increase in $MW(\mathbf{X})$ value was 10%. Furthermore, under 3-mode case, Algorithm 2 was able to determine the minimum X^{max} value for 25% of the instances $TC(\mathbf{X})$ of \widehat{EP}^2 was slightly less than or equal to the $TC(\mathbf{X})$ of EP^2 for 6% of the instances.

Our comparison of PF and \widehat{PF} based on extreme and density points can be summarized as follows. On average, Algorithm 2 is able to determine extreme points that are close to the actual extreme points in terms of both $TC(\mathbf{X})$ and $MW(\mathbf{X})$ values. Furthermore, the density points of PF and \widehat{PF} are close as well. Recalling that both algorithms return similar number of solutions, we can say that Algorithm 2 finds a close point for each actual non-dominated point on average.

The second set of measures compare PF and \widehat{PF} over the objective space of $PF \cup \widehat{PF}$. For notational simplicity, let us define where $TC^{max} = \max_{X \in PE \cup \widehat{PE}} \{TC(X)\}$, $TC^{min} = \min_{X \in PE \cup \widehat{PE}} \{TC(X)\}$, $MW^{max} = \max_{X \in PE \cup \widehat{PE}} \{MW(X)\}$, and $MW^{min} = \min_{X \in PE \cup \widehat{PE}} \{MW(X)\}$. Furthermore, let MW^m and TC^m denote the $MW(X)$ and $TC(X)$ values for the solution X corresponding to the m^{th} point in P such that $1 \leq m \leq |P|$; and, without loss of generality, we assume that the points within P are ordered such that $TC^{m-1} \leq TC^m$ and $MW^{m-1} \geq MW^m$.

The measures considered based on the objective space are the actual and percent differences between the hypervolumes of PF and \widehat{PF} . Hypervolume is typically used to compare Pareto fronts (see, e.g., Knowles and Corne (2002), Zitzler et al. (2008)) and it is defined based on a reference point. The hypervolume for a set of points P , denoted by $HV(P)$, is defined as $HP(P) = \sum_{m=1}^{|P|} [MW^{m-1} - MW^m] \times [TC^{|P|+1} - TC^m]$, where MW^0 is the MW of the reference point and $TC^{|P|+1}$ is the TC of the reference point. Then, the total volume, denoted by TV , will be $TV = (TC^{|P|+1} - TC^{min}) \times (MW^0 - MW^{min})$. Similar to Minella et al. (2011), we define the reference point for the objective space using 20% increments from the worst objective function values; that is, $[1.2TC^{max}, 1.2MW^{max}]$ is the reference point. We define hypervolumes of the Pareto fronts as the percentages of the total volume captured. In particular, we define hypervolumes of PF and \widehat{PF} , denoted by HV and \widehat{HV} , by letting $HV = 100 \times \frac{HV(PF)}{TV} \%$ and $\widehat{HV} = 100 \times \frac{HV(\widehat{PF})}{TV} \%$. Note that hypervolume defines the area, which is dominated by a Pareto front; therefore, a larger hypervolume implies a better Pareto front. It should

be noted that, by definition, we will have $1 \geq HV \geq \widehat{HV} \geq 0$ because PF is the actual exact Pareto front for **CSP**. Then, the two difference measures between HV and \widehat{HV} are: actual difference $ADV = (HV - \widehat{HV})$ and percent difference $PDV = \left(100 \times \frac{HV - \widehat{HV}}{HV} \%\right)$ (see also Kovacs et al. (2015)).

The last set of measures are unary measures, which assign a single value for comparing PF and \widehat{PF} . The first unary measure we consider is the unary-indicator, denoted by UI , such that $UI = \max_{X' \in PE} \left\{ \min_{X'' \in \widehat{PE}} \left\{ \max \left\{ \frac{TC(X'')}{TC(X')}, \frac{TC(X')}{TC(X'')} \right\} \right\} \right\}$ (see, e.g., Zitzler et al. (2003), Kovacs et al. (2015)). Note that when $PF \equiv \widehat{PF}$, $UI = 1$; and we have $UI \geq 1$ by definition. We convert the UI value to percentage by letting $UI \rightarrow 100 \times UI\%$. The second unary measure considered is the generational distance, denoted by GD , such that

$$GD = \frac{\sqrt{\sum_{X'' \in \widehat{PE}} d(X'')^2}}{|\widehat{PF}|}, \text{ where } d(X'') = \min_{X' \in PE} \left\{ \sqrt{(TC(X'') - TC(X'))^2 + (MW(X'') - MW(X'))^2} \right\} \text{ for a } X'' \in \widehat{PE} \text{ (see, e.g.,}$$

Rudolph (1998), Van Veldhuizen and Lamont (2000), Kovacs et al. (2015)). That is, $d(X'')$ defines the minimum of the distances from the point corresponding to X'' within \widehat{PF} to the points within PF . To get relative distance measure, we redefine $d(X'')$ as a percentage of the maximum possible distance, denoted by MD , which is defined as

$$MD = \sqrt{(TC^{max} - TC^{min})^2 + (MW^{max} - MW^{min})^2}. \text{ That is, we let } d(X'') \rightarrow 100 \times \frac{d(X'')}{MD} \% \text{ while calculating } GD.$$

Table 6 documents the averages of ADV , PDV , UI , and GD values over the 10 problem instances solved within each problem set under 2- and 3-mode cases. Figure 4

shows how these average values change for problem sets with increasing $|I|$ for 2- and 3-mode cases.

We have the following observations from Table 6 and Figure 4.

- Based on the actual and percent differences (i.e., ADV and PDV) of the hypervolumes of PF and \widehat{PF} documented in Table 6, we can see that these differences are around 6% on overall average for both 2- and 3-mode cases (6% and 6.24% for 2-mode cases and 6% and 6.52% for 3-mode cases). As expected, \widehat{PF} dominates a smaller area than PF does; however, the difference is relatively small, which implies that PF and \widehat{PF} are close to each other. Furthermore, we can observe from Figure 4 that, these differences do not follow an increasing or a decreasing pattern as $|I|$ grows, which indicates that how close \widehat{PF} is to PF does not change with the problem size.

Table 6. Comparison of PF and \widehat{PF} based results on hypervolume and unary measures.

$ I $	2-mode results				3-mode results			
	ADV	PDV	UI	GD	ADV	PDV	UI	GD
10	6.11%	6.35%	103.63%	2.73%	6.17%	6.71%	105.14%	1.51%
12	4.84%	5.02%	102.37%	3.40%	4.98%	5.40%	104.11%	1.74%
14	5.05%	5.26%	102.93%	3.11%	6.16%	6.68%	105.23%	2.04%
16	4.81%	5.02%	102.70%	2.27%	6.71%	7.32%	105.24%	1.57%
18	5.76%	6.01%	103.11%	2.36%	5.90%	6.39%	104.88%	1.81%
20	5.49%	5.74%	102.84%	2.24%	4.92%	5.34%	104.03%	1.93%
25	7.02%	7.31%	103.46%	1.99%	5.79%	6.28%	104.46%	1.56%
30	6.42%	6.69%	103.15%	2.69%	7.14%	7.73%	105.61%	1.22%
35	7.60%	7.94%	103.62%	2.15%	5.87%	6.36%	104.78%	1.53%
40	6.78%	7.08%	103.14%	2.08%	6.44%	7.01%	105.20%	1.48%
Avg.	5.99%	6.24%	103.09%	2.50%	6.01%	6.52%	104.87%	1.64%

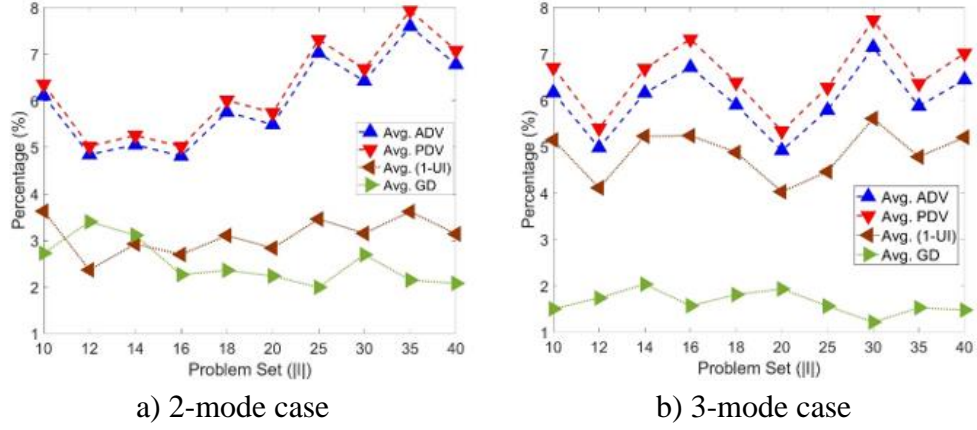


Figure 4. Illustration of average extreme.

- Based on unary-indicator (i.e., UI), we can see that it is around 103% for 2-mode cases and 105% for 3-mode cases on overall average. These indicate that, for each point on PF , \widehat{PF} had a point that deviates by at most 3% and 5% on overall average for 2- and 3-mode cases, respectively. Note that these numbers are consistent with the percent differences between average density points of PF and PF , i.e., average DP and \widehat{DP} , documented in Tables 4 and 5. Furthermore, it can be observed from Figure 4 that average UI values do not follow an increasing or a decreasing pattern as $|I|$ grows, which indicates that the unary-indicator value does not change with the problem size.
- Based on generational distance (as a percentage of the maximum distance, i.e., GD), we have similar observations with UI .

Our comparison of PF and \widehat{PF} based on hypervolume differences and the two unary indicators can be summarized as follows. On average, Algorithm 2 can find points that do not significantly deviate from the points returned by Algorithm 1. Particularly, all these measures (hypervolume differences and unary indicators) imply that PF are PF are

in close proximity; and the problem size does not have an observable impact on this proximity.

5. CONCLUDING REMARKS

Motivated from construction planning requirements in a semiconductor manufacturing facility, we presented a bi-objective multi-mode flexible resource profile project scheduling problem with a single unconstrained renewable discrete resource under discrete time. The project activities are installation, demolition, and modification of the machines/tools within the manufacturing facility and the resource is the labor utilized for. Individual activities have work-content requirements, time windows, and lower and upper limits on the resource that can be simultaneously used. Furthermore, the project schedule has a deadline. The objectives considered are total labor cost minimization and maximum resource (labor) usage minimization throughout the project schedule. Finally, preemption is allowed. To the best of our knowledge, a project scheduling problem with these settings has not been investigated in the literature.

We first present the bi-objective optimization model for this problem. After that, we discuss the implementation of the well-known classical ε -constraint method for generating the exact Pareto front of the problem. Given the computational complexity of the problem, we then develop a simple approximation method. This approximation method is based on partial linear relaxation of the problem and uses rounding and improvement procedures to find near Pareto efficient solutions. Based on a set of numerical studies and qualitative comparison metrics, we believe that the proposed

approximation method is computationally very efficient and finds solutions within close proximity of the exact Pareto front.

We realize that generalized settings remain as future research directions. One immediate resource direction is to consider multiple constrained and/or unconstrained resources (renewable and nonrenewable). Furthermore, different resource leveling objectives can be considered. Another potential research direction is to analyze different heuristic methods for the problem and its possible extensions. We believe that the problem instances we generated and the solution method we proposed can be useful in such future research studies.

REFERENCES

- Atan, T., & Eren, E. (2018). Optimal project duration for resource leveling. *European Journal of Operational Research*, 266(2), 508-520.
- Ballestín, F., & Blanco, R. (2011). Theoretical and practical fundamentals for multi-objective optimisation in resource-constrained project scheduling problems. *Computers & Operations Research*, 38(1), 51-62.
- Ballestín, F., & Blanco, R. (2015). Theoretical and Practical Fundamentals. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 411-427). Cham: Springer.
- Ballestín, F., Valls, V., & Quintanilla, S. (2008). Pre-emption in resource-constrained project scheduling. *European Journal of Operational Research*, 189(3), 1136-1152.
- Baumann, P., & Trautmann, N. (2014). An MILP formulation for scheduling of work-content-constrained projects. *Proceedings of the 14th International Conference on Project Management and Scheduling* (pp. 24-39). TUM School of Management.
- Baumann, P., Fündeling, C.-U., & Trautmann, N. (2015). The resource-constrained project scheduling problem with work-content constraints. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 533-544). Cham: Springer.
- Blazewicz, J., Lenstra, J. K., & Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1), 11-24.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1), 3-41.
- Caramia, M., & Dell'Olmo, P. (2003). Assessing the resource usage in scheduling with incompatibilities. *OR Spectrum*, 25(4), 521-547.
- Cheng, J., Fowler, J., & Kempf, K. (2012). Simulation-based multi-mode resource-constrained project scheduling of semiconductor equipment installation and qualification. *Proceedings of the 2012 Winter Simulation Conference (WSC)* (pp. 1-12). IEEE.

- Cheng, J., Fowler, J., Kempf, K., & Mason, S. (2015). Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. *Computers & Operations Research*, 53, 275-287.
- Coughlan, E. T., Lübbecke, M. E., & Schulz, J. (2010). A branch-and-price algorithm for multi-mode resource leveling. *The 9th International Symposium on Experimental Algorithms* (pp. 226-238). Springer.
- Coughlan, E. T., Lübbecke, M. E., & Schulz, J. (2015). A branch-price-and-cut algorithm for multi-mode resource leveling. *European Journal of Operational Research*, 245(1), 70-80.
- Damci, A., Arditi, D., & Polat, G. (2016). Impacts of different objective functions on resource leveling in line-of-balance scheduling. *KSCE Journal of Civil Engineering*, 20(1), 58-67.
- Demeulemeester, E. L., & Herroelen, W. S. (2002). *Project scheduling: a research handbook*. Dordrecht: Kluwer Academic Publishers.
- Dwyer, P., & Konur, D. (2018). Schedule Optimization for Construction in Manufacturing Environments. *Proceedings of the 2018 IISE Annual Conference* (pp. 1765-1770). IISE.
- Dwyer, P., & Konur, D. (2020). Schedule optimization for construction: comparison of resource leveling techniques. *Proceedings of the 2020 IISE Annual Conference. Accepted*. IISE.
- Fowler, J. W., & Mönch, L. (2017). Modeling and analysis of semiconductor manufacturing. In A. Tolk, J. W. Fowler, G. Shao, & E. Yücesan, *Advances in Modeling and Simulation* (pp. 301-313). Cham: Springer.
- Fündeling, C.-U., & Trautmann, N. (2010). A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, 203(3), 568-574.
- Gupta, A. K., & Sivakumar, A. I. (2006). Job shop scheduling techniques in semiconductor manufacturing. *The International Journal of Advanced Manufacturing Technology*, 27(11-12), 1163-1169.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1-14.
- Herroelen, W. (2005). Project scheduling—Theory and practice. *Production and Operations Management*, 14(4), 413-432.

- Herroelen, W., De Reyck, B., & Demeulemeester, E. (1998). Resource-constrained project scheduling: a survey of recent developments. *Computers & Operations Research*, 25(4), 279-302.
- Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. *Proceedings of the 2002 Congress on Evolutionary Computation* (pp. 711-716). IEEE.
- Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *Omega*, 29(3), 249-272.
- Konur, D., & Schaefer, B. (2016). Economic and environmental comparison of grouping strategies in coordinated multi-item inventory systems. *Journal of the Operational Research Society*, 67(3), 421-436.
- Konur, D., Campbell, J. F., & Monfared, S. A. (2017). Economic and environmental considerations in a stochastic inventory control model with order splitting under different delivery schedules among suppliers. *Omega*, 71, 46-65.
- Kovacs, A. A., Parragh, S. N., & Hartl, R. F. (2015). The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research*, 247(2), 441-458.
- Kreter, S., Rieck, J., & Zimmermann, J. (2014). The total adjustment cost problem: Applications, models, and solution algorithms. *Journal of Scheduling*, 17(2), 145-160.
- Kreter, S., Schutt, A., Stuckey, P. J., & Zimmerman, J. (2018). Mixed-integer linear programming and constraint programming formulations for solving resource availability cost problems. *European Journal of Operational Research*, 266(2), 472-486.
- Mika, M., Waligóra, G., & Węglarz, J. (2015). Overview and state of the art. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling* Vol. 1 (pp. 445-490). Cham: Springer.
- Miller, J. L., & Franz, L. S. (1996). A binary-rounding heuristic for multi-period variable-task-duration assignment problems. *Computers & Operations Research*, 23(8), 819-828.
- Minella, G., Ruiz, R., & Ciavotta, M. (2011). Restarted iterated Pareto greedy algorithm for multi-objective flowshop scheduling problems. *Computers & Operations Research*, 38(11), 1521-1533.

- Mönch, L., Chien, C.-F., Dauzère-Pérès, S., Ehm, H., & Fowler, J. W. (2018c). Modelling and analysis of semiconductor supply chains. *International Journal of Production Research*, 56(13), 4521-4523.
- Mönch, L., Fowler, J. W., & Mason, S. J. (2012). *Production planning and control for semiconductor wafer fabrication facilities*. New York: Springer.
- Mönch, L., Fowler, J. W., Dauzère-Pérès, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *Journal of Scheduling*, 14(6), 583-599.
- Mönch, L., Uzsoy, R., & Fowler, J. W. (2018a). A survey of semiconductor supply chain models part I: semiconductor supply chains, strategic network design, and supply chain simulation. *International Journal of Production Research*, 56(13), 4524-4545.
- Mönch, L., Uzsoy, R., & Fowler, J. W. (2018b). A survey of semiconductor supply chain models part III: master planning, production planning, and demand fulfilment. *International Journal of Production Research*, 56(13), 4565-4584.
- Naber, A. (2017). Resource-constrained project scheduling with flexible resource profiles in continuous time. *Computers & Operations Research*, 84, 33-45.
- Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 239(2), 335-348.
- Neumann, K., & Zimmermann, J. (1999). Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, 117(3), 591-605.
- Neumann, K., Schwindt, C., & Zimmermann, J. (2003). *Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions*. Springer.
- Özdamar, L., & Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. *IIE Transactions*, 27(5), 574-586.
- Özlen, M., & Azizoglu, M. (2009). Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1), 25-35.
- Quintanilla, S., Lino, R., Pérez, Á., Ballestín, F., & Valls, V. (2015). Integer preemption problems. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 231-250). Cham: Springer.

- Ranjbar, M., & Kianfar, F. (2010). Resource-constrained project scheduling problem with flexible work profiles: a genetic algorithm approach. *Transaction E: Industrial Engineering*, 17(1), 25-35.
- Rieck, J., & Zimmermann, J. (2015). Exact methods for resource leveling problems. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 361-387). Cham: Springer.
- Rieck, J., Zimmermann, J., & Gather, T. (2012). Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research*, 221(1), 27-37.
- Rodrigues, S. B., & Yamashita, D. S. (2010). An exact algorithm for minimizing resource availability costs in project scheduling. *European Journal of Operational Research*, 206(3), 562-568.
- Rodrigues, S. B., & Yamashita, D. S. (2015). Exact Methods for the Resource Availability. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 319-338). Cham: Springer.
- Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the pareto set. *1998 IEEE International Conference on Evolutionary Computation Proceedings* (pp. 511-516). IEEE.
- Saltzman, R. M., & Hillier, F. S. (1992). A heuristic ceiling point algorithm for general integer linear programming. *Management Science*, 38(2), 263-283.
- Schaefer, B., & Konur, D. (2015). Economic and environmental considerations in a continuous review inventory control system with integrated transportation decisions. *Transportation Research Part E: Logistics and Transportation Review*, 80, 142-165.
- Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE Spectrum*, 34(6), 52-59.
- Schwindt, C., & Paetz, T. (2015). Continuous preemption problems. In C. Schwindt, & J. Zimmermann, *Handbook on Project Management and Scheduling Vol. 1* (pp. 251-295). Cham: Springer.
- Schwindt, C., & Zimmermann, J. (2015a). *Handbook on project management and scheduling vol. 1*. Springer International.
- Schwindt, C., & Zimmermann, J. (2015b). *Handbook on project management and scheduling vol. 2*. Springer International.

- Takamoto, M., Yamada, N., Kobayashi, Y., Nonaka, H., & Okoshi, S. (1995). Zero-one quadratic programming algorithm for resource leveling of manufacturing process schedules. *Systems and Computers in Japan*, 26(10), 68-76.
- Tritschler, M., Naber, A., & Kolisch, R. (2014). A genetic algorithm for resource-constrained project scheduling problem with flexible resource profiles. *Proceedings of the 14th International Conference on Project Management and Scheduling* (pp. 230-233). TUM School of Management.
- Tritschler, M., Naber, A., & Kolisch, R. (2017). A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operational Research*, 262(1), 262-273.
- Uzsoy, R., Fowler, J. W., & Mönnch, L. (2018). A survey of semiconductor supply chain models part II: demand planning, inventory management, and capacity planning. *International Journal of Production Research*, 56(13), 4546-4564.
- Uzsoy, R., Lee, C.-Y., & Martin-Vega, L. A. (1992). A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning. *IIE Transactions*, 24(4), 47-60.
- Uzsoy, R., Lee, C.-Y., & Martin-Vega, L. A. (1994). A review of production planning and scheduling models in the semiconductor industry part II: Shop-floor control. *IIE Transactions*, 26(5), 44-55.
- Van Peteghem, V., & Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235(1), 62-72.
- Van Veldhuizen, D. A., & Lamont, G. B. (2000). On measuring multiobjective evolutionary algorithm performance. *Proceedings of the 2000 Congress on Evolutionary Computation* (pp. 204-211). IEEE.
- Viana, A., & de Sousa, J. P. (2000). Using metaheuristics in multiobjective resource constrained project scheduling. *European Journal of Operational Research*, 120(2), 359-374.
- Vohra, R. V. (1988). A quick heuristic for some cyclic staffing problems with breaks. *Journal of the Operational Research Society*, 39(11), 1057-1061.
- Węglarz, J., Józefowska, J., Mika, M., & Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes – A survey. *European Journal of Operational Research*, 208(3), 177-205.

- Zimmermann, A. (2016). An MIP-based heuristic for scheduling projects with work-content constraints. *2016 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1195-1199). IEEE.
- Zitzler, E., Knowles, J., & Thiele, L. (2008). Quality assessment of pareto set approximations. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński, *Multiobjective Optimization* (pp. 373-404). Springer.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117-132.

SECTION

2. CONCLUSIONS AND RECOMMENDATIONS

2.1. CONCLUSIONS

This research presents the development of a novel multi-objective scheduling optimization model for multiple construction projects in a manufacturing and operations environment. The novelty arises from the scheduling flexibility through the use of optional overtime and non-continuous resource allocation to a project once it starts. The multiple objectives that were studied are the minimization of total labor cost and the minimization of total resource leveling. The model was tested to solve resource-constrained problem that were randomly generated to avoid bias. After this proof of concept, this research developed a heuristic utilizing partial linear relaxation and rounding method and compared the heuristic and exact method against case studies of multiple lengths. The analysis of the results prove that the novel model can be scaled to generate near optimum schedule for large projects. The model enables project managers to plan work across multiple projects in a manufacturing setting and properly allocate the available trade resources.

The final step in this research generated multiple Pareto Fronts utilizing various techniques for resource leveling. The techniques that were compared include the minimization of the sum of the absolute deviations in resource usage for a determined time interval, the minimization of the maximum resource usage for a determined time interval, and the minimization of the maximum deviation in resource usage for a

determined time interval. The results from this study demonstrate that the technique utilized in resource leveling greatly affect the schedules that are generated, and the project management team must have clear insight to the risks that are to be minimized prior to chosen a model.

2.2. RECOMMENDATIONS

The model developed and tested in this research is a novel and useful scheduling method for construction in manufacturing environments. The model should be run multiple times throughout the lifespan of a program. Running the model during the positioning and planning section allows the project management team to make an educated and accurate forecast of the total program cost based on how much risk the team is willing to take. The outcome of the model is dependent on the market conditions with the total number of resources available as an input. As the market conditions change the model need to be updated to reflect current conditions.

While this model is not designed for a traditional construction scheduling project, there are many aspects of it that can be expanded upon in future research. Our model is novel in its utilization of resources and the interrelation between a task's duration and the number of allocated resources. Future research could expand on this aspect while adapting the model to shift non-critical activities to achieve the model's objective. As this current model does not have non-critical activities (since each project is independent), we were not able to test its functionality against existing methods in literature.

The first part of our research involved converting the parameters of the manufacturing construction environment into a linear model. To accomplish this task, we had to include additional constraints and parameters to convert a non-linear system into a format to solve utilizing linear programming optimization techniques. Future research can explore how the output of the model described in our research compares to the results of a non-linear approach. We believe that the model developed in this research is a great starting point for continued research on non-traditional construction scenarios.

BIBLIOGRAPHY

- [1] Mellentien, Christoph, and Norbert Trautmann. "Resource allocation with project management software." *OR-Spektrum* 23.3 (2001): 383-394.
- [2] Caramia, Massimiliano, and Paolo Dell'Olmo. "Assessing the resource usage in scheduling with incompatibilities." *OR Spectrum* 25.4 (2003): 521-547.
- [3] Menesi, Wail, and Tarek Hegazy. "Multimode resource-constrained scheduling and leveling for practical-size projects." *Journal of management in engineering* 31.6 (2015): 04014092.1-04014092.7.
- [4] Heon Jun, Dho, and Khaled El-Rayes. "Multiobjective optimization of resource leveling and allocation during construction scheduling." *Journal of construction engineering and management* 137.12 (2011): 1080-1088.
- [5] Son, Jaeho, and Kris G. Mattila. "Binary resource leveling model: Activity splitting allowed." *Journal of construction engineering and management* 130.6 (2004): 887-894.
- [6] Hegazy, Tarek. "Optimization of resource allocation and leveling using genetic algorithms." *Journal of construction engineering and management* 125.3 (1999): 167-175.
- [7] Burgess, A. R., and James B. Killebrew. "Variation in activity level on a cyclical arrow diagram." *Journal of Industrial Engineering* 13.2 (1962): 76-83.
- [8] Son, Jaeho, and Mirosław J. Skibniewski. "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *Journal of construction engineering and management* 125.1 (1999): 23-31.
- [9] Kolisch, R. and Hartmann, S. (2006) Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1), 23–37.
- [10] Benjaoran, Vacharapoom, Wisitsak Tabyang, and Nart Sooksil. "Precedence relationship options for the resource levelling problem using a genetic algorithm." *Construction management and economics* 33.9 (2015): 711-723.
- [11] Rokou, Elena, Manos Dermitzakis, and Konstantinos Kirytopoulos. "Multi-project flexible resource profiles project scheduling with Ant Colony Optimization." *IEEE International Conference on Industrial Engineering and Engineering Management* (2014): 642-646.

- [12] Atan, Tankut, and Elif Eren. "Optimal project duration for resource leveling." *European Journal of Operational Research* 266.2 (2018): 508-520.
- [13] Fündeling, C-U., and Norbert Trautmann. "A priority-rule method for project scheduling with work-content constraints." *European Journal of Operational Research* 203.3 (2010): 568-574.
- [14] Trautmann, Norbert, and Philipp Baumann. "Resource-allocation capabilities of commercial project management software: An experimental analysis." 2009 *International Conference on Computers & Industrial Engineering*. IEEE, 2009.

VITA

Patrick Bernard Dwyer received a Bachelor of Science in Chemical Engineering from Lafayette College in 2008 and a Master of Science in Engineering Management from Stevens Institute of Technology in 2012. He received a Doctor of Philosophy in Engineering Management from Missouri University of Science & Technology in August of 2020. He worked as a Project Manager for the Department of Defense from 2008 until 2012 at Picatinny Arsenal, leading the development of an ammunition industrial base strategic plan to support the Military Service's conventional ammunition requirements. His main focus was to monitor the cost, schedule and performance of projects across the industrial base facilities as well as to manage environmental compliance across the production bases. In 2012, Patrick transitioned to a position at Intel Corporation as a Construction Project Manager acting as an owner's representative. He has managed projects ranging from creation of office buildings and public work projects to the construction of cleanroom semiconductor manufacturing facilities. Along with maintaining project scope, schedule, and budget, Patrick's main focus at Intel was sustaining manufacturing operations during construction activities. In addition to being OSHA10 certified, Patrick received his Project Management Professional certification from the Project Management Institute in 2013.