



THE HONG KONG  
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library

包玉剛圖書館

---

## Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

**By reading and using the thesis, the reader understands and agrees to the following terms:**

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

### IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact [lbsys@polyu.edu.hk](mailto:lbsys@polyu.edu.hk) providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

ROBUST OPTIMIZATION FOR GENERALIZED  
PROJECT NETWORKS

CHEN SIYUAN

MPhil

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University  
Department of Logistics and Maritime Studies  
Robust Optimization for Generalized Project Networks

Chen Siyuan

A thesis submitted in partial fulfilment of the  
requirements for the degree of Master of Philosophy

June 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

\_\_\_\_\_ (Signed)

\_\_\_\_\_ Siyuan CHEN (Name of student)

# Abstract

Completion time estimation is a key component for project management. In addition to uncertain task times, uncertain task outcomes also have a significant impact on the evaluation of completion time. For example, due to some uncontrolled factors in a project, certain task may have a probability of failure, which will result in full repetition or partial rework of the task. Introduction of new task or change of precedence relationships may occur during project execution as a contingency measure. Another example is probabilistic branching, i.e., after the project reaches a milestone, it may be necessary to choose among alternative plans by that time. However, these uncertainties are often ignored in standard approaches, e.g., the program evaluation and review technique (PERT). In this thesis, we introduce the generalized project network to capture all the uncertainties in both task durations and task outcomes. A distributionally robust optimization model is developed to estimate the project completion time as well as the target-based measure of tardiness. We develop an efficient algorithm to solve the distributionally robust optimization model. The performance of the estimates obtained through distributionally robust optimization is evaluated through numerical studies. Numerical studies show that our model is practical and general to capture all the uncertainties in both task durations and task outcomes in completion time estimation problems.

The proposed algorithm demonstrates exceptional efficiency, allowing us to solve one large size project instance, solved by the simulation method in the literature, in 98.2 seconds.

# Acknowledgements

I would like to thank everyone who has supported and contributed to the completion of this thesis and my MPhil degree.

Special thanks go to my supervisor, SONG Miao, for her invaluable guidance, unwavering support, and expertise throughout this research.

I am also grateful to the members of my thesis committee for their valuable input and feedback, which greatly improved this work.

I extend my appreciation to the HK Polyu for providing the necessary resources and research facilities that facilitated the successful completion of this study.

To my family and friends, thank you for your constant encouragement and support throughout this journey. Your belief in me has been a tremendous source of motivation.

Thank you all for your support and encouragement.

# Contents

<b>1</b>	<b>Introduction and Literature Review</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Literature Review . . . . .	5
<b>2</b>	<b>Generalized Project Network</b>	<b>10</b>
<b>3</b>	<b>Formulation and Solution Procedure</b>	<b>18</b>
3.1	Marginal Distribution Model for $\tilde{c}$ . . . . .	18
3.2	Distributionally Robust Model for $\tilde{c}$ and $\tilde{\eta}$ . . . . .	27
3.2.1	Find the Initial Subset $\Omega'$ . . . . .	30
3.2.2	Solve the Pricing Problem . . . . .	35
<b>4</b>	<b>Computational Study</b>	<b>39</b>
4.1	Classical Research and Development Projects Network . . . . .	40
4.1.1	Input Explanation . . . . .	42
4.1.2	Numerical Result . . . . .	43
4.2	Performance Test . . . . .	44
4.2.1	Performance Test in Small-scale and Middle-scale Networks	45
4.2.2	Performance Test in Large-scale Network . . . . .	47



<i>CONTENTS</i>	v
4.3 Solution Quality Check . . . . .	53
<b>5 Conclusion and Future Work</b>	<b>56</b>
5.1 Conclusion . . . . .	57
5.2 Future Work on Formulation Structure . . . . .	58
5.2.1 DRO Model for $\tilde{\eta}$ under Deterministic $\mathbf{c}$ . . . . .	58
5.2.2 DRO Model Allowing Dependence between $\tilde{\eta}$ and $\tilde{\mathbf{c}}$ . . . . .	61
5.2.3 Other Formulation Extensions . . . . .	62
5.3 Research Field Extensions . . . . .	63

# List of Figures

2.1	Illustration of Probabilistic Branching . . . . .	13
3.1	Intuitive Explanation of Algorithm 2 . . . . .	35
4.1	Network of R&D Process for One Project . . . . .	42
4.2	Performance Result in the Large-scale Network with $ \mathcal{A}_p $ . . . . .	48
4.3	Performance Result in Large-scale Network with $ \mathcal{N}_{b_s} $ . . . . .	50
4.4	Case of Solution Quality Check . . . . .	54

# List of Tables

4.1	Numerical Result in Small-scale and Middle-scale Networks . . .	46
4.2	Numerical Result in the Large-scale Network . . . . .	49
4.3	Marginal Distribution Information . . . . .	54

# Chapter 1

## Introduction and Literature Review

### 1.1 Introduction

Many projects, especially those involving research and development activities, have an uncertain scope. For example, new tasks must be introduced during the execution of a project. Another example is probabilistic branching, i.e., after the project reaches a milestone, it may be necessary to choose among alternative plans based on information that has been revealed by that time. However, uncertainty in the project scope is often ignored in standard approaches for project planning, which consider a given list of tasks and assume that it will not be changed during the project lifecycle. Standard approaches simplify projects further in the following two respects. First, it is assumed that the precedence relations of the tasks, i.e., whether one task must be performed before the start of another, are known for certain at the planning phase of the project. Whereas, in reality, precedence relations may change over time as new information arrives. A task that was previously believed to be independent of another may turn out to need the information from

the latter and cannot start before its completion. Second, the outcomes of certain tasks may not meet expectations or quality standards. In this case, it is common to have these tasks repeated or partially reworked. Unfortunately, the practice of task repetition is not considered in standard project planning methods either.

In this study, we generalize the project networks to incorporate uncertainties in the network structure. This framework enables us to capture the introduction of new tasks, task repetition or partial rework, change of precedence, and probabilistic branching. The few existing studies on generalized project networks mainly resort to simulation or Markov chain approaches, which require the complete joint distribution of random task durations and other uncertain events. However, due to the uniqueness of projects, it is always a challenge to determine the joint distribution, which limits the practical applicability of the existing approaches. To address this issue, we adopt the methodologies of distributionally robust optimization, whose strong performance has been proved in the literature when applied to standard project management problems with incomplete information on the distribution of uncertain task durations. Marginal distributions are usually considered to build ambiguity set in the literature. The network uncertainty can be only captured by Bernoulli variables because the precedence relation has only two outcomes. For Bernoulli random variables, if we know moments information, we know Marginal distributions. Thus, in this study, we apply the Marginal distribution model. As a first step to studying generalized project networks using distributionally robust optimization, we focus on the evaluation of project makespan. The universal “triple constraint” defines priorities for time, cost, and scope in projects (Kerzner 2013, pp. 8-9). In reality, however, time and cost performance typically show a strong positive correlation. Companies usually need to pay higher labor and material

costs for projects with longer completion time. This work studies the time issue under the uncertain scope and therefore addresses all three components of the triple constraint.

This thesis is organized as follows.

- In Chapter 1, we provide a comprehensive introduction to show the background and motivations and review some literature.
- In Chapter 2, we delve into the specifics of the generalized project network. Here, we outline the generalization and uncertainties we will consider in the execution of projects.
- Chapter 3 focuses on the formulation and solution procedure. Within this chapter, we elaborate on the establishment of the model and the step-by-step derivation process leading to formulation results.
- In Chapter 4, we shift our attention to computational research. Here, we apply our approach to one instance from the literature. By solving the instance using the proposed approach, we demonstrate the practicality and generality of our research. Additionally, we conduct three sets of performance experiments to test the practicality of models and the efficiency of the algorithm in different scale networks.
- Finally, in the last chapter, we draw conclusions based on our research findings and reflect on the broader implications of our work. We also outline future work.

## 1.2 Literature Review

Hall (2016) describes some project management research directions and some mainstream branching as follows: 1. Project expediting. Delays in previous tasks, changed circumstances, and modifications of the project scope motivate the need to expedite. Crashing decisions with determined task time can be modeled in linear programming, however, the uncertainty of task time makes crashing decisions more complex, which requires the estimation of uncertain task durations over the remainder of the project. Two standard approaches, PERT and Monte Carlo simulation, assume full knowledge of the probability distributions of the arc lengths and independence of the task durations, instead, robust optimization provides a more efficient way to find crashing decisions without the need of these assumptions. 2. Project selection. Selection of the right projects can bring better returns and ease of management. Selecting individually and accepting multiple projects simultaneously are two fundamental ways to select the project. Control of risk through diversification, better resource utilization, the potential to model dependency between projects, and optimization of overall portfolio performance are provided by the project portfolio planning approach to project selection. Project selection with determined returns can be solved by the knapsack model. For uncertainty, robust optimization is applied to solve a project selection problem as well as the target-measured underperformance risk. 3. Estimation of project completion time. Accurate estimation of completion time or makespan, can help a company prevent expensive crashing costs and loss of a contract when bidding. The standard method is PERT, however, it ignores many uncertainties of modern complicated and unique projects. Limited distributional information on task duration and de-

pendence between both tasks and precedence relationships stimulate more application of DRO in project management. Among these project management research branches, completing time estimation is crucial not only because it is a basic need of project management but it also plays an important role in other aspects, such as crashing decisions. In practice, accurate estimation of completion tasks can help the company save cashing money and prevent loss of contracts.

The earliest algorithms for scheduling projects and estimating their duration, i.e., the critical path method (CPM) (Kelley and Walker 1959) and the program evaluation and review technique (PERT) (Fazar 1959, Malcolm et al. 1959), were proposed in the late 1950s. CPM assumes deterministic durations for all tasks, from which the project makespan is calculated as the sum of the durations of the tasks on the longest path in the project network. PERT uses the expected duration of each task to determine the critical path, as in CPM. Further, recognizing uncertainties in the task durations, it estimates the variance of the project makespan as the sum of variances of the tasks on the critical path. Although PERT remains a standard method for makespan estimation, it has been widely documented that PERT often significantly underestimates the expected makespan (see, e.g., Welsh 1965, Hartley and Wortham 1966, Klingel 1966, and Schonberger 1981). This error is mainly caused by merging paths and hence is known as the *merge bias*. Intuitively, the cause of merge bias is that the longest path in expectation is not the longest in realization due to uncertain task durations. Therefore, by basing its estimates solely on the longest path in expectation, PERT ignores important relevant information.

Hagstrom (1988) shows that the problem of computing either the expected makespan or the probability to complete a project by a given time is #P-complete.



Due to this severe intractability, the literature predominantly focuses on approximation approaches. One exception is Kulkarni and Adlakha (1986) which analyzes the distribution of project makespan and the probability for a path to be critical for independent and exponentially distributed task durations. Methods to compensate for the merge bias are based on (a) details of merging activity (Gong and Hugsted 1993, Gong and Rowings 1995, and Banerjee and Paul 2008) or (b) consideration of multiple paths (see, e.g., Anklesaria and Drezner 1986, Sculli and Shum 1991, and Jun and El-Rayes 2011).

Another active research stream identifies bounds on the distribution and/or expectation of project makespan. This work includes Fulkerson (1962), Kleindorfer (1971), Robillard and Trahan (1976), Spelde (1976), Shogan (1977), Dodin (1985), Ludwig et al. (2001), and Möhring (2001). More recent studies apply distributionally robust optimization, which finds the worst-case expectation among a structured set of distributions. In Meilijson and Nádas (1979), task durations are specified by marginal distributions. They identify a computational procedure to find the worst-case expected project makespan and the corresponding probability for a path to be longest. Bertsimas et al. (2004, 2006) determine the worst-case expected tardiness (and makespan) among all distributions of task durations with given marginal moments. They also compute the persistency under the worst-case distribution, i.e. the probability for a task to be critical sometimes called the criticality index. Natarajan et al. (2011) then consider the case where the distribution information on task duration is specified by both marginal moments and cross moments. An approximation based on a completely positive representation is developed to estimate the worst-case expected makespan and criticality index. In Doan and Natarajan (2012), tasks are partitioned into non-overlapping sets, and

the joint distribution of the task durations in each set is given. Assuming task durations follow discrete distributions, they identify cases under which the worst-case expected makespan can be computed in polynomial time. Zheng et al. (2016) consider task durations that follow a multivariate normal distribution and propose the least squares normal approximation and the least squares quadratic approximation to the task duration, which can be obtained from the criticality index.

All the above studies assume that the set of tasks and the precedence relationships among them are known deterministically. Yet, in reality, uncertainty in network structure is present in many projects, especially those for research and development, new product and service development, and pharmaceutical and software development. In such situations, when a project starts, even the list of tasks is not known with certainty. Some early studies consider uncertain network structure. Eisner (1962) considers probabilistic branching, i.e., after completing a task, the project will take exactly one of several alternative paths with a certain probability, and studies the network based on enumerating possible outcomes. Elmaghraby (1964) presents a graphic representation and an algebra to analyze networks with probabilistic branching and task repetition. These features are also enabled in the simulation package Graphical Evaluation and Review Technique (GERT) and its successor Q-GERT (see, e.g., Pritsker 1966, 1979, Moore and Taylor 1977, and Taylor and Moore 1980). Carracosa et al. (1998) and Ahmadi et al. (2001) develop Markov chain models for projects with task repetition and information dependency. Cho and Eppinger (2005) consider similar project networks with resource constraints and use simulation and heuristics to estimate the makespan. Implementing all these approaches requires knowledge of a joint probability distribution of the task durations and the network structure. Problematically, since projects are

unique, this distributional information is rarely available. Consequently, distributionally robust optimization, which relies on only partial information about the joint distribution, is a natural and potentially powerful approach to studying the makespan of projects with uncertain network structures. Robust optimization has been applied to project management to model crashing decisions during project execution by Goh and Hall (2013) and project portfolio selection by Hall et al. (2015). The work proposed here will exploit this approach. Moreover, Birge and Maddox (1995) provide that the tardiness of a project rises linearly after a deadline is passed, but it is zero before the deadline. Tardiness is more difficult to obtain distribution information than completion time, however, since tardiness includes more information than completion time, it has stimulated many research motivations to how to derive it. In this thesis, we model our objective as a target-measured of tardiness.

## Chapter 2

# Generalized Project Network

Before introducing the generalized project network, we first consider a project with a deterministic network structure, i.e., one that can be analyzed by the classical approaches of CPM or PERT. The project can be represented as a set of tasks denoted by  $\mathcal{N}$  and a set of precedence relations  $\mathcal{A}$ . Every task  $i \in \mathcal{N}$  must be executed during the project. If  $(i, j) \in \mathcal{A}$  for some  $i, j \in \mathcal{N}$ , then task  $i$  must be completed before the start of task  $j$ . Without loss of generality, we define dummy start and end tasks  $s, t \in \mathcal{N}$ , which represent the start and completion of the project, respectively. Furthermore, for any task  $i \in \mathcal{N} \setminus \{s, t\}$ , task  $i$  can only be started after the start of the project, and the project is completed only if task  $i$  is completed, implying  $(s, i), (i, t) \in \mathcal{A}$ . For any task  $i \in \mathcal{N}$ , let  $\tilde{c}_i$  denote the uncertain time to complete task  $i$ . Given the vector  $\tilde{c}$  of  $\tilde{c}_i$  for all  $i \in \mathcal{N}$ , the project makespan, i.e., the elapsed time of the entire project, can be obtained by solving the following

integer program:

$$\max \sum_{(i,j) \in \mathcal{A}: i \neq s} \tilde{c}_i x_{ij} \quad (2.1a)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \mathcal{A}} x_{ij} = \sum_{j:(j,i) \in \mathcal{A}} x_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (2.1b)$$

$$\sum_{i:(s,i) \in \mathcal{A}} x_{si} = 1, \quad (2.1c)$$

$$\sum_{i:(i,t) \in \mathcal{A}} x_{it} = 1, \quad (2.1d)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}. \quad (2.1e)$$

If we consider the activity-on-node network for this project, i.e., the network with nodes  $\mathcal{N}$  and arcs  $\mathcal{A}$ , and set the length of any arc  $(i, j) \in \mathcal{A}$  to  $\tilde{c}_i$ , the above optimization problem finds the longest path from  $s$  to  $t$ , which corresponds to the project makespan. The binary decision variable  $x_{ij}$  denotes whether arc  $(i, j)$  is in the longest path. Constraints (2.1b)-(2.1d) ensure flow balance at each node.

In this study, we consider a generalized project network with an uncertain structure. The potential sources of uncertainty are: (a) tasks can be repeated or partially reworked due to failure or unacceptable quality, (b) new tasks not previously considered necessary can become necessary, (c) precedence requirements can be added or removed due to a better understanding of the relationships among the tasks, and (d) choice between alternative plans may be realized based on uncertain scenarios to be revealed during the project execution, which is also referred to as probabilistic branching in the literature (see, e.g., Eisner 1962, Elmaghraby 1964, GERT, and Q-GERT). We would like to model the project makespan for such a generalized project as an optimization model. Towards this end, we first

introduce two sets of tasks that may not necessarily be executed during the project.

- First, as mentioned in generalization (b), certain new tasks, denoted by set  $\mathcal{N}_N$ , could be introduced after the project starts. The necessity of executing them cannot be determined before the execution of the project. For example, for a project that is developing a new consumer electronic product, various elaborations, and improvements will become necessary in the event that a competitor company unexpectedly releases a similar product.
- Second, the aforementioned generalization (d) considers choices between alternative plans, i.e., probabilistic branching. The tasks in alternative plans, denoted by the set  $\mathcal{N}_B$ , may not be executed either. An example is shown in Figure 2.1. After completing task 1, the project takes exactly one of two alternative plans, plans 1 and 2, with equal probability, but not both. For example, task 1 may represent collecting customers' feedback on a current prototype of a product. Based on the uncertain feedback, exactly one of two alternative features, represented by plans 1 and 2, will be incorporated into the product design. At the planning phase of the project, we only know that the two plans will be followed with equal probability. Plan 1 consists of tasks 2, 3, and 4, while Plan 2 consists of tasks 5 and 6. These two plans merge at task 7. Therefore, tasks 2-6 are all in the alternative plans, i.e.,  $\{2, \dots, 6\} \subseteq \mathcal{N}_B$ .

Based on the introduction of new tasks and tasks on probabilistic branching, a task should be included in set  $\mathcal{N}$  as long as its probability to be executed is positive, implying  $\mathcal{N}_N, \mathcal{N}_B \subseteq \mathcal{N}$ . In addition, we also have dummy tasks  $s, t \in \mathcal{N}$  to include the start and end of the project.

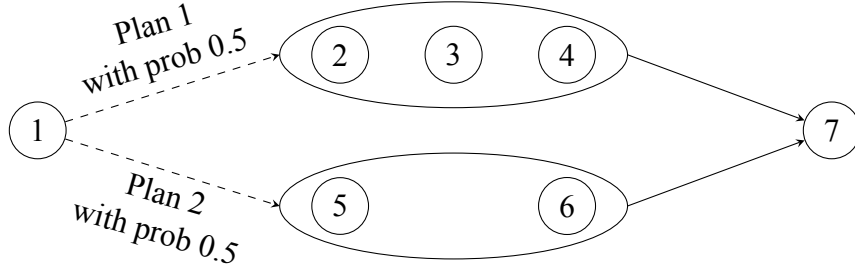


Figure 2.1: Illustration of Probabilistic Branching

Recall that model (2.1) for a project with a deterministic network considers the uncertain parameters  $\tilde{c}_i$ . To model the generalized network, we introduce another group of uncertain parameters  $\tilde{\eta}_{ij}$  associated with the precedence restrictions  $(i, j) \in \mathcal{A}$ . Using  $\tilde{c}_i$  and  $\tilde{\eta}_{ij}$ , we can address the aforementioned generalizations (a)-(d) as follows:

- (a) For generalization (a), we capture the possible repeats and reworks for each task through the uncertain time  $\tilde{c}_i$  to complete this task. Here we focus on the tasks that must be executed, i.e., the tasks in  $\mathcal{N} \setminus \mathcal{N}_N \setminus \mathcal{N}_B$ . The tasks that may be introduced during the project (i.e.,  $\mathcal{N}_N$ ) and the tasks on alternative plans (i.e.,  $\mathcal{N}_B$ ) will be further discussed when modeling generalizations (b) and (d), respectively. Consider any task  $i \in \mathcal{N} \setminus \mathcal{N}_N \setminus \mathcal{N}_B$ . We use  $\tilde{c}_i$  to represent the total time spent in task  $i$ , including possible repetitions and partial reworks, instead of the time to complete task  $i$  once. Let  $\tilde{m}_i$  denote the number of times that task  $i$  is fully executed, and let  $\tilde{n}_i$  denote the number of times that task  $i$  is partially reworked. For any  $k \in \{1, 2, \dots\}$ ,  $\tilde{c}_{i,k}^c$  and  $\tilde{c}_{i,k}^p$  represent the time required to complete the task for the  $k$ th time, where  $k = 1$  represents the first time execution, and the time required to complete the  $k$ th partial rework, respectively. Our modeling of task repetition is dif-

ferent from that in GERT and Q-GERT. Nevertheless, the distribution of  $\tilde{m}_i$  and  $\tilde{n}_i$  can be easily determined from the GERT or Q-GERT networks by simple Markov chain models, while the distributions of  $\tilde{c}_{i,k}^c$  and  $\tilde{c}_{i,k}^p$  are also specified in GERT and Q-GERT. Then, the total time spent on task  $i$  is

$$\tilde{c}_i = \sum_{k=1}^{\tilde{m}_i} \tilde{c}_{i,k}^c + \sum_{k=1}^{\tilde{n}_i} \tilde{c}_{i,k}^p. \quad (2.2)$$

- (b) Generalization (b) considers new tasks that could be potentially introduced during the project. Some of the new tasks could be part of alternative plans. For these new tasks, i.e., tasks in the set  $\mathcal{N}_N \cap \mathcal{N}_B$ , we will introduce how to model them in generalization (d). Here we consider new tasks not in alternative plans, i.e., tasks in  $\mathcal{N}_N \setminus \mathcal{N}_B$ . For any task  $i \in \mathcal{N}_N \setminus \mathcal{N}_B$ , it is sufficient to make sure that there is a positive probability to skip the task, i.e.,  $\tilde{c}_i = 0$  with a positive probability. Consider the random variables  $\tilde{m}_i$ ,  $\tilde{n}_i$ ,  $\tilde{c}_{i,k}^c$ , and  $\tilde{c}_{i,k}^p$  defined in generalization (a). We can still calculate  $\tilde{c}_i$  based on the formula in (2.2). Note that  $\tilde{c}_i = 0$  when  $\tilde{m}_i = \tilde{n}_i = 0$ . In other words, we can model the new task  $i$  by allowing  $\tilde{m}_i = \tilde{n}_i = 0$  with a positive probability.
- (c) To model generalization (c), let the set  $\mathcal{A}$  contain all possible precedence requirements among any two tasks in  $\mathcal{N}$ . A Bernoulli random variable  $\tilde{\eta}_{ij}$  is introduced for any precedence restriction  $(i, j) \in \mathcal{A}$  such that  $\tilde{\eta}_{ij} = 1$  if and only if the precedence requirement  $(i, j)$  must be satisfied, i.e., task  $j$  can only start after task  $i$  completes. The change in the precedence requirement can then be modeled by  $\tilde{\eta}_{ij}$ . In addition, to ensure that all the tasks not in



alternative plans will be considered in a path from  $s$  to  $t$ , for all  $i \notin \mathcal{N}_B$ , we set  $(s, i), (i, t) \in \mathcal{A}$  with  $P\{\tilde{\eta}_{si} = 1\} = P\{\tilde{\eta}_{it} = 1\} = 1$ .

- (d) For generalization (d), first, we determine the marginal distribution of  $\tilde{c}_i$  for tasks in alternative plans resulted from probabilistic branching. For any task  $i \in \mathcal{N}_B$ , we can define  $\tilde{c}_i$  in a way similar to generalization (b). As in generalization (b), we also let  $\tilde{m}_i = \tilde{n}_i = 0$  with a positive probability if task  $i$  could be a new task introduced during the project, i.e.,  $i \in \mathcal{N}_N \cap \mathcal{N}_B$ . The only difference is that  $\tilde{m}_i$  and  $\tilde{n}_i$  now represent the numbers of executions and partial reworks, respectively, given that an alternative plan including task  $i$  is executed. For example, for the task  $i \in \{2, 3, 4\}$  in Figure 2.1,  $\tilde{m}_i$  and  $\tilde{n}_i$  are the times of executions and partial reworks under the condition that plan 1 is chosen.

Second, we use the Bernoulli random variables  $\tilde{\eta}_{ij}$  to model probabilistic branching. For example, consider Figure 2.1. Tasks 2-4 will be executed only if plan 1 is chosen. Thus, we can introduce the random variables  $\tilde{\eta}_{12}, \tilde{\eta}_{13}, \tilde{\eta}_{14}$  such that  $P\{\tilde{\eta}_{12} = \tilde{\eta}_{13} = \tilde{\eta}_{14} = 1\} = 1 - P\{\tilde{\eta}_{12} = \tilde{\eta}_{13} = \tilde{\eta}_{14} = 0\} = 0.5$ . Similarly,  $P\{\tilde{\eta}_{15} = \tilde{\eta}_{16} = 1\} = 1 - P\{\tilde{\eta}_{15} = \tilde{\eta}_{16} = 0\} = 0.5$ . We have  $\tilde{\eta}_{1i} + \tilde{\eta}_{1j} = 1$  for any  $i \in \{2, 3, 4\}$  and  $j \in \{5, 6\}$  to ensure that exactly one of plans 1 and 2 will be selected. Moreover, if either plan 1 or plan 2 is executed, the tasks in the chosen plan must be performed before task 7. Therefore, we also have the precedence requirements  $(i, 7)$  with  $\tilde{\eta}_{i7} = 1$  for all  $i \in \{2, \dots, 6\}$ . Now, consider the general case. Let  $\mathcal{N}_{B_s} \subseteq \mathcal{N}$  denote the set of tasks after which we need to choose among several alternative plans. For example, we have  $1 \in \mathcal{N}_{B_s}$  in Figure 2.1. For any  $i \in \mathcal{N}_{B_s}$ , let  $g_i$  denote

the number of alternative plans after task  $i$ , e.g.,  $g_1 = 2$  in Figure 2.1. For any  $k \in \{1, \dots, g_i\}$ , plan  $k$  after task  $i$  consists of tasks in the set  $G_i^k$ . In Figure 2.1, we can set  $G_1^1 = \{2, 3, 4\}$  and  $G_1^2 = \{5, 6\}$ . For all  $j \in G_i^k$ , we have  $(i, j) \in \mathcal{A}$  and  $\mathbb{P}\{\tilde{\eta}_{ij} = 1\} > 0$  corresponds to the probability that the alternative plan  $G_i^k$  will be chosen after finishing task  $i$ . Thus, we have  $\tilde{\eta}_{ij} = \tilde{\eta}_{ij'}$  for all  $j, j' \in G_i^k$ . Also note that  $\sum_{k=1}^{g_i} \tilde{\eta}_{ij_k} = 1$  for all  $j_k \in G_i^k$  since exactly one alternative plan will be chosen. Let  $t_i$  denote the task to be executed after completing any one of the alternative plans following task  $i$ , i.e., tasks in  $G_i^k$  for any  $k \in \{1, \dots, g_i\}$ . Here we have  $t_1 = 7$  in Figure 2.1. As the tasks in alternative plans must be executed before  $t_i$ , we have the precedence requirement  $(j, t_i) \in \mathcal{A}$  with  $\tilde{\eta}_{jt_i} = 1$  for all  $k \in \{1, \dots, g_i\}$  and  $j \in \cup_{k=1}^{g_i} G_i^k$ .

Let  $\tilde{\boldsymbol{\eta}}$  and  $\tilde{\boldsymbol{c}}$  denote the random vectors of  $\tilde{\eta}_{ij}$  and  $\tilde{c}_i$ , respectively. The project makespan can be written as the following model, for given  $\tilde{\boldsymbol{\eta}}$  and  $\tilde{\boldsymbol{c}}$ :

$$Z(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{c}}) = \max \sum_{(i,j) \in \mathcal{A}: i \neq s} \tilde{c}_i \tilde{\eta}_{ij} x_{ij} \quad (2.3a)$$

$$\text{s.t.} \quad \sum_{j: (i,j) \in \mathcal{A}} \tilde{\eta}_{ij} x_{ij} = \sum_{j: (j,i) \in \mathcal{A}} \tilde{\eta}_{ji} x_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (2.3b)$$

$$(2.1c), (2.1d), (2.1e). \quad (2.3c)$$

For many projects, there exists a target project completion time of  $T$ . We are particularly interested in the expected project tardiness defined as  $\mathbb{E}[(Z(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{c}}) - T)^+]$ , which measures the expected delay if the project cannot be completed within the target time  $T$ . Motivated by typical contract penalties, and used earlier by Bertsimas et al. (2004, 2006), this is a standard measure of project performance.

If  $T = 0$ , then the expected tardiness and expected makespan are equivalent. In the following sections, we apply the method of distributionally robust optimization to provide estimates of  $E[(Z(\tilde{\eta}, \tilde{c}) - T)^+]$ .

# Chapter 3

## Formulation and Solution Procedure

### 3.1 Marginal Distribution Model for $\tilde{c}$

In this section, we consider the marginal distribution model for  $\tilde{c}$ , which captures the uncertainties in generalizations (a) and (b). For any task  $i \in \mathcal{N}$ , the time  $\tilde{c}_i$  required to complete task  $i$  is assumed to be a discrete random variable with a finite support  $\mathcal{C}_i$  and a probability mass function  $p_i(\cdot)$ . Then, the set of all possible joint distributions of  $\tilde{c}$  can be written as

$$\Theta_c = \{\theta_c : P_{\theta_c}\{\tilde{c}_i = c\} = p_i(c) \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i\},$$

where  $P_{\theta_c}\{\cdot\}$  represents the probability under the distribution  $\theta_c$ . Given the vector  $\tilde{\eta}$ , the worst-case expected tardiness among all distributions of  $\tilde{c}$  in the set  $\Theta_c$  is

$$Z^*(\tilde{\eta}) = \max_{\theta_c \in \Theta_c} E_{\theta_c}[(Z(\tilde{\eta}, \tilde{c}) - T)^+], \quad (3.1)$$

where  $E_{\theta_c}[\cdot]$  denotes the expectation taken with respect to the distribution  $\theta_c$ . We refer to model (3.1) as the marginal distribution model since it only uses the marginal distributions of  $\tilde{c}$ .

To solve model (3.1), we first consider an arbitrary binary integer linear program with the uncertain objective vector  $\tilde{c}$  defined as

$$\mathcal{Z}(\tilde{c}) = \max\{\tilde{c}^\top \mathbf{y} : \mathbf{y} \in \mathcal{Y}\}, \quad (3.2)$$

where  $\mathcal{Y} \subseteq \{0, 1\}^{|\mathcal{N} \setminus \{s, t\}|}$ . Given the ambiguity set  $\Theta_c$  for the distribution of  $\tilde{c}$ , we can define the following distributionally robust model for  $\mathcal{Z}(\tilde{c})$ :

$$\mathcal{Z}^* = \max_{\theta \in \Theta_c} E_\theta[(\mathcal{Z}(\tilde{c}) - T)^+]. \quad (3.3)$$

In the following theorem, we represent  $\mathcal{Z}^*$  as a linear program with a potentially exponential number of decision variables.

**Theorem 1.**  *$\mathcal{Z}^*$  is equivalent to the following linear problem:*

$$\bar{\mathcal{Z}} = \max \quad \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} cq_i(c) - \sum_{\mathbf{y} \in \mathcal{Y}} T\lambda(\mathbf{y}) \quad (3.4a)$$

$$s.t. \quad q_i(c) \leq p_i(c) \quad \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i, \quad (3.4b)$$

$$\sum_{c \in \mathcal{C}_i} q_i(c) - \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda(\mathbf{y}) = 0 \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (3.4c)$$

$$\sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y}) \leq 1, \quad (3.4d)$$

$$q_i(c) \geq 0 \quad \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i, \quad (3.4e)$$

$$\lambda(\mathbf{y}) \geq 0 \quad \forall \mathbf{y} \in \mathcal{Y}, \quad (3.4f)$$

where the decision variables  $\lambda(\mathbf{y})$  and  $q_i(c)$  can be interpreted as the following probabilities:

$$\begin{aligned}\lambda(\mathbf{y}) &= \text{P}\{\mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \forall \mathbf{y} \in \mathcal{Y}, \\ q_i(c) &= \\ &\text{P}\{\mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \tilde{c}_i = c, \text{the } i\text{th component of } \mathbf{y}^*(\tilde{\mathbf{c}}) \text{ is } 1\} \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i.\end{aligned}\tag{3.5}$$

*Proof.* Consider the optimization problem  $\mathcal{Z}(\tilde{\mathbf{c}})$  defined in (3.2). For any given  $\tilde{\mathbf{c}}$ , let  $\mathbf{y}^*(\tilde{\mathbf{c}})$  be the corresponding optimal solution with ties broken by lexicographic order. First conditioning on the value of  $\mathcal{Z}^*(\tilde{\mathbf{c}})$  and then conditioning on the optimal solution  $\mathbf{y}^*(\tilde{\mathbf{c}})$ , given  $\lambda(\mathbf{y})$  defined in (3.5), we have

$$\begin{aligned}\text{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] &= \text{P}(\mathcal{Z}^*\{\tilde{\mathbf{c}}\} \geq T) \text{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T] \\ &\quad + \text{P}\{\mathcal{Z}^*(\tilde{\mathbf{c}}) < T\} \text{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) < T] \\ &= \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y}) \text{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}] \\ &\quad + \left(1 - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y})\right) \text{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) < T].\end{aligned}$$

Note that

$$\begin{aligned}
& \mathbb{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}] \\
&= \mathbb{E}[\mathcal{Z}^*(\tilde{\mathbf{c}}) - T \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}] \\
&= \mathbb{E}[\tilde{\mathbf{c}}^T \mathbf{y} \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}] - T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \mathbb{E}[\tilde{c}_i \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}] \cdot y_i - T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}} c \cdot \mathbb{P}\{\tilde{c}_i = c \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \cdot y_i - T
\end{aligned}$$

and  $\mathbb{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+ \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) < T] = 0$ . Therefore,

$$\begin{aligned}
& \mathbb{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] \\
&= \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y}) \left( \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} c \cdot \mathbb{P}\{\tilde{c}_i = c \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \cdot y_i - T \right).
\end{aligned}$$

Rearranging the terms yields

$$\begin{aligned}
& \mathbb{E}[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{y} \in \mathcal{Y}} c \cdot \lambda(\mathbf{y}) \cdot \mathbb{P}\{\tilde{c}_i = c \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \cdot y_i - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y}) T.
\end{aligned} \tag{3.6}$$

According to the definition of  $\lambda(\mathbf{y})$ , we have

$$\lambda(\mathbf{y}) \cdot \mathbb{P}\{\tilde{c}_i = c \mid \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} = \mathbb{P}\{\tilde{c}_i = c, \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\}$$

and so

$$\begin{aligned}
& E[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{y} \in \mathcal{Y}} c \cdot \mathbb{P}\{\tilde{c}_i = c, \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \cdot y_i - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y})T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} c \cdot \mathbb{P}\{\tilde{c}_i = c, \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y})T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} c \left( \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \mathbb{P}\{\tilde{c}_i = c, \mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \mathbf{y}^*(\tilde{\mathbf{c}}) = \mathbf{y}\} \right) - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y})T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} cq_i(c) - \sum_{\mathbf{y} \in \mathcal{Y}} T\lambda(\mathbf{y}),
\end{aligned}$$

where the second and last equalities are obtained from  $y_i \in \{0, 1\}$  for all  $\mathbf{y} \in \mathcal{Y}$  and the definition of  $q_i(c)$  in (3.5), respectively. According to the definitions of  $\lambda(\mathbf{y})$  and  $q_i(c)$  in (3.5), we have  $\lambda(\mathbf{y}) \geq 0$ ,  $\sum_{\mathbf{y} \in \mathcal{Y}} \lambda(\mathbf{y}) \leq 1$ ,  $0 \leq q_i(c) \leq p_i(c)$ , and

$$\sum_{c \in \mathcal{C}_i} q_i(c) = \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda(\mathbf{y}) = \mathbb{P}\{\mathcal{Z}^*(\tilde{\mathbf{c}}) \geq T, \text{the } i\text{th component of } \mathbf{y}^*(\tilde{\mathbf{c}}) \text{ is } 1\}.$$

Therefore, we obtain  $\mathcal{Z}^* \leq \bar{\mathcal{Z}}$ .

Next, we show  $\mathcal{Z}^* \geq \bar{\mathcal{Z}}$ . Let  $(q_i^*(c), \lambda^*(\mathbf{y}))$  denote an optimal solution to (3.4).

We can construct a distribution  $\theta$  for  $\tilde{\mathbf{c}}$  as follows:

- For any  $\mathbf{y} \in \mathcal{Y}$ , choose  $\mathbf{y}$  with probability  $\lambda^*(\mathbf{y})$ . For any  $i$  such that  $y_i = 1$ , generate  $\tilde{c}_i \sim q_i^*(c) / \sum_{c' \in \mathcal{C}_i} q_i^*(c')$ . For any  $i$  such that  $y_i = 0$ , generate  $\tilde{c}_i \sim (p_i - q_i^*(c)) / (1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c'))$ .
- With probability  $1 - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y})$ , for any  $i \in \mathcal{N} \setminus \{s, t\}$ , generate  $\tilde{c}_i \sim (p_i - q_i^*(c)) / (1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c'))$ .



As  $0 \leq q_i^*(c) \leq p_i(c)$  and  $\sum_{c \in \mathcal{C}_i} p_i(c) = 1$ , it is straightforward that  $q_i^*(c) / \sum_{c' \in \mathcal{C}_i} q_i^*(c')$  and  $(p_i - q_i^*(c)) / (1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c'))$  are valid probability mass functions. Under this distribution  $\theta$ , for any  $i \in \mathcal{N} \setminus \{s, t\}$ , we have

$$\begin{aligned}
P_\theta(\tilde{c}_i = c) &= \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda^*(\mathbf{y}) \frac{q_i^*(c)}{\sum_{c' \in \mathcal{C}_i} q_i^*(c')} + \sum_{\mathbf{y} \in \mathcal{Y}: y_i=0} \lambda^*(\mathbf{y}) \frac{p_i(c) - q_i^*(c)}{1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c')} \\
&\quad + \left(1 - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y})\right) \frac{p_i(c) - q_i^*(c)}{1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c')} \\
&= \frac{q_i^*(c)}{\sum_{c' \in \mathcal{C}_i} q_i^*(c')} \left( \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda^*(\mathbf{y}) \right) \\
&\quad + \frac{p_i(c) - q_i^*(c)}{1 - \sum_{c' \in \mathcal{C}_i} q_i^*(c')} \left(1 - \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda^*(\mathbf{y})\right) \\
&= p_i(c),
\end{aligned}$$

where the last equality follows from (3.4c). Thus, we have  $\theta \in \Theta_c$  and hence

$$\mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{c}) - T)^+] \leq \mathcal{Z}^*. \quad (3.7)$$

Also, note that

$$\begin{aligned}
&\mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{c}) - T)^+] \\
&= \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) \mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{c}) - T)^+ \mid \mathbf{y} \text{ is chosen}] \\
&\quad + \left(1 - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y})\right) \mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{c}) - T)^+ \mid \text{none of } \mathbf{y} \in \mathcal{Y} \text{ is chosen}] \\
&\geq \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) \mathbb{E}_\theta[\mathcal{Z}^*(\tilde{c}) - T \mid \mathbf{y} \text{ is chosen}] \geq \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) \mathbb{E}_\theta[\tilde{c}^\top \mathbf{y} - T \mid \mathbf{y} \text{ is chosen}].
\end{aligned}$$

Similarly to the derivation of (3.6), we have

$$\begin{aligned}
\mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] &\geq \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) \mathbb{E}_\theta[\tilde{\mathbf{c}}^\top \mathbf{y} - T \mid \mathbf{y} \text{ is chosen}] \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{y} \in \mathcal{Y}} c \cdot \lambda^*(\mathbf{y}) \cdot \mathbb{P}_\theta\{\tilde{c}_i = c \mid \mathbf{y} \text{ is chosen}\} \cdot y_i \\
&\quad - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) T.
\end{aligned}$$

Applying  $y_i \in \{0, 1\}$  for any  $\mathbf{y} \in \mathcal{Y}$  and the construction of  $\theta$ , we have

$$\begin{aligned}
\mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] &\geq \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} c \cdot \lambda^*(\mathbf{y}) \cdot \frac{q_i^*(c)}{\sum_{c' \in \mathcal{C}_i} q_i^*(c')} - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} c \cdot \left( \sum_{\mathbf{y} \in \mathcal{Y}: y_i=1} \lambda^*(\mathbf{y}) \right) \cdot \frac{q_i^*(c)}{\sum_{c' \in \mathcal{C}_i} q_i^*(c')} \\
&\quad - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) T \\
&= \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} c q_i^*(c) - \sum_{\mathbf{y} \in \mathcal{Y}} \lambda^*(\mathbf{y}) T \\
&= \bar{\mathcal{Z}},
\end{aligned}$$

where the second last equality and the last equality follow from (3.4c) and (3.4a), respectively. According to (3.7), we obtain  $\bar{\mathcal{Z}} \leq \mathbb{E}_\theta[(\mathcal{Z}^*(\tilde{\mathbf{c}}) - T)^+] \leq \mathcal{Z}^*$ , which yields  $\mathcal{Z}^* = \bar{\mathcal{Z}}$  by combining with  $\mathcal{Z}^* \leq \bar{\mathcal{Z}}$ .  $\square$

In Theorem 1,  $\lambda(\mathbf{y})$  represents the probability that a feasible solution  $\mathbf{y}$  to model (3.2) is optimal and its optimal value is at least  $T$ .  $q_i^*(c)$  is the probability that (i) the optimal value of model (3.2) is at least  $T$ , (ii)  $\tilde{c}_i$  is equal to  $c_i$ , and (iii) the  $i$ th component in the optimal solution to (3.2) is 1. An optimal solution to model (3.4) then corresponds to these probabilities under the worst-case distribution in

the distributionally robust model (3.3). In the proof of Theorem 1, we also show how to generate the worst-case distribution based on the optimal solution to (3.4). Also, note that Theorem 1 can be interpreted as the dual form of the result obtained in Meilijson and Nadas (1979). Our result is derived based on the probability that an arbitrary solution to (3.2) is optimal, which is different from the approach adopted by Meilijson and Nadas (1979).

Next, we apply Theorem 1 to the project management model (3.1) and obtain an equivalent linear programming formulation for model (3.1).

**Theorem 2.**  $Z^*(\tilde{\eta})$  can be computed by solving the following linear program:

$$\min_{u,v,w,z(\cdot)} \sum_{i \in \mathcal{N} \setminus \{s,t\}} \sum_{c \in \mathcal{C}_i} p_i(c) z_i(c) + u \quad (3.8a)$$

$$s.t. \quad z_i(c) + v_i \geq c \quad \forall i \in \mathcal{N} \setminus \{s,t\}, c \in \mathcal{C}_i, \quad (3.8b)$$

$$w_s - w_t \leq T + u, \quad (3.8c)$$

$$\tilde{\eta}_{ij}(w_i - w_j) \geq \tilde{\eta}_{ij} v_i \quad \forall (i,j) \in \mathcal{A}, i,j \in \mathcal{N} \setminus \{s,t\}, \quad (3.8d)$$

$$w_s - w_i \geq 0 \quad \forall (s,i) \in \mathcal{A}, \quad (3.8e)$$

$$w_i - w_t \geq v_i \quad \forall (i,t) \in \mathcal{A}, \quad (3.8f)$$

$$z_i(c) \geq 0 \quad \forall i \in \mathcal{N} \setminus \{s,t\}, c \in \mathcal{C}_i, \quad (3.8g)$$

$$u \geq 0. \quad (3.8h)$$

*Proof.* Consider the linear program (3.4). Let  $z_i(c)$ ,  $v_i$ , and  $u$  be the dual variables corresponding to constraints (3.4b), (3.4c), and (3.4d), respectively. The dual form

of (3.4) is written as

$$Z^* = \min_{u, \mathbf{v}, \mathbf{z}(\cdot)} \left\{ \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} p_i(c) z_i(c) + u \left| \begin{array}{l} \sum_{i: y_i=1} v_i \leq T + u \quad \forall \mathbf{y} \in \mathcal{Y}, \\ (3.8b), (3.8g), (3.8h) \end{array} \right. \right\}. \quad (3.9)$$

If we let  $y_i = \sum_{j: (i,j) \in \mathcal{A}} \tilde{\eta}_{ij} x_{ij} = \sum_{j: (j,i) \in \mathcal{A}} \tilde{\eta}_{ji} x_{ji}$  for all  $i \in \mathcal{N} \setminus \{s, t\}$ , model  $Z(\tilde{\boldsymbol{\eta}}, \tilde{\mathbf{c}})$  with given  $\tilde{\boldsymbol{\eta}}$  can be written as

$$Z(\tilde{\boldsymbol{\eta}}, \tilde{\mathbf{c}}) = \max\{\tilde{\mathbf{c}}^T \mathbf{y} : \mathbf{y} \in \mathcal{Y}(\tilde{\boldsymbol{\eta}})\}, \quad (3.10)$$

where

$$\mathcal{Y}(\tilde{\boldsymbol{\eta}}) = \left\{ \mathbf{y} \in \{0, 1\}^{|\mathcal{N} \setminus \{s, t\}|} \left| \begin{array}{l} \sum_{j: (i,j) \in \mathcal{A}} \tilde{\eta}_{ij} x_{ij} \\ = \sum_{j: (j,i) \in \mathcal{A}} \tilde{\eta}_{ji} x_{ji} = y_i \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \\ (2.1c), (2.1d), (2.1e) \end{array} \right. \right\}.$$

Applying the general result in (3.9) to  $Z^*(\tilde{\boldsymbol{\eta}})$  with  $Z(\tilde{\boldsymbol{\eta}}, \tilde{\mathbf{c}})$  formulated in (3.10),  $Z^*(\tilde{\boldsymbol{\eta}})$  is equivalent to the following model:

$$Z^*(\tilde{\boldsymbol{\eta}}) = \min_{u, \mathbf{v}, \mathbf{z}(\cdot)} \left\{ \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} p_i(c) z_i(c) + u \left| \begin{array}{l} Z(\tilde{\boldsymbol{\eta}}, \mathbf{v}) \leq T + u, \\ (3.8b), (3.8g), (3.8h) \end{array} \right. \right\}. \quad (3.11)$$

Consider the formulation of  $Z(\tilde{\boldsymbol{\eta}}, \tilde{\mathbf{c}})$  in (2.3). As (2.3) is the longest path problem, we can replace the binary constraints  $x_{ij} \in \{0, 1\}$  with  $x_{ij} \geq 0$  for all  $(i, j) \in \mathcal{A}$ ,

i.e.,

$$Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v}) = \max \left\{ \sum_{(i,j) \in \mathcal{A}} v_i \tilde{\eta}_{ij} x_{ij} : (2.3b), (2.1c), (2.1d), x_{ij} \geq 0 \forall (i,j) \in \mathcal{A} \right\}.$$

Let  $w_i$ ,  $w_s$ , and  $w_t$  be the dual variables corresponding to constraints (2.3b), (2.1c), (2.1d), respectively. The dual form of  $Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v})$  is formulated as

$$Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v}) = \min \{w_s - w_t : (3.8d), (3.8e), (3.8f)\}.$$

Next we complete the proof by showing that (3.8) is equivalent to  $Z^*(\tilde{\boldsymbol{\eta}})$  formulated in (3.11). Note that these two models have the same objective function. It suffices to show that their feasible regions are equivalent. First, consider any  $(u, \boldsymbol{v}, \boldsymbol{z}(\cdot))$  feasible to (3.11). There exists a dual optimal solution  $\boldsymbol{w}^*$  to  $Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v})$  such that  $w_s^* - w_t^* = Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v}) \leq T + u$ . Thus,  $(u, \boldsymbol{v}, \boldsymbol{w}^*, \boldsymbol{z}(\cdot))$  is feasible to (3.8). Now consider any  $(u, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{z}(\cdot))$  feasible to (3.8). As  $\boldsymbol{w}$  is a feasible dual solution to  $Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v})$ , we have  $Z(\tilde{\boldsymbol{\eta}}, \boldsymbol{v}) \leq w_s - w_t \leq T + u$ . Therefore,  $(u, \boldsymbol{v}, \boldsymbol{z}(\cdot))$  is feasible to (3.11), which completes the proof.  $\square$

## 3.2 Distributionally Robust Model for $\tilde{\boldsymbol{c}}$ and $\tilde{\boldsymbol{\eta}}$

In this section, we further consider the uncertain precedence restrictions  $\tilde{\boldsymbol{\eta}}$  in the distributionally robust optimization model so that generalizations (c) and (d) can also be incorporated when evaluating the project makespan. Recall that  $\tilde{\eta}_{ij}$  for any  $(i, j) \in \mathcal{A}$  is the Bernoulli random variable such that task  $i$  should be completed before the start of  $j$  if and only if  $\tilde{\eta}_{ij} = 1$ . Here we assume that the marginal

distribution of  $\tilde{\eta}_{ij}$  is known, which can be easily specified by the parameter  $\bar{\eta}_{ij}$  defined as  $\bar{\eta}_{ij} = \text{P}\{\tilde{\eta}_{ij} = 1\}$ .

According to the description of generalization (d), we can further identify some information regarding the joint distribution of  $\tilde{\eta}$ . Recall that  $\mathcal{N}_{B_s}$  denotes the set of tasks after which alternative plans will be considered. For any  $i \in \mathcal{N}_{B_s}$ ,  $g_i$  is the number of alternative plans after task  $i$  and, for any  $k \in \{1, \dots, g_i\}$ ,  $G_i^k$  is the set of tasks in the  $k$ th alternative plan after task  $i$ . Observe that any task  $j \in G_i^k$  is executed if and only if the alternative plan  $G_i^k$  after task  $i \in \mathcal{N}_{B_s}$  is selected. Therefore, for the  $k$ th alternative plan  $G_i^k$ , we can use an arbitrary task denoted by  $j_i^k \in G_i^k$  to represent whether the alternative plan  $G_i^k$  is selected. In other words,  $G_i^k$  is selected if and only if  $\tilde{\eta}_{ij_i^k} = 1$  and the probability of selecting  $G_i^k$  is  $\text{P}\{\tilde{\eta}_{ij_i^k} = 1\} = \bar{\eta}_{ij_i^k}$ . This implies that  $\tilde{\eta}_{ij} = \tilde{\eta}_{ij_i^k}$  and hence  $\bar{\eta}_{ij} = \text{P}\{\tilde{\eta}_{ij} = 1\} = \text{P}\{\tilde{\eta}_{ij_i^k} = 1\} = \bar{\eta}_{ij_i^k}$  for all  $j \in G_i^k$ . Furthermore, as exactly one of the alternatives after  $i$  will be executed, we have  $\sum_{k=1}^{g_i} \tilde{\eta}_{ij_i^k} = 1$  with probability 1, which also implies  $\sum_{k=1}^{g_i} \bar{\eta}_{ij_i^k} = 1$ .

Additionally, let  $\mathcal{A}_f$  be a subset of the set of precedence relations  $\mathcal{A}$  such that

$$\mathcal{A}_f = \{(i, j) \in \mathcal{A} : \bar{\eta}_{ij} = 1\}, \quad (3.12)$$

i.e.,  $\tilde{\eta}_{ij} = 1$  with probability 1 for any  $(i, j) \in \mathcal{A}_f$ . Section 2 shows that  $\mathcal{A}_f$  at least includes the following precedence relations. First, for any task not in the alternatives plans, it must be executed and so it is linked with the dummy start and end tasks, i.e.,  $s$  and  $t$ . In other words, we have  $\tilde{\eta}_{si} = \tilde{\eta}_{ti} = 1$  for any  $i \in \mathcal{N} \setminus \{s, t\} \setminus \mathcal{N}_B$ , where  $\mathcal{N}_B$ , as defined in Section 2, represents the tasks in alternative plans and satisfies  $\mathcal{N}_B = \cup_{i \in \mathcal{N}_{B_s}} \cup_{k=1}^{g_i} G_i^k$ . Second, for any task  $i \in \mathcal{N}_{B_s}$  after which

probabilistic branching will occur, we use  $t_i$  to denote the task to be executed after completing any selected alternative plan after task  $i$ . The tasks in the alternative plans after  $i$ , if executed, must be completed before the initiation of  $t_i$ . This yields  $\tilde{\eta}_{jt_i} = 1$  for all  $j$  in any alternative plan after  $i$ .

Putting together all these restrictions, we obtain that the support of  $\tilde{\eta}$  should be contained in the following set  $\Omega$ :

$$\Omega = \left\{ \boldsymbol{\eta} \in \{0, 1\}^{|\mathcal{A}|} \left| \begin{array}{ll} \sum_{k=1}^{g_i} \eta_{ij^k} = 1, & \forall i \in \mathcal{N}_{B_s}, \\ \eta_{ij} = \eta_{ij^k}, & \forall i \in \mathcal{N}_{B_s}, k \in \{1, \dots, g_i\}, \\ & j \in G_i^k \setminus \{j_i^k\}, \\ \eta_{ij} = 1 & \forall (i, j) \in \mathcal{A}_f \end{array} \right. \right\}. \quad (3.13)$$

Combining with the marginal distribution of  $\tilde{\eta}$  specified by the parameters  $\bar{\eta}_{ij}$ , the joint distribution of  $\tilde{\eta}_{ij}$  should be contained within the following set:

$$\Theta_\eta = \left\{ \theta_\eta : \sum_{\boldsymbol{\eta} \in \Omega} \mathbb{P}_{\theta_\eta} \{ \tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} \} = 1, \mathbb{P}_{\theta_\eta} \{ \tilde{\eta}_{ij} = 1 \} = \bar{\eta}_{ij} \forall (i, j) \in \mathcal{A} \right\}.$$

Without further information regarding the joint distribution of  $\tilde{\boldsymbol{c}}$  and  $\tilde{\boldsymbol{\eta}}$ , we can obtain an upper bound on the expected tardiness by solving the following distributionally robust optimization model:

$$\max_{\theta_\eta \in \Theta_\eta} \mathbb{E}_{\theta_\eta} [Z^*(\tilde{\boldsymbol{\eta}})] = \max_{\theta_\eta \in \Theta_\eta} \mathbb{E}_{\theta_\eta} \left[ \max_{\theta_c \in \Theta_c} \mathbb{E}_{\theta_c} [(Z(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{c}}) - T)^+] \right]. \quad (3.14)$$

For any  $\boldsymbol{\eta} \in \Omega$ , let  $p(\boldsymbol{\eta}) = \mathbb{P}\{\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\}$ . Model (3.14) can be formulated as

$$\max \sum_{\boldsymbol{\eta} \in \Omega} Z^*(\boldsymbol{\eta})p(\boldsymbol{\eta}) \quad (3.15a)$$

$$\text{s.t.} \quad \sum_{\boldsymbol{\eta} \in \Omega: \eta_{ij}=1} p(\boldsymbol{\eta}) = \bar{\eta}_{ij} \quad \forall (i, j) \in \mathcal{A}, \quad (3.15b)$$

$$\sum_{\boldsymbol{\eta} \in \Omega} p(\boldsymbol{\eta}) = 1, \quad (3.15c)$$

$$p(\boldsymbol{\eta}) \geq 0 \quad \forall \boldsymbol{\eta} \in \Omega. \quad (3.15d)$$

Model (3.15) is a linear program with  $|\Omega|$  decision variables, whose number could be exponential in  $|\mathcal{A}|$ . To tackle the numerical challenge, we present in Algorithm 1 a column generation framework to solve the linear program.

To execute Algorithm 1, we first need to find an initial subset  $\Omega'$  of  $\Omega$  that ensures the feasibility of the restricted model (3.16). Furthermore, the pricing problem (3.17) must be solved for each column generation iteration. These two issues are addressed in the following two subsections, respectively.

### 3.2.1 Find the Initial Subset $\Omega'$

In this subsection, we propose a low polynomial time algorithm to obtain  $\Omega' \subseteq \Omega$  such that the restricted model (3.16) based on  $\Omega'$  is feasible.

Recall that we have defined in a subset  $\mathcal{A}_f$  of  $\mathcal{A}$  in (3.12), which corresponds to the precedence relations that must be satisfied. For any  $i \in \mathcal{N}_{B_s}$ , let  $\mathcal{A}_b^i = \{(i, j^k) : k \in \{1, \dots, g_i\}\}$  and  $\mathcal{A}_f^i = \{(i, j) : j \in \cup_{k=1}^{g_i} G_i^k \setminus \{j^k\}\}$ , which corresponds to the precedence relations between the node in  $\mathcal{N}_{B_s}$  and the node in  $\mathcal{N}_B$ . Furthermore, let  $\mathcal{A}_p = \mathcal{A} \setminus \mathcal{A}_f \setminus (\cup_{i \in \mathcal{N}_{B_s}} (\mathcal{A}_b^i \cup \mathcal{A}_f^i))$ , which corresponds to the



---

**Algorithm 1:** Column Generation Framework
 

---

1 Consider  $\Omega' \subseteq \Omega$  such that the following restricted model is feasible:

$$\max \sum_{\boldsymbol{\eta} \in \Omega'} Z^*(\boldsymbol{\eta})p(\boldsymbol{\eta}), \quad (3.16a)$$

$$\text{s.t.} \quad \sum_{\boldsymbol{\eta} \in \Omega': \eta_{ij}=1} p(\boldsymbol{\eta}) = \bar{\eta}_{ij} \quad \forall (i, j) \in \mathcal{A}, \quad (3.16b)$$

$$\sum_{\boldsymbol{\eta} \in \Omega'} p(\boldsymbol{\eta}) = 1, \quad (3.16c)$$

$$p(\boldsymbol{\eta}) \geq 0 \quad \forall \boldsymbol{\eta} \in \Omega'. \quad (3.16d)$$

- 2 Solve the restricted model (3.16) to obtain the primal optimal solution ( $p^*(\boldsymbol{\eta}) \forall \boldsymbol{\eta} \in \Omega'$ ) and the dual optimal solution ( $\boldsymbol{\alpha}^*, \beta^*$ ), where  $\boldsymbol{\alpha}^*$  and  $\beta^*$  correspond to the optimal dual variables for constraints (3.16b) and (3.16c), respectively.
- 3 Consider the following pricing problem:

$$\max_{\boldsymbol{\eta} \in \Omega} \left\{ Z^*(\boldsymbol{\eta}) - \sum_{(i,j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \right\}. \quad (3.17)$$

Let  $\boldsymbol{\eta}^*$  and  $\mathcal{P}^*$  denote its optimal solution and optimal value respectively. Given the dual variable ( $\boldsymbol{\alpha}^*, \beta^*$ ) for the original model (3.15),  $\mathcal{P}^* - \beta^*$  is the maximum reduced cost for all decision variables  $p(\boldsymbol{\eta})$  where  $\boldsymbol{\eta} \in \Omega$ .

- 4 If  $\mathcal{P}^* - \beta^* \leq 0$ , i.e.,  $p(\boldsymbol{\eta})$  for all  $\boldsymbol{\eta} \in \Omega$  has a non-positive reduced cost, an optimal solution ( $p^*(\boldsymbol{\eta}), \forall \boldsymbol{\eta} \in \Omega$ ) to the original problem (3.15) can be obtained from the current optimal solution ( $p^*(\boldsymbol{\eta}) \forall \boldsymbol{\eta} \in \Omega'$ ) to the restricted model (3.16) by setting  $p^*(\boldsymbol{\eta}) = 0$  for all  $\boldsymbol{\eta} \in \Omega \setminus \Omega'$ . Otherwise, i.e.,  $\mathcal{P}^* - \beta^* > 0$ , let  $\Omega' \leftarrow \Omega' \cup \{\boldsymbol{\eta}^*\}$  and go to Step 2.
-

precedence relations with the positive probability of being unsatisfied in the real project. Obviously,  $\mathcal{A}_f$ ,  $\mathcal{A}_p$ ,  $\mathcal{A}_b^i$  and  $\mathcal{A}_f^i$  for all  $i \in \mathcal{N}_{B_s}$  correspond to a partition of  $\mathcal{A}$ . Therefore, for any  $\boldsymbol{\eta} \in \Omega$ , we can decompose it into subvectors  $\boldsymbol{\eta}(\mathcal{A}_f)$ ,  $\boldsymbol{\eta}(\mathcal{A}_p)$ ,  $\boldsymbol{\eta}(\mathcal{A}_b^i)$  and  $\boldsymbol{\eta}(\mathcal{A}_f^i)$  for all  $i \in \mathcal{N}_{B_s}$ , each of which represent the entries in  $\boldsymbol{\eta}$  corresponding to  $\mathcal{A}_f$ ,  $\mathcal{A}_p$ ,  $\mathcal{A}_b^i$  and  $\mathcal{A}_f^i$  for all  $i \in \mathcal{N}_{B_s}$ , respectively. Next, for each of these subvectors, we propose its possible values for all  $\boldsymbol{\eta}$  in the desired subset  $\Omega'$ .

- According to the last constraint in (3.13), for any  $\boldsymbol{\eta} \in \Omega$ ,  $\boldsymbol{\eta}(\mathcal{A}_f) = \mathbf{1}$  where  $\mathbf{1}$  denotes a vector of all 1's of the proper dimension. Thus, for any  $\boldsymbol{\eta} \in \Omega'$ , we also have  $\boldsymbol{\eta}(\mathcal{A}_f) = \mathbf{1}$ .
- Consider the entries corresponding to  $\mathcal{A}_p$ . Without loss of generality, we assume that  $\mathcal{A}_p = \{(i_1, j_1), \dots, (i_{|\mathcal{A}_p|}, j_{|\mathcal{A}_p|})\}$  with  $\bar{\eta}_{i_k, j_k} \leq \bar{\eta}_{i_{k+1}, j_{k+1}}$  for all  $k \in \{1, \dots, |\mathcal{A}_p| - 1\}$ , i.e., the precedence relations in  $\mathcal{A}_p$  are sorted according to the ascending order of  $\bar{\eta}_{i_j, j}$ . For any  $k \in \{1, \dots, |\mathcal{A}_p| + 1\}$ , let  $\boldsymbol{\eta}_0^k$  be the  $|\mathcal{A}_p|$ -dimensional vector where the first  $k - 1$  entries are 0's and the other entries are 1's. For all  $\boldsymbol{\eta} \in \Omega'$ , we consider  $\boldsymbol{\eta}(\mathcal{A}_p)$  to be  $\boldsymbol{\eta}_0^k$  for some  $k \in \{1, \dots, |\mathcal{A}_p| + 1\}$ . Furthermore, let  $p_0^1 = \bar{\eta}_{i_1, j_1}$ ,  $p_0^k = \bar{\eta}_{i_k, j_k} - \bar{\eta}_{i_{k-1}, j_{k-1}}$  for all  $k \in \{2, \dots, |\mathcal{A}_p|\}$ , and  $p_0^{|\mathcal{A}_p|+1} = 1 - \bar{\eta}_{i_{|\mathcal{A}_p|}, j_{|\mathcal{A}_p|}}$ . It is straightforward that  $\sum_{k=1}^{|\mathcal{A}_p|+1} p_0^k = 1$  and  $\sum_{k=1}^{|\mathcal{A}_p|+1} p_0^k \boldsymbol{\eta}_0^k = [\bar{\eta}_{i_1, j_1}, \dots, \bar{\eta}_{i_{|\mathcal{A}_p|}, j_{|\mathcal{A}_p|}}]^\top$ . Therefore,  $p_0^k$  can be viewed as the probability for  $\tilde{\boldsymbol{\eta}}(\mathcal{A}_p) = \boldsymbol{\eta}_0^k$  if the support of  $\tilde{\boldsymbol{\eta}}$  is  $\Omega'$ .

For example, suppose that  $\mathcal{A}_p = \{(7, 8), (7, 9), (8, 9)\}$  with  $\bar{\eta}_{78} = 0.2$ ,  $\bar{\eta}_{79} =$

0.6, and  $\bar{\eta}_{89} = 0.9$ . We have

$$\begin{array}{r} \eta_0^1 \quad \eta_0^2 \quad \eta_0^3 \quad \eta_0^4 \\ \eta_{78} \left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ \eta_{79} \left[ \begin{array}{cccc} 1 & 1 & 0 & 0 \\ \eta_{89} \left[ \begin{array}{cccc} 1 & 1 & 1 & 0 \end{array} \right] \right. \\ p_0^k \quad 0.2 \quad 0.4 \quad 0.3 \quad 0.1 \end{array} \right. \end{array} \right. \end{array} \quad (3.18)$$

- For any  $i \in \mathcal{N}_{B_s}$ , given  $\boldsymbol{\eta} \in \Omega$ , the first constraint in (3.13) shows that  $\boldsymbol{\eta}(\mathcal{A}_b^i)$  must be a  $g_i$ -dimensional standard basis vector, i.e., a  $g_i$ -dimensional vector with only one entry equal to 1 and the others 0's, which implies only one alternative plan is chosen after any branching starting node  $i \in \mathcal{N}_{B_s}$ . Therefore, for any  $\boldsymbol{\eta} \in \Omega'$ ,  $\boldsymbol{\eta}(\mathcal{A}_b^i) \in \{\mathbf{b}_i^1, \dots, \mathbf{b}_i^{g_i}\}$  where  $\mathbf{b}_i^k$  for any  $k \in \{1, \dots, g_i\}$  is the  $g_i$ -dimensional vector with the  $k$ th entry to be 1 and the others 0's. Since  $\sum_{k=1}^{g_i} \bar{\eta}_{ij_i^k} \mathbf{b}_i^k = [\bar{\eta}_{ij_i^1}, \dots, \bar{\eta}_{ij_i^{g_i}}]^\top$  and  $\sum_{k=1}^{g_i} \bar{\eta}_{ij_i^k} = 1$ ,  $\bar{\eta}_{ij_i^k}$  can be viewed as the probability of  $\tilde{\boldsymbol{\eta}}(\mathcal{A}_b^i) = \mathbf{b}_i^k$  given that the support of  $\tilde{\boldsymbol{\eta}}$  is  $\Omega'$ . According to the second constraint in (3.13), for  $\boldsymbol{\eta} \in \Omega$ ,  $\boldsymbol{\eta}(\mathcal{A}_f^i)$  can be determined by  $\boldsymbol{\eta}(\mathcal{A}_b^i)$ . As a result, for  $\boldsymbol{\eta} \in \Omega'$ , given  $\boldsymbol{\eta}(\mathcal{A}_b^i) = \mathbf{b}_i^k$  for some  $k \in \{1, \dots, g_i\}$ , the second constraint in (3.13) yields a unique vector  $\mathbf{f}_i^k$  such that  $\boldsymbol{\eta}(\mathcal{A}_f^i) = \mathbf{f}_i^k$ . As there is a one-to-one correspondence between  $\mathbf{b}_i^k$  and  $\mathbf{f}_i^k$ , if we restrict the support of  $\tilde{\boldsymbol{\eta}}$  to  $\Omega'$ , the probability for  $\tilde{\boldsymbol{\eta}}(\mathcal{A}_f^i) = \mathbf{f}_i^k$  is  $\bar{\eta}_{ij_i^k}$ .

For example, consider the probabilistic branching after task 1 in Figure 2.1. We have  $i = 1$ ,  $g_1 = 2$ ,  $G_1^1 = \{2, 3, 4\}$ , and  $G_1^2 = \{5, 6\}$ . Suppose that  $j_1^1 = 2$  and  $j_1^2 = 5$ . Then  $\mathcal{A}_b^1 = \{(1, 2), (1, 5)\}$  with  $\bar{\eta}_{12} = \bar{\eta}_{15} = 0.5$  and

$$\mathcal{A}_f^1 = \{(1, 3), (1, 4), (1, 6)\}.$$

$$\begin{array}{ccc} & \mathbf{b}_1^1 & \mathbf{b}_1^2 \\ \eta_{12} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \eta_{15} & \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \bar{\eta}_{1j_1^k} & 0.5 & 0.5 \end{array} \quad \text{and} \quad \begin{array}{cc} \mathbf{f}_1^1 & \mathbf{f}_1^2 \\ \eta_{13} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \eta_{14} & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ \eta_{16} & \begin{bmatrix} 0 & 1 \end{bmatrix} \\ \bar{\eta}_{1j_1^k} & 0.5 & 0.5 \end{array}. \quad (3.19)$$

Given  $\boldsymbol{\eta} \in \Omega$ , the above analysis obtains the possible values of the subvectors  $\boldsymbol{\eta}(\mathcal{A}_f)$ ,  $\boldsymbol{\eta}(\mathcal{A}_p)$ ,  $\boldsymbol{\eta}(\mathcal{A}_b^i)$  and  $\boldsymbol{\eta}(\mathcal{A}_f^i)$  for all  $i \in \mathcal{N}_{Bs}$ . It also computes the probability for each subvector to take a certain value when  $\Omega'$  is the support of  $\tilde{\boldsymbol{\eta}}$ . Based on this information, we can use Algorithm 2 to combine the subvectors into the elements of  $\Omega'$  and find a feasible solution  $(p(\boldsymbol{\eta}) \forall \boldsymbol{\eta} \in \Omega')$  to (3.16). For example, consider a project network consisting of  $\mathcal{A}_f$ ,  $\mathcal{A}_p = \{(7, 8), (7, 9), (8, 9)\}$  with  $\bar{\eta}_{78} = 0.2$ ,  $\bar{\eta}_{78} = 0.6$ , and  $\bar{\eta}_{89} = 0.9$ , and the probabilistic branching in Figure 2.1. We can use (3.18) and (3.19) to obtain  $\Omega' = \{\boldsymbol{\eta}^1, \dots, \boldsymbol{\eta}^5\}$  and a corresponding feasible solution  $(p(\boldsymbol{\eta}) \forall \boldsymbol{\eta} \in \Omega')$  shown as follows:

$$\begin{array}{ccccc} & \boldsymbol{\eta}^1 & \boldsymbol{\eta}^2 & \boldsymbol{\eta}^3 & \boldsymbol{\eta}^4 & \boldsymbol{\eta}^5 \\ \boldsymbol{\eta}(\mathcal{A}_f) & \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ \boldsymbol{\eta}(\mathcal{A}_p) & \begin{bmatrix} \boldsymbol{\eta}_0^1 & \boldsymbol{\eta}_0^2 & \boldsymbol{\eta}_0^2 & \boldsymbol{\eta}_0^3 & \boldsymbol{\eta}_0^4 \end{bmatrix} \\ \boldsymbol{\eta}(\mathcal{A}_b^1) & \begin{bmatrix} \mathbf{b}_1^1 & \mathbf{b}_1^1 & \mathbf{b}_1^2 & \mathbf{b}_1^2 & \mathbf{b}_1^2 \end{bmatrix} \\ \boldsymbol{\eta}(\mathcal{A}_f^1) & \begin{bmatrix} \mathbf{f}_1^1 & \mathbf{f}_1^1 & \mathbf{f}_1^2 & \mathbf{f}_1^2 & \mathbf{f}_1^2 \end{bmatrix} \\ p(\boldsymbol{\eta}) & 0.2 & 0.3 & 0.1 & 0.3 & 0.1 \end{array}$$

Let us discuss the general idea of Algorithm 2 here. The vectors  $\eta_0^k$  correspond to the uncertain precedence relations in  $\mathcal{A}_p$ . They are generated in a way that we try to put the most number of precedence relationships in a scenario as long as the expectation requirement (3.15b) is not violated. The vectors  $\mathbf{b}_i^k$  and  $\mathbf{f}_i^k$  correspond to the probabilistic branching after task  $i$ . In particular, each pair of  $\mathbf{b}_i^k$  and  $\mathbf{f}_i^k$  represents a given branch after task  $i$ . We then combine these vectors of ones for  $\mathcal{A}_f$ , the vectors  $\eta_0^k$ , and the pairs  $(\mathbf{b}_i^k, \mathbf{f}_i^k)$  so as to satisfy the expectation requirement (3.15b). The more intuitive explanation of how Algorithm 2 gets the feasible solution in the last paragraph is shown as Figure 3.1. In Figure 3.1, we only present the combination process of  $\eta_0^k$  and  $(\mathbf{b}_i^k, \mathbf{f}_i^k)$ , as  $\eta(\mathcal{A}_f)$  does not affect  $|\Omega'|$  and  $p(\eta)$ ,  $\forall \eta \in \Omega'$ .

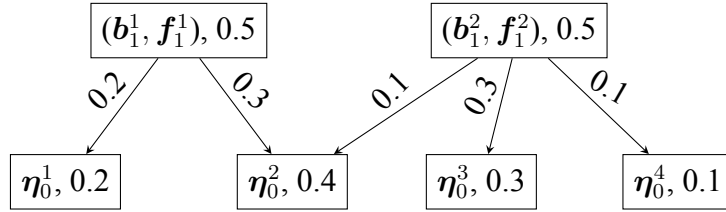


Figure 3.1: Intuitive Explanation of Algorithm 2

In Algorithm 2, the while loop from line 4 to 12 will be executed for at most  $|\mathcal{A}_p| + \sum_{i \in \mathcal{N}_{B_s}} g_i \leq |\mathcal{A}|$  times. Thus, the subset  $\Omega'$  returned by Algorithm 2 has at most  $O(|\mathcal{A}|)$  elements. Also note that each  $\eta \in \Omega'$  has  $|\mathcal{A}|$  entries. The computational complexity is bounded by  $O(|\mathcal{A}|^2)$ .

### 3.2.2 Solve the Pricing Problem

To solve the pricing problem in (3.17), we reformulate it as a compact mixed integer linear program shown in Theorem 3, which can be readily solved by commer-

---

**Algorithm 2:** Find the Initial Subset  $\Omega' \subseteq \Omega$

---

**Input:**  $\mathcal{N}_{B_s}, \mathcal{A}_p, \mathcal{A}_f, \mathcal{A}_b^i \forall i \in \mathcal{N}_{B_s}, \mathcal{A}_f^i \forall i \in \mathcal{N}_{B_s}, \bar{\eta}_{ij} \forall (i, j) \in \mathcal{A}, \mathbf{b}_i, \mathbf{f}_i, \boldsymbol{\eta}_0$

**Output:**  $\Omega' \subseteq \Omega, (p(\boldsymbol{\eta}) \forall \boldsymbol{\eta} \in \Omega')$  feasible to (3.16)

- 1 Label the elements in  $\mathcal{A}_p$  as  $\{(i_1, j_1), \dots, (i_{|\mathcal{A}_p|}, j_{|\mathcal{A}_p|})\}$  with  $\bar{\eta}_{i_k, j_k} \leq \bar{\eta}_{i_{k+1}, j_{k+1}}$  for all  $k \in \{1, \dots, |\mathcal{A}_p| - 1\}$
  - 2  $p_0^1 \leftarrow \bar{\eta}_{i_1, j_1}, p_0^k \leftarrow \bar{\eta}_{i_k, j_k} - \bar{\eta}_{i_{k-1}, j_{k-1}}$  for all  $k \in \{2, \dots, |\mathcal{A}_p|\}$ ,  
 $p_0^{|\mathcal{A}_p|+1} \leftarrow 1 - \bar{\eta}_{i_{|\mathcal{A}_p|}, j_{|\mathcal{A}_p|}}$
  - 3  $\Omega' \leftarrow \emptyset, P \leftarrow 0, k_0 \leftarrow 1, k_i \leftarrow 1$  for all  $i \in \mathcal{N}_{B_s}$
  - 4 **while**  $P < 1$  **do**
  - 5      $q \leftarrow \min\{p_0^{k_0}, \bar{\eta}_{i, j_i}^{k_i} \forall i \in \mathcal{N}_{B_s}\}$
  - 6     **if**  $q > 0$  **then**
  - 7         Generate  $\boldsymbol{\eta}$  such that  $\boldsymbol{\eta}(\mathcal{A}_f) \leftarrow \mathbf{1}, \boldsymbol{\eta}(\mathcal{A}_p) \leftarrow \boldsymbol{\eta}_0^{k_0}, \boldsymbol{\eta}(\mathcal{A}_b^i) \leftarrow \mathbf{b}_i^{k_i}$   
        and  $\boldsymbol{\eta}(\mathcal{A}_f^i) \leftarrow \mathbf{f}_i^{k_i}$  for all  $i \in \mathcal{N}_{B_s}$
  - 8          $\Omega' \leftarrow \Omega' \cup \{\boldsymbol{\eta}\}, p(\boldsymbol{\eta}) \leftarrow q$
  - 9          $P \leftarrow P + q, p_0^{k_0} \leftarrow p_0^{k_0} - q, \bar{\eta}_{i, j_i}^{k_i} \leftarrow \bar{\eta}_{i, j_i}^{k_i} - q$  for all  $i \in \mathcal{N}_{B_s}$
  - 10     **if**  $p_0^{k_0} = 0$  **then**  $k_0 \leftarrow k_0 + 1;$
  - 11     **for**  $i \in \mathcal{N}_{B_s}$  **do**
  - 12         **if**  $\bar{\eta}_{i, j_i}^{k_i} = 0$  **then**  $k_i \leftarrow k_i + 1;$
-

cial optimization packages.

**Theorem 3.** *The pricing problem in (3.17) is equivalent to*

$$\max_{\mathbf{y}, \mathbf{q}(\cdot), \lambda, \boldsymbol{\eta}} \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} cq_i(c) - T\lambda - \sum_{(i, j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \quad (3.20a)$$

$$s.t. \quad \sum_{c \in \mathcal{C}_i} q_i(c) = \sum_{(i, j) \in \mathcal{A}} y_{ij} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (3.20b)$$

$$\sum_{j: (i, j) \in \mathcal{A}} y_{ij} = \sum_{j: (j, i) \in \mathcal{A}} y_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (3.20c)$$

$$\sum_{i: (s, i) \in \mathcal{A}} y_{si} = \lambda, \quad (3.20d)$$

$$\sum_{i: (i, t) \in \mathcal{A}} y_{it} = \lambda, \quad (3.20e)$$

$$0 \leq q_i(c) \leq p_i(c) \quad \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i, \quad (3.20f)$$

$$0 \leq \lambda \leq 1, \quad (3.20g)$$

$$0 \leq y_{ij} \leq \eta_{ij}, \quad \forall (i, j) \in \mathcal{A}, \quad (3.20h)$$

$$\boldsymbol{\eta} \in \Omega. \quad (3.20i)$$

*Proof.* Recall that  $Z^*(\boldsymbol{\eta})$  can be formulated as the minimization problem in (3.8).

Let  $q_i(c)$ ,  $\lambda$ ,  $x_{ij}$ ,  $x_{si}$ , and  $x_{it}$  be the dual variables corresponding to constraints (3.8b), (3.8c), (3.8d), (3.8e), and (3.8f), respectively. Applying strong duality to

the linear program, we have

$$Z^*(\boldsymbol{\eta}) = \max_{\mathbf{x}, \mathbf{q}(\cdot), \lambda} \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} cq_i(c) - T\lambda \quad (3.21a)$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}_i} q_i(c) = \sum_{(i, j) \in \mathcal{A}} \eta_{ij} x_{ij} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (3.21b)$$

$$\sum_{j: (i, j) \in \mathcal{A}} \eta_{ij} x_{ij} = \sum_{j: (j, i) \in \mathcal{A}} \eta_{ji} x_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (3.21c)$$

$$\sum_{i: (s, i) \in \mathcal{A}} x_{si} = \lambda, \quad (3.21d)$$

$$\sum_{i: (i, t) \in \mathcal{A}} x_{it} = \lambda, \quad (3.21e)$$

$$x_{ij} \geq 0 \quad \forall (i, j) \in \mathcal{A}, \quad (3.21f)$$

$$(3.20f), (3.20g). \quad (3.21g)$$

Inserting the above formulation of  $Z^*(\boldsymbol{\eta})$  into (3.17), the pricing problem can be reformulated as follows:

$$\begin{aligned} & \max_{\mathbf{x}, \mathbf{q}(\cdot), \lambda, \boldsymbol{\eta}} \sum_{i \in \mathcal{N} \setminus \{s, t\}} \sum_{c \in \mathcal{C}_i} cq_i(c) - T\lambda - \sum_{(i, j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \\ & \text{s.t.} \quad (3.21b), (3.21c), (3.21d), (3.21e), (3.21f), (3.20f), (3.20g), \\ & \quad \boldsymbol{\eta} \in \Omega. \end{aligned}$$

Due to the network flow balance constraints and  $0 \leq \lambda \leq 1$ , we have  $0 \leq x_{ij} \leq 1, \forall (i, j) \in \mathcal{A}$ . Also, note that  $\eta_{ij}$  is a binary decision variable. Therefore,  $\eta_{ij} x_{ij}$  is always within  $[0, \eta_{ij}]$ . Replacing  $\eta_{ij} x_{ij}$  with a new decision variable  $0 \leq y_{ij} \leq \eta_{ij}$ , we obtain Theorem 3.  $\square$



# Chapter 4

## Computational Study

In this section, we conduct a computational study. The objectives of the computational study are as follows. First, in Section 4.1, we compute the instance of classical research and development projects from the literature to demonstrate the practicality of our approach. Second, Section 4.2 conducts numerical experiments in small-scale, middle-scale, and large-scale networks to test the performance of the proposed approach and obtain findings. Third, Section 4.3 presents one small instance to check solution quality. In this thesis, we use the Python programming language and call Gurobi solver. The computation device is a Linux server operating on Ubuntu 18.04.5 LTS. It is equipped with 80 Intel Xeon Gold 5218R processors running at 2.10 GHz, and it has a generous memory capacity of 256GB RAM.

## 4.1 Classical Research and Development Projects Network

Research and development projects (R&D projects) are essential when a new technology, product, or proposal is developed. It is practical and widely applied in various industries, such as electronic products, aerospace, commerce, and pharmaceuticals. According to Moore and Taylor (1977), network analysis has been applied to plan and control of R&D projects since the 1950s. The classical procedure for R&D projects is as follows: problem definition (PD), research implementation (RI), solution proposal (SP), prototype development (PtD), and solution implementation (SI). However, Moore and Taylor (1977) present that four special uncertainties in real project execution need to be considered.

- First, experience has shown that problem definition is frequently failed, and project executors need to redefine the problem until that problem definition is successfully completed.
- Second, the projects team will meet four choices after the solution proposal: going directly to prototype development, doing more research, redefining the problem, and washout.
- Third, the project team still encounters the possibility of redefining the problem after doing more research; otherwise, go directly to the solution proposal.
- Fourth, prototype development sometimes fails, so the project team needs to redevelop the prototype until prototype development is successfully com-

pleted.

Every time the project team enters the problem definition or prototype development task, they have to ensure successful completion before leaving, which means that rework would often be necessary. Considering these uncertainties, Moore and Taylor (1977) use the numerical simulation method to estimate the completion time of R&D projects for the subsequent four projects.

Our generalized and robust model can solve this instance more efficiently based on the below fitness.

- Model (2.2) can be applied to model rework of the tasks problem definition and prototype development. Observe that  $\tilde{m}_i$  in the model (2.2) represents the number of times the task problem definition needs to be completed before achieving success, following a Geometric distribution. If the distribution of completion time of problem definition once and the probability of successful problem definition each time are given, the distribution of completion time of successful problem definition can be determined easily. In the same way, the time distribution of successful prototype development completion can also be determined. Therefore, the task notations successful problem definition (PDS) and successful prototype development (PtDS) can be used as replacements for the task notations problem definition (PD) and prototype development (PtD) in the network.
- Choices after solution proposal can be modeled using  $\tilde{\eta}$ . To apply our model, we adopt a fitting approach to handle situations where additional branching start nodes are encountered after the initial branching start node. Listing all extended alternative plans starting from the initial branching start

node, all subsequent branching start nodes are included in these extended alternative plans. As a result, certain alternative plans may have a significantly low probability of being chosen. By setting a threshold of 0.01, alternative plans with a probability below this threshold are automatically excluded, and the corresponding probabilities are absorbed into the washout choice.

The instance in Moore and Taylor (1977) can be reorganized as the generalized project network shown in Figure 4.1. Moore and Taylor (1977) organize the R&D projects process into a series of four consecutive projects. The network structure of the four projects remains consistent, with minor variations in the completion time of individual tasks and the probabilities associated with alternative plans. We present one project network in Figure 4.1, but we will perform numerical computations on these four projects in the next step.

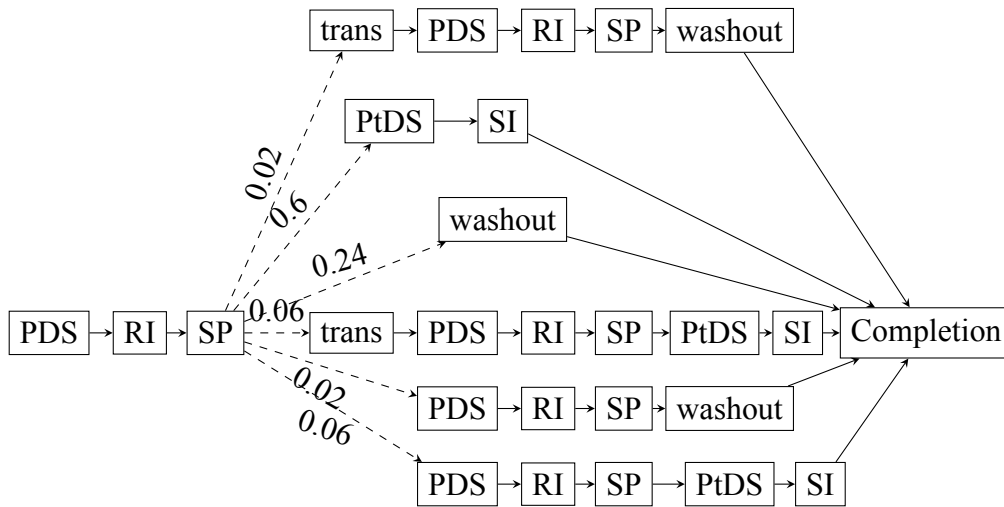


Figure 4.1: Network of R&D Process for One Project

### 4.1.1 Input Explanation

This subsection introduces the input data as follows:

- Distribution of random variables ( $\tilde{c}_i$ ): randomly generated real numbers within the range defined by the minimum and maximum values and their corresponding random probability value. In practice, the specific project will determine the specific distribution of  $\tilde{c}_i$ .
- Nodes and arcs set: to fit the instance into our model framework, the tasks repeatedly included in multiple alternative plans are considered disjoint. As a result, there are 142 nodes within the network structure. There are 26 arcs in  $\mathcal{A}_b$ , while all other existing arcs are in  $\mathcal{A}_f$ .
- The expectation  $\bar{\eta}_{ij}$  of Bernoulli random variables  $\tilde{\eta}_{ij}$  is 1 for each  $(i, j)$  in  $\mathcal{A}_f$ , and it's the probability that the alternative plan including  $(i, j)$  is chosen for each  $(i, j)$  in  $\mathcal{A}_b^i, \forall i \in \mathcal{N}_{b_s}$ , respectively. The probability values are sourced from Moore and Taylor (1977).

### 4.1.2 Numerical Result

When reorganizing this network, we have 142 tasks, four alternative plans start nodes within  $\mathcal{N}_{b_s}$ , and a total of 26 arcs within  $\bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i$  after these four alternative plans start nodes. There are 395 arcs within  $\mathcal{A}$  and 267 arcs within  $\mathcal{A}_f$ . We add two arcs to  $\mathcal{A}_p$  to fit the model. By implementing computation, the numerical result is shown in the first row of Table 4.2. The column generation algorithm was executed for 156 iterations, with a CPU runtime of 98.2 seconds. The computation time for solving the master problem and the pricing problem once was remarkably short, at 0.23 seconds and 0.4 seconds, respectively. Based on these results, we can infer that our model is well-suited for project completion time estimation problems in networks with uncertainties in task duration and task outcomes. Our method

exhibits several advantages over literature approaches. It demonstrates superior generality and can replace numerical and simulation approaches while requiring significantly shorter computation time. Moreover, our approach is robust even when provided with limited distributional information.

## 4.2 Performance Test

The computational study conducted in this section is a pivotal component of our research, offering valuable insights into the robust optimization and mixed programming aspects of our proposed model and solution algorithm. Through systematic experimentation on networks of varying sizes, we aim to evaluate the performance and scalability of our approach comprehensively. By analyzing the results obtained from various computational scenarios, we can draw meaningful conclusions regarding the effectiveness and efficiency of our model. Specifically, our attention is directed towards the following key performance indicators: the number of iterations, the solving time of the pricing problem in each iteration, and the solving time of the master problem in each iteration used in the column generation algorithm. These metrics provide crucial insights into the computational complexity and runtime efficiency of our proposed model and solution approach. By examining the behavior of our algorithm across different network sizes, we aim to identify any patterns or trends that may emerge. The findings from this study will validate the practicality of our approach and highlight potential areas for further research to improve and optimize.

While developing our model and algorithm, it is of utmost importance to consider the impact of network scale. Specifically, we need to take into account the

number of arcs in  $\mathcal{A}_p$  and the number of task nodes in  $\mathcal{N}_{b_s}$ , as these factors have a significant impact on overall performance and computing time. Therefore, in the following subsections, we will thoroughly examine three sets of computations conducted on small-scale, middle-scale, and large-scale networks. Within each network scale, we will carefully adjust the number of arcs within  $\mathcal{A}_p$ , as well as the number of task nodes within  $\mathcal{N}_{b_s}$ . With this systematic approach, we aim to demonstrate the practicality of our research across different network scales. Moreover, we seek to identify any discernible relationships between computation time and the numbers of arcs within  $\mathcal{A}_p$  and task nodes within  $\mathcal{N}_{b_s}$ .

### 4.2.1 Performance Test in Small-scale and Middle-scale Networks

In this section, our primary focus is conducting numerical experiments on small-scale and middle-scale networks. The small-scale network comprises 126 arcs and 50 tasks, while the middle-scale network comprises 192 arcs and 80 tasks. The completion time of each task is randomly generated within the range of 1 to 10. The expectations  $\bar{\eta}_{ij}$  are randomly assigned values between 0-1 for  $(i, j)$  in  $\mathcal{A}_p$  and  $\bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i$ , with the constraint that  $\sum_{(i,j) \in \mathcal{A}_b^i} \bar{\eta}_{ij} = 1, \forall i \in \mathcal{N}_{b_s}$ . We vary the number of arcs in  $\mathcal{A}_p$  and adjust the number of task nodes in  $\mathcal{N}_{b_s}$ . Due to the short computation time required for the small and middle-scale numerical experiments, we present the combined results in Table 4.1. The table includes a total of 7 numerical experiments, with the first three small-scale network experiments and the remaining six middle-scale experiments. Each row contains the input data in the second through sixth columns, followed by the result data in the last four

columns of each experiment. Columns two and three represent the total number of nodes in  $\mathcal{N}$  and total arcs in  $\mathcal{A}$ , indicating the network size. Columns four, five, and six represent the numbers of branching start nodes, all alternative plans that follow these branching start nodes, and precedence requirements that may not be satisfied, respectively.

index	$ \mathcal{N} $	$ \mathcal{A} $	$ \mathcal{N}_{b_s} $	$ \bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i $	$ \mathcal{A}_p $	itera. #	CPUtime	last_master_time	pricing_time
1	50	126	4	8	22	97	36.076s	0.158s	0.21s
2	50	126	7	15	22	120	50.3s	0.21s	0.28s
3	50	126	7	15	12	70	27.5s	0.145s	0.19s
4	79	192	14	33	7	50	10.6s	0.1s	0.18s
5	79	192	14	33	12	106	36.076s	0.158s	0.25s
6	79	192	14	33	17	150	70.6s	0.358s	0.28s
7	79	192	14	33	22	218	110.6s	0.659s	0.3s
8	79	192	10	24	22	167	75.2s	0.355s	0.28s
9	79	192	5	12	22	106	37.6s	0.159s	0.22s

Table 4.1: Numerical Result in Small-scale and Middle-scale Networks

In this table, itera. # represents the number of iterations the column generation algorithm needs in the computation process. CPU time is the total time to solve each numerical experiment. The last two columns are the time to solve the master problem in the last iteration and the average time to solve the pricing problem once. We have findings as follows.

- In both small and middle-scale experiments, we have successfully demonstrated the efficiency of our formulation as shown, the total computation time for small-scale and middle-scale cases remains within one minute and two minutes, respectively.
- Basically, as more columns join the master problem, the runtime of the master problem gradually increases. Experiment results support this point and reveal that the runtime spent on solving the pricing problem remains almost unchanged.



- Furthermore, we have observed a trend where the addition of 10 arcs in  $\mathcal{A}_p$  or in  $\bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i$  can at least double the computation time significantly.
- Finally, in experiments 1 2 3 4 5 9, the time taken to solve the master problem in the last iteration was shorter than the average time taken to solve the pricing problem once. We can argue that for small and middle cases with fewer than 130 iterations, the pricing problem accounts for the majority of the computation time in the entire solving process. Conversely, above this size, we observe that the master problem dominates the computation time.

These findings demonstrate the efficiency and practicality of our approach and provide insights for further enhancing algorithms for small-scale and middle-scale networks. In the following subsection, we will move on to large-scale cases to explore the scalability of our approach and draw further conclusions.

## 4.2.2 Performance Test in Large-scale Network

This section presents the results of computational experiments conducted on the large-scale network. The input data for these experiments are as follows: the number of task nodes is approximately 140, the number of total arcs is approximately 400, and the distribution of each task completion time is randomly generated within 1 to 10. Expectations  $\bar{\eta}_{ij}$  are random values between 0-1 for  $(i, j)$  in  $\mathcal{A}_p$  and  $\bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i$ , and  $\sum_{(i,j) \in \mathcal{A}_b^i} \bar{\eta}_{ij} = 1, \forall i \in \mathcal{N}_{b_s}$ . Additionally, the experimental setup involves adjusting the number of arcs in  $\mathcal{A}_p$ , with a maximum of 74, and the number of branching start nodes in  $\mathcal{N}_{b_s}$ , with a maximum of 26 (total of 53 arcs in  $\bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_b^i$ ).

Let's introduce key output metrics and their corresponding column names in

Table 4.2 as follows: total time to solve each case (CPUtime), the number of iterations in the column generation process (itera. #), the computation time of the master problem in the last iteration (last\_master\_time), the average computation time of the pricing problem per iteration (pricing\_time), and the optimal objective value (Result). By examining these outputs, we aim to gain insights into the performance and efficiency of our computational approach in handling large-scale network flows. Table 4.2 shows the detailed input and output. The column names are defined similarly to the small-scale network table.

Figure 4.2 provides a more intuitive representation of the relationship between the number of arcs in  $\mathcal{A}_p$  and system performance. The x-axis represents the number of arcs in  $\mathcal{A}_p$  and the y-axis represents the number of iterations in the column generation algorithm and CPU computational time. Similarly, Figure 4.3 shows the relationship between the number of task nodes in  $\mathcal{N}_{b_s}$  and system performance. The x-axis represents the number of task nodes in  $\mathcal{N}_{b_s}$  and the y-axis represents the number of iterations in the column generation algorithm and CPU computational time.

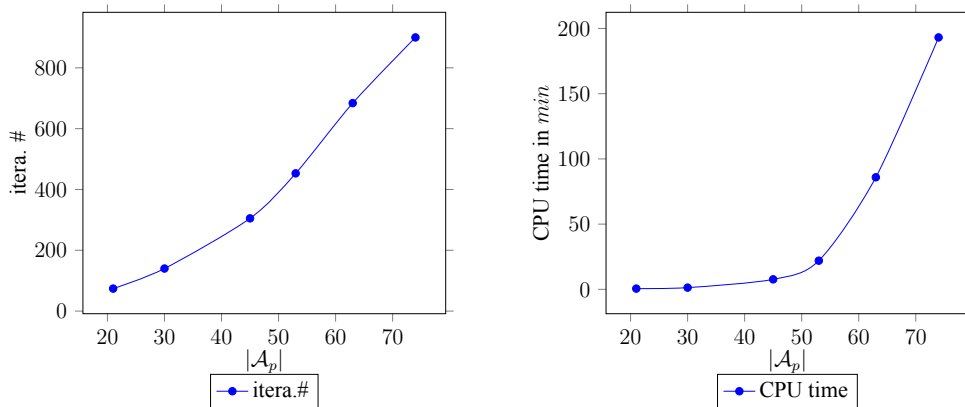


Figure 4.2: Performance Result in the Large-scale Network with  $|\mathcal{A}_p|$

index	$ \mathcal{N} $	$ \mathcal{A} $	$ \mathcal{N}_{b_s} $	$ \bigcup_{i \in \mathcal{N}_{b_s}} \mathcal{A}_i^b $	$ \mathcal{A}_p $	$ \mathcal{A}_f $	itera. #	CPUtime	last_master_time	pricing_time	Result
R&D	142	395	4	26	2	267	156	98.2s	0.232s	0.4s	123.1
10	124	383	3	6	21	352	74	34s	0.221s	0.25s	226
11	124	383	3	6	30	343	140	79.4s	0.578s	0.265s	226
12	124	383	3	6	45	328	305	7.66min	3.04s	0.35s	214
13	124	383	3	6	53	320	453	21.98min	7s	0.31s	213
14	124	383	3	6	63	310	684	85.9min	21s	0.4s	209
15	124	383	3	6	74	299	900	3.219h	36.6s	0.45s	204
16	124	375	6	12	40	316	196	3.13min	1.68s	0.4s	221.2
17	124	361	9	19	40	291	221	4.414min	1.965s	0.38s	251.68
18	140	394	14	29	40	314	279	9.27min	3.7s	0.5s	306.757
19	155	420	20	41	40	327	378	22.07min	7.94s	0.55s	384.579
20	170	446	26	53	40	340	587	1.01h	16.2s	0.61s	460.465

Table 4.2: Numerical Result in the Large-scale Network

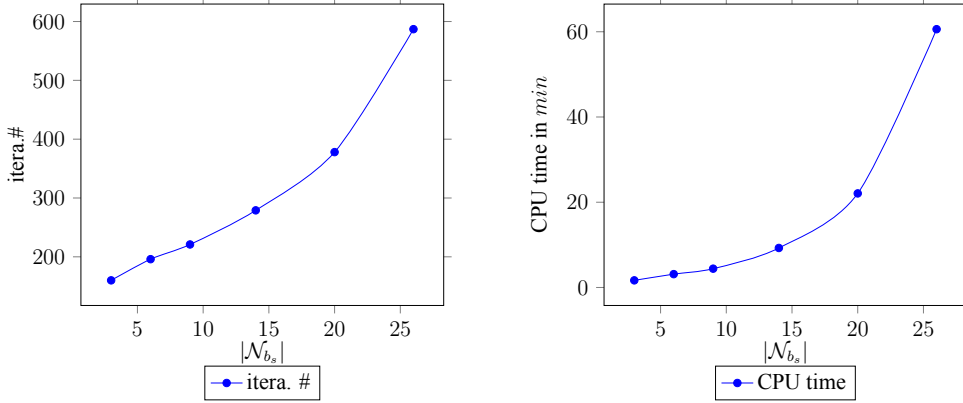


Figure 4.3: Performance Result in Large-scale Network with  $|\mathcal{N}_{b_s}|$

In the conducted 11 experiments, several noteworthy findings about the influence of  $|\mathcal{A}_p|$  on system performance are obtained as follows.

- Firstly, in the first 6 cases, it is observed that the CPU runtime of an individual experiment would at least double upon the addition of 10 arcs in  $\mathcal{A}_p$ . This indicates that the computational time increases significantly with the  $|\mathcal{A}_p|$  increase. In the experimental setup, a maximum of 74 arcs in  $\mathcal{A}_p$  is considered, resulting in a running time of 3.22 hours.
- Similar to the middle-scale experiments, we observe that most of the CPU time is spent computing the master problem. The computation time of the master problem shows an increase in different iterations. This implies that with the inclusion of more columns in different iterations, the computation time for the master problem escalates notably. Across experiments, the computational time of the last master problem significantly increases with the increase of  $|\mathcal{A}_p|$ . Specifically, it is observed that the runtime of the last master problem could double after the addition of every ten arcs in  $\mathcal{A}_p$ . For example, in the case of 74 arcs in  $\mathcal{A}_p$ , the time to solve the last master problem

is 36.6 seconds.

- However, the time required to solve the pricing problem remains relatively constant across iterations in each experiment. Although this indicator increases with the increase of  $|\mathcal{A}_p|$ , the magnitude of the increase is small. For example, although the number of arcs in  $\mathcal{A}_p$  is increased to 74, the average time to solve the pricing problem once is only 0.45 seconds. Specifically, an approximate increase of 0.05 seconds is observed for every additional ten arcs in  $\mathcal{A}_p$ .
- Regarding the optimal objective function value, it is discovered that it decreases as the number of arcs in  $\mathcal{A}_p$  increases. By changing fixed arcs to arcs in  $\mathcal{A}_p$ , more precedence requirements get relaxed, reducing the overall project completion time.

Again, in terms of the influence of  $|\mathcal{N}_{b_s}|$  on system performance, we obtain findings as follows.

- The largest network case consists of 26 branching start nodes, with 53 branchings originating from these nodes. This network requires 587 column generation iterations and a CPU running time of 1.01 hours. It is evident that our approach can handle network sizes beyond this scale.
- The total running time and the last master problem time in each experiment approximately double with the addition of every five nodes in  $\mathcal{N}_{b_s}$ .
- Furthermore, our observations indicate that most of the computational time is spent solving the master problem. In different experiments, the computation time for the last master problem at least doubles with adding every

five branching starting nodes. In experiment 11, the computation time for the last master problem has been increased to 16.2 seconds.

- Similarly, the time required to solve the pricing problem remains relatively constant across iterations in each experiment. Although the time for the pricing problem also increases, the difference is less noticeable. For instance, in the experiment with 26 nodes in  $\mathcal{N}_{b_s}$ , the average time required to solve the pricing problem once is 0.61 seconds.

In summary, in the large-scale network, the entire computing time at least doubles with the addition of every ten arcs in  $\mathcal{A}_p$  or every five nodes in  $\mathcal{N}_{b_s}$ . In each iteration, the running time required to solve the pricing problem remains relatively constant. The pricing problem in different experiments shows an increase in computation time with the increase of  $|\mathcal{N}_{b_s}|$  or  $|\mathcal{A}_p|$ , but to a lesser extent. The master problem's computation time significantly increases with more columns in each experiment. Across different experiments, the computation time of the last master problem substantially increases with the increase of  $|\mathcal{N}_{b_s}|$  or  $|\mathcal{A}_p|$ . The optimal objective function value decreases as the number of arcs in  $\mathcal{A}_p$  increases. The experiments suggest that the model can efficiently handle networks with more than 26 branching starting nodes. In the future, we can conduct more experiments on larger  $|\mathcal{N}_{b_s}|$  or different network structures to obtain more general findings.

To summarize the findings from the small-scale, middle-scale, and large-scale experiments, we can conclude that when the number of arcs in  $\mathcal{A}_p$  exceeds 70, the computation time surpasses 3 hours. Similarly, when the number of branching start nodes exceeds 26, the calculation time extends beyond 1 hour. Based on these observations, we make an initial estimation that we can handle larger network

structures consisting of more than 300 nodes and over 1000 arcs with a potential computing time below 12 hours.  $|\mathcal{N}_{b_s}|$  and  $|\mathcal{A}_p|$  significantly influence the complexity of the proposed algorithm. However, it is essential to conduct more large-scale experiments in the future on different network structures that may impact computational efficiency differently. In the future, we can enhance our algorithm based on these findings. For example, we can add more columns in each iteration of the proposed algorithm framework to solve the pricing problem and the master problem fewer times, which can easily reduce the total CPU time.

### 4.3 Solution Quality Check

We develop a DRO model to solve the tardiness estimation considering uncertainties on tasks and precedence relations. The worst-case expected tardiness can be easily obtained from our approach. Next step, solution quality should be considered. We can compare solutions obtained from our approach to solutions in the literature. However, considering more uncertainties than existing studies, the literature cannot give us enough samples. The simulation method is motivated to generate some instances. We randomly generate 1000 deterministic  $(\tilde{\eta}, \tilde{c})$  for (2.3) based on same distribution information of  $\tilde{\eta}$  and  $\tilde{c}$  as our DRO model, solve  $(Z(\tilde{\eta}, \tilde{c}) - T)^+$  1000 times, and then get  $E[(Z(\tilde{\eta}, \tilde{c}) - T)^+]$ . Obviously, this result will be less than the worst-case expected tardiness obtained from our DRO model. Here, we start from one small network Figure 4.4.

We assume marginal task durations distributions  $\tilde{c}_i, \forall i \in \mathcal{N} \setminus \{s, t\}$ , as discrete uniform distributions with probabilities 0.2 for values 3.0, 3.2, 3.4, 3.6, and 3.8.

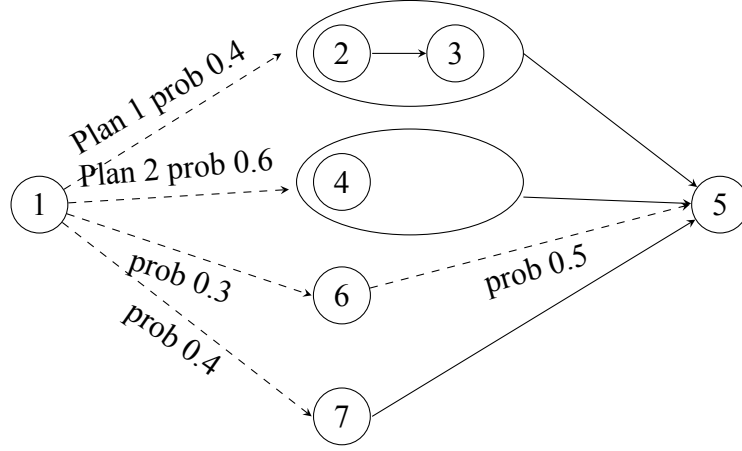


Figure 4.4: Case of Solution Quality Check

Then, the set of all possible joint distributions of  $\tilde{c}$  can be written as

$$\Theta_c = \{\theta_c : P_{\theta_c}\{\tilde{c}_i = c\} = 0.2, \forall i \in \{1, \dots, 7\}, c \in \{3.0, 3.2, 3.4, 3.6, 3.8\}\}.$$

	(1, 6)	(6, 5)	(1, 7)	(1, 2)	(1, 3)	(1, 4)
$\bar{\eta}$	0.3	0.5	0.4	0.4	0.4	0.6

Table 4.3: Marginal Distribution Information

In this network, we have  $(s, 1), (s, 5), (s, 6), (s, 7), (1, t), (5, t), (6, t), (7, t), (7, 5), (2, 3), (3, 5), (2, 5), (4, 5) \in \mathcal{A}_f$ . Thus, we have  $P\{\tilde{\eta}_{ij} = 1\} = 1$  for any  $(i, j) \in \mathcal{A}_f$ .

In addition, we assume marginal distributions for  $\tilde{\eta}_{ij}, (i, j) \notin \mathcal{A}_f$  as Table 4.3.

If  $T = 10$ , based on these input distribution information, the worst-case expected tardiness obtained from our robust approach is 1.78, and the expected tardiness obtained from the simulation is 1.58. The result from our approach is larger than the result from simulation 12.6%. The quality of our solution is initially examined, and we will conduct more comparative experiments in different networks



in the future.

# Chapter 5

## Conclusion and Future Work

This chapter aims to provide an initial conclusion before delving into further investigations and future work. In future research, our focus extends beyond formulation extensions to encompass applied research fields. Furthermore, we will implement more numerical experiments in different network structures, especially those with different probabilistic branching structures. In terms of formulation extensions, we intend to delve deeper into the study of formulation structure and solution procedures, exploring various aspects. Although not limited to these aspects, our plan focuses on two key areas, which are as follows:

- Firstly, we propose excluding the random variables  $\tilde{c}_i$  from our investigation. This exclusion will streamline the model.
- Secondly, we aim to explore alternative methods for modeling the uncertainty set of distributions in the distributionally robust model. This exploration will involve considering any dependence structure that may exist between  $\tilde{\eta}$  and  $\tilde{c}$ .

By addressing these two aspects, we seek to enhance our understanding of the formulation structure and solution procedures, contributing to the overall advancement of the field. Moreover, we will explore extensions of various research fields and contemplate integrating our research into alternative branch of project management.

## 5.1 Conclusion

This research models the problem of project completion time estimation more generally, including uncertainties in task durations and task outcomes such as rework, the introduction of new tasks, changes in precedence, and alternative plans. Our information set for task durations and outcomes is sufficiently general to model practical project management problems. We formulate a DRO model as well as a target-measured of tardiness based on limited distribution information. We develop a specific column generation framework to solve mix-integer programming with an exponential number of decision variables. The performance of the estimates obtained through distributionally robust optimization is evaluated using numerical studies. Numerical studies show that our model is practical and can capture all the uncertainties in both task durations and outcomes in completion time estimation problems. The proposed algorithm demonstrates exceptional efficiency, allowing us to solve one large-size project instance, solved by the simulation method in the literature, in 98.2 seconds. One simulation method is developed to examine the solution quality, and it is initially applied to a tiny instance. In the future, we plan to improve our algorithm based on our findings. We are considering remodeling the formulation by excluding random variables for task

durations. Simplifying the problem formulation can potentially enhance the algorithm's efficiency. Additionally, we can adjust the algorithm framework based on computational studies to reduce the running time. We aim to provide project managers with a general and efficient tool to solve project management problems in projects with various uncertainties. Moreover, more extensive numerical studies are expected to evaluate the performance and check the solution quality.

## 5.2 Future Work on Formulation Structure

### 5.2.1 DRO Model for $\tilde{\eta}$ under Deterministic $c$

We believe that we can simplify our model by eliminating random variables for task time uncertainty, specifically in the pricing problem formulation within the column generation framework. This will lead to significant improvements in the solution procedure. We present a preliminary model framework and plan to develop more advanced algorithms in future research.

We consider replacing the uncertain task time  $\tilde{c}$  with  $\tilde{\eta}$ . If the task durations are discrete random variables, then we can also use the  $\tilde{\eta}$  to model the uncertainty while keeping  $\tilde{c}$  deterministic. For example, suppose that task  $i$  may take 6, 8, 11, or 15 units of time with equal probability. Then we can consider tasks  $\{i_1, i_2, i_3, i_4\}$  in the set  $\mathcal{N}$ , each with deterministic completion times of 6, 8, 11, and 15. We consider four alternative plans  $\{i_1\}$ ,  $\{i_2\}$ ,  $\{i_3\}$ , and  $\{i_4\}$  after any predecessor  $j$  of task  $i$ . The set  $\mathcal{A}$  of precedence relations includes  $(j, i_k)$  for any predecessor  $j$  of task  $i$  with corresponding Bernoulli variables  $\tilde{\eta}_{ji_k}$  for all  $k \in \{1, 2, 3, 4\}$ , and  $(i_k, m)$  for any successor  $m$  of task  $i$  for all  $k \in \{1, 2, 3, 4\}$ . In this case,  $\bar{\eta}_{ji_k} =$

0.25 for all  $k \in \{1, 2, 3, 4\}$  for any predecessor  $j$ , and  $P\{\tilde{\eta}_{i_k m} = 1\} = 1$  for all  $k \in \{1, 2, 3, 4\}$  for any successor  $m$ . If this plan is applicable, the entire project completion time under the random variable  $\tilde{\eta}$  is modeled as follows:

$$Z^*(\tilde{\eta}) = \max \sum_{(i,j) \in \mathcal{A}: i \neq s} c_i \tilde{\eta}_{ij} x_{ij} \quad (5.1a)$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \mathcal{A}} \tilde{\eta}_{ij} x_{ij} = \sum_{j:(j,i) \in \mathcal{A}} \tilde{\eta}_{ji} x_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (5.1b)$$

$$\sum_{i:(s,i) \in \mathcal{A}} x_{si} = 1, \quad (5.1c)$$

$$\sum_{i:(i,t) \in \mathcal{A}} x_{it} = 1, \quad (5.1d)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A} \quad (5.1e)$$

The support of  $\tilde{\eta}$  should be contained in the following set  $\Omega$ :

$$\Omega = \left\{ \boldsymbol{\eta} \in \{0, 1\}^{|\mathcal{A}|} \left| \begin{array}{l} \sum_{k=1}^{g_i} \eta_{ij_i^k} = 1, \quad \forall i \in \mathcal{N}_{B_s}, \\ \eta_{ij} = \eta_{ij_i^k}, \quad \forall i \in \mathcal{N}_{B_s}, k \in \{1, \dots, g_i\}, \\ \quad \quad \quad j \in G_i^k \setminus \{j_i^k\}, \\ \eta_{ij} = 1 \quad \forall (i, j) \in \mathcal{A}_f \end{array} \right. \right\}. \quad (5.2)$$

Combining with the marginal distribution of  $\tilde{\eta}$  specified by the parameters  $\bar{\eta}_{ij}$ , the joint distribution of  $\tilde{\eta}_{ij}$  should be contained within the following set:

$$\Theta_\eta = \left\{ \theta_\eta : \sum_{\boldsymbol{\eta} \in \Omega} P_{\theta_\eta} \{\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\} = 1, P_{\theta_\eta} \{\tilde{\eta}_{ij} = 1\} = \bar{\eta}_{ij} \quad \forall (i, j) \in \mathcal{A} \right\}.$$

No considering target, the worst case expectation completion time is

$$Z^* = \max_{\theta_\eta \in \Theta_\eta, \eta \in \Omega} E_{\theta_\eta}[Z^*(\eta)]. \quad (5.3)$$

For any  $\eta \in \Omega$ , let  $p(\eta) = P\{\tilde{\eta} = \eta\}$ . Model (5.3) can be formulated as

$$\max \sum_{\eta \in \Omega} Z^*(\eta) p(\eta) \quad (5.4a)$$

$$\text{s.t.} \quad \sum_{\eta \in \Omega: \eta_{ij}=1} p(\eta) = \bar{\eta}_{ij} \quad \forall (i, j) \in \mathcal{A}, \quad (5.4b)$$

$$\sum_{\eta \in \Omega} p(\eta) = 1, \quad (5.4c)$$

$$p(\eta) \geq 0 \quad \forall \eta \in \Omega. \quad (5.4d)$$

Applying column generation to solve (5.4), the pricing problem is

$$\max_{\eta \in \Omega} \left\{ Z^*(\eta) - \sum_{(i,j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \right\}. \quad (5.5)$$

Inserting (5.1), the pricing problem (5.5) can be reformulated as

$$Z^*(\tilde{\eta}) = \max \sum_{(i,j) \in \mathcal{A}: i \neq s} c_i \eta_{ij} x_{ij} - \sum_{(i,j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \quad (5.6a)$$

$$\text{s.t.} \quad \sum_{j: (i,j) \in \mathcal{A}} \eta_{ij} x_{ij} = \sum_{j: (j,i) \in \mathcal{A}} \eta_{ji} x_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (5.6b)$$

$$(5.1c), (5.1d) \quad (5.6c)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A} \quad (5.6d)$$

$$\eta \in \Omega \quad (5.6e)$$

Replacing  $\eta_{ij}x_{ij}$  with a new decision variable  $y_{ij}$ , we obtain pricing problem model as

$$\mathcal{P}^* = \max \quad \sum_{(i,j) \in \mathcal{A}: i \neq s} c_i y_{ij} - \sum_{(i,j) \in \mathcal{A}} \alpha_{ij}^* \eta_{ij} \quad (5.7a)$$

$$\text{s.t.} \quad \sum_{j: (i,j) \in \mathcal{A}} y_{ij} = \sum_{j: (j,i) \in \mathcal{A}} y_{ji} \quad \forall i \in \mathcal{N} \setminus \{s, t\}, \quad (5.7b)$$

$$\sum_{i: (s,i) \in \mathcal{A}} y_{si} = 1, \quad (5.7c)$$

$$\sum_{i: (i,t) \in \mathcal{A}} y_{it} = 1, \quad (5.7d)$$

$$0 \leq y_{ij} \leq \eta_{ij} \quad \forall (i, j) \in \mathcal{A} \quad (5.7e)$$

$$\boldsymbol{\eta} \in \Omega \quad (5.7f)$$

Observe that the above model is simpler than the original pricing problem formulation in the column generation framework. Hopefully, we can develop a more efficient algorithm in the future.

### 5.2.2 DRO Model Allowing Dependence between $\tilde{\boldsymbol{\eta}}$ and $\tilde{\boldsymbol{c}}$

Based on model (3.14), we will consider alternative ways to model the ambiguity set of the distributionally robust model. First, although model (3.14) does not require independence between  $\tilde{\boldsymbol{\eta}}$  and  $\tilde{\boldsymbol{c}}$ ,  $\tilde{\boldsymbol{c}}$  is assumed to have the same marginal distributions for any realization of  $\tilde{\boldsymbol{\eta}}$ . To allow any dependence structure between  $\tilde{\boldsymbol{\eta}}$  and  $\tilde{\boldsymbol{c}}$ , we will consider

$$\max_{\theta \in \Theta} E_{\theta}[(Z(\tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{c}}) - T)^+], \quad (5.8)$$

where

$$\Theta = \left\{ \theta \left| \begin{array}{ll} \mathbb{P}_\theta\{\tilde{c}_i = c\} = p_i(c) & \forall i \in \mathcal{N} \setminus \{s, t\}, c \in \mathcal{C}_i, \\ \sum_{\boldsymbol{\eta} \in \Omega} \mathbb{P}_\theta\{\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}\} = 1, & \\ \mathbb{P}_\theta\{\tilde{\eta}_{ij} = 1\} = \bar{\eta}_{ij} & \forall (i, j) \in \mathcal{A} \end{array} \right. \right\}. \quad (5.9)$$

Second, alternative ways exist to model the distributional information of  $\tilde{c}_i$ . For example, according to the definition of  $\tilde{c}_i$  in (2.2), instead of specifying the marginal distribution of  $\tilde{c}_i$ , we can define the distribution of  $\tilde{c}_i$  using the marginal distributions of  $\tilde{m}_i$  and  $\tilde{n}_i$  as well as the marginal distributions or moments of  $\tilde{c}_{i,k}^c$  and  $\tilde{c}_{i,k}^p$ .

For these extensions and future work, we will develop efficient solution algorithms, validate their performance by comparing their results with those from simulations and other benchmarks, and develop insights of value for companies that manage projects with generalized networks.

### 5.2.3 Other Formulation Extensions

Some formulation alternatives are worthwhile to be considered. For objective function, we can consider not just the expectation, but some risk measures. For the DRO model, we can incorporate moments information usually considered by literature. Additionally, when characterizing the ambiguity set, we can use Wasserstein distance. These methods are expected to be applied, and corresponding experiments are conducted to check performance.



### **5.3 Research Field Extensions**

This thesis primarily focuses on the issue of estimating project completion time, which holds great significance in project management. Accurate time estimates can assist companies in saving costs and avoiding contract losses. Moreover, the approach proposed in this study can be applied to various other aspects of project management. For instance, in project portfolio problems, uncertainties exist regarding each project's profit, completion time, and outcomes. By employing a similar method, we can utilize the model to evaluate completion time or profit to cater to the needs of project managers. Similarly, when faced with decision-making problems in the project planning phase, where human resources and materials are subject to uncertainties, our model can be employed to aid project managers in devising the optimal plan.

# Bibliography

- Ahmadi, R., T.A. Roemer, and R.H. Wang. 2001. Structuring product development processes. *European Journal of Operational Research*, **130**(3), 539-558.
- Anklesaria, K.P. and Z. Drezner. 1986. A multivariate approach to estimating the completion time for PERT networks. *Journal of the Operational Research Society*, **37**(8), 811-815.
- Banerjee, A. and A. Paul. 2008. On path correlation and PERT bias. *European Journal of Operational Research*, **189**(3), 1208-1216.
- Bertsimas, D., K. Natarajan, and C.-P. Teo. 2004. Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds. *SIAM Journal on Optimization*, **15**(1), 185-209.
- Bertsimas, D., K. Natarajan, and C.-P. Teo. 2006. Persistence in discrete optimization under data uncertainty. *Mathematical Programming*, **108**(2), 251-274.
- Birge, J., R and Maddox, M., J. 1995. Bounds on expected project tardiness. *Operations Research*, **43**(5), 838-850.

- Carracosa, M., S.D. Eppinger, and D.E. Whitney. 1998. Using the design structure matrix to estimate product development time. *Proceedings of DETC '98, 1998 ASME Design Engineering Technical Conferences*, Atlanta, GA, USA.
- Cho, S.-H. and S.D. Eppinger. 2005. A simulation-based process model for managing complex design projects. *IEEE Transactions on Engineering Management*, **52**(3), 316-328.
- Demeulemeester, E., M. Vanhoucke, and W. Herroelen. 2003. RanGen: A random network generator for activity-on-the-node networks. *Journal of Scheduling*, **6**(1), 17-38.
- Doan X.V. and K. Natarajan. 2012. On the complexity of nonoverlapping multivariate marginal bounds for probabilistic combinatorial optimization problems. *Operations Research*, **60**(1), 138-149.
- Dodin, B. 1985. Bounding the project completion time distribution in PERT networks. *Operations Research*, **33**(4), 862-881.
- Eisner, H. 1962. A generalized network approach to the planning and scheduling of a research program. *Operations Research*, **10**(1), 115-125.
- Elmaghraby, S.E. 1964. An algebra for the analysis of generalized activity networks. *Management Science*, **10**(3), 494-514.
- Fazar, W. 1959. Program evaluation and review technique. *American Statistician*, **13**(2), 10.
- Fulkerson, D.R. 1962. Expected critical path length in PERT networks. *Operations Research*, **10**(6), 808-817.

- Goh, J. and N.G. Hall. 2013. Total cost control in project management via satisficing. *Management Science*, **59**(6), 1354-1372.
- Gong, D. and R. Hugsted. 1993. Time-uncertainty analysis in project networks with a new merge-event time-estimation technique. *International Journal of Project Management*, **11**(3), 165-173.
- Gong, D. and J.E. Rowings, Jr. 1995. Calculation of safe float use in risk-analysis-oriented network scheduling. *International Journal of Project Management*, **13**(3), 187-194.
- Hagstrom, J.N. 1988. Computational complexity of PERT problems. *Networks*, **18**(2), 139-147.
- Hall, N.G. 2016. Research and teaching opportunities in project management. *Optimization Challenges in Complex, Networked and Risky Systems*, INFORMS, Catonsville, MD, USA, 329-388.
- Hall, N.G., D.Z. Long, J. Qi, and M. Sim. 2015b. Managing underperformance risk in project portfolio selection. *Operations Research*, **63**(3), 660-675.
- Hartley, H.O. and A.W. Wortham. 1966. A statistical theory for PERT critical path analysis. *Management Science*, **12**(10), B469-B481.
- Jun, D.H. and K. El-Rayes. 2011. Fast and accurate risk evaluation for scheduling large-scale construction projects. *Journal of Computing in Civil Engineering*, **25**(5), 407-417.
- Kelley, J. and Walker, M. 1959. Critical-path planning and scheduling. In *Proceedings of the Eastern Joint Computer Conference*.

- Kerzner, H. 2013. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 11th edition. Wiley, Hoboken, NJ, USA.
- Kleindorfer, G.B. 1971. Bounding distributions for a stochastic acyclic network. *Operations Research*, **19**(7), 1586-1601.
- Klingel Jr, A.R. 1966. Bias in PERT project completion time calculations for a real network. *Management Science*, **13**(4), B194-B201.
- Kulkarni, V.G. and V.G. Adlakha. 1986. Markov and Markov-regenerative PERT networks. *Operations Research*, **34**(5), 769-781.
- Li, Y., X. Li, J. Shu, M. Song, and K. Zhang. 2022. A General Model and Efficient Algorithms for Reliable Facility Location Problem Under Uncertain Disruptions. *INFORMS Journal on Computing*, **34**(1), 407-426.
- Lu, M., L. Ran, and Z.J.M. Shen. 2015. Reliable facility location design under uncertain correlated disruptions. *Manufacturing & Service Operations Management*, **17**(4), 445-455.
- Ludwig, A., R.H. Möhring, and F. Stork. A computational study on bounding the makespan distribution in stochastic PERT networks. *Annals of Operations Research*, **102**(1), 49-64.
- Malcolm, D.G., Roseboom, J.H., Clark, C.E., and Fazar, W. 1959. Application of a technique for research and development program evaluation. *Operations Research*, **7**(5), 646-669.
- Meilijson, I. and A. Nádas. 1979. Convex majorization with an application to the length of critical paths. *Journal of Applied Probability* **16**(3), 671-677.

- Möhring, R.H. 2001. Scheduling under uncertainty: Bounding the makespan distribution. H. Alt, ed. *Computational Discrete Mathematics: Advanced Lectures, Lecture Notes in Computer Science*, Vol. 2122. Springer, Berlin, 79-97.
- Moore, L.J. and B.W. Taylor. 1977. Multiteam, multiproject research and development planning with GERT. *Management Science*, **24**(4), 401-410.
- Natarajan, K., C.-P. Teo, and Z. Zheng. 2011. Mixed zero-one linear programs under objective uncertainty: A completely positive representation. *Operations Research*, **59**(3), 713-728.
- Pritsker, A.A.B. 1966. GERT: Graphical evaluation and review technique. Memorandum RM-4973-NASA, The Rand Corporation, Santa Monica, CA, USA.
- Pritsker, A.A.B. 1979. *Modeling and Analysis Using Q-GERT Networks*, 2nd edition. John Wiley/Halsted Press, New York, NY, USA.
- Robillard, P. and M. Trahan. 1976. Expected completion time in PERT networks. *Operations Research*, **24**(1), 177-182.
- Schonberger, R.J. 1981. Why projects are “always” late: A rationale based on manual simulation of a PERT/CPM network. *Interfaces*, **11**(5), 66-70.
- Sculli, D. and Y.W. Shum. 1991. An approximate solution to the PERT problem. *Computers & Mathematics with Applications*, **21**(8), 1-7.
- Shogan, A.W. 1977. Bounding distributions for a stochastic PERT network. *Networks*, **7**(4), 359-381.

- Spelde, H.G. 1976. Stochastische Netzpläne und Ihre Anwendung im Baubetrieb. PhD thesis, Rheinisch-Westfälische Technische Hochschule, Aachen, Germany.
- Taylor, B.W. and L.J. Moore. 1980. R&D project planning with Q-GERT network modeling and simulation. *Management Science*, **26**(1), 44-59.
- Welsh, D.J.A. 1965. Errors introduced by a PERT assumption. *Operations Research*, **13**(1), 141-143.
- Zheng, Z., K. Natarajan, and C.-P. Teo. 2016. Least Squares Approximation to the Distribution of Project Completion Times with Gaussian Uncertainty. *Operations Research*, **64**(6), 1406-1421.