Electronic Theses and Dissertations

2019

Stochastic Optimization and Applications with Endogenous Uncertainties Via Discrete Choice Models

Mengnan Chen
University of Central Florida

STOCHASTIC OPTIMIZATION AND APPLICATIONS WITH ENDOGENOUS
UNCERTAINTIES VIA DISCRETE CHOICE MODELS

by

MENGNAN CHEN
M.S. The George Washington University, 2014
B.S. Tianjin University of Finance and Economics, 2012

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2019

Major Professor: Qipeng Zheng

ABSTRACT

Stochastic optimization is an optimization method that solves stochastic problems for minimizing or maximizing an objective function when there is randomness in the optimization process. In this dissertation, various stochastic optimization problems from the areas of Manufacturing, Health care, and Information Cascade are investigated in networks systems. These stochastic optimization problems aim to make plan for using existing resources to improve production efficiency, customer satisfaction, and information influence within limitation. Since the strategies are made for future planning, there are environmental uncertainties in the network systems. Sometimes, the environment may be changed due to the action of the decision maker. To handle this decision-dependent situation, the discrete choice model is applied to estimate the dynamic environment in the stochastic programming model. In the manufacturing project, production planning of lot allocation is performed to maximize the expected output within a limited time horizon. In the health care project, physician is allocated to different local clinics to maximize the patient utilization. In the information cascade project, seed selection of the source user helps the information holder to diffuse the message to target users using the independent cascade model to reach influence maximization.

The computation complexities of the three projects mentioned above grow exponentially by the network size. To solve the stochastic optimization problems of large-scale networks within a reasonable time, several problem-specific algorithms are designed for each project. In the manufacturing project, the sampling average approximation method is applied to reduce the scenario size. In the health care project, both the guided local search with gradient ascent and large neighborhood search with Tabu search are developed to approach the optimal solution. In the information cascade project, the myopic policy is used to separate stochastic programming by discrete time, and the Markov decision process is implemented in policy

evaluation and updating.

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my advisor, Dr. Qipeng P. Zheng for his professional guidance, continuous encouragement and impressive kindness. He is a great mentor, friend and collaborator. It has been an absolute pleasure time to work with such a nice and resourceful advisor. He is always willing to help and give insightful guidance. I am very grateful to be one of his students. His enthusiasm, deep insight and keen observation toward scientific research deserve my greatest respect. I am also grateful to all my lovely research colleagues in our group. I am proud of being part of the group with these hardworking graduate students and we spent a lot of pleasant time cooperating and sharing experience with each other.

I would like to extend my thanks my committee members: Dr. Vladimir Boginski, Dr. Aden E. Vela, and Dr. Muge Yayla Kullu, for their help and effort for being the committee members, and providing me many valuable comments, advises and suggestions for my research and dissertation. From them I have learnt a lot.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Review of Stochastic Optimization

Stochastic optimization (SO) is a common method in mathematics and operations research, and it is an optimization method using random variables. For the formulation of the stochastic optimization problems, the random variables appear in objective functions or constraints. Stochastic programming is an approach to modeling the stochastic optimization problem. It has been widely applied in optimization problems with uncertainty. Uncertainty is usually measured by a probability distribution on the parameters. Stochastic programming deals with problems of maximizing or minimizing objective functions of decision variables and random variables subject to constraints. Stochastic programming has been applied to a wide variety of areas. The traditional applied and studied stochastic programming models are two-stage (linear) programs. In two-stage stochastic programming, the decisions are made at current time, they should be based on data available at this time point, instead of depending on future observations. In the first stage, the decision maker takes some action. After a random event is observed, the outcome of the first-stage decision will be affected by the random variables. Based on the outcome of the first-stage decision, a recourse decision is made in the second stage depending on the random event that occurred. The optimal policy from such a model is a single first-stage decision and a collection of recourse decisions (decision rule) defining which second-stage action should be taken in response to each random outcome. The typical two-stage stochastic programming with the minimization problem can be formulated as

$$\min_{x \in X} \quad f(x) + \mathbb{E}\big[Q(x, \xi)\big]$$

where $Q(x, \xi)$ is the optimal value of the second-stage problem based on the solution first-stage decision $\hat{x}$,

$$\min_{y \in Y(\xi)} \quad g(y, \xi)$$

$$s.t. \quad h(\hat{x}, y, \xi) \leq 0$$

If we consider the two-stage problem as linear, that means the model has a linear objective function, subject to linear equality and linear inequality constraints; then, the two-stage stochastic linear programming can be expressed as

$$\min_{x \in X} \quad c^\mathsf{T} x + \sum_{s \in S} P_s \cdot d_s^\mathsf{T} y_s$$

$$s.t. \quad Ax + B_s y_s \leq h^s \qquad \forall \ s \in S$$

where $s \in S$ is the scenario index and set, while P_s is the probability that the scenario s will happen in the future.

The objective function is optimizing (minimizing) the cost $c^\mathsf{T} x$ of the first-stage decision and the expected cost of the (optimal) second-stage decision. The second-stage problem can simply be considered as an optimization problem that describes the supposedly optimal behavior of the first stage when the uncertain data is revealed. The solution of the second stage is a recourse action where y_s is the cost of this recourse action restricted by the constraint $B_s y_s \leq h^s - A\hat{x}$. For multi-stage problems, the decisions may be made at several time periods $t = 1, 2, \cdots T$. The standard form of multi-stage stochastic programming can be formulated as:

$$\min_{x \in X} \quad F(x_0, x_1, \cdots x_{T-1})$$

$$= \mathbb{E}\big[Q(x_0, x_1, \cdots, x_{T-1}, \xi_1, \xi_2, \cdots \xi_T)\big]$$

$$s.t. \quad x_t \in \mathcal{F}_t \qquad\qquad\qquad\qquad \forall\, t \in T$$

where x_t represent the decision vector, chosen at stage/time t, and $\xi_t, t = 1, 2, \cdots, T$ represent a sequence of random variables with a specified probability distribution. In addition, $x_t \in \mathcal{F}_t$ is nonanticipativity constraint. $F_t \subseteq F_{t+1}$ means all information happening before time t should be kept to next time period $t + 1$. For example, the decisions made today are influenced by previous decisions and outcomes.

Probabilistic dynamic programming is similar to multi-stage stochastic programming, as shown in Figure 1.1. The difference is the dynamic programming is based on a state system, and the information from the previous time period is not considered to make the current decision. When the probability distribution of the random variables is known, and decision is make to control the Markov process, this is known as a Markov Decision Process (MDP). The standard form of probabilistic dynamic programming can be formulated as:

$$\min_{x \in X} \quad F(x_0, x_1, \cdots x_{T-1})$$

$$= \mathbb{E}\big[Q_1(\zeta_1), Q_2(\zeta_2), \cdots, Q_T(\zeta_T)\big]$$

$$s.t. \quad \zeta_t = z_0 \qquad\qquad\qquad\qquad t = 0$$

$$\zeta_t = f_t(\zeta_{t-1}, x_{t-1}, \xi_{t+1}) \qquad\qquad t = 1, 2, \cdots, |T|$$

1.2 Review of Discrete Choice Models

Discrete choice models are used to explain or predict a choice from a set of two or more discrete (i.e., distinct and separable; mutually exclusive) alternatives. For example, a discrete

choice model may be used to analyze why people choose to drive, take the subway, or walk to work, or to analyze the factors causing people to pick one job over another. Techniques like logistic regression and probit regression can be used for empirical analysis of discrete choice. Assume person n has a choice set I, which includes all possible alternatives i.

- Choice y_{ni}: dummy variable, if $y_{ni} = 1$, then person n chooses alternative i
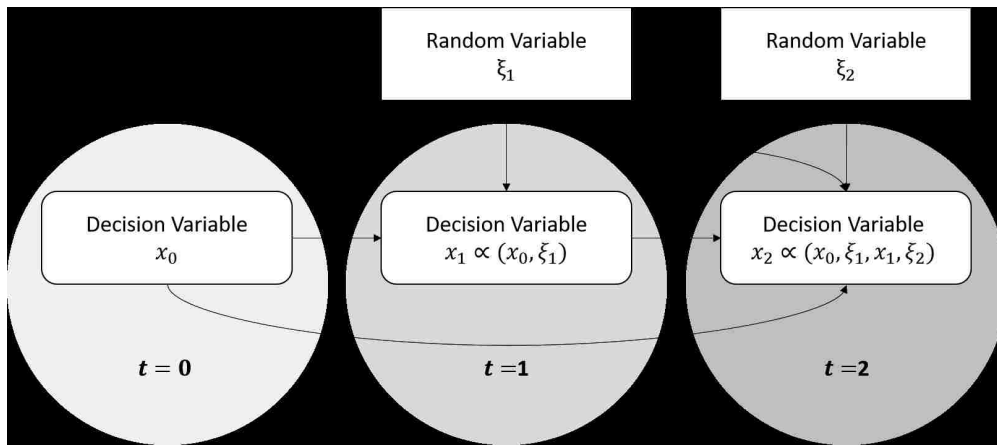
(a) Multi-stage Stochastic programming
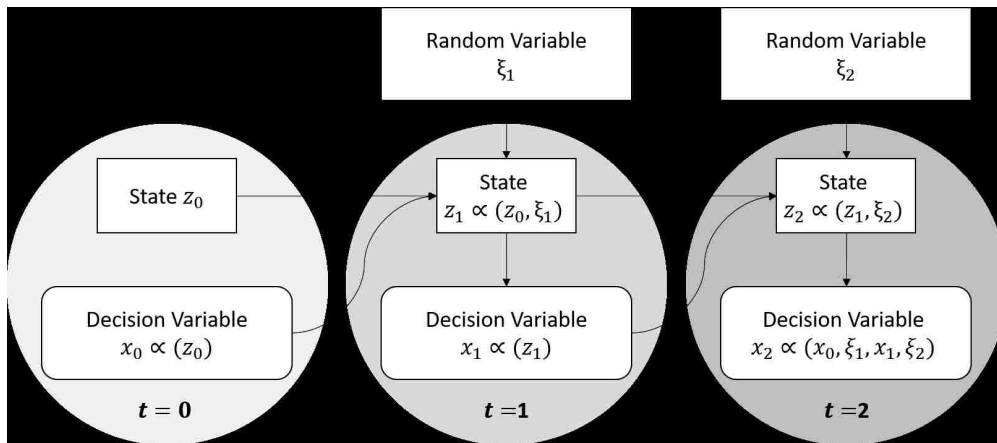
(b) Probabilistic Dynamic programming

Figure 1.1: Comparison between Multi-stage Stochastic programming and Probabilistic Dynamic programming

- Utility U_{ni}: the utility that person n obtains from choosing alternative i.

- Probability P_{ni}: $\Pr(y_{ni} = 1)$, the probability that person n will choose alternative i.

The utility depends on many factors, where some are observed z_{ni}, some are not observed ε_{ni}:

$$U_{ni} = \beta z_{ni} + \varepsilon_{ni}$$

Here, $z_{ni} = z(x_{ni}, s_n)$, where x_{ni} is the a vector of attributes of alternative i faced by person n and s_n is a vector of characteristics of person n. The behavior of the person is utility maximizing: Person n chooses the alternative that provides the highest utility.

$$y_{ni} = \begin{cases} 1 & U_{ni} > U_{nj} \quad \forall j \neq i \\ 0 & \text{otherwise} \end{cases}$$

The choice probability is

$$
\begin{aligned}
P_{ni} &= Pr(y_{ni} = 1) \\
&= Pr\left(\bigcap_{j \neq i} \left(U_{ni} > U_{nj}\right)\right) \\
&= Pr\left(\bigcap_{j \neq i} \left(\beta z_{ni} + \varepsilon_{ni} > \beta z_{nj} + \varepsilon_{nj}\right)\right) \\
&= Pr\left(\bigcap_{j \neq i} \left(\varepsilon_{nj} - \varepsilon_{ni} < \beta z_{ni} - \beta z_{nj}\right)\right)
\end{aligned}
$$

In the Discrete Choice Model, the random variable is following two types of distribution:

- Extreme Value Distribution – Type I Gumbel (Logit), with the probability density

function

$$f(x; \mu, \sigma, \xi = 0) = e^{-\frac{x-\mu}{\sigma}} e^{-e^{-\frac{x-\mu}{\sigma}}}$$

- Normal Distribution (Probit), with the probability density function

$$f(x; \mu, \sigma) = \varphi\left(\frac{x - \mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The random variable ε_{ni} is normalized, and then the probability density function of these distributions will be

$$f(\varepsilon_{ni}) = e^{-\varepsilon_{ni}} e^{-e^{-\varepsilon_{ni}}} \text{ and } \varphi(\varepsilon_{ni}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\varepsilon_{ni}^2}$$

Let us start with binary choice. The choice set of person n has two alternatives. The utility of each alternative U_{ni} depends on observed variable z_{ni} and unobserved/random variable ε_{ni}:

$$U_{ni} = \beta z_{ni} + \varepsilon_{ni}$$

The choice probability of alternative $i = 1$ is

$$P_{n1} = Pr(y_{n1} = 1) = Pr(U_{n1} > U_{n2})$$
$$= Pr(\beta z_{n1} + \varepsilon_{n1} > \beta z_{n2} + \varepsilon_{n2})$$
$$= Pr(\varepsilon_{n2} - \varepsilon_{n1} < \beta z_{n1} - \beta z_{n2})$$

6

If the random variables of each utility is iid, then the choice probability is based on joint probability distribution:

$$P_{n1} = \int_{\varepsilon_{n1}=-\infty}^{\varepsilon_{n1}=+\infty} \int_{\varepsilon_{n2}=-\infty}^{\varepsilon_{n2}=\beta z_{n1}-\beta z_{n2}+\varepsilon_{n1}} e^{-\varepsilon_{n1}} e^{-e^{-\varepsilon_{n1}}} \cdot e^{-\varepsilon_{n2}} e^{-e^{-\varepsilon_{n2}}} \, d\varepsilon_{n1} d\varepsilon_{n2}$$

$$= \int_{\varepsilon_{n1}=-\infty}^{\varepsilon_{n1}=+\infty} e^{-\varepsilon_{n1}} e^{-e^{-\varepsilon_{n1}}} \cdot \left(\int_{t_{n2}=+\infty}^{t_{n2}=exp(-(\beta z_{n1}-\beta z_{n2}+\varepsilon_{n1}))} e^{-t_{n2}} \left(-dt_{n2} \right) \right) d\varepsilon_{n1}$$

$$(\text{let } t_{n2} = e^{-\varepsilon_{n2}})$$

$$= \int_{\varepsilon_{n1}=-\infty}^{\varepsilon_{n1}=+\infty} e^{-\varepsilon_{n1}} e^{-e^{-\varepsilon_{n1}}} \cdot e^{-e^{-(\beta z_{n1}-\beta z_{n2}+\varepsilon_{n1})}} \, d\varepsilon_{n1}$$

$$= \int_{t_{n1}=+\infty}^{t_{n1}=0} e^{-t_{n1}} \cdot e^{-t_{n1} \cdot e^{-(\beta z_{n1}-\beta z_{n2})}} \left(-dt_{n1} \right) \qquad (\text{let } t_{n1} = e^{-\varepsilon_{n1}})$$

$$= \frac{1}{1 + exp(\beta z_{n1} - \beta z_{n2})}$$

$$= \frac{exp(\beta z_{n1})}{exp(\beta z_{n1}) + exp(\beta z_{n2})}$$

The normal distribution has the property

$$X \sim N(\mu_X, \sigma_X^2)$$

$$Y \sim N(\mu_Y, \sigma_Y^2)$$

$$X \pm Y \sim N(\mu_X \pm \mu_Y, \sigma_X^2 + \sigma_Y^2)$$

Then, the choice probability with normal distribution is $P_{n1} = \Phi(\beta z_{n1} - \beta z_{n2})$.

Multinomial choice has two types; one is multinomial choice without correlation among alternatives, while the other is multinomial choice with correlation among alternatives. The difference is whether it has correlation between alternatives. For extreme value distribution, the two types of multinomial choice model are as follows

- No attributes of the alternatives

$$P_{ni} = \frac{exp(\beta z_{ni})}{\sum\limits_{j=1}^{J} exp(\beta z_{nj})}$$

- Generalized nested logit

  Nests of alternatives are labeled $B_1, B_2, \cdots, B_K$. Each alternative can be a member of more than one nest:

$$P_{ni} = \sum_{k=1}^{K} P_{ni|B_k} \cdot P_{nk}$$

where the probabilities of nest k and alternative i given nest k are

$$P_{nk} = \frac{\sum_{j \in B_k} \left(\alpha_{jk}e^{V_{nj}}\right)^{1/\lambda_k}}{\sum_{l=1}^{K} \left(\sum_{j \in B_l}(\alpha_{jl}e^{V_{nj}})^{1/\lambda_l}\right)^{\lambda_l}}$$

$$P_{ni|B_k} = \frac{\left(\alpha_{ik}e^{V_{ni}}\right)^{1/\lambda_k}}{\sum_{j \in B_k} \left(\alpha_{jk}e^{V_{nj}}\right)^{1/\lambda_k}}$$

For normal distribution, the multinomial choice model is as follows:

$$P_{ni} = \int I(V_{ni} + \varepsilon_{ni} > V_{nj} + \varepsilon_{nj} \ \forall j \neq i) \, \phi(\varepsilon_n|\Omega) \, d\varepsilon_n$$

where some parameters are defined as

- $U_{ni}^r = V_{ni} + \varepsilon_{ni}$
- $I = 1$, if $U_{ni} > U_{nj}$; and $I = 0$, otherwise
- $\phi(\varepsilon_n|\Omega)$ is joint normal density function with mean 0 and covariance Ω

1.3 Outline of This Dissertation

This dissertation is motivated by real-world operations research problems in resource allocation. It aims at developing the optimal strategy to satisfy customer demand by solving the stochastic programming problem in a large-scale network. Several mathematical models and efficient algorithms are developed to deal with the exponential computation complexity of each research project. The structure of this dissertation is as follows: In Chapter 2, we study the two-stage stochastic programming physician location problem with patients having discrete choices. Chapter 3 investigates decision-dependent multistage stochastic programming for the information cascade problem under user discrete choice. In Chapter 4, we study the multi-stage stochastic programming job-shop Problem in Semiconductor Manufacturing. Chapter 5 concludes the dissertation.

CHAPTER 2: DECISION-DEPENDENT STOCHASTIC PROGRAMMING FOR PHYSICIAN ALLOCATION CONSIDERING PATIENTS' DISCRETE CHOICES

In this chapter, we study a stochastic facility location problem considering customer preference. The model is motivated by a physician scheduling problem in local clinics. The model aims to improve hospital efficiency by allocating the hospital resource (sending physicians to local clinics), as well as to match the patient preference and maximize patient satisfaction. A two-stage stochastic programming model is proposed for the physician/clinic facility location and patient assignment problem, where the patient preference is considered as endogenous uncertainty. Instead of being prefixed, scenario probabilities are defined through discrete choice theory, considering various features of patient preference. The two-stage stochastic programming model is computationally intractable due to the exponentially growing number of scenarios. To solve the model, this paper designs hybrid algorithms via the combination of the Large Neighborhood Search and Tabu Search to solve the location problem in the first stage and Sample Average Approximation to estimate the value function of the second stage. Computational experiments show that the proposed hybrid algorithms can outperform existing hill-climbing techniques, such as Guided Local Search and Gradient Descent method, in terms of both solution quality and computational time.

2.1 Introduction

Physician shortages are continuously increasing in both primary and specialty care. In the 2018 updated report from the Association of American Medical Colleges (AAMC), Dall et al.

[17] show a projected shortage of between 42,600 and 121,300 physicians by the end of the next decade in Figure 2.1. The main reason leading to the huge shortage is that physician demand will grow faster than supply. Typically, physicians can be categorized to two types, namely, primary care and non-primary care physicians. The primary care shortage is between 14,800 and 49,300 physicians, while the non-primary care shortage is between 33,800 and 72,700 physicians. The shortage of physicians is a critical problem in health care systems; besides, there is another challenge in matching physicians with patients. In reality, patients facing higher barriers to accessing care (racial and ethnic minorities, the uninsured, and those living outside metropolitan areas) have lower health care utilization than those patients with fewer barriers to access. Figure 2.2 shows that it will require more physicians than the equivalent care utilization (Scenario 1), if we consider the care utilization of different pairs

(a) Total Physicians

(b) Primary Care Physicians

(c) Non-Primary Care Physicians
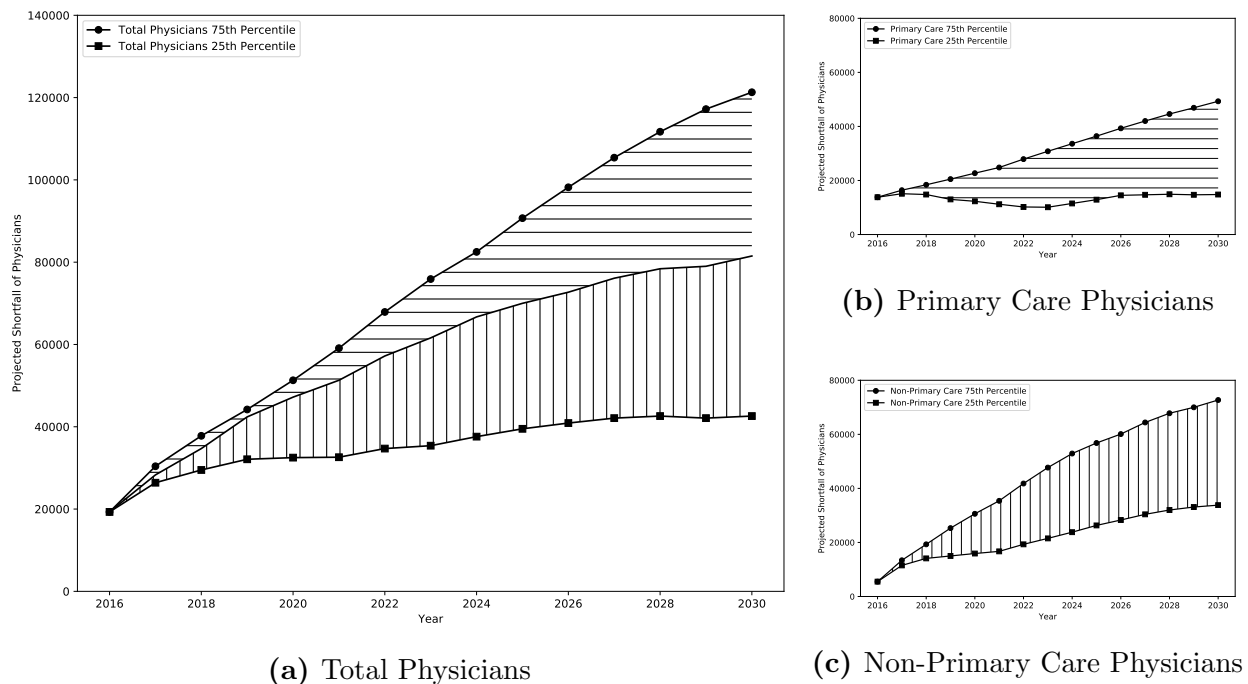
Figure 2.1: Projected Physician Shortfall Range, 2016-2030. Reprinted from "The Complexities of Physician Supply and Demand: Projections from 2016 to 2030," by IHS Markit Ltd., 2018. Copyright (2018) by Association of American Medical Colleges.
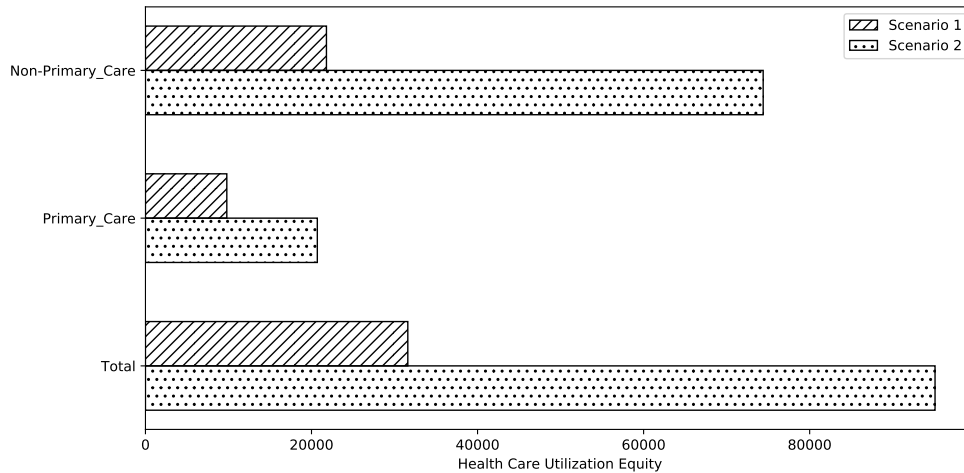
Figure 2.2: Additional Physician Demand to Achieve health care Utilization Equity, 2016. Reprinted from "The Complexities of Physician Supply and Demand: Projections from 2016 to 2030," by IHS Markit Ltd., 2018. Copyright (2018) by Association of American Medical Colleges.

of physician and patient are not equivalent (Scenario 2) based on the AAMC health care utilization equity (HCUE) analysis model. Due to the different patient barriers, more and more medical procedures are moving into outpatient facilities, which reduces complications and allows patients to return home sooner [1]. With the growing outpatient care, however, how to match the physician supply and patient demand in Clinically Integrated Networks (CINs) is becoming a big issue. As a network problem, the physician allocation in CINs can be modeled as a facility location problem.

In this paper, we present two-stage stochastic facility location models while considering the random discrete choices of the patients due to their preferences. These models directly address physician scheduling problems among various local clinics. In recent years, the physician shortage and resource limits have created tough problems to provide necessary health care access to patients. One way to alleviate this burden is to send physicians from central hospital(s) to local clinics, as in the case of Veterans Affairs (VA) hospitals [41].

12

Instead of the central hospital, patients can visit local clinics close by to avoid congestion and unnecessary trips. However, patients' decisions on which clinic they choose to visit are not certainly known to the scheduler in charge of the physician allocation. From the viewpoint of the scheduler, this can be modeled as a stochastic facility location problem, where the demand for each clinic or physician is random. It is challenging to model these demand uncertainties while assuming prefixed distributions. This is due to patients' different preferences toward the physician, the travel distance to clinics, patient race and ethnicity, and so on. The distributions of random demands at various locations are intrinsically endogenous, corresponding to the whereabouts of physicians. We can borrow the Discrete Choice Theory that we first introduced in [44] to model the decision-dependent uncertain demands, since each patient has a discrete number of selections of clinics or physicians. Then, the total demand of a particular location can be calculated by aggregating the patients choosing to visit this location.

2.1.1   Literature Review

The study of physician allocation and patient assignment in health care is a growing area of research. Here, we only focus on the papers considering resource allocation and customer choice in health care, which is most relevant to our research. Relman [52] finds the utilization of health care system is determined largely by the collective decision of physicians. The business model for health care resources in human terms is identified by Kluge [34]. In this paper, the patient becomes a service consumer, the physician is a service provider, and the physician–patient relationship is the key question for health care resource allocation. Nicholson and Levy [48] consider the physician allocation strategy for large-scale medical services to reduce the burden of health care costs by increasing efficiency. Barz and Rajaram [6] develop the approximate dynamic programming by formulating the MDP for the patient

admission problem with multiple resource constraints. Operations research methods, such as the integer programming model, have been introduced to solve the physician allocation and patient assignment problem with the capacity constraint in [35, 41]. Balasubramanian builds a two-stage capacity allocation model with an uncertain patient demand in which he assumes the distribution of uncertainty is known [4]. Wang formulates a two-server network model to maximize the patient benefit, combining of analytical calculation and simulation-based optimization [61].

The prescriptive mathematical model was developed to assist the facility location decision, which is used to find optimal solution. A typical normative model is the mathematical programming model [50]. In general, the classic facility location problems consider the following elements: characteristics of the facility, characteristics of the served population, and objectives. An important factor of facility characteristics is spacing. According to the facility spacing property, the facility location can be categorized into capacitated and uncapacitated problems [59]. For characteristics of the served population, customer demand is a key factor, which may be splittable or unsplittable [36]. The facility location problem is an NP-hard problem, and it can be reached by reduction from the set-packing-covering-partitioning problems [31]. The neighborhood search procedure is one of the earliest heuristic/approximation algorithms in the facility location problem [37]. The algorithm has exhibited good practical performance and proved to be a good approximation algorithm with guaranteed performance in polynomial time [14].

Many facility location problems involve strategic decisions that must hold for some considerable time. In such cases, it is important to embed uncertainty in the models. To this end, many research endeavors based on stochastic programming and robust optimization have been undertaken. The common uncertainty of a facility location problem includes demand levels, travel time or cost for supplying the customers, location of the customers, presence or

absence of the customers, and price for the commodities [16]. There are two popular ways of handling the uncertainty, namely, stochastic programming and robust optimization [56]. If the information is probabilistic and the uncertain parameters can be represented through random variables, then stochastic programming models and methods can be used to deal with the problem. In those models, the objective is usually to minimize the expected cost [45, 54]. If no probabilistic information is available but some ranges of uncertainties are known, robust optimization can be used for evaluating the performance of the system. In those models, two classical objectives are often considered–minmax cost and minmax regret [5, 28].

Patient demand is the main concern for decision makers in the health care facility location problem. Early research on patient demand shows that the assignment between the physicians and the local clinics strongly influences the patient's choice [15, 21]. Lawton uses the conditional logit model to estimate the probability of an individual physician or patient choosing a specific hospital [8], following McFadden's random utility approach [44, 43]. Atlas, et al. measure physician performance by categorizing all patients seen in a large primary care network [3]. In Güneş et al. [29]'s paper, the researchers show how to match patient and physician preference to arrive at the central planner's objectives, which are maximizing the coverage–the percentage of assigned patients–and minimizing the average patient travel distance. This study case is from Sakarya, Turkey, and shows the patient's preference is highly correlated with travel distance. Griffin develop a model to determine the best location and number of new Community Health Centers (CHCs) to maximize the coverage of the weighted demand in the population within a limited budget and facility capacity [26].

The previous studies have realized the importance of the patient's preference and the physician's characteristics in providing high-quality health care services, while there is no work on matching the customer (patient) choice and provider (physician/clinic) availability. The

mismatching between the demand of customers and the supply of providers may reduce customer satisfactions.

2.1.2 Our Contributions

To address the mismatching of customer and provider in health care, we model the problem as a facility location problem with the maximal patient satisfaction. Since this problem involves three elements–clinic, physician,patient–we decompose this problem to two sub-problems, namely, the physician allocation problem and the patient assignment problem. In this paper, we propose a large-scale planning model through a two-stage, decision-dependent, stochastic programming approach. The stochastic formulation considers endogenous uncertainty, which is represented by the discrete probability distribution of patient preference.

The computation challenge of our model is decision-dependent uncertainty with the non-linear probability mass function and exponentially increasing number of scenarios. Goel and Grossmann [22] define the stochastic programming model with decision-dependent uncertainty and use the Lagrangian Duality based Branch and Bound (LDB&B) algorithm to solve the model. The necessary condition to apply this algorithm by relaxing the non-anticipativity constraints is that the scenarios s and s' should either differ exclusively in their multiple parameters with exogenous uncertainty, or one parameter with endogenous uncertainty. Due to the nonlinear probability of the uncertainty, intuitively, an initial attempt to solve our model by LDB&B involves linearizing the non-linear terms. The probability constraints can seem to be non-anticipativity constraints, since the probability distribution of patient preference is the same for all scenarios. Before applying the Lagrangian relaxation of the probability constraints, we check the necessary condition of LDB&B within the linearization model. The scenarios are jointed with each other based on the realization for

the choice of patient preferred physician, with the result that LDB&B is not applicable in our model. In addition to focusing on the stochastic programming, we look at the property of the decision variables. Since our model is based on the facility location problem, the main decision is to allocate different physicians to the proper local clinic, which is a typical combinatorial optimization problem. Zlochin et al. [66] summarize some model-based search algorithms that can be used on our model, such as the stochastic gradient ascent and estimation of distribution algorithms. Fu [19] introduces simulation-based methods for estimating gradients in the optimization problem. Based on the property of combinatorial optimization, we design two problem-specific hybrid algorithms to solve our model; one is a combination of the guided local search and gradient decent method, and the other one is a combination of the Tabu search and large neighborhood search.

We summarize the contributions of this project as follows:

- We introduce the discrete choice model in the facility location and assignment problem with customer preference and address its health care applications;

- We develop practical algorithms for solving the two-stage stochastic programming problem with endogenous uncertainty;

- To avoid dealing directly with the exponentially growing number of scenarios, we take advantage of implicit Sample Average Approximate with Monte Carlo Simulation; and

- We apply the sensitivity analysis while increasing the penalty of unpreferred assignment to shown the percentage change between the assignment types.

The rest of this chapter is structured as follows: We briefly describe the the physician allocation and patient assignment problem, and provide three models with different assumptions in section 2. In section 3, we design two hybrid heuristic algorithms to solve the problem and compare the computation results. Section 4 concludes the study and discusses options

17

for future field-work and implementation.

2.2 Mathematical Models

We formulate the health care problem of multiple physicians as a two-stage stochastic programming model consisting of *physician allocation* followed by *patient assignment*. In the first stage, we assume that, for each physician l, there are several outreach clinics to be allocated. The problem is deciding which outreach clinic j the physician l needs to go to. The capacities of the clinic and physician are unknown until the allocation is decided. In the second state, following the allocation decision, the problem is to decide how many patients from group k are assigned to physician l within the limits of the clinic and physician capacities.

In economics, Utility Maximization is when, in making a purchase decision, a consumer attempts to obtain the greatest value possible while spending the least amount of money. His or her objective is maximizing the total value derived from the available money. In our problem, we have the same idea that the objective is maximizing the patient's total utility derived from the proper physician allocation and patient assignment. The patient's total utility is define as Health Care Utilization by Carrasquillo [10]. For different physician assignment, patients have different levels of Personal Utility (PU). No assignment lead to a PU of 0; we call this *dissatisfaction*. We develop three following models to achieve Maximization on Health Care Utilization: the Basic Model (BP), Stochastic programming Model with Equally Likely Scenarios (SPEL), Decision-dependent Stochastic programming with Discrete Choice Uncertainty (SPDC). The purpose of these three models is the same, to maximize the patient total utility. The physician allocation decision of BP assumes the patient always prefers to be assigned to the physician with a higher PU. However, the two stochastic models,

SPEL and SPDC consider service quality more than BP does by adding the uncertainty of patient preference. The difference between SPEL and SPDC is the assumption of probability distribution.

2.2.1 The Basic Model

The basic model considers the arrangement between the three factors (patient, physician and clinic) of the health care system without uncertainty. There are two types of decision: one is the physician allocation y_{lj} which are binary variables that decide which clinic does the physician visit; another one is patient assignment x_{ki} which are continues variables that decide which physician and how many patient are assigned. In the basic model, the physician allocation and patient assignment does not happen in time sequence, which means the central hospital can make these two types of decision at the same time. Our model considers a short-term case, such as daily health care services. To avoid the waste of available working hours caused by the physician traveling, we assume each physician is only allowed to allocate once,

(a) Before Decision   **(b)** After Decision
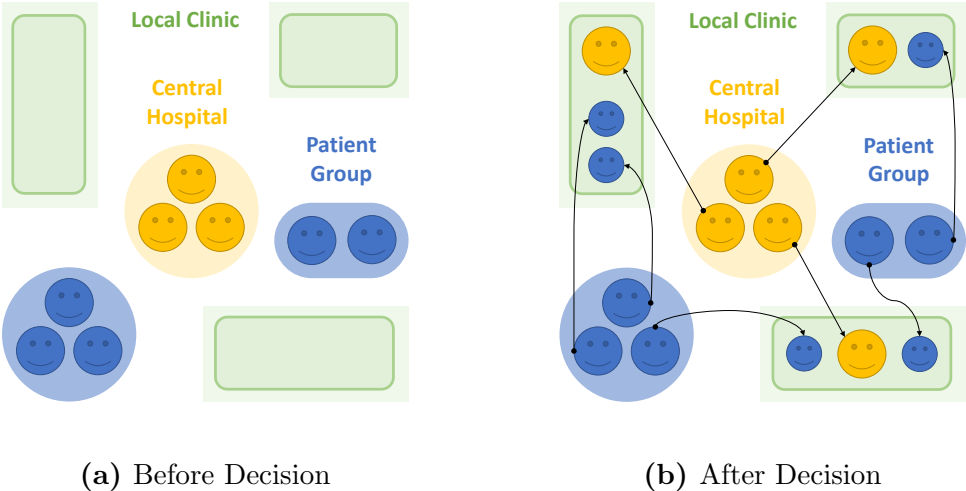
Figure 2.3: The Physician Allocation and Patient Assignment in Basic Model

19

Table 2.1: Notation of Basic Model

Symbol	Definition
	Indices and Sets
$j \in J$	outreach clinic
$k \in K$	patient group
$L \in L$	physician
	Parameters
d_k	the total people size of patient group k
g_{lj}	the available surgery people size for physician l, if it is assigned to outreach clinic j
u_{klj}	the utility of patient k, if it is assigned to physician l and visit outreach clinic j
	Decision Variable
y_{jl}	binary variable, the allocation decision between physician l and outreach clinic j
x_k^0	continuous variable, the number of unsatisfied patient in group k
x_{kl}^1	continuous variable, the number of patient in group k assigned to physician l

which means the physician only can visit and work in one clinic. It is a Capacitated Facility Location Problem (CFLP), that the availability of the patient assignment is restricted by the physician working hours and clinic spacing. Let consider a small case, we have 2 patient locations, 3 local clinics and 3 physician in the central hospital, as shown int Figure 4.2. Both physician and patient need to visit the local clinic to provide services or take treatments. Due to the capacity constraint, the patient group may be assigned to multiple physician separately. The notation of basic model is shown in Table 2.1. The deterministic model of

basic model is shown below:

$$[\textbf{BP}] \quad \max_{x,y} \quad \sum_{k \in K} \sum_{l \in L} \sum_{j \in J} \left(u_{klj} \cdot y_{jl} \cdot x_{kl}^1 \right) + m_0 \cdot \sum_{k \in K} x_k^{0s} \tag{2.1a}$$

$$s.t. \quad \sum_{j \in J} y_{lj} = 1 \qquad\qquad \forall\, l \in L \tag{2.1b}$$

$$x_k^0 + \sum_{l \in L} x_{kl}^1 = d_k \qquad\qquad \forall\, k \in K \tag{2.1c}$$

$$\sum_{k \in K} x_{kl}^1 \leq \sum_{j \in J} \left(g_{lj} \cdot y_{lj} \right) \qquad\qquad \forall\, l \in L \tag{2.1d}$$

$$\sum_{k \in K} \sum_{l \in L} \left(y_{lj} \cdot x_{kl}^1 \right) \leq h_j \qquad\qquad \forall\, j \in J \tag{2.1e}$$

$$y_{jl} \in B, \quad x_k^0 \in R^+, \quad x_{kl}^1 \in R^+$$

The objective (2) is to maximize the quality of the patient assignment measured by the patient utility respect to the certain physician and clinic. Constraint (2.1b) tells that each physician is only allowed to allocate to one clinic. The constraint (2.1c) means all patients should be assigned or unsatisfied. The constraint (2.1d) and (2.1e) means the capacity of physician and clinics is limited. For any physician $l \in L$ in constraint (2.1d), $\sum_{j \in J} \left(g_{lj} \cdot y_{lj} \right)$ is the maximal working time which may varies by physician allocation variable y_{lj}. For any clinic $j \in J$ in constraint, the total assigned patient of all location k respected to the allocated physician (2.1e).

2.2.2 The Stochastic programming Model with Equally Likely Scenarios

Everyone has personal preferences, and these may affect the choices we make, while the basic model (BP) doesn't consider the patient choices. It is important for health care systems to recognize that patient preferences are associated with how individuals use health

21

(a) Physician Allocation Decision

(b) Patient Preferred Assignment

(c) Patient Unpreferred Assignment

(d) Utility Maximization
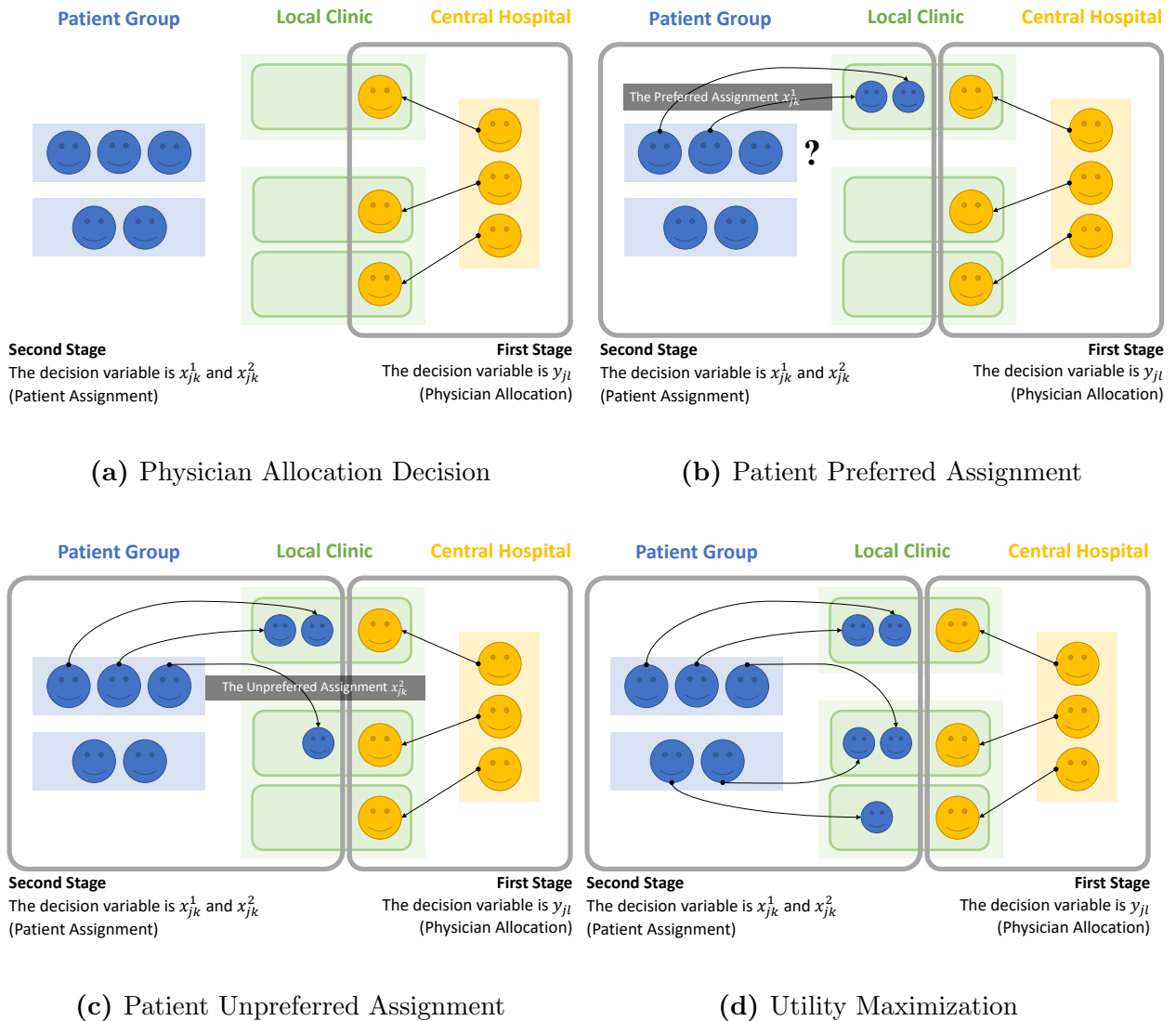
Figure 2.4: The Physician Allocation and Patient Assignment in Stochastic programming Model

care services, such as giving them choices to choose the physician [42]. So we develop the stochastic programming model using patient choice as uncertainty, which is following the discrete uniform distribution, as shown in Figure 2.4.

In this model, we involve the new concept *Patient Preference for Physician*, which is denoted as A. a_{kl} is the element of row k and column l in matrix A. If the patient k would like to

choose physician l, this physician is considered as *preferred* ($a_{kl} = 1$) and other physician l' is *unpreferred* ($a_{kl} = 0$). The stochastic patient preference is approximated by a scenario tree, which is constructed according to the following steps:

- *Individual Probability*: For any patient k, there are $|L|$ options to choose the preferred physician l. Since we have the assumption that the uncertainty is following the discrete uniform distribution, as shown in n Figure 2.5a, the probability of patient choice is

$$p_{kl} = \text{Prob}(a_{kl} = 1) = 1/|L| \tag{2.2a}$$

- *Scenario Probability*: For each scenario s, the uncertainty size is $|K|$, since each patient has the difference choice, ie. preferred physician. The patient choice is independent, so the scenario probability is the joint probability of individual probability, as shown in Figure 2.5b and Figure2.5c.

$$P^s = \prod_{k \in K} \text{Prob}(a_{kl} = 1) = \prod_{k \in K} \left(\sum_{l \in L} p_{kl} \cdot a_{kl} \right) = \frac{1}{|L|^k} = \frac{1}{|S|} \tag{2.2b}$$

(a) Individual Probability
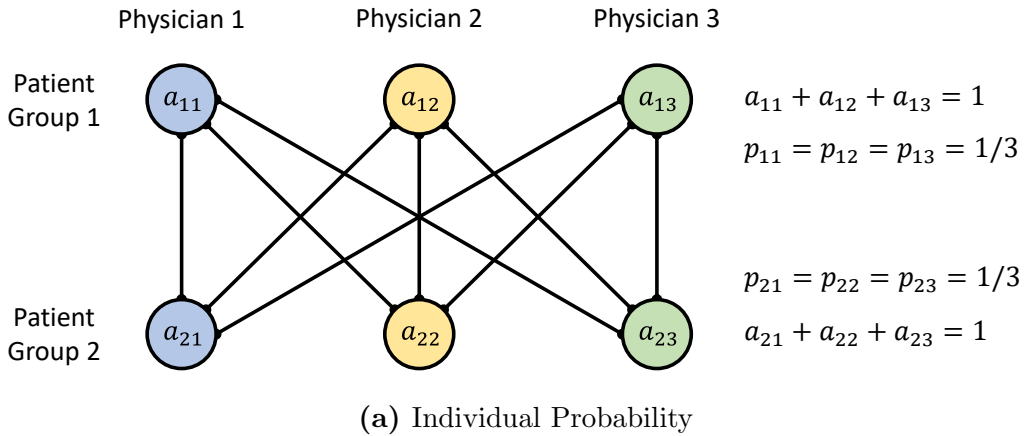
Figure 2.5: Scenario Tree
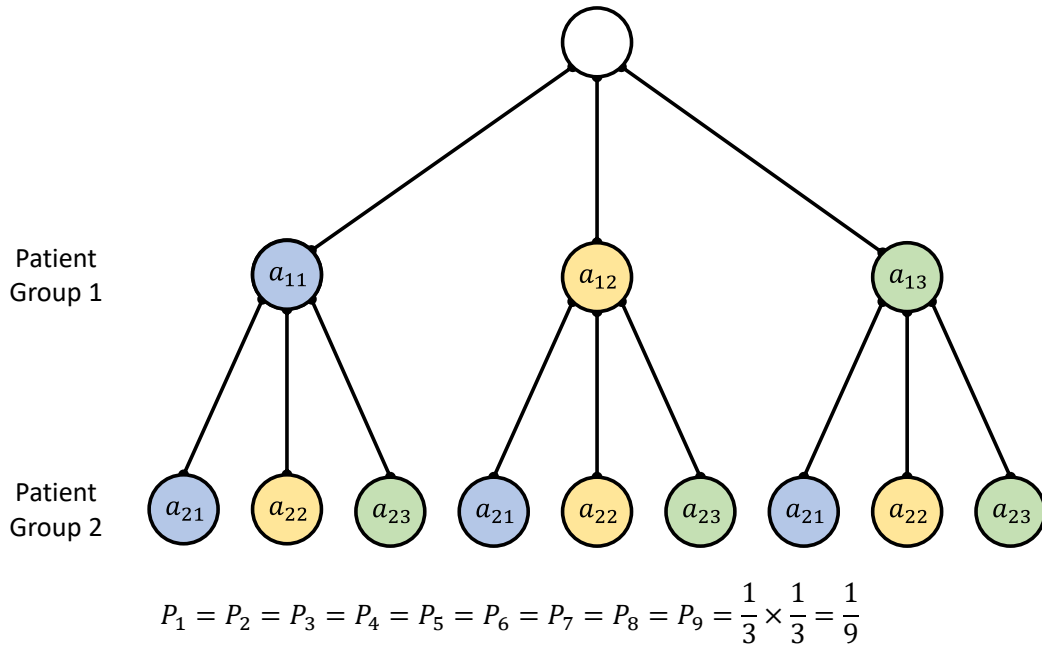
(b) Scenario Probability
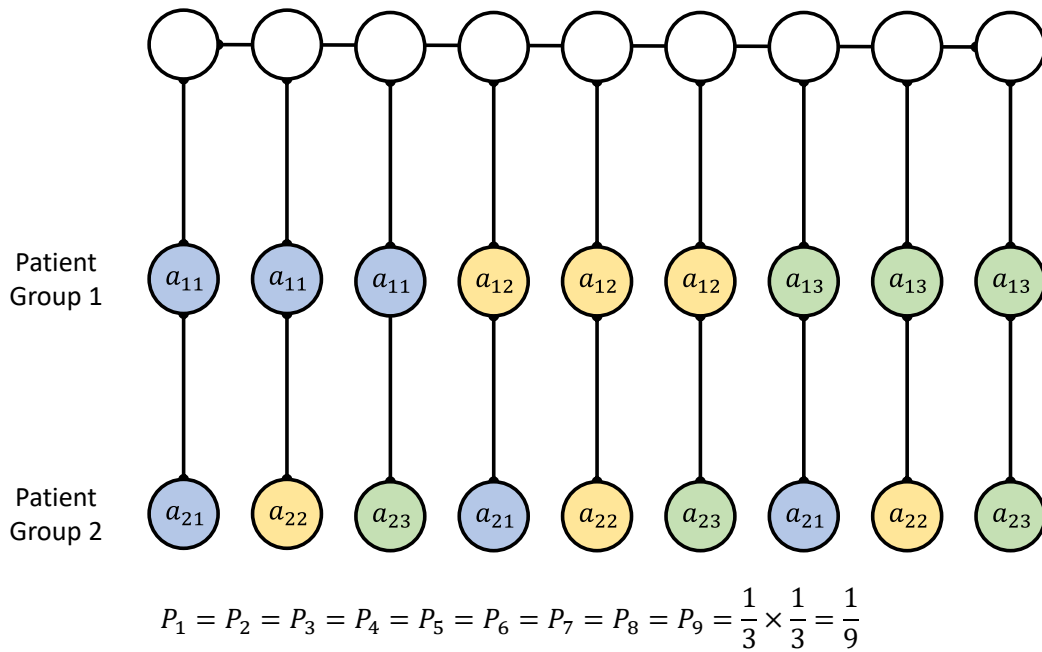
(c) Deterministic Equivalent

Figure 2.5: Scenario Tree (cont.)

Table 2.2: Notation of Stochastic programming Model

Symbol	Definition
	Indices and Sets
$s \in S$	scenario
	Parameters
m	the penalty if the patient is assigned to unpreferred physician
a_{kl}^s	the choice between patient k and physician l
	Decision Variable
x_k^{0s}	continuous variable, the number of unsatisfied patient in group k
x_{kl}^{1s}	continuous variable, the number of patient in group k assigned to preferred physician l
x_{kl}^{2s}	continuous variable, the number of patient in group k assigned to unpreferred physician l

Based on the patient choices, the patient assignment is separate to two variables. The additional notation is shown in Table 2.2.

If patient k choose physician l, we will try to satisfy its preference firstly. But sometimes there is no space for preferred physician l to accept patient k, we will try to reassigned patient k to another unpreferred physician l' depending on the utility. In that case, there is discount factor on patient utility with unpreferred assignment. The deterministic equivalent is reformulate as below:

$$[\textbf{SPEL}] \quad \max_{x,y} \ \mathbb{E}\left[Q\left(x,y;\omega\right)\right] = \sum_{s \in S} P^s \cdot Q^s\left(x,y\right) = \frac{1}{|S|} \cdot \sum_{s \in S} Q^s\left(x,y\right) \tag{2.3a}$$

$$s.t. \ \ Q^s\left(x,y\right) = \sum_{k \in K}\sum_{l \in L}\sum_{j \in J}\left[u_{jkl} \cdot y_{jl} \cdot \left(x_{kl}^{1s} + m_2 \cdot x_{kl}^{2s}\right)\right] + m_0 \cdot \sum_{k \in K} x_k^0$$

$$\forall \ s \in S \tag{2.3b}$$

$$\sum_{j \in J} y_{jl} = 1 \qquad\qquad \forall\, l \in L \qquad\qquad (2.3c)$$

$$x_k^{0s} + \sum_{l \in L} \left(x_{kl}^{1s} + x_{kl}^{2s} \right) = d_k \qquad\qquad \forall\, s \in S,\, k \in K \qquad\qquad (2.3d)$$

$$\sum_{k \in K} \left(x_{kl}^{1s} + x_{kl}^{2s} \right) \leq \sum_{j \in J} \left(g_{jl} \cdot y_{jl} \right) \leq 0 \qquad\qquad \forall\, s \in S,\, l \in L \qquad\qquad (2.3e)$$

$$\sum_{k \in K} \sum_{l \in L} y_{jl} \cdot \left(x_{kl}^{1s} + x_{kl}^{2s} \right) \leq h_j \qquad\qquad \forall\, s \in S,\, j \in J \qquad\qquad (2.3f)$$

$$x_{kl}^{1s} \leq a_{kl}^s \cdot d_k \qquad\qquad \forall\, s \in S,\, k \in K,\, l \in L \qquad (2.3g)$$

$$x_{kl}^{2s} \leq (1 - a_{kl}^s) \cdot d_k \qquad\qquad \forall\, s \in S,\, k \in K,\, l \in L \qquad (2.3h)$$

$$y_{jl} \in B, \quad x_{kl}^{1s} \in R^+, \quad x_{kl}^{2s} \in R^+, \quad x_k^{0s} \in R^+$$

Objective function (2) is maximizing the expected total utility considering all the scenarios. The total utility of each scenario is defined in equation (2.3b), which is similar to objective function (2) of Basic Model. Constraints (2.3c - 2.3f) are similar to constraint (2.1b - 2.1e) of Basic Model. The difference is that we separate the assignment to two types: preferred and unpreferred. Due to the patients' preference, there are two additional constraints (2.3g, 2.3h) which is used to distinguish the assignment types. The preferred assignment should be followed patient's preference, as shown in (2.3g). The unpreferred assignment is only allowed when the preferred physician doesn't accept the patient for the reason of limited capacity, as shown in (2.3h).

2.2.3   The Decision-dependent Stochastic programming with Discrete Choice Uncertainty (SPDC)

The previous stochastic model (SPEL) considers only the exogenous uncertainty where patients' preferences are not related to physician allocation. In the real world, if health care

planners knew more about patients' health-related preferences, the provided health care would most likely be more effective, and closer to the individuals' desires. in reality, the characteristics of physicians and locations of clinics also have effect on the patient preference. For instance, if a physician is allocated to a remote area, even through the reputation of this physician is good, the patient will unlikely visit there due to the long travel distance.In such case, interaction between the decisions and uncertainty would exist in many cases and thus the endogenous uncertainty must be included. In [65, 64], the probability distributions of uncertain parameters is impacted by optimization decisions. In our uncertainty definition, each patient has the discrete random choice on preferred physician. To describe patient choices between multiple physician, we implement an economics choices prediction model in our model, ie. Discrete Choice Model (DCM). Since the choice of preferred physician is independent that fulfilled with multinomial choice without correlation among alternatives, we use multinomial logit (MNL) model of DCM to find the probability of the patient preference.

$$p_{kl}(y) = \frac{\sum\limits_{j \in J} \left(e^{u_{jkl}} \cdot y_{jl} \right)}{\sum\limits_{\tau \in L} \sum\limits_{j \in J} \left(e^{u_{jk\tau}} \cdot y_{j\tau} \right)} \qquad \forall k, l$$

The distribution of patient preference is depended on physician allocation which means the uncertainty is endogenous (decision-dependent uncertainty).

$$\max_{x,y} \quad \mathbb{E}_s \left[Q(x, y; s) \right] = \sum_{s \in S} P^s(y) Q^s(x, y) \tag{2.4a}$$

$$\text{s.t.} \quad P^s(y) = \prod_{k \in K} \prod_{l \in L} [p_{kl}(y)]^{a_{kl}^s} \qquad \qquad \forall s \in S \tag{2.4b}$$

$$p_{kl}(y) = \frac{\sum\limits_{j \in J} \left(e^{u_{jkl}} \cdot y_{jl} \right)}{\sum\limits_{\tau \in L} \sum\limits_{j \in J} \left(e^{u_{jk\tau}} \cdot y_{j\tau} \right)} \qquad \qquad \forall k \in K, l \in L \tag{2.4c}$$

Model (SPEL) Constraint (2.3b - 2.3h)

Due to the dicrete choice model, model (SPDC) is non-linear, non-convex problem, which does not has the existed algorithm to solve it. The easiest method is prefixing the probability to avoid the non-linear probability function, while it is unsuccesful method. The following theorem shows the relationship between the optimal solution of model (SPDC) and the optimal solution of stochastic programming with prefixed probability, which tells why we cannot use this prefixing method.

Theorem 1. *If the distribution of scenarios is fixed based on the optimal solution of model (SPDC), the new optimal solution of the stochastic programming with exogenous uncertainty may be not feasible in the original model.*

Proof. We consider the model (SPDC) in the generic form (SPGF),

$$[\textbf{SPGF}] \quad \max_{\boldsymbol{x},\boldsymbol{y}} \quad z(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{p}^\mathsf{T}\boldsymbol{U}\boldsymbol{x} \tag{2.5a}$$

$$s.t. \quad \boldsymbol{p} = f(\boldsymbol{y}) \tag{2.5b}$$

$$g(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{h} \tag{2.5c}$$

$$\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}$$

where $\boldsymbol{p}$ is the vector of scenario probability, $\boldsymbol{U}$ is the matrix of patient utility. In the right hand side of equation (2.5b), f is the probability mass function with physician allocation $\boldsymbol{y}$. The equality constraint (2.5c) is the capacity constraints, either physician and clinic. Set $\mathcal{X}$ and set $\mathcal{Y}$ is the other constraints that only obtain the variable $\boldsymbol{x}$ or $\boldsymbol{y}$, ie. patient choice and unique allocation. The optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ can be presented below,

$$(\boldsymbol{x}^*, \boldsymbol{y}^*) = \operatorname*{arg\,max}_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y}} \left\{ z(\boldsymbol{x},\boldsymbol{y}) | \boldsymbol{p} = f(\boldsymbol{y}), g(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{h} \right\} \tag{2.6a}$$

Consider the nonempty polyhedral set $\mathcal{P} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{p} = \boldsymbol{p}^* \right\}$ where $\boldsymbol{p}^* = f(\boldsymbol{y}^*)$. Instead of constraint (2.5b), solve the model (SPGF) within the polyhedral set $\mathcal{P}$, then $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$ is the optimal solution of this stochastic programming with exogenous uncertainty,

$$(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ) = \arg\max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}} \left\{ z(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{p} = f(\boldsymbol{y}^*), g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{h} \right\} \tag{2.6b}$$

We prove this theorem by contradiction. Suppose the solution $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$ is a feasible solution in model (SPGF), then

$$z(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ) \leq \max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}} \left\{ z(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{p} = f(\boldsymbol{y}), g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{h} \right\} = z(\boldsymbol{x}^*, \boldsymbol{y}^*) \tag{2.7}$$

Since $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ fall in the polyhedral set $\mathcal{P} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{p} = f(\boldsymbol{y}^*) \right\}$ and is validated with constraint $g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{h}$, then it is feasible solution of model with fixed probability $\max_{\boldsymbol{x} \in X, \boldsymbol{y} \in Y} \left\{ z(\boldsymbol{x}, \boldsymbol{y}) \right.$ $\left| \boldsymbol{p} = f(\boldsymbol{y}^*), g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{h} \right\}$. Due to the property of optimality,

$$z(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ) = \boldsymbol{p}^{*\intercal} \boldsymbol{U} \boldsymbol{x}^\circ \geq \boldsymbol{p}^{*\intercal} \boldsymbol{U} \boldsymbol{x}^* = z(\boldsymbol{x}^*, \boldsymbol{y}^*) \tag{2.8}$$

When $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ) \neq (\boldsymbol{x}^*, \boldsymbol{y}^*)$ and there is unique optimal solution in model with fixed probability $\max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}} \left\{ z(\boldsymbol{x}, \boldsymbol{y}) | \boldsymbol{p} = f(\boldsymbol{y}^*), g(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{h} \right\}$. The objective function value of $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$ is strictly less then the objective function value of $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$. The relationship between solution $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$ and solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ from our assumption in inequality (2.7) is contradicted to the fact in (2.8). Thus, $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ)$ is infeasible solution in model (SPDC) when $(\boldsymbol{x}^\circ, \boldsymbol{y}^\circ) \neq (\boldsymbol{x}^*, \boldsymbol{y}^*)$. $\square$

2.3 Solution Approaches

The physician allocation problem is combinatorial optimization problem and the patient assignment is based on the uncertain patient preference. According the model structure that first stage binary variables and second stage continuous variables, the integer L-shaped method is the traditional method to solve the mix-integer stochastic programming. Lagrangian relaxation is a method to handle the large-scale problem. While the decision-dependent uncertainty is a difficulty that we cannot use these two methods. In this section, we propose the approximation algorithms to solve the health care planning problems at reasonable computational costs.

2.3.1 Sample Average Approximation

Firstly, we consider large-scale problem size caused by the scenario size. This difficulty happens in the computation of second stage. The scenario size $|S| = |L|^{|K|}$ which is exponential growth. Since the exogenous uncertain parameter of scenarios cannot be differ exclusively with each other, Lagrangian relaxation is not able to apply in our model. To handle the large amount scenarios, Kleywegt, Shapiro, and Mello [33] design the new efficiency method, sample average approximation (SAA) method. Different to the original SAA, we aim to solve the problem in reasonable time, so the sample size is pre-defined with the certain confidence level and confidence interval instead updating by the estimated optimality gap. The modified SAA is shown in Algorithm 1.

There is another difficulty caused by the decision-dependent uncertainty, that is nonlinear probability function which transit the allocation decision from first stage to the probability distribution of patient choice in second stage. The basic logic to solve this problem is variable

---

Algorithm 1 Sample Average Approximation (SAA)

1: Initialization: given confidence level (CL), the significance level $\alpha = 1 - \text{CL}$ and confidence interval (CI)

2: **for each** patient index $k \in \mathcal{K}$ **do**   ▷ Calculate sample size for each patient choice

3:      Calculate the variance σ_k^2 and the mean μ_k

4:      Define the sample size N_k, that the interval $\left(\mu_k - \dfrac{z_{\alpha/2} \cdot \sigma_k}{\sqrt{N_k}}, \mu_k + \dfrac{z_{\alpha/2} \cdot \sigma_k}{\sqrt{N_k}} \right)$ is within the confidence interval

5:      **return** the minimal required sample size $N_k \geq \underline{N_k} = \dfrac{z_{\alpha/2}^2 \cdot \sigma_k^2}{\text{CI}^2}$

6: **end for**

7: Define the sample set $\mathcal{N}$, where the set size is equal to the minimal required sample size for scenarios $|\mathcal{N}| = \max\limits_{k \in \mathcal{K}} \underline{N_k} \leq ||\mathcal{S}||$ ▷ Find sample size for scenarios

8: **for each** sample index $n \in \mathcal{N}$ **do**

9:      Use Monte Carlo method randomly generate the patient choice matrix A^n

10:     Update the parameters of model (SPDC)

11:     Solve subproblem to get the objective value Q^{n*}

12: **end for**

13: The objective value of all scenarios can be approximated by $\dfrac{1}{|\mathcal{N}|} \sum\limits_{n \in \mathcal{N}} Q^{n*}$

---

separation which is similar to Benders Decomposition (BD). The solution is updated by adding optimality cut by using BD method, while we design two hybrid algorithms, one is based on Gradient Ascent Algorithm with minor updated by Guided Local Search, another is based on Tabu Search Algorithm with minor updated by Large Neighborhood Search. Both hybrid algorithms can reduce the computation time from exponential to polynomial time.

2.3.2 Gradient Ascent and Guided Local Search

The first hybrid algorithm method presented here combines Gradient Ascent (GA) and Guided Local Search (GLS) methods. In this hybrid algorithm, GA provide the framework to iterative update the physician allocation. Within each iteration of GA, we use the GLS to create the gap list, which can give the patient sequence to update the physician allocation one by one. Between two iterations, physician allocation is updated by GA. The computation complexity is $O(L \times (J \times N + \log(L)))$, where L is the size of the physician and J is the size of the clinic. The algorithm of GA-GLS method is shown in Algorithm 2 and Algorithm 3.

Algorithm 2 Gradient Ascent (GA)

1: Initialization: set iteration $i = 0$, physician allocation $\boldsymbol{y}^i = 0$, sequence $\boldsymbol{s}^i = \{0, 1, ..., |L| - 1\}$, build and solve model, **return** maximal objective value $\bar{z}$ $\triangleright$ Start from Trivial Solution

2: time limit $= TL$, iteration limit $= IT$, allocation_update $= 1$ $\triangleright$ Stopping Criteria

3: build computation history list $\boldsymbol{Y}$ to save computation time

4: **if** computation time $< TL$, i $< IT$, allocation_update $== 1$ **then**

5:     Run Algorithm 3 Guided Local Search (GLS)

6: **else**

7:     Output Stop Reason "Overtime" or "Overiteration" or "Cannot find better solution"

8: **end if**

9: The best solution is $\bar{z}$

Within each iteration, the Guided Local Search (GLS) aimed to find the better solution following the sequence of gap list. For example, we consider the case of 7 physicians and 3 clinics. At the iteration 0, we assume all the physician will be allocated to clinic 0 and
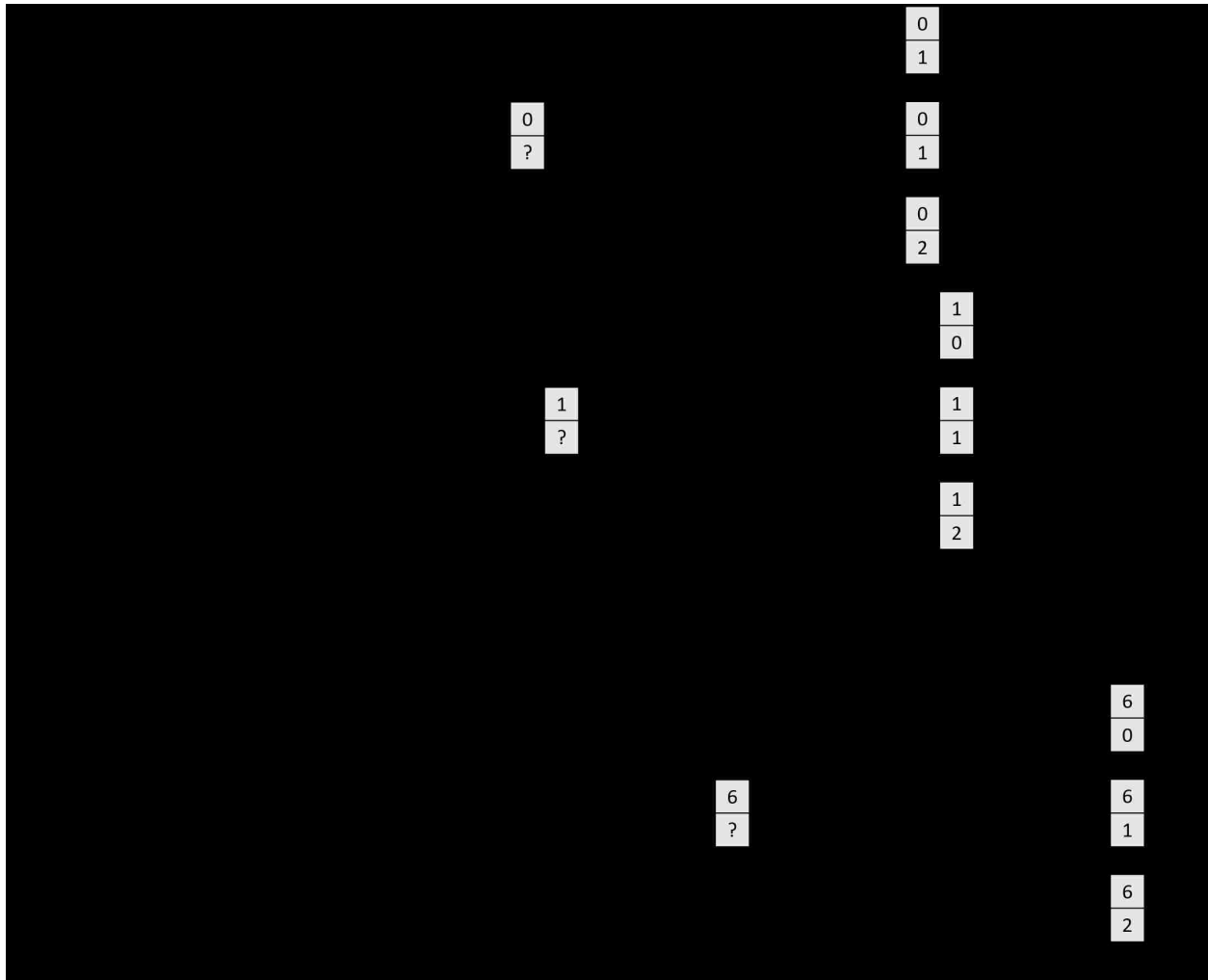
Figure 2.6: GA-GLS: Iteration update

the sequence is from physician 0 to physician 6. Within the iteration 0, we compare the objective function value of different clinic 0,1 and 2 for physician 0 at first, in the meantime, fix the other physician allocation. The flow chart is shown in figure 2.6.

We pick the best clinics of physician 0 and calculate the gap of the objective function value between the best clinic and previous clinic. Then we do the same thing for next physician from the sequence list. After we updated all the physician allocation, we get the GAP list.

33

Algorithm 3 Guided Local Search (GLS)

1: allocation_update = 0, last_solution = $\bar{z}$

2: create SEQUENCE_GAP_LIST

3: **for each** sequence index $n \in L$ **do**

4:     **return** change_physician $\hat{l} = s_n$

5:     clean the allocation of $\hat{l}$: $y_{\hat{l}j} = 0 \quad \forall j \in J$

6:     **for each** clinic $j \in J$ **do**

7:         **return** $y_{\hat{l}j} = 1$

8:         check if this allocation is already in the list, if YES, skip, else do

9:         Update physician allocation $\boldsymbol{y}^i$ and Run Algorithm 1 Sample Average Appoximation (SAA)

10:         **return** objective value z_j, and add allocation decision and objective value in the computation history list

11:     **end for**

12:     $z = \max_j \{z_j\}$

13:     **if** $z \geq \bar{z}$ **then**

14:         $\bar{z} = z$

15:         **return** allocation_update = 1

16:     **end if**

17:     **return** sequence_gap$_n = z - $last_solution

18: **end for**

19: $i = i + 1$

20: sort sequence_gap, and update the sequence of next iteration $\boldsymbol{s}^i$

The patient who has the lower gap will have higher priority to allocate in the next iteration. The flow chart is shown in figure 2.7.
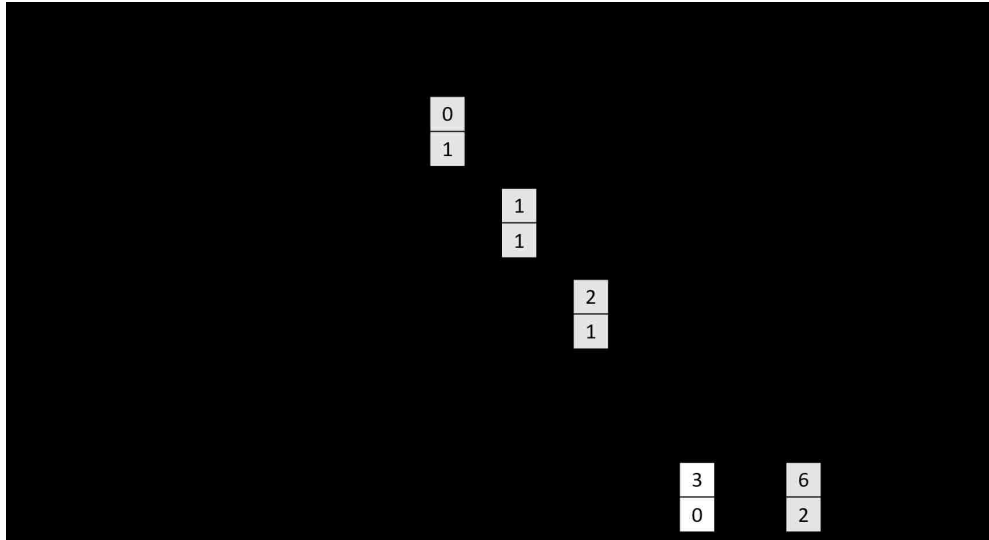
Figure 2.7: GA-GLS: Create gap list and Find sequence

2.3.3 Tabu Search and Large Neighborhood Search

The second hybrid algorithm method presented here combines Tabu Search (TB) and Large Neighborhood Search (LNS) methods. In this algorithm, we use the Tabu Search method to create the tabu list of each iteration, which can jump out of the local optimum by forbidding the same physician allocation. Within the iteration, physician allocation is updated by Tabu Search method. The algorithm of TB-LNS method is shown in Algorithm 4.

We consider the same example with the previous algorithm. Within the iteration, we only change one physician allocation, and pick the best change into the tabu list. In the next iteration, if the new objective function value is lower than the best solution, we keep add the new allocation decision into the tabu list, until we find the one solution is better than

---

Algorithm 4 Tabu Search (TB) - Large Neighborhood Search (LNS)

1: Initialization: set iteration $i = 0$, physician allocation $\boldsymbol{y}^i = 0$, create the tabu list which

    is used to avoid the local minimal solution y, build and solve model $\triangleright$ Start from

    Trivial Solution

2: **return** maximal objective value $\bar{z}$, and add it into the tabu list

3: time limit $= TL$, iteration limit $= IT$, allocation_update $= 1$ $\triangleright$ Stopping Criteria

4: build computation history list $\boldsymbol{Y}$ to save computation time

5: **if** computation time $< TL$, i $< IT$, allocation_update $== 1$ **then**

6:      **if** not in the Tabu list **then** $\triangleright$ at iteration $i = 0$, tabu list is empty

7:         Run Algorithm 1 Sample Average Appoximation (SAA)

8:         add the $\hat{y}$ and $\hat{Q}(x; s)$ in the tabu list

9:      **end if**

10:     do nothing, change to another direction $\triangleright$ in the tabu list

11:     **if** $\hat{Q}(x) \geq LB$ **then**

12:        UPDATE the lower bound $LB = \hat{Q}(x)$

13:     **end if**

14: **else**

15:     output solution $\triangleright$ meet the stopping criteria

16: **end if**

---

all solution of the tabu list. Then the tabu list will be cleaned and add the current best allocation decision into the tabu list. Any physician allocation decision in the tabu will be forbidden in the next iteration. The computation complexity is $O(L \times (J \times N + 1))$.
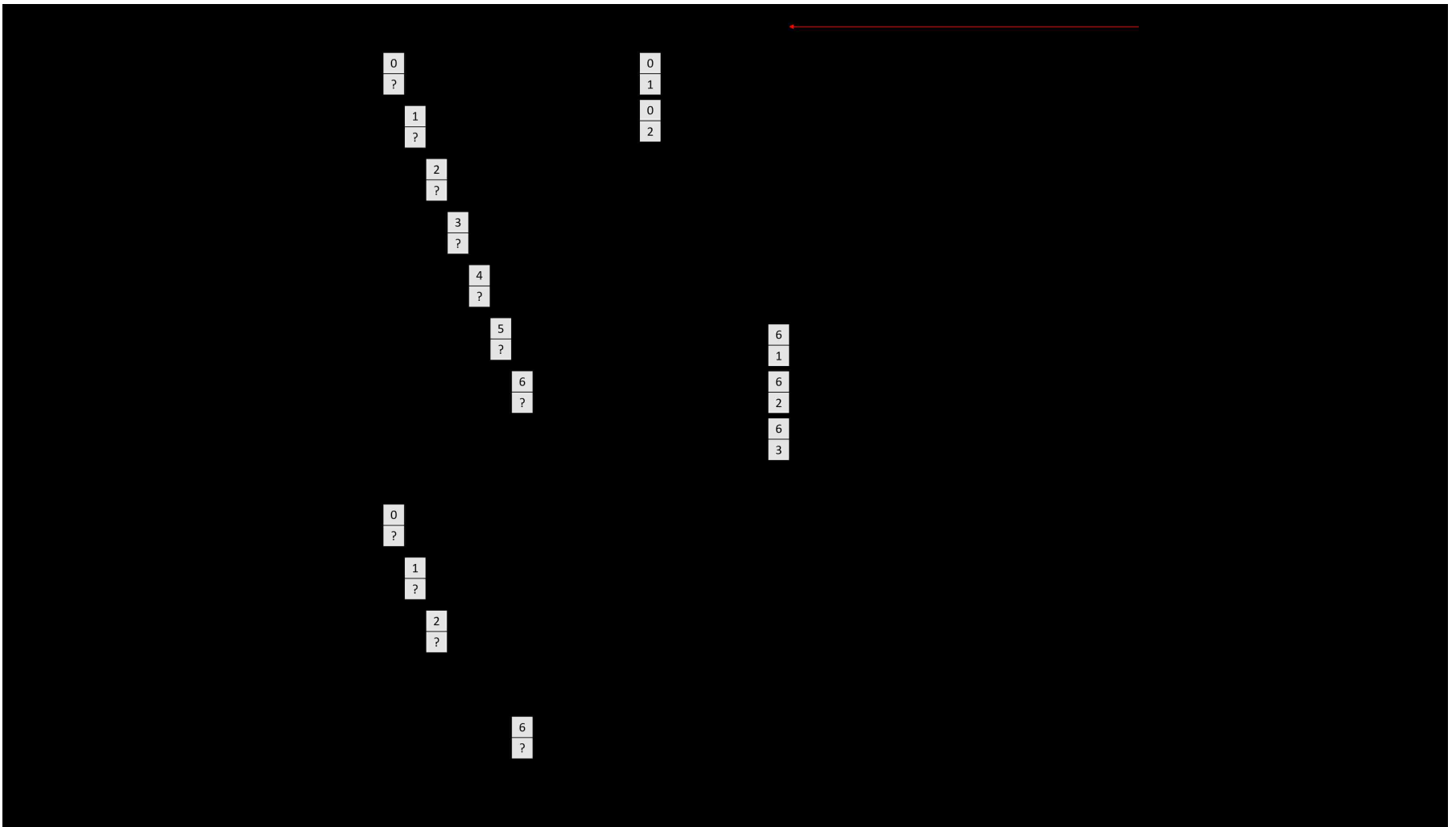
Figure 2.8: TB-LNS: Iteration update

2.4 Computational Results

Numerical experiments and results of different algorithms are presented in this section on solving model (SPDC). We randomly generate 5 data sets, from small size (2 Patient Groups, 4 Physicians, 2 Clinics) to large size (10 Patient Groups, 50 Physicians, 5 Clinics). The largest date set is similar to a health care system of median city. The algorithms are coded in Microsoft Visual Studio 2015 C++ linked with CPLEX 12.8. All the programs are run in Microsoft Windows 10 Professional operating system with Intel Xeon CPU E3-1535M v6 3.10GHz and 16GB RAM.

2.4.1 Algorithm Comparison

In the SPDC model, the GA-GLS method is increasing faster than TB-LNS method at beginning, but the TB-LNS can jump out of the local optimal solution. Figure 2.9 shows the objective value updating with the time increasing. In table 2.3, we compare the computation

(a) date set $(2, 4, 2)$ **(b)** date set $(3, 5, 2)$

Figure 2.9: GD-GLS and TB-LNS computation time and lower bound

38
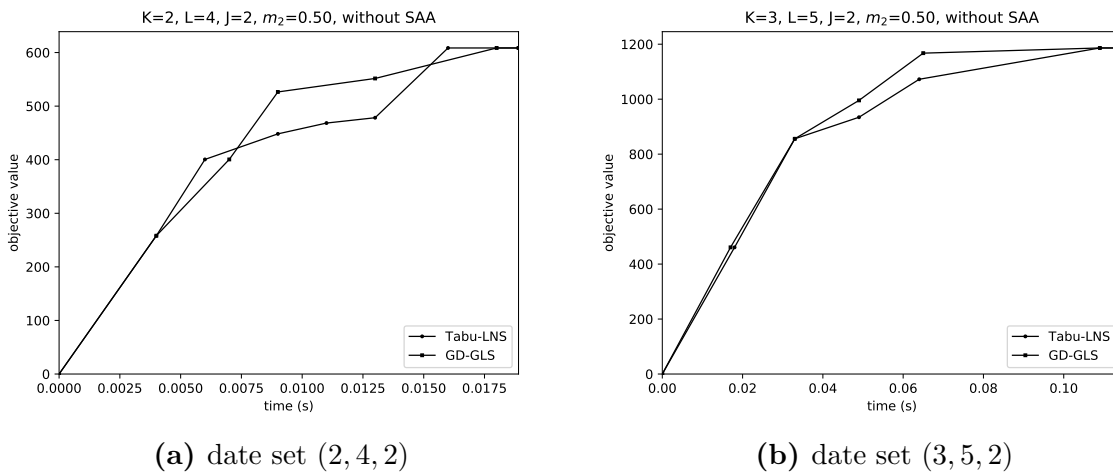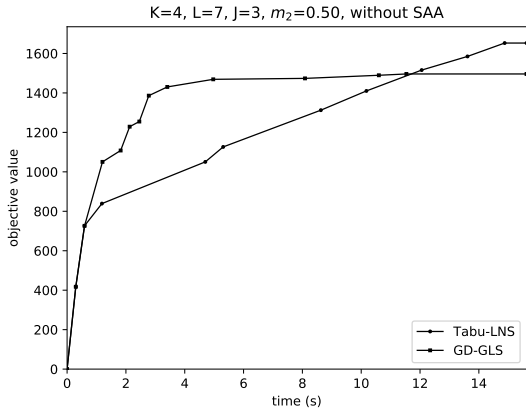
(c) date set $(4, 7, 3)$
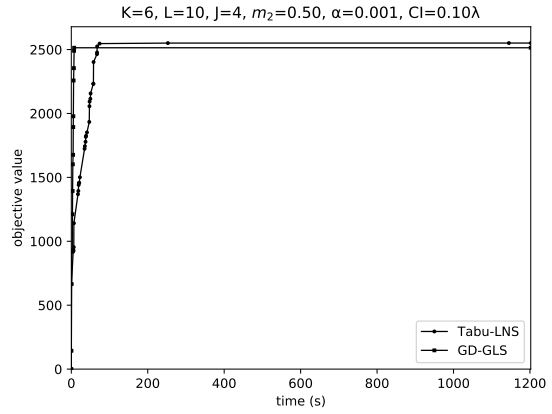
(d) date set $(6, 10, 4)$
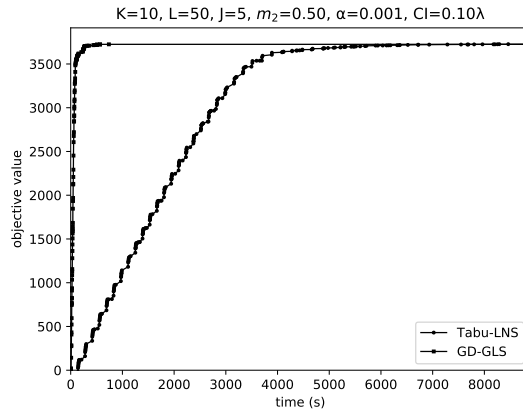
(e) date set $(10, 50, 5)$

Figure 2.9: GD-GLS and TB-LNS computation time and lower bound (cont.)

result accuracy and time in five methods: direct method, GA-GLS, TB-LNS, GA-GLS with SAA and TB-LNS with SAA. The parameter of objective function is defined with discount factor of unpreferred assignment $m_2 = 0.5$ and penalty of no assignment $m_0 = 5$. To decide the Monte Calo sample size in sample average approximation algorithm, we use 3 different confidence level (two-side) of each data set, that are 90.0%, 99.0% and 99.9%. Since the data set size is exponentially increasing, we define the confidence intervals of the small data sets are 0.01μ, the large data sets are 0.10μ.

39

Table 2.3: Computation Time [1] and Solution Comparison

Data Set ($|K|, |L|, |J|$)		Direct Method	GA-GLS	TB-LNS	Confidence Interval	GA-GLS with SAA			TB-LNS with SAA		
						90.0%	99.0%	99.9%	90.0%	99.0%	99.9%
(2, 4, 2)	Time	0.044	0.018	0.016	0.01	0.534	1.301	5.334	1.255	3.213	5.001
	Optimal	608.54	608.54	608.54		608.738	608.64	608.61	608.73	608.64	608.61
	Gap	N/A	0.00%	0.00%		0.032%	0.016%	0.011%	0.032%	0.016%	0.011%
(3, 5, 2)	Time	0.459	0.109	0.109	0.01	0.110	0.159	0.226	0.090	0.146	0.200
	Optimal	1186.29	1186.29	1186.29		1186.8	1186.7	1186.4	1186.85	1186.70	1186.46
	Gap	N/A	0.00%	0.00%		0.046%	0.035%	0.014%	0.046%	0.035%	0.013%
(4, 7, 3)	Time	653.424	11.536	14.877	0.01	4.846	10.298	20.895	11.320	28.604	48.613
	Optimal	1652.87	1496.18	1652.87		1496.30	1496.43	1496.48	1653.11	1652.88	1652.86
	Gap	N/A	9.48%	0.00%		9.473%	9.465%	9.462%	0.015%	0.001%	0.001%
(6, 10, 4)	Time	– [2]	3050.75	233.733	0.10	3.485	5.609	7.501	516.30	1580.03	1144.83
	Optimal	–	2511.84	664.877		2513.13	2513.98	2514.11	2555.45	2557.97	2551.01
	Gap	N/A	N/A	N/A		N/A	N/A	N/A	N/A	N/A	N/A
(10, 50, 5)	Time	–	–	–	0.10	720.977	720.310	738.437	8444.37	8465.27	8461.92
	Optimal	–	–	–		3724.59	3724.59	3724.59	3726.43	3726.43	3726.43
	Gap	N/A	N/A	N/A		N/A	N/A	N/A	N/A	N/A	N/A

---

[1] The computation time of the largest dataset $(10, 50, 5)$ is limited in 18000 sec. The other dataset is solved in 3600 sec.

[2] –: out of memory or timelimit.

In this section, we discuss the correlation between the assignment type and the assignment penalty. To analysis the sensitivity of the total utilization of all patients, we change the weight of the unpreferred assignment and penalty of the unsatisfied patient.

We run the Algorithm TB-LNS with date set ($K = 4$, $L = 7$, $J = 3$), the sensitivity analysis result is shown in Figure 2.10. The weight of no assignment m_0 has barely impact on the solution, while the objective value has significantly positive correlation with the weight of unpreferred assignment m_2. We also compare two different stochastic programming models, model (SPEL) and model (SPDC). To see which model is able to get higher patient satisfaction, suppose we just replaced the physician allocation optimal decisions with discrete
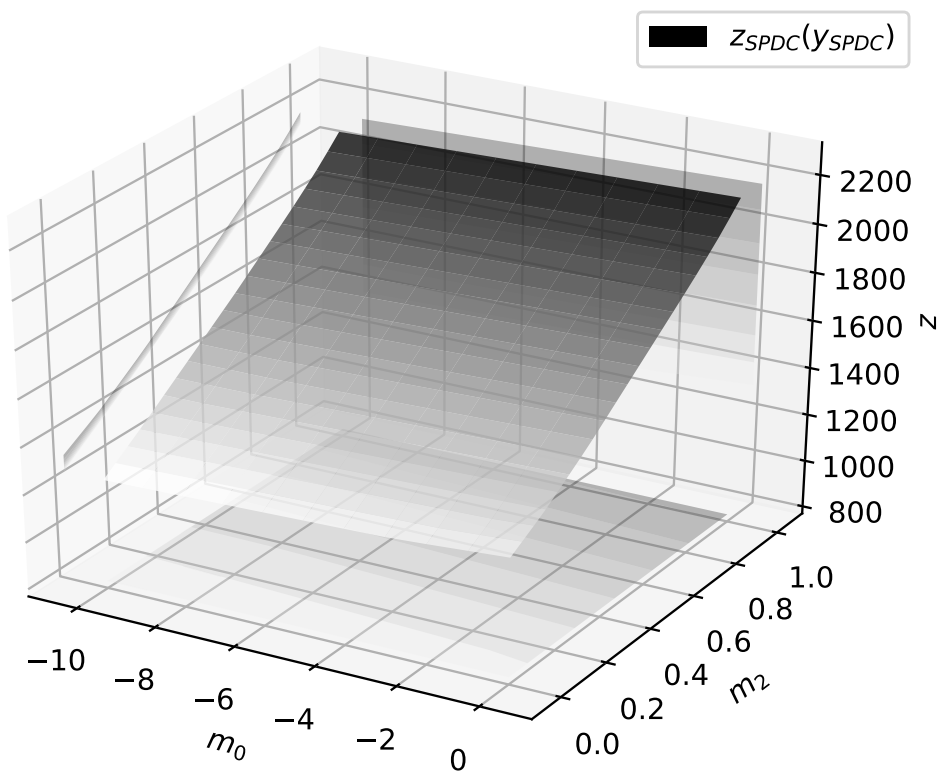
Figure 2.10: Sensitivity Analysis of Data Set: $K = 4$, $L = 7$, $J = 3$

choice model by Equally Likely Scenarios decisions and solved that problem, then we define the Value of Discrete Choice Model (VDC) as below:

$$VDC = z_{SPDC}(x^*) - z_{SPDC}(x^o)$$

$$x^* = \arg\max_x z_{SPDC}(x)$$

$$x^o = \arg\max_x z_{SPEL}(x)$$

When the there is no penalty on unsatisfied assignment m_0 and the no difference on the weight of preferred and unpreferred assignment m_2, the value of discrete choice model $VDC = 0$. Figure 2.11a shows the VDC has negative correlation with the weight of unpreferred assignment, except the allocation decision changed which shown in figure 2.11b.
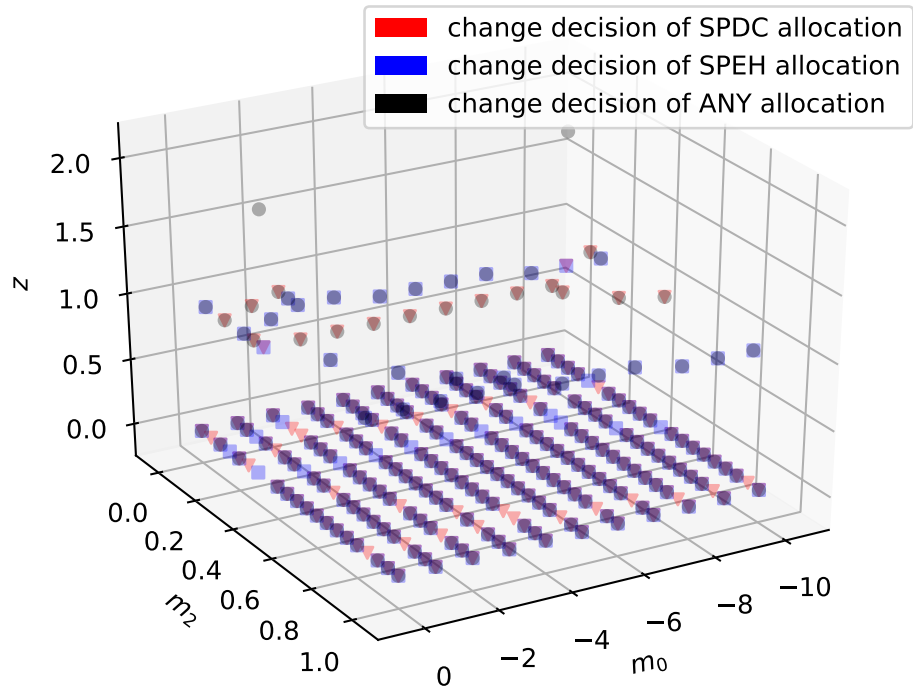
(a) Value of Discrete Choice Model
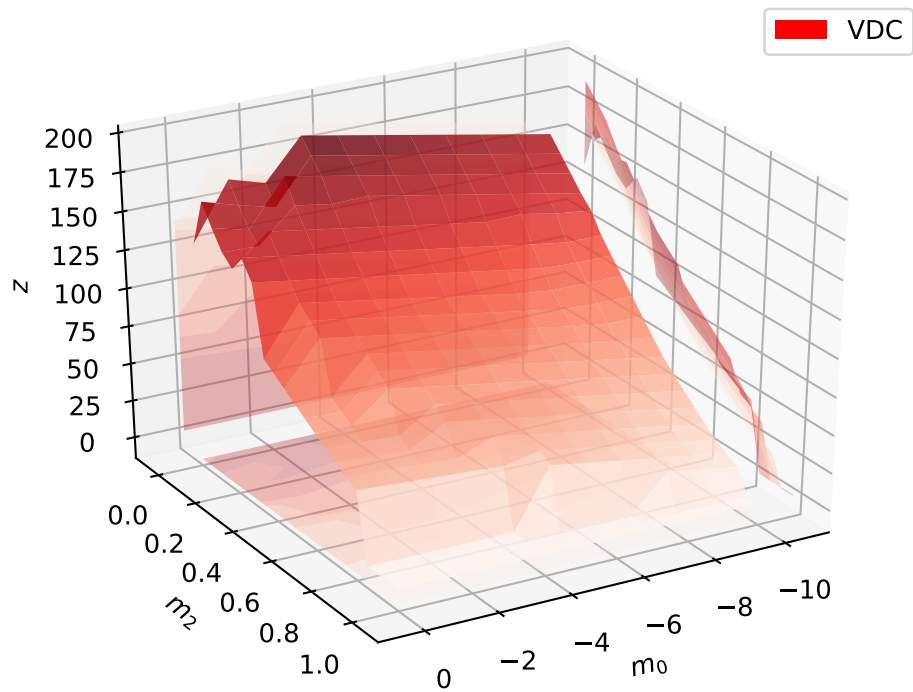
(b) Change of Allocation Decision

Figure 2.11: Value of Discrete Choice Model of Data Set: $K = 4$, $L = 7$, $J = 3$

CHAPTER 3: REINFORCEMENT LEARNING IN INFORMATION CASCADE BASED ON DYNAMIC USER BEHAVIOR

In this project, we study the Influence Maximization problem based on Information Cascade within a Random Graph, where the network structure is dynamically changed by the user's uncertain behavior. We use the Discrete Choice Model to build the probability distribution of the directed arc between any two nodes in a random graph. In our problem, the Discrete Choice Model provides a good description and prediction of user behavior in terms of following or not following the neighbor node. To find the maximal influence at the end of a finite time horizon, we model this problem by Multi-Stage Stochastic Programming, which can help the decision maker to select the optimal seed node to broadcast messages efficiently. Since the computation complexity grows exponentially with the network size and time horizon, the original model is not solvable within a reasonable time. We have two approaches for approximating the optimal decision: One is the Myopic Two-Stage Stochastic Programming at each time period, while the other one is Reinforcement Learning using the MDP. Computational experiments shows that the Reinforcement Learning method has better performance than Myopic method does in a large-scale network.

3.1 Introduction

Cascading phenomena are typically characterized by a dynamic process of information propagation between nodes in a network, where nodes can rebroadcast or repost information from and to their neighbors. Moreover, the content and value of information may affect

44

not only the reach (or depth) of a cascade but also the topology of the underlying network. This is due to the effects whereby nodes may either sever their ties with neighboring nodes where the transmitted information is deemed unreliable and/or malicious or form new ties with nodes transmitting "reliable" information. In an information cascade, people observe the choices of others and make decisions based on these observations while considering their personal preference. This phenomenon usually arises in the field of behavioral economics and other social sciences. For example, in Viral Marketing, information cascade is the process of spreading information about a product with other people in their social networks, where the objective is to promote a product using existing social networks. A recent study of social networks suggests that such processes may occur in a "bursty" fashion, that is, the patterns of network links change abruptly as a result of significant information cascades. Thus, new information may create a burst of node activations and edge activations/deactivations in a network. In a decentralized autonomous network, agents or nodes act independently and behave according to their utility functions. To model their autonomous behaviors, we implement the concepts of discrete choice models from behavioral economics.

3.1.1 Literature Review

In general, the nature of information cascades can be described as follows. When a node of a network adopts certain information, it is "activated" [32]. The definition presented in [27] states, an *activation sequence* is an ordered set of nodes capturing the order in which the nodes of the network adopted a piece of information. The first node in the activation sequence is *seed node*. *Spreading cascade* is a directed tree having as a root the first node of the activation sequence. The tree captures the influence between nodes (branches represent who transmitted the information to whom) and unfolds in the same order as the activation sequence. There are two typical information diffusion models, namely,

Independent Cascade [23] and Linear Threshold [25]. Our study is based on the assumption of Independent Cascade model. Saito, Nakano, and Kimura [53] propose the Expectation Maximization algorithm to predict the information diffusion probabilities in the Independent Cascade model. Chen, Wang, and Wang [11] apply the Influence Maximization problem with an Independent Cascade model in the prevalent viral marketing. Furthermore, Wang, Chen, and Wang [60] has shown for the first time that the computing influence spread in the Independent Cascade model is NP-Hard; these researches have designed a new heuristic algorithm that can easily scale up compared with the greedy algorithm proposed by Kempe, Kleinberg, and Tardos [32].

Distinct from the previous research on the Independent Cascade model, we consider the information diffusion probabilities or the network topology probabilities as dynamically changing with the user behavior. Oinas-Kukkonen [49] has introduced the concept of behavior change support systems. Based on this work, Ploderer et al. [51] find ample evidence of the strong influence exerted by social interaction on people's behaviors. Yu et al. [63] conduct extensive statistical analysis on large-scale real data and find that the general form of Exponential, Rayleigh and Weibull distribution can well preserve the characteristics of behavioral dynamics. From Yu et al. [63]'s paper, the Networked Weibull Regression model for behavioral dynamics modeling significantly improves the interpretability and generality of traditional survival models.

3.1.2 Our Contributions

To maximize the influence of the information provider within a limited time, we model problem as an seed selection problem of information spreading in dynamic networks with random graphs. In the social network, each user may has three roles, which are those of source

user, message sender (followee of neighbors), and message receiver (follower of neighbors). We decompose our problem into two process;

- Seed Selection: This can be controlled by the information provider, which selects a proper set of initial seeds that will initialize the information diffusion process;

- Information Cascade: This includes two variables. One is the node activation status, which describe the process that the user receives message from their followee. The other one is the node repost decision, which is controlled by the message receiver. In our model, the repost decision depends on the user preference and the received message type.

In this project, we propose an information maximization model through independent cascade with random graphs. For the network properties, the network size and node preference is given and fixed, while the friendships between any two users (arc connection) are dynamically changed. Our model can help the decision maker choose the optimal action when facing an uncertain network topology. The stochastic formulation considers endogenous uncertainty, which is represented by the binary choice probability distribution of arc connection between any two nodes. To solve this problem, we design two problem-specific algorithms, one is two-stage stochastic programming with a myopic policy, while the other is reinforcement learning with the MDP.

We summarize the contributions of this project as follows:

- We introduce the discrete choice model in the information maximization problem, where the distribution of network topology is dynamically changing during the Independent Cascade;

- We develop the practical algorithms for solving the multi-stage stochastic programming problem with endogenous uncertainty;

- To avoid directly dealing with large state spaces of node activation, we take advantage of the implicit Monte Carlo-Based Partially Observable Markov Decision Process (MC-POMDP); and

- We compare the results using two algorithms and different sample size.

The rest of this chapter is structured as follows: We briefly describe the information maximization and information cascade problem in random graphs with a finite time horizon, and we provide the original multi-stage stochastic programming models with several assumptions in Section 3.2. In Section 3.3, we design two algorithms to solve this problem. The computational results are shown in Section 3.4.

3.2 Mathematical Models

In a social network, the information is transmitted between users. Initially, some nodes will be selected as seed nodes, which are the source users for broadcasting messages in the network. During the information cascade, each node may have two roles, that of message receiver, who is activated with a certain message by a neighbor, and message sender, who reposts the received message to the neighbor. Information providers have several messages on hand, and they want to maximize their influence in a network. While the users of the network may have different preferences on the different messages, which node is good to be a seed node is a problem that the information provider faces.

For each period, the information provider will select the seed node to post a certain message in the social network. Sometimes, it is the initial posting of a certain message, while sometimes it is a repeated post to increase the network activity. Once the source user posts the message, the followers of source user automatically receive the information. The follower

make decisions message based on their preferences. As the user has multiple roles in the social network, the follower also acts as the followee of other users. The information flows are always from followee to follower. The track of the information transmission has an influence on the network topology, which means the user relationship or the arc connection is dynamically changed. Since the influence maximization problem has uncertainty on the network topology, we model this problem by stochastic programming, and the objective is maximizing the total influence on the finite time horizon. The influence is measured by the seed cost and node activation.

3.2.1 Problem Description

To show the information cascade process of our problem clearly, we give a simple example here. Considering viral marketing in a random network $G(n, p)$, a company wants to promote two products in a network with uncertain topology. To maximize the company influence, the company wants to select certain nodes as influencers to post the promotion message in the network. Figures 3.1 and 3.2 give us an information cascade example in a 4-node network. Before seed selection, we know the node preference in terms of the message type. During the
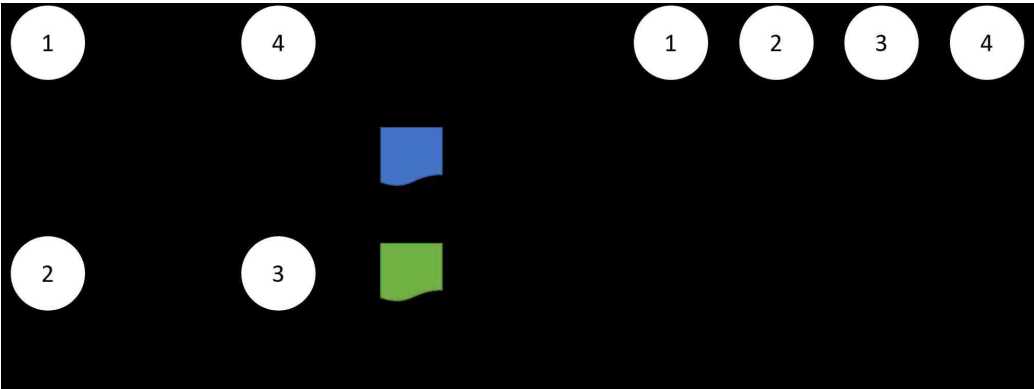
Figure 3.1: Given Network Properties

49

information cascade, the network topology is dynamically changed and decision-dependent. Assume there are two types of message, blue and green, and the initial arc probability of the random graph $p = 0.5$. Some node may already know the messages before information cascade. All the given network properties are shown in Figure 3.1.

Within one period, the information cascade usually includes four steps, as follows: seed selection, message transmission (node send messages), node activation (node receive messages), and updating network topology probability. When the message provider selects the seed, the message is broadcast by the seed node in the network, but it cannot guarantee all the other nodes of the network will received the message. Only followers are able to receive the message
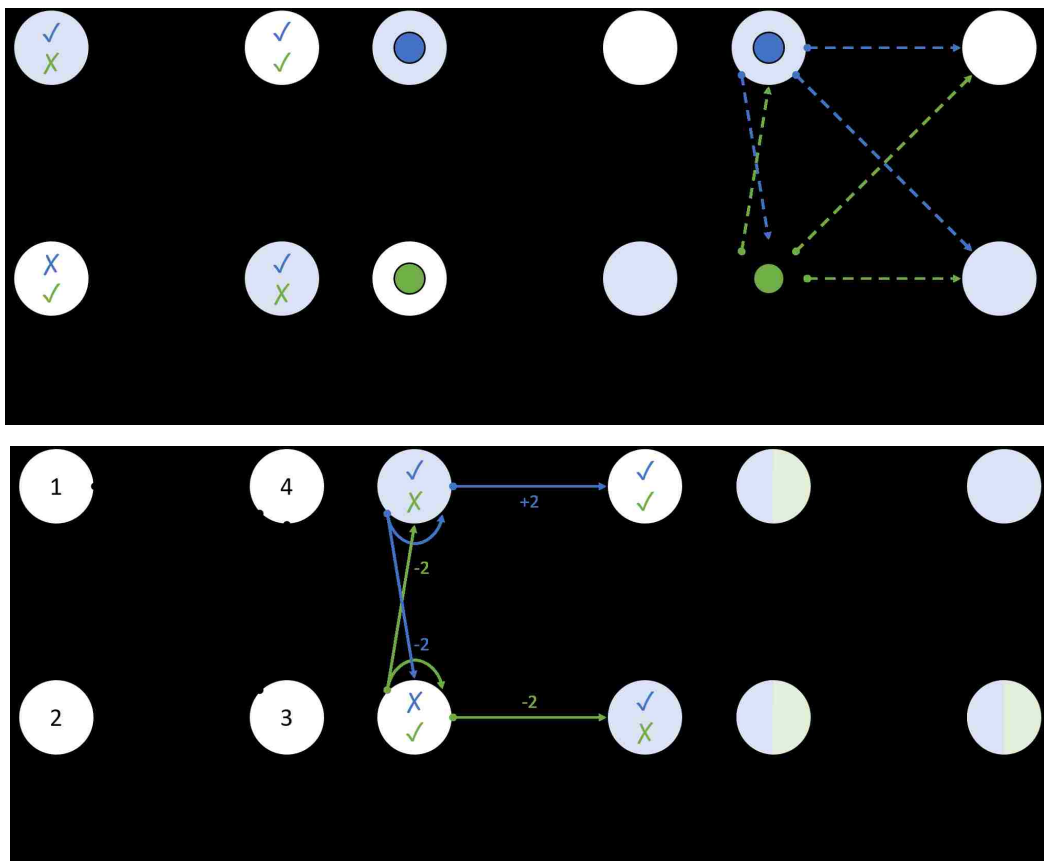
Figure 3.2: Information Cascade in Time Horizon $|\mathcal{T}| = 2$
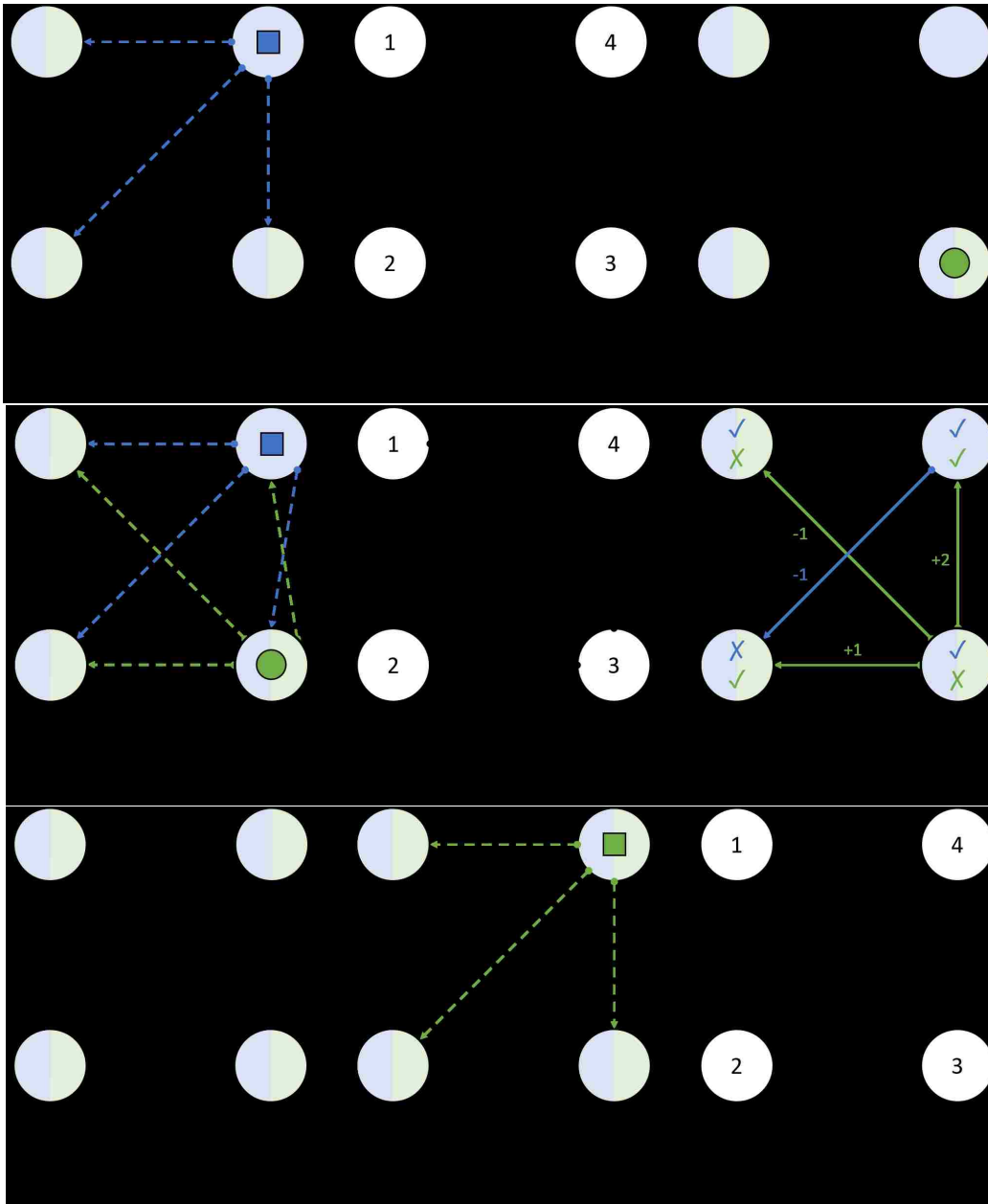
Figure 3.2: Information Cascade in Time Horizon $|\mathcal{T}| = 2$ (cont.)

from the message sender. After the information transmission, the network topology may be changed. For a message receiver, it has a high chance of disconnecting the link from the followee if the received message and follower's preference is mismatched. This means some directed arcs will break down even if it may be connected in the last time period, which is

due to the uncertain topology. This uncertain topology is modeled by discrete choice model with two alternatives. Figure 3.2 shows the information cascade of two messages in a 4-node network within the time horizon $\mathcal{T} = 2$.

At time $t = 0$, node $i = 1$ is selected as the seed node of message BLUE and node $i = 2$ is selected as the seed node of message GREEN. Then, these two nodes will broadcast messages in the network. The initial probability of the directed arc connection between any two nodes is 0.5. When message transmission occurs, the real topology is as shown in the fourth picture of Figure 3.2. The arc from node 1 to node 3 is disconnected, and the arc from node 2 to node 4 is disconnected; this means node 3 cannot receive message BLUE and node 4 cannot receive message GREEN. Since nodes 1 and 2 are seed nodes, they are activated alone. Node 2 is activated from message BLUE by node 1. Since node 2 dislikes message BLUE, it will break the friendship from node 1 to node 2. We use the utility of measuring the friendship. When the node initially receives the message, we assume it has double effect on the utility changing. We reduce the 2 utility from node 1 to node 2, because it is the first time node 1 to receives this message. Node 4 is also activated with message BLUE by node 1. Since node 4 likes this message and never receive this message in all the previous time periods, node 4 will decide to repost this message in the network. The utility from node 1 to node 4 will be increased by 2.

The topology probability of the directed arc connection at the next time period is updated by the utility changing. For example, the probability of a directed arc from note 1 to node 2 is updated as

$$\mathrm{Prob}(a_{12}^{t=1} = 1) = \mathrm{Prob}(a_{12}^{t=1} = 1 | a_{12}^{t=0} = 1)$$
$$= \frac{1}{1 + exp(-u_{12}^{t=0})} = 0.1192$$

where a_{12}^t is the directed arc connection status at time t and u_{12}^t is the utility at time t if $a_{12}^t = 1$. The details of probability updating are explained in Subsection 3.2.2.

3.2.2 Mathematical Formulation

We formulation the Independent Cascade within Random Graph (ICRG) problem by using stochastic programming model. In our model, the independent cascade include 3 decision variables, seed selection x, node activation y, message transmission z. The notation is shown in Table 3.1.

The original stochastic programming model [SP] is shown below:

$$[SP] \quad \max_{x,y,z} \ \mathbb{E}\left(Q(x), R(y); \varepsilon\right) = \sum_{s \in \mathcal{S}} P^s(a) \cdot (R^s(y) - Q^s(x)) \tag{3.1a}$$

$$s.t. \ P^s(a_{ij}) = \prod_{t \in \mathcal{T}} \prod_{i \in \mathcal{I}} \prod_{j \in \mathcal{I} \setminus \{i\}} \text{Prob}(a_{ij}^{t,s} = 1) \qquad \forall s \in \mathcal{S} \tag{3.1b}$$

$$R^s(y) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} w_k \cdot (2b_{ki} - 1) \cdot (y_{ki}^{t=|\mathcal{T}|,s} - c_{ki}) \quad \forall s \in \mathcal{S} \tag{3.1c}$$

$$Q^s(x) = \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} x_{ki}^{t,s} \qquad \forall s \in \mathcal{S} \tag{3.1d}$$

$$x^{t,s} = x^{t,s+1} \qquad t \in \mathcal{T}, s \in \mathcal{S} \setminus \bar{\mathcal{S}}^t \tag{3.1e}$$

$$y_{ki}^{t,s} = \max\{c_{ki}, x_{ki}^{t,s}\} \qquad \forall t = 0, k \in \mathcal{K}, i \in \mathcal{I} \tag{3.1f}$$

$$z_{ki}^{t,s} = x_{ki}^{t,s} \qquad \forall t = 0, k \in \mathcal{K}, i \in \mathcal{I} \tag{3.1g}$$

$$y_{ki}^{t,s} = \max\left\{x_{ki}^{t,s}, y_{ki}^{t-1,s}, \max_{j \in \mathcal{I} \setminus \{i\}}\{a_{ji}^{t,s} \cdot z_{kj}^{t-1,s}\}\right\} \qquad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I} \tag{3.1h}$$

$$z_{ki}^{t,s} = \max\left\{x_{ki}^{t,s}, b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1.s})\right\} \qquad \forall t \in \mathcal{T}, s \in \mathcal{S}, k \in \mathcal{K}, i \in \mathcal{I} \tag{3.1i}$$

Table 3.1: Notation of Multi-Stage Stochastic programming Model

Symbol	Definition
	Indices and Sets
$i \in \mathcal{I}$	node
$k \in \mathcal{K}$	message
$t \in \mathcal{T}$	time
$s \in \mathcal{S}$	scenario
	Parameters
$a_{ij}^{t,s}$	the directed arc from node i to node j
b_{ki}	the information preference of node i with respect to message k
c_{ki}	the pre-activation, that node i has known or has not known the message k before the seed selection
w_k	the influence weight of message k
	Decision Variable
x_{ki}^{t}	binary variable, seed selection, whether the node i is selected as the seed node of message k at time t
$y_{ki}^{t,s}$	binary variable, node activation, whether the node i is activated by message k at time t and scenario s
$z_{ki}^{t,s}$	binary variable, message transmission, whether the node i decide to transmit message k to its neighbor at time t and scenario s

$$x \in \mathbb{B}, \quad y \in \mathbb{B}, \quad z \in \mathbb{B}$$

In objective function (3.1a), the total influence has two parts: one is the seed cost $Q(x)$, the other one is activation reward $R(y)$. Constraint (3.1b) shows the probability of scenario s depend on the probability of arcs between any two nodes. The directed arc a_{ij} from node i to node j is random variable, which is following logit binary choice model with utility U_{ij}.

Utility U_{ij} is a function to measure the user friendship or the strength of arc connection, which includes two term: observed utility u_{ij} and unobserved utility ε_{ij}. The observed utility $u_{ij}^{t,s}$ at time t and scenario s is cumulative impact from node i to node j with all kinds of message type. The current direct arc $a_{ij}^{t,s}$ from node i to node j decide the impact happen or not, the impact sign is decided by the preference b_{kj} of message k and node j, and the impact amount is decided by the transmission decision $z_{ki}^{t-1,s}$ of message k and node i at last moment. The unobserved utility $\varepsilon_{ij}^{t,s}$ is assumed to have a logistic distribution.

$$U_{ij}^{t,s} = u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{U}_{ij}^{t,s} = u_{ij}^{t-1,s} + \bar{\varepsilon}_{ij}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$a_{ij}^{t+1,s} = \begin{cases} 1, & U_{ij}^{t,s} > \bar{U}_{ij}^{t,s} \\ 0, & U_{ij}^{t,s} \leq \bar{U}_{ij}^{t,s} \end{cases} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\varepsilon_{ij}^{t,s} \sim \text{Logistic} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

Before the information cascade, there is no message transmission and each node doesn't know anything from the other nodes. Whether connect or disconnect, the observed utility is always be 0.

$$u_{ij}^{t,s} = \bar{u}_{ij}^{t,s} = 0 \qquad \forall\, t = -1,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$U_{ij}^{t,s} = 0 + \varepsilon_{ij}^{t,s} \qquad \forall\, t = -1,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{U}_{ij}^{t,s} = 0 + \bar{\varepsilon}_{ij}^{t,s} \qquad \forall\, t = -1,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\text{Prob}(a_{ij}^{t+1,s} = 1) = \text{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = 0.5 \qquad \forall\, t = -1,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

At the initial time period $t = 0$, seed node broadcast the message in the network, and some

node may received message from the seed node.

$$u_{ij}^{t,s} = \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot x_{ki}^{t,s} \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{u}_{ij}^{t,s} = 0 \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$U_{ij}^{t,s} = u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{U}_{ij}^{t,s} = 0 + \bar{\varepsilon}_{ij}^{t,s} \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\mathrm{Prob}(a_{ij}^{t+1,s} = 1) = \mathrm{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = \frac{1}{1 + exp(-u_{ij}^{t,s})} \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

From time $t = 1$ to the end of time horizon $t = \mathcal{T}$, except the seed node, the other node who received message also involve in the message transmission.

$$u_{ij}^{t,s} = \sum_{\tau=0}^{t} \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{u}_{ij}^{t,s} = \sum_{\tau=0}^{t-1} \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\Delta u_{ij}^{t,s} = u_{ij}^{t,s} - \bar{u}_{ij}^{t,s} = \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot a_{ij}^{t,s} \cdot z_{ki}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$U_{ij}^{t,s} = u_{ij}^{t,s} + \varepsilon_{ij}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\bar{U}_{ij}^{t,s} = \bar{u}_{ij}^{t,s} + \bar{\varepsilon}_{ij}^{t,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

$$\mathrm{Prob}(a_{ij}^{t+1,s} = 1) = \mathrm{Prob}(U_{ij}^{t,s} > \bar{U}_{ij}^{t,s}) = \frac{1}{1 + exp(-\Delta u_{ij}^{t,s})} \qquad \forall\, t = 0,\, s \in \mathcal{S},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$

The total seed cost equals to the number of seed node. The reward equals to the weighted average of final active node amount. Constraint (3.1c) shows the activation reward depends on message weight, node preference and node activation status y at end of the time horizon $t = |\mathcal{T}|$. Constraint (3.1e) is nonanticipativity constraint, the scenario subset $\bar{\mathcal{S}}^t$ define as

below:

$$\bar{S}^t = \{s \in \mathcal{S} \mid s = |\mathcal{S}| \cdot \frac{\tau}{|\mathcal{A}|^t} \quad \forall \tau = 1, \cdots, |\mathcal{A}|^t \quad \forall t \in \mathcal{T} \cup \{0\}\}$$

where the directed arc size is $\mathcal{I} \cdot (\mathcal{I} - 1)$, the combination of all arcs status is $|\mathcal{A}| = 2^{\mathcal{I} \cdot (\mathcal{I}-1)}$, and the scenario set cardinality $|\mathcal{S}| = |\mathcal{A}|^{|\mathcal{T}|} = 2^{|\mathcal{I}| \cdot (|\mathcal{I}|-1) \cdot |\mathcal{T}|}$.

The information cascade process is limited by 4 constraints. Constraints (3.1f, 3.1g) define the initial node activation and transmission decision at time $t = 0$. Constraints (3.1h, 3.1i) define the information diffusion rule from time $t = 1$ to the end $t = |\mathcal{T}|$.

In constraint (3.1f), some node are active node at beginning because it has already known this message c_{ki} or it is selected as seed x_{ki}. So the initial time period $t = 0$, node is not active node if and only if it didn't know the message before k and it is not selected as seed node. Due to the binary property, constraint (3.1f) can be linearized by the equation below:

$$1 - y_{ki}^t = (1 - c_{ki}) \cdot (1 - x_{ki}^t) \qquad \forall t = 0, \, k \in \mathcal{K}, \, i \in \mathcal{I} \qquad \text{(3.1f-L)}$$

The initial message transmission happen if and only if the node is selected as seed node, shown in constraint (3.1g).

Except the seed selection, the node may also be activated by two causes from time $t = 1$ to the end $t = |\mathcal{T}|$, shown in constraint (3.1h). One is once node i was activated by message k at previous time period $t - 1$, it will be active node in the future. The other one is at least one of the followees transmit the message k at the previous time period $t - 1$. Constraint (3.1h) can be linearized by the following inequalities:

$$y_{ki}^{t,s} \geq x_{ki}^t \qquad \forall t \in \mathcal{T}, \, s \in \mathcal{S}, \, k \in \mathcal{K}, \, i \in \mathcal{I} \qquad \text{(3.1h-L1)}$$

$$y_{ki}^{t,s} \geq y_{ki}^{t-1,s} \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1h-L2)}$$

$$y_{ki}^{t,s} \geq a_{ji}^{t,s} \cdot z_{kj}^{t-1,s} \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}$$
$$\text{(3.1h-L3)}$$

$$y_{ki}^{t,s} \leq x_{ki}^{t} + y_{ki}^{t-1,s} + \sum_{j \in \mathcal{I}\setminus\{i\}} a_{ji}^{t,s} \cdot z_{kj}^{t-1,s} \qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1h-L4)}$$

Constraint (3.1h-L3) is based on independent cascade assumption, that means the node will be activated ($y_{ki}^{t,s} = 1$) if the neighbor node ($a_{ji}^{t,s} = 1$) decide to transmit message ($z_{kj}^{t-1,s} = 1$). For node i, we define the number of all the neighbors as degree $DEG_i = \sum_{j \in \mathcal{I}\setminus\{i\}} a_{ji}$. Since one of the neighbor transmit message, the receiver node will be activated, constraint (3.1h-L3) for all neighbour node j can be aggregated by the receiver node i.

$$y_{ki}^{t,s} \geq \frac{\sum_{j \in \mathcal{I}\setminus\{i\}} a_{ji}^{t,s} \cdot z_{kj}^{t-1,s}}{\sum_{j \in \mathcal{I}\setminus\{i\}} a_{ji}^{t,s}} \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1h-L3-A)}$$

Constraint (3.1h-L4) shows the node is deactivated if all the possible activation causes are failed.

Constraint (3.1i) shows node i has two motivation to transmit message k. One is node i is selected as seed, the other one is node i is new active node of message k and like this message. Constraint (3.1i) can be linearized by the following inequalities:

$$z_{ki}^{t,s} \geq x_{ki}^{t} \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1i-L1)}$$

$$z_{ki}^{t,s} \geq b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1,s}) \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1i-L2)}$$

$$z_{ki}^{t,s} \leq x_{ki}^{t} + b_{ki} \cdot (y_{ki}^{t,s} - y_{ki}^{t-1,s}) \qquad\qquad \forall\, t \in \mathcal{T},\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \qquad \text{(3.1i-L3)}$$

Constraint (3.1i-L2) is based on independent cascade assumption, that means the node is

willing to transmit message ($z_{ki}^{t,s} = 1$) if it like this message ($b_{ki} = 1$) and it just activated ($y_{kj}^{t,s} = 1$) and never know this message before ($y_{kj}^{t-1,s} = 0$). Constraint (3.1i-L3) shows the node decide not to transmit message if all the transmission motivation are invalid.

The computation complexity of this model is $O(2^{|\mathcal{K}||\mathcal{I}| \cdot log_{|T|}|S| \cdot |T|})$. To reduce the complexity, we add an assumption of seed selection, that the decision maker only allow to select one seed node of each message within one time period. It is formulated by the following constraint:

$$\sum_{i \in \mathcal{I}} x_{ki}^{t,s} = 1 \ \ \forall\, s \in \mathcal{S},\, t \in \mathcal{T} \cup \{0\},\, k \in \mathcal{K} \tag{3.1d-A}$$

After adding this assumption, the computation complexity is reduced to $O(|\mathcal{I}|^{|\mathcal{K}| \cdot log_{|T|}|S| \cdot |T|})$ and the objective function (3.1a) can be simplified as below:

$$\max_{x,y,z} \ \mathbb{E}\left(Q(x), R(y); \varepsilon\right) = \sum_{s \in \mathcal{S}} P^s(a) \cdot (R^s(y) - Q^s(x)) = -|\mathcal{T} + 1| \cdot |\mathcal{K}| + \sum_{s \in \mathcal{S}} P^s(a) \cdot R^s(y))$$

$$\tag{3.1a-A}$$

3.3 Solution Approaches

Since the network topology is dynamic changed, the decision maker is faced to an unstable node friendship. The uncertain directed arc connection lead to the scenario size exponentially growth with the network size $|I|$ and time horizon $|T|$. To handle the large-scale scenarios, we have two approaches to solve the information cascade in random graph problem:

- Myopic Policy: does not explicitly use any forecasted network topology and separate the multi-stage into several two-stage problems (MYSP) by discrete time.
- Reinforcement Learning: reformulate the Stochastic programming model to Markov Decision Process (MDP)

3.3.1 Two-Stage Stochastic programming with Myopic Policy

Different to the original model, myopic model focus on current network topology and ignore the future changing on arc. The seed selection(x^t) is only based on current user connection (a^t) and aims to find the local maximal influence on node activation of next time period (y^{t+1}).

$$x^t = \arg\max R(y^{t+1}, a^t)$$

Table 3.2: Notation of Myopic Two-Stage Stochastic programming Model

Symbol	Definition
	Indices and Sets
$i \in \mathcal{I}$	node
$k \in \mathcal{K}$	message
$s \in \mathcal{S}$	scenario
	Parameters
a_{ij}^s	the directed arc from node i to node j
b_{ki}	the information preference of node i with respect to message k
c_{ki}	the pre-activation, that node i has known or has not known the message k before the seed selection
d_{ki}	the node repost decision, that node i will repost message k in the network
w_k	the influence weight of message k
	Decision Variable
x_{ki}	binary variable, seed selection, whether the node i is selected as the seed node of message k at time t
y_{ki}^s	binary variable, node activation, whether the node i is activated by message k at time t and scenario s

By using the myopic method, the multi-stage problem is decomposed to several two-stage problem. The first stage variable is seed selection, and the second stage variable is node activation and node repost decision. The given parameters are including the node preference, the probability of current network, and the node repost decision of the previous time period. Since we select seed to find the maximal expected influence at current time period, the decision only happens within one time period. Then the time index and set can be removed and the node repost decision of the previous time period should be added in the known parameter. The notation of myopic model is shown in Table 3.2. The mathematic formulation of myopic model is shown below:

$$[\text{MYSP}] \quad \max_{x,y} \quad \mathbb{E}(R(y); \varepsilon) = \sum_{s \in \mathcal{S}} P^s(a) \cdot R^s(y) \tag{3.2a}$$

$$s.t. \quad P^s(a) = \prod_{i \in \mathcal{I}} \prod_{j \in \mathcal{I} \setminus \{i\}} \text{Prob}(a_{ij}^s = 1) \qquad \forall\, s \in \mathcal{S} \tag{3.2b}$$

$$R^s(y) = \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} w_k \cdot (2b_{ki} - 1) \cdot (y_{ki}^s - c_{ki}) \qquad \forall\, s \in \mathcal{S} \tag{3.2c}$$

$$\sum_{i \in \mathcal{I}} x_{ki} = 1 \qquad \forall\, k \in \mathcal{K} \tag{3.2d}$$

$$y_{ki}^s \geq c_{ki} \qquad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \tag{3.2e}$$

$$y_{ki}^s \geq x_{ki} \qquad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \tag{3.2f}$$

$$y_{ki}^s \geq \frac{\displaystyle\sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^s \cdot (d_{ki} + x_{kj} - d_{ki} \cdot x_{kj})}{\displaystyle\sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^s} \qquad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I} \tag{3.2g}$$

$$y_{ki}^s \leq c_{ki} + x_{ki} + \sum_{j \in \mathcal{I} \setminus \{i\}} a_{ji}^s \cdot (d_{ki} + x_{kj} - d_{ki} \cdot x_{kj}) \qquad \forall\, s \in \mathcal{S},\, k \in \mathcal{K},\, i \in \mathcal{I}$$

$$\text{(3.2h)}$$

$$x \in \mathbb{B}, \quad y \in \mathbb{B}, \quad z \in \mathbb{B}$$

When time $t > 0$, some known parameters is given by the previous myopic model.

$$
\begin{aligned}
c_{ki} &= \hat{y}_{ki} & \forall\, k \in \mathcal{K},\, i \in \mathcal{I} \\[4pt]
d_{ki} &= b_{ki} \cdot (\hat{y}_{ki} - \hat{c}_{ki}) & \forall\, k \in \mathcal{K},\, i \in \mathcal{I} \\[4pt]
u_{ij} &= \sum_{k \in \mathcal{K}} (2b_{kj} - 1) \cdot \hat{a}_{ij} \cdot (\hat{d}_{ki} + \hat{x}_{kj} - \hat{d}_{ki} \cdot \hat{x}_{kj}) & \forall\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\} \\[4pt]
\text{Prob}(a_{ij}^s = 1) &= \frac{1}{1 + exp(-u_{ij})} & \forall\, i \in \mathcal{I},\, j \in \mathcal{I} \setminus \{i\}
\end{aligned}
$$

where $\hat{y}_{ki}$ is the activation status using the decision of previous seed selection $\hat{x}_{kj}$, $\hat{c}_{ki}$ is the parameter of previous myopic model, and $\hat{d}_{ki}$ is the node repost decision using the decision of previous seed selection $\hat{x}_{kj}$. The parameter transition between two myopic models is shown in Figure 3.3.

3.3.2 Reinforcement Learning with Markov Decision Process

Our problem can be defined as a Markov Decision Process (MDP), that how information provider choose source user when facing the given information activation status of all user in the network. We use the Reinforcement Learning to learn the policy based on state-action pairs $(\boldsymbol{s}, \boldsymbol{a})$. The notation of reinforcement learning with Markov decision process model is shown in Table 3.3. In general, MDP is described by a 4-tuple (S, A, P, R), which are the states, actions, transitions, and reward. In our problem, these four terms are defined as below.

- S: the finite set of state, ie, activation status, $\boldsymbol{s} \in S$

- A: the finite set of action, ie, source user selection, $\boldsymbol{a} \in A$

- P: the probability of transition from s to s' through action a, $P_a(\boldsymbol{s}, \boldsymbol{s'})$

- R: the expected reward of transition from s to s' through action a, ie, weighted information influence, $R_a(\boldsymbol{s}, \boldsymbol{s'})$

The probability function is not unknown since the network topology is uncertainty. The reward function is shown below:

$$R(\boldsymbol{s}, \boldsymbol{s'}) = \sum_{k \in K} \sum_{i \in I} w_K \cdot (s'_{ki} - s_{ki}) \tag{3.3a}$$

We will introduce the Q-learning algorithm to compute optimal policies, which includes policy evaluation and policy inprovement.
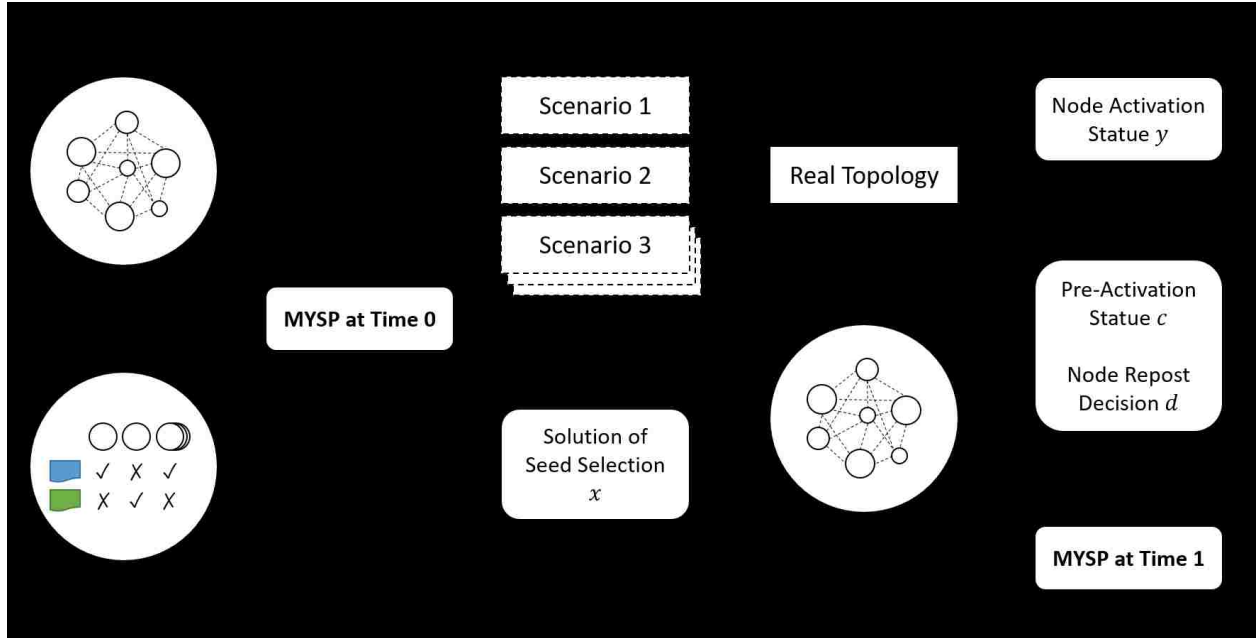
Figure 3.3: Myopic Model: Parameter Transition

Policy Evaluation

If we have a policy, the probability of actions taken at each state are known. Then the MDP is turned into a Markov chain (with rewards). We can compute the expected total reward collected over time using this policy. For given policy $\pi(\boldsymbol{s})$, the state-value function $Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ is used the evaluated the policy value.

$$Q^{\pi}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}^{\pi}\big(R(\boldsymbol{s}, \boldsymbol{s}') + \gamma \cdot \sum_{\boldsymbol{a}' \in A} \pi(\boldsymbol{s}', \boldsymbol{a}') \cdot Q^{\pi}(\boldsymbol{s}', \boldsymbol{a}')\big) \ \forall \ \boldsymbol{s} \in S, \ \boldsymbol{a} \in A \qquad (3.3\mathrm{b})$$

where γ is the discount factor and $\pi(\boldsymbol{s}, \boldsymbol{a})$ is the probability to take action $\boldsymbol{a}$ at state $\boldsymbol{s}$.

Consider a network with node size $|\mathcal{I}| = 4$ and information size $|\mathcal{K}| = 2$. The size of state

Table 3.3: Notation of Reinforcement Learning with Markov Decision Process Model

Symbol	Definition
	Indices and Sets
$i \in \mathcal{I}$	node
$k \in \mathcal{K}$	message
	Parameters
b_{ki}	the information preference of node i with respect to message k
w_k	the influence weight of message k
ρ_{ij}	the probability of arc connection from node i to node j
	Variable
σ_{ki}	the element of state matrix $\boldsymbol{s} \in S$ in row k and column i, that the activation status of node i by message k
α_{ki}	the element of action matrix $\boldsymbol{a} \in A$ in row k and column i, that the seed selection of node i by message k

set is $|S| = 2^{|\mathcal{K}| \cdot |\mathcal{I}|} = 256$ and the size of action set $|A| = |\mathcal{I}|^{|\mathcal{K}|} = 16$. Given initial state (no activation) $\boldsymbol{s}$, the information provider has a trivial policy $\pi(\boldsymbol{s})$, that each node has equally probability to be seed.

$$
\boldsymbol{s} = \left. \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\[2mm] \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \end{pmatrix} \right\} \text{information } k = \begin{pmatrix} 0 & 0 & 0 & 0 \\[2mm] 0 & 0 & 0 & 0 \end{pmatrix}
$$
$$
\underbrace{\phantom{\sigma_{11} \quad \sigma_{12} \quad \sigma_{13} \quad \sigma_{14}}}_{\text{user } i}
$$

$$
\pi(\boldsymbol{s}) = \left. \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\[2mm] \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \end{pmatrix} \right\} \text{information } k = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\[2mm] 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}
$$
$$
\underbrace{\phantom{\alpha_{11} \quad \alpha_{12} \quad \alpha_{13} \quad \alpha_{14}}}_{\text{user } i}
$$

We run several simulations of independent cascade with random actions and discount factor $\gamma = 1$. The average final influence of each action is shown in Table 3.4. Figure 3.4 shows the same policy is applied in different state to calculate the expected total reward, that is the

Table 3.4: Example of Policy Evaluation

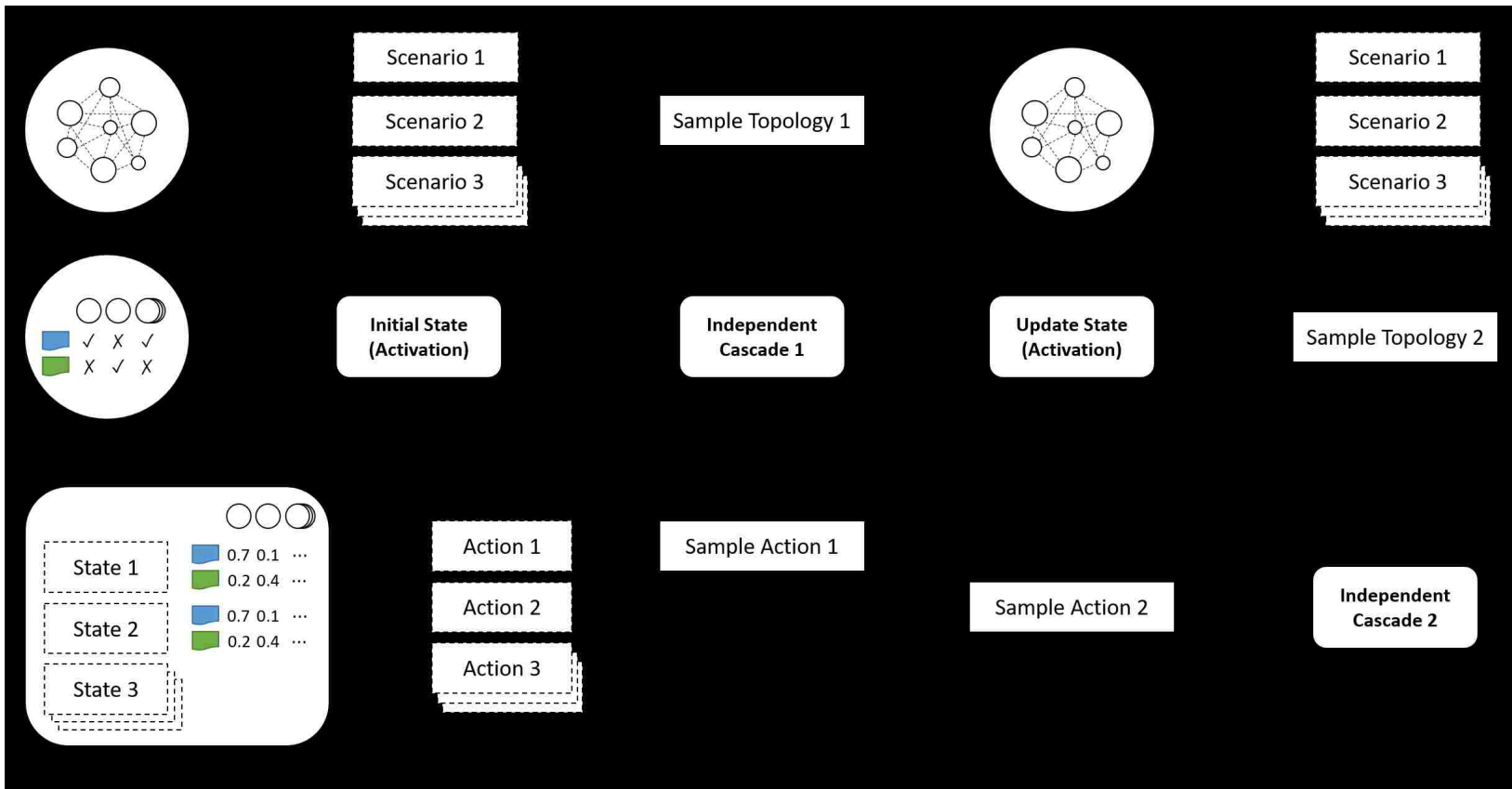| state $\boldsymbol{s}$	action $\boldsymbol{a}$	influence $Q^\pi(\boldsymbol{s}, \boldsymbol{a})$
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$	3.27869
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	3.09836
$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	3.22414
	$\vdots$	$\vdots$
	$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	3.90909

Figure 3.4: Reinforcement Learning: Policy Evaluation

total activated node at end of the time horizon.

Policy Improvement

Based on the simulation result, we create a final reward (weighted total influence) list $Q(\boldsymbol{s}, \boldsymbol{a})$ by state and action, which is used to improve the policy. $\pi(\boldsymbol{s}, \boldsymbol{a})$ and $\pi'(\boldsymbol{s}, \boldsymbol{a})$ are old policy and new policy. The action set A is splitted to two subset. A^1 is the set of all happened action, A^0 is the set of all unhappened action.

$$
\pi'(\boldsymbol{s}, \boldsymbol{a}) =
\begin{cases}
(1 - \sum\limits_{a \in A^0} \pi(\boldsymbol{s}, \boldsymbol{a})) \cdot \dfrac{Q(\boldsymbol{s}, \boldsymbol{a}) - \hat{Q}(s, a)}{\sum\limits_{a \in A^1} Q^\pi(\boldsymbol{s}, \boldsymbol{a}) - \hat{Q}^\pi(\boldsymbol{s}, \boldsymbol{a})}, & \forall\, \boldsymbol{a} \in A^1,\ \boldsymbol{s} \in S \\[4mm]
\pi(s, a), & \forall\, \boldsymbol{a} \in A^0,\ \boldsymbol{s} \in S
\end{cases}
$$

$$\hat{Q}(\boldsymbol{s}, \boldsymbol{a}) = \lambda \cdot \min_{\boldsymbol{a} \in A^1} Q(\boldsymbol{s}, \boldsymbol{a})$$

Table 3.5: Example of Policy Improvement

state $\boldsymbol{s}$	action $\boldsymbol{a}$	inital policy $\pi(\boldsymbol{s}, \boldsymbol{a})$	updated policy $\pi'(\boldsymbol{s}, \boldsymbol{a})$
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$	0.0625	0.0463788
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	0.0625	0.0515202
$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	0.0625	0.0554653
	$\vdots$	$\vdots$	$\vdots$
	$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	0.0625	0.0737093

where λ is the stepsize, which is decided by the iteration number and policy improved value.

$$\lambda = \frac{m}{itr^n} \cdot \sum_{s \in S} \sum_{a \in A} \left(\pi_{itr}(s, a) \cdot Q^{\pi_{itr}}(s, a) - \pi_{itr-1}(s, a) \cdot Q^{\pi_{itr-1}}(s, a) \right)$$

For the example of Policy Evaluation, the updated policy is shown in Table 3.5. If we summarized the policy by information k and user i, it will be

$$\pi(s) = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \end{pmatrix} \left.\vphantom{\begin{pmatrix}a\\a\end{pmatrix}}\right\} \text{information } k = \begin{pmatrix} 0.2056 & 0.2086 & 0.2959 & 0.2899 \\ 0.2255 & 0.2505 & 0.2697 & 0.2542 \end{pmatrix}$$

$$\underbrace{\phantom{\alpha_{11} \quad \alpha_{12} \quad \alpha_{13} \quad \alpha_{14}}}_{\text{user } i}$$

3.4 Computational Results

Numerical experiments and results of different algorithms are presented in this section on solving the information maximization problem. We randomly generate three data sets, small size (2 message, 4 node), medium size (2 message, 7 node), and large size (3 message, 7 node). The algorithms are coded in Microsoft Visual Studio 2019 C++ linked with CPLEX 12.9. All the programs are run in Microsoft Windows 10 Professional operating system with Intel Xeon CPU E-2186 2.90GHz and 32GB RAM.

In Figure 3.5 and Figure 3.5, we choose two sample sizes to test the algorithm of reinforcement learning with Markov decision process. The result shows the policy learned from small sample size cannot converge, because the policy evaluation using Monte Carlo simulation has low accuracy with small sample size. The policy learned from large sample size is significantly improved.

In Figure 3.7, we compare the algorithm of Two-Stage Stochastic programming with Myopic
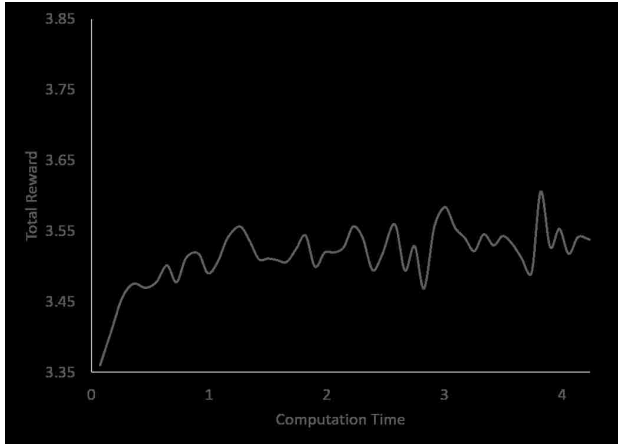
(a) Sample Size 10000
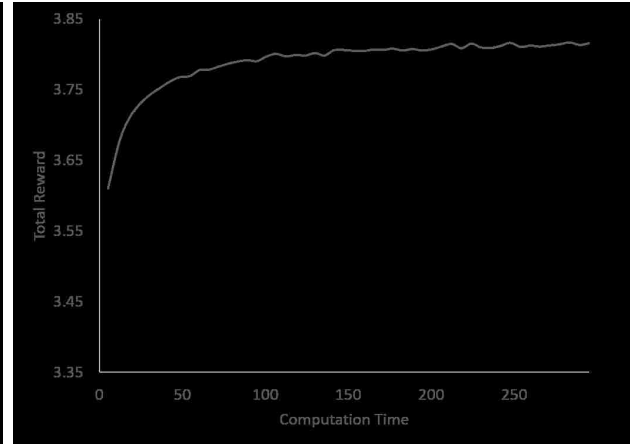
(b) Sample Size 1000000

Figure 3.5: Date Set (2,4) of Sample Size using RL-MDP
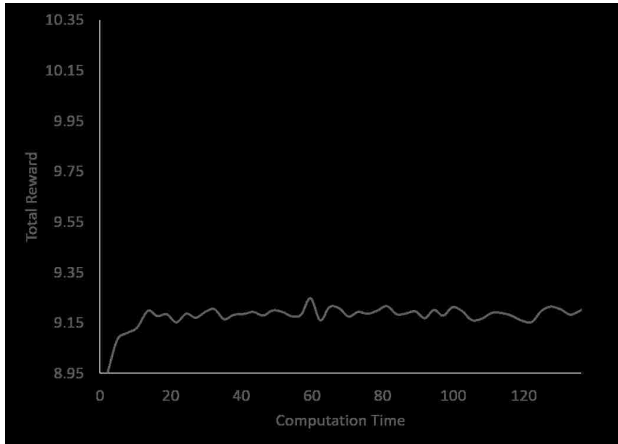
(a) Sample Size 10000
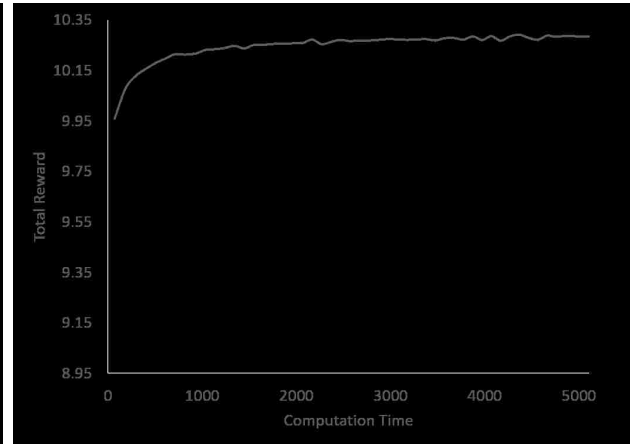
(b) Sample Size 1000000

Figure 3.6: Date Set (2,7) of Sample Size using RL-MDP

Policy (SP-MYOPIC) and the algorithm of Reinforcement Learning with Markov Decision Process (RL-MDP) using the different data set. The RL-MDP method can provide better performance for our influence maximization problem.
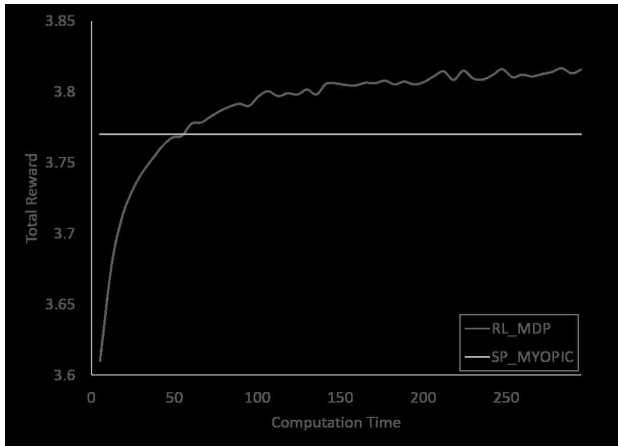
(a) Date Set (2,4)   **(b)** Date Set (2,7)

Figure 3.7: Algorithm Comparison, Sample Size 1000000

CHAPTER 4: MULTI-STAGE STOCHASTIC PROGRAMMING JOB-SHOP PROBLEM IN SEMICONDUCTOR MANUFACTURING

In a semiconductor wafer fabrication, there are multiple product types that have different due dates and different process flows. From all fabrication processes, photolithography process can be considered as the bottleneck step of each photo layer in wafer production, our model is designed to increase the 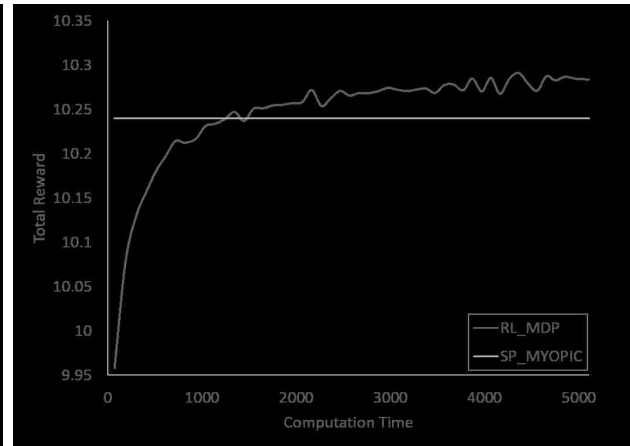efficiency of the production system by controlling the wafer flow of photolithography process to meet target production quantity. The production scheduling and dispatching of manufacturing can be modeled as a Job-Shop Scheduling Problem with Limited Capacity, which is used to find the optimal resource allocation and job dispatching of equipment and product lots. In the reality, the equipment capacity is unknown, because exact time down time is not able to predicted before the real happens. To solve this problem, we build the multi-stage stochastic programming model to plan the shift production scheduling and dispatching which can reduce the violation of the shift target by increasing the utilization of the equipment and increase the production efficiency.

4.1 Introduction

In information society, silicon microchips is required in a lot of area, such as computer, mobile phones, human-like robots and vehicles. Since the commercialization of new technologies such as AI and big data is promoting the digital transformation of human life, the integrated circuits (IC) are expected to evolve further in future intelligent society. The need for electronic components is growing exponentially. The electronics industry is already facing

Figure 4.1: Average Annual Growth Rates of Each IC Product Category, 2018-2023

several shortages and high volume demands for electronic components. The shortage was caused by a classic case of demand far outstripping supply. Besides some market factors, such as poorly-forecasted demands, new high-demand industries, and longer wait times on raw material, the main reason is fabrication shortage. Components across the board are going out of stock, such as small commodity type capacitors and resistors. For customer integrated circuits (IC), it need longer lead times due to the customer complex design. The McClean Report 2019 of IC Insights shows that the average annual growth rates from 2018 to 2023 is 6.8%, in Figure 4.1.

Since the supply is lower than the market demand, to get more customer order, efficient scheduling and dispatching can increase the equipment utilization and productivity. For higher performances, the modern electronic circuits have been designed into ultra-large-scaled integrated (ULSI) circuits. In 2007, '45 nm' commercial technology node is in volume production which need a lithograph capability of 65nm half pitch(HP) for the metal lines of DRAM. Photolithography (LITHO) can be considered as the bottleneck step of each photo

layer in wafer production. In our study, we build model to control the wafer flow of LITHO process to meet target production quantity.

Job-shop scheduling problem (JSP) is typical combinatorial optimization problems in operations research in which jobs are assigned to resources at particular times. In our job-shop model, we use the maximum target production quantity as the objective function instead of minimum makespan, which is converted by the product due date and order quantity. Sometimes, it may not be able to reach the target production quantity due to the limited equipment capacity, the objective is to find the optimal planing of entire product by allocating the wafer to different equipment, which aims to minimize the shortage between the expected production quantity and target production quantity. Since we cannot know the equipment down time before the real happens, the equipment capacity is considered as the uncertainty in the job-shop problem. Based the historical data, we build the equipment reliability distribution, which can give the probability of equipment work status in the future. Based on the equipment reliability, we can build the time-based multi-stage stochastic programming model. Since the scenarios is exponential increasing, deterministic equivalent of the stochastic programming is still difficult to solve. We have 3 approaches to approximate the optimal solution.

4.1.1 Literature Review

Graham [24] firstly define a multiprocessing system, then based on this system, Taillard [57] build three basic models with makespan objective, that are job-shop scheduling, Open-shop scheduling, and Flow shop scheduling. In our project, we focus on job-shop scheduling problem. Job-shop problem (JSP) is an best known combinatorial optimization problem in operations research in which jobs are assigned to resources at particular times [58], which is

proved as NP-complete problem by Garey and Johnson [20]. Due to researchers efforts, many effective algorithms are developed for the basic job-shop scheduling model [12, 13]. Adams, Balas, and Zawack [2] gives an approximation method for solving the minimum makespan problem, that is an $O(n)$ longest path algorithm. Nakano and Yamada [47] find conventional genetic algorithm is able to solve job-shop problem effectively.

In real world, as one of the most complex of manufacturing environments, manufacturing scheduling problem in semiconductor fabrication facilities is more complex than the basic job-shop problem. Reasons for this include tightly constrained production processes, re-entrant process flows, expensive sophisticated equipment, variable demand, high levels of automation [46]. The production of a single wafer requires about 1000 processing steps and takes couple months [62]. With the emergence of highly automated wafer fabrication facilities (fabs), there is a compelling trend to extend the traditional automation scope to integrate with advanced decision technologies. Gupta and Sivakumar [30] give a brief review of the scheduling techniques in scheduling the semiconductor manufacturing processes, such as dispatching heuristics, mathematical programming techniques, neighborhood search methods, and AI techniques. Błażewicz, Domschke, and Pesch [7] summarize several exact or heuristic algorithm to solve the deterministic job-shop problem.

While the solution of deterministic job-shop problem is not suitable in real production, since the scheduled job may be failed to allocation due to the broken machine. Foo and Takefuji [18] define the job-shop problem by stochastic neural network, which is the first involve the uncertainty in the machine scheduling. Then Buzacott and Shanthikumar [9] introduce several stochastic models applied in the manufacturing systems. Tavakkoli-Moghaddam et al. [58] develop the stochastic programming model to minimize the difference between the delivery and the completion times of jobs. They also propose simulated annealing algorithm while it is suitable to use in the large-scale problems. Li and Gao [40] propose a hybrid

algorithm (HA) which hybridizes the genetic algorithm (GA) and tabu search (TS) in flexible job-shop scheduling problem , which gives a method to solve the large-scale job-shop problem.

For manufacturing production planning and scheduling, equipment throughput is one of the most critical parameters. We optimize equipment throughput by multiple performance measures at the same time. There are two key performance indicators to show the wafer production status, Work-In-Process(WIP) and Ideal Production Quantity (IPQ) [55]. WIP shows the manufacturing lots in the factory not yet completed, which is waiting to process in the assigned machine. IPQ is the concept based on the cycle time, which is defined by Leachman, Kang, and Lin [39]. IPQ is used to calculate the production quantity needed to bring the actual downstream WIP up to the target WIP level by the end of the shift, considering that one more shift's worth of fab outs must be added to the downstream WIP to replace the fab outs due in the current shift [38]. In our problem, we use the IPQ as the shift target production quantity, that gives the amount of units need to be completed by the end of the shift to meet the target cycle time and the target fab outs. We schedule the job allocation considering the uncertain equipment status.

4.1.2   Our Contributions

To increase the equipment utilization and maximize the total throughput, we model problem as a problem of product lots to matched equipment. In the semiconductor manufacturing, we assume the production environment of the LITHO process is fully automotive, that the lot transportation time between two equipment and chemistry refill time can be ignored.

In this project, we propose an scheduling and dispatching model through semiconductor fabrication LITHO process with one shift time period. In this production environment, the operation sequence and the matching between operation and equipment are given and fixed.

While the equipment capacity is changed, because some machines of the equipment type may be broken, which will reduce the equipment capacity. Our model can help the production planner to schedule the time to dispatch the certain product quantity to their matched equipment. The stochastic formulation considers exogenous uncertainty, which represents the probability of the equipment capacity level. We design a two-stage stochastic linear programming to solve this production planning problem.

We summarize the contributions of this project as follows:

- We introduce the concept of uncertain equipment capacity in the manufacturing production planning problem.
- We implicate the practical algorithms for solving the two-stage stochastic programming with exogenous uncertainty.
- We compare the result of stochastic programming model with the deterministic model and calculate the value of stochastic solution.

The rest of this chapter is structured as follows. We briefly describe the production planning production with uncertain equipment capacity by lot scheduling and dispatching in finite time horizon, and provide the two-stage stochastic programming models with several assumptions in Section 3.2. In Section 3.3, we use the sample average approximation (SAA) to solve this problem with large-scale production plan. The computational results are shown in the Section 3.4.

4.2 Mathematical Models

Different product are using same equipment type while the equipment capacity is limits. Now we meet the problems: which product should be assigned and how many quantity should be

dispatched? To solve these problems, we make a plan in order to reduce the shortage from the production target by building the stochastic model. The target is defined by the shift Ideal Production Quantity (IPQ) value of key operation of each layer. Since the Litho is used as the first operation of each layer, our model only consider the LITHO operation to schedule the start time of the layer. Let's start with the scheduling and dispatching of single product.

For example, consider single product planning with 4 operations in future 3 time periods. For each time period, the released wafer quantity is shown in Table 4.1. Assume each operation use the unique equipment (no capacity share). The process time and capacity is shown in Table 4.2.

Table 4.1: Single Product: Released Wafer Quantity

	Time 1	Time 2	Time 3
Operation 1	3	2	1
Operation 2	0	2	0
Operation 3	2	3	1
Operation 4	0	0	0

Table 4.2: Single Product: Operation Process Time and Equipment Capacity

	Operation Process Time	Equipment Capacity
Operation 1	1	2
Operation 2	1	3
Operation 3	2	3
Operation 4	1	1

The supply of equipment is fixed all the time, but the demand from the operation is changed
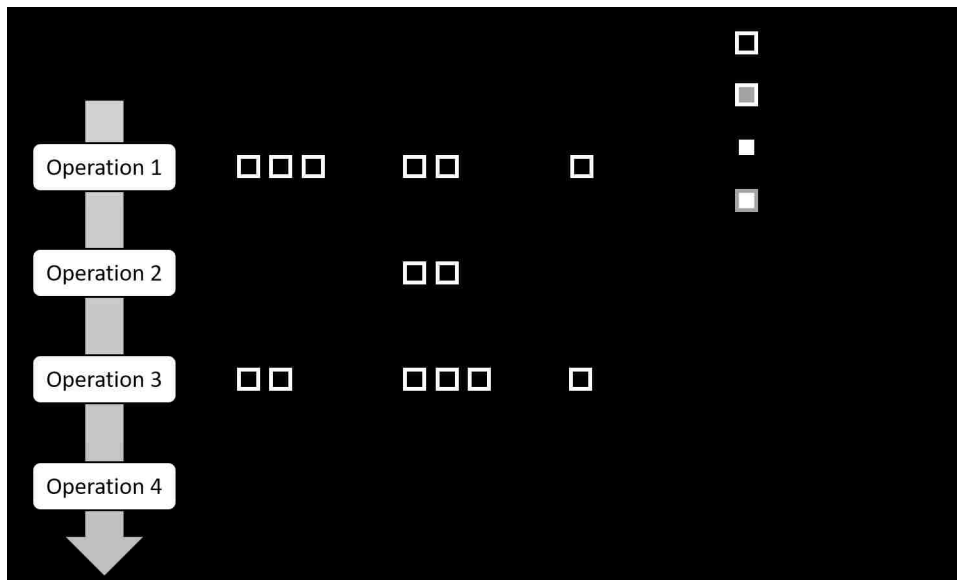
(a) Before Decision
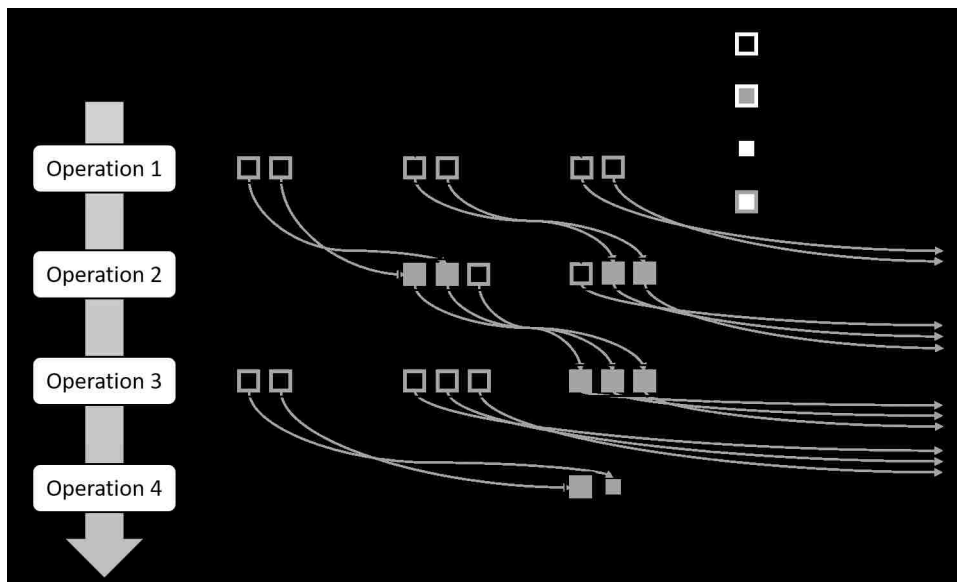
(b) After Decision

Figure 4.2: Single Product: Lot Scheduling and Dispatching

over time. Figure 4.2a shows the release time and quantity of different wafer lot. In Figure 4.2b, operation 1 has three lots waiting in the line, while the capacity only has two. So one

Table 4.3: Two Products: Matching between Operation and Equipment

	Product 1	Product 2
Equipment 1	Operation 1 Operation 2	Operation 1 Operation 3
Equipment 2	Operation 2 Operation 3 Operation 4	Operation 2 Operation 3

lot will be hold for next available time to dispatch. These two lots is completed by operation 1 after one process time, then they are become new wafer lot of operation 2 at time 2.

Consider there are multiple products, some operations are using the same equipment. Since the capacity is limited, the competition happens not only between products but also between operations. For example, there are two product is currently produced by two equipment in the fabrication. The product 1 has 4 operation and product 2 has operation. The matching
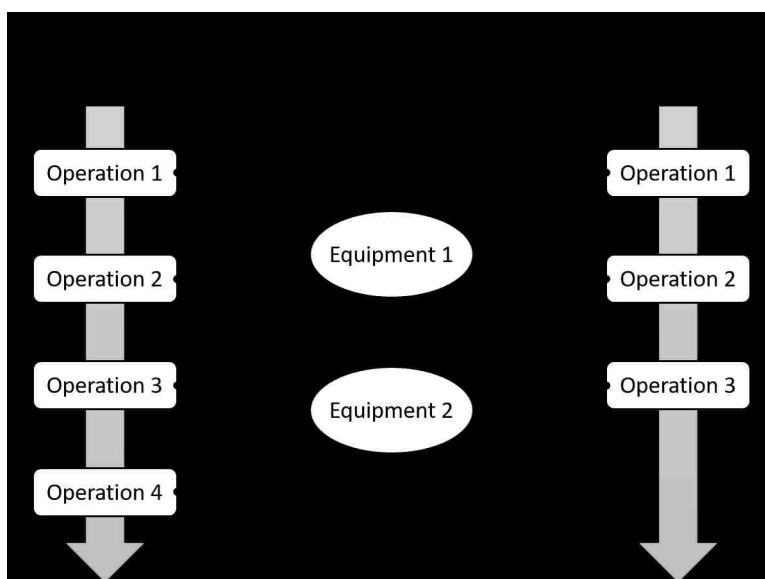
Figure 4.3: Two Products: Matching between Operation and Equipment

between operation and equipment is shown in Table 4.3 and Figure 4.3.

4.2.1 Deterministic Model

We consider the problem within the discrete finite-time horizon which aims to improve the production efficiency and equipment utilization. The objective can be measured by the total quantity of the operation completed wafers. Assume for different operation it has different

Table 4.4: Notation of Deterministic Model

Symbol	Definition
	Indices and Sets
$i \in I$	Product
$k \in K$	Equipment
$n \in N_i$	Operation, the set of the operation depends on product type
$t \in T$	Time
	Parameters
w_{in}	the weight of the operation n in product i, which represent the priority
d_{in}	the process time of the operation n in product i
b_{in}^k	the matching between the operation n in product i and the equipment k
$c^{t,k}$	the capacity of equipment k at time t
a_{in}^t	the quantity of the product released to dispatch at time i
	Decision Variable
$x_{in}^{t,k}$	continuous variable, the quantity of operation n in product i allocated to equipment k at time t
y_{in}	continues variable, the quantity completed at end of planning and partial dispatched
z_{in}^t	the quantity of product that is waiting to allocate from previous complete operation

time priority which is given from the concept of ideal production quantity (IPQ) and schedule score (SS). The notation of deterministic model is shown in Table 4.4. The formulation of the deterministic model is shown below:

$$[DP] \max_{x,y,z} \sum_{i \in I} \sum_{n \in N_i} w_{in} \cdot y_{in} \tag{4.1}$$

$$s.t. \quad y_{i,n} = \sum_{t \in T \setminus \mathcal{T}_{in}^3} \sum_{k \in K} x_{i,n}^{t,k} + \sum_{t \in \mathcal{T}_{in}^3} \sum_{k \in K} \frac{|T| - t}{d_{i,n}} \cdot x_{i,n}^{\tau,k} \quad \forall\, i \in I,\, n \in N_i \tag{4.2}$$

$$\sum_{\tau=0}^{t} \sum_{k \in K} x_{i,n}^{\tau,k} \le \sum_{\tau=0}^{t} \left(a_{i,n}^{\tau} + z_{i,n}^{\tau} \right) \quad \forall\, i \in I,\, n \in N_i,\, t \in T \tag{4.3}$$

$$z_{i,n}^t = 0 \quad \forall\, i \in I,\, n \in N_i,\, t \in \mathcal{T}_{in}^1 \tag{4.4}$$

$$z_{i,n}^t = \sum_{k \in K} x_{i,n-1}^{t-d_{i,n},k} \quad \forall\, i \in I,\, n \in N_i,\, t \in T \setminus \mathcal{T}_{in}^1$$

$$\tag{4.5}$$

$$\sum_{i \in I} \sum_{n \in N_i} x_{i,n}^{t,k} \le c^{t,k} \quad \forall\, i \in I,\, n \in N_i,\, t \in T \tag{4.6}$$

$$x \in \mathbb{R}^+, y \in \mathbb{R}^+, z \in \mathbb{R}^+$$

where Time set is separated to three subsets $\mathcal{T}_{in}^1$, $\mathcal{T}_{in}^2$ and $\mathcal{T}_{in}^3$, depending on the process time d_{in}, as shown in Figure 4.4.

- d_{in} - parameter, the process time of product i in operation n
- $\mathcal{T}_{in}^1$ - set, time period from the beginning to d_{in}
- $\mathcal{T}_{in}^3$ - set, time period from $(|T| - d_{in})$ to the end
- $\mathcal{T}_{in}^2$ - set, the absolute complement of set $\mathcal{T}_{in}^1 \cup \mathcal{T}_{in}^3$

The objective function (4.1) is maximizing the weighted quantity of operation completion, which depends on the product importance and the final processed quantity. The constraint
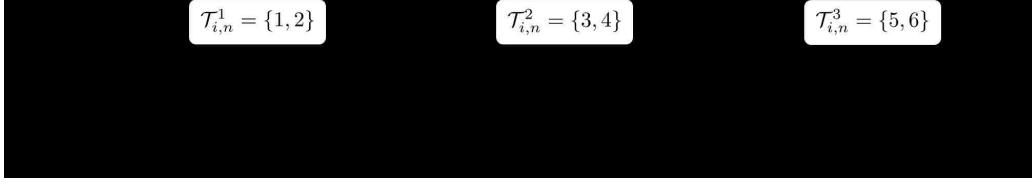
Figure 4.4: Partition of Time Set

(4.2) shows the final processed quantity of each product and operation at end of the planning, it has two parts:

- $\sum_{t \in T \setminus \mathcal{T}_{in}^3} \sum_{k \in K} x_{i,n}^{t,k}$ is the total quantity of wafer which is fully completed with target layer
- $\sum_{t \in \mathcal{T}_{in}^3} \sum_{k \in K} \frac{|T| - t}{d_{i,n}} \cdot x_{i,n}^{\tau,k}$ is the partial completion with the percentage of processing time

The dispatching demand constraint (4.3) shows the wafer quantity availability for dispatching. It obtains two parts:

- a_{in}^t - the quantity of the wafer released to dispatch at time i
- z_{in}^t - the quantity of the wafer from previous completed operation waiting to dispatch

With the same matching of operation and equipment in Figure 4.3, the dispatching decision for equipment 1 is shown in Figure 4.5. In constraint (4.3), the left hand side is the cumulative
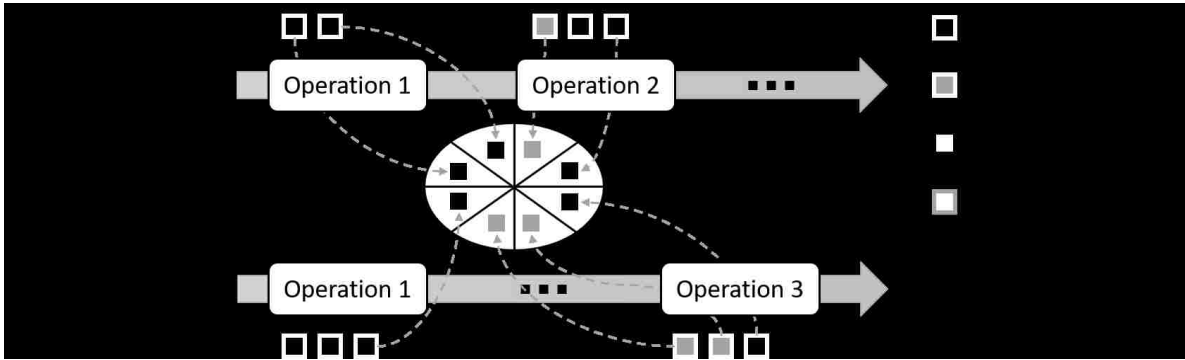
Figure 4.5: Two products with shared capacity

quantity of dispatched wafer, the right hand side is the cumulative quantity of released wafers and fresh wafers. There is a simple example (Table 4.5) shown how these constraints handle the production decision for each time period.

Table 4.5: Example of Dispatching

	Time 1	Time 2
Released Wafers	25	0
Fresh Wafers	50	10
Equipment 1 Capacity	50	50
Equipment 2 Capacity	50	50
Dispatched Wafers	50	35
Cumulative Available Wafers	75	85
Cumulative Dispatched Wafers	50	85
Waiting Wafers	25	0

Before operation time d_{in}, there is no fresh wafers produced, shown in constraint (4.4).Constraint (4.5) shows the product is going the next operation after it complete the previous operation. If it start the previous operation $n-1$ at time t, then it will be completed after d_{in} unit time. For the operation n, all wafer completed the previous operation before time t, is available to dispatch to matched equipment, which obtain three types:

- the fresh wafers $z_{i,n}^t$ just completed from the latest time period
- the released wafers from holding bank
- the waiting-in-line wafers $a_{i,n}^t$ due to the capacity limit

Constraint 4.6 shows the all dispatched wafer from different product and operation cannot exceed the equipment capacity.

The uncertainty of the stochastic model is the equipment capacity. Some equipment may be unexpected broken or needmaintenance, then the capacity will be reduced due to the shutdown equipment. For example, there are two equipment. The run status of equipment k and its probability p_k^t at time t are shown in Table 4.6.

Table 4.6: Probability of Equipment Status

	UP $a_k^t = 1$	**DOWN** $a_k^t = 0$
Equipment 1	0.7	0.3
Equipment 2	0.5	0.5

Table 4.7: Probability of Outcome

	Equipment Status	**Probability**
Outcome 1	UP/UP	0.35
Outcome 2	UP/DOWN	0.35
Outcome 3	DOWN/UP	0.15
Outcome 4	DOWN/DOWN	0.15

We define the combination of the equipment status as outcome. The size of outcomes is $2^{|K|}$. Table 4.7 shows there are four possible outcomes may happen at time t. The scenario is including the situations of all the time period. Let consider 2 time periods in Figure 4.6. The size of scenarios is $2^{|K| \cdot |T|}$. Based on the equipment status a_k^t and its probability p_k^t , the probability of the scenario can be formulated as below:

$$P^s = \prod_{t \in T} \prod_{k \in K} p_k^t (a_k^{t,s} = 1)$$
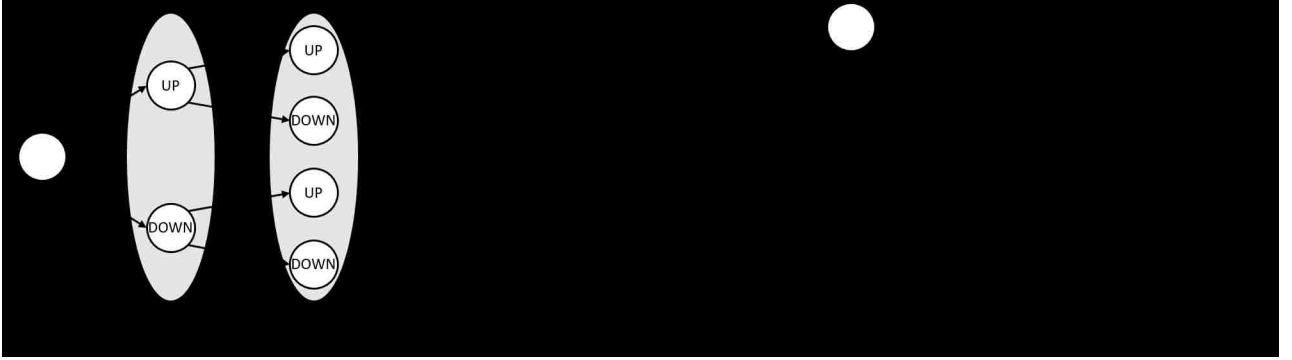
Figure 4.6: Construction of Scenario Tree

The two stage stochastic programming model of production planning problem can be formulated as,

$$[\text{SP}] \max_{\delta,x,y,z} \mathbb{E}\left[Q^{max}(x,y,z);\theta\right] = \sum_{s\in S} P^s \cdot Q^{max,s} \tag{4.7a}$$

$$s.t.\ Q^{max,s} = \max_{x^s,y^s,z^s} \sum_{i\in I}\sum_{n\in N_i} w_{in}\cdot y_{in}^s \qquad \forall\, s\in S \tag{4.7b}$$

$$y_{i,n}^s = \sum_{t\in T\setminus\mathcal{T}_{in}^3}\sum_{k\in K} x_{i,n}^{t,k,s} + \sum_{t\in\mathcal{T}_{in}^3}\sum_{k\in K} \frac{|T|-t}{d_{i,n}}\cdot x_{i,n}^{\tau,k,s} \qquad \forall\, s\in S,\, i\in I,\, n\in N_i \tag{4.7c}$$

$$\sum_{\tau=0}^{t}\sum_{k\in K} x_{i,n}^{\tau,k,s} \leq \sum_{\tau=0}^{t}\left(a_{i,n}^{\tau} + z_{i,n}^{\tau,s}\right) \qquad \forall\, s\in S,\, i\in I,\, n\in N_i,\, t\in T \tag{4.7d}$$

$$z_{i,n}^{t,s} = 0 \qquad \forall\, i\in I,\, n\in N_i,\, t\in \mathcal{T}_{in}^1 \tag{4.7e}$$

$$z_{i,n}^{t,s} = \sum_{k\in K} x_{i,n-1}^{t-d_{i,n},k,s} \qquad \forall\, i\in I,\, n\in N_i,\, t\in T\setminus\mathcal{T}_{in}^1 \tag{4.7f}$$

$$\sum_{i\in I}\sum_{n\in N_i} x_{i,n}^{t,k,s} \leq c^{t,k,s} \qquad \forall\, i\in I,\, n\in N_i,\, t\in T \tag{4.7g}$$

85

$$x \in \mathbb{R}^+, y \in \mathbb{R}^+, z \in \mathbb{R}^+$$

4.3 Solution Approaches

Difficulty of Solving Stochastic Model is that the scenario size is exponential growth, which lead to the computation time of solving stochastic model also increase exponentially. For the same length of time horizon $|T|$, the computation complexity of deterministic model is $O(|I| \cdot |K|)$, the computation complexity of stochastic model is $O(2^{|I| \cdot |K|})$. To handle this large-scale scenarios, we implement sample average approximation (SAA) design a problem-specific Algorithm 5 for our problem, which is similar to Algorithm 1 in Chapter 2.

Algorithm 5 Sample Average Approximation (SAA)

1: Initialization: given confidence level (CL), the significance level $\alpha = 1 - \text{CL}$ and confidence interval (CI)

2: **for each** equipment index $k \in \mathcal{K}$ **do**   ▷ Calculate sample size for each patient choice

3:     Calculate the variance σ_k^2 and the mean μ_k

4:     Define the sample size N_k, that the interval $\left(\mu_k - \dfrac{z_{\alpha/2} \cdot \sigma_k}{\sqrt{N_k}}, \mu_k + \dfrac{z_{\alpha/2} \cdot \sigma_k}{\sqrt{N_k}} \right)$ is within the confidence interval

5:     **return** the minimal required sample size $N_k \geq \underline{N_k} = \dfrac{z_{\alpha/2}^2 \cdot \sigma_k^2}{\text{CI}^2}$

6: **end for**

7: Define the sample set $\mathcal{N}$, where the set size is equal to the minimal required sample size for scenarios $|\mathcal{N}| = \max_{k \in \mathcal{K}} \underline{N_k} \leq ||\mathcal{S}||$ ▷ Find sample size for scenarios

---

8: **for each** sample index $n \in \mathcal{N}$ **do**

9:      Use Monte Carlo method randomly generate the equipment capacity matrix C^n

10:     Update the parameters of model (SP)

11:     Solve subproblem to get the objective value $Q^{max,n}$

12: **end for**

13: The objective value of all scenarios can be approximated by $\dfrac{1}{|\mathcal{N}|} \sum\limits_{n \in \mathcal{N}} Q^{max,n}$

4.4 Computational Results

Numerical experiments and results are presented in this section on solving deterministic model (DP) and stochatic model (SP). We randomly generate 3 data sets, small size (3 Product, 2 Equipment), medium size (4 Product, 2 Equipment), and large size (6 Product, 3 Equipment). The largest date set is similar to a 80% production quantity in a real semiconductor manufacturing. The algorithms are coded in Microsoft Visual Studio 2015 C++ linked with CPLEX 12.8. All the programs are run in Microsoft Windows 10 Professional operating system with Intel Xeon CPU E3-1535M v63.10GHz and 16GB RAM. The computation time (sec) of these three data set is shown in Table 4.8. The confidence level of SAA is 99% and the confidence interval is 0.01μ.

Table 4.8: Computation Time

| Data set (I, K)	Deterministic Model	Stochastic Model using Direct Method	Stochastic Model using SAA Method
$(3, 2)$	0.027	8.527	0.132
$(4, 2)$	0.038	34.622	3.940
$(6, 3)$	0.082	13577.986	801.573

We also compare solution of these two different models, deterministic model (DP) and stochastic model (SP). To see which model is able to get higher product throughput within limited time, suppose we just replaced the decisions of wafer scheduling time and dispatching quantity from deterministic model solved that problem, then the Value of Stochastic Solution (VSS) is defined as below:
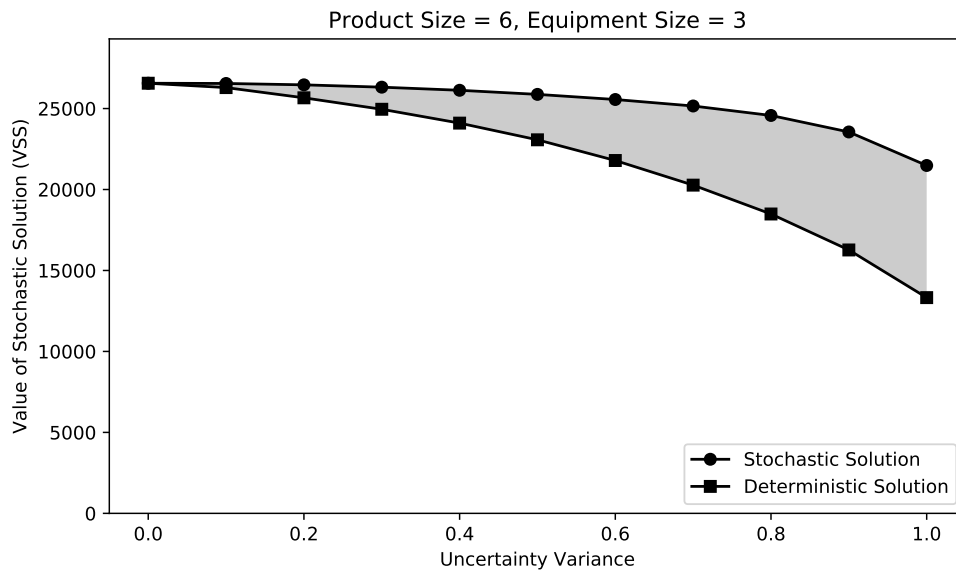
$$VSS = z_{SP}(x^*) - z_{SP}(x^o)$$

$$x^* = \arg \max_x z_{SP}(x)$$

$$x^o = \arg \max_x z_{DP}(x)$$

VSS is the area between the two lines in Figure 4.7. As the uncertainty variance increasing, the production planner will get more benefits from stochastic model.
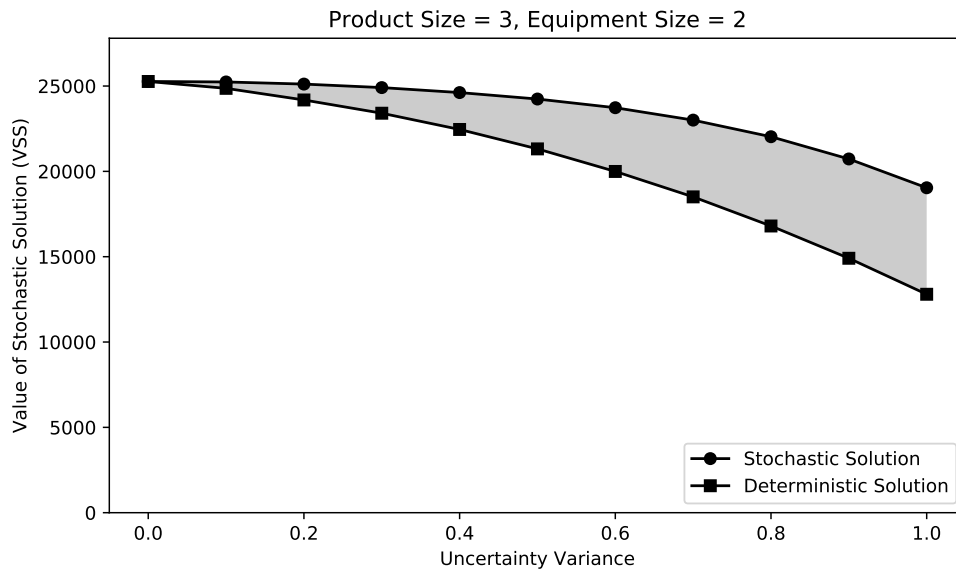
(a) VSS of date set $(3, 2)$

(b) VSS of date set $(6, 3)$

Figure 4.7: Value of Stochastic Solution

CHAPTER 5: CONCLUSION

This dissertation studied several stochastic optimization problems, applying the operations research methodologies in different areas, namely, health care, information networks and manufacturing.

Chapter 2 focused on a stochastic facility location problem considering customer preference, helping a hospital manager improve efficiency by allocating the hospital resource (sending physicians to local clinics), as well as matching the patient preference and maximizing patient satisfaction. The two-stage stochastic programming model was proposed for the physician/clinic facility location and patient assignment problem, where the patient preference was considered as endogenous uncertainty. To solve the model, we designed hybrid algorithms via the combination of the Large Neighborhood Search and Tabu Search to solve the location problem in the first stage, and Sample Average Approximation to estimate the value function in the second stage. The computational experiments showed that the proposed hybrid algorithms could outperform existing hill-climbing techniques, such as the Guided Local Search and Gradient Descent method, in terms of both solution quality and computational time.

Chapter 3 studied the network structure based on the user preference in a finite-time information cascade. Information networks are usually composed of autonomous nodes that make decisions when forming links with other nodes and transmitting information We used the Discrete Choice Model to build the node preference distribution, and the dynamic changing of network structure was modeled by Stochastic Dynamic Programming, which can be solve by the Markov Decision Process (MDP). In our model, DCM provided a good description and prediction of behavior for dynamic optimization under uncertainty. By solving our model,

we can analysis the changing of network structure by controlling information flow, which can be used in the Information maximization problem.

Chapter 4 studied the job-shop scheduling problem in semiconductor manufacturing. We built a multi-stage stochastic programming model to provide the optimal planning to the fabrication operation manager by allocating the product lot to the matched machine at the proper time. In the production system, there are several product types processed in the different operation steps, and they may share the same equipment. Thus, this problem can be considered as a resource allocation problem and assignment problem with limited capacity. The multi-stage stochastic programming model aims to make a shift production plan that can increase the utilization of the equipment and increase the production efficiency by reducing the violation of the shift target.

The contributions of this dissertation can be summarized as follows:

1. The dissertation presented three applications for stochastic optimization models in a complex network system. Each model studied the problem from a different perspective.

2. The three projects studied the network flow with uncertainty. Two of them incorporated the discrete choice model with decision-dependent probability for each stochastic programming model to faithfully model the real-world customer decision-making process.

3. This dissertation developed solution approaches that applied the decomposition, approximation, simulation, meta-heuristic and reinforcement learning algorithm techniques to reduce and solve real-world large-scale optimization problems.

LIST OF REFERENCES

[1] Ken Abrams, Andreea Balan-Cohen, and Priyanshi Durbha. "Growth in outpatient care The role of quality and value incentives". In: *Deloitte Center for Health Solutions* (2018).

[2] Joseph Adams, Egon Balas, and Daniel Zawack. "The shifting bottleneck procedure for job shop scheduling". In: *Management science* 34.3 (1988), pp. 391–401.

[3] Steven J Atlas, Richard W Grant, Timothy G Ferris, Yuchiao Chang, and Michael J Barry. "Patient–physician connectedness and quality of primary care". In: *Annals of Internal Medicine* 150.5 (2009), pp. 325–335.

[4] Hari Balasubramanian, Ana Muriel, Asli Ozen, Liang Wang, Xiaoling Gao, and Jan Hippchen. "Capacity allocation and flexibility in primary care". In: *Handbook of Healthcare Operations Management*. Springer, 2013, pp. 205–228.

[5] Opher Baron, Joseph Milner, and Hussein Naseraldin. "Facility location: A robust optimization approach". In: *Production and Operations Management* 20.5 (2011), pp. 772–785.

[6] Christiane Barz and Kumar Rajaram. "Elective patient admission and scheduling under multiple resource constraints". In: *Production and Operations Management* 24.12 (2015), pp. 1907–1930.

[7] Jacek Błażewicz, Wolfgang Domschke, and Erwin Pesch. "The job shop scheduling problem: Conventional and new solution techniques". In: *European journal of operational research* 93.1 (1996), pp. 1–33.

[8] Lawton R Burns and Douglas R Wholey. "The impact of physician characteristics in conditional choice models for hospital care". In: *Journal of Health Economics* 11.1 (1992), pp. 43–62.

[9] John A Buzacott and J George Shanthikumar. *Stochastic models of manufacturing systems.* Vol. 4. Prentice Hall Englewood Cliffs, NJ, 1993.

[10] Olveen Carrasquillo. "Health care utilization". In: *Encyclopedia of Behavioral Medicine.* Ed. by Marc D. Gellman and J. Rick Turner. New York, NY: Springer New York, 2013, pp. 909–910. ISBN: 978-1-4419-1005-9. DOI: 10.1007/978-1-4419-1005-9_885. URL: https://doi.org/10.1007/978-1-4419-1005-9_885.

[11] Wei Chen, Chi Wang, and Yajun Wang. "Scalable influence maximization for prevalent viral marketing in large-scale social networks". In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 2010, pp. 1029–1038.

[12] Runwei Cheng, Mitsuo Gen, and Yasuhiro Tsujimura. "A tutorial survey of job-shop scheduling problems using genetic algorithms—I. Representation". In: *Computers & industrial engineering* 30.4 (1996), pp. 983–997.

[13] Runwei Cheng, Mitsuo Gen, and Yasuhiro Tsujimura. "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies". In: *Computers & Industrial Engineering* 36.2 (1999), pp. 343–364.

[14] Fabián A Chudak and David P Williamson. "Improved approximation algorithms for capacitated facility location problems". In: *International Conference on Integer Programming and Combinatorial Optimization.* Springer. 1999, pp. 99–113.

[15] Morris A Cohen and Hau L Lee. "The determinants of spatial distribution of hospital utilization in a region". In: *Medical Care* (1985), pp. 27–38.

[16]   Isabel Correia and Francisco Saldanha da Gama. "Facility location under uncertainty". In: *Location Science.* Springer, 2015, pp. 177–203.

[17]   Tim Dall, Terry West, Ritashree Chakrabarti, Ryan Reynolds, and Will Iacobucci. "The Complexities of Physician Supply and Demand: Projections from 2016 to 2030". In: Association of American Medical Colleges, 2018. URL: `https://aamc-black.global.ssl.fastly.net/production/media/filer_public/bc/a9/bca9725e-3507-4e35-87e3-d71a68717d06/aamc_2018_workforce_projections_update_april_11_2018.pdf`.

[18]   Yoon-Pin Simon Foo and Yoshiyasu Takefuji. "Stochastic neural networks for solving job-shop scheduling: Part 1. Problem representation". In: *IEEE International Conference on Neural Networks.* Vol. 2. San Diego CA, USA. 1988, pp. 275–282.

[19]   Michael C Fu. "Stochastic gradient estimation". In: *Handbook of Simulation Optimization.* Springer, 2015, pp. 105–147.

[20]   Michael R Garey and David S Johnson. *Computers and intractability.* Vol. 29. wh freeman New York, 2002.

[21]   Deborah W Garnick, Harold S Luft, James C Robinson, and Janice Tetreault. "Appropriate measures of hospital market areas." In: *Health Services Research* 22.1 (1987), p. 69.

[22]   Vikas Goel and Ignacio E Grossmann. "A class of stochastic programs with decision dependent uncertainty". In: *Mathematical Programming* 108.2-3 (2006), pp. 355–394.

[23]   Jacob Goldenberg, Barak Libai, and Eitan Muller. "Talk of the network: A complex systems look at the underlying process of word-of-mouth". In: *Marketing letters* 12.3 (2001), pp. 211–223.

[24] Ronald L Graham. "Bounds for certain multiprocessing anomalies". In: *Bell System Technical Journal* 45.9 (1966), pp. 1563–1581.

[25] Mark Granovetter. "Threshold models of collective behavior". In: *American journal of sociology* 83.6 (1978), pp. 1420–1443.

[26] Paul M Griffin, Christina R Scherrer, and Julie L Swann. "Optimization of community health center locations and service offerings with statistical need estimation". In: *IIE Transactions* 40.9 (2008), pp. 880–892.

[27] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. "Information diffusion in online social networks: A survey". In: *ACM Sigmod Record* 42.2 (2013), pp. 17–28.

[28] Nalan Gülpınar, Dessislava Pachamanova, and Ethem Çanakoğlu. "Robust strategies for facility location under uncertainty". In: *European Journal of Operational Research* 225.1 (2013), pp. 21–35.

[29] Evrim Didem Güneş, Hande Yaman, Bora Çekyay, and Vedat Verter. "Matching patient and physician preferences in designing a primary care facility network". In: *Journal of the Operational Research Society* 65.4 (2014), pp. 483–496.

[30] Amit Kumar Gupta and Appa Iyer Sivakumar. "Job shop scheduling techniques in semiconductor manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 27.11-12 (2006), pp. 1163–1169.

[31] KRA Jakob and Peter Mark Pruzan. "The simple plant location problem: Survey and synthesis". In: *European Journal of Operational Research* 12 (1983), pp. 36–81.

[32] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM. 2003, pp. 137–146.

[33] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. "The sample average approximation method for stochastic discrete optimization". In: *SIAM Journal on Optimization* 12.2 (2002), pp. 479–502.

[34] Eike-Henner W Kluge. "Resource allocation in healthcare: Implications of models of medicine as a profession". In: *Medscape General Medicine* 51(1).57 (2007).

[35] Nan Kong and Peter J Fabri. "Optimal physician scheduling in a multi-clinic network to improve patient accessibility to outpatient care". In: *IIE Annual Conference. Proceedings.* Institute of Industrial and Systems Engineers (IISE). 2009, p. 1808.

[36] Madhukar R Korupolu, C Greg Plaxton, and Rajmohan Rajaraman. "Analysis of a local search heuristic for facility location problems". In: *Journal of Algorithms* 37.1 (2000), pp. 146–188.

[37] Alfred A Kuehn and Michael J Hamburger. "A heuristic program for locating warehouses". In: *Management Science* 9.4 (1963), pp. 643–666.

[38] Robert C Leachman and Shengwei Ding. "Integration of speed economics into decision-making for manufacturing management". In: *International Journal of Production Economics* 107.1 (2007), pp. 39–55.

[39] Robert C Leachman, Jeenyoung Kang, and Vincent Lin. "SLIM: Short cycle time and low inventory in manufacturing at Samsung Electronics". In: *Interfaces* 32.1 (2002), pp. 61–77.

[40] Xinyu Li and Liang Gao. "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem". In: *International Journal of Production Economics* 174 (2016), pp. 93–110.

[41] Yan Li, Nan Kong, Mengnan Chen, and Qipeng P Zheng. "Optimal physician assignment and patient demand allocation in an outpatient care network". In: *Computers & Operations Research* 72 (2016), pp. 107–117.

[42] Charles D MacLean, Beth Susi, Nancy Phifer, Linda Schultz, Deborah Bynum, Mark Franco, Andria Klioze, Michael Monroe, Joanne Garrett, and Sam Cykert. "Patient preference for physician discussion and practice of spirituality: Results from a multicenter patient survey". In: *Journal of General Internal Medicine* 18.1 (2003), pp. 38–43.

[43] Daniel McFadden. "Econometric models of probabilistic choice". In: *Structural Analysis of Discrete Data with Econometric Applications* 198272 (1981).

[44] Daniel McFadden et al. "Conditional logit analysis of qualitative choice behavior". In: (1973).

[45] Huseyin Onur Mete and Zelda B Zabinsky. "Stochastic optimization of medical supply location and distribution in disaster management". In: *International Journal of Production Economics* 126.1 (2010), pp. 76–84.

[46] Lars Mönch, John W Fowler, Stephane Dauzere-Peres, Scott J Mason, and Oliver Rose. "A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations". In: *Journal of scheduling* 14.6 (2011), pp. 583–599.

[47] Ryohei Nakano and Takeshi Yamada. "Conventional genetic algorithm for job shop problems." In: *ICGA*. Vol. 91. 1991, pp. 474–479.

[48] Peter J Nicholson and Nimrod Levy. "Solving the primary care crisis - A proposed strategy for physician allocation". In: *British Columbia Medical Journal* 51.1 (2009), pp. 12–13.

[49] Harri Oinas-Kukkonen. "A foundation for the study of behavior change support systems". In: *Personal and ubiquitous computing* 17.6 (2013), pp. 1223–1235.

[50] Susan Hesse Owen and Mark S Daskin. "Strategic facility location: A review". In: *European Journal of Operational Research* 111.3 (1998), pp. 423–447.

[51] Bernd Ploderer, Wolfgang Reitberger, Harri Oinas-Kukkonen, and Julia Gemert-Pijnen. "Social interaction and reflection for behaviour change". In: *Personal and ubiquitous computing* 18.7 (2014), pp. 1667–1676.

[52] Arnold S Relman. "The allocation of medical resources by physicians." In: *Academic Medicine* 55.2 (1980), pp. 99–104.

[53] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. "Prediction of information diffusion probabilities for independent cascade model". In: *International conference on knowledge-based and intelligent information and engineering systems*. Springer. 2008, pp. 67–75.

[54] Tjendera Santoso, Shabbir Ahmed, Marc Goetschalckx, and Alexander Shapiro. "A stochastic programming approach for supply chain network design under uncertainty". In: *European Journal of Operational Research* 167.1 (2005), pp. 96–115.

[55] Li Shi, Xiaohui Zhang, and Li Li. "Simulation and analysis of scheduling rules for semiconductor manufacturing line". In: *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*. IEEE. 2008, pp. 1–5.

[56] Lawrence V Snyder. "Facility location under uncertainty: a review". In: *IIE Transactions* 38.7 (2006), pp. 547–564.

[57] Eric Taillard. "Benchmarks for basic scheduling problems". In: *european journal of operational research* 64.2 (1993), pp. 278–285.

[58]   Reza Tavakkoli-Moghaddam, Fariborz Jolai, F Vaziri, PK Ahmed, and Amir Azaron.
       "A hybrid method for solving stochastic job shop scheduling problems". In: *Applied mathematics and computation* 170.1 (2005), pp. 185–206.

[59]   Vedat Verter. "Uncapacitated and capacitated facility location problems". In: *Foundations of Location Analysis*. Springer, 2011, pp. 25–37.

[60]   Chi Wang, Wei Chen, and Yajun Wang. "Scalable influence maximization for independent cascade model in large-scale social networks". In: *Data Mining and Knowledge Discovery* 25.3 (2012), pp. 545–576.

[61]   Dongyang Wang, Douglas J Morrice, Kumar Muthuraman, Jonathan F Bard, Luci K Leykum, and Susan H Noorily. "Coordinated Scheduling for a Multi-server Network in Outpatient Pre-operative Care". In: *Production and Operations Management* 27.3 (2018), pp. 458–479.

[62]   Lawrence M Wein. "Scheduling semiconductor wafer fabrication". In: *IEEE Transactions on semiconductor manufacturing* 1.3 (1988), pp. 115–130.

[63]   Linyun Yu, Peng Cui, Fei Wang, Chaoming Song, and Shiqiang Yang. "From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics". In: *arXiv preprint arXiv:1505.07193* (2015).

[64]   Yiduo Zhan and Qipeng P Zheng. "A multistage decision-dependent stochastic bilevel programming approach for power generation investment expansion planning". In: *IISE Transactions* 50.8 (2018), pp. 720–734.

[65]   Yiduo Zhan, Qipeng P Zheng, Jianhui Wang, and Pierre Pinson. "Generation expansion planning with large amounts of wind power via decision-dependent stochastic programming". In: *IEEE Transactions on Power Systems* 32.4 (2017), pp. 3015–3026.

[66] Mark Zlochin, Mauro Birattari, Nicolas Meuleau, and Marco Dorigo. "Model-based search for combinatorial optimization: A critical survey". In: *Annals of Operations Research* 131.1-4 (2004), pp. 373–395.