國立臺灣大學管理學院資訊管理學系碩士論文

Department of Information Management, College of Management

National Taiwan University

Master Thesis

達成網路存活性最大化之
近似最佳化網路防禦資源配置策略

Near Optimal Network Defense Resource Allocation Policies for Maximization of Network Survivability

溫雅芳

Ya-Fang Wen

指導教授：林永松，顏宏旭 博士

Advisor: Yeong-Sung Lin, Hong-Hsu Yen, Ph.D.

中華民國 96 年 7 月

July, 2007

國立臺灣大學碩士學位論文
口試委員會審定書

達成網路存活性最大化之
近似最佳化網路防禦資源配置策略

本論文係 溫雅芳 君（學號 R94725048 ）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 96 年 7 月 19 日承下列考試委員審查通過及口試及格，特此證明

口試委員： 林永松

顏宏旭

孫雅麗 祝國忠

所　　長： 孫雅麗

誌　　謝

當在寫這篇謝詞時，正代表著兩年來的研究生生活即將結束，心中除了些許的輕鬆，更多的是感謝與不捨。

首先，想感謝我的父母—溫春貞先生與彭梅貞女士，還有我的姐妹與朋友們—大姐、惠、琳、廖哥、文泰、順元…，我總是因為自己的忙碌，因而捨棄了許多與你們相處的機會，卻從未聽過你們的埋怨，只有聽到你們不斷地加油打氣，真的很感謝你們的包容與體諒。

當然，這兩年求學過程中，使我獲益最多的是我的指導教授林永松及顏宏旭博士，兩位老師悉心的教導使我得以一窺拉格蘭日鬆弛法的深奧。老師們對做學問的嚴謹、執著，與做人處世態度更是我學習的典範。林老師，我會把您的教誨：「不要這麼快就被 compromised!」永遠放在心中的。

此外，感謝柏皓學長帶領我進入資訊安全的領域，也特別謝謝建宏、義倫學長與中蓮學姐三位，你們總是不厭其煩的指出我研究中的缺失，且總能在我迷惘時為我解惑，由於有你們寶貴經驗的傳承，讓我減少許多盲目摸索的時間與挫折。

最後，要感謝實驗室的戰友們—岦毅、坤道、承賓、翊恆、俊維。因為你們，兩年的研究生活變得活潑有趣。回憶兩年來，幾乎每週都要 meeting 到半夜 11、12 點，雖然大家都很疲倦且焦味四溢，但大家在吃宵夜時，仍不忘互相鼓勵，偶爾還有元傑和鼎盛的加入，這些充滿歡樂的聚會，最後總是讓我帶著愉快的心情回家，使我又有滿滿的活力去面對未來的每一個挑戰。

溫雅芳　謹識

于台大資管所

九十六年七月

論文摘要

論文題目：達成網路存活性最大化之近似最佳化網路防禦資源配置策略

作者：溫雅芳 **九十六年七月**

指導教授：林永松、顏宏旭　博士

由於電腦硬體成本逐漸下降、軟體性能逐漸上昇，大部份的關鍵性網路都已電腦化控制。這些與日常生活息息相關的網路系統，一旦其毀損，除了對我們的生活造成極大的不方便，更是在生命與財產方面，引起不小的損失。所以，有效地評估與衡量關鍵性網路系統的存活性，是現今資訊安全領域中亟需重視的議題。

有鑑於此，我們提出一個全新且簡單的網路存活性指標—網路分隔度(Degree Of Separation, DOS)。這是一種網路傷害指標，用來衡量網路遭受毀損的平均程度。DOS 值愈大，代表其網路毀損愈嚴重，即表示必須付出更大的代價去修復整個網路。倘若其損害程度大於某一門檻值，則我們宣稱該網路已全然毀損。

因此，我們模擬一個網路攻防情境以建立一個最佳化資源配置目標之數學線性規劃模型，並加入 DOS 指標的概念來評估其存活性。在求解的過程之中，利用“拉格蘭日鬆弛法”與“梯度法”來幫助我們逐漸找到最佳解。

最後，經由實驗證明，不僅我們所提出的三階段選擇 (3-Stage Selection, 3SS) 攻擊演算法能夠有效評估攻擊成本，而且針對不同的網路拓樸所提出的網路資源配置策略效果顯著。

關鍵詞：網路分隔度、拉格蘭日鬆弛法、網路存活性、最佳化、資源配置、無尺度網路

THESIS ABSTRACT

THESIS TOPIC: Near Optimal Network Defense Resource Allocation Policies for Maximization of Network Survivability

NAME: Ya-Fang Wen **DATE: July, 2007**

ADVISER: Yeong-Sung Lin, Hong-Hsu Yen, Ph.D.

Due to the decreasing cost of computer hardware and the increasing capacity of computer software, most critical networks are being progressively computerized. If one of these systems were to fail, it would not only cause extreme inconvenience in our daily lives, but could even have catastrophic or fatal consequences. Thus, how to assess and evaluate the survivability of a system effectively is a crucial issue in the field of information security.

In this thesis, we propose a simple and novel metric of network survivability, called Degree of Separation (DOS). DOS is a survivability metric used to measure the average damage level of a system; naturally, the larger the DOS value, the more serious the network damage will be. If the DOS value is larger than a pre-established threshold, we say that the network has been compromised.

We express the scenario of network attack-defense as a mathematical linear programming model to near-optimize the resource allocation policies. In the process of problem solving, we adopt the concept of DOS to assess the network survivability and use the Lagrangean Relaxation method and the subgradient method to approach the optimal solution.

Finally, based on the experiment results, not only can the 3-Stage Selection (3SS) attack algorithm we proposed evaluate the attack cost effectively, but are the results of different defense budget allocation policies to different network topologies quite significant.

Keywords: Degree of Separation, Lagrangean Relaxation, Network Survivability, Optimization, Resource Allocation, Scale-free Network

Table of Contents

List of Figures

List of Tables

Chapter 1 INTRODUCTION

1.1 Background

How to assess and evaluate the survivability of a critical system effectively is now an urgent and crucial issue in the field of information security. With the continuous growth of the Internet and the World Wide Web, people are now connected far more closely than ever before. Communication networks have become not only a means of transmitting information, but also an essential part of our daily lives. As Taylor et al. note: "**We have become dependent on the computer networks that support our daily lives and the reliance on these networks have made us more vulnerable to their disruption.**" [6]

Many threats extend from the real world to the cyber space due to the popularity of the Internet. Familiar cyber threats include the spread of computer viruses, e-mail eavesdropping, information leakage, webpage defacement, and Denial of Service due to malicious attacks. New threats to the Internet are evolving almost daily. Indeed, computer viruses are evolving so fast that they cannot always be neutralized or eliminated by anti-virus products in a timely manner. The techniques of cyber criminals and computer hackers are advancing ceaselessly so that there is no such a thing as a perfect anti-intrusion or attack-proof system.

Due to the decreasing cost of computer hardware and the increasing capacity of computer software, most critical infrastructure systems, such as telecommunication systems, transportation systems, energy generation systems, banking systems, financial systems, medical care systems, and defense systems, are being progressively computerized. If one or more of these systems were to fail, it would not only cause extreme inconvenience in our daily lives, but could even have catastrophic or and fatal consequences.

A few years ago, the dependability and reliability of a system were important measurements of information security; however, these concepts no longer adequately express practical needs because they do not explore the notion of degraded service as an explicit requirement [1]. In the computer science, dependability is defined as the ability to deliver service that can justifiably be trusted [23]. Reliability is a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time [5]. Therefore, neither dependability nor reliability is enough to describe the real state of an attacked or a failed system, not to mention the Parkerian Hexad, including Confidentiality, Possession or Control, Integrity, Authenticity, Availability, and Utility, created by Donn B. Parker (http://en.wikipedia.org/wiki/Parkerian_hexad). We also need to consider the issue of survivability. As Knight et al. note: “To deal with events in information systems that might disrupt service leads to the notion of survivability.” [3]

In summary, we must be able to describe system states more exactly after disruptive incidents occur. Although the concept of system survivability is not new, until recently, it was not deemed an important metric of information security. We believe the issue warrants further study and research.

1.2 Motivation

A number of questions related to network survivability concern some researchers and experts in the field of information security; for example: “How seriously must a network be damaged before we say it is compromised?” and “How can we measure the survivability of an attacked network in our simulations by using Internet-like graphs?”

Maybe questions like those above are asked constantly because computerized systems, on which we are increasingly dependent, consist of a variety of imperfect and heterogeneous components. However, **these systems are not necessarily completely functional or dysfunctional, but may operate on a spectrum between the two extremes**.

The goal of science is to find meaningful simplicity in a situation of chaotic complexity. Therefore, in view of the previous questions, we try to create a network attack and defense model that maximizes survivability and allocates defense resources effectively.

Actually, in 1970, Zeitlin, a Dutch researcher, regarded the network attack and defense issue as an integer resource allocation problem with a min-max objective function [21]. However, even though Zeitlin and other contemporary researchers devoted themselves to the study of this issue, they did not consider the concept of survivability in their studies.

In our network attack and defense model, we not only adopt the concept of network survivability, but also propose a novel metric, called the Degree of Separation (DOS), to calculate network survivability. **DOS can be viewed as a network damage metric used to measure the average damage level of a network, i.e., the larger the DOS value, the more serious the network damage will be**. Thus, we should make every effort to fully recover a failed network. **The DOS value can be also used as a threshold to define a network crash**; that is, if the DOS value is larger than a pre-established threshold, we regard the network as out of control.

By applying the concept of DOS, we can calculate network survivability simply and intuitively, and also allocate limited network resources effectively.

1.3 Literature Survey

1.3.1 Network Survivability

With the continuous growth of the Internet and the prevalence of distributed networks, an increasing number of information security researchers are focusing on the issue of network survivability. While security has traditionally focused on confidentiality of information, the problems of greatest concern today relate to the availability of information and continued services [8].

Most systems have more than two security states; i.e., safe or compromised. Between these two extremes, there still exists an enormous vague zone. Thus, we should express more explicitly how secure (or vulnerable) a system is when we are planning and evaluating the system security blueprint. There are many potential problems, such as hardware failures, power failures, operator mistakes, disasters, and even sabotage. Therefore, we must be able to evaluate exactly how well a critical system can operate under different kinds and different levels of failure.

Although the definition of network survivability is very important, researchers' opinions differ, as shown by the summary in Table 1-1.

Table 1- 1: Summary of Survivability Definition

Researcher	***Definition***
J.C. Knight et al. [1] [2] [3]	A system that has the ability to continue providing service (possibly degraded or different) in a given operating environment when various events cause major damage to the system or its operating environment.
S.C. Liew et al. [4]	Expected survivability *E[S]* is the expected value of the survivability after a negative event. Worst-case survivability s^w is the minimum value of survivability; r-percentile survivability s_r is the probability that survivability would be less than r% of the total survivability; zero survivability, P_0, is the probability that survivability would be equal to zero.
V.R. Westmark [5]	The ability of a given system with a given intended usage to provide a pre-specified minimum level of service in the event of one or more pre-specified threats.
R.J. Ellison et al. [7]	Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures, and recover full services in a timely manner.
M.S. Deutsch et al. [22]	Survivability is the degree to which essential functions are still available, even though some part of the system is down.

From the works cited in Table 1-1, we observe a number of commonly used terms, as shown in Table 1-2.

Table 1- 2: Terms Commonly Used in Definitions of Survivability

Term	*Description*
The degree/ The ability/ The capability	Means that the concept of survivability is expressed as a percentage, not a constant.
A system/ A network	Describes the entity that needs to be protected.
Continuity and recovery of essential services/ Fulfillment of mission/ Provision of continuous operation	Describes the goal of an entity and to expresses the entity's level of robustness.
In the face of threats/ In the face of adverse environments/ under demanding conditions	Describes a negative event, including natural disasters, hostile attacks, or accidental errors.
Within a specified time period/ In a timely manner	Describes the conditions that must be satisfied.

In summary, the concept of survivability describes the adaptability of a system in the event of an abnormal incident. A survivable system is able to continue to provide tolerable service after some damage has occurred as a result of malicious attacks or unintentional failures.

Survivability can also be categorized in terms of network performance, network connectivity, network capacity, and traffic capacity. Furthermore, [4] formulates a general framework that includes and extends the existing definitions of network survivability, such that survivability is expressed as a function, not a single value. Therefore, we can derive a number of measurements from the survivability function.

As noted in [4], the measurement of survivability must be flexible enough to meet various conditions.

1.3.2 Scale-free Networks

Since the 9/11 terrorist attacks, researchers have focused increasingly on reducing the risks or minimizing the consequences of incidents or disasters. It is important to understand critical and complex distributed networks in modern society in order to explore their vulnerability. Therefore, many researchers spare no effort in trying to understand the relationship between a network's topological features and its vulnerability.

For decades, it has been assumed that many complex and large network topologies follow the random graph model (ER model) proposed by Paul Erdos and A. Renyi in the 1950s [16]. However, the statistical results of studies of real networks reveal that network topologies are far from completely random. The most obvious difference is that they follow a power law distribution. Power-laws are expressions of the connectivity distribution $P(k) \sim k^{-\gamma}$, which is the probability that a node is connected to k other nodes and the exponent γ is a constant.

Actually, power-law distributions are ubiquitous in the real world; for example, the Internet [14] [15], the World Wide Web (Figure 1-1a) [18] [19], actor collaboration graphs (Figure 1-1b) [19], science citation graphs (Figure 1-1c) [13], Newman's science co-authorship graphs, R.J. Williams et al.'s food web, Liljeros et al.'s web of human sexual contacts, Pareto's income distribution (which first introduced power-laws in 1896), Zipf's frequency law of English words, the human respiratory system, automobile networks, the size and location of earthquakes, stock-price fluctuations, biological cellular networks, and many more.

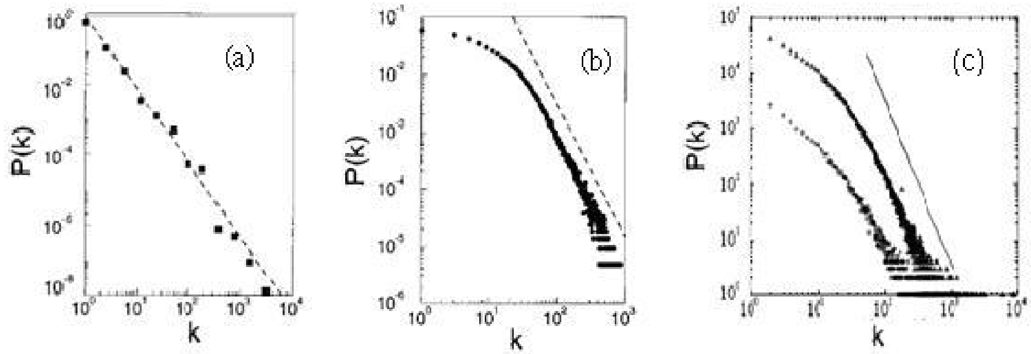


Figure 1- 1: Power-law Distribution Samples

(a) WWW graph with 325,729 nodes, where the average connectivity ⟨k⟩ = 5.46 and the slope $\gamma = 2.1$ [19].

(b) Actor collaboration graph in which the nodes denote actors and the links denote joint casting with 212,250 nodes; the average connectivity ⟨k⟩ = 28.78 and the slope $\gamma = 2.3$ [19].

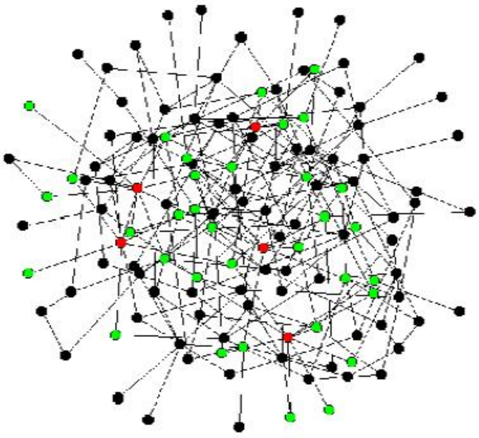
(c) Science citation graph in which the nodes denote papers and the links denote citations from the 783,339 papers in the ISI data set and the 24,296 papers in the PRD data set, with the slope $\gamma = 3$ [21].

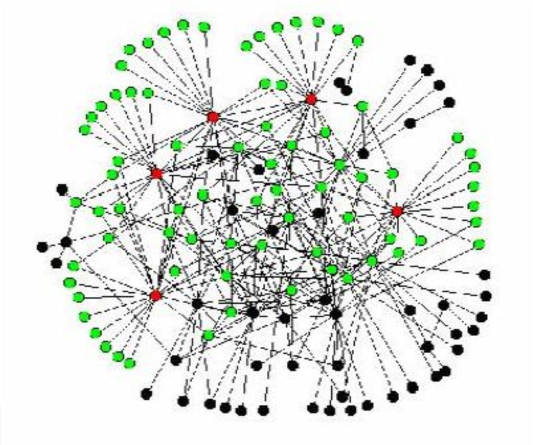
Barabási and Albert proposed a scale-free network that can describe the scaling properties of networks. They found that new nodes attach preferentially to existing nodes that are well connected. This discovery shows that the probability that a new node will connect to existing nodes is neither a uniform nor a random distribution.

However, there is a rather high probability that a new node will connect to an existing node that is well connected. Therefore, a network continuously expands due to the addition of new nodes. This is the so-called "rich-get-richer" phenomenon. In other words, the development of the power-law scale indicates that two critical factors play important roles in network development: **growth** and **preferential attachment**.

In view of the above description, today's complex networks can be divided into two major categories based on their network connectivity:

Table 1- 3: Network Distribution Samples

Exponential Networks	The *P(k)* peaks at an average <*k*> and decays exponentially for large *k*. The examples are the ER random graph model and the small-world model of Watts and Strogatz.  Figure 1- 2: Exponential Network [20] The ER model of random network yields a network with an exponential tail; it has two properties: **low clustering coefficient** and **high network diameter** [17].

Scale-free Networks	The *P(k)* decays as a power-law distribution. The probability that a node has a huge number of connections is statistically significant in scale-free networks. The examples were described earlier in this section.  Figure 1- 3: Scale-free Network [20] The BA model of scale-free networks has a power-law tail; its properties are similar to the “small-world” phenomenon: **high clustering coefficient** and **low network diameter** [17].

In terms of the “small-world” phenomenon, the interconnectedness of a network is described by its diameter, defined as the average number of hops on the shortest path between any two nodes in the network. Networks with a huge number of nodes can have a rather small diameter; for example the diameter of the WWW, with over 800 million nodes, is around 19 nodes, while social networks with over six billion people have a diameter of around 6 nodes.

Once we prove the existence of a power-law scale network, we can derive many advantages, such as designing more efficient network protocols, creating more accurate artificial models for simulation purposes, and estimating topological parameters, to make sure that our simulations accurately reflect real world scenarios. However, Barabási emphasizes that scale-free networks also have a number of vulnerabilities. Removal of 80% of the least-connected nodes would have little effect on the stability of a network; however, compromising some hubs, i.e., the most-connected nodes, can cause a network to crash. Scale-free networks with a high error tolerance of random failure come at a high price — they are extremely vulnerable to attack because an attacker targets the hubs in order to incur the most severe damage.

In other words, the error tolerance comes at the expense of survivability [20].

1.4 Thesis Organization

The remainder of the thesis is organized as follows. The next chapter explains and illustrates the concept of DOS. Chapter 3 uses a mathematical model to implement network attack and defense scenario with the concept of DOS. Chapter 4 introduces the Lagrangean Relaxation (LR) method for solving the model proposed in Chapter 3. Chapter 5 describes the computing experiment. Finally, in Chapter 6, we present our conclusions and discuss future work.

Chapter 2 DEGREE OF SEPARATION

2.1 Introduction

In [9], the author proposed two extreme survivability metrics to measure network survivability. The first metric is based on the connectivity of all critical Origin-Destination (OD) pairs. However, it is too strict to comply with the real case; thus, the second metric only considers the connectivity of at least one OD pair.

Based on [9], we believe that we can develop a more flexible metric for network survivability, no matter how many OD pairs are connected or disconnected. The proposed survivability metric, which assesses the average damage level of a network, is called the Degree of Separation (DOS); it is also called the Degree of Segmentation, Degree of Segregation, or Degree of Disconnectivity.

We begin by describing for the concept of DOS, and how it is calculated.

2.2 Illustration

In this section, we describe the concept and the calculation of DOS and provide some illustrations. Initially, the example network is intact (see Figure 2-1). We assume there is one OD pair (O is the source, and D is the destination) that can exist on more than one path between O and D. However, the transmission cost of the shortest path (thick line) is 7ε (the transmission cost of each link or each node is 1ε) and the DOS value, denoted as S’, is zero; that is, there is no any damage to this OD pair.

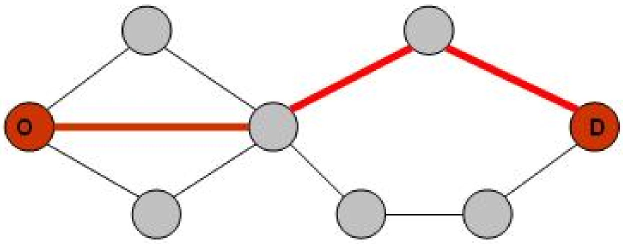


Figure 2- 1: Initial Network (1)

After the network has suffered a malicious attack, one node is dysfunctional (shown as a dotted circle in Figure 2-2), so the transmission cost turns into M, a enormous number. Therefore, the path of the smallest cost of this OD pair has to be changed to another route (thick line), whose path cost is 9ε. The S’ is still zero, which means the OD pair remains interconnected.

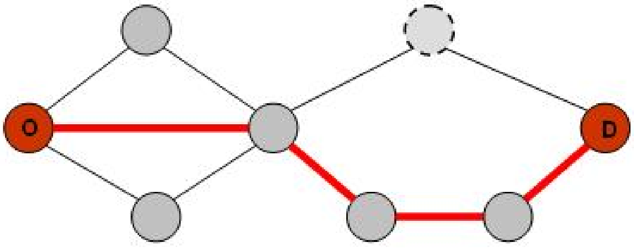


Figure 2- 2: Attacked Network with One Dysfunctional Node

In a more serious case (see Figure 2-3), the network is attacked such that two nodes, shown as dotted circles, are dysfunctional. At this time, if the OD pair wishes to keep communicating, it must pass through one broken node; that is, **we must pay a cost to fully recover this OD pair**. Therefore, the transmission cost of this pair is 1M+8ε, and the S'=1.

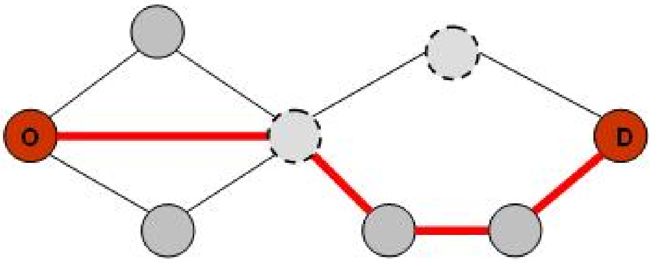


Figure 2- 3: Attacked Network with Two Dysfunctional Nodes

In the worst case (see Figure 2-4), the network is attacked so that three nodes are dysfunctional. The smallest transmission cost of this OD pair is 2M+5ε (thick line), and the S'=2; that is, **we have to fully recover two nodes to ensure that this OD pair can communicate**.

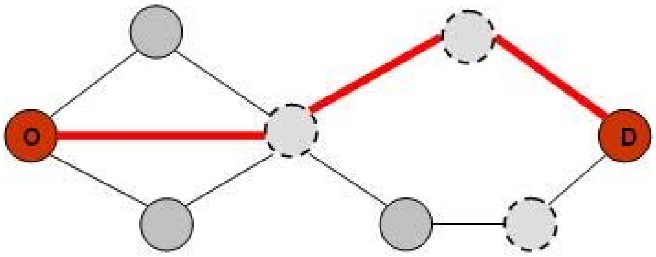


Figure 2- 4: Attacked Network with Three Dysfunctional Nodes (1)

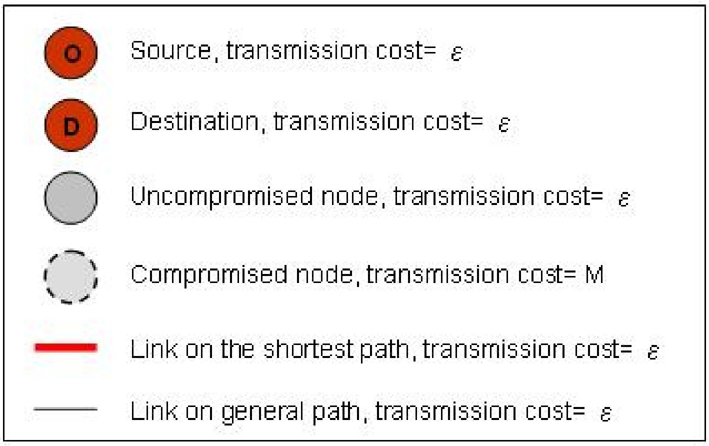

O Source, transmission cost= ε
D Destination, transmission cost= ε
Uncompromised node, transmission cost= ε
Compromised node, transmission cost= M
Link on the shortest path, transmission cost= ε
Link on general path, transmission cost= ε

Figure 2- 5: DOS Legend (1)

The above illustrations are for one OD pair in the network. We elaborate on the DOS calculation of a network with two dysfunctional pairs (see Figure 2-6). Initially, the network is intact and there exist two OD pairs (O_1-D_1 and O_2-D_2). The smallest transmission costs of the two OD pairs, respectively, are Cost$_{O1\text{-}D1}$=11ε and Cost$_{O2\text{-}D2}$=9ε; and the S’ $_{O1\text{-}D1}$=0 and S’ $_{O2\text{-}D2}$=0. That is, the two OD pairs keep communicating.

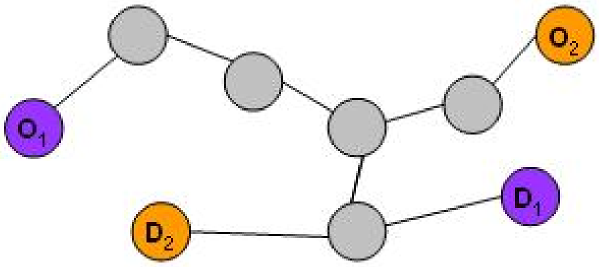


Figure 2- 6: Initial Network (2)

After the network has suffered a malicious attack, for example, three nodes are dysfunctional and two OD pairs are therefore disconnected (see Figure 2-7), and the smallest transmission costs respectively are $Cost_{O1\text{-}D1}$=3M+8ε and $Cost_{O2\text{-}D2}$=2M+7ε; and $S'_{O1\text{-}D1}$=3 and $S'_{O2\text{-}D2}$=2. The average damage to the whole network is $S'_{network}$=(3+2)/2=2.5; that is, **each OD pair has to pay 2.5 times the cost, on average, to fully recover network communications**.

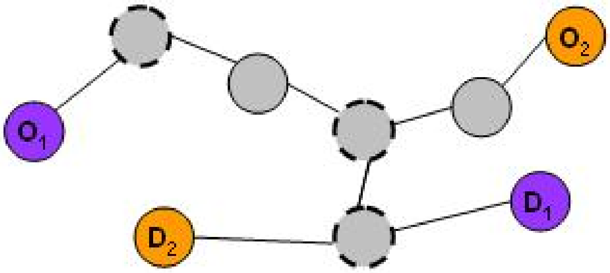


Figure 2- 7: Attacked Network with Three Dysfunctional Nodes (2)

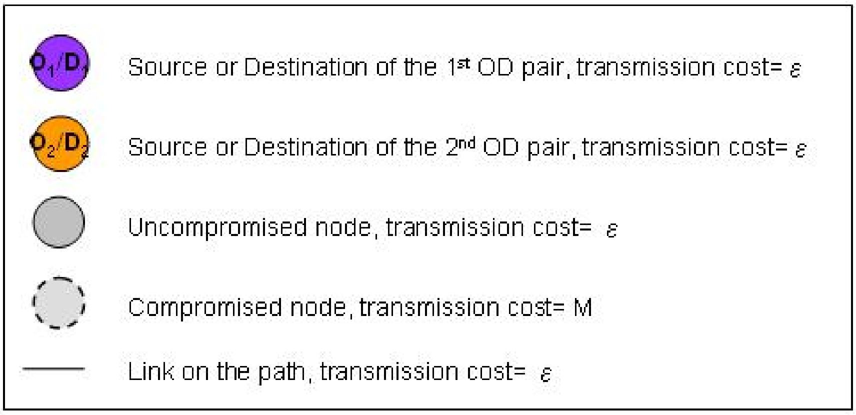

O_1/D_1
Source or Destination of the 1st OD pair, transmission cost= ε
O_2/D_2
Source or Destination of the 2nd OD pair, transmission cost= ε
Uncompromised node, transmission cost= ε
Compromised node, transmission cost= M
Link on the path, transmission cost= ε

Figure 2- 8: DOS Legend (2)

The DOS value, denoted as S', is the average number of broken nodes of each OD pair, calculated as $S'=\frac{\sum(\text{No. of Broken Nodes of Each OD Pair})}{\text{No. of All OD Pairs of a Network}}$. That is, **the DOS value is the mean cost that each OD pair has to pay in order to communicate in an attacked network**.

The concept of DOS can be viewed as not only an "index of network damage" but also as a "metric of network survivability". For example, according to some network properties, the threshold of a network crash is established. Afterwards, should the network suffer from attack or failure, we can calculate the eventual average damage by the DOS concept. If the DOS value is larger than the pre-defined threshold, we say that the network is out of control. Therefore, **the greater the DOS value, the smaller the network survivability.** The DOS value represents not only the average damage level of all OD pairs in a network, but also the average cost we should pay to fully recover all OD pairs.

2.3 Lemma

The upper bound of the DOS value is subject to the network's topology. The DOS value must be related to |V| or |L|, where |V| and |L| are the numbers of nodes and links, respectively. We describe how to calculate the upper bound of the DOS value by the following two lemmas.

Lemma 1:

If the network's topology is simple and regular, for example, linear or ring, we can easily deduce the formulation of the upper bound of the DOS value, denoted as $\hat{S}$.

1. Linear topology:

$$\hat{S} = \frac{V+1}{3}, \quad \text{V : \# of nodes}$$

2. Ring topology:

$$\hat{S} = \frac{\left[\sum_{L=1}^{\left\lfloor \frac{V}{2} \right\rfloor - 1} L \times V + \left\lfloor \frac{V}{2} \right\rfloor \times \left(C_2^V - V\left(\left\lfloor \frac{V}{2} \right\rfloor - 1 \right) \right) \right]}{C_2^V}, \quad \text{V : \# of nodes, L : \# of links}$$

Lemma 2:

If the network's topology is irregular, i.e., an arbitrary undirected, connected graph, (for example, scale-free and random), we can make every OD pair run the shortest path algorithm to find its minimal hop count. The DOS value is the sum of the minimal hop count of each OD pair divided by |W|, i.e., the number of OD pairs. The entire procedure is numerical, so the time complexity is in polynomial time.

Chapter 3 PROBLEM FORMULATION

Network operators must decide how to allocate their limited defense budgets to prevent malicious attacks or intrusions (see **3.2 Model 2**). We simulate the role of an attacker who wants to compromise a targeted network at the minimal cost (see **3.1 Model 1**).

3.1 Model 1

3.1.1 Problem Description and Assumptions

To defend a network against attack, the network operator uses the concept of DOS described in the preceding chapter to calculate the network's survivability. We know the measurement of survivability can be categorized as: network performance, network connectivity, network capacity, traffic capacity, and so on. **The field of network survivability we discuss here focuses on network connectivity**.

In Model 1, we assume that the allocated defense budget for each node, *bi*, in the network is a given parameter. After an attack is launched, if node *i* should be compromised, the minimal attack cost will be $\hat{a}_i$. Naturally, the relationship between the defense cost *bi* and the attack cost $\hat{a}_i$ of node *i* is inter-dependent.

After an attack, the network operator can calculate the DOS value of the network. If the value is larger than a given parameter, *S*, the network operator can determine if the network is compromised and which nodes are the most vulnerable to attack.

Table 3- 1: Problem Description of Model 1

Given:

1. The network topology (for example, the network size, a set of OD pairs, and a set of candidate paths for each OD pair)
2. The damage threshold at which the network can be compromised
3. The cost of compromising a node is a function of the budget allocated to it.
4. The defense budget allocation policy

Objective:

An attacker is to minimize the total attack cost.

Subject to:

The average damage of the target network being greater than a given threshold, *S*

To determine:

Which nodes will be attacked

Table 3- 2: Problem Assumptions of Model 1

1. Any two nodes in the network can form an OD pair.
2. Both the attacker and the defender have complete information about the targeted network topology.
3. We consider static networks only. (We do not consider the growth of the network over time.)
4. We consider node attacks only. (Link attacks are not considered.) The transmission cost of an uncompromised node is ε, which is a small enough number; conversely, the cost of a compromised node is *M*, which is a large enough number.
5. We consider intelligent malicious attacks only. (Random errors are not considered.)
6. The defender's budget allocation policy is a given parameter. (In Model 1, we only consider the attacker's behavior, not that of the defenders.)

3.1.2 Mathematical Model

Table 3- 3: Given Parameters of Model 1

Notation	*Description*
V	The index set of all nodes in the network
W	The index set of all OD pairs in the network
P_w	The index set of all candidate paths of an OD pair *w,* where $w \in W$
M	A large enough number that indicates a node has been compromised
ε	A small enough number that indicates a node is functional
δ_{pi}	An indicator function, which is 1 if node *i* is on path *p,* and 0

	otherwise, where $i \in V$, $p \in P_w$
$\hat{a}_i$	The threshold of attack cost leading to a successful node attack
S	The threshold of a network crash, which is the average damage level of all OD pairs
R_w	The weight of OD pair w

Table 3- 4: Decision Variables of Model 1

Notation	*Description*
y_i	1 if node i is compromised, and 0 otherwise, where $i \in V$
t_{wi}	1 if node i is used by an OD pair w, and 0 otherwise, where $i \in V$, $w \in W$
x_p	1 if path p is chosen, and 0 otherwise, where $p \in P_w$
c_i	The transmission cost of node i, which is ε if node i is functional, and M if node i is broken, where $i \in V$

We formulate the problem as follows:

Objective function:

$$\min_{y_i} \sum_{i \in V} y_i \hat{a}_i, \qquad \text{(IP 1)}$$

Subject to:

$$c_i = y_i M + (1 - y_i)\varepsilon \qquad \forall i \in V \qquad \text{(IP 1.1)}$$

$$\sum_{i \in V} t_{wi} c_i \le \sum_{i \in V} \delta_{pi} c_i \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 1.2)}$$

$$\sum_{p\in P_w} x_p \delta_{pi} = t_{wi} \qquad \forall i \in V,\ w \in W \qquad \text{(IP 1.3)}$$

$$\frac{\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i}{|W| \times M} \geq S \qquad \text{(IP 1.4)}$$

$$\sum_{p\in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 1.5)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 1.6)}$$

$$y_i = 0 \text{ or } 1 \qquad \forall i \in V \qquad \text{(IP 1.7)}$$

$$t_{wi} = 0\ or\ 1 \qquad \forall i \in V,\ w \in W. \qquad \text{(IP 1.8)}$$

Explanation of the objective function:

(IP 1) is to minimize the total attack cost. That is, an attacker tries to minimize the objective value by deciding which node, denoted by y_i, should be compromised.

Explanation of the constraints:

(IP 1.1) describes the definition of the transmission cost of node *i*, which is ε if *i* is functional, and *M* if *i* is compromised.

(IP 1.2) requires that the selected path for an OD pair *w* should be the minimal cost path.

(IP 1.3) denotes the relationship between t_{wi} and $x_p\delta_{pi}$. To simplify the problem-solving procedure, we use the auxiliary set of decision variables t_{wi} to replace the sum of all $x_p\delta_{pi}$.

(IP 1.4) determines whether a target network has been compromised.

(IP 1.5) and (IP 1.6) jointly require that only one of the candidate paths of an OD pair w is selected.

(IP 1.7) determines whether node i is compromised.

(IP 1.8) determines whether node i is used to form the minimal cost path of an OD pair w.

3.1.3 Problem Reformulation

To solve Model 1 more efficiently, we reformulate the original problem. Note that the attack cost of node *i*, $\hat{a}_i$, is a function of the defense budget of node *i*, b_i (where $i \in V$); therefore, we replace $\hat{a}_i$ with $\hat{a}_i(b_i)$. Additionally, we adjust some constraints slightly without influencing the original problem structure and the optimal solution.

Therefore, Model 1 is reformulated as follows:

Objective function:

$$\min_{y_i} \sum_{i \in V} y_i \hat{a}_i(b_i), \qquad \text{(IP 2)}$$

Subject to:

$$c_i \le y_i M + (1 - y_i)\varepsilon \qquad \forall i \in V \qquad \text{(IP 2.1)}$$

$$\sum_{i \in V} t_{wi} c_i \le \sum_{i \in V} \delta_{pi} c_i \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 2.2)}$$

$$\sum_{p \in P_w} x_p \delta_{pi} \le t_{wi} \qquad \forall i \in V,\ w \in W \qquad \text{(IP 2.3)}$$

$$\frac{\sum_{w \in W} R_w \sum_{i \in V} t_{wi} c_i}{|W| \times M} \ge S \qquad \text{(IP 2.4)}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 2.5)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 2.6)}$$

$$y_i = 0 \text{ or } 1 \qquad \forall i \in V \qquad \text{(IP 2.7)}$$

$$t_{wi} = 0\ or\ 1 \qquad \forall i \in V,\ w \in W \qquad \text{(IP 2.8)}$$

$$c_i = M\ or\ \varepsilon \qquad \forall i \in V. \qquad \text{(IP 2.9)}$$

Explanation of the objective function:

(IP 2) is the adjusted result, according to the relationship between attack cost and defense cost.

Explanation of the constraints:

(IP 2.1) is the relaxed result of (IP 1.1). Note that the relaxation of the equation into an inequality does not violate its optimal solution.

(IP 2.3) is the relaxed result of (IP 1.3). To simply the problem-solving process, we slightly adjust this constraint from an equation into an inequality, i.e. from '=' into '≦', without violating its optimal solution.

(IP 2.9) is a redundant constraint, since the value of each c_i should be either ' ε ' or '*M*'. This constraint will be used in Lagrangean relaxation problem-solving.

3.2 Model 2

3.2.1 Problem Description and Assumptions

Due to the limited budget, how a network operator allocates the defense budget to each node of the network to maximize the defense is a significant issue. An intelligent and experienced attacker ceaselessly explores ways to minimize his/her attack costs. Naturally, a defender should continually try to maximize the attacker's costs by implementing an effective defense budget allocation policy.

For this model, a problem with a max-min structure is formulated in Model 2. As with Model 1, we use the concept of DOS to calculate the network's survivability. Recall that we focus on network connectivity.

The problem description and the assumptions of Model 2 are listed below: *(Italic font indicates that the point is different to Model 1)*

Table 3- 5: Problem Description of Model 2

Given:

1. The network topology (for example, the network size, a set of OD pairs, and a set of candidate paths for each OD pair)
2. The cost of compromising a node is a function of the budget allocated to it.
3. The damage threshold at which the network can be compromised

4. The total defense budget

Objective:

A defender is to maximize the attacker's minimal attack cost

Subject to:

1. The average damage to the target network being greater than a given threshold so that the network will be compromised.
2. *The total budget of the defender*

To determine:

1. Which nodes will be attacked by an attacker
2. *The defender's budget allocation policy*

Table 3- 6: Problem Assumptions of Model 2

1. Any two nodes in the network will form an OD pair.
2. Both the attacker and the defender have complete information about the targeted network topology.
3. We consider static networks only. (We do not consider the growth of a network over time.)
4. We consider node attacks only. (Link attacks are not considered.) The transmission cost of an uncompromised node is ε , which is a small enough number; conversely, the cost of a compromised node is *M*, which is a large enough number.
5. We consider intelligent malicious attacks only. (Random errors are not considered.)

3.2.2 Mathematical Model

The notations in Model 2 (Table 3-7 and Table 3-8) are partially the same as those in Model 1, except that the give parameter $\hat{a}_i$ in Model 1 becomes a decision variable $\hat{a}_i$ in Model 2; however, but it still denotes the cost of attacking node *i*. In addition, we add a new given parameter, *B*, to limit the defense budget.

Table 3- 7: Given Parameters of Model 2

Notation	*Description*
V	The index set of all nodes in the network
W	The index set of all OD pairs in the network
P_w	The index set of all candidate paths of an OD pair *w,* where $w\in W$
M	A large enough number that indicates a node has been compromised
ε	A small enough number that indicates a node is functional
δ_{pi}	An indicator function, which is 1 if node *i* is on path *p,* and 0 otherwise, where $i\in V$, $p\in P_w$
$\hat{a}_i$	The threshold of attack cost leading to a successful node attack
S	The threshold of a network crash, which is the average damage level of all OD pairs
R_w	The weight of OD pair *w*
B	*The total defense budget*

Table 3- 8: Decision Variables of Model 2

Notation	*Description*
y_i	1 if node *i* is compromised, and 0 otherwise, where $i\in V$
t_{wi}	1 if node *i* is used by an OD pair *w,* and 0 otherwise, where $i\in V$, $w\in W$
x_p	1 if path *p* is chosen, and 0 otherwise, where $p\in P_w$
c_i	The transmission cost of node *i*, which is *ε* if node *i* is functional, and *M* if node *i* is broken, where $i\in V$
b_i	*The budget allocated to node i*

We formulate the problem as follows:

Objective function:

$$\max_{\hat{a}_i} \min_{y_i} \sum_{i\in V} y_i \hat{a}_i, \qquad \text{(IP 3)}$$

Subject to:

$$c_i = y_i M + (1-y_i)\varepsilon \qquad \forall i\in V \qquad \text{(IP 3.1)}$$

$$\sum_{i\in V} t_{wi} c_i \le \sum_{i\in V} \delta_{pi} c_i \qquad \forall p\in P_w,\ w\in W \qquad \text{(IP 3.2)}$$

$$\sum_{p\in P_w} x_p \delta_{pi} = t_{wi} \qquad \forall i\in V,\ w\in W \qquad \text{(IP 3.3)}$$

$$\frac{\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i}{|W| \times M} \ge S \qquad \text{(IP 3.4)}$$

$$\sum_{p\in P_w} x_p = 1 \qquad \forall w\in W \qquad \text{(IP 3.5)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 3.6)}$$

$$y_i = 0 \text{ or } 1 \qquad \forall i \in V \qquad \text{(IP 3.7)}$$

$$t_{wi} = 0\ or\ 1 \qquad \forall i \in V, w \in W \qquad \text{(IP 3.8)}$$

$$\sum_{i \in V} b_i = B \qquad \text{(IP 3.9)}$$

$$0 \le b_i \le B \qquad \forall i \in V. \qquad \text{(IP 3.10)}$$

Explanation of the objective function:

(IP 3) is to maximize the attacker's minimal attack cost.

Explanation of the constraints:

(IP 3.1) defines the transmission cost of node *i*, which is *ε* if *i* is functional, and *M* if *i* is compromised.

(IP 3.2) requires that the selected path for an OD pair *w* should be the minimal cost path.

(IP 3.3) denotes the relationship between t_{wi} and $x_p\delta_{pi}$. We use the auxiliary set of decision variables t_{wi} to replace the sum of all $x_p\delta_{pi}$ in order to simplify the problem-solving procedure.

(IP 3.4) determines whether a target network has been compromised.

(IP 3.5) and (IP 3.6) jointly require that only one of the candidate paths of an OD pair *w* is selected.

(IP 3.7) determines whether node *i* has been compromised.

(IP 3.8) determines whether node *i* is used to form the minimal cost path of an OD pair *w*.

(IP 3.9) reflects that the optimality condition for the defender holds *if and only if* the total budget, *B*, is fully used.

(IP 3.10) requires that the set of decision variables, *bi*, is continuous and bounded by 0 and *B*.

3.2.3 Problem Reformulation

To solve Model 2 more efficiently, we reformulate the original problem. Note that the attack cost of node *i*, $\hat{a}_i$, is a function of the defense budget of node *i*, *bi* (where $i \in V$); therefore, we replace $\hat{a}_i$ with $\hat{a}_i(b_i)$. Additionally, some constraints can be slightly adjusted without affecting the original problem structure or the optimal solution.

Model 2 is reformulated as follows:

Objective function:

$$\max_{b_i} \min_{y_i} \sum_{i \in V} y_i \hat{a}_i(b_i), \qquad \text{(IP 4)}$$

Subject to:

$$c_i \le y_i M + (1 - y_i)\varepsilon \qquad \forall i \in V \qquad \text{(IP 4.1)}$$

$$\sum_{i \in V} t_{wi} c_i \le \sum_{i \in V} \delta_{pi} c_i \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 4.2)}$$

$$\sum_{p \in P_w} x_p \delta_{pi} \le t_{wi} \qquad \forall i \in V, w \in W \qquad \text{(IP 4.3)}$$

$$\frac{\sum_{w \in W} R_w \sum_{i \in V} t_{wi} c_i}{|W| \times M} \ge S \qquad \text{(IP 4.4)}$$

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(IP 4.5)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w,\ w \in W \qquad \text{(IP 4.6)}$$

$$y_i = 0 \text{ or } 1 \qquad \forall i \in V \qquad \text{(IP 4.7)}$$

$$t_{wi} = 0\ or\ 1 \qquad \forall i \in V, w \in W \qquad \text{(IP 4.8)}$$

$$\sum_{i \in V} b_i = B \qquad \text{(IP 4.9)}$$

$$0 \le b_i \le B \qquad \forall i \in V \qquad \text{(IP 4.10)}$$

$$c_i = M\ or\ \varepsilon \qquad \forall i \in V. \qquad \text{(IP 4.11)}$$

Explanation of the objective function:

(IP 4) is the adjusted result, according to the assumption of the relationship between the attack cost and the defense cost.

Explanation of the constraints:

(IP 4.1) is the relaxed result of (IP 3.1). Note that the relaxation of an equation into an inequality does not violate its optimal solution.

(IP 4.3) is the relaxed result of (IP 3.3). To simplify the problem solving process, we slightly adjust this constraint from an equation into an inequality, i.e. from '=' into '≦', without violating its optimal solution.

(IP 4.11) is a redundant constraint, since the value of each c_i should be either ' ε ' or

‘*M*’. This constraint will be used in Lagrangean relaxation problem-solving.

Chapter 4 SOLUTION APPROACH

4.1 Lagrangean Relaxation Method

In recent decades, Lagrangian Relaxation (LR) has evolved from a complete theoretical concept into a very practical tool that is the backbone of a number of large-scale applications. It is one of the few problem-solving methods for optimization that cuts across the domains of linear and integer programming, combinatorial optimization, and non-linear programming [10].

When we encounter great difficulty in solving large-scale and complicated problems, which are usually solved in *exponential time* and are normally formulated as optimization objective functions with a set of side constraints, the LR method is regarded as an effective problem-solving tool. It makes use of a “decomposition” or “divide-and-conquer” technique to make the original problem easier to solve. Such problems are usually solved in *polynomial time.*

LR is based on the observation that many difficult integer programming problems can be modeled as relatively easy problems complicated by a set of side constraints [12]. Next, we explain how LR simplifies complicated problems:

1) To simplify a problem, we can relax one or more complicated constraints and combine them with the associated Lagrangian multipliers "*u*" into the primal objective function. Therefore, the primal problem (P) is transformed into a Lagrangian relaxation problem (LR). The value of the objective function of the primal problem, Z, is also transformed into the LR problem, $Z_D(u)$.

2) Based on the different decision variables, we can decompose, or "divide", the Lagrangian relaxation problem (LR) into several easily-solvable and stand-alone smaller problems, called "sub-problems".

3) Each sub-problem can be optimally solved, or "overcome" by well-known algorithms, such as Dijkstra's algorithm.

4) Aggregating the optimal solutions of all the sub-problems, we obtain the optimal objective function value, $Z_D(u)$.

However, the optimal objective function value, $Z_D(u)$, obtained by relaxing the constraints may be not a feasible solution of the primal problem. The LR problem may provide a lower bound (LB, for a minimization problem) of the optimal value of the original problem. Although the LB may be not a feasible solution, fortunately, we can obtain some valuable hints from the Lagrangian multipliers "*u*" to explore a real feasible solution. That is, we can raise the LB to the feasible solution area by constantly adjusting the Lagrangian multipliers "*u*". The procedure, called the "subgradient method", is illustrated in Figure 4-2.

In addition, we can obtain an upper bound (UB, for a minimization problem) of the primal problem heuristically. This UB can be regarded as a limitation of the LB. Then, the area constructed by LB and UB, called the "gap", is the feasible solution area. The optimal objective function value of the primal problem will fall in this gap. Therefore, to obtain the tightest gap, we must search the best algorithm for each sub-problem and the best heuristic for the original problem.

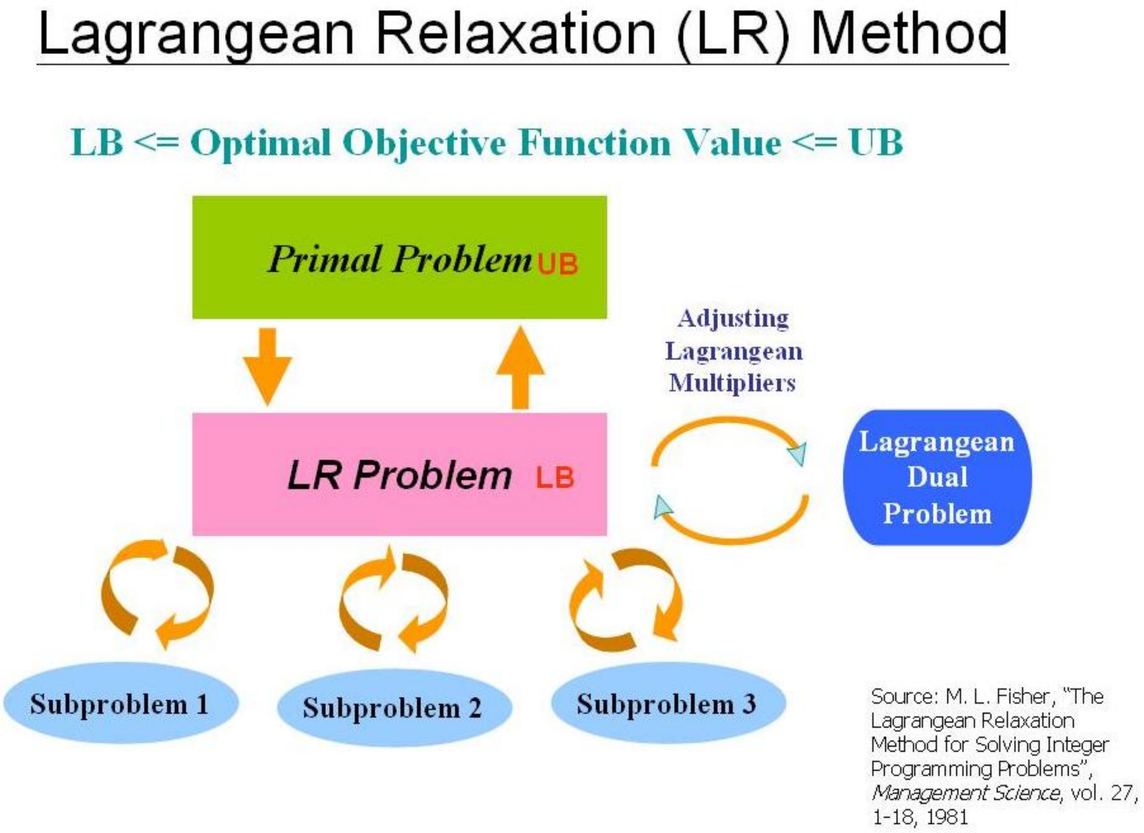
The complete procedure of LR is shown in Figure 4-1. The advantages of the LR solution strategy are as follows:

1) It is a comparatively quick and flexible problem-solving method.

2) It can develop a comparatively tight gap on the optimal value of the objective function.

3) Each decomposed sub-problem is regarded as stand-alone model.

4) It can provide hints about devising effective heuristic solution methods for solving complex combinatorial optimization problems and integer programs [11].

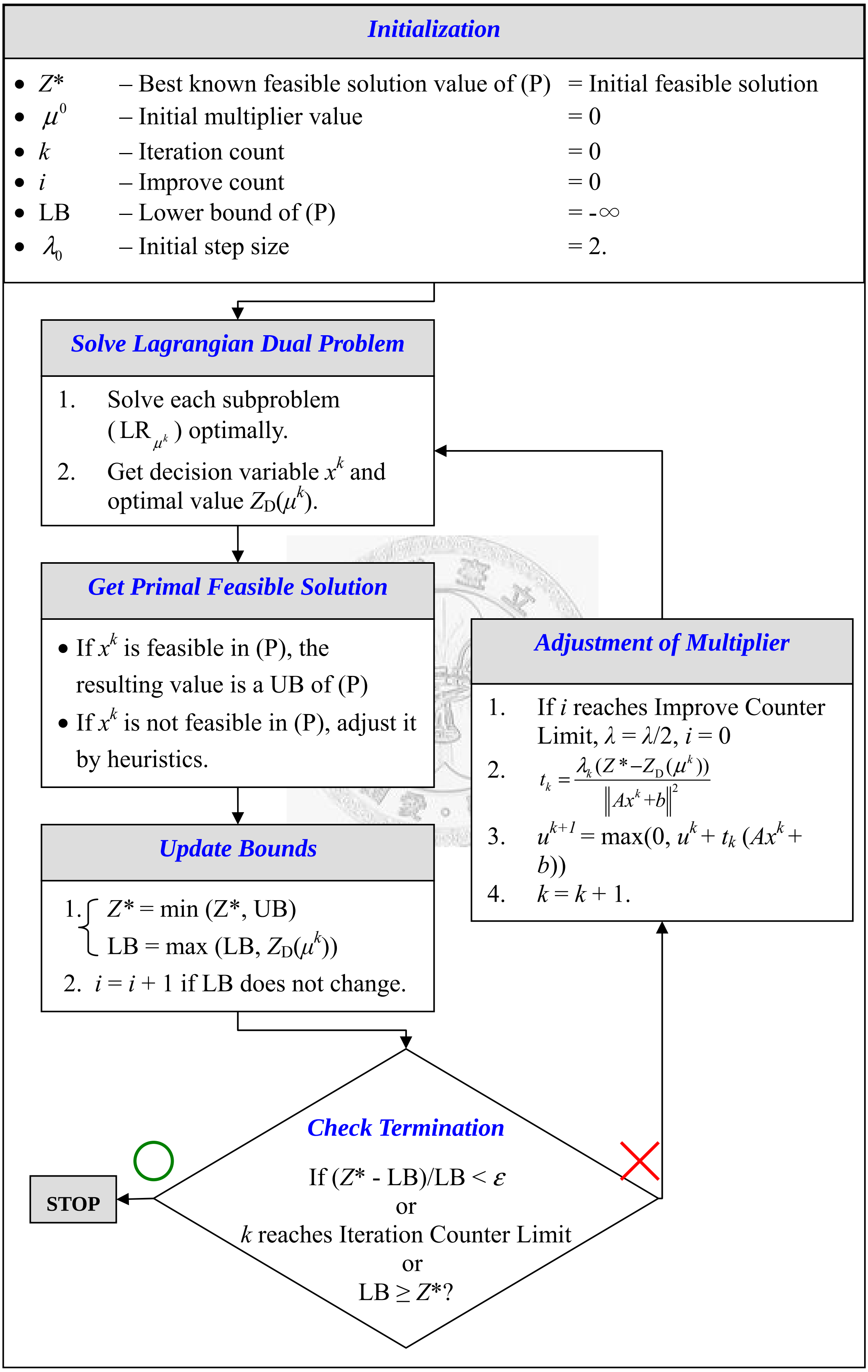


Lagrangean Relaxation (LR) Method

LB <= Optimal Objective Function Value <= UB

Source: M. L. Fisher, "The Lagrangean Relaxation Method for Solving Integer Programming Problems", *Management Science*, vol. 27, 1-18, 1981

Figure 4- 1: Lagrangean Relaxation Method [12]



- Z^* – Best known feasible solution value of (P) = Initial feasible solution
- μ^0 – Initial multiplier value = 0
- k – Iteration count = 0
- i – Improve count = 0
- LB – Lower bound of (P) = $-\infty$
- λ_0 – Initial step size = 2.

1. Solve each subproblem (LR_{μ^k}) optimally.
2. Get decision variable x^k and optimal value $Z_D(\mu^k)$.

- If x^k is feasible in (P), the resulting value is a UB of (P)
- If x^k is not feasible in (P), adjust it by heuristics.

1. $\begin{cases} Z^* = \min(Z^*, \mathrm{UB}) \\ \mathrm{LB} = \max(\mathrm{LB}, Z_D(\mu^k)) \end{cases}$
2. $i = i + 1$ if LB does not change.

1. If i reaches Improve Counter Limit, $\lambda = \lambda/2$, $i = 0$
2. $t_k = \dfrac{\lambda_k (Z^* - Z_D(\mu^k))}{\|Ax^k + b\|^2}$
3. $u^{k+1} = \max(0, u^k + t_k (Ax^k + b))$
4. $k = k + 1$.

If $(Z^* - \mathrm{LB})/\mathrm{LB} < \varepsilon$
or
k reaches Iteration Counter Limit
or
$\mathrm{LB} \geq Z^*$?

Figure 4- 2: Flow Chart for Lagrangean Relaxation Method

4.2 Solution Approach for Model 1

4.2.1 Lagrangean Relaxation

In Model 1, we use LR to solve the problem. We relax Constraints (IP 2.1), (IP 2.2), (IP 2.3), and (IP 2.4) and combine them, respectively, with the associated Lagrangean multipliers, u^1, u^2, u^3, and u^4 of primal problem (P) to obtain the Lagrangean relaxation problem (LR) as follows:

Optimization problem:

$$
\begin{aligned}
& Z_D(u^1,u^2,u^3,u^4) \\
& = \min_{y_i} \sum_{i\in V} y_i \hat{a}_i(b_i) \\
& + \sum_{i\in V} u_i^1 [c_i - (y_i M + (1-y_i)\varepsilon)] \\
& + \sum_{w\in W} \sum_{p\in P_w} u_{wp}^2 \left[\sum_{i\in V} t_{wi} c_i - \sum_{i\in V} \delta_{pi} c_i \right] \\
& + \sum_{w\in W} \sum_{i\in V} u_{wi}^3 [(\sum_{p\in P_w} x_p \delta_{pi}) - t_{wi}] \\
& + u^4 [S\times |W| \times M - \sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i],
\end{aligned}
\qquad \text{(LR)}
$$

Subject to:

$$\sum_{p\in P_w} x_p = 1 \qquad \forall w\in W \qquad \text{(LR 1)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p\in P_w,\ w\in W \qquad \text{(LR 2)}$$

$y_i = 0$ or 1 $\quad \forall i \in V \quad$ (LR 3)

$t_{wi} = 0$ or 1 $\quad \forall i \in V, w \in W \quad$ (LR 4)

$c_i = M$ or ε $\quad \forall i \in V. \quad$ (LR 5)

Constraints (IP 2.5) ~ (IP 2.9) of the objective function (IP 2) of Model 1 are not relaxed, but denoted as (LR 1) ~ (LR 5) in the LR problem. The Lagrangean multipliers u^1, u^2, u^3, and u^4 are, respectively, the vectors of $\{u_i^1\}$, $\{u_{wp}^2\}$, $\{u_{wi}^3\}$, and $\{u^4\}$. According to the procedure mentioned above, we decompose the dual problem into three sub-problems, which we describe below.

4.2.1.1 Subproblem 1 (related to decision variable x_p)

Subproblem 1:

$$Z_{Sub1}(u^3) = \min \sum_{w \in W} \sum_{i \in V} \sum_{p \in P_w} u_{wi}^3 \delta_{pi} x_p, \qquad \text{(Sub 1)}$$

Subject to:

$$\sum_{p \in P_w} x_p = 1 \qquad \forall w \in W \qquad \text{(Sub 1.1)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in p_w, w \in W. \qquad \text{(Sub 1.2)}$$

Subproblem 1 is decomposed into |W| smaller problems of the independent shortest cost path. We can individually determine the value of x_p for each OD pair, and u_{wi}^3 is assigned as the cost of node i in OD pair w. We adopt Dijkstra's algorithm to find the value of x_p for each |W| OD pair. The time complexity of Dijkstra's algorithm is O(|V|^2), where |V| is the number of nodes; therefore, the time complexity of subproblem 1 is O(|W|×|V|^2).

4.2.1.2 Subproblem 2 (related to decision variable y_i)

Subproblem 2:

$$\begin{aligned} &Z_{Sub2}(u^1) \\ &= \min \sum_{i \in V} y_i \hat{a}_i(b_i) + \sum_{i \in V} u_i^1 y_i \varepsilon + \sum_{i \in V} u_i^1 y_i(-M) \\ &= \min \sum_{i \in V} \left[\hat{a}_i(b_i) + u_i^1 \varepsilon + u_i^1(-M)\right] y_i, \end{aligned} \qquad \text{(Sub 2)}$$

Subject to:

$y_i = 0$ or 1 $\qquad \forall i \in V.$ (Sub 2.1)

Subproblem 2 can also be decomposed into |V| smaller problems. We can determine the value of decision variable y_i by its coefficient $\left[\hat{a}_i(b_i) + u_i^1 \varepsilon + u_i^1(-M)\right]$. If the coefficient is positive, y_i is set as zero; otherwise it is set to one. The time complexity of Subproblem 2 is O(|V|).

4.2.1.3 Subproblem 3 (related to decision variable t_{wi}, c_i)

Subproblem 3:

$$
\begin{aligned}
&Z_{Sub3}(u^1,u^2,u^3,u^4) \\
&= \min \sum_{i\in V} u_i^1 c_i + \sum_{w\in W}\sum_{p\in P_w} u_{wp}^2 \sum_{i\in V} t_{wi} c_i + \sum_{w\in W}\sum_{p\in P_w} u_{wp}^2 \sum_{i\in V}(-\delta_{pi} c_i) \\
&+ \sum_{w\in W}\sum_{i\in V} u_{wi}^3(-t_{wi}) + u^4(-\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i) \\
&= \min \sum_{i\in V} \left\{ \left[u_i^1 - \sum_{w\in W}\sum_{p\in P_w} u_{wp}^2 \delta_{pi} + \sum_{w\in W}\left((\sum_{p\in P_w} u_{wp}^2) - u^4 R_w \right) t_{wi} \right] c_i - \sum_{w\in W} u_{wi}^3 t_{wi} \right\},
\end{aligned}
\qquad \text{(Sub 3)}
$$

Subject to:

$$t_{wi} = 0 \; or \; 1 \qquad \forall i \in V, w \in W \qquad \text{(Sub 3.1)}$$

$$c_i = M \; or \; \varepsilon \qquad \forall i \in V. \qquad \text{(Sub 3.2)}$$

Subproblem 3 is decomposed into |V| smaller problems. However, according to Constraints (Sub 3.1) and (Sub 3.2), decision variables t_{wi} and c_i have two choices, depending on which combination results in the smallest value derived by the exhaustive search. The time complexity is O(|V|x|W|).

4.2.2 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem, for any set of multipliers, $(u^1, u^2, u^3, u^4) \geqq 0$, $Z_D(u^1, u^2, u^3, u^4)$ is a lower bound on Z_{IP2}. We explore the maximal lower bound to tighten the gap between the upper bound and the lower bound; that is, the area of feasible solutions. Thus, the dual problem is constructed as follows:

Dual Problem:

$$Z_D = \max Z_D(u^1, u^2, u^3, u^4), \qquad \text{(D)}$$

Subject to:

$(u^1, u^2, u^3, u^4) \geqq 0.$

Although there are many ways to solve dual problem (D), we adopt the subgradient method because it is simple and intuitive. Table 4-1 defines the notations used to solve the dual problem (D).

Table 4- 1: Notations for Subgradient Method

Notation	*Description*
k	A vector, which is a subgradient of $Z_D(u^1, u^2, u^3, u^4)$
p	The iteration counter of the subgradient procedure

t^p	The step size
z_{IP}^*	The best upper bound on the primal objective function value after the p^{th} iteration.
λ	A value between 0 and 2, which is initially 2 and halved whenever the best objective function value does not improve within a given iteration count

In iteration p of the multiplier vector, $u = (u^1, u^2, u^3, u^4)$ is updated by

$$u^{p+1} = u^p + t^p k^p ,$$

where the step size t^p is calculated by

$$t^p = \lambda \frac{z_{IP}^* - Z_D(u^p)}{\left\| k^p \right\|^2} .$$

4.2.3 Getting Primal Feasible Solutions

To get an optimal heuristic, we partition the problem-solving procedure into three stages, called 3-Stage Selection (3SS). Assume that there are two buckets, one is for the nodes not yet attacked, called the Safety-Bucket, and the other is for the nodes already attacked, called the Attacked-Bucket. **Stage 1 describes how we select nodes from the Safety-Bucket to transfer to the Attacked-Bucket**. Recall that Subproblem 1 is used to solve decision variable x_p $(p\in P_w)$, which determines the path to be chosen for an OD pair. With this hint from Subproblem 1, we can count the number of times a node is traversed by all paths and then decide the attack order. That is, the more times a node is traversed, the more likely it will be attacked. Each time we transfer a node from the Safety-Bucket to the Attacked-Bucket, we must rerun the Dijkstra's algorithm to calculate the total transmission cost (TC). The time complexity of Stage 1 is O(|V|^3), where |V| is the number of nodes. The algorithm for Stage 1 is detailed in Table 4-2.

Table 4- 2: Algorithm for Stage 1 of 3SS

```
//Initiate all parameters for the stop condition, (IP2.4)
TC (Transmission Cost)= $\varepsilon$, (M, if attacked.);
SumOfRTC (the result of $\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i$)= 0;
Threshold (a given parameter)= (S×|W|×M);

WHILE (SumOfRTC < Threshold)
{
    Find “ToBeIn”, the node that has been traversed the most times by all paths,
    called the “Popular Node (PN)”;
    Change PN’s TC as M;
    Apply Dijkstra’s algorithm to calculate the SumOfRTC value;

    IF (every node in the Safety-Bucket is chosen)
    {
        break;
    }
    else
    {
        Find the next “ToBeIn” from the Safety-Bucket.
    }
}//end of while
```

Stage 2 of 3SS is used to adjust the nodes transferred to the Attacked-Bucket from the Safety-Bucket in Stage 1. In Stage 1, we chose some nodes to attack based on their popularity. However, we can not guarantee that the result will be optimal because the most popular node may not have the smallest attack cost. To get optimal solution, we adjust the result of Stage 1 by comparing the node with the largest attack cost in the Attacked-Bucket with the most popular node (PN) in the Safety-Bucket. By repeating this procedure, we can derive the value of the minimal objective function, i.e., the minimal attack cost. The time complexity of Stage 2 is O(|V|^4).

Table 4- 3: Algorithm for Stage 2 of 3SS

```
WHILE (any node in the Safety-Bucket is not checked)
{
    In the Attacked-Bucket, find "ToBeOut", the node with the largest attack cost (AC);
    In the Safety-Bucket, find "ToBeIn", the most popular node (PN);
    Use Dijkstra's algorithm to calculate new SumOfRTC value;

    IF (SumOfRTC > Threshold)
    {
        IF (AC [ToBeIn] < AC [ToBeOut]) // which has the smaller AC?
        {
            Choose "ToBeIn" into the Attacked-Bucket;
            Discard "ToBeOut";
        }
        ELSE
        {
            Keep "ToBeOut" in the Attacked-Bucket;
            Keep "ToBeIn" in the Safety-Bucket;
        }
    }
    ELSE
    {
        Keep "ToBeIn" in the Safety-Bucket;
        Find the next "ToBeIn" from the Safety-Bucket;
    }
}//end of while
```

In the previous two stages, we only required that the $\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i$ value of the chosen nodes should be larger than the threshold, ($S\times|W|\times M$). However, in the last stage, we want to make sure the $\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i$ value is greater than and the closest to threshold. So we try to discard the least popular node in the Attacked-Bucket and see if the new $\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i$ value is still greater than the threshold. If it is, we keep discarding the next node until the $\sum_{w \in W} R_w \sum_{i \in V} t_{wi} c_i$ value is smaller than the threshold. The time complexity of Stage 3 is $O(|V|^3)$. Table 4-4 details this stage.

Table 4- 4: Algorithm for Stage 3 of 3SS

```
Find “ToBeOut”, the least popular node, in the Attacked-Bucket;
Use Dijkstra’s algorithm to calculate the new SumOfRTC value;

WHILE (SumOfRTC > Threshold)
{
    Discard “ToBeOut” from the Attacked-Bucket;
    Find “ToBeOut”, the least popular node, from the Attacked-Bucket;
    Use Dijkstra’s algorithm to calculate new SumOfRTC value;
}//end of while
```

In summary, the time complexity of whole heuristic is $O(|V|^4)$. The flow chart of the procedure for getting the primal feasible solution is shown below. Figures 4-3, 4-4, and 4-5 are, respectively, the flow charts of Stage 1, 2, and 3 of 3SS algorithm.

Figure 4- 3: Flow Chart for Stage 1 of 3SS

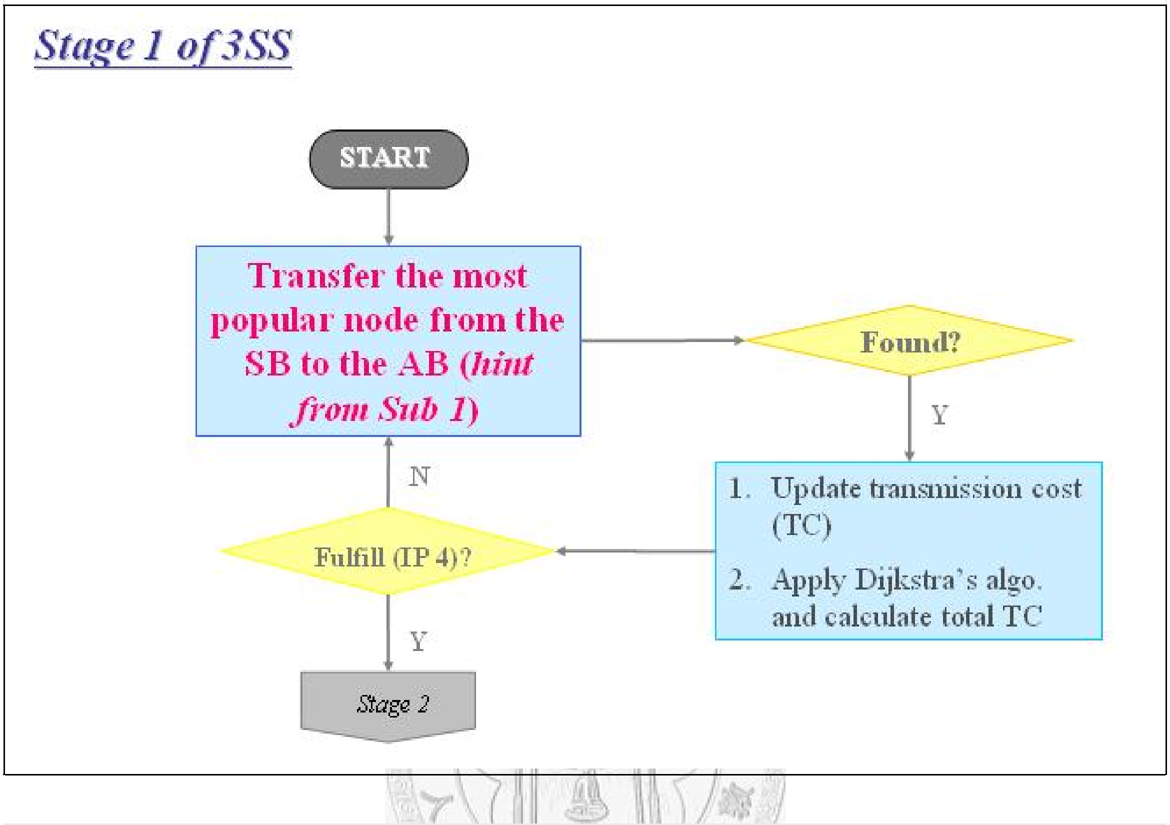

Stage 1 of 3SS
START
Transfer the most popular node from the SB to the AB (hint from Sub 1)
Found?
Y
1. Update transmission cost (TC)
2. Apply Dijkstra's algo. and calculate total TC
Fulfill (IP 4)?
N
Y
Stage 2

Figure 4- 4: Flow Chart for Stage 2 of 3SS

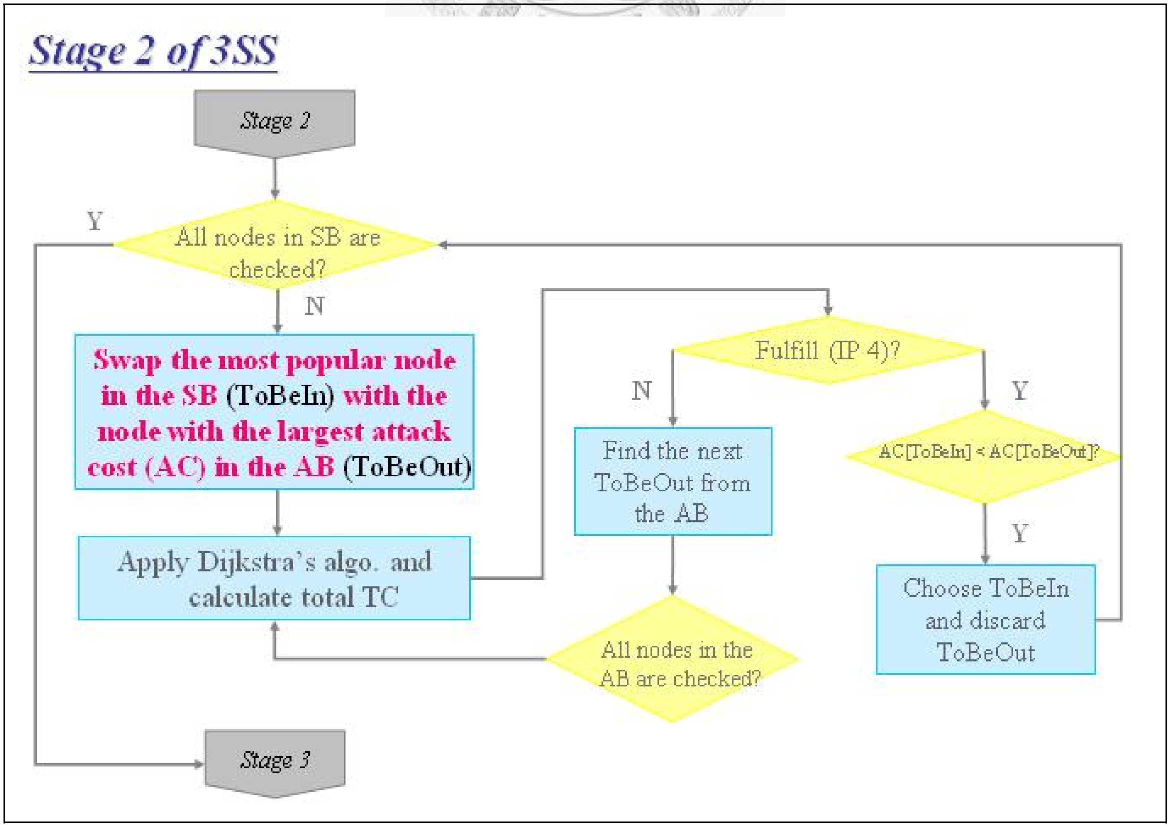
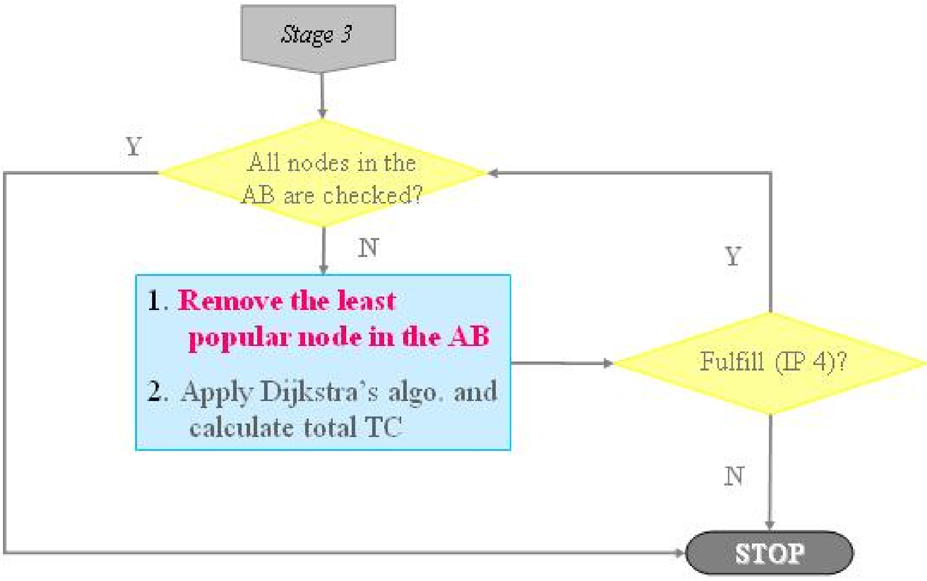

Stage 2 of 3SS
Stage 2
All nodes in SB are checked?
Y
N
Swap the most popular node in the SB (ToBeIn) with the node with the largest attack cost (AC) in the AB (ToBeOut)
Apply Dijkstra's algo. and calculate total TC
Fulfill (IP 4)?
N
Y
Find the next ToBeOut from the AB
AC[ToBeIn] < AC[ToBeOut]?
Y
Choose ToBeIn and discard ToBeOut
All nodes in the AB are checked?
Stage 3

Figure 4- 5: Flow Chart for Stage 3 of 3SS

Stage 3 of 3SS


Stage 3
All nodes in the AB are checked?
Y
N
1. Remove the least popular node in the AB
2. Apply Dijkstra's algo. and calculate total TC
Fulfill (IP 4)?
Y
N
STOP

Chapter 5 COMPUTATIONAL EXPERIMENTS

5.1 Simple Algorithm

In the previous chapter, by using the Lagrangean Relaxation (LR) method and the Subgradient method, we not only derived a lower bound (LB) of the primal objective function value, we also obtained good hints for getting primal feasible solutions.

The algorithm derived from those hints and proposed in Section **4.2.3 Getting Primal Feasible Solutions** is called 3SS. The objective function value derived from 3SS is regarded as an upper bound (UB) of the primal objective function (IP 2).

To prove our 3SS is effective, we propose two simple algorithms, namely, “**5.1.1 DAA**” and “**5.1.2 PAA**”, and compare their performances with that of 3SS.

5.1.1 Degree-based Attack Algorithm (DAA)

Recall that in Section **3.1.3 Problem Reformulation**, the stop condition of the whole experiment on Model 1 is confined to ***Constraint (IP2.4)*** $\frac{\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i}{|W| \times M} \geq S$, and we let the result of the ***Objective Function (IP 2)*** $\min_{y_i} \sum_{i\in V} y_i \hat{a}_i(b_i)$ be minimal. Intuitively, our attack order is from the node with the largest number of degrees, because we assume that it will help us achieve the requirement of ***Constraint (IP2.4)*** more quickly. The time complexity of DAA is O(|V|^4).

Table 5- 1: Degree-based Attack Algorithm (DAA)

```
getDAA()
{
    //Set initial values of all parameters
    TC (Transmission Cost) = $\varepsilon$, (M, if attacked);
    SumOfRTC (the result of $\sum_{w\in W} R_w \sum_{i\in V} t_{wi} c_i$) = 0;
    Threshold = (S*|W|*M);

    WHILE (SumOfRTC < Threshold)
    {
        Transfer "ToBeIn", the node with the largest number of degrees (DN), from
      the Safety-Bucket to the Attacked-Bucket;
        Set its TC as M ;
        Apply Dijkstra's algorithm to recalculate the SumOfRTC value;
        IF (every node in the Safety-Bucket is chosen)
        {
            break;
        }
        ELSE
        {
            Transfer the next “ToBeIn” from the Safety-Bucket to the Attacked-Bucket;
        }
    }
}//end of getDAA()
```

5.1.2 Popularity-based Attack Algorithm (PAA)

To further prove of the performance of 3SS, the second simple algorithm is extracted from the Stage 1 of 3SS. With PAA, we can observe more clearly the effectiveness of the Stage 2 and 3 of 3SS. We define “popularity” as the number times that a node is traversed by all OD pairs in a network. The time complexity of PAA is O(|V|^4).

Table 5- 2: Popularity-based Attack Algorithm (PAA)

```
getPAA()
{
    //Set initial values of all parameters
    TC = $\varepsilon$, (M, if attacked);
    SumOfRTC = 0;
    Threshold = (S*|W|*M);

    WHILE (SumOfRTC < Threshold)
    {
        Transfer “ToBeIn”, the most popular node, from the Safety-Bucket to the Attacked-Bucket;
        Set its TC as M;
        Apply Dijkstra’s algorithm to recalculate the SumOfRTC value;
        IF (every node in the Safety-Bucket is chosen)
        {
            break;
        }
        ELSE
        {
            Transfer the next “ToBeIn” from the Safety-Bucket to the Attacked-Bucket;
            }
        }
}//end of getPAA()
```

5.2 Experiment Environment

All the proposed algorithms are implemented in Dev-C++ IDE and run on a PC with an Intel(R) Pentium(R) 4 CPU 3.00GHz & 2.99GHz, 504 MB RAM, and MW XP Professional V2002 SP2.

Three kinds of the network topologies **(NT)** are simulated as attacked targets, namely, a grid network **(GD)**, a scale-free network **(SF)**, and a random network **(RD)**. The GD is a relatively regular network; the SF features each newly added node connected to the nodes with the largest number of degrees; and the RD features each newly node connected to arbitrary nodes. For comparison, the total number of degrees of a RD is the same as that of a SF.

There are two initial budget allocation policies (**BAP):** uniform (**Uni**) and degree-based (**Deg**).

However, to observe the effect of the budget re-allocation policy (**BRAP**), we also propose three mechanisms. The first is the uniform (**Uni**), where the budget uniformly extracted from un-attacked nodes will be uniformly distributed to attacked nodes. The second is degree-based (**Deg**), where the higher the degree of an un-attacked node, the lower the budget extracted from it. The extraction percentage (**EP**), the proportion of budget extracted from each un-attacked node, is formulated as

$$\mathbf{EP} = (\mathbf{1} - \frac{\textbf{Degree No. of Unattacked Node}_{\mathbf{i}}}{\textbf{Total Degree No. of Unattacked Nodes}}) \times \frac{\mathbf{1}}{\textbf{No. of Unattacked Nodes}}.$$

Conversely, the higher the degree of an attacked node, the greater the budget distributed to it will be. The distribution percentage (**PD**), the proportion of extra budget distributed to each attacked node, is formulated as

$$\mathbf{DP} = \frac{\textbf{Degree No. of Attakced Node}_{\mathbf{i}}}{\textbf{Total Degree No. of Attacked Nodes}}.$$

The third mechanism is popularity-based (**Pop**). The higher the popularity of a un-attack node, the lower the budget extracted from it. The EP is formulated as

$$\mathbf{EP} = (\mathbf{1} - \frac{\textbf{Popularity No. of Unattacked Node}_{\mathbf{i}}}{\textbf{Total Popularity No. Of Unattacked Nodes}}) \times \frac{\mathbf{1}}{\textbf{No. of Unattacked Nodes}}.$$

Conversely, the higher the popularity of an attacked node, the greater the budget distributed to it will be. The DP is formulated as

$$\mathbf{DP} = \frac{\textbf{Popularity No. of Attakced Node}_{\mathbf{i}}}{\textbf{Total Popularity No. Of Attacked Nodes}}.$$

Because of time considerations, we only select two kinds BRAP to implement: Uni and Deg. Additionally, we set the attack cost of each node as a function of its defense budget, denoted as $\hat{a}_i(b_i)$. The step size scalar, λ, is initialized as 2, and is halved if the objective function value, Z_D, does not improve after performing iterations up to the Max Improvement Count. The Percentages of Damage (**PD**) are different damage levels of the network leading to different network crashes.

More parameters are shown below:

Table 5- 3: Experiment Parameter Settings

Parameters	***Value***
Test Platform	1. CPU: Pentium(R) 3.0GHz & 2.99GHz 2. RAM: 504MB 3. OS: MS Win XP Professional 2002
Network Topology (NT)	1. Grid (GD) 2. Scale-free (SF) 3. Random (RD)
Budget Allocation Policy (BAP)	1. Uniform (Uni) 2. Degree-based (Deg)
Budget Re-allocation Policy (BRAP)	1. Uniform (Uni) 2. Degree-based (Deg)
Total Defense Budget (B)	|V|, (No. of nodes)
Attack Cost Function	$\hat{a}_i(b_i) = b_i + m$, (*m*: a constant)
Initial Scalar of Step Size (λ)	2.0
Initial Transmission Cost (ε)	0.1
Transmission Cost after Attack (M)	|V|×ε
Max Outer Counter	500
Max Inner Counter	5000
Max Improvement Counter	1000
Percentages of Damage (PD)	80%, 60%, 40%, 20%
No. of OD pairs (|W|)	72, 240, 600, 1260

5.3 Experiment Results and Discussion

We present all experiment results in tabular form. The notations used in the tables of the experiment results are defined as follows: ***APL*** denotes the Average Path Length; ***S***, calculated by (APL×PD), is the damage threshold at which a network can be compromised; ***ZIP*** is the upper bound of the objective function value and calculated by 3SS attack algorithm; ***LB*** is the lower bound of the objective function value and calculated by the LR problem; ***Gap*** is calculated by $(\frac{ZIP-LB}{LB}\times 100\%)$. ***ZDAA*** and ***ZPAA*** are the results of DAA and PAA, respectively; and ***ImpR.D*** and ***ImpR.P***, which denote the Improvement Rate of DAA and PAA, respectively, are calculated by $(\frac{ZDAA-ZIP}{ZIP}\times 100\%)$ and $(\frac{ZPAA-ZIP}{ZIP}\times 100\%)$.

5.3.1 Experiment Results of Model 1

The purpose of this experiment is to compare the performance of 3SS with that of two other simple algorithms, DAA and UAA, under two budget allocation policies (BAP), Uni and Deg.

Case 1: Extra-small-scale Networks (|W|= 72)

Table 5- 4: Experiment Results of Extra-small-scale Networks for Model 1

APL = 3	**NT = 3×3 GD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.60	1.20	1.80	2.40
	ZIP	2.0000	4.0000	5.0000	7.0000
	ZDAA	2.2500	4.2750	5.4000	7.0714
	ImpR.D	12.50	6.88	8.00	1.02
	ZPAA	2.2500	4.0500	6.1200	8.1000
	ImpR.P	12.50	1.25	22.40	15.71
	Gap	6.12	5.80	3.18	1.10
BAP = Deg	PD	20%	40%	60%	80%
	S	0.60	1.20	1.80	2.40
	ZIP	1.5000	3.7500	5.2500	7.5000
	ZDAA	2.3625	4.6312	6.1500	7.5536
	ImpR.D	57.50	23.50	17.14	0.71
	ZPAA	2.1375	4.2750	6.4500	8.3250
	ImpR.P	42.50	14.00	22.86	11.00
	Gap	6.95	6.00	3.29	1.12
APL = 2. 61	**NT = SF**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.52	1.04	1.57	2.09
	ZIP	2.0000	3.0000	4.0000	7.0000

	ZDAA	2.7000	3.6000	5.4000	7.2000
	ImpR.D	35.00	20.00	35.00	2.86
	ZPAA	2.2500	4.5000	5.1750	7.2000
	ImpR.P	12.50	50.00	29.38	2.86
	Gap	6.35	5.10	2.99	2.10
BAP = Deg	PD	20%	40%	60%	80%
	S	0.52	1.04	1.57	2.09
	ZIP	1.2000	2.4000	5.1000	7.2000
	ZDAA	3.2400	4.3200	6.3400	7.8343
	ImpR.D	170.00	80.00	24.31	8.81
	ZPAA	2.4300	5.1300	6.1950	7.8343
	ImpR.P	102.50	113.75	21.47	8.81
	Gap	4.89	4.10	2.00	1.03
APL = 2.61	**NT = RD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.52	1.04	1.57	2.09
	ZIP	2.0000	3.0000	5.0000	7.0000
	ZDAA	2.1000	4.2000	6.1200	7.9714
	ImpR.D	5.00	40.00	22.40	13.88
	ZPAA	2.7000	3.9000	6.8400	7.9714
	ImpR.P	35.00	30.00	36.80	13.88
	Gap	2.98	2.40	1.11	0.13
BAP = Deg	PD	20%	40%	60%	80%
	S	0.52	1.04	1.57	2.09
	ZIP	1.8000	3.3000	5.1000	7.2000
	ZDAA	2.4300	4.6700	6.4020	8.0743
	ImpR.D	35.00	41.52	25.53	12.14
	ZPAA	2.5200	3.5200	6.3420	7.4743
	ImpR.P	40.00	6.67	24.35	3.81
	Gap	2.77	1.76	0.78	0.01

Case 2: Small-scale Networks (|W|= 240)

Table 5- 5: Experiment Results of Small-scale Networks for Model 1

APL = 3.67	**NT = 4×4 GD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.73	1.47	2.20	2.93
	ZIP	4.0000	7.0000	10.0000	13.0000
	ZDAA	5.3000	8.1143	10.2800	13.1769
	ImpR.D	32.50	15.92	2.80	1.36
	ZPAA	5.3000	7.8857	11.3400	14.0769
	ImpR.P	32.50	12.65	13.40	8.28
	Gap	19.88	15.28	9.44	5.26
BAP = Deg	PD	20%	40%	60%	80%
	S	0.73	1.47	2.20	2.93
	ZIP	3.3333	6.6667	10.0000	13.3333
	ZDAA	5.7000	9.1333	11.2267	14.1128
	ImpR.D	71.00	37.00	12.27	5.85
	ZPAA	5.0833	8.3333	11.5267	14.3795
	ImpR.P	52.50	25.00	15.27	7.85
	Gap	18.95	12.22	8.00	4.29
APL = 2.94	**NT = SF**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.59	1.18	1.77	2.35
	ZIP	2.0000	4.0000	8.0000	12.0000
	ZDAA	3.4000	5.7000	8.6000	12.2667
	ImpR.D	70.00	42.50	7.50	2.22
	ZPAA	5.2000	5.3000	9.3000	12.2667
	ImpR.P	160.00	32.50	16.25	2.22
	Gap	35.11	28.40	11.66	7.39
BAP = Deg	PD	20%	40%	60%	80%
	S	0.59	1.18	1.77	2.35
	ZIP	2.2069	6.8966	9.6552	12.9655
	ZDAA	5.7172	8.6759	12.0138	13.7724
	ImpR.D	159.06	25.80	24.43	6.22
	ZPAA	6.7034	8.1793	11.3000	13.7724

	ImpR.P	203.75	18.60	17.04	6.22
	Gap	40.71	30.34	15.25	6.79
APL = 3.1	**NT = RD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.62	1.24	1.86	2.48
	ZIP	3.0000	5.0000	8.0000	12.0000
	ZDAA	4.3000	6.3600	9.3000	12.1333
	ImpR.D	43.33	27.20	16.25	1.11
	ZPAA	3.6000	8.1600	11.1000	15.1000
	ImpR.P	20.00	63.20	38.75	25.83
	Gap	20.73	13.81	7.99	3.18
BAP = Deg	PD	20%	40%	60%	80%
	S	0.62	1.24	1.86	2.48
	ZIP	1.6552	5.2414	8.2759	12.4138
	ZDAA	4.9655	7.9448	11.0455	13.9586
	ImpR.D	200.00	51.58	33.47	12.44
	ZPAA	2.9793	7.2690	10.8359	15.0069
	ImpR.P	80.00	38.68	30.93	20.89
	Gap	15.81	11.00	6.07	4.44

Case 3: Medium-scale Networks (|W|= 600)

Table 5- 6: Experiment Results of Medium-scale Networks for Model 1

APL = 4.33	**NT = 5×5 GD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.87	1.73	2.60	3.47
	ZIP	7.0000	11.0000	15.0000	20.0000
	ZDAA	8.9857	12.6182	16.5667	20.3750
	ImpR.D	28.37	14.71	10.44	1.88
	ZPAA	7.9143	12.8364	16.9667	22.1750
	ImpR.P	13.06	16.69	13.11	10.88
	Gap	66.67	40.43	20.23	10.11
BAP = Deg	PD	20%	40%	60%	80%
	S	0.87	1.73	2.60	3.47
	ZIP	5.6250	10.3125	15.6250	20.6250
	ZDAA	9.4018	13.9176	17.5898	21.6219
	ImpR.D	67.14	34.96	12.58	4.83
	ZPAA	8.2009	13.4347	17.6719	22.5188
	ImpR.P	45.79	30.28	13.10	9.18
	Gap	52.01	33.33	19.11	9.43
APL = 3.21	**NT = SF**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.64	1.28	1.92	2.57
	ZIP	3.0000	6.0000	10.0000	17.0000
	ZDAA	5.2667	7.9000	11.5000	18.4059
	ImpR.D	75.56	31.67	15.00	8.27
	ZPAA	5.2667	8.3833	13.9500	18.4059
	ImpR.P	75.56	39.72	39.50	8.27
	Gap	72.22	44.22	28.94	15.64
BAP = Deg	PD	20%	40%	60%	80%
	S	0.64	1.28	1.92	2.57
	ZIP	5.8511	10.9043	15.9574	20.2128
	ZDAA	9.6543	12.4012	18.0036	21.3298
	ImpR.D	65.00	13.73	12.82	5.53
	ZPAA	9.1223	13.0661	17.6499	21.3298

	ImpR.P	55.91	19.83	10.61	5.53
	Gap	51.89	30.01	19.02	11.43
APL = 3.4	**NT = RD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.68	1.36	2.04	2.72
	ZIP	5.0000	9.0000	13.0000	18.0000
	ZDAA	6.0000	9.7667	14.9077	18.5611
	ImpR.D	20.00	8.52	14.67	3.12
	ZPAA	7.3000	13.3667	17.9308	21.8833
	ImpR.P	46.00	48.52	37.93	21.58
	Gap	79.01	51.77	30.29	19.82
BAP = Deg	PD	20%	40%	60%	80%
	S	0.68	1.36	2.04	2.72
	ZIP	3.9894	8.7766	15.1596	20.2128
	ZDAA	7.7872	11.4894	17.9293	21.6013
	ImpR.D	95.20	30.91	18.27	6.87
	ZPAA	7.2287	13.1804	16.6698	21.2010
	ImpR.P	81.20	50.18	9.96	4.89
	Gap	76.66	52.22	30.93	19.83

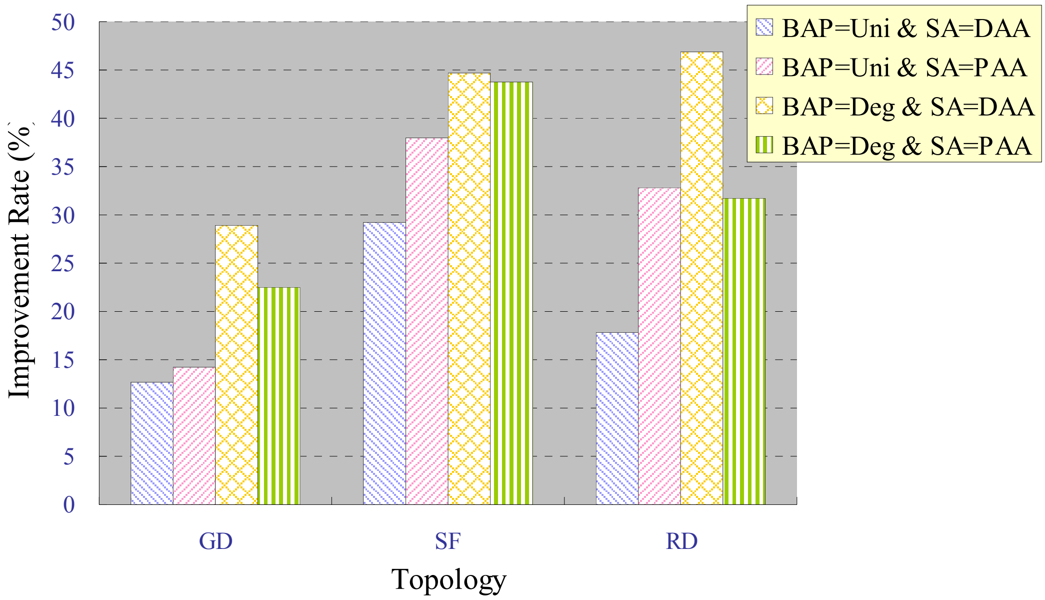
Case 4: Large-scale Networks (|W|= 1260)

Table 5- 7: Experiment Results of Large-scale Networks for Model 1

APL = 5	NT = 6×6 GD				
BAP= Uni	PD	20%	40%	60%	80%
	S	1.00	2.00	3.00	4.00
	ZIP	12.0000	16.0000	22.0000	28.0000
	ZDAA	15.0000	19.8000	24.7091	29.5714
	ImpR.D	25.00	23.75	12.31	5.61
	ZPAA	12.6000	19.3500	25.1182	32.0143
	ImpR.P	5.00	20.94	14.17	14.34
	Gap	49.09	41.11	28.18	10.33
BAP = Deg	PD	20%	40%	60%	80%
	S	1.00	2.00	3.00	4.00
	ZIP	9.9000	16.2000	22.5000	29.7000
	ZDAA	15.4575	22.4220	26.6632	31.0231
	ImpR.D	56.14	38.41	18.50	4.45
	ZPAA	12.9975	18.6840	25.4305	32.6948
	ImpR.P	31.29	15.33	13.02	10.08
	Gap	63.33	50.05	31.00	17.01
APL = 3.58	NT = SF				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.72	1.43	2.15	2.87
	ZIP	6.0000	10.0000	16.0000	25.0000
	ZDAA	9.9000	13.5000	18.6750	26.4240
	ImpR.D	65.00	35.00	16.72	5.70
	ZPAA	8.7000	15.8400	20.0250	27.3240
	ImpR.P	45.00	58.40	25.16	9.30
	Gap	88.28	64.44	48.23	31.11
BAP = Deg	PD	20%	40%	60%	80%
	S	0.72	1.43	2.15	2.87
	ZIP	8.0870	16.4348	23.2174	28.9565
	ZDAA	13.8410	20.1534	27.8750	30.8288
	ImpR.D	71.15	22.63	20.06	6.46
	ZPAA	13.4311	21.4838	26.2370	31.0375

	ImpR.P	66.08	30.72	13.01	7.19
	Gap	89.44	70.03	45.98	29.19
APL = 3.74	**NT = RD**				
BAP = Uni	PD	20%	40%	60%	80%
	S	0.75	1.49	2.24	2.99
	ZIP	8.0000	13.0000	19.0000	25.0000
	ZDAA	10.3500	15.9231	21.0316	26.8560
	ImpR.D	29.38	22.49	10.69	7.42
	ZPAA	9.4500	17.7923	24.4421	30.7800
	ImpR.P	18.13	36.86	28.64	23.12
	Gap	90.22	77.89	67.00	42.22
BAP= Deg	PD	20%	40%	60%	80%
	S	0.75	1.49	2.24	2.99
	ZIP	5.4783	14.3478	21.9130	29.7391
	ZDAA	12.2087	19.7530	26.2462	31.7037
	ImpR.D	122.86	37.67	19.77	6.61
	ZPAA	8.9217	19.6904	24.4201	30.7906
	ImpR.P	62.86	37.24	11.44	3.54
	Gap	81.23	60.61	55.11	49.77

5.3.2 Discussion of Experiment Results for Model 1



NT	GD	SF	RD
BAP=Uni & SA=DAA	12.69	29.25	17.84
BAP=Uni & SA=PAA	14.18	37.91	32.76
BAP=Deg & SA=DAA	28.87	44.75	46.86
BAP=Deg & SA=PAA	22.44	43.81	31.66

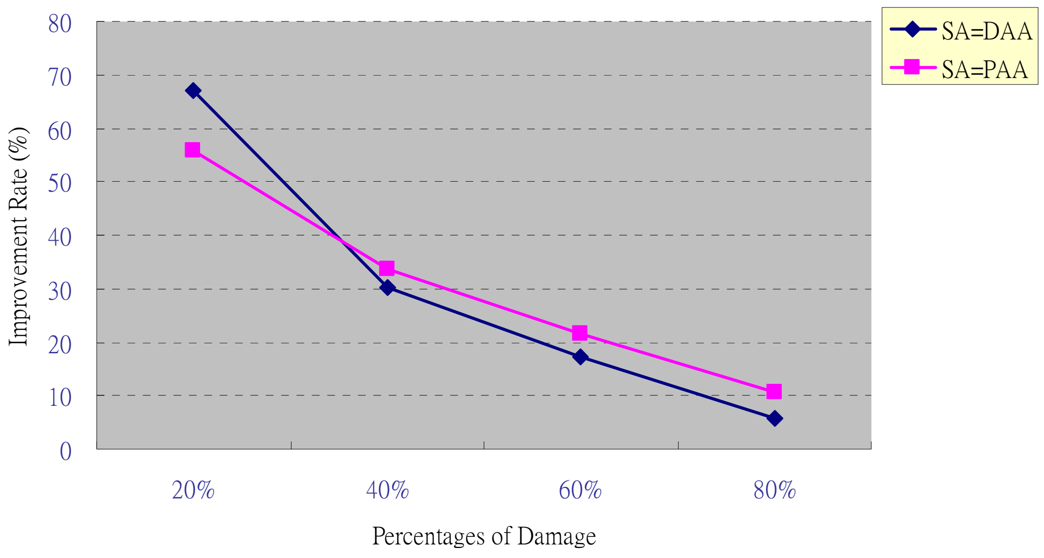
* *This table is the summary of the Table 5-4 ~ Table 5-7.*

Figure 5- 1: Improvement Rates under Different Attack Algorithms

The greater the improvement rate, the better the performance of 3SS. From Figure 5-1, we observe the following phenomena:

- According to three kinds of network topologies, the performance of 3SS is the best in the SF topology, on average.
- As a whole, when BAP=Deg and SA=DAA, the performance of 3SS is the best.

The reason is that attack order of a node in accordance with the defense budget on it will result in the growth of total attack cost. If we use the 3SS attack algorithm, we can avoid this problem, and can decrease our attack cost effectively.



PD	20%	40%	60%	80%
SA= DAA	67.1533	30.2634	17.1227	5.6408
SA= PAA	55.8176	33.7921	21.6060	10.6350

* *This table is the summary of the Table 5-4 ~ Table 5-7.*

Figure 5- 2: Improvement Rates under Different Percentages of Damage

From Figure 5-2, we observe that the lower the percentage of damage, the better the performance of 3SS. That is, an attacker using 3SS could decide which nodes could be attacked more precisely than using the other two algorithms.

5.3.3 Experiment Results of Model 2

Because of time considerations, we only adopt the Uni budget allocation policy (BAP) in the implementation of Model 2, not the Deg policy. To compare the results of Model 2, we adopt two budget re-allocation policies (BRAP): Uni and Deg. ***ImpR.1*** and ***ImpR.2***, which denote the Improvement Rate of Uni and Deg, respectively, are calculated by $(\frac{Uni-Deg}{Deg}\times 100\%)$ and $(\frac{Deg-Uni}{Uni}\times 100\%)$.

Case 1: Extra-small-scale Networks (|W|= 72)

Table 5- 8: Experiment Results of Extra-small-scale Networks for Model 2

NT	BRAP	PD			
		20%	40%	60%	80%
GD	Uni	1.6497	3.7401	5.4093	7.6520
	Deg	1.5018	3.3918	5.2823	7.0149
	ImpR.1	0.0985	0.1027	0.0240	0.0908
	ImpR.2	-0.0897	-0.0931	-0.0235	-0.0833
SF	Uni	1.6497	3.3617	5.5002	8.1047
	Deg	1.4827	3.0630	5.5848	8.9996
	ImpR.1	0.1126	0.0975	-0.0151	-0.0994
	ImpR.2	-0.1012	-0.0889	0.0154	0.1104
RD	Uni	1.5673	3.4815	5.0463	6.5007
	Deg	1.6497	3.6625	5.2149	7.5530
	ImpR.1	-0.0499	-0.0494	-0.0323	-0.1393
	ImpR.2	0.0526	0.0520	0.0334	0.1619

Case 2: Small-scale Networks (|W|= 240)

Table 5- 9: Experiment Results of Small-scale Networks for Model 2

NT	BRAP	PD			
		20%	40%	60%	80%
GD	Uni	3.3909	6.5495	9.7925	13.5510
	Deg	2.9369	6.2285	9.3253	12.7187
	ImpR.1	0.1546	0.0515	0.0501	0.0654
	ImpR.2	-0.1339	-0.0490	-0.0477	-0.0614
SF	Uni	2.4958	5.7234	9.0558	14.2382
	Deg	3.2896	6.9390	10.8232	15.8095
	ImpR.1	-0.2413	-0.1752	-0.1633	-0.0994
	ImpR.2	0.3181	0.2124	0.1952	0.1104
RD	Uni	2.6918	6.2410	11.6403	14.8549
	Deg	2.5248	7.0672	14.4417	15.8846
	ImpR.1	0.0661	-0.1169	-0.1940	-0.0648
	ImpR.2	-0.0620	0.1324	0.2407	0.0693

Case 3: Medium-scale Networks (|W|= 600)

Table 5- 10: Experiment Results of Medium-scale Networks for Model 2

NT	BRAP	PD			
		20%	40%	60%	80%
GD	Uni	5.0460	11.1238	15.2407	22.1697
	Deg	5.6562	10.2497	15.3211	21.8958
	ImpR.1	-0.1079	0.0853	-0.0052	0.0125
	ImpR.2	0.1209	-0.0786	0.0053	-0.0124
SF	Uni	5.3884	10.1925	19.1700	22.9000
	Deg	5.6064	12.2871	19.5696	24.1486
	ImpR.1	-0.0389	-0.1705	-0.0204	-0.0517
	ImpR.2	0.0405	0.2055	0.0208	0.0545
RD	Uni	4.6034	9.9127	16.4048	23.8316
	Deg	4.7489	9.9405	18.1490	24.4907
	ImpR.1	-0.0306	-0.0028	-0.0961	-0.0269
	ImpR.2	0.0316	0.0028	0.1063	0.0277

Case 4: Large-scale Networks (|W|= 1260)

Table 5- 11: Experiment Results of Large-scale Networks for Model 2

NT	BRAP	PD			
		20%	40%	60%	80%
GD	Uni	8.8942	16.1263	22.4339	30.6775
	Deg	9.4200	16.1976	22.6252	30.0161
	ImpR.1	-0.0558	-0.0044	-0.0085	0.0220
	ImpR.2	0.0591	0.0044	0.0085	-0.0216
SF	Uni	8.1336	23.6278	31.2036	32.5037
	Deg	10.6275	23.6471	32.1730	32.5131
	ImpR.1	-0.2347	-0.0008	-0.0301	-0.0003
	ImpR.2	0.3066	0.0008	0.0311	0.0003
RD	Uni	6.6333	14.5887	24.4705	30.8294
	Deg	7.7851	14.6457	23.6859	31.6248
	ImpR.1	-0.1479	-0.0039	0.0331	-0.0252
	ImpR.2	0.1736	0.0039	-0.0321	0.0258

5.3.4 Discussion of Experiment Results for Model 2

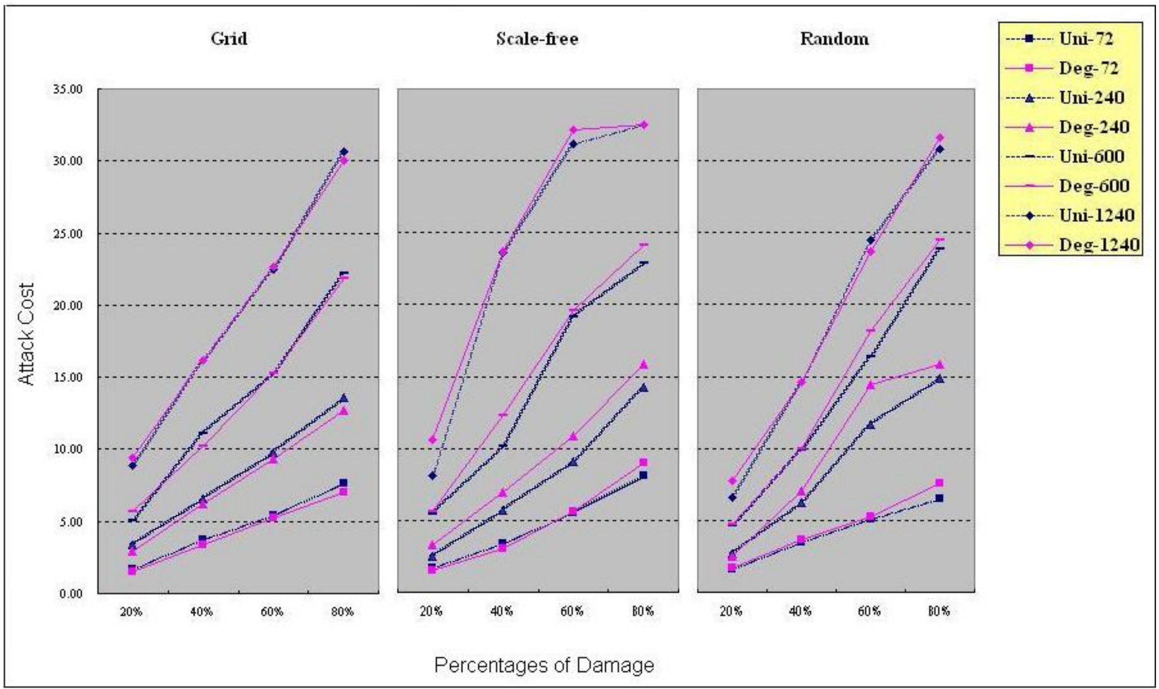


Figure 5- 3: Attack Costs under Different Percentages of Damage and Different Network Topologies

In the Grid graph of Figure 5-3, all the blue lines are a little higher than the red lines under different topology scales. After comparison, the performance of Uni BRAP is 0.58% as good as that of Deg BRAP.

In the Scale-free and Random graphs of Figure 5-3, all the blue lines are lower than the red lines under different topology scales. After comparison, the performance of Deg BRAP is 1.43% as good as that of Uni BRAP in the Scale-free graph. The performance of Deg BRAP is 1.02% as good as that of Uni BRAP in the Random graph.

Chapter 6 SUMMARY AND FUTURE WORK

6.1 Summary

In this work, we address two issues. First, we want to determine how to evaluate the survivability of a network effectively and efficiently. Second, we want to find a good budget re-allocation policy leading to a more robust ability of network defense.

The main contributions of this work are as follows:

1. **A new survivability metric – DOS (Degree of Separation)**

   - The DOS metric is an intuitively-realized and simply-calculated tool for measuring network survivability, shown as Section 2.1 and 2.2.
   - Some lemmas derived from DOS can be applied directly to calculate the upper bound of the DOS value, including for linear topology, ring topology, shown as Section 2.3.

2. **A double-layer mathematical model – max-min model**

   - The double-layer model demonstrates a subtle relationship between an attacker and a defender. By this double-layer model, we can understand the policy change of both parties over time. That is, through this model, we can see how a defender defends a target network will affect how an attacker attacks it, and vice versa.

- In the inner-layer of the model, we proposed a 3-Stage Selection (3SS) heuristic. We not only know what the minimal attack cost of the attacker is, but also realize how robust the network of a defender is. That is, the greater the minimal attack cost, the more robust the network will be
- In the outer-layer of the model, we provide network operators two budget re-allocation policies.

6.2 Future Work

1. Other DOS Metrics

Based on the quantitative method, we can propose other DOS metrics to calculate the damage level of a network. In addition to “average DOS” described in Chapter 2, there are two other survivability metrics: **Longest Damaged Path (LDP)** and **Minimal Recovery Node (MRN)**.

LDP-DOS metric used to measure the damage level of the OD pair with the most damaged among all OD pairs. The LDP-DOS value can be regarded as the most effort we should pay to recover the most damaged OD pair in a network. For example, in Figure 6-1, there are only two disconnected OD pairs: O_1-D_1 and O_2-D_2. The damage level of the first pair is four (marked by the thick and red line), and that of the second is two. Then, the LDP-DOS value is the damage level value of the first pair, i.e., four.

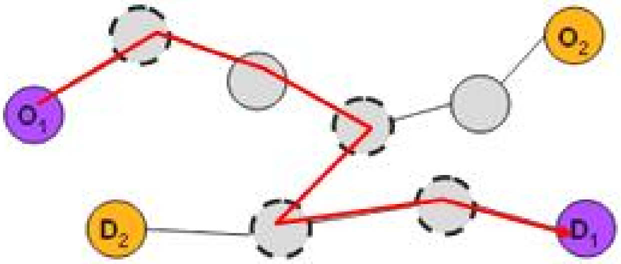


Figure 6- 1: Longest Damaged Path (LDP)

In addition, we define the **MRN-DOS** value as the minimal number of nodes that must be recovered to ensure that all OD pairs reconnect. That is, we must repair the least number of nodes to ensure that all OD pairs can communicate. For example, in Figure 6-2, the network has three disconnected OD pairs and three broken nodes. We only recover one node (circled by the thick and red line) so that the three OD pairs can reconnect. Therefore, the MRN-DOS value is one.

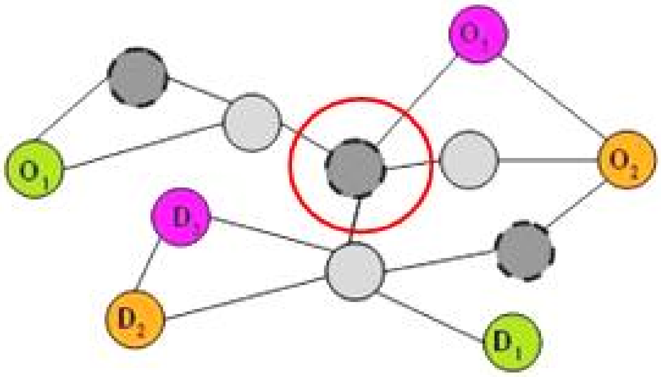


Figure 6- 2: Minimal Recovery Node (MRN)

2. Another Survivability Metric –Greatest Residual Region (GRR)

The GRR is also used to measure network survivability. In contrast to DOS, the GRR is used to measure the sound level of a network. For example, in Figure 6-3, the left-hand side is an intact network and the right-hand side is the attacked network with two broken nodes (colored black). The result of right-hand graph of Figure 6-3 is that whole network breaks into three regions and each region contains 5, 4 and 3 nodes, respectively. Because the GRR value is defined as the sound level of the biggest region of a broken network, we can see that the GRR value of Figure 6-3 is five. Naturally, if the GRR value is smaller than a given threshold, we say the network is compromised.

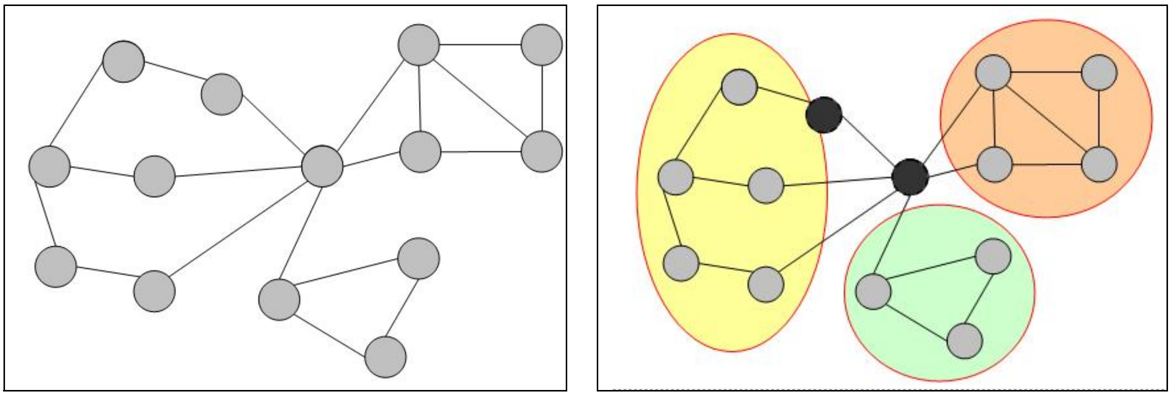


Figure 6- 3: Greatest Residual Region (GRR)

REFERENCES

[1] J.C. Knight and K.J. Sullivan, "On the Definition of Survivability," Technical Report CS-TR-33-00, Department of Computer Science, University of Virginia, December 2000.

[2] J.C. Knight, E.A. Strunk, and K.J. Sullivan, "Towards a Rigorous Definition of Information System Survivability," *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2003)*, Vol. 1, pp.78-89, April 2003.

[3] J.C. Knight, K. Sullivan, M.C. Elder, and C. Wang, "Survivability Architectures: Issues and Approaches," *Proceedings of the DARPA Information Survivability Conference and Exposition*, pp. 157-171, January 2000.

[4] S.C. Liew, and K.W. Lu, "A Framework for Network Survivability Characterization," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 1, pp. 52-58, January 1994 (ICC, 1992).

[5] V.R. Westmark, "A Definition for Information System Survivability," *IEEE Proceedings of the 37th Hawaii International Conference on System Sciences*, Vol. 9, pp. 90303.1, 2004.

[6] C. Taylor, P. Oman, and A. Krings, "Assessing Power Substation Network Security and Survivability: A Work in Progress Report," *Proceedings of the International Conference on Security and Management (SAM'03)*, Las Vegas, 2003.

[7] R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T. Longstaff, and N.R. Mead, "Survivable Network Systems: An Emerging Discipline," Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon

University, November 1997 (Revised: May 1999).

[8] H.F. Lipson, and D.A. Fisher, “Survivability—A New Technical and Business Perspective on Security,” New Security Paradigms Workshop, *Proceedings of the 1999 workshop on New security paradigms, ACM*, September 1999.

[9] Y.S. Lin, P.H. Tsang, C.H. Chen, C.L. Tseng, Y.L. Lin, “Evaluation of Network Robustness for Given Resource Allocation Strategies”, *Proceedings of the First International Conference on Availability, Reliability and Security, IEEE*, 2006.

[10] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, “Network Flows: Theory, Algorithms, and Applications: Chapter 16 Lagrangian Relaxation and Network Optimization,” Prentice-Hall, pp. 598-639, 1993.

[11] M.L. Fisher, “The Lagrangean Relaxation Method for Solving Integer Programming Problems,” *Management Science*, Vol. 27, No. 1, pp. 1-18, January 1981.

[12] M.L. Fisher, “An Application Oriented Guide to Lagrangean Relaxation,” *Interfaces*, Vol. 15, No. 2, pp. 10-21, March-April 1985.

[13] S. Redner, “How Popular Is Your Paper? An Empirical Study of the Citation Distribution,” *European Physical Journal B - Condensed Matter and Complex Systems*, pp. 131-134, 1998.

[14] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On Power-Law Relationships of the Internet Topology,” *ACM SIGCOMM Computer Communications Review*, Vol. 29, Number 4, pp. 251-263, September 1999.

[15] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, “Power-Laws and the AS-level Internet Topology,” *IEEE/ACM Transactions on Networking*, Vol. 11, Issue 4, pp. 514-524, 2003.

[16] P. Erdos, and A. Renyi, “On the Evolution of Random Graphs,” *Publ. Math. Inst. Hung. Acad. Sci.*, Vol. 5, pp. 17-60, 1960.

[17] D.J. Watts, and S.H. Strogatz, “Collective Dynamics of ‘Small-World’ Networks,” *Nature*, Vol. 393, pp. 440-442, 1998.

[18] R. Albert, H. Jeong, and A.L. Barabasi, “Diameter of the World-Wide Web,” *Nature*, Vol. 401, pp. 130-131, 1999.

[19] A.L. Barabasi, and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, Vol. 286, pp. 509-512, October 1999.

[20] R. Albert, H. Jeong, and A.L. Barabasi, “Error and Attack Tolerance of Complex Networks”, *Nature*, Vol. 406, pp. 378-382, July 2000.

[21] Z. Zeitlin, “Integer Allocation Problems of Min-Max Type with Quasiconvex Separable Functions,” Delft University of Technology, Netherlands, 1981.

[22] M.S. Deutsch, and R.R. Willis, “Software Quality Engineering: A Total Technical and Management Approach”, Englewood Cliffs, NJ: Prentice-Hall, 1988.

[23] A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr, “Basic Concepts and Taxonomy of Dependable and Secure Computing”, *IEEE Transactions on Dependable and Secure Computing*, Vol. 1, No. 1, January-March 2004.