國立臺灣大學管理學院資訊管理學研究所

碩士論文

Graduate Institute of Information Management
College of Management
National Taiwan University

Master Thesis

考慮攻擊環境下達到違反服務品質最小化之近似最佳化網路規劃及防禦資源配置策略

Near Optimal Network Planning and Defense Resource Allocation Strategies for Minimizing Quality-of-Service (QoS) Violations under Attacks

謝孜謙
Tzu-Chen Hsieh

指導教授：林永松 博士
Advisor: Yeong-Sung Lin, Ph.D.

中華民國 97 年 7 月
July, 2008

考慮攻擊環境下達到違反服務品質最小化之近似最佳化網路規劃及防禦資源配置策略

Near Optimal Network Planning and Defense Resource Allocation Strategies for Minimizing Quality-of-Service (QoS) Violations under Attacks

本論文係提交國立台灣大學
資訊管理學研究所作為完成碩士
學位所需條件之一部份

研究生：謝孜謙　撰

中華民國九十七年七月

博碩士論文電子檔案上網授權書

（提供授權人裝釘於紙本論文書名頁之次頁用）

本授權書所授權之論文為授權人在國立臺灣大學資訊管理學研究所 96 學年度第 二 學期取得博士、碩士學位之論文。

論文題目： 考慮攻擊環境下達到達反服務品質最小化之近似最佳化網路規劃及防禦資源配置策略

指導教授： 林永松博士

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權本人畢業系所，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

☐ 上列論文為授權人向經濟部智慧財產局申請專利之附件或相關文件之一（專利申請案號：　　　　　　），請於　　年　　月　　日後再將上列論文公開或上載網路。

☐ 因上列論文尚未正式對外發表，請於　　年　　月　　日後再將上列論文公開或上載網路。

授權人簽名： 謝孜謙

中　華　民　國 97 年 7 月　　　日

國立臺灣大學碩士學位論文
口試委員會審定書

考慮攻擊環境下達到違反服務品質最小化
之近似最佳化網路規劃及防禦資源配置策略

Near Optimal Network Planning and Defense Resource Allocation Strategies for Minimizing Quality-of-Service (QoS) Violations under Attacks

本論文係 謝孜謙 君（學號 R95725009）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 97 年 7 月 15 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：

趙啟超

孫雅麗 林永松

所　　長： 陳靜枝

謝誌

這篇論文要獻給我的父母：謝榮輝先生及鄧秀蘭女士，謝謝你們在我求學生涯的過程中給予最全力的支持，並在研究受挫時給予最溫暖的鼓勵，讓我更能勇於面對及接受挫折，才能更順利的完成我的學業。也謝謝我的哥哥謝劭謙，容忍我在水深火熱的階段提出的許多不合理的要求。

在這二年的研究生涯，最感謝的就是林永松老師的指導，在研究方面，無論是研究方向的擬定、研究方法的細節及論文的撰寫等等，都提供了最專業且最大的協助；另一方面，老師您不時的教導我們無論身在何處，都要表現我們做人應有的態度，這更是比做研究更重要的收獲阿！此外，也特別感謝清大通訊趙啟超教授、輔大資工呂俊賢教授以及本校孫雅麗教授和莊裕澤教授在論文口試的過程中提供寶貴的建議和指正，讓這篇論文能夠更加嚴謹且更完善。

另外，特別要感謝的是柏皓學長，您是這篇論文完成的大功臣，您總是在忙得不可開交的事務中，抽空來指導我們正確的方向，從帶領我到資安的領域，一路上的督導，到論文的前一刻都在為我們加油與協助，除了說感激還是感激。感謝霈語學姐給予許多研究及論文撰寫的建議，並在口試前給我最有信心的加油。感謝俊維學長精確的LR教學，也感謝雅芳學姐傳授我許多的口試經驗。

感謝奐廷、至浩、政祐及志元在這二年的研究生涯中，一起打拼，一起歡樂。感謝睿斌、竣韋、培維、猷順、冠瑋、宴毅及友仁，因為有你們，讓我可以無後顧之憂的全心準備口試。感謝子超、家禎在我撰寫論文遇到瓶頸時給予適時的幫助，感謝奕伃、立穎、偉倫等同學在研究生涯給予歡樂，感謝所有該感謝的人。最後把最真摯的誠意感謝神明，保佑我平安，保佑我研究、口試順利。

謝孜謙 謹識

于台大資訊管理研究所

民國九十七年七月

論文摘要

論文題目：考慮攻擊環境下達到違反服務品質最小化之近似最佳化網路規劃及防禦資源配置策略

作者：謝孜謙　　九十七年七月

指導教授：林永松 博士

隨著網際網路的方便性，資訊安全的問題也越來越重要。近幾年來，有意及無心的網路犯罪事件層出不窮。其中，攻克網路中某些特定的伺服器並降低其處理能力，是影響網路服務品質最常見的網路犯罪手法之一。因此我們應發展出有效的策略來防範如此的攻擊，例如防禦資源的配置。此外，網路規劃也必須納入資訊安全的考量。

在這篇論文中，我們提出一個最小最大化的數學規劃問題來塑造網路管理者和攻擊者間相互的行為。在內層問題（ARRAS問題）中，考慮的是一個攻擊者該選擇哪些節點來攻擊並有效配置其有限的攻擊資源，以最大化因為違反服務品質而網路管理者必須付出的代價，例如賠償。在外層問題（NPDRAS問題）中，網路管理者則希望在有限的預算中，設計一個良好的網路並有效的配置防禦資源，來最小化必須付出的代價。為了求得此問題的最佳解，我們利用拉格蘭日鬆弛法為基礎的演算法來處理內層的問題，並利用內層問題的解和調整預算的演算法來處理外層的問題。

關鍵詞：資訊安全、服務品質、數學規劃、資源配置、拉格蘭日鬆弛法、最佳化

THESIS ABSTRACT

Near Optimal Network Planning and Defense Resource Allocation Strategies for Minimizing Quality-of-Service (QoS) Violations under Attacks

Name：Tzu-Chen Hsieh July 2008

Advisor：Yeong-Sung Lin, Ph. D.

With the convenience of Internet, the problem of information security has caught more and more attentions. Events of witting or unwitting cybercrimes emerge in an endless stream in past years. Among them, to compromise particular servers and then degrade their process capability is one of the most popular cybercrimes in order to further affect the Quality-of-Service (QoS) of the network. For taking precautions against such attacks, we should develop effective defense strategies such as defense resources allocation. Besides, the network planning has to be considered in the realm of information security.

In the thesis, we propose a min-max mathematical programming problem to model the mutual behavior between a network administrator and an attacker. In the inner problem, called the ARRAS problem, the attacker would like to maximize the total penalty the administrator has to pay for due to QoS violations by deciding which node to attack and allocating the limited attack budget effectively. In the outer problem,

called the NPDRAS problem, the network administrator hopes to minimize the total penalty by planning a well network and allocating defense resources intelligently under a limited budget. For obtaining near optimal solutions, we use the Lagrangean relaxation-based algorithm to solve the ARRAS problem and exploit the solutions of ARRAS problem and the proposed budget adjustment procedure to solve the NPDRAS problem.

Keywords: Information Security, Quality-of-Service, Mathematical Programming, Resource Allocation, Lagrangean Relaxation, Optimization

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

According to Alvin Toffler's talk in "The Third Wave" in 1980 [1], as the first agrarian revolution ten thousand years ago and the second industrial revolution in the nineteen century, people will face the third revolution which is going to change people's lifestyle and economical view in the twentieth century, the so-called post-industrial revolution or information revolution. Indeed, with the popularity of computer and the rise of internet, the usage of computer extends increasingly from national defense and science to human entertainment, communication, and commercial affair. Many applications of emerging technology have also replaced numerous human physical behaviors in our daily lives. Due to the extensive usage of e-mail, web phone, electronic commerce, digital product and so forth, network services are indivisible from our daily lives. Therefore, the applications on the network services are developed quickly for the arrival of new age.

Among them, multimedia in the distributed environment is one of the popular applications on the network services. Common cited examples include Video-on-Demand (VoD), Multimedia-on-Demand (MoD), distance learning, videoconferencing, distributed games, distributed databases, and mass mailing [2]. In such applications, a network service provider has to guarantee the Quality-of-Service (QoS) requirements requested by users. For this reason, a network planner hopes to design an optimal communication planning in order to satisfy the QoS requirements, such as bandwidth, delay, delay jitter, packet loss, etc.

In order to achieve this one-to-many communication planning, multicast routing is the most frequent technology. Multicast represents the data transmission from a single source to multiple destinations belonging to the same group in a communication network. Multicast routing refers to the path selection for data transmission which has to satisfy the QoS requirements requested by the downstream users. Finally, a tree rooted at a single source and terminated at all destinations is generated, which is the so-called multicast routing tree. A Steiner Minimal Tree (SMT) is the multicast routing tree with the minimal overall cost. The algorithm of determining a Steiner minimal tree is known as NP-Complete problem [3].

However, with the convenience of information, the problem of information security has caught more and more attentions. Events of witting or unwitting cybercrimes emerge in an endless stream in past years, which is shown in **Figure 1-1** [4]. Besides, nature disasters also damage components in a network to break the data transmission. Therefore, the network planning has increasingly subsumed the realm of information security.

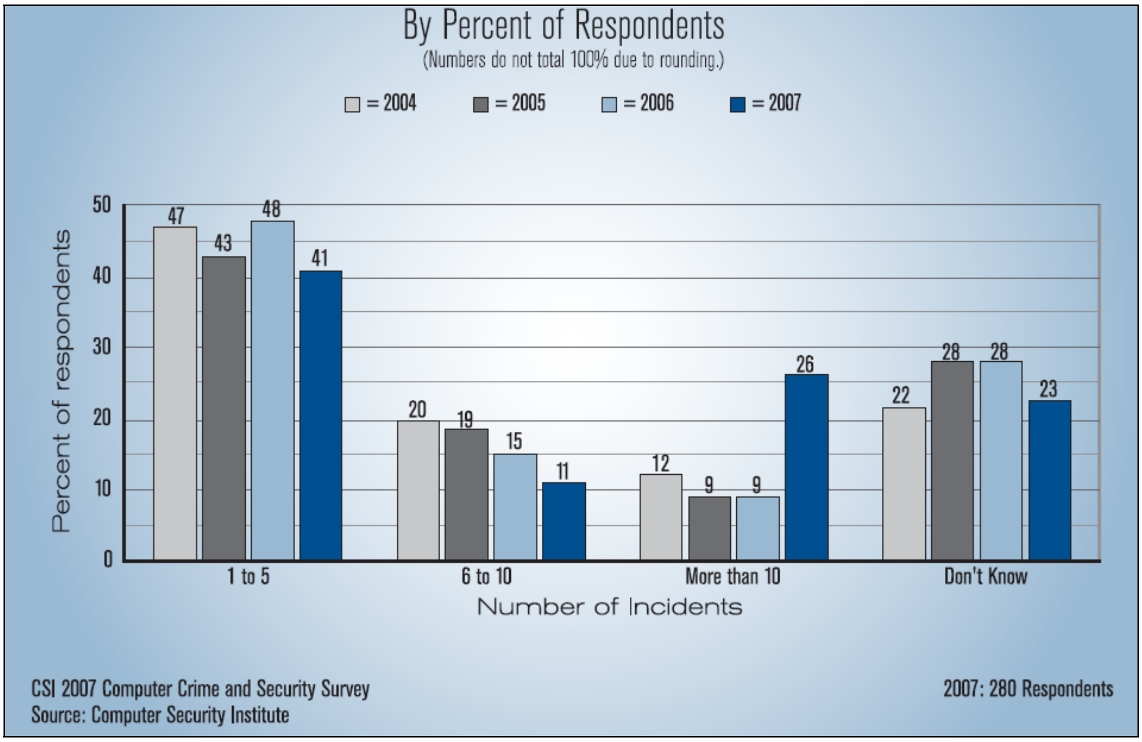


Figure 1-1. How Many Incidents in the Past 12 Months?

A great deal of security technologies have been proposed to strengthen the network robustness against malicious attacks and nature disasters in recent years as **Figure 1-2** shows [4]. Nevertheless, because there is no perfect technology and communication protocol, and the behavior of an attacker is unexpected, the network administrator can't guarantee the robustness of the network out and out. The attacker is always capable of finding the vulnerabilities of the network and then maximizing the damage of the network by the most powerful attacks. However, the network administrator could change the network planning and defense resource allocation strategies to degrade the damage of the network under such attacks. In another word, the attacker and the network administrator could constantly modify their strategies to resist the other side until the optimal defense strategy can be generated to maximize the network survivability.

Many scholars have researched in the field of survivability for a while. However, the definition and the measurement of network survivability are not consistent among them. According to the survey of [5], “the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures or accidents,” proposed by Ellison *et al.* in 1999 [6], is the most frequent definition of survivability.

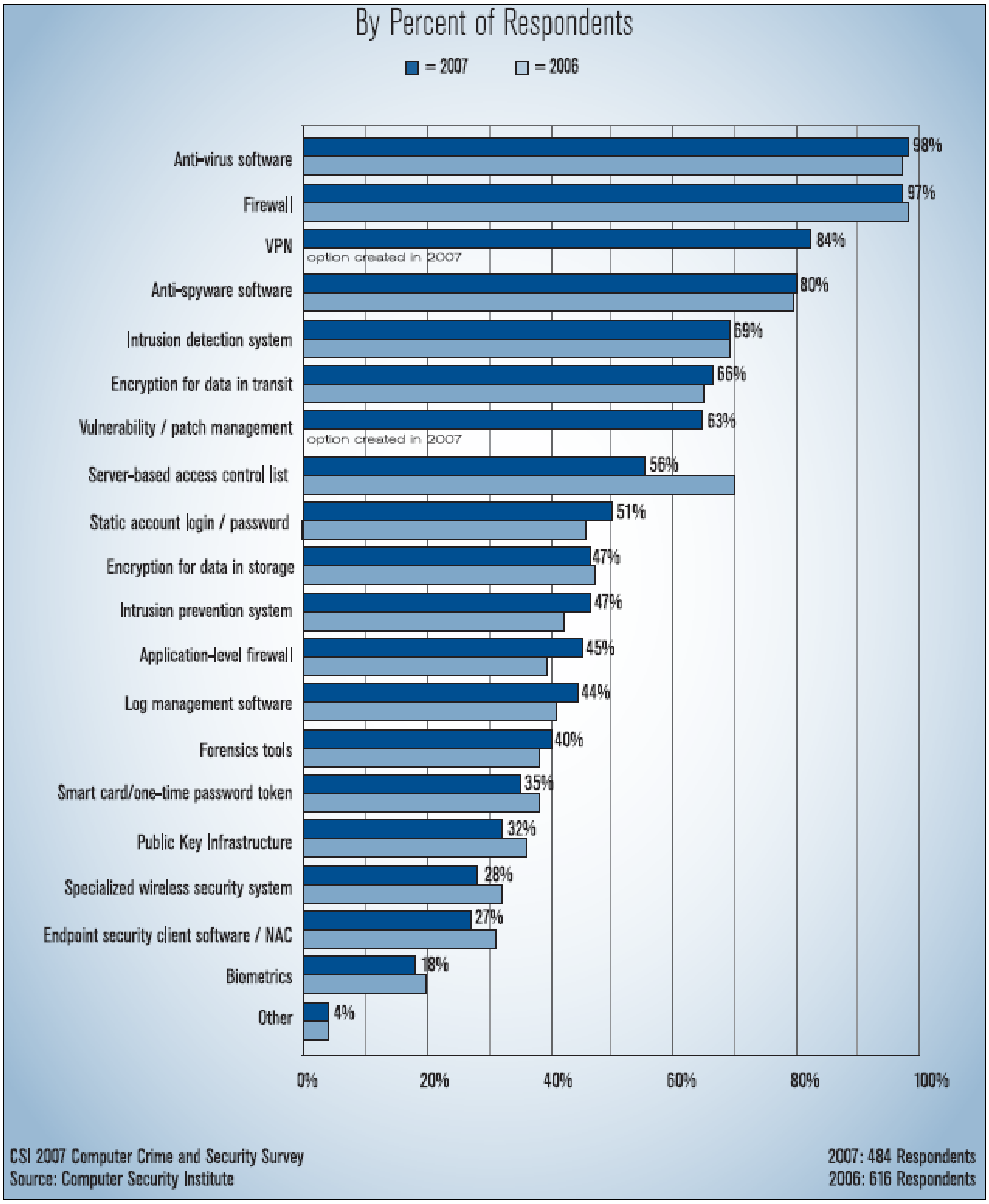


Figure 1-2. Security Technologies Used

1.2 Motivation

In a distributed environment, an attacker can attack the critical points of multicast routing trees and affect the QoS requested by users. For instance, an attacker could embed some useless programs in the critical points to degrade their operating capabilities and then cause slow transmissions or even fail transmissions. The more the ability to provide reliable QoS under attacks is, the more the users' willingness of paying for network services is. On the contrary, when QoS violations occur, the user would request the penalty for contract violations, or even cancel the contract.

With the limited budget, a network administrator needs to deploy defense budget effectively to decline the penalty due to QoS violations. Similarly, an attacker will allocate attack budget appropriately with the limited attack budget. The two opposites will constantly change their respective strategies according to the other's strategy. Through our surveys, however, there are few theoretical researches using mathematical manners to discuss the mutual behavior between a network administrator and an attacker. Therefore, we propose a mathematical model to formulate the mutual behavior and solve it by our proposed solution approaches. Finally, we will also provide the useful indicator of defense strategies to a network administrator to minimize the penalty under attacks.

From related researches, moreover, the defense resource allocation is mostly considered after network planning. We hope to consider the realm of defense in the phase of network planning. Therefore, we can implement extra the capacities of links and nodes by investing some budget to decrease the time of transmissions, and even to decline the chance of QoS violations.

1.3 Literature Survey

1.3.1 IP Multicast

Multicast means the data transmission from a single source to multiple destinations in a group. In generally, a spanning tree is one of the most efficient methods to achieve the data transmission to connect all the members in the group. The algorithm of constructing a spanning tree for the group is called multicast routing algorithm.

For multicast algorithms nowadays, according to the research proposed by Bin Wang *et al*. in 2000 [7], there are two types of tree: *the source-based tree* and *the core-based tree* (or *the share tree* [8]), which depends on how a tree is generated.

A source-based tree is a source-rooted tree composed of the shortest paths among the source and all destinations in a multicast group. That is to say, the source-based tree can mainly be characterized by a Shortest Path Tree (SPT). Generally, in a multicast group, there may have many separate SPTs, one for each source. Reverse Path Forwarding (RPF) is one of the common routing mechanisms to derive the shortest path to build a SPT [8]. The Multicast extensions for Open Shortest Path First protocol (MOSPF) and Distance-Vector Multicast Routing Protocol (DVMRP) are the cited source-based tree protocols using SPT [6].

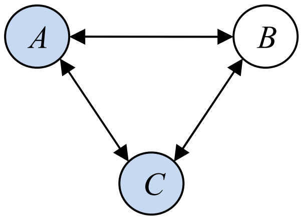
Of course, the primary advantage of a SPT is the minimal end-to-end delay from a source to each destination. The characteristic makes the SPT be suitable to timely applications, such as videoconferencing, which are mainly delay-sensitive and have a high bandwidth requirement [2][3]. With a large number of multicast groups and sources, however, the routers' memories could be exhausted. In other words, we assume there are m groups in a network, and n sources for each group, then $m \times n$ routing tables have to be stored in the routers of the network [9].

In order to solve this storage problem, the core-based tree or the shared tree has been proposed. There is only a tree used by all the sources of a multicast group. Each source has to sends data to a single node which called core, center, or Rendezvous Point (RP) [7] and the RP then forwards the data to the designate destinations. Core Based Tree (CBT) and Protocol Independent Multicast-Sparse Mode (PIM-SM) are the famous protocols of core-based tree [6].

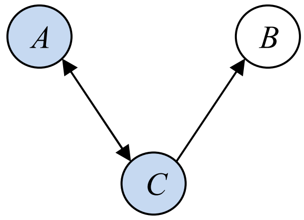
The main advantage of a core-based tree is to save the router storages because of the tree sharing. There are only *m* routing tables to be stored in the routers while the network has *m* groups. But the path from a source to a destination through the RP may cause much delay than the minimal. Besides, there exists a critical problem for data transmission, which means traffic concentration. The bottleneck is the RP when all sources in a group transmit data in the meantime. Furthermore, how to choose the optimal RP in the core-based tree is an NP-Complete problem [8].

Figure 1-3 [10] shows an example of traffic concentration. There are three members *A*, *B*, and *C*, in a multicast group connected with directed link as **Figure 1-3(a)** shows. Among them, node *A* and *C* are two sources with the same sending rate. **Figure 1-3(b)** shows a core-based tree used by all the sources of the group. **Figure 1-3(c)** shows two SPTs, one for each source. Clearly, link *CB* has two flows in **Figure 1-3(b)**, but all links have only one flow at most in **Figure1-3(c)**.

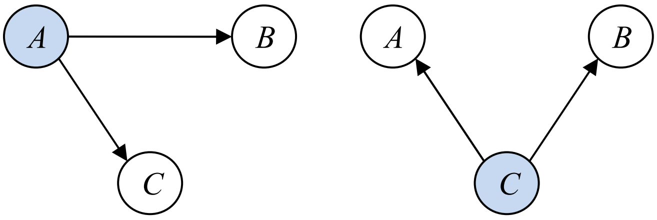
In generally, the type of tree is an alternative which depends on the distribution of destinations throughout a network. A source-based tree is optimized for densely distributed destinations and a core-based tree is suitable for sparse mode [8].



(a) Sample network



(b) Core-based tree



(c) Source-based tree

Figure 1-3. Traffic Concentration Example

1.3.2 QoS Routing

With the development of multimedia applications, the demand for QoS has been increasingly considered in multicast routing. The multicast routing tree has to satisfy the QoS requirements, such as bandwidth, delay, and delay jitter, requested by users. In other words, the QoS requirements have to be characterized by some constraints for solving a problem of multicast routing.

Bin Wang *et al*. [7] propose two categories of such constraints: *link constraints* and *tree constraints*. The link constraints are the usage limitations of links while routing. For example, the total consumed bandwidth of any link cannot exceed the capacity of the link. The tree constraints include the restrictions of all end-to-end transmissions from the source to destinations and the limitations between all transmissions in a multicast routing tree. For example, the end-to-end delay of any transmission and the delay jitter between any two transmissions must satisfy to the requirement request by users.

Clearly, a tree constraint is composed of some link metrics along with the multicast routing tree. According to the relationship between a tree constraint and the corresponding link metrics, the tree constraints can be divided into three types as following [7]:

1. *Transitive tree constraints* (or *Concave tree constraints* [11]): Available bandwidth is one of transitive tree constraints. For example, we assume $bw(R_1 \rightarrow R_2)$ is the available bandwidth from node R_1 to R_2 and $bw(R_2 \rightarrow R_3)$ is the available bandwidth from node R_2 to R_3, then the available bandwidth from node R_1 to R_3 through R_2 is

$$bw(R_1 \rightarrow R_2 \rightarrow R_3) = min[bw(R_1 \rightarrow R_2), bw(R_2 \rightarrow R_3)].$$

2. *Additive tree constraints*: End-to-end delay is one of additive tree constraints. For example, we assume $d(R_1 \rightarrow R_2)$ is the delay from node R_1 to R_2 and $d(R_2 \rightarrow R_3)$ is the delay from node R_2 to R_3, then the delay from node R_1 to R_3 through R_2 is

$$d(R_1 \rightarrow R_2 \rightarrow R_3) = d(R_1 \rightarrow R_2) + d(R_2 \rightarrow R_3).$$

3. *Multiplicative tree constraints*: Reliability is one of multiplicative tree constraints. For example, we assume $r(R_1 \rightarrow R_2)$ is the reliability from node R_1 to R_2 and $r(R_2 \rightarrow R_3)$ is the reliability from node R_2 to R_3, then the reliability from node R_1 to R_3

through R_2 is

$$r(R_1 \rightarrow R_2 \rightarrow R_3) = r(R_1 \rightarrow R_2) \times r(R_2 \rightarrow R_3).$$

Besides, a multiplicative tree constraint can be transformed into an additive tree constraint using logarithm.

Zheng Wang *et al*. [12] have proved that a path routing problem with multiple additive tree constraints and/or multiple multiplicative tree constraints in any combination is NP-Complete.

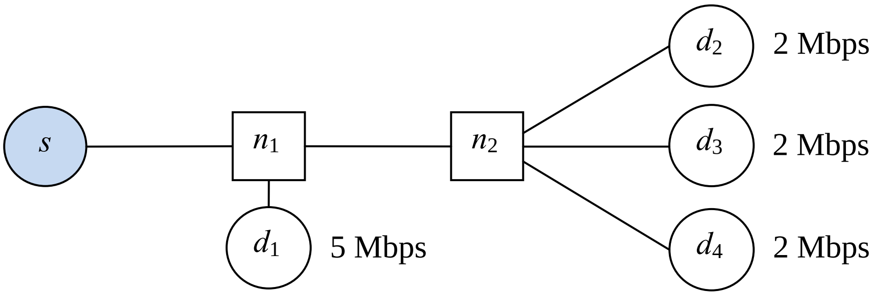
With the difference of constraints and the difference of objective function, the QoS multicast routing problems can be classified into twelve categories as **Table 1-1** shows [7].

1.3.3 Single-Application Multiple-Stream

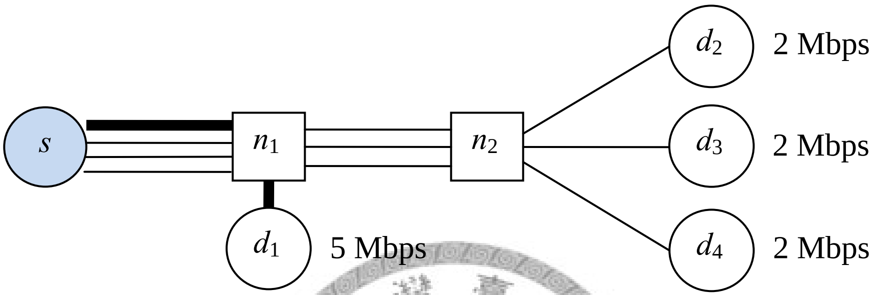
In a QoS multicast routing problem, there may have several significantly varied bandwidth requirements because of the heterogeneity of network and the different qualities requested by different destinations as **Figure 1-4(a)** shows. Node s is the source and node d_1, d_2, d_3, and d_4 are destinations in a multicast group where node d_1 requests 5 Mbps bandwidth requirement and nodes d_2, d_3, and d_4 request 2 Mbps bandwidth requirement respectively.

Table 1-1. A Taxonomy of Multicast Routing Problems

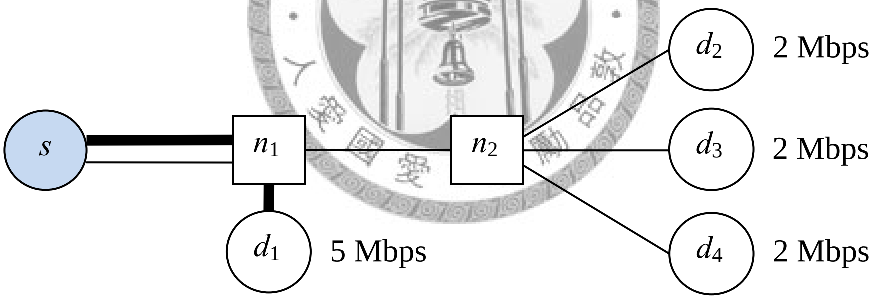
	No optimization	Complexity	Example
Null constraint			
Link constraint	(1) Link-constrained	Polynomial time	Bandwidth-constrained routing
	(2) Multiple-link-constrained	Polynomial time	Bandwidth- and buffer-constrained routing
Tree constraint	(3) Tree-constrained	Polynomial time	Delay-constrained routing
	(4) Multiple-tree-constrained	NP-complete	Delay- and interreceiver-delay-jitter-constrained routing
Link and tree constraints	(5) Link- and tree-constrained	Polynomial time	Delay- and bandwidth-constrained routing
	Link optimization	**Complexity**	**Example**
Null constraint	(6) Link optimization	Polynomial time	Maximization of the link bandwidth over on-tree links in a multicast tree
Link constraint	(7) Link-constrained link optimization	Polynomial time	The bandwidth-constrained buffer optimization problem
Tree constraint	(8) Tree-constrained link optimization	Polynomial time	The delay-constrained bandwidth optimization problem
Link and tree constraints			
	Tree optimization	**Complexity**	**Example**
Null constraint	(9) Tree optimization	NP-complete	Minimization of the total cost of a multicast tree
Link constraint	(10) Link-constrained tree optimization	NP-complete	The bandwidth-constrained Steiner tree problem
Tree constraint	(11) Tree-constrained tree optimization	NP-complete	The delay-constrained Steiner tree problem
Link and tree constraints	(12) Link- and tree-constrained tree optimization	NP-complete	The bandwidth- and delay-constrained tree optimization problem



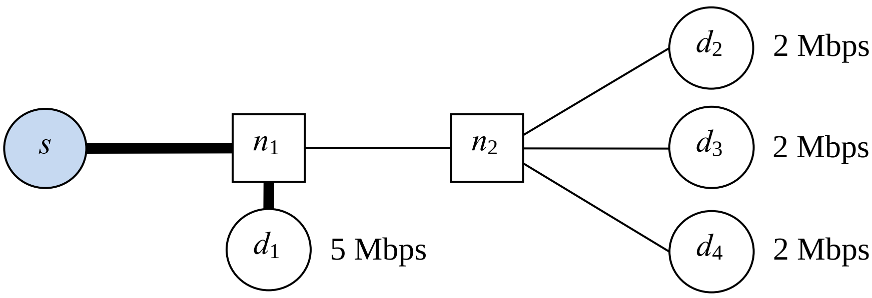
(a) Example Network



(b) Unicast video distribution



(c) Multicast video distribution



(d) Multicast video distribution with multi-layered coding

Figure 1-4. Video Distribution [13]

Figure 1-4(b) illustrates the transmissions from the source node to all destinations using unicast video distribution. There is an 11 Mbps bandwidth requirement for the link from node s to n_1 and a 6 Mbps bandwidth requirement for the link from node n_1 to n_2.

Figure 1-4(c) shows the transmissions using multicast video distribution. There is a 7 Mbps bandwidth requirement for the link from node s to n_1 and a 2 Mbps bandwidth requirement for the link from node n_1 to n_2. The bandwidth requirement of multicast is less than this of unicast because many destinations share the same traffic.

With the usage of a video gateway or progress coder, and the advance of video encoding and transmission technologies such as the multi-layered coding method [14], a source and video gateways transmit only one signal that is sufficient for the highest bandwidth requirement of downstream destinations. The concept is called Single-Application Multiple-Stream (SAMS) [13]. **Figure 1-4(d)** is an instance of SAMS. Thus, there is only a 5 Mbps bandwidth requirement for the link from node s to n_1. Therefore, SAMS has attracted more and more attention in multicast routing problem in recent years.

1.3.4 Survivability

In the generation full of information, the incidents of cybercrime have increased greatly with the growth of internet. The problems of such events are threatening our daily lives nowadays. Therefore, a large number of businesses and people have increasingly attached great importance to the domain of information security. By this trend, the term *survivability* has appeared in recent years.

The concept of survivability is not equal to this of security. According to [5], an application with security mechanisms such as encryption is probably dedicate yet whereas a survivability application has to be capable of surviving under attacks. Hence, security is included to survivability.

A great quantity of research on survivability has been proposed in recent years as **Table 1-2** shows. However, the precise definition of survivability is varied. In general, the definition of survivability is to measure the degree of anticipations of all users [15].

The definition of survivability in [6] is the most common one [5]. The terms *system*, *mission*, *attack*, *failure*, and *accident* are described as follows:

1. *System*: A system refers to a network or a large-scale system.
2. *Mission*: A mission represents a set of very high-level requirements or goals.
3. *Attack*: Attacks are the potentially damaging events caused by a malicious adversary. Attacks include intrusions, probes, denials of service (DoS), distributed DoS (DDoS), and etc.

Table 1-2. Definitions of Survivability

No.	Researcher(s)	Definition	Year	Ref.
1.	Louca, Pitsillides, and Samaras	The ability of a network to maintain or restore an acceptable level of performance during network failure conditions by applying various restoration techniques.	1999	[16]
2.	Ellison, Fisher, and Linger	The capacity of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents.	1999	[6]
3.	Knight and Sullivan	The ability to continue to provide service, possibly degraded or different, in a given operating environment when various events cause major damage to the system or its operating environment.	2000	[17]
4.	Westmark	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.	2004	[15]

4. *Failure*: Failures are the potentially damaging events caused by the deficiencies in the system. Failures include software design errors, hardware degradation, human errors, corrupted data, and so forth.
5. *Accident*: Accidents are the potentially damaging events caused by randomly occurring. With the contrast to failures, accidents are generated outside the system. A natural disaster is an example of accident.

Westmark divided the measurement of survivability into three categories: *connectivity*, *network performance*, and *a function of other quality or cost measures* [15]. We use the performance metric as the measurement of survivability in our model.

That is to say, the more the degree of satisfying the QoS under malicious attacks is, the more the survivability is.

1.4 Proposed Approach

We model the problem as a min-max optimization problem, which is also a nonlinear mathematical programming problem. Because of its high complexity, we are going to apply the Lagrangean relaxation and the subgradient method, and design optimization-based heuristics to solve the problem.

1.5 Thesis Organization

The remainder of the thesis is organized as follows. In **Chapter 2**, we propose the NPDRAS and the APRAS problems, and formulate them as mathematical models. In **Chapter 3**, we apply the Lagrangean relaxation approach to decompose the APRAS problem into several subproblems and solve each subproblem optimally. In **Chapter 4**, we propose heuristics for the two problems to get primal feasible solutions. In **Chapter 5**, we present our computational experiments and results for the two problems. Finally, in **Chapter 6**, we summary our conclusions and suggest some possible direction for the future works.

Chapter 2 Problem Formulation

2.1 Problem Description

The problem we discuss is at the Autonomous System (AS) level. There is a lot of network domains such as sets of subnets in the AS and no connection between any two domains. A user group is an application requesting for data transmissions like multimedia in the AS, which transmits data from a single domain called source to multiple domains called destinations. Each destination of different user groups may request various QoS requirements including traffic, end-to-end delay, and multiple paths demands. Therefore, a network administrator has to decide which connections to set and the capacities of them for data transmissions. In order to illustrate the problem conveniently, we model the AS as a graph where domains are depicted as nodes and there is no link between any two nodes. Furthermore, we assume that all nodes in the AS have video encoding and transmission technologies for data transmissions.

After the AS topology is generated by the network administrator, an attacker outside the AS will attack nodes in the AS through entry nodes. A node is compromised if the attacker applies adequate attack budget to break the nodal defense capability and finds a path from the attacker's source to the target node where all intermediate nodes on the path are compromised. After compromising a node, the attacker can apply extra attack budget to the node to degrade its capacity. For instance, the attacker could embed useless programs to a node to exhaust its CPU process capability. The effect of the degradation of nodal capacity may cause the increment of the end-to-end delay of each transmission through that node. Once the end-to-end delay is violated, the network administrator has to pay for the penalty to corresponding destinations. The objective of the attacker is to maximize the total penalty for which the network administrator has to pay by deciding which nodes to compromise and allocating the attack budget effectively to degrade nodal capacities within the limited attack budget.

From the network administrator perspective, he/she can allocate defense budget to protect the network as **Figure 2-1** shows. The defense budget can be divided into two categories: one is to strengthen the nodal defense capability from compromising, and the other is to enhance the extra nodal capacity. The relationship among the budget for strengthening the defense capabilities and the extra capacities of nodes is a trade off because the budget is limited. The objective of the network administrator is to minimize the total penalty incurred by the attacker by allocating the defense budget appropriately.

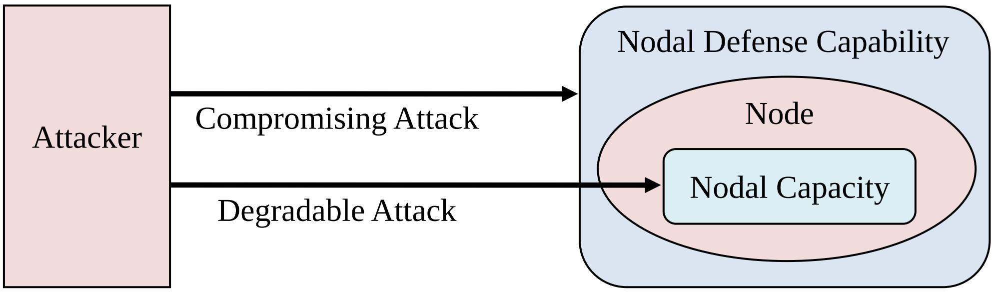


Figure 2-1. In-depth defenses against corresponding attacks

In the worst case scenario, the attacker has complete information about the network and the strategy of the network administrator, and then the attacker can always find the most powerful attack strategy to maximize the total penalty. In the mean time, the network administrator also has complete information about the strategy of the attacker. In response to the attack, hence, the network administrator can adjust his/her strategy to minimize the total penalty. The phenomenon is like a battle between the network administrator and the attacker, and it is dynamic until the network administrator finds an optimal solution to minimize the maximized total penalty.

2.2 Problem Formulation of the NPDRAS Problem

In order to formulate the problem conveniently, we summarize some key points of problem assumptions and problem descriptions as **Table 2-1** and **Table 2-2** show respectively. Furthermore, we denominate the problem as a Network Planning and Defense Resources Allocation Strategy (NPDRAS) problem.

Table 2-1. Problem Assumptions of the NPDRAS Problem

Problem Assumptions
● All nodes have video encoding and transmission technologies such as a progress coder or video gateway.
● Paths which are chosen for connecting the source to a destination in a multicast group are dis-joint paths in terms of link.
● Both the network administrator and the attacker have complete information.
● Both the network administrator and the attacker have budget limitations.
● The objective of the attacker is to maximize the total penalty caused by QoS violations in terms of delay by deciding which nodes to attack and allocating attack budget effectively.
● The objective of the network administrator is to minimize the total penalty caused by the attacker by choosing which links to set and allocating defense budget appropriately.
● Only nodal attacks are considered. (No link attacks are considered.)
● Only malicious attacks are considered. (No random errors are considered.)
● A node is only subject to attack if a path exists from attacker's source to that node, and all the intermediate nodes on the path have been compromised.
● A node is compromised if the attack budget applied to the node is equal to or greater than the defense capability of the node.
● The attacker can apply extra attack budget to degrade the nodal capacity only if the node is compromised.

Table 2-2. Problem Descriptions of the NPDRAS Problem

Problem Descriptions
Given:
● A set of nodes in the AS
● A set of feasible links in the AS
● A set of multicast groups
● The requirements of traffic, end-to-end delay, and multiple paths for each destination of each multicast group
● The implementation cost of each feasible link
● The defense capability function of each node
● The delay function of each feasible link
● The penalty function of each destination of each multicast group
● The total defense budget of the network administrator
● The total attack budget of the attacker
Objective:
● To minimize the maximized total penalty caused by QoS violations in terms of delay.
Subject to:
● Routing constraints
● Capacity constraints
● Delay constraints
● Multiple paths constraints
● Attack budget constraints
● Defense budget constraints
To Determine:
● Network administrator:
✓ Which links to set and their capacity
✓ The defense budget allocation strategy
● Attacker:
✓ Which nodes to attack and which paths to reach the nodes
✓ The amount of attack budget allocated to each compromised node to degrade the nodal capacity

We first convert the AS to a directed graph and all domains are depicted as nodes where no link between any two nodes as **Figure 2-2** shows. As the topology is generated by the network administrator, the attacker could entry the AS by artificial links to entry nodes as **Figure 2-3** shows. In order to measure the nodal capacity, we use the node splitting technology which splits a node into two dummy nodes and generates an artificial link between them. For example, **Figure 2-4** is converted from **Figure 2-3** using node splitting technology. Later we propose a mathematical model to formulate the mutual behavior and solve it by our proposed solution approaches. It is a min-max problem where the inner problem is the attacker perspective and the outer problem is the network administrator perspective.

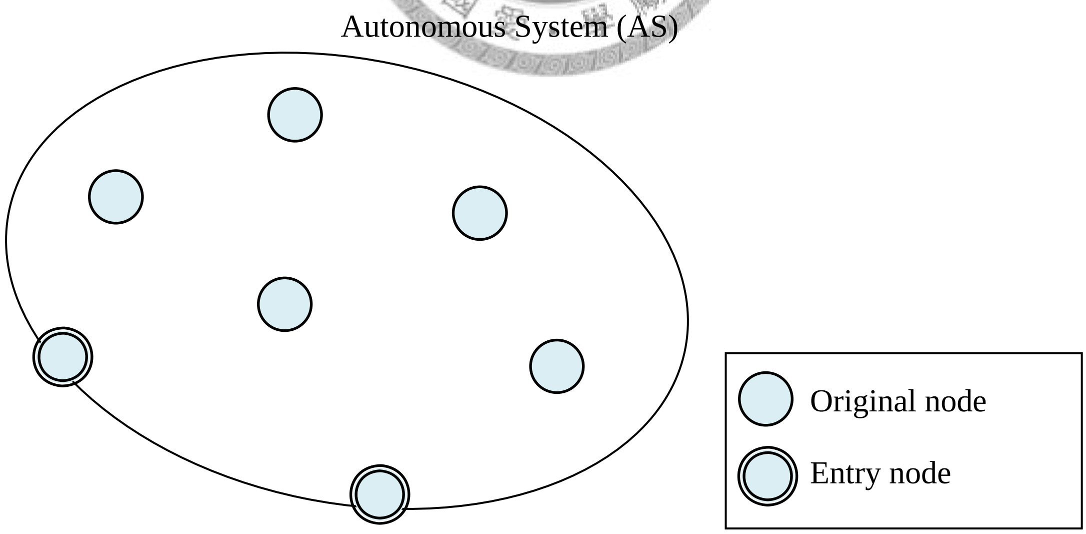


Figure 2-2. Graph of the Autonomous System (AS)

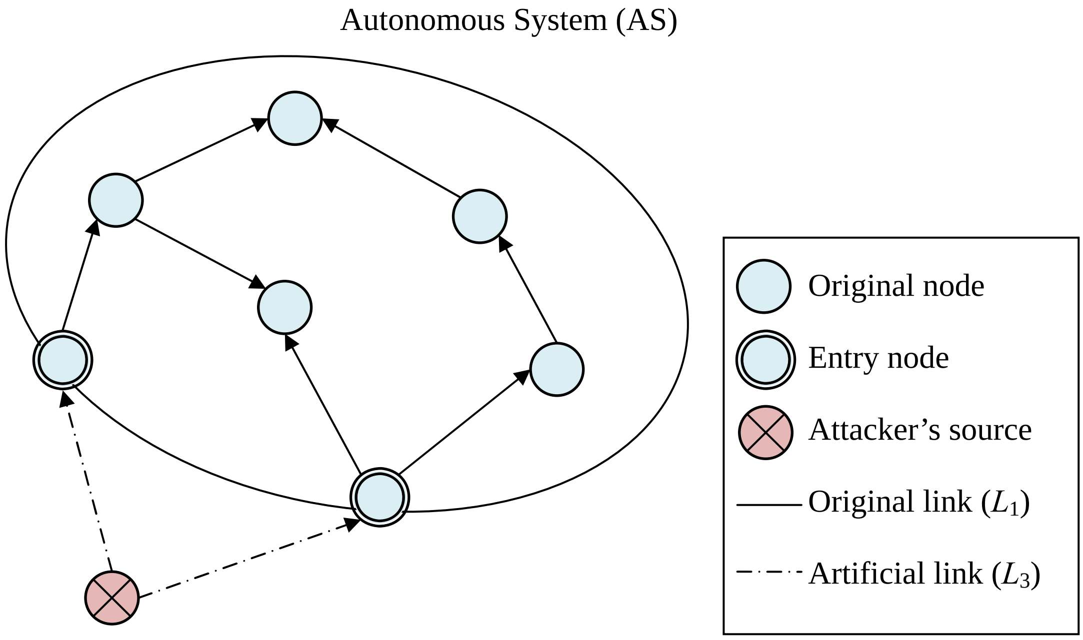


Figure 2-3. An Attack Scenario

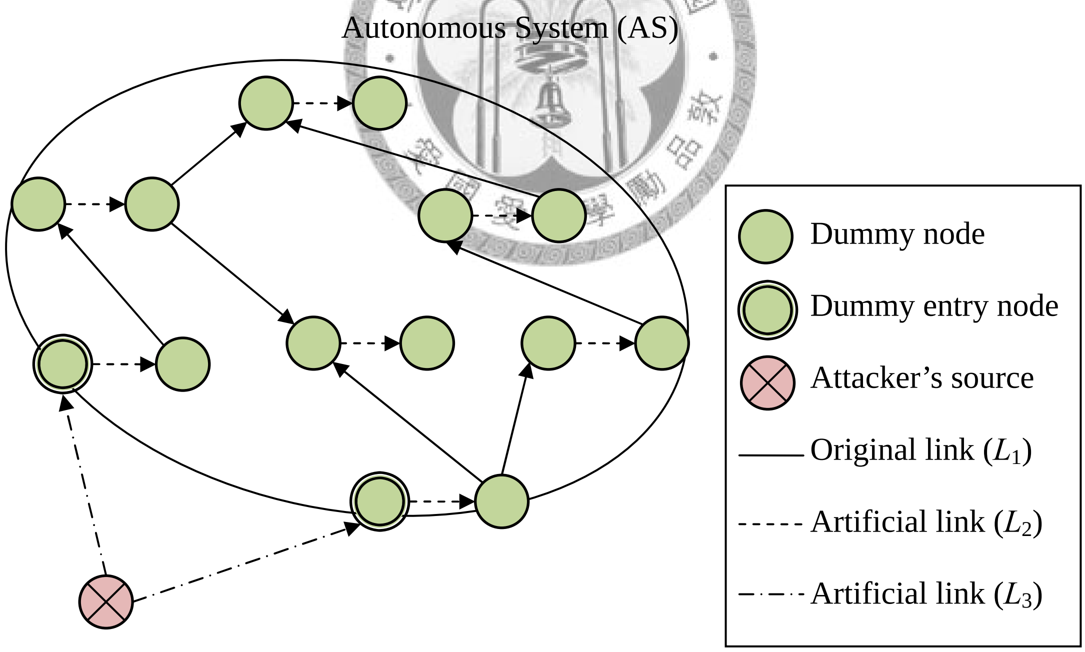


Figure 2-4. An Attack Scenario with Node Splitting

The given parameters and the decision variables used in the NPDRAS problem are defined in **Table 2-3** and **Table 2-4** respectively.

Table 2-3. Given Parameters of the NPDRAS Problem

Given Parameters	
Notation	**Description**
N	The index set of all nodes
L	The index set of all links, $L = L_1 \cup L_2 \cup L_3$
L_1	The index set of all candidate links
L_2	The index set of all artificial links which are original nodes
L_3	The index set of all artificial links from attacker's source node not in the AS to the entry nodes of AS
G	The index set of all multicast groups
D_g	The index set of all destinations of multicast group g, where $g\in G$
R_{gd}	The index set of all candidate paths which destination d of multicast group g may use, where $d\in D_g$, $g\in G$
σ_{rl}	The indicator function, which is 1 if link l is on path r, and 0 otherwise (where $l\in L$, $r\in R_{gd}$)
α_{gd}	The delay requirement of the destination d of multicast group g, where $d\in D_g$, $g\in G$
β_{gd}	The traffic requirement of the destination d of multicast group g, where $d\in D_g$, $g\in G$
γ_{gd}	The multiple paths requirement of the destination d of multicast group g, where $d\in D_g$, $g\in G$
Uh_{gd}	The maximum allowable end-to-end delay of the destination d of multicast group g, where $d\in D_g$, $g\in G$
W	The index set of all Origin-Destination (O-D) pairs for attack
P_w	The index set of all candidate paths for O-D pair w, where $w\in W$
δ_{pl}	The indicator function, which is 1 if link l is on path p, and 0 otherwise (where $l\in L$, $p\in P_w$)
A	The total attack budget of the attacker
A_l^c	All possible value of a_l^c, where $l\in L_2$
B	The total defense budget of the network administrator
s_l	The implementation cost of link l, where $l\in L_1$

Table 2-4. Decision Variables of the NPDRAS Problem

Decision Variables	
Notation	**Description**
v_{gdr}	1 if path r is selected to transmit for group g and destined at destination d and 0 otherwise, where $g\in G$, $d\in D_g$, $r\in R_{gd}$
m_{gl}	The maximum traffic requirement of destinations in multicast group g that are connected from the source through link l, where $g\in G$, $l\in L$
M_l	The aggregate traffic flow on link l, where $l\in L$
z_l	1 if link l is selected to implement, and 0 otherwise (where $l\in L$)
b_l^t	The budget allocated to link l to enhance the link's defense capability, where $l\in L_2$
b_l^c	The budget allocated to link l to enhance the link capacity, where $l\in L$
$\hat{a}_l^t(b_l^t)$	The threshold of the attack cost leading to a successful attack, where $l\in L_2$
a_l^t	The attack budget allocated to link l to compromise the link, where $l\in L_2$
a_l^c	The attack budget allocated to link l to degrade the link capacity, where $l\in L_2$
$c_l(a_l^c, b_l^c)$	The capacity of link l, where $l\in L$
$t_l(c_l, M_l)$	The traffic delay of link l, where $l\in L$
h_{gdr}	The end-to-end delay of the destination d of multicast group g in path r, where $g\in G$, $d\in D_g$, $r\in R_{gd}$
Lh_{gd}	The lower bound of end-to-end delay of the destination d of multicast group g, where $d\in D_g$, $g\in G$
θ_l	The maximum allowable link delay for link l
$p_{gd}(h_{gdr}, \alpha_{gd})$	The delay penalty of the destination d of multicast group g in path r, where $g\in G$, $d\in D_g$, $r\in R_{gd}$
x_p	1 if path p is selected as the attack path, and 0 otherwise (where $p\in P_w$)
y_l	1 if link l is attacked, and 0 otherwise (where $l\in L_2$)

The NPDRAS problem is then formulated as the following problem **(IP 1)**.

Objective function:

$$Z_{IP1} = \min_{z_l, b_l^t, b_l^c} \max_{x_p, y_l, a_l^t, a_l^c} \sum_{g\in G} \sum_{d\in D_g} \sum_{r\in R_{gd}} v_{gdr}\, p_{gd}(h_{gdr}, \alpha_{gd}) \qquad \textbf{(IP 1)}$$

Subject to:

$$v_{gdr} \beta_{gd} \sigma_{rl} \le m_{gl} \qquad \forall\, g\in G,\ d\in D_g,\ l\in L \qquad \text{(IP 1.1)}$$

$$M_l = \sum_{g\in G} m_{gl} \qquad \forall\, l\in L \qquad \text{(IP 1.2)}$$

$$M_l \le c_l(a_l^c, b_l^c) \qquad \forall\, l\in L \qquad \text{(IP 1.3)}$$

$$0 \le m_{gl} \le \max_{d\in D_g} \beta_{gd} \qquad \forall\, g\in G,\ l\in L \qquad \text{(IP 1.4)}$$

$$\sum_{r\in R_{gd}} v_{gdr} \sigma_{rl} \le z_l \qquad \forall\, g\in G,\ d\in D_g,\ l\in L_1 \qquad \text{(IP 1.5)}$$

$$\sum_{r\in R_{gd}} v_{gdr} = \gamma_{gd} \qquad \forall\, g\in G,\ d\in D_g \qquad \text{(IP 1.6)}$$

$$z_l = 0 \text{ or } 1 \qquad \forall\, l\in L_1 \qquad \text{(IP 1.7)}$$

$$v_{gdr} = 0 \text{ or } 1 \qquad \forall\, g\in G,\ d\in D_g,\ r\in R_{gd} \qquad \text{(IP 1.8)}$$

$$\sum_{l\in L} t_l(c_l(a_l^c, b_l^c), M_l) v_{gdr} \sigma_{rl} = h_{gdr} \qquad \forall\, g\in G,\ r\in R_{gd},\ d\in D_g \qquad \text{(IP 1.9)}$$

$$Lh_{gd} \le h_{gdr} \le Uh_{gd} \qquad \forall\, g\in G,\ r\in R_{gd},\ d\in D_g \qquad \text{(IP 1.10)}$$

$$t_l(c_l(a_l^c, b_l^c), M_l) \le \theta_l \qquad \forall\, l\in L_2 \qquad \text{(IP 1.11)}$$

$$\sum_{l\in L_2} b_l^t + \sum_{l\in L} (b_l^c + z_l s_l) \le B \qquad \text{(IP 1.12)}$$

$$0 \le b_l^t \le B \qquad \forall\, l\in L_2 \qquad \text{(IP 1.13)}$$

$$0 \le b_l^c \le B \qquad \forall\, l\in L \qquad \text{(IP 1.14)}$$

$$\sum_{l\in L_2} a_l^t + \sum_{l\in L_2} a_l^c \le A \qquad \text{(IP 1.15)}$$

$$0 \le a_l^t \le \hat{a}_l^t(b_l^t) \qquad \forall\, l \in L_2 \qquad \text{(IP 1.16)}$$

$$\hat{a}_l^t(b_l^t)\, y_l \le a_l^t \qquad \forall\, l \in L_2 \qquad \text{(IP 1.17)}$$

$$\min\{A_l^c\} \le a_l^c \le \max\{A_l^c\} \qquad \forall\, l \in L_2 \qquad \text{(IP 1.18)}$$

$$a_l^c \in A_l^c \qquad \forall\, l \in L_2 \qquad \text{(IP 1.19)}$$

$$a_l^c \le y_l A \qquad \forall\, l \in L_2 \qquad \text{(IP 1.20)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le z_l \qquad \forall\, l \in L_1,\ w \in W \qquad \text{(IP 1.21)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le y_l \qquad \forall\, l \in L_2,\ w \in W \qquad \text{(IP 1.22)}$$

$$\sum_{p \in P_w} x_p = y_l \qquad \forall\, l \in L_2,\ w = (s, l) \qquad \text{(IP 1.23)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall\, w \in W \qquad \text{(IP 1.24)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w,\ w \in W \qquad \text{(IP 1.25)}$$

$$y_l = 0 \text{ or } 1 \qquad \forall\, l \in L_2. \qquad \text{(IP 1.26)}$$

<u>Explanation of the mathematical formulations</u>:

- **Objective Function**: The objective is to minimize the maximized total penalty caused by QoS violations in terms of delay. In the inner problem, an attacker would like to maximize the total penalty by deciding which artificial links to attack and allocating attack budget effectively. In outer problem, the network administrator would like to minimize the penalty caused by the attacker by choosing which original links to set and allocating defense resources appropriately.

- **Constraints (IP 1.1) ~ (IP 1.4)** represent the capacity constraints. In **Constraint (IP 1.1)**, m_{gl} can be interpreted as the "estimate" of the aggregate flows for multicast group g on link l. **Constraint (IP 1.2)** denotes that M_l refers to the total aggregate flows for all groups on link l. **Constraint (IP 1.3)** limits the total aggregate flows on a link does not exceed its capacity. The capacity of a link is a function of two parameters, which are the attack budget for degradation applied to the link by an attacker and the budget for enhancement allocated to the link by a network administrator. **Constraint (IP 1.4)** is a redundant constraint, which provides upper bound and lower bound on the maximum traffic requirement for multicast group g on link l.
- **Constraint (IP 1.5)** enforces that if a path is chosen for transmission for an Origin-Destination pair (O-D pair), all original links on the path have to be set
- **Constraint (IP 1.6)** requires that the amount of connection for each O-D pair has to satisfy its corresponding QoS requirement.
- **Constraints (IP 1.7)** and **(IP1.8)** limit the value of z_l and v_{gdr} to 0 or 1. Therefore, **Constraints (IP 1.5)** and **(IP 1.7)** jointly require that an original link has to be chosen once at most for one multicast group.
- **Constraint (IP 1.9)** denotes that the end-to-end delay of the transmission of an O-D pair is the sum of the traffic delay of all links on the path. The traffic delay of

a link is a function of two parameters, which are the capacity and the total aggregate flows of the link.

- **Constraint (IP 1.10)** restricts that the end-to-end delay has to be between the lower bound and upper bound. It is noted that the Lh_{gd} value is the basic delay calculated from v_{gdr}.
- **Constraint (IP 1.11)** restricts that the link delay has to be smaller than or equal to upper bound. It is noted that the θ_l value is calculated from v_{gdr} and Lh_{gd}.
- **Constraint (IP 1.12)** restricts that the total allocated budget, including the budget for enhancing an artificial link's defense capability, for enhancing the capacity of a link, and for setting a link, has not to exceed the total budget of the network administrator.
- **Constraint (IP 1.13)** restricts that the defense budget for enhancing an artificial link's defense capability has to be nonnegative and not exceed the total budget of the network administrator.
- **Constraint (IP 1.14)** restricts that the budget for enhancing the capacity of a link has to be nonnegative and not exceed the total budget of the network administrator and be nonnegative.
- **Constraint (IP 1.15)** restricts that the total allocated attack budget, including the attack budget for compromising an artificial link and for degrading the capacity of

an artificial link, has not to exceed the total attack budget of an attacker.

- **Constraint (IP 1.16)** restricts that the attack budget for compromising an artificial link has to be nonnegative and not exceed the link's defense capability because it would be a waste of budget.
- **Constraint (IP 1.17)** enforces that if an artificial link is compromised, the attack budget for compromising the link has to equal to or greater than the link's defense capability.
- **Constraints (IP 1.18)** and **(IP 1.19)** restricts that the attack budget for degrading the capacity of an artificial link has to be chosen from the set A_l^c .
- **Constraint (IP 1.20)** enforces that the attack budget for degrading the capacity of an artificial link is applied only if the link is compromised.
- **Constraint (IP 1.21)** enforces that an original link is chosen for an attack path only if the link is set.
- **Constraint (IP 1.22)** requires that all artificial links on an attack path are compromised.
- **Constraint (IP 1.23)** enforces that if an artificial link is chosen for attack, the attacker has to find a path from the source to the targeted link.
- **Constraint (IP 1.24)** enforces that if an artificial link is chosen for attack, the attack path for it has to be only one.

- **Constraints (IP 1.25)** and **(IP 1.26)** limit the value of x_p and y_l to 0 or 1.

2.3 Problem Formulation of the ARRAS Problem

In order to solve the NPDRAS problem, we first try to analyze the inner problem of the NPDRAS problem, that is, the Attack Routing and Resource Allocation Strategy (ARRAS) problem. The ARRAS problem is to predict the future action of the attacker. In another words, in the ARRAS problem, we assume that the network administrator's strategy is given and find the corresponding optimal strategy of the attacker. The result of ARRAS problem is used as an input to adjust the strategy of network administrator in NPDRAS problem and finally generate a best strategy for the network administrator against the attacker.

The assumptions of the ARRAS problem are the same as those of the NPDRAS problem. The given parameters and the decision variables of the APRAS problem are defined in **Table 2-5** and **Table 2-6** respectively.

Table 2-5. Given Parameters of the ARRAS Problem

Given Parameters	
Notation	**Description**
N	The index set of all nodes
L	The index set of all links, $L = L_1 \cup L_2 \cup L_3$
L_1	The index set of all candidate links
L_2	The index set of all artificial links which are original nodes
L_3	The index set of all artificial links from attacker's source node not in the AS to the entry nodes of AS
G	The index set of all multicast groups
D_g	The index set of all destinations of multicast group g, where $g \in G$
R_{gd}	The index set of all candidate paths which destination d of multicast group g may use, where $d \in D_g$, $g \in G$
σ_{rl}	The indicator function, which is 1 if link l is on path r, and 0 otherwise (where $l \in L$, $r \in R_{gd}$)
α_{gd}	The delay requirement of the destination d of multicast group g, where $d \in D_g$, $g \in G$
Lh_{gd}	The lower bound of end-to-end delay of the destination d of multicast group g, where $d \in D_g$, $g \in G$
Uh_{gd}	The maximum allowable end-to-end delay of the destination d of multicast group g, where $d \in D_g$, $g \in G$
θ_l	The maximum allowable link delay for link l
W	The index set of all Origin-Destination (O-D) pairs for attack
P_w	The index set of all candidate paths for O-D pair w, where $w \in W$
δ_{pl}	The indicator function, which is 1 if link l is on path p, and 0 otherwise (where $l \in L$, $p \in P_w$)
A	The total attack budget of the attacker
A_l^c	All possible value of a_l^c, where $l \in L_2$
v_{gdr}	1 if path r is selected to transmit for group g and destined at destination d and 0 otherwise, where $g \in G$, $d \in D_g$, $r \in R_{gd}$
M_l	The aggregate traffic flow on link l, where $l \in L$
z_l	1 if link l is selected to implement, and 0 otherwise (where $l \in L$)
b_l^c	The budget allocated to link l to enhance the link capacity, where $l \in L$
$\hat{a}_l^t(b_l^t)$	The threshold of the attack cost leading to a successful attack, where $l \in L_2$

Table 2-6. Decision Variables of the ARRAS Problem

Decision Variables	
Notation	**Description**
a_l^t	The attack budget allocated to link l to compromise the link, where $l \in L_2$
a_l^c	The attack budget allocated to link l to degrade the link capacity, where $l \in L_2$
$c_l(a_l^c, b_l^c)$	The capacity of link l, where $l \in L$
$t_l(c_l, M_l)$	The traffic delay of link l, where $l \in L$
h_{gdr}	The end-to-end delay of the destination d of multicast group g in path r, where $g \in G$, $d \in D_g$, $r \in R_{gd}$
$p_{gd}(h_{gdr}, \alpha_{gd})$	The delay penalty of the destination d of multicast group g in path r, where $g \in G$, $d \in D_g$, $r \in R_{gd}$
x_p	1 if path p is selected as the attack path, and 0 otherwise (where $p \in P_w$)
y_l	1 if link l is attacked, and 0 otherwise (where $l \in L_2$)

The ARRAS problem is formulated as the following problem **(IP 2)**.

Objective function:

$$Z_{IP2} = \max_{x_p, y_l, a_l^t, a_l^c} \sum_{g \in G} \sum_{d \in D_g} \sum_{r \in R_{gd}} v_{gdr} p_{gd}(h_{gdr}, \alpha_{gd}) = -\min_{x_p, y_l, a_l^t, a_l^c} \sum_{g \in G} \sum_{d \in D_g} \sum_{r \in R_{gd}} v_{gdr} p_{gd}(h_{gdr}, \alpha_{gd}) \qquad \textbf{(IP 2)}$$

Subject to:

$$M_l \le c_l(a_l^c, b_l^c) \qquad \forall\, l \in L \qquad \text{(IP 2.1)}$$

$$\sum_{l \in L} t_l(c_l(a_l^c, b_l^c), M_l) v_{gdr} \sigma_{rl} = h_{gdr} \qquad \forall\, g \in G, r \in R_{gd}, d \in D_g \qquad \text{(IP 2.2)}$$

$$Lh_{gd} \le h_{gdr} \le Uh_{gd} \qquad \forall\, g \in G, r \in R_{gd}, d \in D_g \qquad \text{(IP 2.3)}$$

$$t_l(c_l(a_l^c, b_l^c), M_l) \le \theta_l \qquad \forall\, l \in L_2 \qquad \text{(IP 2.4)}$$

$$\sum_{l \in L_2} a_l^t + \sum_{l \in L_2} a_l^c \le A \qquad \text{(IP 2.5)}$$

$$0 \le a_l^t \le \hat{a}_l^t(b_l^t) \qquad \forall\, l \in L_2 \qquad \text{(IP 2.6)}$$

$$\hat{a}_l^t(b_l^t)\, y_l \le a_l^t \qquad \forall\, l \in L_2 \qquad \text{(IP 2.7)}$$

$$\min\{A_l^c\} \le a_l^c \le \max\{A_l^c\} \qquad \forall\, l \in L_2 \qquad \text{(IP 2.8)}$$

$$a_l^c \in A_l^c \qquad \forall\, l \in L_2 \qquad \text{(IP 2.9)}$$

$$a_l^c \le y_l A \qquad \forall\, l \in L_2 \qquad \text{(IP 2.10)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le z_l \qquad \forall\, l \in L_1,\ w \in W \qquad \text{(IP 2.11)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le y_l \qquad \forall\, l \in L_2,\ w \in W \qquad \text{(IP 2.12)}$$

$$\sum_{p \in P_w} x_p = y_l \qquad \forall\, l \in L_2,\ w = (s, l) \qquad \text{(IP 2.13)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall\, w \in W \qquad \text{(IP 2.14)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w,\ w \in W \qquad \text{(IP 2.15)}$$

$$y_l = 0 \text{ or } 1 \qquad \forall\, l \in L_2. \qquad \text{(IP 2.16)}$$

Explanation of the mathematical formulations:

- **Objective Function**: The objective function is to maximize the total penalty caused by QoS Violations in terms of delay by deciding which artificial links to attack and allocating attack budget effectively. The objective function is also the inner problem of the NPDRAS problem. For convenience, we transform **(IP 2)** from a maximization problem into an equivalent minimization problem and does not affect the problem structure or the optimality conditions

- **Constraints (IP 2.1)**, **(IP 2.2), (IP 2.3)** and **(IP 2.4)** are equal to **Constraints (IP 1.3)**, **(IP 1.9), (IP 1.10)** and **(IP 1.11).**
- **Constraints (IP 2.5)** ~ **(IP 2.16)** are the same to **Constraints (IP 1.15)** ~ **(IP 1.26)**.

Chapter 3 Solution Approach

3.1 Lagrangean Relaxation Method

There are a lot of researches on the Lagrangean relaxation method after 1970s [18][19]. It is one of the most useful methodologies to solve large-scale mathematical programming applications including linear, dynamic, and integer programming nowadays. The concept of the method comes from the observation that a complicated programming problem can be sighted as a related easily-solved problem with side constraints. Because of its reduction of complexity and excellent performance for solving a difficult programming problem, we exploit the Lagrangean relaxation method to solve the ARRAS problem proposed in **Chapter 2**.

The basic idea of the Lagrangean relaxation method is shown in **Figure 3-1**. First, some constraints are removed and added into the objective function with corresponding Lagrangean multipliers in order to convert the primal problem to an easily-solved form, which is called the Lagrangean relaxation (LR) problem. Then we can use the decomposition technique to disintegrate the LR problem into several independent subproblems which can be solved optimally. By solving the LR problem, we can obtain a lower bound (LB) of the optimal value for the original minimization problem. Furthermore, for the sake of getting the best solution, we use the subgradient optimization technique which is one of the cited Lagrangean dual problems to derive the tightest LB by adjusting the Lagrangean multipliers.

From resolving the LR problem, besides, we could obtain some useful information for designing some proper heuristic approaches to get the feasible solutions of the primal problem, which is also the upper bound (UB) of the optimal value. Clearly, the optimal solution of the primal problem is guaranteed to be between the LB and the UB. The detail procedure of Lagrangean relaxation method is shown in **Figure 3-2**.

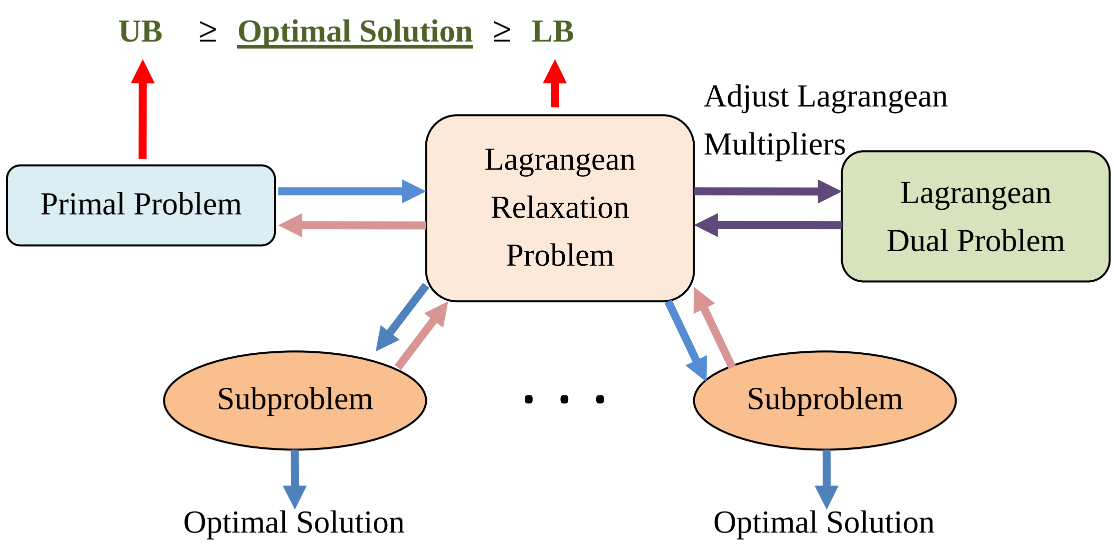


Figure 3-1. Idea of Lagrangean Relaxation Method

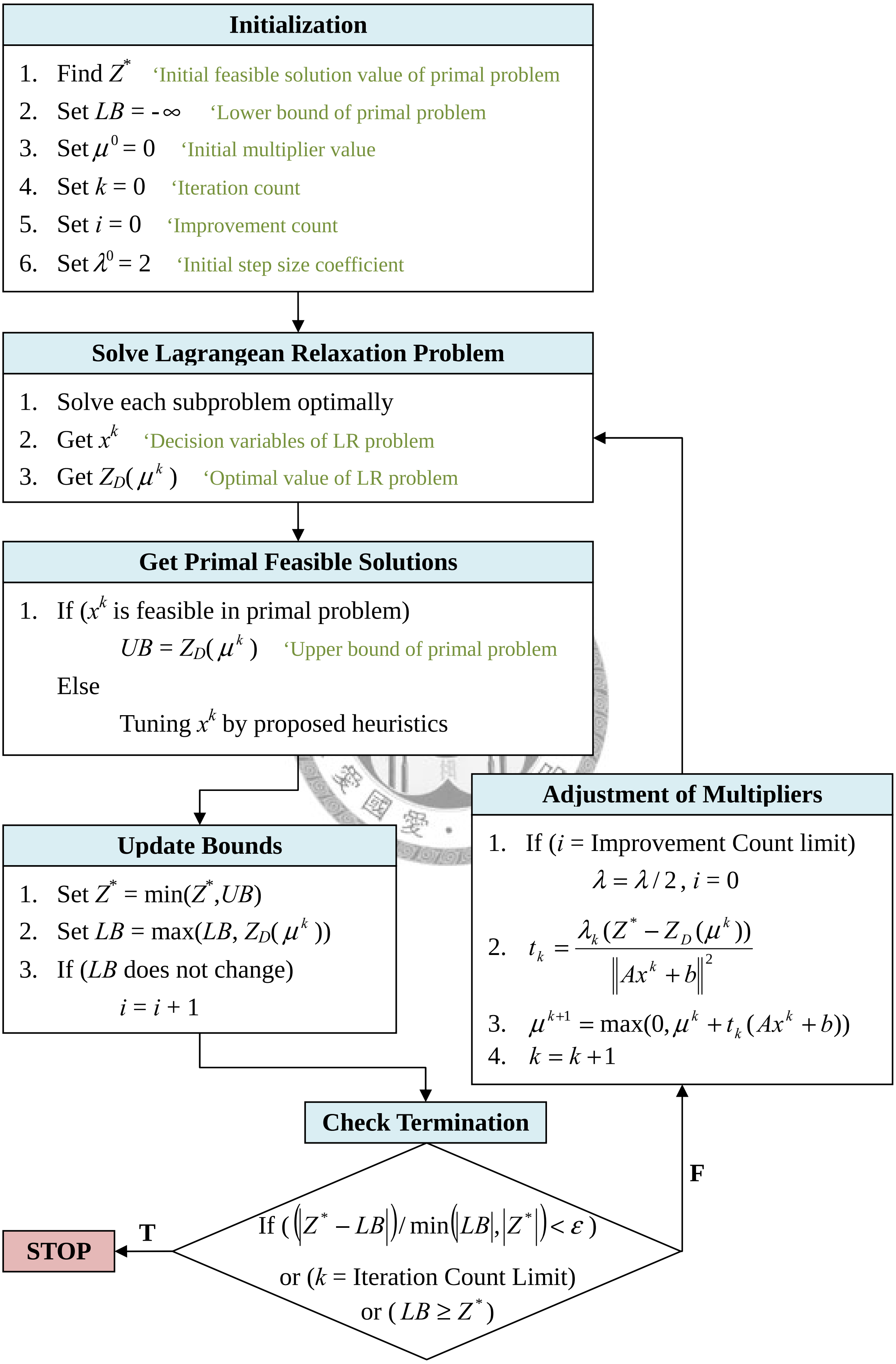


Figure 3-2. Detail Procedure of Lagrangean Relaxation Method

3.2 The Solution Approach for the ARRAS Problem

We relax **Constraints (IP 2.2)**, **(IP 2.5)**, **(IP 2.12)**, and **(IP 2.13)** with associated Lagrangean multipliers to add into the objective function of **(IP 2)** and thus the Lagrangean relaxation problem **(LR 1)** can be obtained.

3.2.1 Lagrangean Relaxation

Optimization Problem (LR):

$$
\begin{aligned}
&Z_D(\mu_1,\mu_2,\mu_3,\mu_4)\\
&=\min -\sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}} v_{gdr}\, p_{gd}(h_{gdr},\alpha_{gd})\\
&+\sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}} \mu^1_{gdr}\Big[\sum_{l\in L} t_l(c_l(a_l^c,b_l^c),M_l)\, v_{gdr}\sigma_{rl}-h_{gdr}\Big]+\mu^2\Big[\Big(\sum_{l\in L_2} a_l^t+\sum_{l\in L_2} a_l^c\Big)-A\Big]\\
&+\sum_{w\in W}\sum_{l\in L_2}\mu^3_{wl}\Big(\sum_{p\in P_w} x_p\delta_{pl}-y_l\Big)+\sum_{l\in L_2}\mu^4_l\Big(\sum_{p\in P_{(s,t)}} x_p-y_l\Big)
\end{aligned}
$$

(LR 1)

Subject to:

$$M_l \le c_l(a_l^c,b_l^c) \qquad \forall\, l\in L \qquad \text{(LR 1.1)}$$

$$v_{gdr}Lh_{gd} \le h_{gdr} \le v_{gdr}Uh_{gd} \qquad \forall\, g\in G,\ r\in R_{gd},\ d\in D_g \qquad \text{(LR 1.2)}$$

$$t_l(c_l(a_l^c,b_l^c),M_l) \le \theta_l \qquad \forall\, l\in L_2 \qquad \text{(LR 1.3)}$$

$$0 \le a_l^t \le \hat{a}_l^t(b_l^t) \qquad \forall\, l\in L_2 \qquad \text{(LR 1.4)}$$

$$\hat{a}_l^t(b_l^t)\, y_l \le a_l^t \qquad \forall\, l\in L_2 \qquad \text{(LR 1.5)}$$

$$\min\{A_l^c\} \le a_l^c \le \max\{A_l^c\} \qquad \forall\, l\in L_2 \qquad \text{(LR 1.6)}$$

$$a_l^c \in A_l^c \qquad \forall\, l\in L_2 \qquad \text{(LR 1.7)}$$

$$a_l^c \le y_l A \qquad \forall l \in L_2 \qquad \text{(LR 1.8)}$$

$$\sum_{p \in P_w} x_p \delta_{pl} \le z_l \qquad \forall l \in L_1, w \in W \qquad \text{(LR 1.9)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(LR 1.10)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W \qquad \text{(LR 1.11)}$$

$$y_l = 0 \text{ or } 1 \qquad \forall l \in L_2. \qquad \text{(LR 1.12)}$$

Among Lagrangean multipliers, μ_1 and μ_4 are unrestricted variable where μ_1 is a three-dimensional vector and μ_4 is a one-dimensional vector. Besides, μ_2 and μ_3 are non-negative variables where μ_3 is two-dimensional vectors.

We then decompose **(LR 1)** into three independent optimization subproblems which are easy-solved as follows.

Subproblem 1: (related to decision variable x_p)

$$Z_{Sub1}(\mu_3, \mu_4) = \min \sum_{w \in W} \sum_{l \in L_2} \sum_{p \in P_w} \mu_{wl}^3 x_p \delta_{pl} + \sum_{l \in L_2} \sum_{p \in P_{(s,l)}} \mu_l^4 x_p \qquad \textbf{(Sub 1)}$$

Subject to:

$$\sum_{p \in P_w} x_p \delta_{pl} \le z_l \qquad \forall l \in L_1, w \in W \qquad \text{(Sub 1.1)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(Sub 1.2)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall p \in P_w, w \in W. \qquad \text{(Sub 1.3)}$$

In the problem, because **Constraint (Sub 1.2)** enforces only one path to be chosen for an O-D pair, we can transform $\sum_{l \in L_2} \sum_{p \in P_{(s,l)}} \mu_l^4 x_p$ into $\sum_{w \in W} \sum_{p \in P_w} \mu_l^4 x_p + \sum_{p \in P_{(s,s)}} \mu_s^4 x_p$. However, no path starts and ends at the same artificial link, so $\sum_{p \in P_{(s,s)}} \mu_s^4 x_p$ can be ignored. Then we can further decomposed the problem into $|W|$ independent subproblems and one for each O-D pair $w \in W$ as follows.

Subproblem 1': (related to decision variable x_p)

$$Z_{Sub1}(\mu_3, \mu_4) = \min \sum_{p \in P_w} (\sum_{j \in L_2} \mu_{wj}^3 \delta_{pj} + \mu_l^4) x_p \qquad \textbf{(Sub 1')}$$

Subject to:

$$\sum_{p \in P_w} x_p \delta_{pl} \le z_l \qquad \forall\, l \in L_1, w \in W \qquad \text{(Sub 1'.1)}$$

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall\, w \in W \qquad \text{(Sub 1'.2)}$$

$$x_p = 0 \text{ or } 1 \qquad \forall\, p \in P_w, w \in W. \qquad \text{(Sub 1'.3)}$$

The algorithm for solving **(Sub 1)** is described below.

Step 1: By using the values of μ_{wj}^3 as the arc weight of the corresponding artificial link respectively, we use Dijkstra's algorithm to find the shortest path for each O-D pair $w \in W$.

Step 2: For paths which are not chosen for any O-D pair, we assign zero to the corresponding x_p.

Step 3: For the path which is chosen for each O-D pair $w \in W$, we examine its total cost and the μ_l^4 value of its destination artificial link. We assign one to the corresponding x_p if the resulting value is non-positive, and zero otherwise.

The time complexity of Dijkstra's algorithm is $O(|L_2|^2)$. Therefore, the computational complexity of **(Sub 1)** is $O(|W|\times|L_2|^2)$.

Subproblem 2: (related to decision variable y_l, a_l^t, a_l^c)

(Sub 2)

$$
\begin{aligned}
&Z_{Sub2}(\mu_1,\mu_2,\mu_3,\mu_4)\\
&=\min\sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}}\mu_{gdr}^1\sum_{l\in L}t_l(c_l(a_l^c,b_l^c),M_l)v_{gdr}\sigma_{rl}+\mu^2(\sum_{l\in L_2}a_l^t+\sum_{l\in L_2}a_l^c)\\
&-\sum_{w\in W}\sum_{l\in L_2}\mu_{wl}^3 y_l-\sum_{l\in L_2}\mu_l^4 y_l\\
&=\min\sum_{l\in L}[\sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}}\mu_{gdr}^1 v_{gdr}\sigma_{rl}t_l(c_l(a_l^c,b_l^c),M_l)+\mu^2(a_l^t+a_l^c)-(\sum_{w\in W}\mu_{wl}^3+\mu^4)y_l]
\end{aligned}
$$

Subject to:

$$M_l \le c_l(a_l^c,b_l^c) \qquad \forall\, l\in L \qquad \text{(Sub 2.1)}$$

$$t_l(c_l(a_l^c,b_l^c),M_l)\le\theta_l \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.2)}$$

$$y_l = 0 \text{ or } 1 \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.3)}$$

$$0\le a_l^t \le \hat{a}_l^t(b_l^t) \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.4)}$$

$$\hat{a}_l^t(b_l^t)y_l \le a_l^t \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.5)}$$

$$0=\min\{A_l^c\}\le a_l^c\le\max\{A_l^c\} \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.6)}$$

$$a_l^c \in A_l^c \qquad \forall\, l\in L_2 \qquad \text{(Sub 2.7)}$$

$$a_l^c \le y_l A \qquad \forall\, l\in L_2. \qquad \text{(Sub 2.8)}$$

(Sub 2) can be further decomposed into |*L*| independent subproblems and one for each link. According to the constraints related to y_l, a_l^t, and a_l^c, we can conclude the relationship among them showed in **Table 3-1**.

Table 3-1. The Relationship among y_l, a_l^t, and a_l^c

y_l's value	a_l^t's value	a_l^c's value
0	$[0,\ \hat{a}_l^t(b_l^t)]$	0
1	$\hat{a}_l^t(b_l^t)$	$M_l \le c_l(a_l^c, b_l^c)$ and $0 \le a_l^c \le \max\{A_l^c\}$ and $a_l^c \in A_l^c$ and $t_l(c_l(a_l^c, b_l^c), M_l) \le \theta$

Since this is a minimization problem and the value of μ^2 is non-negative, the value of a_l^t has to be set to zero when the value of y_l is zero. For each subproblem, we can examine all the possible combinations of y_l, a_l^t, and a_l^c, and then obtain the optimal combination result among them to minimize the objective value.

The computational complexity of **(Sub 2)** is $O(|L| \times |A_l^c|)$.

Subproblem 3: (related to decision variable h_{gdr})

$$
\begin{aligned}
Z_{Sub5}(\mu_1) &= \min - \sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}} v_{gdr}\, p_{gd}(h_{gdr}, \alpha_{gd}) - \sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}} \mu_{gdr}^1 h_{gdr} \\
&= \max \sum_{g\in G}\sum_{d\in D_g}\sum_{r\in R_{gd}} [v_{gdr}\, p_{gd}(h_{gdr}, \alpha_{gd}) + \mu_{gdr}^1 h_{gdr}]
\end{aligned}
\quad \textbf{(Sub 3)}
$$

Subject to:

$$v_{gdr} Lh_{gd} \le h_{gdr} \le v_{gdr} Uh_{gd} \qquad \forall g \in G, r \in R_{gd}, d \in D_g. \qquad \text{(Sub 3.1)}$$

(Sub 3) can be decomposed into $|G|\times|D_g|\times|R_{gd}|$ independent subproblems and one for each path $r\in R_{gd}$. For each subproblem, we can solve it by the exhausted search of the value of h_{gdr}, and then find the optimal value of h_{gdr} to maximize the objective value.

The computational complexity of **(Sub 3)** is $O(|G|\times|D_g|\times|R_{gd}|\times|h_{gdr}|)$.

3.2.2 The Dual Problem and the Subgradient Method

According to the weak Lagrangean duality theorem [20], for any $\mu_2, \mu_3 \ge 0$, $Z_D(\mu_1, \mu_2, \mu_3, \mu_4)$ is a LB of Z_{IP2}. For obtaining the tightest LB, we construct the dual problem **(D 1)** and solve it by the subgradient method [18][19] as follows.

Dual Problem (D 1):

$$Z_D = \max Z_D(\mu_1, \mu_2, \mu_3, \mu_4) \qquad \textbf{(D 1)}$$

Subject to:

$$\mu_2, \mu_3 \ge 0. \qquad \text{(D 1.1)}$$

Let a vector s be a subgradient of $Z_D(\mu_1, \mu_2, \mu_3, \mu_4)$. Then, in iteration k of the subgradient optimization procedure, the multiplier vector $\mu^k = (\mu_1^k, \mu_2^k, \mu_3^k, \mu_4^k)$ is update by $\mu^{k+1} = \mu^k + t^k s^k$

where

$$s^k(u_1^k, u_2^k, u_3^k, u_4^k)$$
$$= (\sum_{l \in L} t_l(c_l(a_l^c, b_l^c), M_l) v_{gdr} \sigma_{rl} - h_{gdr}, \sum_{l \in L_2} (a_l^t + a_l^c) - A, \sum_{p \in P_w} x_p \delta_{pl} - y_l, \sum_{p \in P_{(s,l)}} x_p - y_l)$$

; and the step size t^k is determined by $t^k = \lambda \frac{Z_{IP2}^* - Z_D(\mu^k)}{\left\| s^k \right\|^2}$.

In this equation, Z_{IP2}^* is the tightest UB of the optimal value for the primal problem obtained by iteration *k* and λ is a constant where $0 \le \lambda \le 2$.

3.2.3 Getting Primal Feasible Solutions

If the solution to **(LR 1)** is not feasible to **(IP 2)**, we have to modify it to be a feasible primal solution by a getting primal feasible solutions' heuristic. To get a primal feasible solution for **(IP 2)**, the results obtained from the procedures of Lagrangean relaxation and the subgradient method may provide some useful hints. That is to say, the solution to **(LR 1)** and the Lagrangean multipliers gained from **(D 1)** are useful hints to the heuristic's design. The proposed heuristic for getting primal feasible solutions is shown in **Table 3-2** and described below.

The heuristic has two stages. In the first stage (**Step 1** to **Step 5**), we let each attack path whose x_p's value derived from **(Sub 1)** is equal to one as the candidate attack path. We then assign each candidate attack path a weight, $\min_{a_l^c} \frac{\overline{\hat{a}_l^t(b_l^t)} + a_l^c + |N| u_l^4}{P_l}$, where the artificial link *l* of a_l^c, u_l^4, and P_l means the target node of the candidate attack path, i.e. the terminal node of the candidate attack path from the attack source node. $\overline{\hat{a}_l^t(b_l^t)}$ represents the attack budget allocated to compromise all un-compromised nodes on the candidate attack path in order to reach the target node and then attack its capacity. $|N|u_l^4$ reflects the punishment of inconsistency between the values of x_p and y_l, where the target node is compromised but there is no attack path to it. The value of a_l^c is the attack budget allocated to attack the target nodal capacity and can be tuned to minimize the weight using the feasible quota which is the remainder of attack budget minus $\overline{\hat{a}_l^t(b_l^t)}$. P_l is the total penalty caused by a_l^c. The weight's concept shows mainly the ratio of the attack cost to the penalty gained. It is remarkable to address that the less the weight of a candidate attack path is, the more the effectiveness for attack is. Moreover, each path whose $\overline{\hat{a}_l^t(b_l^t)}$ is greater than the remainder of attack budget is removed from candidate attack paths because the attacker can't afford to compromise the target node.

After assigning the weight of each candidate attack path, we select the one with the smallest weight among them to attack. We then move it away from candidate attack paths and re-calculate the weight of each candidate attack path again. The steps are continued until there is no candidate attack path, and then an attack subtree is generated.

If there is excess attack budget yet, the second stage is performed (**Step 6** to **Step 12**). We use $\hat{a}_l^t(b_l^t)$ as each nodal cost and apply Dijkstra's algorithm to determine the minimal cost from the attack subtree to the target node of each un-attacked path. The paths obtained from Dijkstra's algorithm are considered as candidate attack paths. We can calculate the weight of each candidate attack path, remove the paths which the attacker can't afford to compromise the target node, and select the one of the smallest weight to attack, which is the same procedure to the **Step 2** to **Step 4** of the first stage. We then remove the attacked path from candidate attack paths, re-apply Dijkstra's algorithm, and re-calculate the weight of each candidate attack path again. The steps are repeated until there is no candidate attack path, and then a final attack tree is generated.

The computational complexity of this heuristic is $O(|L|^3 + |L|^2|A_l^c|)$.

Table 3-2. The Proposed Heuristic for getting primal feasible solutions

Step 1.	Let each attack path whose x_p's value is equal to one as the candidate attack path.
Step 2.	Use $\min\limits_{a_l^c} \frac{\overline{\hat{a}_l^t(b_l^t)} + a_l^c + \lvert N \rvert u_l^4}{P_l}$ as each candidate attack path's weight, where the artificial link l of $\overline{\hat{a}_l^t(b_l^t)}$, a_l^c, u_l^4, and P_l is the target node of the candidate attack path, $\overline{\hat{a}_l^t(b_l^t)}$ is the total compromise cost from the attack source to the target node, P_l is the caused penalty, and the value of a_l^c can be tuned to minimize this weight.
Step 3.	Remove each candidate attack path whose $\overline{\hat{a}_l^t(b_l^t)}$ is greater than the remainder of attack budget.

Step 4. Choose the candidate attack path with the smallest weight to attack.

Step 5. Remove the attacked path and return to **Step 2** until there is no candidate attack path.

Step 6. If there is no excess attack budget, go to **Step 12**; otherwise go to **Step 7**.

Step 7. Use $\hat{a}_l^t(b_l^t)$ as each nodal cost and apply Dijkstra's algorithm to determine the minimal compromise cost from the attack subtree to the target node of each un-attacked path and the paths obtained from Dijkstra's algorithm are considered as candidate attack paths.

Step 8. Use $\min_{a_l^c} \frac{\overline{\hat{a}_l^t(b_l^t)} + a_l^c + |N| u_l^4}{P_l}$ as each candidate attack path's weight.

Step 9. Remove each candidate attack path whose $\overline{\hat{a}_l^t(b_l^t)}$ is greater than the remainder of attack budget.

Step 10. Choose the candidate attack path with the smallest weight to attack.

Step 11. Remove the attacked path and return to **Step 8** until there is no candidate attack path.

Step 12. Stop.

3.3 The Solution Approach for the NPDRAS Problem

The solution of the ARRAS problem is the best strategy for attacking a network where defense resource allocation and network planning strategies are known. That is to say, with different strategy of a network administrator, an attacker can change his/her strategy to compromise the network optimally. As mention before, the objective of the NPDRAS problem is to minimize the total penalty due to QoS violations caused by the attacker. Therefore, we can use the solution of the ARRAS problem as the input of the NPDRAS problem and adjust the strategy of the network administrator according to corresponding attack strategy in order to degrade the total penalty. The two opponents would change their strategies until a balance is reached and then the optimal solution of the NPDRAS problem is obtained. The concept of solving the NPDRAS problem is shown in **Figure 3-4**.

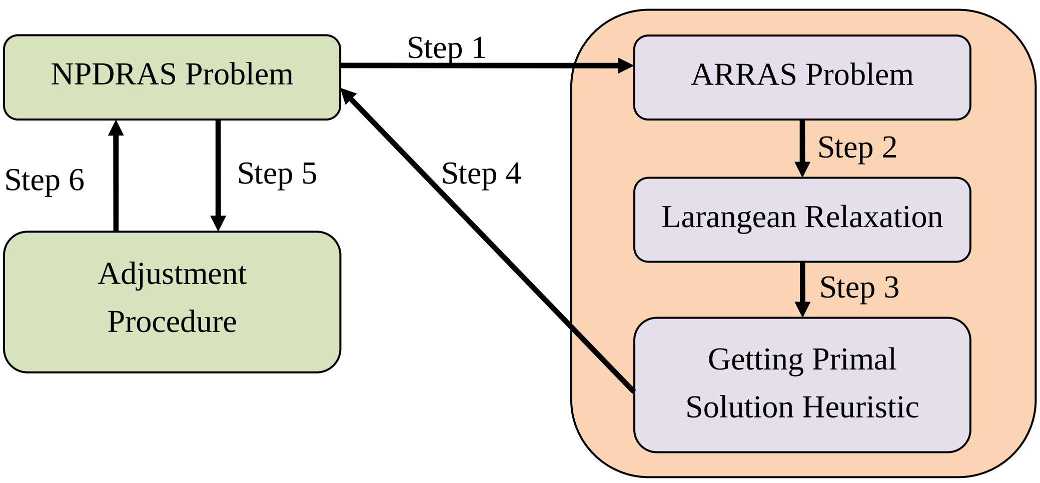


Figure 3-3. Solution Approach for the NPDRAS Problem

The concept of the adjustment procedure is to let the waste budget to be useful. It implies that the budget allocated to uncompromised node is too much and a certain proportion of it can be extracted to some compromised nodes. The extraction ratio of each uncompromised node is equal to the step size coefficient, denoted as θ. Moreover, the distribution of total extracted budget to each compromised node is according to the reward ratio of each node. That is, we add excess ten percentage of total defense budget to each compromised node and calculate the reduced penalty of that node. The proportion among the reduced penalties of compromised node is exactly the nodal reward ratio.

If the solution of the NPDRAS problem is not improved, it means the extracted budget of each uncompromised node is too much. Then the step size coefficient is halved to extract the less budget from uncompromised nodes. The adjustment procedure is executed to improve the defense strategy according to the corresponding attack strategy repeatedly until the defense is not improved within a certain number of iterations.

The proposed heuristic of the adjustment procedure is shown in **Table 3-3**.

Table 3-3. The Adjustment Procedure

Step 1.	Calculate the reduced penalty of each compromised node by adding excess ten percentage of total defense budget to the nodes respectively.
Step 2.	Extract θ ratio of budget from each uncompromised node, where θ is the step size coefficient.
Step 3.	Allocate the extracted budget to each compromised node according to the proportion among the reduced penalties of the nodes.
Step 4.	If the solution is not improved more than a certain number of iterations, go to **Step 6**; Otherwise, go to **Step 5**;
Step 5.	If the solution is not improved, θ is halved and go to **Step 2**; Otherwise, θ is set to initial value and go to **Step 1**.
Step 6.	Stop.

Chapter 4 Computational Experiments

4.1 Computational Experiments with the ARRAS Model

4.1.1 Simple Algorithms

For the comparison purpose with our proposed heuristic, we develop two simple algorithms to solve the ARRAS problem. The two algorithms are shown in Table 4-1 and **Table 4-2** respectively.

The two simple algorithms are similar to the second stage of our proposed heuristic, and the only difference is the weight of candidate attack path. The computational complexities of them are the same as $O(|L|^3 + |L|^2 |A_l^c|)$.

Table 4-1. Simple Algorithm 1

Step 1.	Use $\hat{a}_l^t (b_l^t)$ as each nodal cost and apply Dijkstra's algorithm to determine the minimal compromise cost from the attack subtree to the target node of each un-attacked path and the paths obtained from Dijkstra's algorithm are considered as candidate attack paths.

Step 2.	Use $\min_{a_l^c} \frac{\overline{\hat{a}_l^t(b_l^t)} + a_l^c}{P_l}$ as each candidate attack path's weight.
Step 3.	Remove each candidate attack path whose $\overline{\hat{a}_l^t(b_l^t)}$ is greater than the remainder of attack budget.
Step 4.	Choose the candidate attack path with the smallest weight to attack.
Step 5.	Remove the attacked path and return to **Step 1** until there is no candidate attack path.
Step 6.	Stop.

Table 4-2. Simple Algorithm 2

Step 1.	Use $\hat{a}_l^t(b_l^t)$ as each nodal cost and apply Dijkstra's algorithm to determine the minimal compromise cost from the attack subtree to the target node of each un-attacked path and the paths obtained from Dijkstra's algorithm are considered as candidate attack paths.
Step 2.	Use the $\frac{1}{\deg_l}$ as each candidate attack path's weight which $\deg_l$ is the degree of its target node.
Step 3.	Remove each candidate attack path which the attacker can't afford to compromise the target node.
Step 4.	Choose the candidate attack path with the smallest weight to attack.
Step 5.	Remove the attacked path and return to **Step 1** until there is no candidate attack path.
Step 6.	Stop.

4.1.2 Experiment Environment

The algorithms we proposed for ARRAS model are coded in Visual C++ and implemented on a PC with an INTEL Pentium 4 (3.00 GHz). The Iteration Counter Limit and Improve Counter Limit are set to 1000 and 20, respectively. The initial UB is set to 10^{10} to represent the infinity value.

The capacity of each link and node is a function that is monotonically decreasing to defense budget and monotonically increasing to attack budget. For example, we use the form $c_l(a_l^c, b_l^c) = 100 \times \ln(1 + \frac{20 \times b_l^c}{1 + a_l^c})$ as the capacity function.

Refer to previous research[21], each nodal buffer is modeled as an *M*/*M*/1 queue. It is remarkable to note that the delay function can be extended to any non *M*/*M*/1 model with monotonically increasing and convexity performance metrics. For illustration purpose, the delay function will be based on the *M*/*M*/1 model.

In order to observe the effect of penalty function, we adopt three different types of penalty function which are a linear form, a convex form, and a concave form.

We design two defense budget distribution strategies to determine how to distribute defense budget to each node is more effective under different scenarios. The first strategy is “uniform” distribution, where the total defense budget distributes averagely to each node. The second strategy is "degree-based" distribution, where each node is allocated the budget according to the percentage of that node's degree over the total degree of the network.

For each defense budget distribution strategy, we also perform ten different defense budget allocation ratio strategies to determine how to allocate the distributed budget to nodal capacity and nodal defense capability for a node is better. The ratios for the ten strategies are 0:10, 1:9, …, and 10:0, respectively. Each strategy is denoted as R_i, where i is the ratio to nodal capacity.

The concave function of nodal defense capability is considered to be close to real situation, say, the marginal nodal defense capability is decreased with the addition of defense budget. For example, we use the form $\hat{a}_l^t(b_l^t) = 2 + 2 \times \ln(10 \times b_l^t + 1)$ as the nodal defense capability function.

The test platform, the parameters of LR, and the parameters of the ARRAS model are shown in **Table 4-3**, **Table 4-4**, and **Table 4-5**, respectively.

Table 4-3. Test Platform

Test Platform	
CPU	Intel Pentium 4 (3.00 GHz)
RAM	1 GB
OS	Microsoft Windows XP Professional Version 2002 SP2

Table 4-4. Experimental Parameters of LR

Parameters of LR	
Parameter	**Value**
Iteration Counter Limit	1000
Improvement Counter Limit	20
Initial UB	10^{10}
Initial Lagrangean Multipliers	$\mu^1 = \mu^2 = \mu^3 = \mu^4 = 0$
Initial Scalar of Step Size	2

Table 4-5. Experimental Parameters of the ARRAS Model

Parameters of the ARRAS Model	
Parameter	**Value**
Network Size (Number of Nodes)	25, 64, 100
Number of Multicast Groups	$\left\lfloor \frac{\lvert N \rvert}{3} \right\rfloor$
Number of Destinations	1 ~ 3 (per a multicast group)
Delay Requirement	0.1 ~ 0.5 (sec)
Bandwidth Requirement	20 ~ 100 (packet/sec)
Multiple Path Requirement	1 ~ 2
Total Defense Budget	$3 \times \lvert N \rvert$
Total Attack Budget	20, 40, 60, 80, 100
Configurations of A_l^c	$A_l^c = \{1,2,..., A\}$
Capacity Function	$c_l(a_l^c, b_l^c) = 100 \times \ln(1 + \frac{20 \times b_l^c}{1 + a_l^c})$ (packets/sec)
Delay Function	$t_l(c_l(a_l^c, b_l^c), M_l) = \frac{1}{c_l(a_l^c, b_l^c) - M_l}$ (sec/packet)
Maximum Allowable End-to-End Delay	2 (sec)
Penalty Function	Linear $p_{gd}(h_{gdr}, \alpha_{gd}) = \begin{cases} 0 & , if\ h_{gdr} \le \alpha_{gd} \\ h_{gdr} - \alpha_{gd} & , if\ h_{gdr} > \alpha_{gd} \end{cases}$

Convex	$p_{gd}(h_{gdr},\alpha_{gd})=\begin{cases}0 & ,if\ h_{gdr}\le\alpha_{gd}\\(h_{gdr}-\alpha_{gd})^2 & ,if\ h_{gdr}>\alpha_{gd}\end{cases}$
Concave	$p_{gd}(h_{gdr},\alpha_{gd})=\begin{cases}0 & ,if\ h_{gdr}\le\alpha_{gd}\\\sqrt{h_{gdr}-\alpha_{gd}} & ,if\ h_{gdr}>\alpha_{gd}\end{cases}$
Defense Budget Distribution Strategy	Uniform distribution, Degree-based distribution
Defense Budget Allocation Ratio Strategy	0:10, 1:9, …, 10:0 (denoted as R_i, where i is the ratio to nodal capacity and 10 minus i is the ratio to nodal defense capability)
Nodal Defense Capability	$\hat{a}_l^t(b_l^t)=2+2\times\ln(10\times b_l^t+1)$

4.1.3 Experiment Results

The UB value is obtained from the LR process and the LR value is derived from the "getting primal feasible solution algorithm". In order to illustrate easily, we transform the two values into being positive by obtaining the absolute value, respectively. The two values also represent the upper bound and the lower bound of the optimal value. The gap between UB and LR is calculated by $\frac{\text{UB - LR}}{\text{LR}}\times100\%$.

Moreover, the SA_1 and SA_2 are the solutions obtained from simple algorithm 1 and 2. The improvement ratios of the two simple algorithms are calculated by $\frac{\text{LR - SA}_1}{\text{SA}_1}\times100\%$ and $\frac{\text{LR - SA}_2}{\text{SA}_2}\times100\%$, respectively.

Table 4-6. The Experiment Results (A=80, |N|=25, Uniform Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	24.1	19	26.84211	19	0	17.18	10.59371
	R_1	24.1	18.7106	28.80399	18.6962	0.077021	17.2106	8.715559
	R_2	22.7719	18.6696	21.97315	18.6696	0	17.2173	8.43512
	R_3	21.5529	17.261	24.86472	17.2412	0.114841	17.2203	0.236349
	R_4	20.3264	17.242	17.88888	17.2355	0.037713	17.222	0.116131
	R_5	20.2084	17.2389	17.22558	17.2338	0.029593	17.2232	0.091156
	R_6	20.1996	17.2332	17.21329	17.2332	0	17.224	0.053414
	R_7	20.1295	17.2335	16.80448	17.2335	0	17.2246	0.05167
	R_8	19.9604	17.2348	15.81451	17.2348	0	17.2251	0.056313
	R_9	19.9309	17.2459	15.56892	17.2379	0.046409	17.2255	0.118429
	R_{10}	19.9917	17.255	15.86033	17.2485	0.037684	17.2491	0.034205
Convex	R_0	41.77	33.0168	26.51135	33.0168	0	29.6932	11.19314
	R_1	41.77	32.0496	30.32924	32.0041	0.142169	29.7996	7.550437
	R_2	39.5746	31.9166	23.99378	29.9188	6.677407	29.8227	7.021162
	R_3	37.2048	29.9775	24.10908	29.9175	0.200552	29.8329	0.4847
	R_4	34.7702	29.9124	16.24009	29.8952	0.057534	29.8388	0.246659
	R_5	34.6021	29.9005	15.72415	29.888	0.041823	29.8426	0.194018
	R_6	34.5888	29.8856	15.73735	29.8856	0	29.8454	0.134694
	R_7	34.4824	29.8863	15.37862	29.8863	0	29.8476	0.129659
	R_8	34.1765	29.8912	14.33633	29.8912	0	29.8493	0.140372
	R_9	34.1451	29.926	14.09844	29.9025	0.078589	29.8507	0.252255
	R_{10}	34.1859	29.9566	14.11809	29.9428	0.046088	29.9377	0.063131
Concave	R_0	18.3526	14.4458	27.04454	14.4458	0	13.0973	10.29602
	R_1	18.3526	14.3337	28.03812	14.328	0.039782	13.109	9.342436
	R_2	17.5223	14.3176	22.38294	14.3176	0	13.1115	9.198795
	R_3	16.7942	13.128	27.92657	13.1202	0.05945	13.1127	0.116681
	R_4	15.9017	13.1207	21.19552	13.118	0.020582	13.1133	0.056431
	R_5	15.5759	13.1195	18.72327	13.1174	0.016009	13.1138	0.043466
	R_6	15.8288	13.1176	20.66841	13.1171	0.003812	13.1141	0.026689
	R_7	15.8572	13.1179	20.88215	13.1172	0.005337	13.1143	0.027451
	R_8	15.6378	13.1181	19.20781	13.1178	0.002287	13.1145	0.027451
	R_9	15.6451	13.1222	19.2262	13.1189	0.025155	13.1147	0.057188
	R_{10}	15.6573	13.1256	19.28826	13.1227	0.022099	13.1233	0.017526

Table 4-7. The Experiment Results (*A*=80, |*N*|=25, Degree-based Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	24.1	18.98	26.97576	18.98	0	13.6015	39.54343
	R_1	24.1	18.6524	29.20589	18.6294	0.123461	13.6243	36.90538
	R_2	22.8841	17.2142	32.93734	17.2077	0.037774	13.6302	26.29455
	R_3	22.204	17.2178	28.95957	17.2016	0.094177	13.6328	26.29687
	R_4	21.6201	17.2065	25.65077	17.2048	0.009881	13.6363	26.18159
	R_5	21.0022	17.2031	22.08381	17.2026	0.002907	13.6354	26.16498
	R_6	20.6405	17.2028	19.98337	17.1868	0.093095	13.6361	26.15631
	R_7	20.3142	17.2041	18.07767	17.187	0.099494	13.6367	26.16029
	R_8	20.3578	17.2058	18.3194	17.2034	0.013951	13.6372	26.16813
	R_9	19.9613	17.2195	15.92265	17.2129	0.038343	13.6375	26.26581
	R_{10}	20.3259	18.6962	8.716745	18.6962	0	13.6378	37.09103
Convex	R_0	41.77	32.966	26.7063	29.823	10.53885	23.2737	41.64486
	R_1	41.77	31.8597	31.10607	31.7842	0.237539	23.3547	36.41665
	R_2	39.6604	29.8232	32.98506	29.8146	0.028845	23.3746	27.58807
	R_3	38.4219	29.8349	28.78173	29.7937	0.138284	23.3833	27.59063
	R_4	37.1193	29.804	24.54469	29.804	0	23.3948	27.39583
	R_5	35.938	29.7944	20.61998	29.7944	0	23.3916	27.37222
	R_6	35.4944	29.7779	19.19712	29.7352	0.143601	23.394	27.28862
	R_7	34.9065	29.7821	17.20631	29.7354	0.157052	23.3958	27.29678
	R_8	34.8473	29.7952	16.95609	29.7952	0	23.3972	27.34515
	R_9	34.3772	29.8379	15.2132	29.8299	0.026819	23.3983	27.52166
	R_{10}	34.822	32.0061	8.79801	32.0061	0	23.3993	36.7823
Concave	R_0	18.3526	14.4387	27.10701	14.4371	0.011083	10.4229	38.52862
	R_1	18.3526	14.3109	28.24211	14.3019	0.062929	10.4315	37.18928
	R_2	17.5725	13.1106	34.03277	13.1064	0.032045	10.4338	25.65508
	R_3	17.3451	13.1111	32.29325	13.1041	0.053418	10.4348	25.64783
	R_4	17.0359	13.1071	29.97459	13.1054	0.012972	10.4361	25.59385
	R_5	16.5992	13.1058	26.65537	13.1046	0.009157	10.4358	25.58501
	R_6	16.2222	13.1057	23.77973	13.0988	0.052677	10.4361	25.58044
	R_7	15.9488	13.1062	21.68897	13.099	0.054966	10.4363	25.58282
	R_8	15.4936	13.1069	18.20949	13.105	0.014498	10.4365	25.58712
	R_9	15.4082	13.1119	17.5131	13.1086	0.025174	10.4367	25.63262
	R_{10}	15.9948	14.3279	11.63394	14.3279	0	10.4368	37.2825

Table 4-8. The Experiment Results (*A*=80, |*N*|=64, Uniform Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	46.5516	36.46	27.67855	34.12	6.858148	24.85	46.72032
	R_1	44.4793	35.4696	25.40119	34.6065	2.49404	21.6632	63.73204
	R_2	43.108	31.7192	35.90507	31.6314	0.277572	21.6394	46.58077
	R_3	41.4875	31.7958	30.48107	31.5581	0.753214	21.6307	46.99386
	R_4	40.4135	29.9655	34.86676	27.9759	7.111836	21.6289	38.5438
	R_5	40.15	29.7471	34.97114	27.9813	6.310643	21.6293	37.5315
	R_6	39.7361	29.7516	33.55954	26.8142	10.95464	21.6312	37.54022
	R_7	39.3069	29.7553	32.1005	26.8164	10.95934	21.6347	37.53507
	R_8	38.7029	28.1897	37.29447	26.8182	5.114064	21.629	30.33289
	R_9	38.3306	30.3529	26.28316	26.8473	13.05755	21.6272	40.34595
	R_{10}	39.553	31.8505	24.18329	29.7913	6.912085	21.7042	46.74809
Convex	R_0	79.2949	61.0212	29.94648	58.6912	3.969931	40.9451	49.03175
	R_1	76.2776	57.5512	32.53868	57.5512	0	36.0816	59.5029
	R_2	74.2366	56.0101	32.54145	53.13	5.420855	36.1021	55.14361
	R_3	71.1525	53.4802	33.04457	52.9066	1.084175	36.1062	48.11916
	R_4	70.2063	51.9368	35.17641	46.4682	11.76848	36.1224	43.78004
	R_5	67.7001	50.3731	34.39733	46.4858	8.362339	36.14	39.38323
	R_6	67.6421	49.619	36.32298	45.1944	9.790151	36.1589	37.22486
	R_7	65.6515	51.5865	27.26489	45.2115	14.10039	36.1809	42.57937
	R_8	65.5498	47.1809	38.93292	45.2255	4.323667	36.17	30.44208
	R_9	65.4068	49.4415	32.29129	45.33	9.070152	36.1662	36.70637
	R_{10}	67.5492	53.6907	25.81173	49.6357	8.169523	36.4349	47.36063
Concave	R_0	36.1659	28.2871	27.85298	28.2126	0.264066	19.8647	42.39883
	R_1	35.18	26.9276	30.64662	26.9276	0	17.169	56.83849
	R_2	34.405	25.7851	33.42977	24.488	5.29688	16.9978	51.69669
	R_3	33.5035	24.5257	36.60568	24.4584	0.275161	16.9548	44.65343
	R_4	32.6832	23.3907	39.72733	21.7832	7.37954	16.9217	38.22902
	R_5	32.0769	23.3427	37.41727	21.7853	7.148857	16.8929	38.18054
	R_6	31.7193	23.345	35.87192	20.963	11.36288	16.8654	38.41949
	R_7	31.6015	23.3468	35.35688	20.9524	11.42781	16.8365	38.66778
	R_8	30.8941	21.8658	41.28959	20.9431	4.405747	16.7922	30.21403
	R_9	31.0396	22.1344	40.2324	20.9349	5.729667	16.7567	32.09283
	R_{10}	32.3263	24.5492	31.67965	23.2486	5.594315	16.7858	46.24981

Table 4-9. The Experiment Results (A=80, |N|=64, Degree-based Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	46.4608	35.3	31.617	35.3	0	21.6102	63.34879
	R_1	43.9162	33.8827	29.61246	33.0255	2.59557	21.6474	56.52088
	R_2	42.1528	31.7286	32.85427	29.9639	5.88942	21.6049	46.85835
	R_3	40.3476	30.6719	31.54581	29.9696	2.343375	21.5631	42.24253
	R_4	38.6306	30.34	27.32564	29.9766	1.212279	21.5831	40.57295
	R_5	37.2209	29.9817	24.1454	28.138	6.552349	21.5721	38.98369
	R_6	36.2385	28.1636	28.67141	25.2238	11.65487	21.5695	30.57141
	R_7	35.3686	28.1667	25.56885	25.2238	11.66716	21.584	30.49805
	R_8	34.7635	28.1755	23.38202	25.2257	11.69363	21.5807	30.55879
	R_9	34.9121	28.2039	23.78465	25.2307	11.78406	21.5745	30.72794
	R_{10}	36.7644	31.7765	15.69682	30.4643	4.307337	21.6317	46.89784
Convex	R_0	79.6398	60.1768	32.34303	58.6912	2.531214	35.7383	68.38182
	R_1	75.7481	55.7086	35.972	55.0438	1.207765	36.053	54.51863
	R_2	72.4732	53.474	35.52979	50.3616	6.180105	35.9784	48.62807
	R_3	69.2106	52.6852	31.3663	50.3775	4.580815	35.8938	46.78078
	R_4	66.1818	51.5666	28.34238	50.4006	2.313465	35.9854	43.29867
	R_5	63.975	50.4173	26.89097	48.8846	3.135343	35.9653	40.18318
	R_6	61.907	47.0911	31.46221	42.7673	10.11006	35.9697	30.9188
	R_7	60.7093	47.1011	28.89147	42.7409	10.20147	36.0289	30.73144
	R_8	59.9154	47.1315	27.1239	42.7473	10.25609	36.0263	30.82526
	R_9	60.173	47.2307	27.4023	42.7077	10.5906	36.0117	31.15376
	R_{10}	63.06	53.6151	17.61612	51.9447	3.215727	36.181	48.18579
Concave	R_0	36.1418	28.2871	27.76778	27.199	4.000515	17.2552	63.93377
	R_1	34.5767	25.9649	33.16708	25.6707	1.146054	17.0706	52.10303
	R_2	33.5516	24.5245	36.8085	23.196	5.727281	16.9975	44.28298
	R_3	32.7668	23.4728	39.59477	23.1984	1.18284	16.9397	38.5668
	R_4	31.1589	23.2015	34.29692	22.2185	4.424241	16.9146	37.16848
	R_5	30.464	23.2031	31.2928	22.0001	5.468157	16.8815	37.44691
	R_6	29.6598	21.8558	35.70677	19.6919	10.98878	16.8533	29.68261
	R_7	29.3399	21.857	34.23571	19.645	11.25986	16.8311	29.86079
	R_8	28.9126	21.8604	32.26016	19.6254	11.3883	16.7974	30.14157
	R_9	28.9741	21.8712	32.47604	19.6433	11.34178	16.735	30.69137
	R_{10}	29.8617	24.5431	21.67045	23.3949	4.907907	16.7584	46.45253

Table 4-10. The Experiment Results (A=80, $|N|$=100, Uniform Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	68.6878	52.2525	31.45361	48.34	8.093711	34.74	50.41019
	R_1	67.2159	48.3596	38.99184	48.3596	0	34.7163	39.29941
	R_2	65.8161	48.7477	35.01375	48.4092	0.699247	34.7946	40.10134
	R_3	65.2623	48.4398	34.72867	48.4398	0	34.8519	38.98754
	R_4	64.5087	48.4593	33.11934	46.085	5.152002	34.8968	38.8646
	R_5	63.9292	46.0923	38.69822	45.1158	2.16443	34.931	31.95242
	R_6	63.3887	46.1004	37.50141	45.114	2.186461	34.9583	31.87255
	R_7	63.34	47.616	33.02251	45.1093	5.556947	34.9812	36.11883
	R_8	63.4659	47.6782	33.11304	45.7552	4.202801	34.9952	36.24211
	R_9	64.457	49.6313	29.87167	45.7733	8.428494	35.0107	41.76038
	R_{10}	67.0803	55.9246	19.94775	49.9684	11.91993	41.6626	34.23214
Convex	R_0	118.451	86.7671	36.51603	81.2892	6.73878	57.7582	50.22473
	R_1	115.195	81.87	40.70478	81.3406	0.650843	57.8021	41.63845
	R_2	113.136	81.5043	38.80985	81.5043	0	58.112	40.25382
	R_3	112.521	81.7732	37.60132	79.1963	3.253814	58.3211	40.21203
	R_4	110.477	79.2537	39.39665	79.2537	0	58.4706	35.54453
	R_5	109.985	79.2771	38.73489	75.9477	4.383806	58.5848	35.32025
	R_6	109.108	79.3036	37.58266	75.9684	4.390246	58.6764	35.15417
	R_7	109.259	81.6753	33.77239	75.977	7.500033	58.7532	39.01422
	R_8	109.132	81.8743	33.29213	76.0521	7.655541	58.7999	39.24224
	R_9	110.87	83.2993	33.09836	76.0772	9.49312	58.8518	41.54079
	R_{10}	115.936	95.6169	21.25053	83.9042	13.95961	69.9661	36.66175
Concave	R_0	55.918	40.2064	39.07736	37.5372	7.110813	27.3427	47.0462
	R_1	53.3066	37.5686	41.89137	37.4081	0.429051	27.1559	38.34415
	R_2	52.4527	37.56	39.65043	37.4274	0.354286	27.0765	38.71808
	R_3	51.7743	37.4394	38.28827	37.4394	0	27.0247	38.53771
	R_4	51.27	36.5038	40.45113	35.2404	3.58509	27.0422	34.98828
	R_5	50.9144	35.2444	44.46096	34.9854	0.740309	27.0555	30.26704
	R_6	50.4714	35.2464	43.1959	35.1834	0.179062	27.066	30.2239
	R_7	50.2901	36.473	37.88309	35.23	3.528243	27.0749	34.71149
	R_8	50.4634	36.4977	38.2646	35.2788	3.455049	27.0804	34.77534
	R_9	50.9357	37.8992	34.39782	35.4197	7.000342	27.0863	39.92018
	R_{10}	53.47	42.9198	24.5812	38.6778	10.96753	32.2415	33.11974

Table 4-11. The Experiment Results (A=80, |N|=100, Degree-based Distribution)

Penalty Function	Budget Allocation Ratio	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	R_0	65.5123	49.78	31.60366	48.09	3.514244	34.74	43.29303
	R_1	61.5098	48.3915	27.10869	48.3915	0	34.6777	39.54645
	R_2	59.6141	47.6285	25.16476	45.1912	5.393307	34.7326	37.12909
	R_3	58.0495	45.022	28.93585	43.807	2.773529	34.7886	29.41596
	R_4	56.8219	45.0149	26.22909	41.8377	7.594108	34.8452	29.18537
	R_5	55.9744	43.5291	28.59076	41.8524	4.006222	34.8779	24.80425
	R_6	55.411	43.7562	26.63577	41.8639	4.520124	34.9049	25.35833
	R_7	54.808	43.5508	25.84843	41.8733	4.006133	34.9277	24.68843
	R_8	54.8791	43.5601	25.98479	41.8811	4.008968	34.9477	24.64368
	R_9	54.6722	43.568	25.48705	41.9126	3.949648	34.9646	24.60603
	R_{10}	58.9133	48.5202	21.42015	46.6293	4.055176	35.0153	38.56857
Convex	R_0	112.765	83.6138	34.8641	83.37	0.292431	57.7582	44.76525
	R_1	106.132	81.5542	30.13677	81.4876	0.08173	57.6636	41.43099
	R_2	102.636	81.6319	25.73026	76.3204	6.959476	57.9172	40.94587
	R_3	99.7158	75.666	31.78416	74.4329	1.65666	58.1302	30.16642
	R_4	98.5721	73.6945	33.75774	70.6176	4.357129	58.3193	26.36383
	R_5	96.0407	73.5361	30.60347	70.6663	4.061059	58.4272	25.85936
	R_6	94.5025	73.4571	28.64992	70.7045	3.893104	58.5168	25.53164
	R_7	94.2325	73.4932	28.21935	70.7356	3.898461	58.5928	25.43043
	R_8	93.9468	73.5237	27.77757	70.7615	3.903535	58.6598	25.33916
	R_9	94.6633	73.5501	28.70588	70.8642	3.790207	58.7141	25.26821
	R_{10}	100.755	81.9406	22.96102	78.2928	4.659177	58.8679	39.19403
Concave	R_0	53.1684	38.6927	37.41197	37.4664	3.273066	27.3427	41.51017
	R_1	50.4755	37.4682	34.71557	37.418	0.13416	27.1602	37.95259
	R_2	49.3695	36.4829	35.3223	33.7033	8.247264	27.0622	34.81129
	R_3	47.9469	34.9156	37.32229	33.7125	3.568706	26.9989	29.32231
	R_4	46.9673	34.8974	34.58682	32.3057	8.022423	27.0209	29.14966
	R_5	45.8421	33.6594	36.19405	32.3114	4.171902	27.0336	24.5095
	R_6	45.0782	33.6387	34.00696	32.3159	4.093341	27.0441	24.38462
	R_7	44.4716	33.6182	32.2843	32.3196	4.017995	27.053	24.26792
	R_8	44.4675	33.6218	32.25794	32.3226	4.019479	27.0608	24.2454
	R_9	45.0231	33.6991	33.60327	32.335	4.218649	27.0675	24.50023
	R_{10}	48.1487	37.4674	28.50825	36.0959	3.799601	27.0881	38.31683

Table 4-12. The Experiment Results (R_5, |N|=25, Uniform Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	12.4357	8.23703	50.97311	8.23703	0	8.2289	0.098798
	40	17.3813	13.5978	27.82435	13.5978	0	13.5978	0
	60	19.0798	17.2001	10.92842	17.2001	0	13.637	26.12818
	80	20.2084	17.2389	17.22558	17.2338	0.029593	17.2232	0.091156
	100	21.4246	18.7232	14.42809	18.7111	0.064667	17.2432	8.583094
Convex	20	21.3717	13.667	56.37448	13.667	0	13.6414	0.187664
	40	29.4299	23.2614	26.51818	23.2614	0	23.2614	0
	60	32.7739	29.7681	10.09739	29.7681	0	23.3951	27.24075
	80	34.6021	29.9005	15.72415	29.888	0.041823	29.8426	0.194018
	100	36.8768	32.0926	14.90749	32.0544	0.119172	29.9165	7.273912
Concave	20	10.505	6.412	63.83344	6.412	0	6.40876	0.050556
	40	14.1562	10.4215	35.83649	10.4215	0	10.4215	0
	60	15.5567	13.1047	18.71084	13.1047	0	10.4365	25.56604
	80	15.5759	13.1195	18.72327	13.1174	0.016009	13.1138	0.043466
	100	16.5796	14.3385	15.62995	14.3337	0.033488	13.1211	9.278186

Table 4-13. The Experiment Results (R_5, |N|=25, Degree-based Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	10.0966	8.20883	22.99682	8.20883	0	8.20883	0
	40	15.7101	11.8077	33.04962	11.8077	0	8.20883	43.84145
	60	19.4455	15.2222	27.74435	13.3077	14.38641	13.5661	12.20764
	80	21.0022	17.2031	22.08381	17.2026	0.002907	13.6354	26.16498
	100	23.1579	18.7012	23.83109	18.6569	0.237446	13.6354	37.15183
Convex	20	16.8655	13.5777	24.2147	13.5777	0	13.5777	0
	40	27.1578	20.0733	35.29315	20.0733	0	13.5777	47.84021
	60	32.9342	25.919	27.06586	22.3233	16.10739	23.1587	11.91906
	80	35.938	29.7944	20.61998	29.7944	0	23.3916	27.37222
	100	40.069	32.0226	25.12725	31.8762	0.459277	23.3916	36.89786
Concave	20	8.2908	6.40078	29.52796	6.40078	0	6.40078	0
	40	13.2771	9.08261	46.18155	9.08261	0	6.40078	41.89849
	60	14.8961	11.695	27.37153	10.3074	13.46217	10.409	12.35469
	80	16.5992	13.1058	26.65537	13.1046	0.009157	10.4358	25.58501
	100	17.6506	14.3298	23.17408	14.3125	0.120873	10.4358	37.31386

Table 4-14. The Experiment Results (R_5, $|N|$=64, Uniform Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	20.3867	14.9526	36.34217	9.90682	50.93239	14.9526	0
	40	28.5081	21.4715	32.77181	16.5584	29.67135	15.0441	42.72373
	60	35.7523	26.677	34.01919	22.7453	17.28577	21.5657	23.70106
	80	40.15	29.7471	34.97114	27.9813	6.310643	21.6293	37.5315
	100	44.0658	33.3609	32.08816	31.5797	5.640332	21.6246	54.27291
Convex	20	34.6995	24.9576	39.0338	16.569	50.62828	24.9576	0
	40	49.2018	35.6269	38.10295	27.6862	28.68108	25.2728	40.96934
	60	60.8959	42.689	42.6501	37.287	14.48762	35.9512	18.74152
	80	67.7001	50.3731	34.39733	46.4858	8.362339	36.14	39.38323
	100	75.0461	56.1526	33.64671	52.9796	5.989098	36.1242	55.44317
Concave	20	17.1807	11.7271	46.50425	7.69773	52.34491	11.7271	0
	40	24.083	16.8308	43.08886	12.964	29.82721	11.762	43.09471
	60	29.0116	20.6383	40.57166	17.9808	14.77965	16.8667	22.36122
	80	32.0769	23.3427	37.41727	21.7853	7.148857	16.8929	38.18054
	100	34.7176	25.7991	34.56904	24.4669	5.444907	16.891	52.73874

Table 4-15. The Experiment Results (R_5, $|N|$=64, Degree-based Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	17.6749	9.91032	78.34843	9.91032	0	0.011548	85718.5
	40	24.5236	20.1964	21.4256	18.619	8.471991	15.0439	34.24976
	60	31.7232	23.796	33.31316	22.2378	7.006988	15.0439	58.17707
	80	37.2209	29.9817	24.1454	28.138	6.552349	21.5721	38.98369
	100	42.4587	31.7923	33.55026	31.6326	0.504859	21.6264	47.0069
Convex	20	30.206	16.5813	82.16907	16.5813	0	0	-
	40	42.1667	34.3213	22.85869	31.6759	8.35146	25.2645	35.84793
	60	54.4791	40.82	33.46178	38.2445	6.734302	25.2645	61.57058
	80	63.975	50.4173	26.89097	48.8846	3.135343	35.9653	40.18318
	100	72.7932	53.6602	35.65585	53.3824	0.520396	36.1237	48.54569
Concave	20	14.3266	7.69883	86.08802	7.69883	0	0.151974	4965.886
	40	21.2244	15.5488	36.50185	14.4634	7.50446	11.7742	32.05823
	60	26.6245	18.2309	46.04051	17.1859	6.080566	11.7742	54.8377
	80	30.464	23.2031	31.2928	22.0001	5.468157	16.8815	37.44691
	100	33.5619	24.6413	36.20182	24.6413	0	16.904	45.77201

Table 4-16. The Experiment Results (R_5, $|N|$=100, Uniform Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	29.9533	13.5986	120.2675	11.663	16.59607	13.1348	3.531078
	40	42.9243	30.1086	42.56492	28.275	6.484881	13.1818	128.4104
	60	53.9095	39.0077	38.2022	35.3685	10.28938	34.8824	11.82631
	80	63.9292	46.0923	38.69822	45.1158	2.16443	34.931	31.95242
	100	69.9599	53.9945	29.56857	49.9116	8.180263	41.53	30.01324
Convex	20	50.9125	23.2882	118.6193	19.6113	18.74888	21.6919	7.358968
	40	74.5385	50.7744	46.80331	47.4259	7.060488	21.614	134.9144
	60	94.6065	65.9105	43.53783	60.0358	9.785328	58.4412	12.78088
	80	109.985	79.2771	38.73489	75.9477	4.383806	58.5848	35.32025
	100	120.498	89.2914	34.94917	83.7163	6.659516	69.5215	28.4371
Concave	20	24.8815	10.5869	135.0216	9.09412	16.41478	10.4087	1.712029
	40	35.9141	23.3136	54.04785	22.0174	5.887162	10.5373	121.2483
	60	44.1167	29.9215	47.44147	27.3465	9.416196	27.0354	10.67526
	80	50.9144	35.2444	44.46096	34.9854	0.740309	27.0555	30.26704
	100	55.2724	40.2557	37.30329	38.6557	4.139105	32.1902	25.05576

Table 4-17. The Experiment Results (R_5, $|N|$=100, Degree-based Distribution)

Penalty Function	Attack Budget	UB	LR	Gap (%)	SA1	Imp. Ratio to SA1 (%)	SA2	Imp. Ratio to SA2 (%)
Linear	20	26.9649	13.1532	105.0064	13.1532	0	0.064092	20422.41
	40	38.7354	28.2391	37.16939	28.2391	0	13.1748	114.3418
	60	47.9004	36.724	30.4335	35.3023	4.027216	13.3867	174.332
	80	55.9744	43.5291	28.59076	41.8524	4.006222	34.8779	24.80425
	100	62.3664	47.7158	30.70388	46.6073	2.378383	34.9222	36.63458
Convex	20	45.8835	21.8342	110.1451	21.8342	0	0.002993	729489.3
	40	67.2863	47.3588	42.07771	47.3588	0	21.5996	119.2578
	60	83.0238	61.7789	34.3886	59.8622	3.201854	21.6372	185.5217
	80	96.0407	73.5361	30.60347	70.6663	4.061059	58.4272	25.85936
	100	107.386	84.1488	27.61442	78.2181	7.58226	58.5561	43.70629
Concave	20	21.9243	10.2947	112.9669	10.2947	0	0.369273	2687.829
	40	31.6771	21.8838	44.75137	21.8838	0	10.5314	107.7957
	60	39.5497	28.5488	38.53367	27.1975	4.968471	11.1294	156.517
	80	45.8421	33.6594	36.19405	32.3114	4.171902	27.0336	24.5095
	100	50.7635	38.6569	31.31808	36.0873	7.120511	27.052	42.89849

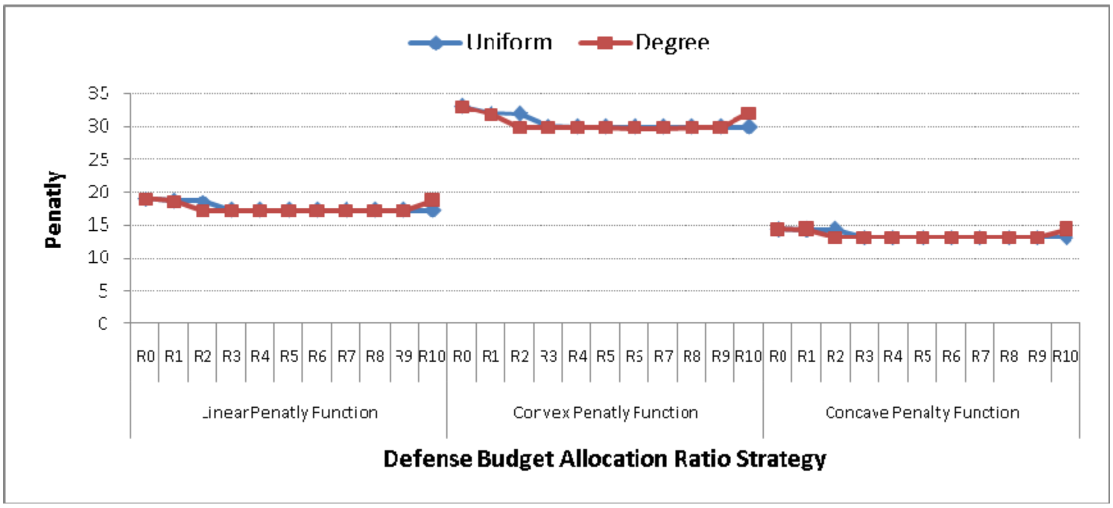


Figure 4-1. Total Penalty under Different Allocation Ratio (*A*=80, |*N*|=25)

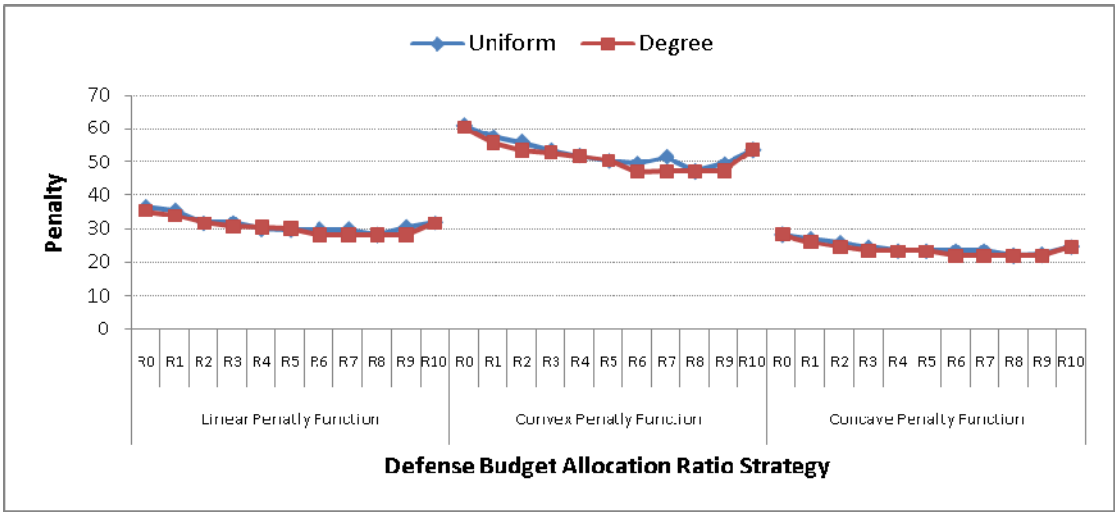


Figure 4-2. Total Penalty under Different Allocation Ratio (*A*=80, |*N*|=64)

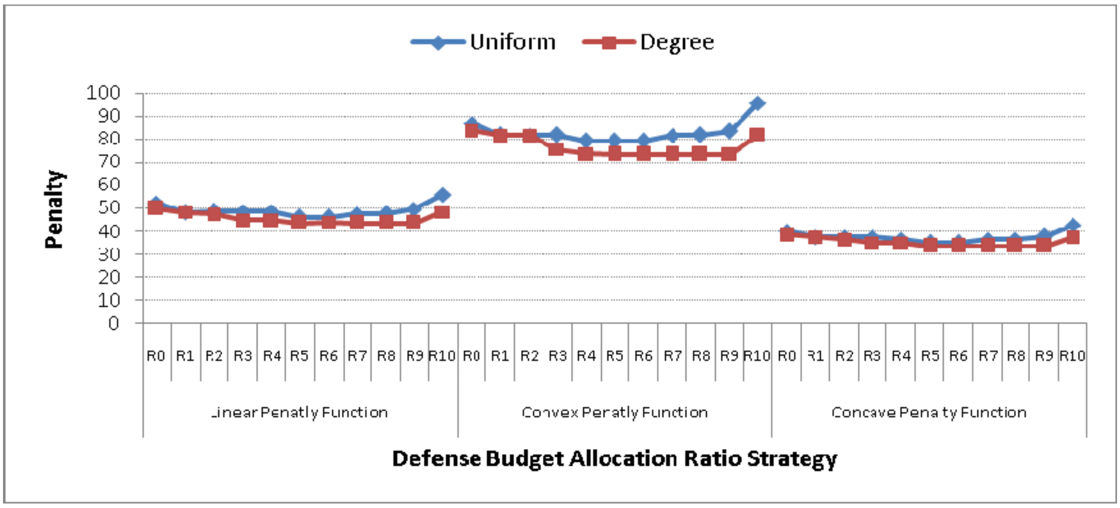


Figure 4-3. Total Penalty under Different Allocation Ratio (*A*=80, |*N*|=100)

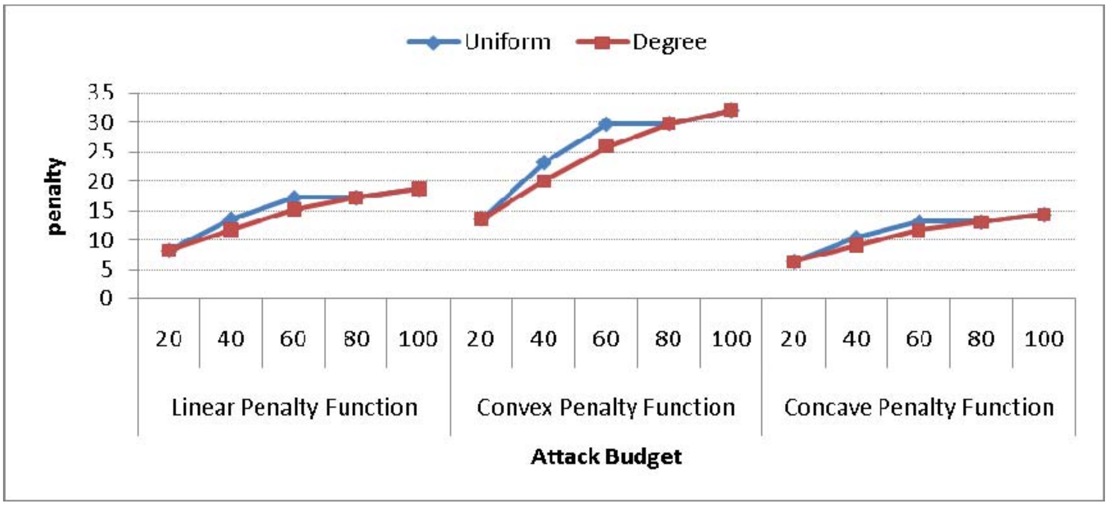


Figure 4-4. Total Penalty under Different Attack Budget (R_5, |N|=25)

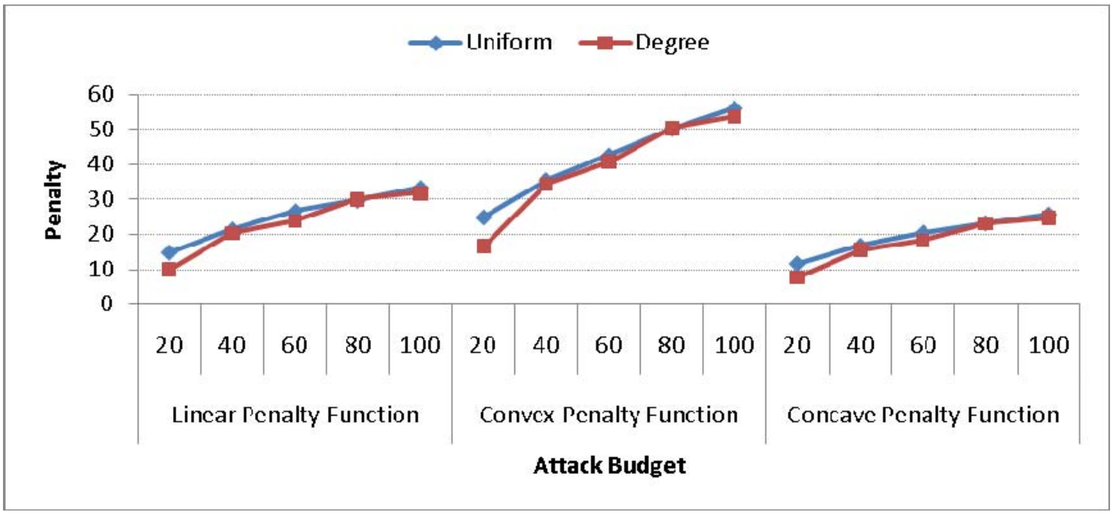


Figure 4-5. Total Penalty under Different Attack Budget (R_5, |N|=64)

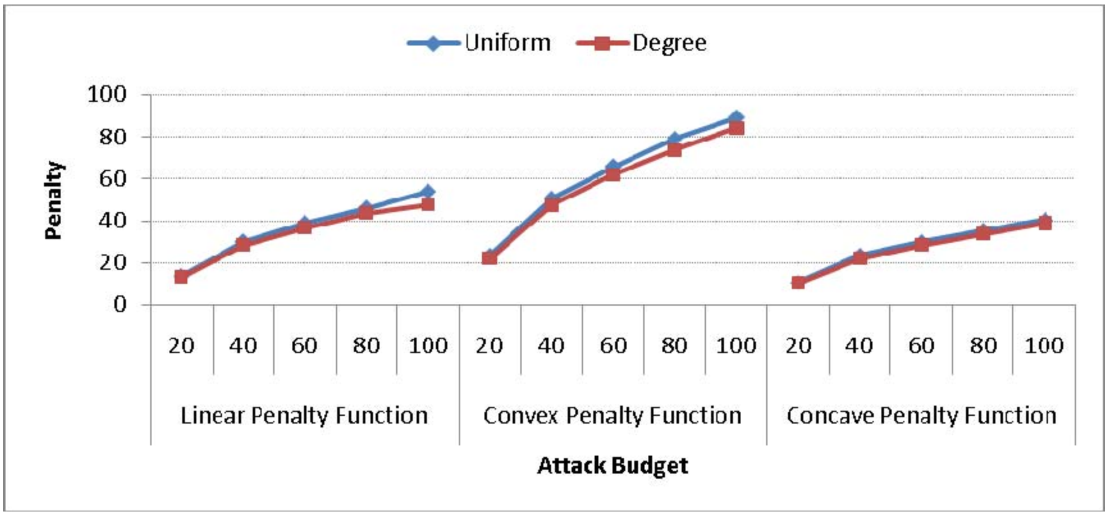


Figure 4-6. Total Penalty under Different Attack Budget (R_5, |N|=100)

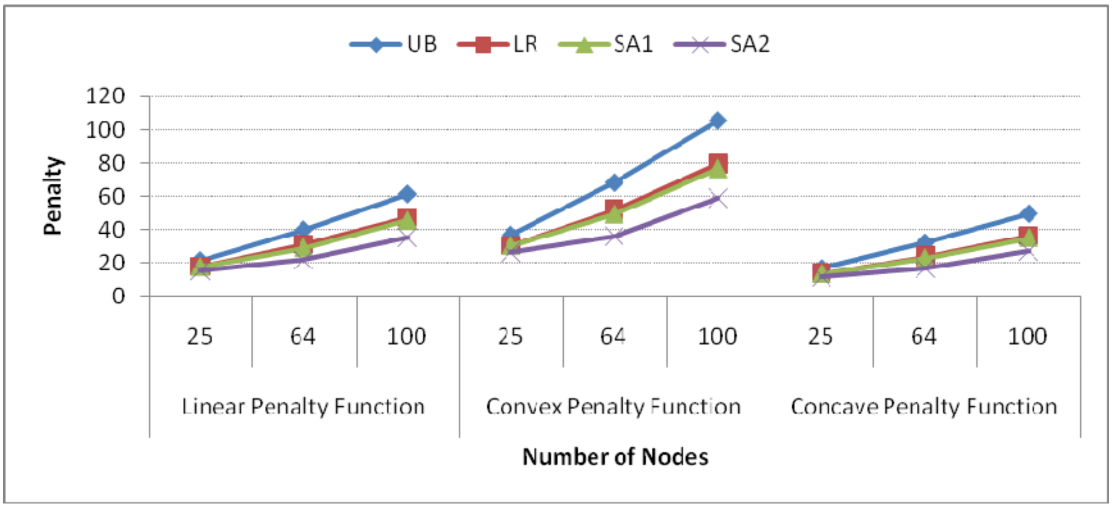


Figure 4-7. Total Penalty under Different Numbers of Nodes (R_5, A=80)

4.1.4 Discussion of Results

Figures 4-1 to **4-3** show the caused penalties under different numbers of nodes, penalty function types, defense budget distribution, and defense budget allocation ratio strategies within the attack budget 80. We can observe the penalty caused by degree-based distribution is less than that caused by uniform distribution in most situations, that is to say, the defense ability of degree-based distribution is better than the other. That is because the degree of a node implies the frequency of the node as a hop-site to connect some O-D pairs. Moreover, the difference between the two distributions gets more obvious with the growth of the number of nodes.

Since the nodal defense capability function is a concave form, too much budget allocated to nodal defense capability may be useless. Therefore, the former of each curve in these figures may fall by shifting useless budget from nodal defense capability to nodal capacity. However, the later of each curve may rise because the shifted budget is too much and the nodal defense capability turns weak quickly. Hence, the curves in these figures all tend to convex form but the best ratio strategy which the minimal values appear at is uncertain under different scenarios. In the experiment cases, the strategies R_5 and R_6 are the most robust.

Figures 4-4 to **4-6** show the effect of different attack budget under different numbers of nodes, penalty function types, and defense budget distribution strategies within the defense budget allocation ratio strategy R_5. It is obvious that all curves tend to concave form with the enlargement of attack budget whatever the scenario is. That is to say, the marginal penalty almost decreases when the attack budget increases.

Moreover, it is also obvious that the penalty caused by convex form is the biggest, the penalty caused by concave form is the smallest, and the penalty caused by linear form is between them.

Figure 4-7 compares the performance of our proposed Lagrangean relaxation-based algorithm with simple algorithm 1 and 2 under different numbers of nodes and different penalty function types. The value of each point is the average penalty of two different defense budget distribution and ten allocation ratio strategies under the same number of nodes and the same penalty function type within the attack budget 80. We could observe that the penalty of our proposed heuristic always higher than that of simple algorithm 1 and 2, namely, our proposed heuristic outperforms the two simple algorithms and the average improvement ratios to them are 4.5% and 30% except special cases respectively. The average gap between UBs and LRs is less than 33%. Moreover, the penalty increases with the enlargement of network size. That is because the more the network size is, the more the amount of choices to attack is.

4.2 Computational Experiments with the NPDRAS Model

4.2.1 Experiment Environment

The algorithms we proposed for NPDRAS model are coded in Visual C++ and implemented on a PC with an INTEL Pentium 4 (3.00 GHz). The Iteration Counter Limit and Improve Counter Limit are set to 50 and 5, respectively. The initial step size coefficient, θ, is set to 0.5.

From the results of the ARRAS model, we can obtain that the degree-based distribution is the best defense budget distribution strategy but the best allocation ratio strategy is uncertain. We therefore execute the ten defense budget allocation ratio strategies mentioned in **Section 4.1.2** for the degree-based distribution and choose the best one as the initial defense strategy for the NPDRAS problem. Besides, the multicast tree of each group is constructed by the shortest path algorithm to approach the minimal end-to-end delay from a source to each destination. We use 80 as the attack budget. Other unmentioned parameters are the same to those in the ARRAS model.

For comparing our proposed adjustment heuristic, denoted as "benefit" re-distribution, we also execute the "uniform" re-distribution where the extracted budget distributes averagely to each compromised node.

4.2.2 Experiment Results

The Init. P. value is obtained from the initial defense strategy, the Bef. P. value is derived from the adjustment procedure, and the Uni. P. value is gained from the uniform re-distribution strategy. The improvement ratios of the two re-distributions are calculated by $\frac{\text{Bef. P. - Init. P.}}{\text{Init. P.}} \times 100\%$ and $\frac{\text{Uni. P. - Init. P.}}{\text{Init. P.}} \times 100\%$, respectively. The experiments results are shown in **Table 4-18**.

4.2.3 Discussion of Results

Figure 4-8 show the improvement by performing our proposed adjustment procedure, and compare the two different re-distributions under different numbers of nodes and penalty function types. We can observe that the benefit re-distribution

Table 4-18. The Experiment Results for the NPDRAS Model

Penalty Function	Number of Nodes	Init. P.	Bef. P.	Imp. Ratio of Bef. P. (%)	Uni. P.	Imp. Ratio of Uni. P. (%)
Linear	25	17.2028	15.1536	13.52286	15.175	13.36277
	64	28.1636	24.5605	14.6703	25.5333	10.30145
	100	43.5291	38.0443	14.41688	39.9285	9.017619
Convex	25	29.7779	25.7619	15.58891	26.9458	10.51036
	64	47.0911	42.6074	10.52329	43.9118	7.240195
	100	73.4571	63.7719	15.18725	65.2593	12.56189
Concave	25	13.1057	11.5909	13.06887	11.6725	12.27843
	64	21.8558	19.5947	11.53934	20.3445	7.428543
	100	33.6182	28.6033	17.53259	28.7654	16.87027

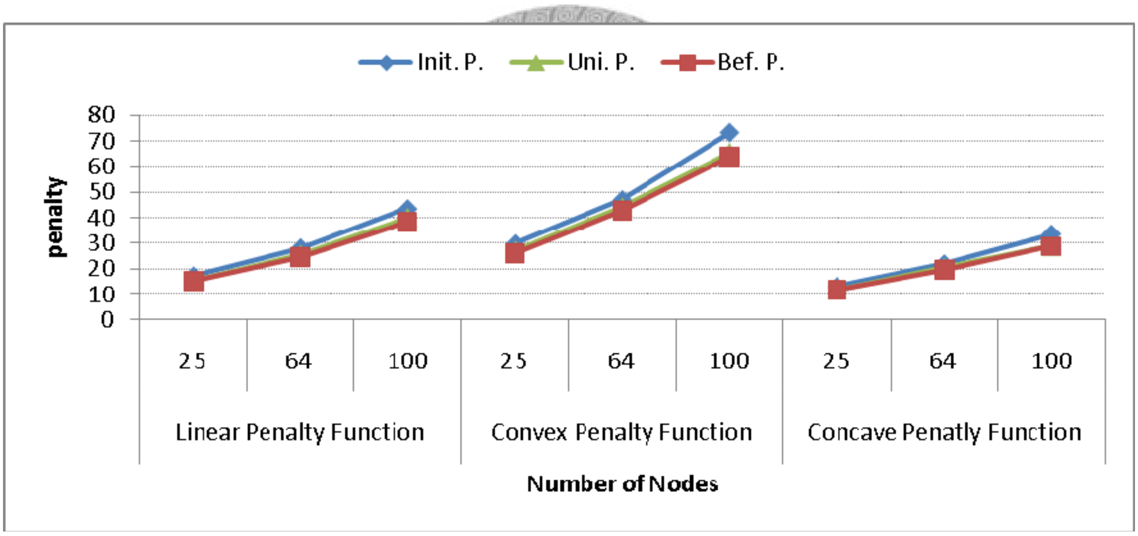


Figure 4-8. The Improvements under Different Numbers of Nodes

strategy gets more improvement than uniform re-distribution strategy. That is because the uniform re-distribution does not consider the important of each node and may allocate the extracted budget to nodes which gain less improvement.

The two re-distributions' improvement ratios to initial value are 14% and 11%, respectively.

Chapter 5 Conclusion

5.1 Summary

With the convenience of Internet, most of network services are indivisible from our daily lives and some of them need to offer the high Quality-of-Service (QoS) requirements of transmissions. However, the transmissions may be interfered with malicious attackers. The network administrator has to endeavor his/her best to guarantee the QoS of each transmission and to minimize the penalty caused by QoS violations.

The main contribution of this research is that we proposed mathematical programming problems which are the ARRAS and the NPDRAS problems to well-model the mutual behavior between a network administrator and an attacker in the real world. We then develop the Lagrangean relaxation-based algorithm to solve the ARRAS problem and exploit the solutions of the ARRAS problem and the adjustment procedure to obtain the near optimal defense strategy for the NPDRAS problem. Most importantly, the obtained solution for NPDRAS problem provides the useful indicator of defense strategies to the network administrator to strengthen the robustness of the network.

Moreover, we use a concave defense capability function in the computational experiments. It is more reasonable and to simulate the real situation more actually. From the experiment results, we can make some observations:

- The degree-based defense budget distribution is more robust than uniform.
- The best budget allocation ratio to defense capability and capacity is uncertain.
- The marginal penalty declines with the enlargement of attack budget.

5.2 Future Work

We address three issues that can be researched further:

- **The requirements of QoS**: In the thesis, we take bandwidth, end-to-end delay, and multiple paths to be QoS requirements. However, other possible QoS requirements should be considered, such as delay-jitter, packet loss and so forth. Besides, we only adopt the unique delay violation, but the combined violations should be considered for approaching the real situation more actually.

- **The experiences of the attacker**: The attacker may gains and accumulates experiences when he/she compromises a node, and further uses the less attack budget to compromise other nodes. Thus, the experiences of the attacker may be considered into the network attack-defense problem in the future.
- **The attack types of the attacker**: In our research, the capacity attack is the only attack type of the attacker, but there are several different attack types in the real world, such as Distributed Denial of Service (DDoS). That is to say, the combined attacks may be taken into account as possible as we can in the future.
- **The network topology**: The network topology may be the important factor to affect the defense capability of the network. The rich connectivity of nodes can benefit not only the data transmission but also the convenience for attack. Therefore, the alternative of setting a link is another discussion for resisting attacks in the realm of network planning.

References

[1] http://en.wikipedia.org/wikipedia.org/wiki/The_Third_Wave_%28book%29.

[2] A. Shaikh and K. Shin, "Destination-Driven Routing for Low-Cost Multicast," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 373-381, April 1997.

[3] B. Zhang and H. T. Mouftah, "Destination-Driven Shortest Path Tree Algorithms," *Journal of High Speed Networks*, Vol. 15, pp. 123-130, 2006.

[4] R. Richardson, "2007 CSI Computer Crime and Security Survey," 2007.

[5] P. Tarvainen, "Survey of the Survivability of IT Systems," *the 9th Nordic Workshop on Secure IT-systems*, November 2004.

[6] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. A. 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).

[7] B. Wang and J. C. Hou, "Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols," *IEEE Network*, Vol. 14, No. 1, pp. 22-36, January/February 2000.

[8] P. Paul and S. V. Raghavan, "Survey of Multicast Routing Algorithms and Protocols," *Proc. of the 15th Int. Conference on Computer Communication*, pp.

902-926, 2002.

[9] A. S. Tanenbaum, Computer Networks, 4th Edition, 2004.

[10] L. Wei and D. Estrin, “The Trade-offs of Multicast Trees and Algorithms,” *Proc. Third Int. Conference on Computer Communications and Networking,* pp. 17-24, 1994.

[11] A. Fei and M. Gerla, “Receiver-Initiated Multicasting with Multiple QoS Constraints,” *Proc. IEEE INFOCOM 2000*, Vol. 1, pp. 62-70, 2000.

[12] Z. Wang and J. Crowcroft, “Quality-of-Service Routing for Supporting Multimedia Applications,” *IEEE JSAC*, Vol. 14, pp. 1228-1234, September, 1996.

[13] H. C. Cheng, “Multicasting Algorithms in Multimedia Networks,” *Department of Information Management*, *National Taiwan University*, July 2005.

[14] T. Turletti and J. C. Bolot, “Issues With Multicast Video Distribution in Heterogeneous Packet Networks,” *Proc. The 6th International Workshop on Packet Video*, pp. F3.1-F3.4, 1994.

[15] V. R. Westmark, “A Definition for Information System Survivability,” *Proc. Of the 37th IEEE Hawaii International Conference on System Sciences*, Vol. 9, 2004.

[16] S. Louca, A. Pitsillides, and G. Samaras, “On Network Survivability Algorithms Based on Trellis Graph Transformations,” *Fourth IEEE Symposium on Computers and Communications* (*ISCC ”99*), pp. 235-243, July 1999.

[17] 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.

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

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

[20] A. M. Geoffrion, “Lagrangean Relaxation and its Use in Integer Programming,” *Mathematical Programming Study*, Vol. 2, pp. 82-114, 1974.

[21] F. Y. S. Lin and J. R. Yee, “A New Multiplier Adjustment Procedure for the Distributed Computation of Routing Assignments in Virtual Circuit Data Networks,” *ORSA Journal on Computing*, Vol. 4, No. 3, pp. 250-266, 1992

簡歷

姓名：謝孜謙

出生地：台灣 台北市

生日：中華民國七十二年十一月二八日

學歷：九十三年九月至九十五年六月

國立台灣科技大學資訊管理學系學士

九十五年九月至九十七年七月

國立台灣大學資訊管理研究所碩士