國立臺灣大學管理學院資訊管理學系

碩士論文

Department of Information Management

College of Management

National Taiwan University

master thesis

考量危害事件與惡意攻擊下系統脆弱度最小化之近似最佳化冗餘配置策略

A Near-Optimal Redundancy Allocation Policy to Minimize System Vulnerability against Hazardous Events and Malicious Attacks

江坤道

Derek Kun-Dao, Jiang

指導教授：林永松 博士

Advisor: Frank Yeong-Song, Lin Ph.D.

中華民國 96 年 7 月

July, 2007

國立台灣大學資訊管理研究所碩士論文

指導教授：林永松 博士

考量危害事件與惡意攻擊下系統脆弱度最小化之近似最佳化冗餘配置策略

A Near-Optimal Redundancy Allocation Policy to Minimize System Vulnerability against Hazardous Events and Malicious 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論文
學位所需條件之一部分

研究生：江坤道 撰

中華民國九十六年七月

國立臺灣大學碩士學位論文

口試委員會審定書

考量危害事件與惡意攻擊下系統脆弱度最小化之近似最佳化冗餘配置策略

本論文係江坤道 君（學號 R94725018）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國 96 年 07 月 19 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

林永松

呂俊賢　　顏宏旭

孫雅麗　　祝國忠

所　　長：孫雅麗

謝辭

撰寫謝辭的心情是矛盾的。它代表著學術研究的訓練暫告一個段落，此時的心境是激昂、澎湃的，它也宣告著必須和所有深厚的情誼、熟稔的人事物別離，此時的心情是難過、不捨的。但我知道，它不過是個換節符號罷了，不代表結束，而是下一個人生階段的開啟。

兩年雖然短暫，但所接受的訓練，不管是專業的學術能力，或是對自我要求的負責，乃至人生視野的開拓，都讓我成長許多。其中，最要感謝的是林永松老師的教誨與指導，每當我遇到研究上的瓶頸時，老師對於問題的系統化剖析，字字珠璣的真知灼見，讓我得以有方向地繼續往下探究。雖然其中有甘有苦、峰迴路轉，但這也就是自我訓練提升的必經過程，讓我更加深切地體察到老師對於治學的嚴謹與要求。而老師對於人與人之間的相處態度，更是讓我學到虛懷若谷的謙卑該如何去親身實踐，而非淪於紙上空談。同時也要非常感謝本系的孫雅麗老師、輔大資工的呂俊賢老師、世新資管的顏宏旭老師、景文資管的祝國忠老師，感謝你們在口試過程中給我的寶貴建議，讓本論文得以更趨完整與豐富。

此外，要感謝實驗室所有的學長姐與學弟們。特別是柏皓學長在資訊安全領域的啟蒙，在人生過程的建議，在生活上點點滴滴的協助，讓我充滿了感激與謝意。也要萬分感謝學弟們，辛苦地處理實驗室的事務與口試過程的協助，讓我們能夠更加無憂無慮的準備研究。接著，當然不可以忘記我親愛的戰友們：俊維、承賓、雅芳、翊恆、岦毅、怡孜。是你們讓這兩年添加了更多豐富的色彩，讓碩士階段變得更加繽紛亮麗，讓我生命中增加了更多值得珍惜的朋友。

最後要感謝的是我的父母親江秋期和詹于嫺，是你們的支持讓我有繼續走下去的動力與勇氣，也感謝我的大姐、二姐和小弟，謝謝你們在我受挫時的鼓勵，讓我更加堅定地去面對失敗。也很感謝 IBM 的 Eric Chao 與 Ken Yang，你們的提攜讓我這兩年可以過的更加平順。當然，最壓軸的感謝要獻給我的皮，是妳的陪伴與支持，才得以讓我順利劃下謝誌的這個最後句點。

論文摘要

論文題目：考量危害事件與惡意攻擊下系統脆弱度最小化之近似最佳化冗餘配置策略

作者：江坤道　　　　　　　　　　　　　　　　　　　　　九十六年七月

指導教授：林永松 博士

現代組織企業越來越倚重資訊科技來協助日常的營運作業。然而，這樣的依賴性卻是建立在危害事件發生頻繁且惡意攻擊層出不窮的環境下，任何的網路斷線或是機器故障都會造成嚴重的經濟損失。因此，為了達到持續性服務的目標，我們提出一個植基於冗餘配置的方法，期望將潛在威脅發生的可能性降到一個可以接受的程度。

在本論文中，我們將攻防雙方的戰役模擬成一個兩階的非線性整數規劃問題。在內層問題中（ARS 模型），攻擊者透過分配有限的攻擊能量來最大化網路元件面對危害事件的脆弱度。相反地，在外層問題中（RAPMA 模型），防守者嘗試在有限的預算限制下，透過冗餘元件的適當部署來最小化攻擊者所帶來的傷害。其中，我們發展一個以拉格蘭日鬆弛法為基礎的演算法來快速地解決此數學規劃問題。

關鍵字：冗餘配置問題、網路最佳化、數學規劃、資源配置、拉格蘭氏鬆弛法、網路脆弱度、網路存活度

THESIS ABSTRACT

DEPARTMENT OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME: DEREK KUN-DAO, JIANG MONTH/YEAR: July 2007

ADVISOR: FRANK YEONG-SUNG, LIN

A Near Optimal Redundancy Allocation Policy to Minimize the Vulnerability against Hazardous Events Considering the Impact of Intelligent Attacks

Modern organizations have increasingly relied on information technology to facilitate daily business operations. However, the dependency is built upon an environment where hazardous events happen frequently and malicious attacks emerge in an endless stream. Any network disconnections or failure of machines may result in serious economic lost. Therefore, to attain the objective of "continuity of services", we propose an approach based on redundancy allocation to reduce the possibility of threats occurring to an acceptable degree.

In the thesis, we formulate a "battle" between the attacker and the network into a two-level programming problem. In the inner problem (ARS model) an attacker allocates the limited attack powers to maximize the vulnerability of components against hazardous events. Contrarily, in the outer problem (RAPMA model) a defender attempts to minimize the damages by deploying redundant components appropriately with the limited budgets. We develop a Lagrangean Relaxation-based algorithm to solve the programming problem efficiently.

Keywords: Redundancy Allocation Problem, Network Optimization, Mathematical Programming, Resources Allocation, Lagrangean Relaxation, Network Vulnerability, Network Survivability

Table of Content

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

Modern organizations have been increasingly reliant on information technology, especially the Internet, to facilitate their daily business operations [3]. Nevertheless, it is noteworthy that the development of Internet brings about not only convenient access to information, but also potential crises. For a profit business, any failure of fiber connections or machines may result in extensive economic lost and even uncountable damages to reputation. As a result, it has become an extremely important issue to design a network configuration or a recovery plan which supports continuous services in the case of hazardous events occurring.

The goal of delivering continuous services, however, is a rigorous challenge, since we are living in a world where is full of potential risks. According to the CSI/FBI 2006 report [20], the organizations have invested large portion of IT budgets to information security activities to prevent from malicious attacks and cybercrime. This phenomenon paints the picture that the importance of information security has drawn much more attention than before. On the other hand, it also reveals a fact that the potential risks have become a constantly evolving threat to business operations.

Apart from the threat incurred by malicious attacks, the hazardous events, such as earthquake, flooding, blizzard, terrorist attack, and information warfare, are other strong adversaries to information security and continuous services. Take 9/11 attacks for example, this disaster had a significant economic impact on the United States and world markets. The New York Stock Exchange (NYSE), the American Stock Exchange (ASE) and NASDAQ did not open on September 11 and remained closed until September 17, because member firms, customers and markets were unable to communicate due to major damage to the information exchange facility near the World Trade Center. This painful experience has completely displayed the urgent requirements on a survivable network configuration or a recovery plan for organizations.

As a consequence, security, which traditionally puts much emphasis on information confidentiality, has been evolving into a brand-new concept, survivability, which mainly focuses on the availability of system and continuity of service [2]. The essential transformation from traditional network security toward the novel concept of survivability has involved not only the change of measurements of security risk, but also the shift in solution approaches. As to the measurements, most researches in computer security focuses on how to propose a mathematical model to quantify the security degree. Generally, the analysis techniques in common to evaluate the system security can be divided into three types, including combinatorial methods, model checking and state-based stochastic methods [13]. A variety of related performance indicators, such as reliability, availability, dependability, and survivability, are proposed to systematically and concretely derive the value of present security degree [5] [6] [11] [12]. Meanwhile, several approaches are developed to construct a robust network immune to equipment failures via rerouting mechanism, survivability constraint, or redundancy allocation [3] [8] [9].

Information security consists of not only technology application, but also strategies management. From the perspectives of business, it has been expanded toward risk management that requires the participation of an organization as a whole (executive manager, security experts, application domain experts, and other stakeholders) to protect mission-critical systems from cyber-attacks, failures, and accidents [2]. Therefore, in this thesis, we try to develop a methodology concerning redundancy allocation in terms of risk management. The ultimate goal is to reduce the occurring possibility of potential threats to an acceptable degree with limited budgets; meanwhile, ensure the continuity of services.

1.2 Motivation

Nowadays, existing services are mostly web-based systems, which exchange or retrieve data through network. Unfortunately, the infrastructure of network is built upon an environment where hazardous events occur frequently and malicious attacks emerge in an endless stream. In order to diminish the impacts incurred by internal and external jeopardy, an organization must spend a large volume of investments on security mechanism, like, firewall, intrusion detection systems, and intrusion prevention systems. However, for an organization, the available resources are so limited that we have to dispute over every detail of budget allocation.

Based on the consideration of finite resources, the question regarding risks control is not whether organizations need more security, but how much to spend for added security. Accordingly, we look forward to proposing an optimization-based framework to maximize the return of limited budgets. With the assistance of the approach, the service providers are capable of planning a resource allocation policy to support continuous services.

For an organization, the deployment of redundant components is one of the best strategies to reduce potential risks due to the advantage of fault tolerance. So-called fault tolerance is a capability of a system to respond gracefully to an unexpected hardware or software failure. There are many levels of fault tolerance, the lowest being the ability to continue operation in the case of hazardous events occurring. Many fault-tolerance computer systems are configured in hot-standby mode and mirror all operations, that is, every operation is performed on two or more duplicate systems, so if one fails the other can take over right away. The nature of redundancy meets the requirements of continuous services; thus, we attempt to design a scheme which adopts the concept of redundancy as the core.

In the realm of reliability, redundancy allocation problem (RAP) has been widely studied for a long time [3] [4] [5] [6] [7]. Those studies mainly focus on parallel-system design or recovery plan without extending to network configuration. Besides, they did not consider the impacts of malicious attacks, which have different characters from natural disasters. Therefore, we want to propose a novel redundancy allocation problem considering the impacts of malicious attacks and being applied to network configuration design. To the best of our knowledge, we are the first one to integrate attacking behavior model with traditional redundancy allocation problem.

1.3 Literature Survey

Conceptually, risk management and survivability have strongly positive correlation in the realm of information security. Risk management focuses on reducing the potential threats to an acceptable degree, whereas the concept of survivability is an indicator to concretely quantify the “degree.” As a consequence, we discuss them together. Moreover, the traditional RAP is also discussed here.

1.3.1 Risk Management

Security and cyber-terrorism have become increasingly important issues for organizations and the society. The Harmantzis et al [21] proposes a risk management framework from a bottom-up perspective, i.e. modeling the different types of attacks that an organization could experience. A quantitative model is presented to measure the economic impact of security risk. In addition to risk management, a further goal of this research is to apply data mining techniques to predict and prevent security attacks in an effective manner. Attack graphs or trees are increasingly formalized to be model for representation of system security based on various attacks. In [22], Dantu et al use attack graph to calculate vulnerabilities and risk of a critical resource in a given network topology. The procedure can be divided into five steps, including creating an attacker profile, constructing attack graph according to the corresponding file, labeling attack paths with behavior attributes, computing risk, and optimizing the risk levels based on the final outcomes. By executing these five steps repeatedly, an optimal security configuration might be obtained, eventually.

In [23], the risk management is conducted in a different perspective from [22]. A new approach based on defense trees is proposed to evaluate the security investments. Bistarelli et al present a mixed qualitative and quantitative approach for evaluation of IT security investments. For this purpose, they model security scenarios by using defense trees, an extension of attack trees with attack countermeasures and use economic quantitative indexes for computing the defender’s return on security investment and the attacker’s return on attack. This approach can be used to evaluate effectiveness and economic profitability of countermeasures as well as their deterrent effect on attackers, thus providing decision makers with a useful tool for performing better evaluation of IT security investments during the risk management process.

1.3.2 Survivability

In recent years, there have been dramatic changes in the character of security problems, in their technical and business contexts, and in the goals and purposes of their stakeholder. As a consequence, many of the assumptions underlying traditional security technologies are no longer valid. Survivability provides a new technical and business perspective on security. In [2], survivability has been defined as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents, where the term “system” is used in the broadest possible sense, and includes networks and large-scale “systems of system.”

Survivability becomes increasingly crucial, since large traffic volumes are multiplexed onto a single fiber. A single cable cut can affect incredibly large groups of users, leading to catastrophic socioeconomic effects. The Molisz [1] defines the network survivability function as the probability function of the percentage of total data flow delivered after failure and survivability attributes, which are the expected percentage of total data flow delivered after failure, the respective *p*-percentile values, the worst case survivability. Let ς denote a failure scenario. This scenario is the set of network components (subset of N_D out of N nodes, or subset of M_D out of M arcs) being not operational due to the catastrophic failure. Each ς occurs with a specific probability. Different scenarios ς may result in similar values of survivability measures. The survivability function is defined as $S(x)=\sum_{\varsigma:X(\varsigma)=x}P(\varsigma)$, where $X(\varsigma)$ (the random variable) equals to percentage of flow delivered after the failure according to scenario ς; and $P(\varsigma)$ equals to probability of scenario ς which is characterized by the percentage x of total data flow still delivered.

In [10], the goal is to assess the survivability of a system when it is subjected to a series of random incidents over time. For this reason, Moitra et al first need to model the process of occurrence of incidents from the point of view of a system or site that experiences this process over time. This is equivalent to a stochastic point process where incidents occur at random points in time; therefore stochastic point process is needed to simulate attack behavior. The survivability also depends on how the system responds to an incident. This will depend on the system configuration, that is, its design and defense mechanisms as defined above. Therefore, Moitra et al model this response as a function of the incident type and configuration. The model will involve a transition matrix that will give the probabilities of the system ending up in any of its possible states after experiencing an incident. These probabilities will depend on the incident type and system configuration. Next, the degree to which it has survived will have to be measured. This will be a function of the state in which it ends up and the amount of compromise that has occurred. For this purpose, Moitra et al develop some new survivability measures that take into account the different dimensions of survivability, that is, the different functionalities and services that can be compromised. The survivability is measured as:

SURV = (performance level at new state s) / (normal performance level)

Let $\varphi(s,k)$ be the degree to which the compromised function/service *k* has survived in state s, and let *w(k)* be the important level of function/service. Then one possible measure of survivability might be in the form of a weighted sum:

$$SURV(s) = \sum_{k} w(k) \times \varphi(s,k)$$

Previous quantitative models of security or survivability have been defined on a range of probable intruder behavior. This measures survivability as a statistic such as mean time to breach. However, this kind of purely stochastic quantification is not suitable. In [16], McDermott proposes an approach based on the most competent intruders the system is likely to face. It is assumed that the potential intelligent attackers will obtain more information about the target and be more likely to compromise the target as time goes by. Similarly, the defenders will learn some experience to adjust defense policy according to the previous attack behavior. As a result, defender should allocate more resources to resist those attackers who are the most competent.

In [14], Zeitlin attempts to formulate attack-defense scenario into a min-max integer resources allocation problem. Attacker tries to compromise as many targeted node as s/he can with limited *M* units of attack resources, whereas defender desires to minimize the damages by allocating finite *N* units of defense resources. Assuming x_i, y_i represent the attack powers and defense powers allocated on targeted node *i*. The damage to node *i* is intuitively defined as $\max\{x_i - q_i y_i;\ 0\}$, where q_i is the defense effect on node *i*. Therefore, the problem is formulated as follow:

$$
\begin{aligned}
&\min_{y} \max_{x} \sum\nolimits_{i=1}^{n} d_i \max\{x_i - q_i y_i ;0\} \\
&s.t. \\
&\sum\nolimits_{i=1}^{n} x_i = M;\ \sum\nolimits_{i=1}^{n} y_i = N;\ x_i, y_i \geq 0 \text{ integer}
\end{aligned}
$$

1.3.3 Redundancy Allocation Problem

The deployment of redundant components is often adopted to support the continuity of service in terms of risk management. As we mention earlier, a computing machine with redundant components is highly possible surviving in the case of hazardous events. Hence, with the assistance of redundancy allocation policy, an organization can assure the effective control over potential threats in the environment.

The objective of the RAP is to determine an optimal system design to maximize system reliability, availability, and survivability given constraints on the system. It is a difficult non-linear integer programming problem that has been extensively studied because it is widely applicable and relevant, but also because it is challenging to solve. The general RAP is classified as NP-hard [17] in terms of computational complexity due to its nonlinearity, nonconvexity, and integrality. So far, there are many works discussing the RAP considering different scenarios, assumptions, constraints, and solution approaches.

In [3], a discrete optimization model is proposed to allocate redundancy to critical IT functions for disaster recovery planning. The objective is to maximize the overall survivability of an organization’s IT functions by selecting their appropriate redundancy levels. A solution procedure based on probabilistic dynamic programming is presented to optimally solve the problem. It is noteworthy that, in [3], the number of redundant components for a specific IT function is restricted to be exactly one. However, in the model of David et al [4], there are multiple, functionally equivalent components available to be used in the system. The design can include a single components selection for each subsystem, or there may be multiple components selected and arranged in parallel. In [4], a new multiple weighted objectives (MWO) heuristic has been developed by transforming the problem into one, which is so-call surrogate problem, with the simultaneous objectives of maximizing each of the subsystem reliability for a series-parallel system.

The Ha et al [7] proposes a new heuristic based on tree structure to solve the general RAP in reliability optimization. The tree heuristic can obtain several local optimal by branching off the main searching path when some criterions are satisfied. Then, the best local optimal is selected for the final solution. The tree heuristic is a simple, efficient, iterative heuristic for any integer nonlinear programming problems with increasing constraint functions. Iterative heuristics are normally trapped in a local optimum. However, the tree heuristic can overcome local optimal by branching the solution path. All of works [3] [4] [7] regarding RAP above formulate the model as a maximization problem with the objective of maximizing the system reliability. The Jose et al [5] formulates RAP in a different perspective. In [5], the RAP is formulated with the objective of maximizing the minimum subsystem reliability for a series-parallel system. This is a new problem formulation that offers several distinct benefits compared to traditional problem formulation. Since time-to-failures of the system is dictated by the minimum subsystem time-to-failure, a logical design strategy is to increase the minimum subsystem reliability as high as possible, given constraints on the system. For some system design problems, a preferred design objective may be to maximize the minimum subsystem reliability. Additionally, the max-min formulation can serve as a useful and efficient surrogate for optimization problems to maximize system reliability.

1.4 Proposed Approach

As we described in the section 1.2, there are no any works regarding RAP considering the impacts of malicious attacks, which launch assaults on specific nodes and are restricted to “continuity constraints.” Therefore, we develop a practical and extensive model taking the impacts of hazardous events and malicious attacks into account at the same time. In the model, which is called “Redundancy Allocation Problem considering Malicious Attack, RAPMA” model, a min-max integer programming problem with nonlinearity is created to formulate a battle between attack and defender.

In our methodology, at first, we extract an inner problem, which is call

"Attacking Redundancy Strategy, ARS" model, concerning attacking behavior from the original model, and then solve it with Lagrangean Relaxation method. After the attacking decision is made, the attack policy is inputted to the RAPMA model to develop the near optimal redundancy allocation policy. Next, the attacker launches the attack again given the pervious redundancy allocation policy. Repeating the procedure until the solution reaches the balance status. A near optimal solution to redundancy allocation policy is eventually obtained. The solution procedure can be illustrated with Figure 1 below.

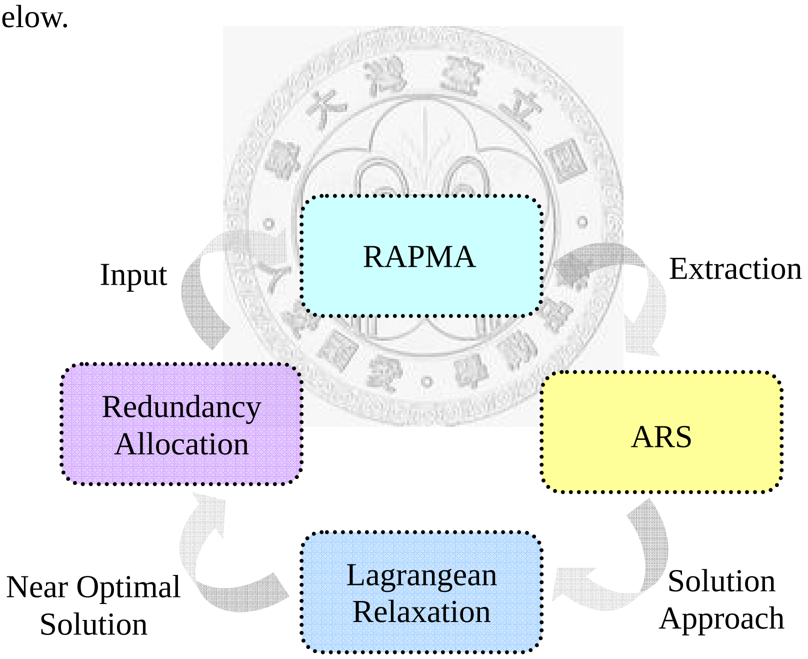


Figure 1 the Solution Procedure

1.5 Thesis Organization

The rest content of the thesis is organized as follows. In Chapter 2, the formulation of the RAPMA and the ARS problems are proposed. In Chapter 3, solution approaches to the AS problem and the DRAS problem are presented; in Section 3.1, solution approaches base on Lagrangean Relaxation are proposed.

Chapter 2 RAPMA and ARS Model

2.1. Problem Description

The problem we discuss is how to deploy the redundant components appropriately to reduce the vulnerability against hazardous events. In other words, we propose a methodology to raise the survivability of the whole network in the case of hazardous events occurring by redundancy allocation. Notably, we also consider the impacts of intelligent malicious attacks due to applicability and practicability. We adopt the concept of optimization to solve this redundancy allocation problem by formulating it as a mathematical model. In this model, there exists a wrestle between defender and attacker. They will dynamically adjust their resources allocation policies according to the decisions made by their opponent.

For attacker, s/he will try the best to compromise as many nodes as s/he can in limited attack powers. The ultimate objective of attacker is to weaken the resistance to the hazardous events instead of crippling the entire network. On the other hand, from the perspective of defender, s/he will choose redundant components in an advisable way to strengthen the capability of withstanding the damages incurred by hazardous events and malicious attacks. The goal of defender lies in providing continuous services by the deployment of redundancy. Basically speaking, a node with redundant components is more likely to survive when hazardous events occurring, since primary component can switch its function to those hot-standby redundancies to reduce the potential threats caused by hazardous events. To demonstrate the applicability and practicality of our model, two real scenarios fit in with our model are given below.

Scenario 1: The first scenario is about hardware attack prior to sequent hazardous events. The intruder launches a targeted attack, which will infect the computer with malicious program, to make the CPU in the status of high temperature. Once the power failure incurred by natural disasters makes the air conditioners of server room dysfunction, those infected computers may be shutdown due to CPU over-heat. Therefore, the entire system becomes more vulnerable to the natural disasters because of malicious attacks.

Scenario 2: The second scenario is about software attack prior to sequent hazardous events. The attacker intrudes the computer and manipulates the privileged configuration files of some services. Once the power cut incurred by natural disasters makes the computer rebooted, the service will not execute functionally because of wrong configuration. Hence, the goal of “continuous” service will be forced to “stop”.

In order to quantify the degree of damages after malicious attacks and hazardous events, we define two metrics, which are antithetic to each other, vulnerability and survivability. Vulnerability is a probability that at least one of the nodes in the network is dysfunction upon the occurrence of hazardous events, such as flooding, earthquake, tsunami, hurricane, tornado, and a large-scale of information war. Given a network topology, each node in the network is composed of just one primary component and several secondary redundancies. Considering the character of redundancy, it is assumed the probability V_{imd}, that presents a component *m* within a node *i* is conquered by the events *d*, is independent and the probability P_d of event *d* is known or can be estimated, where $i \in N, m \in r_i, d \in D$. We define a node is dysfunction while all components, including primary and redundancies, are compromised by hazardous events. As a result, vulnerability V_{id} of node *i* against event *d* can be calculated by multiplying associated V_{imd}. According to the definition about node vulnerability, the vulnerability V_d^* of the entire network against event *d* is also determined by the equations presented as follow.

Equation 1:

Node vulnerability V_{id} against event d= $\prod_{m \in r_i \text{ which are chosen as a component}} V_{imd}$, where $i \in N, d \in D$

Equation 2:

$$\textit{Entire vulnerability}\, V_d^* \textit{ against event } d = P_d \left(1 - \sum_{i \in N} \left(1 - \prod_{m \in r_i \text{ which are chosen as a component}} V_{imd} \right) \right),$$

where $d \in D$

It is noted that an approximation is used to derive the value of entire vulnerability V_d^* against event *d*. The equation 2 above utilizes the operation of summation to replace multiplication as an approximation to overestimate the probability of all nodes survive when event *d* occurring.

As mentioned previously, survivability is antithetic to vulnerability. We define the survivability as a probability S_{imd}, that a component *m* within a node *i* defies event *d* successfully. Based on the definition, we can find the relationship $S_{imd} = 1 - V_{imd}$. Therefore, the other metric, survivability, can be presented as follow. Notably, the equation 3 enforces the definition that a node fails against event *d* only when all its chosen components fail at the same time upon the occurrence of event *d*.

<u>Equation 3</u>:

Node survivability S_{id} *against events d*= $\left(1 - \prod_{m \in r_i \text{ which are chosen as a component}} V_{imd}\right)$, where

$i \in N, d \in D$

<u>Equation 4</u>:

Entire survivability S_d^* *against events d*= $P_d \sum_{i \in N} \left(1 - \prod_{m \in r_i \text{ which are chosen as a component}} V_{imd}\right)$, where

$d \in D$

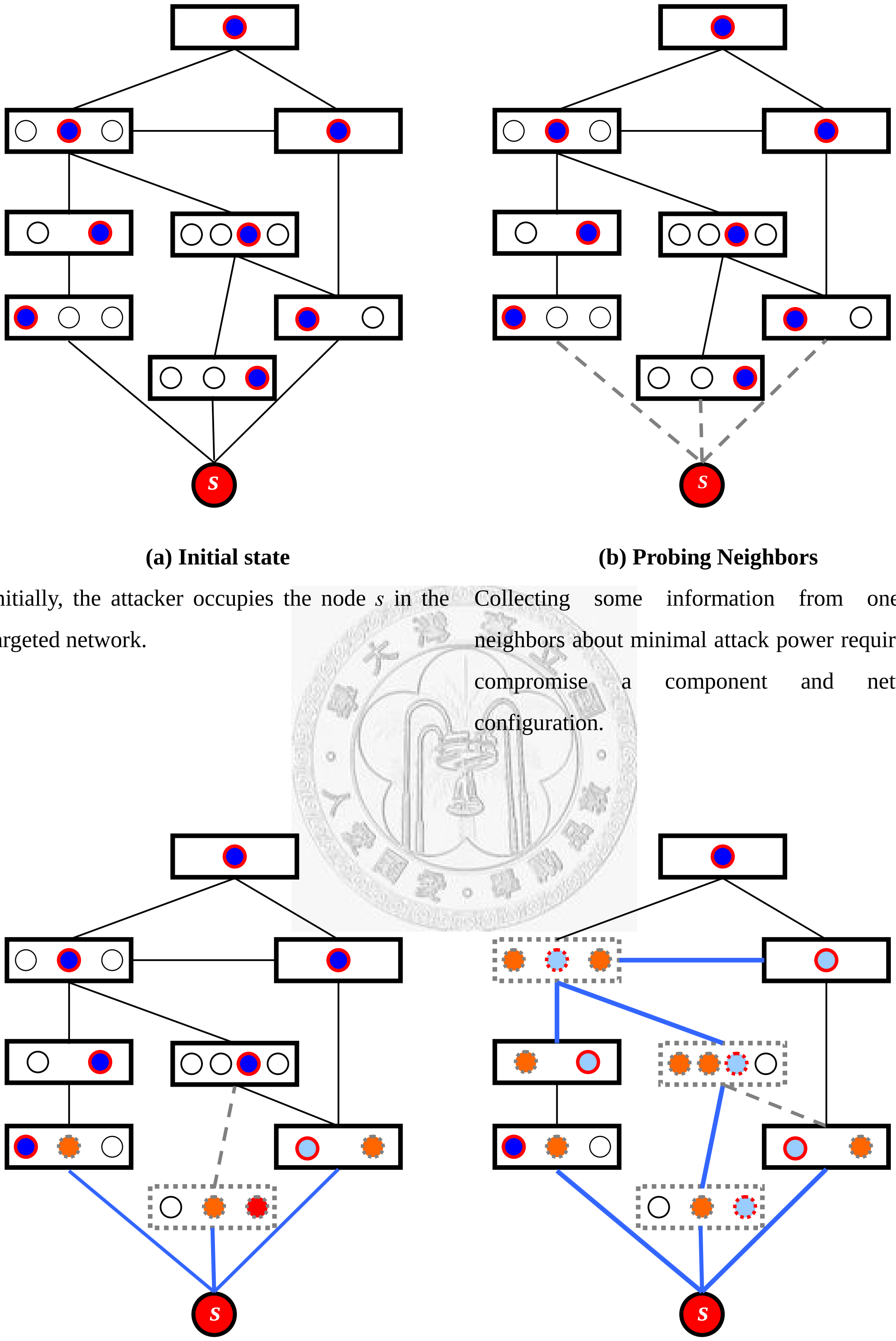
2.2. Problem Formulation of RAPMA Model

Obviously, the conflict between those two roles in the battle is that the intelligent attacker desires to maximize the vulnerability against hazardous events; however, the defender attempts to minimize the maximized vulnerability. As a consequence, we develop a min max integer programming mathematical model to formulate this scenario. By solving this complicated problem, it is expected to obtain a near optimal redundancy allocation policy to protect the targeted network from being devastated by hazardous events.

Assuming both attack and defender possess complete and correct information about the targeted network, including the topology configuration, the network size, and the minimal attack powers required to compromise a component. The attacker will take advantage of that information to determine the targets on which s/he intends to launch assaults. This assumption leads to an adverse situation for defender. Nevertheless, by considering the worst case, we can propose a more robust scheme to develop our redundancy allocation policy. The defender will make use of that information to adjust the defense policy in response to the malicious attacks.

Beside, only node attacks are consider, since they are more common in real world. It is noteworthy that a node is regarded as an AS-level domain and all edges are regarded as inter-domain connections. Therefore, attacker should compromise all nodes on the path linked to the targeted node if s/he wants to reach the targeted node. This is the so-called “continuity constraint.” Moreover, a node is compromised if and only if the primary component within it is also compromised by allocating attack powers more than the predefined minimal threshold $\hat{g}_{im}\left(c_{im}\right)$. Generally speaking, the more a component costs, the more robust it is. Based on the principle, the minimal threshold is designed to be positively proportional to cost. We illustrate the attack behavior with descriptions and these figures below.

Fist of all, the intelligent attacker occupies the position s in the network (Figure 2.a). After that, s/he tries to collect some useful information from one-hop neighbors (Figure 2.b), i.e. those nodes which connect directly with the initial position s. The information that attacker is interested in includes the minimal attack powers required to compromise the primary component, the increasing degree of vulnerability incurred by allocating attack power to secondary redundancy, and network configuration. The attacker will select the targets and apply attack powers to them based on the information s/he obtained (Figure 2.c). Repeating those procedures above until all attack powers are exhausted (Figure 2.d). Finally, an attack tree is constructed to maximize the total vulnerability by this intelligent attacker (Figure 2.e).



(a) Initial state

Initially, the attacker occupies the node *s* in the targeted network.

(b) Probing Neighbors

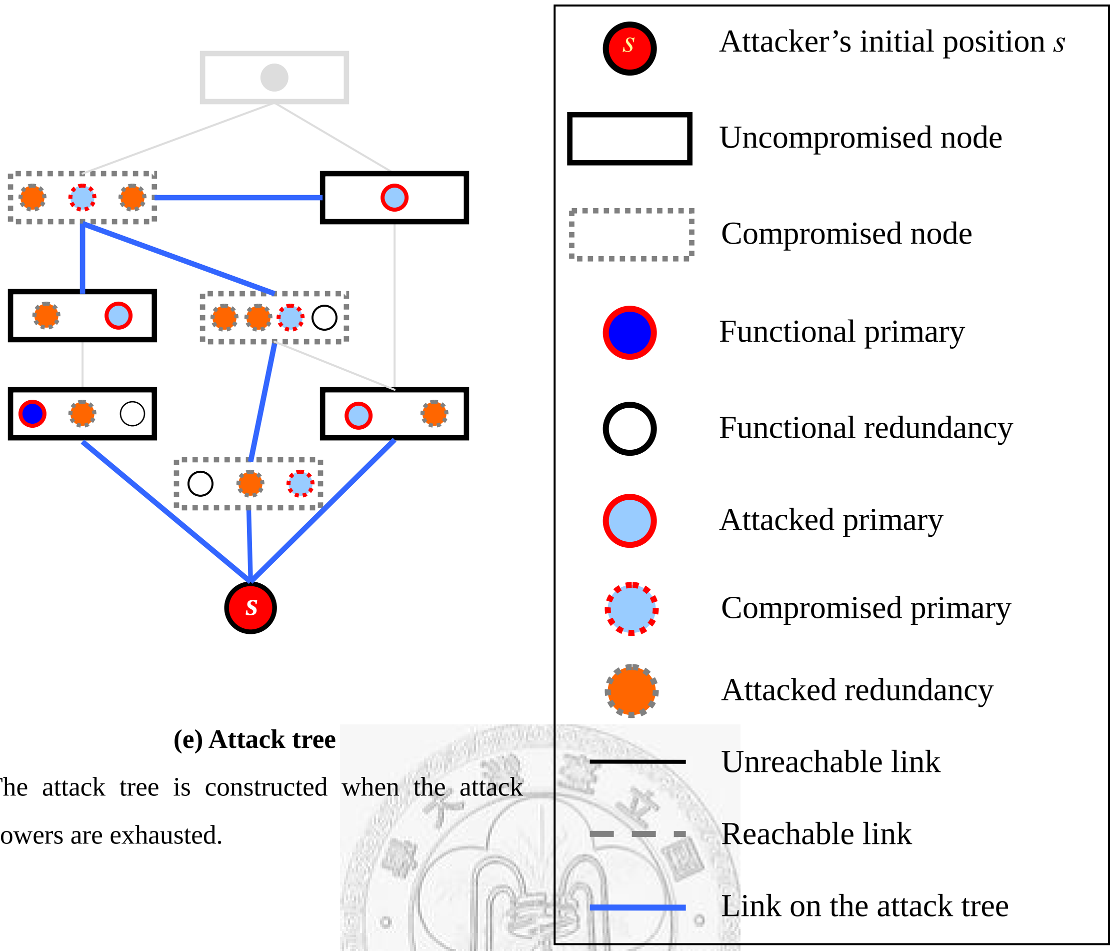
Collecting some information from one-hop neighbors about minimal attack power required to compromise a component and network configuration.

(c) Attacking a target

Determining the attack path based on the collected information and applying attack powers to the targeted components.

(d) Post-attack network state

Continuing launching attacks till the limited attack powers are exhausted.



(e) Attack tree

The attack tree is constructed when the attack powers are exhausted.

Figure 2. (a) Initial state (b) Probing Neighbors (c) Attacking a target (d) Post-attack network state (e) Attack tree

So far, we have defined an optimization-based problem with its specific assumptions, objectives and related parameters. All information is listed in Table 1 below.

Table 1 Problem Assumption and Description of RAPMA Model

Assumptions
• The attacker’s objective is to maximize the total vulnerability against hazardous events by malicious attacks.
• The defender’s objective is to minimize the total vulnerability against hazardous events by redundancy allocation.
• Both attacker and defender have complete and correct information about the network topology.
• Both attacker and defender have resource budget limitations.
• Only node attack is considered.
• Only malicious attacks are considered.
• Only AS-level networks are considered.
• A node is only subject to attack if a path exists from attacker’s position to that node, and all the intermediate nodes on the path have been compromised.
• “A node is compromised” if and only if the primary component deployed to it is compromised by allocating more attack power than the minimum level.
• Failures of individual components are *s*-independent.
• All redundant components are in a hot-standby state.
• All redundant components which are compromised by attacker are never repaired or detected.

Given

- Defense resource budget *B*
- Attack resource budge *A*
- The minimum attack power required to compromise a component.
- Attacker's position *s*, which is connected to the target network
- The network topology and the network size
- The estimated probability of events *d* occurring
- All available redundant components for node *i* to support operating function and provide failure tolerance.

Objective

- For defender, utilizing limited resources to minimize the maximized vulnerability against hazardous events.
- For attacker, utilizing limited attack powers to maximize the vulnerability against hazardous events.

Subject to

- The total defense cost must be no more than B
- The total attack cost most be no more than A
- The node to be attacked must be connected to the existing attack tree

To determine

- Defender: redundancy allocation strategy
- Attacker: which nodes to attack

As we mentioned earlier, we formulate this problem as a min-max integer programming problem. All notations of given parameters used in this model are listed in Table 2 below.

Table 2 Given Parameters of RAPMA Model

Given Parameters	
B	Total available budget under defender's control
N	The index set of nodes in the network
A	Total available resources under attacker's control
W	The set of all OD-pairs, where origin is node s where attacker occupied and the destinations are the nodes i in the given network, where $i, s \in N$
D	The index set of all potential hazardous events with probability P_d, where $P_d \in (0,1), \sum_{d \in D} P_d = 1$
r_i	The index set of all redundant components which provide the same operating function as node i, where $i \in N$
P_w	The index set of all candidate path for an OD-pair w, where $w \in W$
δ_{pi}	The indicate function which returns 1 if node i is on path p; 0 otherwise, where $i \in N, p \in P_w, w \in W$
$level_i$	The redundant level of node i, where $i \in N, level_i \geq 0$
c_{im}	The cost of redundant component m for node i, where $i \in N, m \in r_i$
$\hat{g}_{im}(c_{im})$	The threshold of the attack power required to compromise component m for node i, where $i \in N, m \in r_i$

There are some points regarding those parameters to be clarified first. We utilize historical data of hazardous events to estimate the probability of events d occurring. Considering the scalability and flexibility of our model, we introduce a parameter $level_i$ defined by defender. For core nodes in the given network, defender is capable of determining the minimal redundant levels of those core nodes to ensure they are robust enough to resist hazardous events. The value of $\hat{g}_{im}(c_{im})$ is a concave function governed by the attack powers. Besides, it is assumed that the defender is completely aware of those given parameters, but the attacker only has a priori knowledge of *N*, *A*, and *B*. All notations of decision variables are listed in Table 3.

Table 3 Decision Variables of RAPMA Model

Decision Variables	
α_{im}	1 if redundant component *m* for node *i* is selected as primary to support operating function; 0 otherwise, where $i \in N, m \in r_i$
β_{im}	1 if redundant component *m* for node *i* is selected as secondary one to provide failure tolerance; 0 otherwise, where $i \in N, m \in r_i$
g_{im}	Attack power applied to redundant component *m* for node *i*, where $i \in N, m \in r_i$
y_i	1 if node *i* is compromised, that is, the attack power allocated to the primary component is greater than the threshold, $\hat{a}_{im}(c_{im})$; 0 otherwise, where $i \in N$
x_p	1 if path *p* is selected as attack path; 0 otherwise, where $p \in P_w, w \in W$
$f_{imd}(g_{im})$	The vulnerability of redundant component *m* for node *i* against events *d*, where $i \in N, m \in r_i, d \in D, f_{imd}(g_{im}) \in (0,1)$

The mathematical model (IP 1) of our problem is completed formulated and shown in the next page.

Objective function

$$\min_{\alpha_{im},\beta_{im}} \max_{g_{im}} \sum_{d\in D} p_d \left(1 - \sum_{i\in N} \left(1 - \prod_{m\in r_i} f_{imd} \left(g_{im} \right)^{\alpha_{im}+\beta_{im}} \right) \right) \qquad \textbf{(IP 1)}$$

Subject to

$$\sum_{w\in W} \sum_{p\in P_w} x_p \delta_{pi} \le \left(|N| - 1 \right) y_i \qquad \forall i \in N \qquad \text{(IP 1.1)}$$

$$\sum_{p\in P_w} x_p = y_i \qquad \forall i \in N, w = (s, i) \qquad \text{(IP 1.2)}$$

$$\sum_{p\in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(IP 1.3)}$$

$$\sum_{m\in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}\left(c_{im} \right)} \ge y_i \qquad \forall i \in N \qquad \text{(IP 1.4)}$$

$$x_p = 0 \ or\ 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP 1.5)}$$

$$y_i = 0 \ or\ 1 \qquad \forall i \in N \qquad \text{(IP 1.6)}$$

$$\alpha_{im} = 0 \ or\ 1 \qquad \forall i \in N, m \in r_i \qquad \text{(IP 1.7)}$$

$$\beta_{im} = 0 \ or\ 1 \qquad \forall i \in N, m \in r_i \qquad \text{(IP 1.8)}$$

$$\alpha_{im} + \beta_{im} \le 1 \qquad \forall i \in N, m \in r_i \qquad \text{(IP 1.9)}$$

$$\sum_{m\in r_i} \alpha_{im} = 1 \qquad \forall i \in N \qquad \text{(IP 1.10)}$$

$$\sum_{m\in r_i} \beta_{im} \ge level_i \qquad \forall i \in N \qquad \text{(IP 1.11)}$$

$$0 \le \sum_{m\in r_i} c_{im} \left(\alpha_{im} + \beta_{im} \right) \le B \qquad \forall i \in N \qquad \text{(IP 1.12)}$$

$$\sum_{i\in N} \sum_{m\in r_i} c_{im} \left(\alpha_{im} + \beta_{im} \right) \le B \qquad \text{(IP 1.13)}$$

$$\sum_{i\in N} \sum_{m\in r_i} g_{im} \le A \qquad \text{(IP 1.14)}$$

$$0 \le g_{im} \le A \qquad \forall i \in N,\ m \in r_i \qquad \text{(IP 1.15)}$$

Explanation of RAPMA Model

- Objective function: The objective is to minimize the maximized vulnerability against hazardous events. This is also the battlefield of attack and defender. In the ARS model, the attacker tries to maximize the vulnerability by determining the targets and attack powers. For defender, the ultimate goal is to minimize the total vulnerability by selecting redundant components to provide continuous services. Besides, it is worth accentuating again that a node fails against events *d* only when all its chosen components fail at the same time upon the occurrence of events *d*.
- Constraint (IP 1.1) restricts a node can be transited at most (|*N*|-1) times. This constraint also makes sure the presence of cycle on the attack tree never exists and all nodes on the attack tree are compromised.
- Constraint (IP 1.2) restricts a node is compromised if and only if there exists an attack path which leads to the target node.
- Constraint (IP 1.3) restricts there exists at most one attack path connecting the node *s* with the attacking target *i* in the given network.
- Constraint (IP 1.4) enforces a node is compromised if and only if the attack powers allocated onto it are more or equal to the minimal threshold.
- Constraint (IP 1.5) and constraint (IP 1.6) are integer constraints, both of which

restrict the value of x_p, y_i to be 0 or 1. Notably, constraints (IP 1.1) to (IP 1.6) also enforce the limitation, “a node is only subject to attack if a path exists from attacker’s position to that node, and all the intermediate nodes on the path have been compromised”, upon the model.

- Constraint (IP 1.7) and constraint (IP 1.8) are integer constraints, both of which restrict the value of α_{im}, β_{im} to 0 or 1.
- Constraint (IP 1.9) restricts the role of component is mutually exclusive. In other words, a component is selected to be either primary component, or secondary redundancy, or discarded.
- Constraint (IP 1.10) enforces there must be exactly one primary component deployed to node i in the network.
- Constraint (IP 1.11) enforces the number of secondary redundancies allocated to node i must be satisfied or more than the minimal redundancy level predefined by defender.
- Constraint (IP 1.12) restricts the boundary of budget allocated to node i. Obviously, the lower bound and upper bound are 0 and total budget B, respectively.
- Constraint (IP 1.13) is the total budget constraint, which enforces the total budget used by defender cannot be more than the total available budget B.

- Constraint (IP 1.14) is also the total attack powers constraint, which enforces the total attack powers used by attacker cannot be more than the total available attack powers A.
- Constraint (IP 1.15) restricts the boundary of attack powers allocated to component *m* for node *i*. Obviously, the lower bound and upper bound are 0 and total available attack powers A, respectively.

2.3. Problem Formulation of ARS Model

Generally speaking, it is usually intractable to solve a two levels problem with conflicting objectives directly, because we are not able to predict what will happen in this battle between attacker and defender. Accordingly, to deal with this difficulty, we use a two-phase approach.

First, we extract an ARS model from the original one. Fortunately, the ARS model is a maximization problem which formulates the behavior of attacker into a mathematical model. Then, after solving the ARS model, we input the result into the original one as given parameters to develop redundancy allocation policy. All assumptions are still applicable to ARS model. The given parameters are listed in Table 4 below.

Table 4 Given Parameters of ARS Model

Given Parameters	
B	Total available budget under defender’s control
N	The index set of nodes in the network
A	Total available resources under attacker’s control
W	The set of all OD-pairs, where origin is node *s* where attacker occupied and the destinations are the nodes *i* in the given network, where $i, s \in N$
D	The index set of all potential events with probability P_d, where $P_d \in (0,1), \sum_{d \in D} P_d = 1$
r_i	The index set of all redundant components which provide the same operating function as node *i*, where $i \in N$
P_w	The index set of all candidate path for an OD-pair *w*, where $w \in W$
δ_{pi}	The indicate function which returns 1 if node *i* is on path *p*; 0 otherwise, where $i \in N, p \in P_w, w \in W$
c_{im}	The cost of redundant component *m* for node *i*, where $i \in N, m \in r_i$
$\hat{g}_{im}(c_{im})$	The threshold of the attack power required to compromise component *m* for node *i*, where $i \in N, m \in r_i$
α_{im}	1 if redundant component *m* for node *i* is selected as primary to support operating function; 0 otherwise, where $i \in N, m \in r_i$
β_{im}	1 if redundant component *m* for node *i* is selected as secondary one to provide failure tolerance; 0 otherwise, where $i \in N, m \in r_i$

It is noteworthy that α_{im}, β_{im} in gray, both of which are originally defined as decision variables in RAPMA model, become given parameters in the ARS model.

Because information concerning redundancy allocation policy is known for attacker, s/he can determine the attack powers allocation policy to maximize total vulnerability.

Table 5 listed all decision variables used in the ARS model. Except α_{im}, β_{im}, they are the same as the decision variables of RAPMA model.

Table 5 Decision Variables of ARS Model

Decision Variable	
g_{im}	Attack power applied to redundant component *m* for node *i*, where $i \in N, m \in r_i$
y_i	1 if node *i* is compromised, that is, the attack power allocated to the primary component is greater than the threshold, $\hat{a}_{im}(c_{im})$; 0 otherwise, where $i \in N$
x_p	1 if path *p* is selected as attack path; 0 otherwise, where $p \in P_w, w \in W$
$f_{imd}(g_{im})$	The vulnerability of redundant component *m* for node *i* against events *d*, where $i \in N, m \in r_i, d \in D, f_{imd}(g_{im}) \in (0,1)$

The mathematical model (IP 2) of ARS model, that only formulates attack behavior, is given as follows in the next page. In this model, we transform the objective function from maximization into minimization without changing its optimality.

Objective function

$$\begin{aligned} Z_{IP2} &= \max_{g_{im}} \sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\prod_{m\in r_i} f_{imd}\left(g_{im}\right)^{\alpha_{im}+\beta_{im}}\right)\right) \\ &= \min_{g_{im}} - \sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\prod_{m\in r_i} f_{imd}\left(g_{im}\right)^{\alpha_{im}+\beta_{im}}\right)\right) \end{aligned} \qquad \textbf{(IP 2)}$$

Subject to

$$\sum_{w\in W}\sum_{p\in P_w} x_p \delta_{pi} \le \left(|N|-1\right) y_i \qquad \forall i \in N \qquad \text{(IP 2.1)}$$

$$\sum_{p\in P_w} x_p = y_i \qquad \forall i \in N, w=(s,i) \qquad \text{(IP 2.2)}$$

$$\sum_{p\in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(IP 2.3)}$$

$$\sum_{m\in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}\left(c_{im}\right)} \ge y_i \qquad \forall i \in N \qquad \text{(IP 2.4)}$$

$$x_p = 0 \; or \; 1 \qquad \forall p \in P_w, w \in W \qquad \text{(IP 2.5)}$$

$$y_i = 0 \; or \; 1 \qquad \forall i \in N \qquad \text{(IP 2.6)}$$

$$\sum_{i\in N}\sum_{m\in r_i} g_{im} \le A \qquad \text{(IP 2.7)}$$

$$0 \le g_{im} \le A \qquad \forall i \in N,\ m \in r_i \qquad \text{(IP 2.8)}$$

Explanation of ARS Model

- Objective function: The objective is to maximize vulnerability against hazardous events. In the ARS model, the attacker tries to maximize the vulnerability by determining the targets and attack powers. Again, it is noted that a node fails against events *d* only when all its chosen components fail at the same time upon the occurrence of events *d*.

- Constraints (IP 2.1) to (IP 2.6), which are the same as constraints (IP 1.1) to (IP 1.6) in RAPMA model, enforce the “continuity constraints” upon ARS model.
- Constraint (IP 2.7) and (IP 2.8), which are the same as constraints (IP 1.14) and (IP 1.15), both restrict the boundary of attack powers allocated to component *m* for node *i* and total available attack powers limitation.

Chapter 3

3.1. Solution Approach for the ARS Model

3.1.1 Lagrangean Relaxation Method

In the last decade, Lagrangean relaxation has grown from a successful but largely theoretical concept to a tool that is the backbone of a number of applications [18]. One of the core concepts of Lagrangean Relaxation method is decomposition, which slices up the complicated problem into several easily solvable and independent subproblems. Lagrangean Relaxation method is highly suitable to cope with large-scale problem in terms of scalability and efficiency.

One of the most computationally useful ideas of the 1970s is the observation that many hard problems can be views as easy problems complicated by a relatively small set of side constraints. Dualizing the side constraints produces a Lagrangean problem that is easy to solve and whose optimal value is a lower bound (for minimization problem) on the optimal value of the original problem. Due to a number of advantages over other programming methods, like linear programming, dynamic program, the Lagrangean Relaxation approach has provided the best existing algorithm for intractable combinatorial optimization problems [19].

Lagrangean Relaxation is principally on the basis of the observation that many difficult integer programming problems can be formulated as a relatively easy problem complicated by a set of side constraints. To employ this character, we create a Lagrangean problem, where the complicating constraints are relaxed and added to the objective function with associated Lagrangean multipliers (μ). After the transformation, LR problem (LR_{μ}) is decomposed into several independent subproblems which can be optimally solved by appropriate algorithm. According to the weaken duality theorem, for a minimization problem, the objective function value of LR problem always provides a lower bound to the primal problem. By this character, we attempt to obtain tightest lower bound by creating a Lagrangean dual problem, which tries to increase the lower bound via constantly adjusting the values of LR multiplier (μ). Generally, subgradient-based technique is frequently adopted.

After solving the LR problem, the feasibility of the result for primal problem (P) is checked. If it doe not violated the constraints in (P), a primal feasible solution is smoothly found; otherwise, additional efforts are needed to tune it to become a feasible one. Moreover, each feasible solution is naturally an upper bound for a minimization problem. Therefore, the optimal solution to (P) is guaranteed to locate between the LR lower bound and the primal feasible solution values. The core concepts and flow chart of Lagrangean Relaxation method are demonstrated in detail in Figure 3 and Figure 4, respectively.

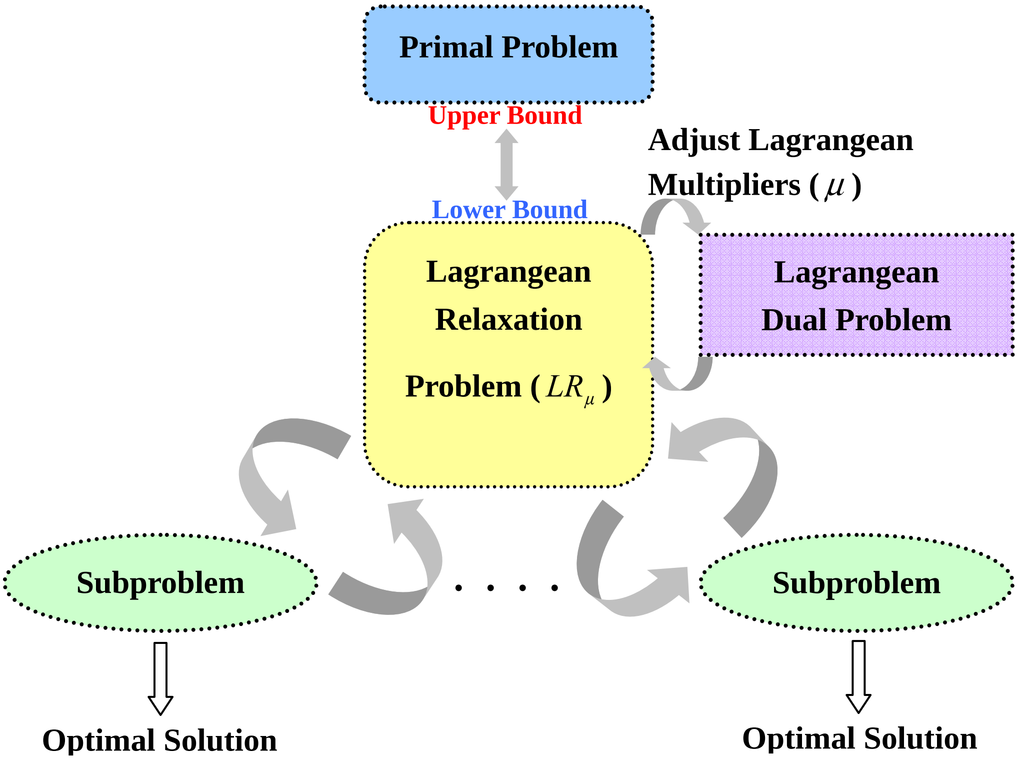


Figure 3. The Core Concepts of Lagrangean Relaxation Method

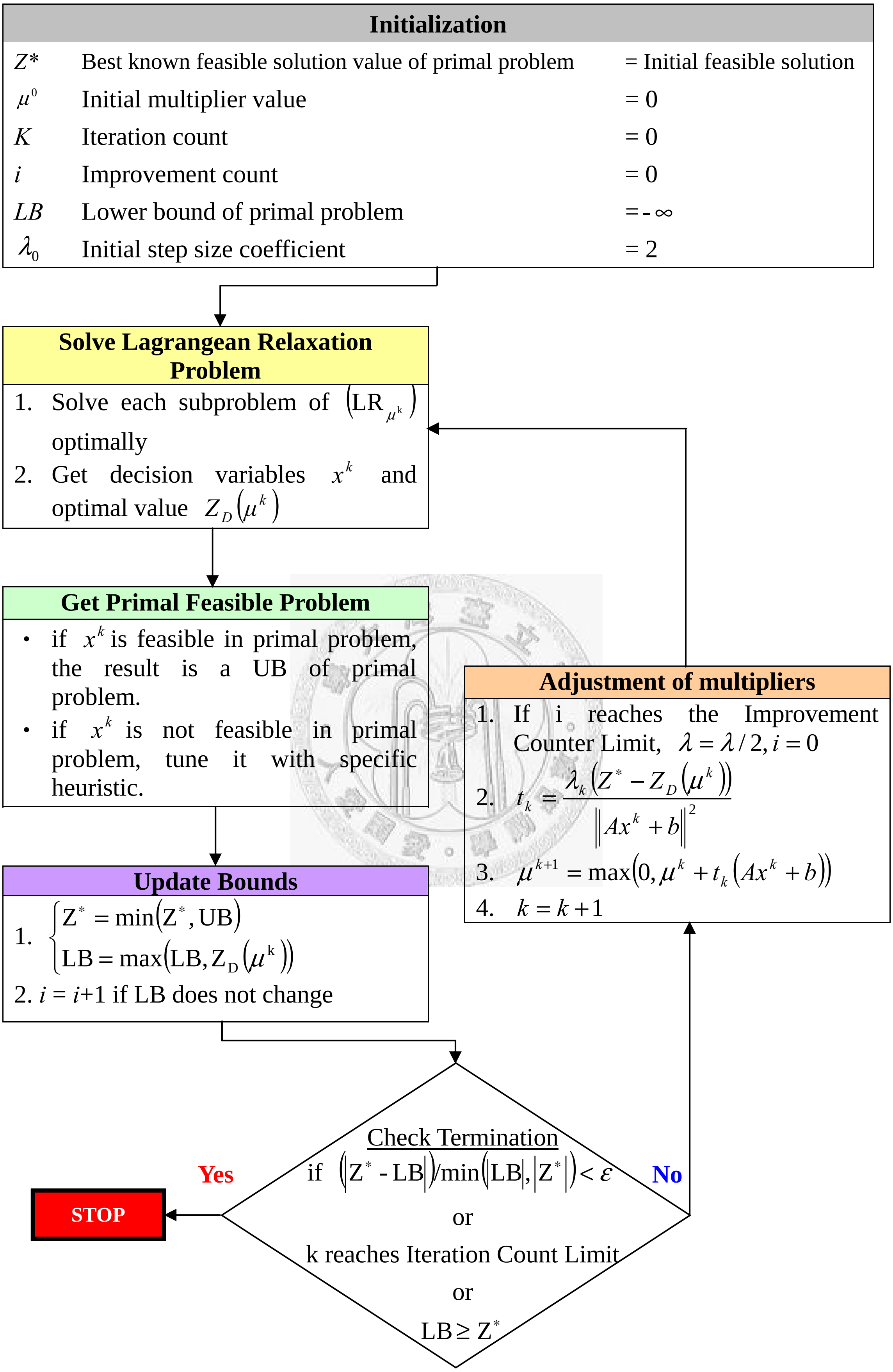

Initialization
Z^* Best known feasible solution value of primal problem = Initial feasible solution
μ^0 Initial multiplier value = 0
K Iteration count = 0
i Improvement count = 0
LB Lower bound of primal problem = - ∞
λ_0 Initial step size coefficient = 2
Solve Lagrangean Relaxation Problem
1. Solve each subproblem of (LR_{μ^k}) optimally
2. Get decision variables x^k and optimal value $Z_D(\mu^k)$
Get Primal Feasible Problem
• if x^k is feasible in primal problem, the result is a UB of primal problem.
• if x^k is not feasible in primal problem, tune it with specific heuristic.
Update Bounds
1. $\begin{cases} Z^* = \min(Z^*, UB) \\ LB = \max(LB, Z_D(\mu^k)) \end{cases}$
2. i = i+1 if LB does not change
Adjustment of multipliers
1. If i reaches the Improvement Counter Limit, $\lambda = \lambda / 2, i = 0$
2. $t_k = \dfrac{\lambda_k (Z^* - Z_D(\mu^k))}{\left\| Ax^k + b \right\|^2}$
3. $\mu^{k+1} = \max(0, \mu^k + t_k (Ax^k + b))$
4. $k = k + 1$
Check Termination
if $(|Z^* - LB|)/\min(|LB|, |Z^*|) < \varepsilon$
or
k reaches Iteration Count Limit
or
$LB \geq Z^*$
Yes
No
STOP

Figure 4. The Flow Chart of Lagrangean Relaxation Method

3.1.2 Lagrangean Problem of ARS Model

We apply Lagrangean Relaxation methodology to develop our solution approach. At the first beginning, we have to conduct adjustments with respect to objective function. The original objective function in (IP 2) is a value calculated by a series of product, which makes this problem intractable and complicated due to its non-linearity. Hence, we transform it, which is presented in product form, to logarithm form without changing its optimality. Beside, it is assumed that $f_{imd}(g_{im})$ follows an exponential distribution with λ, which indicated that the vulnerability will rapidly descend. The procedure and result after transformation are presented as follows.

$$Z_{IP2} = \max_{g_{im}} \sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\prod_{m\in r_i} f_{imd}\left(g_{im}\right)^{\alpha_{im}+\beta_{im}}\right)\right)$$

$$= \min_{g_{im}} -\left(\sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\prod_{m\in r_i} f_{imd}\left(g_{im}\right)^{\alpha_{im}+\beta_{im}}\right)\right)\right)$$

$$\Rightarrow \min_{g_{im}} -\left(\sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\sum_{m\in r_i}\left(\alpha_{im}+\beta_{im}\right)\ln\left(f_{imd}\left(g_{im}\right)\right)\right)\right)\right)$$

Assuming that $f_{imd}(g_{im}) \sim Exponential(\lambda)$

$$\Rightarrow \min_{g_{im}} -\left(\sum_{d\in D} p_d \left(1-\sum_{i\in N}\left(1-\sum_{m\in r_i}\left(\alpha_{im}+\beta_{im}\right)\ln\left(1-e^{-\lambda_{imd} g_{im}}\right)\right)\right)\right)$$

After the transformation, we successfully obtain a surrogate problem with the same optimal solution as primal one. Besides, to simplify the complexity of primal problem, some constraints with complicated mathematical structure are relaxed and decomposed into several independent subproblems. According to the past experience on Lagrangean Relaxation, those constraints are relaxed to acquire the Lagrangean Relaxation problem.

$$\sum_{w\in W}\sum_{p\in P_w} x_p \delta_{pi} \le \left(|N|-1\right) y_i \qquad \forall i\in N \qquad \text{(IP 2.1)}$$

$$\sum_{p\in P_w} x_p = y_i \qquad \forall i\in N, w=(s,i) \qquad \text{(IP 2.2)}$$

$$y_i \le \sum_{m\in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}(c_{im})} \qquad \forall i\in N \qquad \text{(IP 2.4)}$$

The corresponding Lagrangean Relaxation problem is shown as follows.

Objective function

$$\begin{aligned} & Z_D(\mu_1,\mu_2,\mu_3) \\ & = \min_{g_{im}} - \left(\sum_{d\in D} p_d \left(1 - \sum_{i\in N} \left(1 - \sum_{m\in r_i} (\alpha_{im} + \beta_{im}) \ln\left(1 - e^{-\lambda_{imd} g_{im}}\right) \right) \right) \right) \\ & + \sum_{i\in N} \mu_i^1 \left[\sum_{w\in W} \sum_{p\in P_w} x_p \delta_{pi} - \left(|N|-1\right) y_i \right] + \sum_{i\in N} \mu_i^2 \left[\sum_{p\in P_{(s,i)}} x_p - y_i \right] \\ & + \sum_{i\in N} \mu_i^3 \left[y_i - \sum_{m\in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}(c_{im})} \right] \end{aligned} \qquad \textbf{(LR 1)}$$

Subject to

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

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

$$y_i = 0 \ or \ 1 \qquad \forall i\in N \qquad \text{(LR 1.3)}$$

$$0 \le g_{im} \le A \qquad \forall i\in N,\ m\in r_i \qquad \text{(LR 1.4)}$$

$$\sum_{i\in N}\sum_{m\in r_i} g_{im} \le A \qquad \text{(LR 1.5)}$$

The Lagrangean Relaxation multipliers μ_1, μ_2, and μ_3 are the vectors of $\{\mu_i^1\}$, $\{\mu_i^2\}$, and $\{\mu_i^3\}$, respectively, where μ_1 is non-negative, μ_2 is unrestricted, and μ_3 is non-negative. In order to solve the (LR 1) optimally, we decompose it into three absolutely independent and easily solvable optimization subproblem as shown below.

Subproblem 1.1 (related to decision variables x_p)

Objective function

$$Z_D(\mu_1, \mu_2) = \min \sum_{i \in N} \sum_{w \in W} \sum_{p \in P_w} \mu_i^1 x_p \delta_{pi} + \sum_{i \in N} \sum_{p \in P_{(s,i)}} \mu_i^2 x_p \qquad \textbf{(Sub 1.1)}$$

Subject to

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(Sub 1.1.1)}$$

$$x_p = 0 \ or \ 1 \qquad \forall p \in P_w, w \in W \qquad \text{(Sub 1.1.2)}$$

In this problem, we want to determine the value of x_p individually for each O-D pair. Note that Constraint (Sub 1.1.1) allows only one path to be chosen for an O-D pair. As described in the notations, each O-D pair *w* originates from an attacker's position *s* and ends at one target node *i*, where $\forall i \in N$. Therefore, $\sum_{i \in N} \sum_{p \in P_{(s,i)}} \mu_i^2 x_p$ can be transformed into $\sum_{w \in W} \sum_{p \in P_w} \mu_i^2 x_p + \sum_{p \in P_{(s,s)}} \mu_s^2 x_p$, where $\sum_{p \in P_{(s,s)}} \mu_s^2 x_p$ can be ignored because no path starts and ends at the same node. After the transformation, we can further decompose (Sub 1.1) into |*W*| independent subproblems. For each O-D pair *w=(s, i)*, $\forall w \in W, i \in N$,

$$Z_{Sub1.1'} = \min \sum_{p \in P_w} \left(\sum_{j \in N} \mu_j^1 \delta_{Pj} + \mu_i^2 \right) x_p \qquad \textbf{(Sub 1.1')}$$

Subject to:

$$\sum_{p \in P_w} x_p \le 1 \qquad \forall w \in W \qquad \text{(Sub 1.1.1)}$$

$$x_p = 0 \; or \; 1 \qquad \forall p \in P_w, w \in W \qquad \text{(Sub 1.1.2)}$$

Accordingly, the algorithm with further decomposition for solving (Sub 1.1) is presented systematically in Table 6.

Table 6 Heuristic to Solve Subproblem 1.1’

Step1: For each O-D pair $w \in W$, we find the minimum cost shortest path using μ_j^1 as the node weight by Dijkstra’s minimum cost shortest path algorithm. The total cost of a path is the sum of the weights of the nodes on that path.

Step 2: For each O-D pair $w \in W$, we set the x_p value of each path *p* to zero except for the one already chosen to be the minimum cost shortest path for some O-D pair w, since no more than one path can exist between them.

Step3: For each O-D pair $w \in W$, we examine the sum of its minimum path cost and the μ_i^2 value of its destination node. If the value is non-positive, the x_p value of the minimum cost shortest path p between the O-D pair is set to one. The value of x_p is set to zero if its associated parameter is positive.

By applying the approach above, we are able to optimally solve this independent subproblem in a reasonable time. This heuristic is mainly on the basis of shortest path algorithm and all associated weights are non-negative. Consequently, Dijkstra's algorithm is chosen to develop this approach. The time complexity of Dijkstra's algorithm is $O\left(|N|^2\right)$. Since the source of each path is the same, Dijkstra's algorithm only needs to be implemented once since its outcome is the minimum cost shortest path tree; thus, the total time complexity of (Sub 1.1) is $O\left(|N|^2\right)$.

Subproblem 1.2 (related to decision variables y_i)

Objective function

$$Z_D\left(\mu_1,\mu_2,\mu_3\right)=\min\sum_{i\in N}-\left(\mu_i^2+\mu_i^3-\mu_i^1\left(|N|-1\right)\right)y_i \qquad \textbf{(Sub 1.2)}$$

Subject to

$$y_i=0 \; or \; 1 \qquad \forall i\in N \qquad \text{(Sub 1.2.1)}$$

(Sub 1.2) can be further decomposed into |*N*| independent problems. To solve this minimization subproblem is easy. Constraint (Sub 1.2.1) is an integer constraint, restricting the value of y_i to be either zero or one for each node *i*. Apparently, to obtain optimal solution to this subproblem, we only set the y_i with corresponding negative coefficient to one, where $i\in N$. In other words, as far as each node *i* is concerned, if the corresponding coefficient, $-\left(\mu_i^1+\mu_i^2+\mu_i^3\right)$, of y_i is negative, and then it is picked as one; contrarily, if the coefficient is positive, it is assigned to 0. The relation between y_i and its corresponding coefficient can be presented as shown below. The total time complexity of (Sub 1.2) is $O(|N|)$

$$y_i = \begin{cases} 1, \text{ if } -(\mu_i^2 + \mu_i^3 - \mu_i^1(|N|-1)) < 0 \\ 0, \text{ if } -(\mu_i^2 + \mu_i^3 - \mu_i^1(|N|-1)) \geq 0 \end{cases}$$

Subproblem 1.3 (related to decision variables g_{im})

Objective function

$$Z_D(\mu_3) = \min_{g_{im}} -\left(\sum_{d \in D} p_d \left(1 - \sum_{i \in N} \left(1 - \sum_{m \in r_i} (\alpha_{im} + \beta_{im}) \ln\left(1 - e^{-\lambda_{imd} g_{im}}\right) \right) \right) \right)$$
$$- \sum_{i \in N} \mu_i^3 \sum_{m \in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}(c_{im})} \qquad \textbf{(Sub 1.3)}$$

Subject to

$$0 \leq g_{im} \leq A \qquad \forall i \in N,\ m \in r_i \qquad \text{(Sub 1.3.1)}$$

$$\sum_{i \in N} \sum_{m \in r_i} g_{im} \leq A \qquad \text{(Sub 1.3.2)}$$

By the essence of (Sub 1.3), it is a typical fractional knapsack problem, which is also known as continuous knapsack problem. To optimally solve this subproblem, the technique of dynamic programming is adopted. At first, the problem is divided into *A* phases and exact one attacking resource is determined at each phase. Obviously, the “precious” resource will be allocated onto a component which can contribute the most value to the objective function at each phase. Namely, the optimal solution of each phase will be determined after the decision, *gim*, of each phase is made. The solution procedure, which is described in the form of pseudo code in Table 7, is repeated till all attacking resources are completely exhausted. Eventually, the optimal solution is obtained by applying the solution approach. The total time complexity of (Sub. 1.3) is $O(A|C|)$, where C is the number of components and A is total attacking resources.

Table 7. The Pseudo Code of Algorithm to Solve Subproblem 1.3

```
attackPower = A;
attackPolicy = new array(number of available components);

while attackPower is not exhausted {
    for all components to be determined {
        if the new value of this phase is greater than the old value
            attackPolicy is updated
}
```

3.1.3 The Dual Problem and Subgradient Method

By solving above subproblems optimally, the Lagrangean Relaxation problem (*LR1*) can also be solved optimally. According to the weak duality theorem, for any set of the multipliers (μ_1, μ_2, μ_3), $Z_{D1}(\mu_1, \mu_2, \mu_3)$ yields an LB on Z_{IP2}. In the following, we construct a dual problem (D 1) to calculate the tightest LB and solve it by the subgradient method [22] [23].

Dual Problem (D 1)

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

Subject to: $\mu_1 \geq 0, \mu_3 \geq 0$

Let a vector m be a subgradient of $Z_{D1}(\mu_1, \mu_2, \mu_3)$. Then, in iteration k of the subgradient procedure, the multiplier vector $\mu^k = \left(\mu_1^k, \mu_2^k, \mu_3^k\right)$ is updated by

$$\mu^{k+1} = \mu^k + t^k m^k,$$

where

$$m^k\left(\mu_1^k, \mu_2^k, \mu_3^k\right) = \left(\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pi} - \left(|N| - 1\right) y_i,\ \sum_{p \in P_{(s,i)}} x_p - y_i,\ \sum_{m \in r_i} \frac{\alpha_{im} g_{im}}{\hat{g}_{im}\left(c_{im}\right)} - y_i \right)$$

and the step size, t^k, is determined by

$$t^k = \lambda \frac{Z_{IP2}^* - Z_D\left(\mu^k\right)}{\left\|m^k\right\|^2}$$

Z_{IP2}^* represents the best UB on the primal objective function value obtained by iteration *K*. It is noteworthy that λ is a scalar between 0 and 2. Empirically speaking, it is usually initiated with the value of 2 and halved if the objective function value dose not improve within a given iteration count.

3.1.4 Getting Primal Feasible Solutions

According to the flaw chart of Lagrangean Relaxation in Figure 4, after the independent subproblems are optimally solved, we are able to derive a primal feasible solution from the hint of multipliers in Lagrangean Relaxation problem.

The algorithm used to get primal feasible solutions is described in detail below. Firstly, the solution of Subproblem 1.1 is considered as the initial attacking policy and inputted into the algorithm for sequential adjustment. If the attacking policy satisfies all constraints regarding to attacker's behavior, it will be the trunk of the ultimate attacking tree. On the contrary, if the attacking policy violates any constraints of the problem, the wasted attacking power, which is allocated to the leaf node, will be recycled and reallocated to the uncompromised nodes according to the associated weight, $\sum_{i\in N}\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pi}$. The procedure will not be terminated until the attacking policy is available. After the main attacking tree is constructed, the residual resources will be completely allocated to reachable components, which are associated with compromised nodes, according to its side effect on the objective function. Finally, a collection of primal feasible solution is found. The general steps and pseudo code of the algorithm are described in Table 8 and Table 9, respectively.

Table 8. Heuristic for Getting Primal Feasible Solution

Step 1: Utilize the attack policy derived from Subproblem 1.1 as the initial solution to the optimal problem.

Step 2: If the attack tree is available, go to Step 4, otherwise, go to Step 3.

Step 3: Recycle the wasted attack power, which is allocated to the leaf node, and re-allocate the recycled power to the uncompromised nodes according to the associated weight, $\sum_{i\in N}\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pi}$,. Go to Step 2.

Step 4: Allocate residual power to reachable components according to its side effect.

Table 9. Pseudo Code of Algorithm to Get Primal Feasible Solution

```
SortedSet candiateComponents; // sort all elements according to the value of
// $\sum_{d\in D} P_d \lambda_{imd} \quad i\in N, m\in r_i$
List attackTree;

// construct available attack tree
while(attackTree is not available) {
    recycledAttakPower←attackTree.removeLeave();
    tmpTree←Sort.sortByXpDeltaPi(attackTree);
    node←tmpTree.popFirst();
    if(node is not compromised) {
        cost←node.toBeCompromised();
        if(recycledAttackPower >= cost) {
            recycledAttackPower = recycledAttackPower – cost;
            node.makeCompromised();
        }
    }
}

// allocate recycledAttackPower to components in candidateComponents
if(recycledAttackPower > 0) {
  for all nodes node in the network {
      if(node.isCompromised()) {
        candidateComponents.add(node.getAllSecondaryComponents());
        For all adjacent nodes adjacentNode of node {
            candiateComponents.add(adjacentNode.getPrimaryComponents());
          }
      }
  }
  while(!candidateComponents.isEmpty() && recycledAttackPower > 0) {
        node←candidateComponents.removeFirst();
        node.addAttackPower(1);
        recycledAttackPower--;
  }
}
```

3.2. Solution Approach for the RAPMA Model

Since it is assumed that attacker and defender have complete and correct information about the “battle”, both of them are capable of maximizing their benefits according to opponent’s policy. In the ARS model, all decision variables about defense policy are assumed to be known in advance; therefore, the attacker is able to launch malicious attacks to paralyze the network system. After ARS model is solved by Lagrangean Relaxation, the solution of the ARS model, which can be regarded as attacker’s behavior, is inputted into the RAPMA model. In this phase, all decision variables about attacker’s behavior become known; as a result, network defender can dynamically deploy redundant components to strengthen the survivability of the whole network.

To solve the RAPMA model, a degree-based algorithm is proposed. In the beginning, sorting all nodes according to the associated weight, $\sum_{i\in N}\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pi}$, in descending order. The weight stands for the importance of the node. A node with higher weight represents that the node is relatively vital for attacker to successfully launch assault on the network. Consequently, the sorted nodes are checked one by one. If the node is successfully compromised by attacker, we upgrade its protection level, that is, more defense power will be allocated onto it; otherwise, degrade and recycle additional defense resources. After defense power allocated to primary components are determined, residual defense resources will be used to deploy secondary ones to maximize the survivability according to their side effect on protection ability. The detailed procedure is described in Table 10.

Table 10. Heuristic for Solving RAPMA model

Step 1: Sort the nodes according to the associated weight, $\sum_{i\in N}\sum_{w\in W}\sum_{p\in P_w} x_p\delta_{pi}$, in descending order.

Step 2: If the node is compromised, upgrade its protection level; otherwise, degrade and recycle allocated defense resources.

Step 3: Allocate residual resources to secondary components according to its side effect.

Step 4: A practical redundancy allocation policy is found.

Chapter 4 Computational Experiments

4.1. Experiment Environment

We will conduct experiments on the solution approach with respect to the scalability and the applicability. All proposed algorithms are coded in Java 1.6.0 with Eclipse 3.2 and executed on a computer with Intel(R) Pentium 4 CPU 3.00GHz, 512 MB memory.

The experiments are able to be divided into two parts. In the first part, we will run a series of experiments on the ARS model. To verify the scalability of our solution approach, nine scenarios different in topology structure and scale will be executed. Meanwhile, two simple algorithms, which are minimum cost spanning tree algorithm (SA1) and greedy-based algorithm (SA2), will be also conducted under the same conditions to demonstrate the efficiency of our heuristic. Furthermore, to verify the applicability, we will also perform the experiments in six types of topology structures at the same scale.

In the second part, a series of experiments on the RAPMA model will be also executed to show the scalability and the applicability of our solution approach. To demonstrate the efficiency, two different budget reallocation policies are designed to compare with our proposed heuristic. Except the comparative algorithms, the scenarios used in this part are the same as those in the ARS model.

4.2. Simple Algorithms

Two simple algorithms, which are minimum cost spanning tree algorithm and greedy-based algorithm, are designed to compare with the approach we proposed in the ARS model. They are also applied to Lagrangean Relaxation problem to obtain a primal feasible solution. The details of the two comparative methods are described as follows.

4.2.1 Minimum Cost Spanning Tree Algorithm

In the phase of getting primal feasible solution, prim's algorithm is applied to construct a minimum cost spanning tree as the attacking tree. To facilitate the algorithm, $\frac{1}{\text{min(number of hops from attacker)}}$ is used as the edge weight. Because the nature of edge weight and prim's algorithm, the process of paths selection is highly like depth first search algorithm. At first, the attacker will select a path from those adjacent nodes whose associated weights are 1. Next, the attacker will select a node from those nodes which are adjacent to selected nodes and whose associated weights are 1/2. The procedure will not be terminated until a spanning tree rooted at the node occupied by the attacker is constructed. The simple algorithm is described in the form of pseudo code in Table 11.

Table 11. Pseudo Code of Simple Algorithm 1

```
PriorityQueue fringe = {all nodes adjacent to the node occupied by the attacker};
// fringe is a priority queue which sorts all elements according to their associated
// weight, $\frac{1}{\text{min(number of hops from attacker)}}$.
List tree;
// tree is the container which stores all paths of the MST .
while(the spanning tree is not constructed yet) {
    selectedPath←fringe.getMinimalWeightNode();
    tree.add(selectedPath);
    fringe.add(selectedPath.getDestination().getAdjacentNodes());
}
```

4.2.2 Greedy-based Algorithm

The simple algorithm is based upon the concept of hill climbing. The attacker only takes advantage of local information to develop the attacking policy. Obviously, the solution is just a local optimal solution. The simple algorithm is described in the form of pseudo code in Table 12.

Table 12. Pseudo Code of Simple Algorithm 2

```
List attackPolicy;
// attackPolicy is the container which stores all paths.
Node current
// current is a node recording the current node attacker is occupying.
loop do {
    neighbor←current.getHighestValueSuccessor();
    if(value[neighbor] > value[current]) {
        attackPolicy.add(current);
    }
    current←neighbor;
}
```

4.3. Experiment Results

The experiment results can be divided into two parts. In the first part, the experiments for the ARS model will be conducted to verify the scalability and applicability of our proposed solution approach. In the second part, we focus on the RAPMA model to verify the scalability and applicability.

4.3.1. Experiments for ARS Model

To verify the scalability of our proposed solution approach, a series of experiments at three different scales are executed. Besides, they are conducted on two regular networks and one irregular network. The first regular network is grid network; the other one is cellular network. They are shown in Figure 5(a) and Figure 5(b), respectively. The third network is random network, which is shown in Figure 5(c).

All related parameters and scenarios used in the ARS model to verify the scalability are detailed in Table 13 and Table 14. In Table 13, all Lagrangean Relaxation related parameters are listed. In Table 14, all parameters of ARS model to verify scalability are listed.

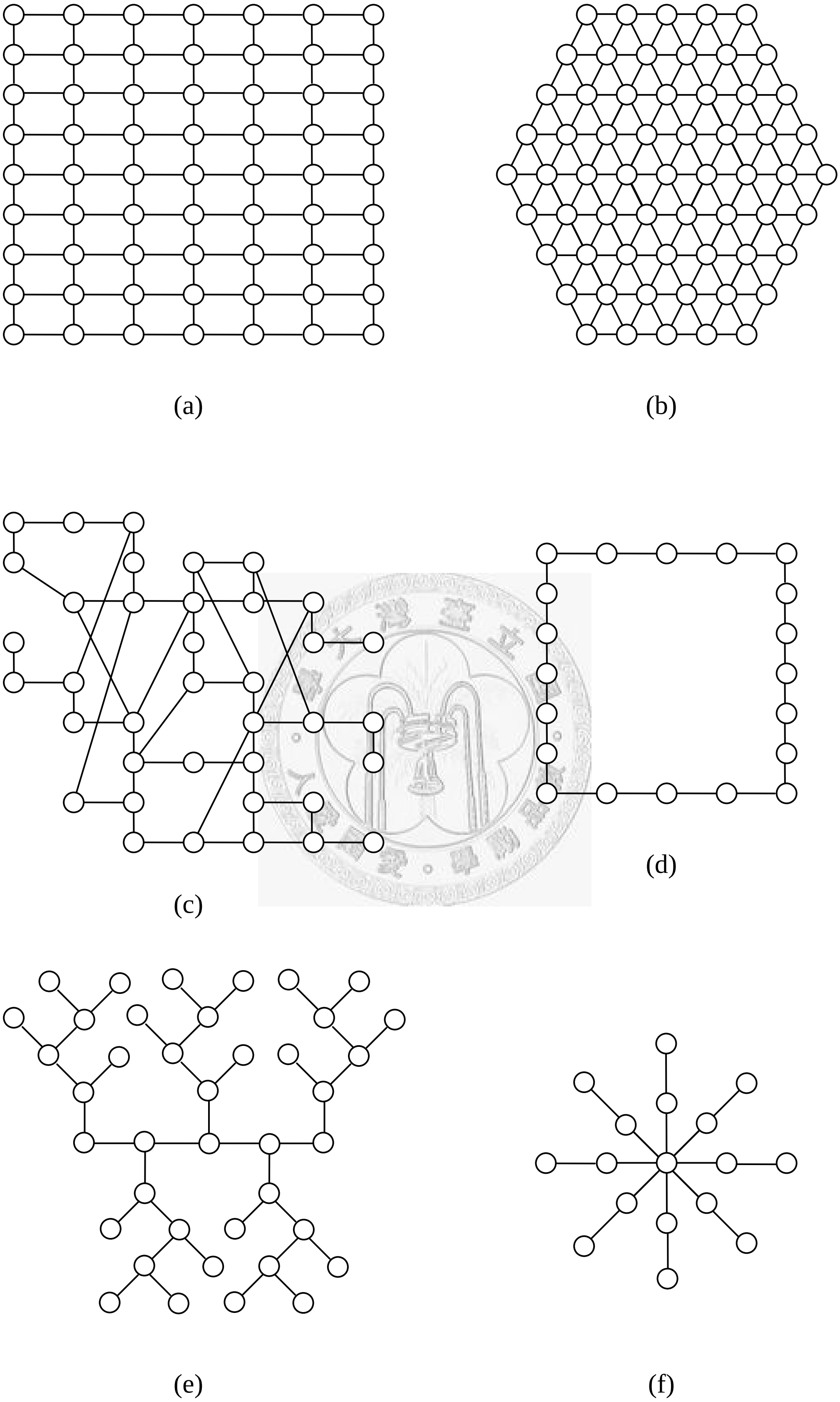


Figure 5. Network topologies: (a) Grid Network (b) Cellular Network (c) Random Network (d) Ring Network (e) Tree Network (f) Star Network.

Table 13. Parameters of LR for ARS Model

Parameters of Lagrangean Relaxation in ARS model	
Parameters	**Value**
Iteration Counter Limit	2000
Improve Counter Limit	60
Initial UB	Positive Limit
Initial Multiplier Value	All multipliers are initiated to be 0
Initial Scalar of Step Size	2
Test Platform	CPU: Intel(R) Pentium(R) 4 3.00GHz RAM: 512MB OS: Windows XP with SP2

Table 14. Parameters of ARS Model to Verify Scalability

Parameters of ARS model to verify scalability		
Parameters	**Value**	
Test Topology	• Grid network • Cellular network • Random network	
Scalability	Number of nodes	Number of components
	16 (Small)	16*5=80
	64 (Medium)	64*5=320
	196 (Large)	196*5=980
Simple Algorithms	• Minimum cost spanning tree (SA1) • Greedy-based algorithm (SA2)	

The experiment results for ARS model to verify scalability are listed in Table 15, Table 16, and Table 17. For readability, the results are also diagramed in Figure 6. In each table, four values, which are vulnerability, GAP, MPI for SA1 and MPI for SA2, are recorded. “Vulnerability” represents the possibility that the sequential hazardous events might cripple the whole network. “GAP” is an indicator used to measure the quality of primal feasible solution and calculated by the formulation, $\frac{\left|\text{UB-LB}\right|}{\min\left(\text{UB,LB}\right)}\times 100\%$. “MPI” is also an indicator used to compare the proposed heuristic with the two simple algorithms and calculated by the formulation, $\frac{V_{ARS}\text{-}V_{SA}}{1\text{-}V_{SA}}\times 100\%$.

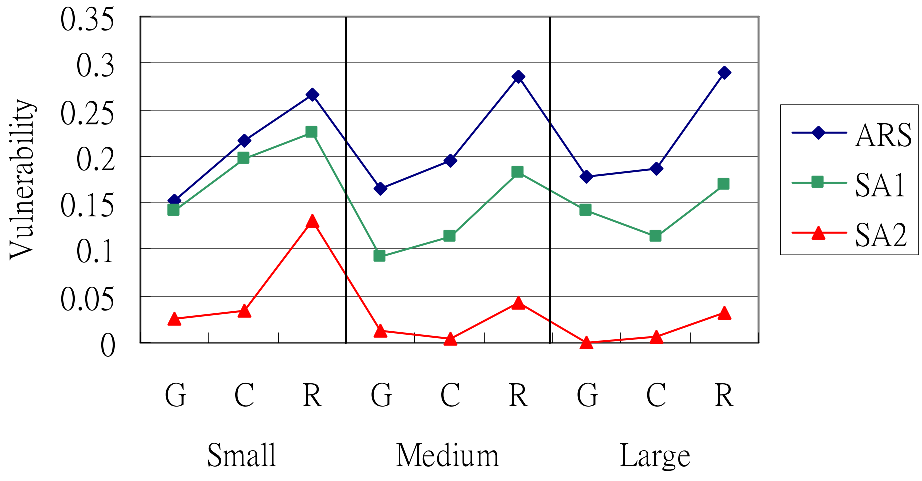


Figure 6. Vulnerability in Different Scenarios to Verify the Scalability

Table 15. Experiment Results of Grid Network for ARS Model at Different Scales

Test Topology: Grid Network				
Scale	ARS		SA1	SA2
	Vulnerability	GAP	MPI	MPI
Small	0.15157984	0.82%	1.26%	12.86%
Medium	0.16621758	1.71%	8.17%	15.45%
Large	0.17754317	4.58%	4.12%	17.67%

Table 16. Experiment Results of Cellular Network for ARS Model at Different Scales

Test Topology: Cellular Network				
Scale	ARS		SA1	SA2
	Vulnerability	GAP	MPI	MPI
Small	0.21771277	0.74%	2.56%	18.94%
Medium	0.19572636	1.97%	9.34%	19.25%
Large	0.18656719	5.27%	8.1%	18.11%

Table 17. Experiment Results of Random Network for ARS Model at Different Scales

Test Topology: Random Network				
Scale	ARS		SA1	SA2
	Vulnerability	GAP	MPI	MPI
Small	0.26585439	1.12%	5.24%	15.62%
Medium	0.28546145	2.36%	12.63%	25.29%
Large	0.28886455	9.62%	14.28%	26.45%

All related parameters and scenarios used in the ARS model to verify the applicability are listed in Table 18. There are total six different topology structures at the same scale used to execute a series of experiments on applicability. Besides the three network topologies used in testifying the scalability, the other three topologies, which are ring, tree and star network, are also used in the experiments. They are illustrated in Figure 5(d), Figure 5(e), and Figure 5(f), respectively.

Table 18. Parameters of ARS Model to Verify Applicability

Parameters of ARS model to verify applicability		
Parameters	**Value**	
Test Topology	• Grid network • Cellular network • Tree Network • Ring Network • Mesh Network • Random network	
Scalability	Number of nodes	Number of components
	49	49*5 = 245
Simple Algorithm	• Minimum cost spanning tree (SA1) • Greedy-based algorithm (SA2)	

The experiment results for ARS model to verify applicability are listed from Table 19 to Table 24. For readability, the results also diagramed in Figure 7.

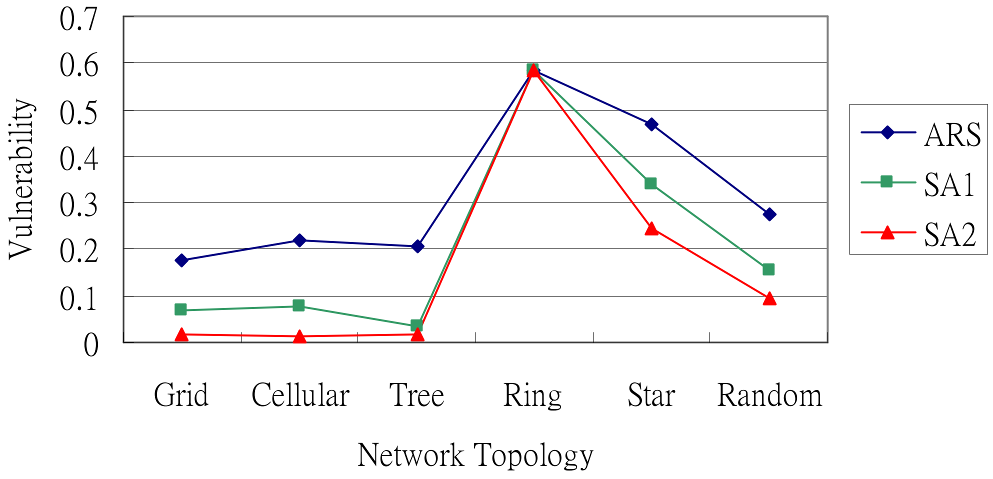


Figure 7. Vulnerability in Different Scenarios to Verify the Applicability

Table 19. Experiment Results of Grid Network for ARS Model

Test Topology: Grid Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.17601521	2.45%	11.32%	16.34%

Table 20. Experiment Results of Cellular Network for ARS Model

Test Topology: Cellular Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.21768319	3.34%	15.36%	20.63%

Table 21. Experiment Results of Tree Network for ARS Model

Test Topology: Tree Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.20758016	1.49%	17.99%	19.38%

Table 22. Experiment Results of Ring Network for ARS Model

Test Topology: Ring Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.58512787	2.34%	0%	0%

Table 23. Experiment Results of Star Network for ARS Model

Test Topology: Star Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.46600688	5.31%	19.34%	29.34%

Table 24. Experiment Results of Random Network for ARS Model

Test Topology: Random Network			
ARS		SA1	SA2
Vulnerability	GAP	MPI	MPI
0.27603519	2.03%	14.34%	20.02%

4.3.2. Experiments for RAPMA Model

In the section, a series of experiments concerning scalability and applicability will be also performed on RAPMA model. In the part of scalability, we will conduct the experiments in three different topology structures at different scales. All the related parameters are detailed in Table 25.

Furthermore, to demonstrate the efficiency of proposed solution approach, two different budget reallocation policies are introduced. The first one is uniform budget allocation policy (B1), where each node is allocated exactly the same resources without considering other factors. The other one is damage-based budget allocation policy (B2), in which each node's resources are determine by the attack power the malicious attacker allocates. In other words, the more damage the node suffers, the more important the node is. Therefore, from the perspective of network operator, more defense resources should be allocated onto a node suffering more damage.

As for the applicability, the experiments will be executed in a variety of topologies, including grid network, cellular network, ring network, tree network, start network, and random network. Those network topologies used in the experiments are illustrated in Figure 5 in section 4.3.1. Similarly, two comparative budge allocation policies will also be performed. All the related parameters are listed in Table 29.

Table 25. Parameters of RAPMA Model to Verify Scalability

Parameters of RAPMA model to verify scalability		
Parameters	**Value**	
Test Topology	• Grid network • Cellular network • Random network	
Scalability	Number of nodes	Number of components
	16 (Small)	16*5=80
	64 (Medium)	64*5=320
	196 (Large)	196*5=980
Budge Reallocation	• Uniform Budget Allocation (B1) • Damage-based Budget Allocation (B2)	

The experiment results for RAPMA model to verify the scalability are listed in Table 26,

Table 27, and Table 28. For readability, the results are also diagramed in Figure 8. In each table, there are three values recorded. The first one is “Survivability”, which is antithetic to the concept of vulnerability and calculated by (1-vulnerability). The second one and third one are MPI, which has been described in section 4.3.1.

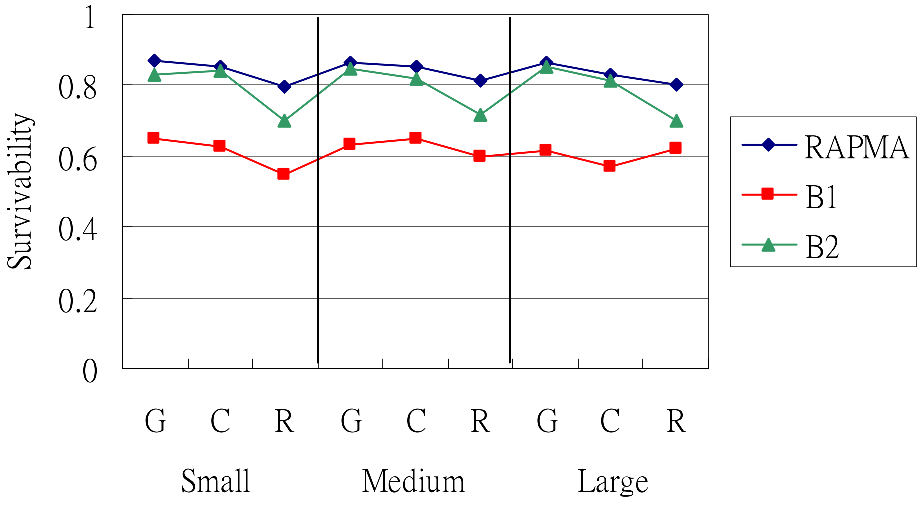


Figure 8. Survivability in Different Scenarios to Verify Scalability

Table 26. Experiment Results of Grid Network for RAPMA Model at Different Scales

Test Topology: Grid Network			
Scale	RAPMA	B1	B2
	Survivability	MPI	MPI
Small	0.87213465	63.22%	35.18%
Medium	0.86542113	63.34%	23.11%
Large	0.86352289	64.56%	6.02%

Table 27. Experiment Results of Cellular Network for RAPMA Model at Different Scales

Test Topology: Cellular Network			
Scale	RAPMA	B1	B2
	Survivability	MPI	MPI
Small	0.85228767	60.62%	28.69%
Medium	0.85344421	58.15%	26.60%
Large	0.83328114	61.38%	11.63%

Table 28. Experiment Results of Random Network for RAPMA Model at Different Scales

Test Topology: Random Network			
Scale	RAPMA	B1	B2
	Survivability	MPI	MPI
Small	0.79862511	55.26%	-1.85%
Medium	0.81238667	53.12%	3.79%
Large	0.80024281	47.60%	-0.23%

All related parameters and scenarios used in the RAPMA model to verify the applicability are listed in Table 29. Similarly, there are total six different topology structures at the same scale used to execute a series of experiments on applicability.

Table 29. Parameters of RAPMA Model to Verify Applicability

Parameters of RAPMA model to verify applicability		
Parameters	**Value**	
Test Topology	• Grid network • Cellular network • Tree Network • Ring Network • Mesh Network • Random network	
Scalability	Number of nodes	Number of components
	49	49*5 = 245
Budge Reallocation	• Uniform Budget Allocation (B1) • Damage-based Budget Allocation (B2)	

The experiment results for RAPMA model to verify applicability are listed from Table 30 to Table 35. For readability, the results are diagramed in Figure 9.

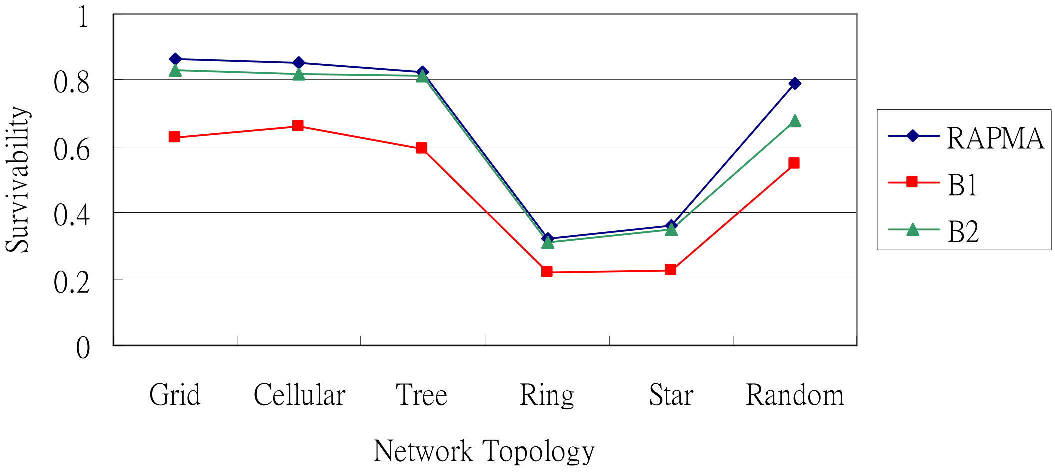


Figure 9. Survivability in Different Scenarios to Verify Applicability

Table 30. Experiment Results of Grid Network for RAPMA Model

Test Topology: Grid Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.86284214	63.43%	18.43%

Table 31. Experiment Results of Cellular Network for RAPMA Model

Test Topology: Cellular Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.85348768	57.06%	19.29%

Table 32. Experiment Results of Tree Network for RAPMA Model

Test Topology: Tree Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.82487913	56.97%	5.93%

Table 33. Experiment Results of Ring Network for RAPMA Model

Test Topology: Ring Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.32452156	13.42%	-44.44%

Table 34. Experiment Results of Star Network for RAPMA Model

Test Topology: Star Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.35983741	17.40%	1.70%

Table 35. Experiment Results of Random Network for RAPMA Model

Test Topology: Random Network		
RAPMA	B1	B2
Survivability	MPI	MPI
0.78994813	53.46%	0.03%

4.4. Discussion of Experiment Results

The discussion will be decomposed into four parts, which are scalability of heuristic for ARS model, applicability of heuristic for ARS model, scalability of heuristic for RAPMA model, and applicability of heuristic for RAPMA model.

- Scalability of Heuristic for ARS Model

  According to the experiment results in Figure 6, no matter in what network topologies or at what scales, our proposed heuristic prominently outperforms another two simple algorithms in terms of vulnerability. SA1 only considers the local information; apparently, the final results must highly underestimate the value. As to SA2, which is similar to DFS algorithm due to the design of edge weight, it is easily affected by the structure of topology. If the topology slopes toward one side, SA2 might explore it along the inclined side and terminate till the attack power is exhausted. As a result, the final result solved by SA2 will be a path, which also highly underestimates the real value. In other words, SA1 is similar to SA2 in some cases. However, our proposed heuristic makes use of the hints provide by LR; it will constantly adjust its direction in a global view. Hence, the solution quality is definitely better than the two simple algorithms.

- Applicability of Heuristic for ARS Model

According to the experiment results in Figure 7, it has been proven that our proposed heuristic for ARS model is applicable in a variety of topologies. It is noteworthy that the three different heuristics will come up with the same attacking policy in ring network. Because each node in ring network has only one adjacent neighbor, no matter what heuristics we adopt, only one solution will be obtained in the condition where attack power is equal.

- Scalability of Heuristic for RAPMA Model

According to the experiment results in Figure 8, it has been proven that our proposed heuristic is capable of coping with a large-scale problem and surpasses another two algorithms in survivability. B1 allocates the same budgets onto each node in the network. Thus, no dynamic adjustments will be performed to response the change of attacking policy. Obviously, B1 will easily lead to weaknesses in the aspect of defense engineering. As for B2, it can also provide a tighter bound in grid network and cellular network; however, the solution quality drops in random network. The reason why existing the difference might lie in the structure of topology. Regular networks are relevantly robust in nature when suffering malicious attacks; contrarily, random network is vulnerable to attacks.

Therefore, it can explain why B2 performs well in regular network but fails in random network.

- Applicability of Heuristic for RAPMA Model

  According to the experiment results in Figure 9, it has been proven that our proposed heuristic for RAPMA model can be applied to a variety of networks. It is noteworthy that B2 is also able to come up with a tight bound to the optimal solution. Moreover, B2 is relevantly easier to be implemented in terms of complexity. Consequently, if time is the most important issue in developing the solution approach, B2 is suggested to be the appropriate policy.

In general, our proposed heuristic is scalable and applicable. By the experiment results, the structure of network plays a decisive role in developing defense policy. A network with higher average degrees will be more robust. Moreover, an interesting phenomenon is found. Those nodes which are relevantly near to the attacker will be allocated relevantly more defense resources and decrease hop by hop. The outer nodes will form a “fosse” to protect the inner nodes. Maybe, the defense engineering techniques used to protect the castles in the past can be transformed to develop defense policy in modern information security world.

Chapter 5 Conclusion and Future Work

5.1. Conclusion

Internet facilitates the flourishing developments on completely new economic activities and provides a worldwide platform to exchange data rapidly. Most business organizations either require Internet to assist in daily operations or directly build their services upon it. Any failure in data communication, no matter incurred by malicious attacks or hazardous events, will result in inestimable damage and economic loss. As a consequence, constructing a network configuration or proposing a recovery plan which supports continuity of service is the most urgent mission for any service providers. To meet the requirement, redundancy allocation planning is one of the key solutions.

In the thesis, we propose a brand-new solution approach based upon redundancy allocation to protect the network against man-made and natural threats. Observing the previous researches on redundancy allocation problem, they mostly focus on the risks incurred by natural disasters or the attacks governed by a random probability. Besides, only one perspective, either attacker or defender, is considered in the previous researches. The insufficiency easily leads to ignore some important facts in the battle between the intelligent attacker and defender. To supplement the existing insufficiency, we formulate the battle into a two-level mathematical problem and take the emerging target attacks into account.

The main contributions of our work consist in proposing a mathematical model which formulates the interaction between the attacker and defender. In the ARS model, we replace the random access attacks governed by a probability with the malicious target attacks conducted by continuity constraints to reflect the popular trend in the information security world. According to our survey, scant works transform the attackers' real behavior into a well-formulated model. Moreover, in the realm of redundancy allocation problem, few works consider the impacts of target attacks and hazardous events at the same time; however, those potential risks indeed bring severe threats. In other words, our model is more generic in handling a variety of scenarios in the real-world.

From the results of computational experiments, our proposed solution approach apparently surpasses other algorithms in terms of survivability. Besides, the designed experiments on scalability and applicability have proven that our heuristic is capable of dealing with a large-scale problem and applicable in all kinds of network structures and environment. Furthermore, the results of computational experiments also reflect a defense guideline. A node with higher degree requires more budgets and a network with higher average degree is more robust.

5.2. Future Works

Our proposed model at least has two interesting directions, which are listed as follows, to be extended in the future.

- Hazardous events occur round by round.

  In our thesis, we only address the scenario where hazardous events, no matter natural disasters or man-made attacks, occur exactly one time after the malicious target attacks. Nevertheless, in some cases, the hazardous events, especially the man-made attacks, will be launched round by round. The objective of first round might be to detect the existing vulnerabilities. After some weaknesses have been found, the attacker might launch several rounds of malicious attacks to conquer the network. Hence, extending the second round of attack to third round, fourth round and so on is worthy paying more attention to discuss.

- Hazardous events occur prior to malicious attacks.

  In our thesis, it is assumed that the malicious attacks are always launched prior to hazardous event occurring. However, in the real-world, it might exist a contrary scenario where natural disasters occur before sequential malicious attacks. When a destructive disaster occurring, it will do a large volume of damage to the network. That is, the network becomes more vulnerable to malicious attacks. The attack will make use of the opportunity to launch attacks to attain his/her goal.

Consequently, the extension to reverse situation is noteworthy and highly interesting.

References

[1] Wojciech Molisz, “Survivability Function — A Measure of Disaster-Based Routing Performance,” *IEEE Journal on Selected Areas in Communication, vol. 22, no. 9, November 2004*

[2] Howard F. Lipson, David A. Fish, “Survivability — A New Technical and Business Perspective on Security,” *CERT* ® *Coordination Center Software Engineering Institute*

[3] Benjamin B.M. Shao, “Optimal Redundancy Allocation for Information Technology Disaster Recovery in the Network Economy,” *IEEE Transactions on Dependable and Secure Computing, Vol. 2, No.3, July-September 2005*

[4] David W. Coit, Abdullah Konak, “Multiple Weighted Objectives Heuristic for the Redundancy Allocation Problem,” *IEEE Transactions on Reliability, Vol. 55, No. 3 September 2006*

[5] Jose E. Ramirez-Marquez, David W. Coit, Abdullah Konak, “Redundancy Allocation for series-parallel systems using a max-min approach,” *IIE Transactions (2004) 36, 891-898*

[6] Yong Jiang, James D. McCalley, Tim Van Voorhis, “Risk-Based Resource Optimization for Transmission System Maintenace,” *IEEE Transactions on Power Systems, Vol. 21, No. 3, August 2006*

[7] Chunghun Ha, Way Kuo, "Multi-Path Heiristic for Redundancy Allocation: The Tree Heuristic," *IEEE Transactions on Reliability, Vol. 55, No. 1, March 2006*

[8] Herve Kerivin, Dritan Nace, Thi-Tuyet-Loan Pham, "Design of Capacitated Survivable Network with a Single Facility," *IEEE/ACM Transactions on Networking, Vol. 13, No. 2, April 2005*

[9] Young-Soo Myung, Hyun-joon Kim, Dong-wan Tcha, "Design of Communication Networks with Survivability Constraints," *Management Science, Vol. 45, No. 2 (Feb., 1999), 238-252*

[10] Soumyo D. Moitra, Suresh L. Konda, "A Simulation Model for Managing Survivability of Networked Information Systems," *Technical Report CMU/SEI-2000-Tr-020 ESC-TR-2000-020*

[11] Chenxi Wang, "A Security Architecture for Survivability Mechanisms," *a dissertation presented to the School of Engineering and Applied Science at the University of Virginia*

[12] John Koroma, Wei Li, Demetrios Kazakos, "A Generalized Model for Network Survivability," *TAPIA'03 October 15-18, 2003, Atlanta, Georgia., USA*

[13] David M. Nicol, William H. Sanders, Kishor S. Trivedi, "Model-Based Evaluation: From Dependability to Security," *IEEE Transactions on Dependable and Secure Computing, Vol.1, No. 1, January-March 2004*

[14] Zeev Zeitlin, “Integer Allocation Problems of Min-Max Type with Quasiconvex Separable Functions,” *Operation Research, Vol. 29, No. 1, January-February 1981*

[15] John C. Knight, Elisabeth A. Strunk, Kevin J. Sullivan, “Towards a Rigorous Definition of Information System Survivability,” *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX’03)*

[16] J. McDermott,” Attack-Potential-Based Survivability Modeling for High-Consequence System,” *Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA’05)*

[17] M. S. Chern, “On The Computational Complexity of Reliability Redundancy Allocation in A Series System,” *IEEE Transactions on Reliability, Vol. R-36, No. 5 pp.621-623, 1987*

[18] Marshall L. Fisher, “An Applications Oriented Guide to Lagrangian Relaxation,” *Interfaces, Vol. 15, No. 2, pp. 10-21, April 1985*

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

[20] L.A. Gordon, M.P. Loeb, W. Lucyshyn, R. Richardson, “2006 CSI/FBI Computer Crime and Security Survey,” *Computer Security Institute, 2006,* http://GoCSI.com

[21] Fotios Harmantzis, Manu Malek, “Security Risk Analysis and Evalution,” *IEEE Communication Society*

[22] Ram Dantu, Kall Loper, Prakash Kolan, "Risk Management using Behavior based Attack Graphs," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*

[23] Stefano Bistarelli, Fabio Fioravanti, Pamela Peretti, "Defense Tree for Economic Evaluation of Security Investments," *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*

簡歷

姓名：江坤道

出生地：台灣省彰化縣

生日：中華民國七十二年元月二十號

學歷：九十年九月至九十四年六月

國立中山大學 資訊管理學系學士

九十四年九月至九十六年六月

國立台灣大學 資訊管理研究所碩士