Redundancy and Defense Resource Allocation Algorithms to Assure Service Continuity against Natural Disasters and Intelligent Attacks

Frank Yeong-Sung Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
yslin@im.ntu.edu.tw

Yu-Shun Wang
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
d98002@im.ntu.edu.tw

Po-Hao Tsang
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
d91002@im.ntu.edu.tw

Jui-Pin Lo
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
r96009@im.ntu.edu.tw

***Abstract*—In this paper, we discuss Redundancy Allocation Problem (RAP) in network environments. By efficiently combining redundancy with extra defense mechanisms, we attempt to ensure the continuity of a network service, and enhance its survivability against malicious attackers that utilize accumulated experience. We construct an attack/defense scenario, in which an attacker and a defender compete against each other, and formulate it as a two-phase nonlinear integer programming problem. We adopt a Lagrangean Relaxation-based solution approach to resolve the above problem, and further prove the efficacy of our approach by computer experiments. The result shows LR-Based attack algorithm is better than other strategies we compared. Further, no matter what kind of attack/defense cost function is adopted, the LR-Based allocation algorithm can always provide a much better defense capability than others.**

Keywords: Service Continuity, Redundancy Allocation Problem (RAP), Survivability, Attack/Defense Scenario, Lagrangean Relaxation

I. INTRODUCTION

Due to the convenience and efficiency of information technology (IT), more and more businesses have been running their routine operations and providing services for their customers with the help of IT in recent years. Although IT undoubtedly brings several advantages, there are still many potential threats against these important IT elements, including earthquakes, tsunamis, hurricanes, floods, abuse of employees, power outages, terrorisms, and hacker attacks. All of above can be mainly divided into four categories: natural disasters, human errors, utility disruptions, and man-made malicious attacks from outsiders. These potential threats are also the main causes of business interruption [1], because a business may suffer from the interruption of IT-supported operations or business processes [2].

The potential threats to the IT operations of a company are not only difficult to predict but also to prevent; however, from the management aspect, businesses still have to make satisfactory preparations for worst case scenarios. Any business interruption can really damage the running of a business, in terms of loss of profits from potential transaction, loss of customers, and serious damage to reputation. According to an electronic survey conducted by Frost & Sullivan and $(ISC)^2$ in 2008 [3], 73% of the respondents view the impact of service downtime as the top priority in risk management. Therefore, most businesses must strive to prevent important business processes from interruption so as to ensure their service continuity.

An important issue based on the concept of service continuity is Business Continuity Planning (BCP), or more generally known as Business Continuity Management (BCM). In the previous literature, Lam (2002) proposed a BCP cycle consisting of eight core steps [4] to provide a stepwise method for IT-related organizations. Since more and more businesses started to pay a lot attention to BCM, the British Standards Institution (BSI) responded by establishing a standard named BS25999 [5] in 2006.

Within the scope of BCM, Disaster Recovery Planning (DRP) is the most IT-related portion, and it is also a critical issue to organizations. In survey [3], the respondents from both America and Asia-Pacific all rank in third place security technology deployed to ensure business continuity and disaster recovery solutions. There are many components that can be taken into consideration for DRP, including backup methods, alternate sites, support teams, equipment replacement [6], and the use of existing compatible on-site equipments to replace the failed ones. This replacement is treated as the concept of redundancy as well.

Redundancy is one of the security approaches commonly used to cope with IT disaster recovery [2]. For a system, allocating redundancy is an absolutely effective solution to mitigate the potential risks of operational interruption. This is because the identically functioning redundant components in a hot-standby state can immediately take over the failed components from possible disasters. This feature of redundancy sufficiently fulfills the requirements of continuous service, and in practice, there are actually many system designs using functionally similar but not exactly the same components in parallel [7]. All of these problems concerned with redundancy allocation are generally defined as Redundancy Allocation Problem (RAP), and is applied to several research fields, such as parallel systems [2] [8 and series-parallel systems [7] [9] [10]. However, RAP is seldom discussed in the network environments even though networks are vital to businesses nowadays. Therefore, we decide to study RAP in the network environments in this work. This is thus the motivation of our present research.

There is no perfectly safe system or network in reality. In the 2008 CSI survey, just over half (51%) of the

978-0-7695-4236-2/10 $26.00 © 2010 IEEE
DOI 10.1109/BWCCA.2010.72

respondents attribute their firms' losses to non-insiders [11], i.e., the attacks from outside of the organization. Thus, we consider malicious attacks when applying RAP to network systems.

We also adopt another extensively studied concept, survivability, as the measurement of resisting malicious attacks. Many researchers and businesses have taken survivability seriously since about 1990s; yet there is still no consistent definition of survivability according to [12], a study containing a through account of survivability. Among all the related research, the most frequently cited is Ellison et al. in 1997 [13], which provides the most accepted definition of survivability.

In summary, this study attempts to compensate the preexisting yet insufficient RAP research about networks. More specifically, we will discuss how to make efficient use of redundancy together with extra defense to ensure the service continuity of a network and simultaneously enhance its survivability when facing intelligent malicious attackers. Therefore, we construct an attack/defense scenario in which an attacker and a defender compete against each other in a given network environment. We then accordingly formulate it as a max-min mathematical programming problem, which consists of the Redundancy Allocation Problem with Extra Defense Mechanisms (RAP-EDM) model and the Attack with Experience Accumulation (AEA) model, and handle it by two phases to get a satisfying solution.

The rest of this paper is organized as follows: the RAP-EDM model and the AEA model are introduced in Section 2; solution approaches based on the Lagrangean Relaxation methods are presented in Section 3; the computer experimental result is illustrated in Section 4; finally, conclusions and possible research directions are provided in Section 5.

II. PROBLEM FORMULATION

The problem we address here is how to make good use of redundancy to assure service continuity and maximize the survivability of the whole network against intelligent malicious attacks at the same time. There are thus two main measurements we must evaluate appropriately: the service availability of each node and the survivability of whole network.

In this work, the service continuity in terms of every node's service availability is assured by the contribution of redundancy; more precisely, if we ensure that the expected number of redundant components in every node is always satisfied with a predefined operating minimum requirement, each node can provide the required service without interruption, even when some of redundant components in it fail due to random errors, natural disasters, or malicious attacks.

From the perspective of deterring attackers, defense resources are used to increase the attack cost that an attacker required for compromising one node successfully [14]. Considering the worst case in which the attacker can achieve his ultimate goal, we treat the total attack cost required for disrupting all mission-critical service in the target network as the measurement of network survivability.

Considering a network consisting in Autonomous System (AS) level nodes with different service functions, such as web server, file transfer protocol (FTP) site, mail server, the plan about which node should provide what kind of service function is predefined and consistent. Furthermore, there are multiple core nodes providing mission-critical service or storing important information, but a non-core node may just provide transmission, rather than a specified service function.

With finite budget, the defender has to enhance the survivability of the whole network and the assuring service availability of each node by exploiting unified purchase to implement redundancy allocation. First of all, there is a catalog of products that lists all available kinds of redundant components providing each specified function with different brands or types, i.e., the redundant component choice set of different specified functions. Besides this, for each type of redundant component, there are several extra defense mechanisms, including firewall, anti-virus, anti-spam, application level firewall, which are especially appropriate for being chosen to provide further protection, and these can be defined as the defense mechanism choice set of different redundant components. Different kinds of redundant components or defense mechanisms have distinct defense costs and attack thresholds, and various kinds of redundant components also have dissimilar reliabilities. In other words, the probability that a redundant component operates properly is not the same.

Because natural disasters and random errors may happen during the operation of redundant components, the defender has to confirm the service availability of each node. Thus, when allocating redundant components, the expected number of redundant components must satisfy the requirement of service availability assurance. On the other hand, the defender also needs to take into consideration the capacity limitation of all nodes. Also, the defender as an operator of this network has to choose the appropriate redundant components together with defense mechanisms to be allocated to each node for maximizing the total attack cost of compromising all core nodes regarding the service availability of every node.

In the most severe situation, the intelligent attacker has the perfect knowledge about the target network, including the topology of the network, the allocation of redundant components and extra defense mechanisms of each node. Furthermore, the attacker also knows the threshold of attacking each kind of redundant component or defense mechanism, and the attacker also compromises the initial node first; then attacks one node at a time until compromising all core nodes. The attacker's ultimate goal is to cause failure in the critical services provided by all core nodes in the target network with minimal attack cost. Moreover, the attacker prefers to penetrate surreptitiously instead of causing destruction before actually reaching the core nodes. Thus, the attacker will compromise the primary redundant component in non-core nodes, and then use such nodes as a hop site to reach further nodes. When reaching the core nodes, the attacker will compromise all redundant

components in the core nodes, thus leading to total dysfunction without doubt.

Accordingly, we consider this worst case for the defender. While penetrating a non-core node, the attacker can always choose to compromise the redundant component to advantageously minimize total attack cost. However, before actually compromising a redundant component, the attacker must compromise all of the extra defense mechanisms that have been deployed to protect it.

Because it is possible that there are some redundant components or defense mechanisms that remain the same in nodes with identical function, the attacker can compromise nodes based on previous experiences. In other words, the attacker has experience accumulation. If the attacker has compromised certain kinds of redundant components or defense mechanisms once, he/she can find some useful methods or develop some efficient hacker tools to cope with them. Afterward the attacker can compromise the same kind of redundant component or defense mechanism with a much lower fixed attack cost.

Therefore, the attacker can decide which redundant components in which nodes to attack for achieving all core nodes and actually disrupt them with the minimum attack cost according to prior accumulation of attack experience.

A. Problem Formulation of RAP-EDM Model

We model the above problem as a max min integer programming problem. The given parameters and decision variables are defined in Table 1 and Table 2 respectively.

TABLE 1. GIVEN PARAMETERS

Notation	Description
B	The total defense budget limitation
N	The index set of all nodes in the network
T	The index set of all core nodes in the network
U	The index set of all non-core nodes in the network
F	The index set of all functions provided by nodes in the network
M_f	The index set of all redundant components which can be selected to provide the same main function f, where $f \in F$
W	The index set of all Origin-Destination (O-D) pairs, the origin is node s and the destination is the other node i, where s, $i \in N$
P_w	The index set of all candidate paths regarding O-D pair w, where $w \in W$
D_m	The index set of all extra defense mechanisms available for the kind of redundant component m, where $m \in M_f$, $f \in F$
α	The threshold of service continuity assurance that defines the minimum expected number of redundant components for each node
β	The capacity limitation of redundant components for each node
σ_{if}	The indicator function, 1 if node i provides function f, otherwise 0 (where $i \in N$, $f \in F$)
δ_{pi}	The indicator function, 1 if node i is on the path p, otherwise 0 (where $i \in N$, $p \in P_w$, $w \in W$)
c_m	The cost of redundant component m, where $m \in M_f$, $f \in F$
$\hat{a}_m(c_m)$	The threshold of the attack cost required to compromise redundant component m, where $m \in M_f$, $f \in F$
λ_m	The consistent ratio that defines the fixed part of the attack cost when compromising redundant component m, where $m \in M_f$, $f \in F$
Q_m	The probability of redundant component m that operates properly, where $m \in M_f$, $f \in F$
c_{md}	The cost of deploying defense mechanism d on redundant component m, where $d \in D_m$, $m \in M_f$, $f \in F$
$\hat{a}_{md}(c_{md})$	The threshold of the attack cost required to compromise defense mechanism d deployed on redundant component m, where $d \in D_m$, $m \in M_f$, $f \in F$
λ_{md}	The consistent ratio that defines the fixed part of the attack cost when compromising defense mechanism d deployed on redundant component m, where $d \in D_m$, $m \in M_f$, $f \in F$

TABLE 2. DECISION VARIABLES

Notation	Description
R_{im}	1 if redundant component m is allocated to node i, otherwise 0 (where $m \in M_f$, $f \in F$, $i \in N$)
R_{imd}	1 if defense mechanism d of redundant component m is allocated to node i, otherwise 0 (where $d \in D_m$, $m \in M_f$, $f \in F$, $i \in N$)
y_i	1 if node i is compromised, otherwise 0 (where $i \in N$)
y_{im}	1 if redundant component m in node i is compromised, otherwise 0 (where $m \in M_f$, $f \in F$, $i \in N$)
y_{imd}	1 if defense mechanism d deployed on redundant component m in node i is compromised, otherwise 0 (where $d \in D_m$, $m \in M_f$, $f \in F$, $i \in N$)
z_m	Times of redundant component m being compromised (where $m \in M_f$, $f \in F$)
z_{md}	Times of defensive mechanism d deployed on redundant component m being compromised (where $d \in D_m$, $m \in M_f$, $f \in F$)
x_p	1 if path p is selected as the attack path, otherwise 0 (where $p \in P_w$, $w \in W$)

The objective of the defender is to maximize the minimized total attack cost for compromising all core nodes in the network. In the inner problem, the AEA model, the attacker attempts to minimize the total attack cost by deciding which redundant components in which nodes to compromise. It is worth mentioning that $\left\lceil \frac{z_m}{|N|} \right\rceil \left[1+(z_m-1)\lambda_m\right]$ and $\left\lceil \frac{z_{md}}{|N|} \right\rceil \left[1+(z_{md}-1)\lambda_{md}\right]$ stand for the impact of the attacker's experience accumulation on redundant components and extra defense mechanisms, respectively. The attacker requires the full attack cost at the first time but only a fixed small portion of original cost is required to spend afterwards. In the outer problem, the defender tries to make best use of limited defense resources to allocate suitable redundant components with extra defense mechanisms to maximize the minimized total attack cost, while also regarding the service availability assurance and capacity limit of every node at the same time.

Objective function:

$$\max_{z_m, z_{md}} \min \sum_{f \in F} \sum_{m \in M_f} \left[\begin{array}{l} \left\lceil \frac{z_m}{|N|} \right\rceil \hat{a}_m(c_m)\left[1+(z_m-1)\lambda_m\right] \\ +\sum_{d \in D_m} \left\lceil \frac{z_{md}}{|N|} \right\rceil \hat{a}_{md}(c_{md})\left[1+(z_{md}-1)\lambda_{md}\right] \end{array} \right] \qquad \text{(IP 1)}$$

subject to:

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

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

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

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad \text{(IP 1.4)}$$

$$y_i = 0 \text{ or } 1 \quad \forall i \in U \quad \text{(IP 1.5)}$$

$$y_i = 1 \quad \forall i \in T \quad \text{(IP 1.6)}$$

$$R_{imd} \le R_{im} \quad \forall i \in N, m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.7)}$$

$$\sum_{m\in M_f} R_{im} Q_m \ge \alpha \quad \forall i \in N, f \in F \quad \text{(IP 1.8)}$$

$$\sum_{m\in M_f} R_{im} \le \beta \quad \forall i \in N, f \in F \quad \text{(IP 1.9)}$$

$$R_{im} = 0 \text{ or } 1 \quad \forall i \in N, m \in M_f, f \in F \quad \text{(IP 1.10)}$$

$$R_{imd} = 0 \text{ or } 1 \quad \forall i \in N, m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.11)}$$

$$y_{im} \le R_{im} \quad \forall i \in N, m \in M_f, f \in F \quad \text{(IP 1.12)}$$

$$y_{imd} \le R_{imd} \quad \forall i \in N, m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.13)}$$

$$y_i = \sum_{m\in M_f} y_{im} \quad \forall i \in U, f \in F \quad \text{(IP 1.14)}$$

$$y_{im} \sum_{d\in D_m} R_{imd} \le \sum_{d\in D_m} y_{imd} \quad \forall i \in U, m \in M_f, f \in F \quad \text{(IP 1.15)}$$

$$y_{im} = R_{im} \quad \forall i \in T, m \in M_f, f \in F \quad \text{(IP 1.16)}$$

$$y_{imd} = R_{imd} \quad \forall i \in T, m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.17)}$$

$$y_{im} = 0 \text{ or } 1 \quad \forall i \in N, m \in M_f, f \in F \quad \text{(IP 1.18)}$$

$$y_{imd} = 0 \text{ or } 1 \quad \forall i \in N, m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.19)}$$

$$z_m = \sum_{i\in N} y_{im} \quad \forall m \in M_f, f \in F \quad \text{(IP 1.20)}$$

$$z_{md} = \sum_{i\in N} y_{imd} \quad \forall m \in M_f, f \in F, d \in D_m \quad \text{(IP 1.21)}$$

$$\sum_{i\in N}\sum_{f\in F} \sigma_{if} \sum_{m\in M_f} \left[R_{im} c_m + \sum_{d\in D_m} R_{imd} c_{md} \right] \le B \quad \text{(IP 1.22)}$$

Constraint (IP 1.1) prevents the attack paths forming loops. Constraint (IP 1.2) and Constraint (IP 1.3) enforces the condition that if the attacker tries to compromise a node, there must be one attack path to that node. Constraints (IP 1.1) to (IP 1.5) jointly compose the "continuity constraints". Constraint (IP 1.8) restricts the expected number of redundant components in each node to being no less than the threshold of service availability assurance. Constraint (IP 1.14) restricts a non-core node being penetrated if one of the redundant components allocated to it has been compromised. Constraint (IP 1.15) restricts the attacker attempting to compromise a redundant component if and only if all extra defense mechanisms deployed for protecting it have been compromised.

III. SOLUTION APPROACH

Since the max min problem we face in the RAP-EDM model changes dynamically, it is difficult to solve immediately. Therefore, we adopt an alternative two-phase approach to cope with it. First, we abstract the inner problem of the RAP-EDM model, the AEA model, as a maximization integer programming problem and handle it by optimization techniques to get the best attack strategy. After this, we treat the solution of the AEA model as the input of the RAP-EDM model, and then solve it to develop the defense plan about how to allocate redundant components and defense mechanisms to each node.

A. Solution Approach for AEA Model

We adopt the Lagrangean relaxation (LR) method to cope with the complicated problems addressed in the AEA model. One of the key concepts of LR is relaxation, and that allows us to remove the limitations caused by the set of relatively troublesome constraints. Instead of considering them directly, we take them into the objective function of the primal problem with corresponding Lagrangean multipliers. Thereafter, a LR problem is then constructed, and we can further decompose the LR problem into several easily solvable subproblems, which are independent of each other.

When dealing with a minimization problem, the objective value of the LR problem is always a lower bound of the primal problem [15], although this solution of the LR problem may be infeasible for the primal one. Based on this, we attempt to acquire the lower bound as tightly as possible by continuously tuning the Lagrangean multiplier. The process of unceasingly tuning the Lagrangean multiplier is known as the Lagrangean dual problem. Accordingly, we adopt one of the most popular approaches for tightening the lower bound, the subgradient method.

When the LR problem is solved, we turn to consider whether the solution is feasible for the primal problem. If the solution meets all the requirements of the primal constraints, a satisfying optimal solution is acquired; otherwise, we have to further develop some appropriate heuristics to make the infeasible solution become a feasible one. Furthermore, each feasible solution of the primal problem also provides an upper bound of the optimal value. Therefore, the actual optimal solution of the primal problem can be guaranteed within the range between the obtained upper bound and lower bound.

In the AEA model, we relax four constraints, i.e., (IP 2.1), (IP 2.2), (IP 2.9), and (IP 2.10), to construct the LR problem and further separate it into four independent subproblems. The first subproblem is related to decision variable x_p. We resolved it mainly by the Dijkstra shortest path algorithm. The second subproblem is about decision variable y_i. Since there are just two possibilities of decision variable y_i for every non-core node, we can use exhausted search to find out if y_i should be 0 or 1 to contribute the

minimum value, for each non-core node i. Handling decision variable y_{im} and z_m is the responsibility of the third subproplem. We solve it by fixing the value of some variables first and then determining the value of the rest of the variables. The last subproblem deals with decision variable y_{imd}, z_{md}. In a similar fashion to the solving of the third subproblem, we adopt an assembling method to resolve it. After solving these subproblems, we resort to some LR multipliers and decision variables to develop a LR-Based heuristic attack algorithm to solve the problem in the AEA model.

In the beginning, the attacker must decide which paths to take as attack paths. We take advantage of the result of the first subproblem, which is related to attack path choice, to create the attack paths. Then, we redefine the path cost of each node and run the Dijkstra shortest path algorithm to construct the attack paths toward all core nodes. These attack paths form an attack tree. We also calculate the lowest attack cost for compromising each non-core node. This result is added to the LR multiplier which is related to the path choices, thus deriving the path cost of each node. Moreover, we believe that the attacker would tend to take those compromised nodes as hop sites when choosing paths to achieve the remaining core nodes. The way of implementing above idea is to let the path cost of each chosen node be zero to replace original one after deciding any attack path, and then rerun Dijkstra's shortest path algorithm for achieving the next core node. This procedure will be executed until reaching all core nodes from the starting node.

According to the description of our attack scenario, the attacker must compromise all redundant components of all core nodes with corresponding defense mechanisms deployed for them. In other words, we have to calculate the corresponding Z_ms and Z_{md}s.

The last step is to determine which redundant component and corresponding defense mechanisms to compromise in those non-core nodes which are on the attack tree. Therefore, we need to figure out the corresponding attack cost. When considering the cost of compromising these non-core nodes, the impact of attack experience accumulation is also in our concern. If there is a kind of redundant component which has been compromised before, its attack cost will only be the fixed part. Of course, this rule is also applied to defense mechanisms. Finally, we choose the redundant component with defense mechanisms that totally contribute the least costs within a non-core node to penetrate on the attack tree. During the attack processes in these non-core nodes, the corresponding Z_ms and Z_{md}s also need to be updated continuously.

B. Solution Approach for RAP-EDM Model

The LR-Based heuristic attack algorithm proposed in section 3.1 makes the attacker achieve the ultimate goal in an efficient way, and it also provides the defender some information about the intelligent attacker's behavior in the meantime. Therefore, we design a heuristic allocation algorithm which is highly related to the LR-Based attack algorithm. We call it the LR-Based allocation algorithm and introduce it below.

The LR-Based allocation algorithm consists of two main parts, initial allocation and allocation adjustment. In the process of initial allocation, the first thing we need to satisfy is the service continuity requirement, so we first allocate redundant components to all nodes following a predefined order. The core nodes have the first priority to get the types of redundant components with higher costs, so we allocate as many different kinds of more expensive redundant components as possible to core nodes under capacity and budget constraints. We then allocate unused types of redundant components to those non-core nodes which are One-Hop away from the starting node or the core nodes, and then allocate redundant components to the remaining non-core nodes lasting the end. Unlike allocating redundant components to core nodes, we allocate the required number of the same kind of redundant components to the non-core nodes. The reason is that we give the attacker no chance to choose a more vulnerable redundant component within a non-core node to penetrate. Moreover, the distribution of each kind of redundant component must be even.

After satisfying the requirement of service continuity, we allocate as many different kinds of defense mechanisms to protect the redundant components in core nodes if the remaining budget is abundant. Then, we execute the LR-Based attack algorithm and record how many times each non-core node has really been compromised within 2,000 iterations. Those non-core nodes with higher records will be allocated with defense mechanisms with the first priority, so they will have a stronger probability to get more expensive defense mechanisms. Because the attacker can arbitrarily choose the most vulnerable redundant component with defense mechanisms to attack, we must allocate the same kind of defense mechanism to every redundant component within a non-core node. For protecting each non-core node as fairly as possible, the allocation of defense mechanisms is in rotation following the priority made by the above record, i.e., one non-core node get allocated a kind of defense mechanism each time. The allocation of defense mechanisms will be executed until the budget is run out, and the initial allocation is finished.

After finishing the initial allocation of this network, we rerun the LR-Based attack algorithm and still record how many times each non-core node has been compromised in 2,000 iterations. If the total allocation cost of redundant components and defense mechanisms in a non-core node with a lower record is higher than the total cost of another same functioning non-core node with a higher record, we completely exchange their allocations. This kind of adjustment will be continuously executed until those non-core nodes that are compromised frequently are guaranteed to get more expensive redundant components and defense mechanisms, thus finishing a round of adjustment.

The process of allocation adjustment is composed of numerous rounds of adjusting. We repeat the procedure described above until the total number of execution reaches

the limit. We then take the allocation that results in the highest total attack cost as the final decision.

IV. COMPUTATION EXPERIMENTS

A. Compared Algorithms

To evaluate the quality of the proposed LR-Based attack algorithm, we further design two simple attack algorithms, SA1 and SA2, for comparison. We adopt the Dijkstra shortest path algorithm as the method to choose attack paths to reach the core nodes in both simple attack algorithms. However, there is one main difference between the two algorithms: the determination of path cost of each node. We take the physical distance, i.e., the hop count to other nodes, as the metric to decide the path cost of each node in SA1. In SA2, for each non-core node, we first calculate the attack cost of compromising each redundant component with its defense mechanisms and set the minimum value to be the path cost. Otherwise, we let the path costs of all core nodes be 0. After deciding the attack paths, the rest steps in the two simple algorithms are the same as the LR-Based attack algorithm.

Because RAP is seldom discussed under a network attack/defense scenario, there is no suitable well-known question for measuring the effectiveness of our LR-Based allocation algorithm. Thus, we provide a randomized allocation algorithm (RA) and a Core-Focused allocation algorithm (CF) to compare with our proposed allocation algorithm. Although RA is a random process, it still fulfills the service continuity requirement, budget constraint, and capacity limitation. CF, on the other hand, prefers to allocate redundant components and defense mechanisms to the core nodes. Afterward, it will randomly allocate redundant components and defense mechanisms to each non-core node. Of course, we must make the allocation comply with all the related constraints.

B. Experiment Environment

We adopt the above attack algorithms and allocation algorithms in a grid network topology since the concept of defense-in-depth is embedded in this topology. The function provided by each node is randomly decided, and each non-core node has a 50% probability of only providing a transmission function. There are six predefined core nodes in the target network, and their positions are consistent. The service continuity requirement forces the number of redundant components in each node not to be smaller than 2, and the total number is restricted by the capacity limit which equals 5.

The prices of different kinds of redundant components are between 50 and 100, and their reliabilities are between 85% and 99%. The prices of different kinds of defense mechanisms are between 1 and 20. We further define the fixed part of attack cost for compromising different redundant components and defense mechanisms as the attack threshold multiplied by a random ratio which is between 1% and 30%.

To prove our LR-Based attack algorithm and LR-Based allocation algorithm are generally suitable for practice, the experiments are implemented under fifteen different parameter settings, and these are composed of various values provided by five main parameters: Number of Nodes, Defense Budget, Number of Functions, Size of Redundant Component and Defense Mechanism Choice Sets, and the relationship between the attack costs and the prices of all redundant components and defense mechanisms.

It should be noted that we set three different kinds of relationships between the attack costs and the prices of all redundant components and defense mechanisms. These are linear, convex, and concave. In order to bring some variations into the attack cost, we randomly adjust the price by adding or removing a portion of it.

C. Experiment Results

The metric we apply for evaluating the efficiency of an attack algorithm is the total attack cost for compromising all core nodes in the target network. For an attack algorithm, the lower total attack cost means it is more effective to achieve the goal. On the other hand, we also make the total attack cost as the metric to evaluate the robustness of an allocation. Regarding an allocation, the more total cost the attacker required to achieve his goal indicates the network is more robust against malicious attacks. The experiment results are demonstrated in the following figures.

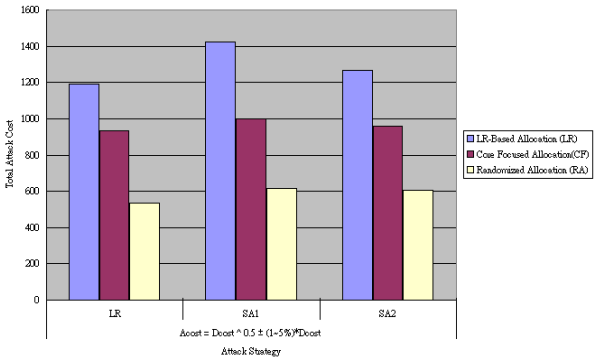

Figure 1. Performance Comparison on Concave Attack/Defense Cost Function

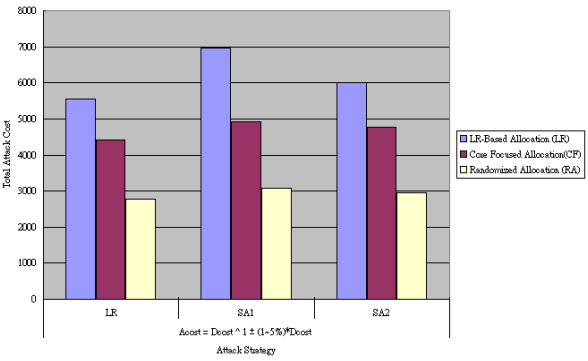

Figure 2. Performance Comparison on Linear Attack/Defense Cost Function

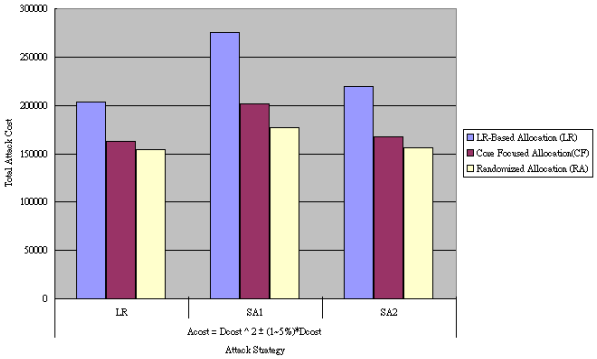

Figure 3. Performance Comparison on Convex Attack/Defense Cost Function

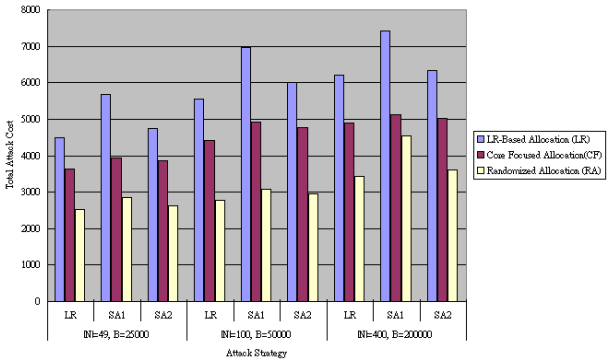

Figure 4. Performance Comparison on Different Network Sizes

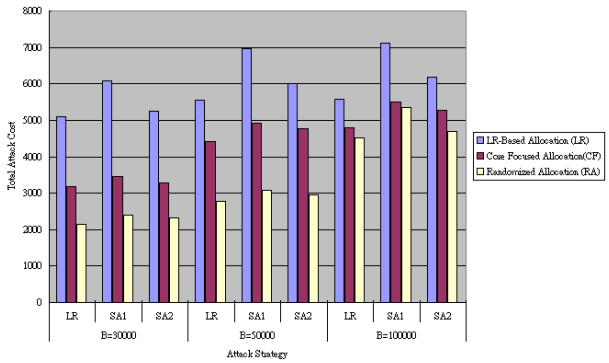

Figure 5. Performance Comparison on Different Defense Budget Settings

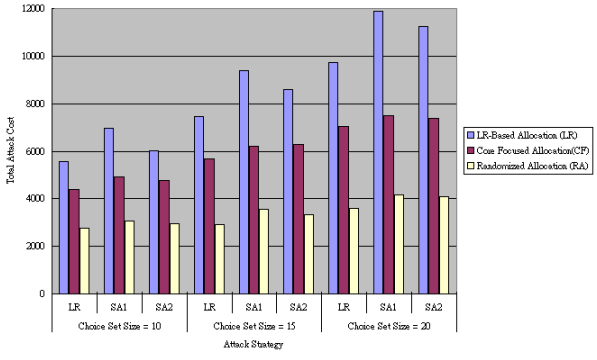

Figure 6. Performance Comparison on Different Choice Set Sizes

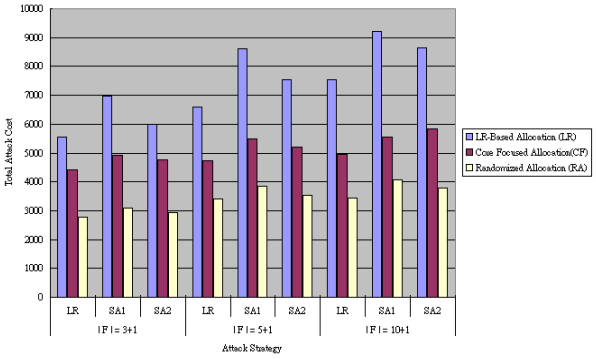

Figure 7. Performance Comparison on Different Numbers of Functions

D. Discussion of Results

In all figures, we find the LR-Based attack algorithm always helps the attacker compromise all core nodes with a lower total cost compared with SA1 and SA2. The LR-Based attack algorithm is about 10% to 35% better than SA1 and about 10% better than SA2 averagely. No matter what kind of attack/defense cost function is adopted, the LR-Based allocation algorithm can always provide a much better defense capability compared with RA and CF.

In Fig. 4, the larger network consumes more of the attacker's budget for compromising the core nodes. However, the enhancement of total attack cost tends to be smaller while the choice set sizes remain the same rather than expanding with the network size. This is due to the impact of the attack experience accumulation. Moreover, the difference between LR-Based allocation, RA, and CF slightly drops down when the number of nodes increases from 100 to 400. This can be explained as that larger network provides the defender more space to randomly allocate different kinds of redundant components and defense mechanisms during randomized allocation.

The impact on total attack cost invoked by defense budget can be observed in Fig. 5. The LR-Based allocation algorithm can produce a greater improvement than RA and CF when the defense budget is not very abundant, for example, 30,000; however, the difference between different allocation algorithms becomes smaller when the defense budget is relatively ample, i.e., 100,000. The reason is that a great amount of defense budget allows the defender to randomly allocate different kinds of redundant components and defense mechanisms into nodes when adopting RA or executing the random allocation procedure of CF. In addition, while doubling the defense budget from 50,000 to 100,000, the marginal effect on total attack cost is comparatively slight, and we attribute this to the capacity limit and the attack experience accumulation.

In Fig. 6, the expansion of choice set sizes brings more kinds of products, i.e., redundant components and defense mechanisms, into this problem. When the defender adopts the LR-Based allocation algorithm which prefers to make the times of allocating each kind of product as uniform as possible, the total attack cost will be enhanced due to the greater diversity of products. On the other hand, RA cannot make good use of the diversity of products to enhance the

total attack cost. Therefore, the difference between the LR-Based allocation and RA becomes more obvious when the diversity of products gets larger. This phenomenon can also be observed from Fig. 7 because the more functions a network needs to provide, the more kinds of different products the defender can choose for allocation.

Furthermore, we have acquired some guidelines for redundancy and defense allocation from the experiment results. The defender can achieve much better defense performance by adopting sophisticated allocation methods, for example, the LR-Based allocation algorithm, in the following three situations: first, when the defense budget is not that abundant; second, when the choices of redundant components and defense mechanisms are rich; and third, when the target network provides many kinds of functions. Moreover, concentrating on strengthening the core nodes significantly enhances the total attack cost at the beginning. However, the best way to enhance the total attack cost is to allocate more non-existing types of redundant components and defense mechanisms to each node. The purpose is to reduce the effect caused by attack experience accumulation which lets the marginal effect of defense investment on total attack cost become slight.

V. CONCLUSIONS

In this research, we discuss RAP in network environments as a defense/attack scenario both considering non man-made failures and malicious attacks. In order to fully describe this attack/defense scenario, we formulate it as two mathematical models, RAP-EDM model and AEA model, to represent the behavior of the defender and the attacker, respectively. We successfully model the competition between both sides into a mathematical problem and further solve it by proposing LR-based heuristics. This can be regarded as the key contribution of this work. We also clearly describe the real attackers' capability of experience accumulation in a mathematical way within our model. This is another contribution of this paper. Moreover, we implement computational experiments to evaluate the effectiveness of our algorithms and further provide guidelines for redundancy and defense allocation according to the experiment results.

Nonetheless, this study has pointed out directions for future research, such as the tradeoff between diversity of products and defender's bargaining power and the concern of maintenance cost, which have the potential to make this research further conform to reality.

REFERENCES

[1] M.J. Cerullo and V. Cerullo (2004). Business Continuity Planning: A Comprehensive Approach. Information Systems Management, 21(3), 70-78.

[2] B. B.M. Shao (2005). Optimal Redundancy Allocation for Information Technology Disaster Recovery in the Network Economy. IEEE Transactions on Dependable and Secure Computing, 2(3), 262-267.

[3] Frost & Sullivan and $(ISC)^2$ (2008). The 2008 $(ISC)^2$ Global Information Security Workforce Study.

[4] W. Lam (2002). Ensuring Business Continuity. IT Professional, 4(3), 19-25.

[5] British Standard Institution, BSI (2006)."BS25999.

[6] P. Fallara (2004). Disaster Recovery Planning. IEEE Potentials, 22(5), 42-44

[7] A.E. Smith and D.W. Coit (1996). Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm. IEEE Transactions on Reliability, 45(2), 254-260.

[8] G. Levitina and K. Hausken (2007). Protection vs. Redundancy in Homogeneous Parallel Systems. Reliability Engineering and System Safety, 93(10), 1444-1451.

[9] Y.-C. Hsieh (2003). A Linear Approximation for Redundant Reliability Problems with Multiple Component Choices. Computers and Industrial Engineering, 44(1), 91-103.

[10] A. Konak, D.W. Coit, and J.E. Ramirez-Marquez (2004). Redundancy Allocation for Series-Parallel Systems Using a Max-min Approach. IIE Transactions, 36(9), 891-898.

[11] R. Richardson (2008). 2008 CSI Computer Crime and Security Survey.

[12] V.R. Westmark (2004). A Definition for Information System Survivability Proceedings of the 37th IEEE Hawaii International Conference on System Sciences.

[13] D.A. Fisher, H.F. Lipson, N.R. Mead, R.C. Linger, R.J. Ellison, and T. Longstaff (1997, Revised: May 1999). Survivable Network Systems: An Emerging Discipline. Technical Report CMU/SEI-97-TR-013.

[14] M.N. Azaiez and V.M. Bier (2007). Optimal Resource Allocation for Security in Reliability Systems. European Journal of Operational Research, 181(2), 773-786.

[15] A.M. Geoffrion (1974). Lagrangean Relaxation for Integer Programming. Mathematical Programming Study, 2, 82-114.