# An Evolutionary Multi-objective Approach for Modelling Network Security

Seyed Mahmood Hashemi, Jingsha He
*(Corresponding author: Seyed Mahmood Hashemi)*
School of Software Engineering, Beijing University of Technology
Beijing Engineering Research Center for IoT Software and Systems
(Email: Hashemi2138@yahoo.com)
*(Received Feb. 20, 2016; revised and accepted July 19 & July 31, 2016)*

## Abstract

Security is the most important issue in a network system. Administrators can more easily understand threats to the network by using a model. In this paper, we present an approach for modelling a network that considers the benefits of the network as well as its limitations. In our approach, we model the system as an optimization problem, which is solved using three algorithms. As the proposed approach is stochastic, it works very efficiently in a network environment. This paper, presents a mathematical model of the system. Model provides easy comprehend of system. Presented model is based on multi-objective optimization problem. One parameter in the presented model is security and another parameter is user productivity. Security is the most important issue in a network system. Administrators can more easily understand threats to the network by using a model. In this paper, we present an approach for modelling a network that considers the benefits of the network as well as its limitations.

*Keywords: Modelling network security, multi-objective approach, network system, optimization*

## 1 Introduction

A model is a tool that facilitates creating a representation of the target object, thereby helping the users to understand that object. A model is necessary for understanding network systems, because these systems typically comprise many sub-systems and having knowledge about all sub-systems is practically impossible. The importance of a model of a network system increases when considering security. Security is the most important issue in a network system and a higher degree of security is constantly being sought. Focusing on security, we can divide network systems into two main groups: open and closed systems. In the first group, network systems are free to join the network. In other words, all machines in the network can access their assets. Despite the open system, no machine in the network can access the assets of a closed system. In practice, because this division is absolute, many network systems fall between open and closed systems in the real world. The main goal of a security model is to represent the security level of a network system.

Security is the generic term for a collection of techniques and tools designed to protect data and prevent counter attacks [7]. Security involves three aspects: confidentiality means hiding the contents of a file, integrity means detecting tampering, and availability means ensuring access to assets. All three security aspects can be applied using an authorization system. In this context, confidentiality means unauthorized disclosure of information, integrity means unauthorized modification, and availability means denial of unauthorized access to information. A security model clearly depicts the level of authorization for each sub-system.

The major benefit of a network is user productivity. In other words, a network is created to facilitate access to users favorite data resources. Therefore, application of security should not be limited to users access to assets.

Contrarily, certain constraints are applicable to an authorization system. The main constraint is an economic one. Given that the financial resources of an organization are normally limited, costs must be constrained. As such, there are two conflicting goals for security (increasing authority as well as user productivity) and one constraint (the economic issue). Any network security model must consider the goals and the constraint. In this paper, we present a model based on evolutionary multi-objective optimization (EMO). We use an evolutionary algorithm because it can adapt to the dynamic nature of a network, and multi-objective optimization because it allows us to optimize a number of conflicting objectives. EMO allows us to optimize Confidentiality, Integrity, Availability and User Productivity simultaneously.

The rest of this paper is organized as follows: In Section 2, we define a number of preliminaries that are needed for our proposed algorithm. We also present an overview of related research. In Section 3, we introduce our pro-

posed algorithm together with our experimental results. Finally, our conclusions are presented in Section 4.

## 2 Related Works

The evolution of the current industrial context and the increase of competition pressure, has led companies to adopt new concepts of management [4]. The implementation of the most important part of the plan phase, consisting of the definition of an appropriate global management plan QSE (Quality, Security and Environment) has been proposed [3]. This implementation is based on the multi-objective influence diagrams (MIDs) [21]. The proposed approach has three phases: Plan phase, Do phase and Check & Art phase. The first phase gathers all quality, security and environmental objectives issued from the requirements, and then analyzes them. In this phase we can define a global management QSE plan. The second phase has the input of the global management plan QSE and the corresponding global monitoring plan generated from the plan phase and will also implement the selected treatments. In the third phase, finalization of the process of integration occurs through measuring the effectiveness of different decisions. Neubauer et al. provide a structured and repeatable process that includes: defining evaluation criteria according to corporate requirements, strategy, assessing and/or refining the existing IT security infrastructure, identifying stakeholder preferences (risks, boundaries), determining the solution space of all efficient (Pareto optimal) safeguard portfolios, and interactively selecting the individually best safeguard portfolio [23]. This paper tries to combine different benefits and costs into one formula. This presents a problem because the authors do not present a multi-objective optimization problem. Kumar et al. focus on PGP (pretty good privacy) [19], which was shown by Zimmerman in 1991 to provide security with available cryptographic algorithms [27]. Algorithms are chosen according to the user requirements of time, cost and required security level. Kumar et al. answer the question: How do you choose appropriate algorithms, from the available pool, to suit the user requirements of time, cost and security? They assign a security level to an algorithm according to its performance P. Authors of [29] investigate security models, which consider risk assessment approaches to be applied for threat modelling, network hardening and risk analysis. Overall, security models can be classified based on the methodologies used to optimally invest into computer security. We have specified the following:

- Risk assessment models;
- Cost-benefit models;
- Game models;
- Multi-objective decision support models.

Cost-benefit analysis looks into intangible costs/returns and addresses the perspective of time. The simplicity of the frameworks can give suitable investment solutions for low risk investments. However, these methods do not consider uncertainty and give misleading indications for long-term investments. In [30], the risk assessment involves a calculation of risk in relation to financial returns, rather than the defined risk of possible losses related to degradation of information security. They demonstrate a novel approach of selecting security countermeasures with respect to both investment cost and the risk of possible degradation of CIA. Their security countermeasure is represented as a binary value. Also, they thought security solutions can be classified based on the function they provide The main challenge Information System (IS) managers face is to strike an appropriate balance between risk exposure and the opportunity to mitigate risk through investments in security. Thus, the authors of [17] propose a decision analytical approach, but the paper does not present a formula for multi-objective optimization. Service provisioning (SP) is defined as the set of interrelated decisions in order to select a service (by a server) to attend to a request (by a client). In [25], the results of the author case study provides evidence in support of the notion that the use of imitation (recall) in DPSP (dynamic provider of service provision) cipher selection process reduces its overheads dramatically. In paper [24], the authors introduce a novel presentation for cyber security problems using the formalization of a Multi-Objective Distributed Constraint Optimization Problem (MO-DCOP). An MO-DCOP is the extension of a mono-objective Distributed Constraint Optimization Problem (DCOP) which is a fundamental problem that can formalize various applications related to multi-agent cooperation. They develop a novel algorithm called Branch and Bound search algorithm (BnB) for solving a cyber security problem. This algorithm utilizes the well-known and widely used branch and bound technique and depth-first search strategy and finds all trade-off solutions. The purpose of any risk analysis is providing decision makers with the best possible information about the probability of loss [6]. Behnia et al. compare several different approaches for risk analysis and declare the weakness and strength for each of them.

## 3 Preliminaries

In this section, we discuss other approaches for modelling network systems, which can be divided into two groups: attack trees and stochastic models.

### 3.1 Attack Tree

An attack tree is one of the main methods for system modelling. In this approach, assets and their related threats are specified simply. Figure 1 shows an example of an attack tree [31], in which nodes depict the desired actions and edges show the required processes. Depending on the type of tree and the type of protection system, nodes and edges may have different values.

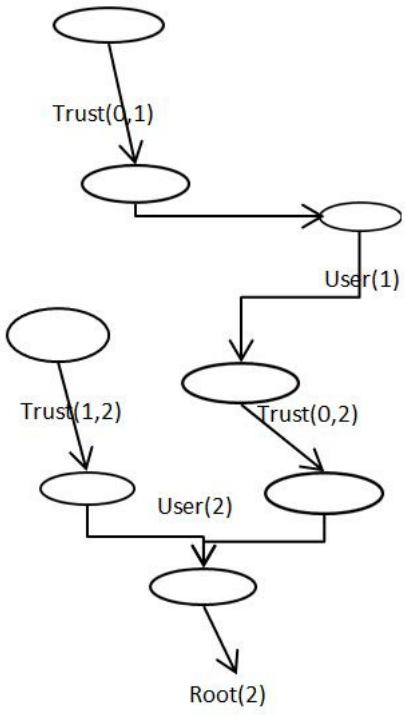


Figure 1: Attack tree

The attack tree models attacking behavior by enumerating all attack scenarios aimed at compromising the root goal. The attack tree model has three components: the root, branches, and leaves. Challenges between security policies and threats are represented by leaves. Branches, which are classified as AND/OR types, move the result from a sub-tree to its ancestor. This process continues until the value reaches the root, at which stage the administrator can make a decision based on the value in the root. Despite the varied use of attack trees, they do suffer from certain problems. Not all useful information about vulnerable systems can be translated into an attack tree. This is a very serious problem because subsequent analysis is sensitive to correct values in the leaves. Because security is a trade-off between user productivity and enhanced security levels, each partner of the system endpoints could have different requirements. Moreover, because an attack tree is static, it is valid for only a limited time.

Many researchers have endeavored to extend the attack tree to suit their target application.

- Dugan et al. presented fault trees [11], which use additional elements such as dependency gates. Fault trees model events according to their exponential distribution. However, there is no evidence that the probability of attack success follows an exponential distribution.

- Fung et al. proposed a MANET network [13]. Their study incorporated the notion of survivability into attack trees. Survivability analysis finds those system components susceptible to attacks and analyzes their ability to survive such attacks.

- Bistarelli et al. proposed the defense tree [8]. Their study, which included quantitative metrics such as return on investment and return on attack, extended the attack tree by using countermeasures to address intrusion attempts at the leaves of the trees.

- Dalton et al. proposed a conversion tree [10]. In this work, steady-state analysis of the resulting generalized stochastic Petri net was performed. Details of attack scenarios can be found in [16].

## 3.2 Stochastic Model

Stochastic models convert the state of the system to a Markov chain, and then analyze it using a steady-state transition matrix. The term stochastic means predicting a set of possible outcomes by their probabilities.

The main property of a Markov chain is that no state can influence the next state. In other words, the probability of any particular future behavior of the process is not altered by additional knowledge about its past behavior. A Markov process is completely defined once its transition probability matrix and initial state have been defined.

Stochastic models have been used extensively in research studies:

- Mandan et al. proposed a model of the behavior of an intrusion tolerant system [20]. This work uses a generic state diagram as a semi-Markov process model that is later solved using an embedded discrete time Markov chain. Quantitative analysis of the model produces two useful metrics: steady-state availability and mean time to security failure.

- Sallhammer et al. used a stochastic model for security and dependability evaluation [26]. This work used game theory to model attack behavior.

### 3.2.1 Multi-objective Optimization

Optimization is a common topic in many scientific fields. When the target of an optimization problem is a single object, we can model the problem as a single-objective optimization problem. Conversely, if the problem has multiple objectives, we model the problem as a multi-objective optimization problem. However, in many problems, objectives conflict with each other. A multi-objective optimization problem is defined as follows [2].

"A vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term 'optimize' means finding such a solution which would give the values of all the objective functions acceptable to the decision maker".

Real world applications frequently have several conflicting objectives. Recently, there has been increased research focus on EMO algorithms. Multi-objective optimization problems (MOPs) are defined as follows [9]:

$$\begin{aligned} \text{Optimize} \quad & [f_1(X), f_2(X), \ldots, f_k(X)] \\ \text{Subject to:} \quad & g_i(X) \leq 0; i = 1, 2, \ldots, m \\ & h_j(X) = 0; j = 1, 2, \ldots, p \end{aligned} \tag{1}$$

where $k$ is the number of objectives, $X$ is a vector of decision variables, $m$ is the number of inequality constraints, and $p$ is the number of equality constraints. The notion of "optimize" in Equation (1) implies setting the decision variables in such a way as to achieve Pareto optimality. We say that a vector of decision variables $X^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $X \in \mathcal{F}$ such that $f_i(X) \leq f_i(X^*)$ for all $i = 1, \cdots, k$ and $f_j(X) < f_j(X^*)$ for at least one $j$. If vector $X^*$ is included in the *Pareto-optimal* set, it is called a non-dominated solution. A vector $\vec{u} = (u_1, u_2, \ldots, u_k)$ is said to *dominate* vector $\vec{v} = (v_1, v_2, \ldots, v_k)$ (denoted by $\vec{u} \preccurlyeq \vec{v}$) if and only if $\vec{u}$ is partially more optimum than $\vec{v}$, i.e., $\forall i \in \{1, \ldots, k\} \Rightarrow u_i \leq v_i, \exists j \in \{1, \ldots, k\} \Rightarrow u_j < v_j$.

Many algorithms have been developed to solve MOPs. In this paper, we use three of these: multi-objective simulated annealing (AMOSA), multi-objective genetic algorithm (MOGA), and multi-objective bee colony (MOBC).

## A. Multi-objective Simulated Annealing (AMOSA)

The basic concept in simulated annealing is the evolution of the solution by simulating decreasing temperature (tmp) in the material, where a higher temperature denotes greater modification of the solution in a generation. If the temperature of a hot material decreases very quickly, its internal structure may change and the material could become hard and brittle. Decreasing the temperature slowly yields higher homogeneity and less brittle material. Evolution of the solution occurs at specific temperature profiles. In the first few iterations, a diverse set of initial solutions for the problem are produced at a higher temperature. These solutions are then evolved while the temperature decreases to obtain their local optima. In a multi-objective situation, there are non-dominated solutions that must be kept in the archive as candidates for the optimal solution.

AMOSA was proposed in [5]. During the execution of the AMOSA algorithm, two solutions exist: the current-so and new-so. Comparison of the two solutions yields one of three states: (i) current-so dominates new-so, (ii) current-so and new-so are non-dominated with respect to each other, and (iii) new-so dominates current-so.

If new-so is dominated by current-so, there may be solutions in the archive that dominate new-so. New-so is accepted into the archive based on the probability:

$$p = \frac{1}{1 + \exp(\Delta * \text{tmp})'} \tag{2}$$

where $\Delta$ is the difference between new-so and the other solutions that dominate new-so. If there are A solutions in the archive,

$$\Delta = (\Sigma_{i=1}^{A} \Delta_i + \Delta)/(A + 1) \tag{3}$$

Solutions can escape from local optima and reach the neighborhood of the global optima by this probable acceptance.

If new-so is dominated by some solutions in the archive, Equation (3) is modified to:

$$\Delta = (\Sigma_{i=1}^{A} \Delta_i)/A. \tag{4}$$

If new-so is not dominated by any of the members in the archive, new-so is set to current-so and is added to the archive.

If new-so dominates some solutions in the archive, new-so is set to current-so and is added to the archive. In addition, any solutions in the archive that are dominated by new-so, are removed.

If new-so is dominated by some solutions in the archive, Equation (2) is changed to:

$$p = \frac{1}{1 + \exp(-\Delta)'} \tag{5}$$

where $\Delta$ is the minimum difference between new-so and the dominating solutions in the archive. New-so is set to current-so with probability Equation (5). If new-so is not dominated by any of the solutions in the archive, it is set to current-so and added to the archive. If new-so dominates some solutions in the archive, it is set to current-so and added to the archive, while all dominated solutions are removed from the archive.

## B. Multi-objective Genetic Algorithm (MOGA)

MOGA, which is based on a single-objective genetic algorithm [12], [15] and [18], comprises various stages. In the first stage, a population of individuals (chromosomes) is created. The number of individuals in the population (pop-size) is determined by the programmer. Each individual contains certain fields, where the number of fields in an individual is equal to the number of variables in the problem, which must be optimum. Each individual has the potential to reach the optimum point, at which optimal values are set in the corresponding fields in the individual. In the first stage of MOGA, all individuals in the population are initialized with random values. The algorithm runs until the stopping conditions are met. There are three types of stopping conditions. The first of these is special values; when the values of individuals are equal

to the default values, the algorithm terminates. The second type of stopping condition occurs when the values of individuals no longer change. The last type of stopping condition is the number of iterations. When the number of iterations of the algorithm reaches the given threshold value (max-generation), the algorithm terminates.

Given that MOGA is an evolutionary algorithm, it is executed for a number of iterations, where each iteration of MOGA is called a generation, inspired by Darwinian evolutionary theory. The programmer can control the evolutionary nature of MOGA using the number of generations. This means that despite the deterministic optimization method, which is controlled by the number of inputs, the programmer can vary the number of generations. In the first generation, individuals are initialized with random values. The values of individuals are changed in each generation using two operators: mutation and cross-over. In mutation, one field of an individual is changed to a different value. There are a number of different methods for mutation, which describe the quality of the altered values. In cross-over, two individuals are combined to produce a new individual. After the genetic algorithm operators (mutation and cross-over) have been applied, several individuals are selected for the next generation. Selection is done stochastically according to the fitness of the individual.

The goal of the optimization algorithm is to find the optimal point. Optimal points can be divided into two categories: local optima and global optima. A local optimum can be any point that is the optimum of all points within a limited range, while a global optimum is a point that is the optimum of all points in an unlimited range. Because deterministic optimization methods compare the current point with points in a limited range, they may be trapped in a local optimum. The stochastic feature of MOGA allows the algorithm to escape from local optima and achieve the global optimum.

Based on the discussion above, MOGA has two advantages: the programmer can control the execution time and the algorithm has the potential to achieve a global optimum point.

MOGA finds an optimum point according to the Pareto set; in other words, a point is optimum if it is not dominated by other points. Indeed, the Pareto principle allows a number of objectives to become optimum simultaneously. Each individual is checked for its domination in the population. Individual i is allocated a rank equal to one plus the number of individuals, $n_i$, dominating individual i. Once ranking has been completed, a raw fitness is assigned to each individual based on its rank using a linear mapping function.

$$F_i = N - \Sigma_{k=1}^{r_i-1}\mu(k) - 0.5\,(\mu(r_i - 1)) \tag{6}$$

where $\mu$ denotes the numbers of individuals in the rank. MOGA incorporates niching among individuals in each rank. The niche count with $\sigma_$share is found first. The distance metric is computed with the objective function values. Thus, the normalized distance between any two individuals $i$ and $j$ in a particular rank is calculated as:

$$d_{ij} = \sqrt{\Sigma_{k=1}^{M}\left(\frac{f_k^{(}i) - f_k^{(j)}}{f_k^{\max} - f_k^{\min}}\right)^2}. \tag{7}$$

The distance is computed for each pair of individuals. Therefore, the niche count is calculated by summing the shared function values:

$$SH(d_{ij}) = \begin{cases} 1 - \dfrac{d_{ij}}{\sigma\_share}, & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0, & \text{otherwise} \end{cases} \qquad nc_i = \Sigma d_{ij}. \tag{8}$$

The shared fitness is calculated as $F_i' = F_i/nc_i$. Shared fitness is used as a basis for stochastically selecting individuals for the next generation.

The above process continues until the stopping condition is satisfied. When the algorithm terminates, the remaining individuals represent the optimum.

## C. Multi-objective Bee Colony (MOBC)

The foraging behavior of bees is characterized by various steps that are used in optimization. The first step is called the Waggle Dance, which is used by bees to convey information to other bees about the direction, distance, and quality of a food source. Upon finding a food source, a bee begins to dance in a figure of eight pattern. The second step in the foraging behavior is following. In this step, follower bees that were waiting inside the hive, follow the dancer bee. The number of follower bees assigned to a path is directly proportional to the quality of the path. In the third step these bees return to the hive. More bees are recruited to the source of the food if the path is still good enough. Bees stop collecting poor-quality food and adjust their strategy for finding food based on information about the location of good-quality food.

Foraging behavior can be used for optimization when it is divided into two phases. The first phase consists of path construction. In this phase, a bee explores the entire food source, but with the exploration limited by constraints. When a bee does a tour (which includes all possible variables), it performs the Waggle Dance. Other bees use this information, expressed as:

$$\text{Pf}_i = \frac{1}{L_i} \tag{9}$$

where $Pf_i$ is the profitability of a bee$_i$ and $L_i$ is its tour. If a colony has $n$ bees, the bee colony average profitability is given by:

$$\text{Pf}_{\text{colony}} = \frac{1}{n}\Sigma_{i=1}^{n} \qquad \text{Pf}_i = \frac{1}{n}\Sigma_{i=1}^{n}\frac{1}{L_i}. \tag{10}$$

The dance duration of any bee is given by:

$$D_i = K * \frac{\text{Pf}_i}{\text{Pf}_{\text{colony}}}, \tag{11}$$

where $K$ is the profitability rating and is adjusted according to the lookup table given in Table 1.

Table 1: Lookup table for adjusting profitability

| Profitability Rating | $K_i$ |
|---|---|
| $\mathrm{Pf}_i < 0.9\mathrm{Pf}_{\mathrm{colony}}$ | 0.60 |
| $0.9\mathrm{Pf}_{\mathrm{colony}} < \mathrm{Pf}_i < 0.95\mathrm{Pf}_{\mathrm{colony}}$ | 0.20 |
| $0.95\mathrm{Pf}_{\mathrm{colony}} < \mathrm{Pf}_i < 1.15\mathrm{Pf}_{\mathrm{colony}}$ | 0.02 |
| $1.15\mathrm{Pf}_{\mathrm{colony}} < \mathrm{Pf}_i$ | 0.00 |

The second phase of the bee algorithm consists of path reconstruction. In this phase, bees in the hive, having received information from the explorer bee, utilize the path. Bees use a transition rule for choosing the appropriate path with the probability denoted by $P_{ij}(t)$, which measures the possibility of moving from $\mathrm{step}_i$ to $\mathrm{step}_j$ at time $t$. In a multi-objective sense, the discussed path must be examined for dominance over other paths. Formula (12) takes into consideration the fitness of all paths:

$$\rho_{ij}(t) = \begin{cases} \lambda & j \in F_i(t) \\ \dfrac{1-\lambda|F_i(t)\cap A_i(t)|}{|A_i(t)|-|F_i(t)\cap A_i(t)|} & j \notin F_i(t) \end{cases} \quad (12)$$

where $\lambda$ is the value (less than one) assigned to the preferred path, $|A_i(t)|$ is the number of allowed next steps, and $|F_i(t) \cap A_i(t)|$ is the number of preferred next steps [1, 14, 22, 28].

Now, we can examine the dominance of all paths according to Section 3.2.1, after which each path is classified as conforming to one of three situations:

1) Dominates another path(s),

2) is dominated by another path, and

3) is not dominated by any other path.

In the first situation, the path is stored in the archive. In the second situation, the path is destroyed, and in the third situation, the path is stored in the archive with the following probability:

$$P_{ij}(t) = \frac{[\rho_{ij(t)}]^{\alpha} * [\frac{1}{d_{ij}}]^{\beta}}{\Sigma_{j\in A_i(t)}[\rho_{ij(t)}]^{\alpha} * [\frac{1}{d_{ij}}]^{\beta}} \quad (13)$$

where $d_{ij}$ is the distance between $\mathrm{step}_i$ and $\mathrm{step}_j$, $\alpha$ is a variable that influences the fitness, and $\beta$ is a variable that influences the distance. A is a collection of all steps that can be reached from the previous step.

# 4 Proposed Algorithm

First, we give an overview of the system. We set up a system with three assets in the network environment. Here, security implies creating confidentiality, integrity, and availability of these assets.

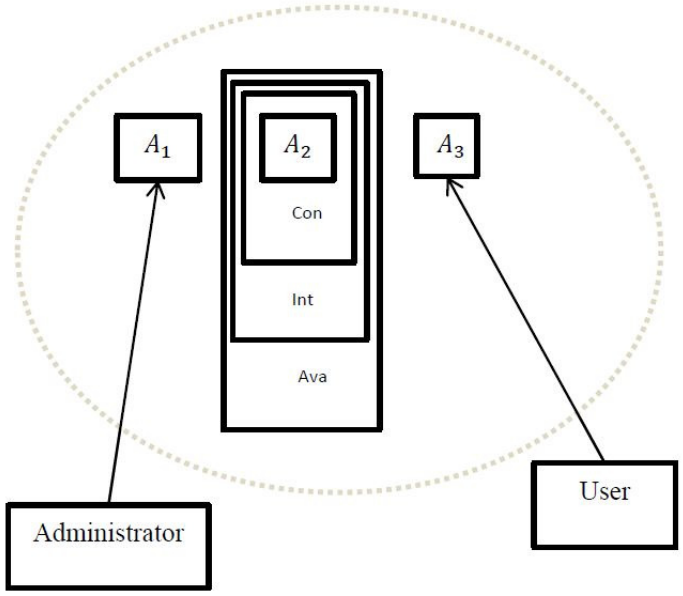


Figure 2: Overview of proposed algorithm

In Figure 2, A represents the assets and *Con*, *Int* and *Ava* denote confidentiality, integrity, and availability. Second, there is a need to create a model of the system. According to the above hits, we use the following optimization model to represent the security of the network model:

**Optimize**

$$\text{Security}\,(A_1, A_2, \ldots, A_n), \text{User Productivity}\,(A_1, A_2, \ldots, A_n)$$

$$\begin{aligned} \text{Subject to: } & \mathrm{Con}(A_1) \le M_1, \mathrm{Con}(A_2) \le M_2, \mathrm{Con}(A_3) \le M_3 \\ & \mathrm{Int}(A_1) \le N_1, \mathrm{Int}(A_2) \le N_2, \mathrm{Int}(A_3) \le N_3 \\ & \mathrm{Ava}(A_1) \le K_1, \mathrm{Ava}(A_2) \le K_2, \mathrm{Ava}(A_3) \le K_3 \end{aligned} \quad (14)$$

where $A_1, A_2, A_3$ are assets in the system, *Int*, *Ava* and *Con* denote the integrity, availability, and confidentiality of the assets, and $M, N, K$ are economic issues applied to each security concept. Security is denoted by the cost function and Optimize means simultaneously maximizing the confidentiality, integrity, and availability of the assets as well as user productivity. Let $M_1 = 5, M_2 = 6, M_3 = 4, N_1 = 4, N_2 = 5, N_3 = 7, K_1 = 5, K_2 = 4, K_3 = 5$ in the range $\{0, 5\}$. These assumptions do not limit the generalization of our modelling. We solve Equation (14) using the three algorithms described in Section 2.2.1, namely, AMOSA, MOGA, and MOBC. The desired levels for confidentiality, integrity, and availability of the assets and user productivity ('user productivity' is defined in Section 1) are in the range $\{0, 5\}$. The final results are listed in Table 2.

Table 2: Final results

| | $Confidentiality(A_1)$ | $Confidentiality(A_2)$ | $Confidentiality(A_3)$ |
|---|---|---|---|
| AMOSA | 5 | 6 | 4 |
| MOGA | 5 | 5 | 4 |
| MOBC | 4 | 5 | 3 |
| | $Integrity(A_1)$ | $Integrity(A_2)$ | $Integrity(A_3)$ |
| AMOSA | 4 | 5 | 7 |
| MOGA | 4 | 4 | 7 |
| MOBC | 4 | 4 | 6 |
| | $Availability(A_1)$ | $Availability(A_2)$ | $Availability(A_3)$ |
| AMOSA | 5 | 4 | 5 |
| MOGA | 4 | 4 | 4 |
| MOBC | 4 | 4 | 5 |
| | $user\ Productivity(A_1)$ | $user\ Productivity(A_2)$ | $user\ Productivity(A_3)$ |
| AMOSA | 9 | 8 | 7 |
| MOGA | 8 | 6 | 7 |
| MOBC | 6 | 7 | 6 |

# 5 Conclusion

In this paper, we presented an approach for modelling network security. The proposed approach is based on EMO. The application of security has two goals (security aspects and user productivity); therefore, we use a multi-objective optimization. In the model, we consider economic limitations applied to the various security aspects. We use an evolutionary method in the proposed approach, because the nature of networks is dynamic.

The model uses three EMO algorithms, all of which are stochastic. This means that different runs may produce different results, but some results are worth highlighting. AMOSA produces the best result, where best means greater maximization of all goals. In future works, we intend to consider a proper unifier for each goal (for example, a fuzzy set).

# Acknowledgments

The work in this paper has been supported by National Natural Science Foundation of China (61272500), National High-tech R&D Program (863 Program) (2015AA017204) and Beijing Natural Science Foundation (4142008).

# References

[1] P. Agrawal, H. Kaur, and D. Bhardwaj, "Analysis and synthesis of enhanced bee colony optimization with the traditional bee colony optimization to solve traveling sales person problem," *International Journal of Computer & Technology*, vol. 2, no. 2, pp. 93–97, 2012.

[2] A. Ameljariczyk, "Multicriteria optimization in engineering design," in *Computing in Civil Engineering*, pp. 318–325, 1994.

[3] A. Badreddine, T. B. Romdhane, and N. B. Amor, "A multi-objective risk management approach to implement an integrated management system: Quality, security, environment," in *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4728–4733, 2009.

[4] A. Badreddine, T. B. Romdhane, and N. B. Amor, "A new process-based approach for implementing an integrated management system: Quality, security, environment," in *Proceedings of the 2009 International Conference on Industrial Engineering*, vol. 2, pp. 18–20, 2009.

[5] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multi-objective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.

[6] A. Behnia, R. A. Rashid, and J. A. Chaudhry, "A survey of information security risk analysis methods," *Smart Computing Review*, vol. 2, no. 1, pp. 79–94, Feb. 2012.

[7] M. Bishop, *Computer Security: Art and Science*, Tsinghua Press, pp. 3–10, 2004.

[8] S. Bistarelli, F. Fioravanti, and P. Peretti, "Defense trees for economic evaluation of security investments," in *The First International Conference on Availability, Reliability and Security*, pp. 8, 2006.

[9] C. A. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-objective Problems*, vol. 242, New York: Kluwer Academic, 2002.

[10] G. C. Dalton, R. F. Mills, J. M. Colombi, and R. A. Raines, "Analyzing attack trees using generalized

stochastic petri nets," in *IEEE Information Assurance Workshop*, pp. 116–123, 2006.

[11] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 363–377, 1992.

[12] C. M. Fonseca and P. J. Fleming, "Genetic algorithm for multiobjective optimization: Formulation, discussion and generalization," in *Proceeding of* $5^{th}$ *International Conference on Genetic Algorithms*, pp. 416–423, 1993.

[13] C. Fung, Y. L. Chen, X. Wang, J. Lee, R. Tarquini, and M. Anderson, "Survivability analysis of distributed systems using attack tree methodology," in *IEEE Military Communications Conference (MILCOM'05)*, pp. 583–589, 2005.

[14] M. Gupta and G. Sharma, "An efficient modified artificial bee colony algorithm for job scheduling problem," *International Journal of Soft Computing and Engineering*, vol. 1, no. 6, pp. 291–296, 2012.

[15] A. Haidine and R. Lehnert, "Multi-case multi-objective simulated annealing (mc-mosa): New approach to adopt simulated annealing to multi-objective optimization," *World Academy of Science*, vol. 4, no. 3, pp. 9, 2008.

[16] M. V. Higuero, J. J. Unzilla, E. Jacob, P. Saiz, M. Aguado, and D. Luengo, "Application of attack trees in security analysis of digital contents e-commerce protocols with copyright protection," in *39th Annual 2005 International Carnahan Conference on Security Technology*, pp. 57–60, 2005.

[17] E. Kiesling, C. Strau, and C. Stummer, "A multi-objective decision support framework for simulation-based security control selection," in *IEEE Seventh International Conference on Availability, Reliability and Security*, pp.454–462, 2012.

[18] A. Konak and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[19] D. Kumar, D. Kashyap, K. K. Mishra, and A. K. Misra, "Security vs cost: An issue of multi-objective optimization for choosing PGP algorithms," in *IEEE International Conference on Computer and Communication Technology*, pp. 532–535, 2010.

[20] B. B. Madan, K. Gogeva-Popstojanova, K. Vaidyanathan, and K. S. Trivedi, "Modeling and quantification of security attributes of software systems," in *International Conference on Dependable Systems and Networks*, pp. 505–514, 2002.

[21] D. Micheal and Y. H. Yacov, "Influence diagrams with multiple objectives and tradeoff analysis," *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 34, no. 3, pp. 293–304, 2004.

[22] R. Murugan and M. R. Mohan, "Artificial bee colony optimization for the combined heat and power economic dispatch problem," *ARPN Journal of Engineering and Applied Sciences*, vol. 5, no. 7, 2012.

[23] T. Neubauer, C. Stummer, and E. Weippl, "Workshop-based multiobjective security safeguard selection," in *Proceedings of IEEE First International Conference on Availability, Reliability and Security*, pp. 8, 2006.

[24] T. Okimoto, N. Ikegai, T. Ribeiro, K. Inoue, H. Okada, and H. Maruyama, "Cyber security problem based on multi-objective distributed constraint optimization technique," in *43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop*, pp. 1–7, 2013.

[25] J. Raissi, "Performance impact of imitation in multi-objective security service provisioning," in *Proceedings of IEEE*, pp. 1–6, 2013.

[26] K. Sallhammar, B. E. Helvik, and S. J. Knapskog, "Towards a stochastic model for integrated security and dependability," in *IEEE First International Conference on Availability, Reliability and Security (ARES'06)*, pp. 8, 2006.

[27] W. Stallings, *Cryptography and Network Security Principles and Practices*, Pearson Education, India, 2004.

[28] S. Suriya, R. Deepalakshmi, S. S. Kannan, and S. P. Shantharajah, "Enhanced bee colony algorithm for complex optimization problems," *International Journal on Computer Science and Engineering*, vol. 4, no. 1,pp. 72, 2012.

[29] V. Viduto, W. Huang, and C. Maple, "Toward optimal multi-objective models of network security: Survey," in *Proceedings of IEEE the 17th International Conference on Automation & Computing*, pp. 6–11, 2011.

[30] V. Viduto, C. Maple, W. Huang, and A. Bochenkov, "A multi-objective genetic algorithm for minimizing network security risk and cost," in *IEEE International Conference on High Performance Computing and Simulation*, pp. 462–467, 2012.

[31] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metrics," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 283–296, 2008.

## Biography

**Seyed Mahmood Hashemi** received his bachelor from Islamic Azad University (Qazvin Branch) in software engineering at 2001 his master from Islamic Azad University (Science and Research Branch) in artificial intelligence at 2003. He is currently PhD candidate in Beijing University of Technology (BJUT). His research interests are Internet of Things (IoT), network security and Artificial Intelligence (AI).

**Jingsha He** received his Master and doctoral degrees in computer engineering from the University of Maryland

at College Park in the US. He is currently a professor in the School of Software Engineering at Beijing University of Technology (BJUT) in Beijing, China. Prior to joining BJUT in 2003, Prof. He worked for several multi-national companies such as IBM Corp., MCI Communications Corp. and Fujitsu Labs in the US. where he published more than 10 papers and received 12 U.S. patents. Since joining BJUT in 2003, Prof. He has published nearly 240 papers in journals and international conferences, received nearly 40 patents and 30 software copyrights in China and co-authored 7 books. He has been the principal investigators of more than 20 research projects. Prof. Hes research interests include information security, wireless networks and digital forensics.