Expert Systems With Applications 64 (2016) 11–23

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Optimizing security and quality of service in a Real-time database system using Multi-objective genetic algorithm

Xuancai Zhao, Qiuzhen Lin*, Jianyong Chen, Xiaomin Wang, Jianping Yu, Zhong Ming

College of Computer Science and Software Engineering, *Shenzhen University*, *Shenzhen*, *P.R. China*, *518060*

ARTICLE INFO

Article history:
Received 3 June 2015
Revised 16 July 2016
Accepted 17 July 2016
Available online 18 July 2016

Keywords:
Multi-objective optimization
Genetic algorithm
Network security
QoS

ABSTRACT

Both network security and quality of service (QoS) consume computational resource of IT system and thus may evidently affect the application services. In the case of limited computational resource, it is important to model the mutual influence between network security and QoS, which can be concurrently optimized in order to provide a better performance under the available computational resource. In this paper, an evaluation model is accordingly presented to describe the mutual influence of network security and QoS, and then a multi-objective genetic algorithm NSGA-II is revised to optimize the multi-objective model. Using the intrinsic information from the target problem, a new crossover approach is designed to further enhance the optimization performance. Simulation results validate that our algorithm can find a set of Pareto-optimal security policies under different network workloads, which can be provided to the potential users as the differentiated security preferences. These obtained Pareto-optimal security policies not only meet the security requirement of the user, but also provide the optimal QoS under the available computational resource.

1. Introduction

Database systems are widely used in today's computer system, which are adopted for storing and accessing data in various application services (Andres, Jose, Ernesto, & Alfredo, 2013; Hababeh, Khalil, & Khreishah, 2015; Tang, Li, Jiang, & Chen, 2014). With the expansion of database application fields, the new applications not only need to maintain a large amount of shared data, but also keep the data fresh for the transaction, such as data communications, e-commerce, and real-time simulation. For traditional database system, it is designed to process the permanent, stable data, and maintain the integrity and consistency of data. Its performance targets mainly focus on the high throughput and the low cost of system. Whereas, a real-time database is designed to use the real-time processing, such that it can handle the workloads whose state is constantly changing (Laura, Jorge, & Viviana, 2005).

With the widespread use of database systems, they are exposed to more and more internal and external threats (Al-Sayid & Aldlaeen, 2013; Poolsappasit, Dewri, & Ray, 2012), as the data stored in databases always involve much sensitive information, such as personal privacy, bank information and commercial secrets. More and more real-time services in database are required, which will highly impact the quality of service (QoS). Real-time database system has become the basis of enterprise information data platform, which is used to process the real-time transaction data for the e-commerce system of the enterprise, to simulate and monitor the system performance for simulation system of the laboratory or to storage historical data for data sharing platform and so on (Laura et al., 2006). Since the mutual influence between network security and QoS, there is a growing interest to figure out their actual relationship on database systems. For example, with the increasing use of the real-time network application services that contain sensitive information, it is required to provide the adequate security service for maintaining the users' security and high QoS to satisfy the real-time requirements.

In most cases, security and QoS are investigated independently. On the improvement of QoS, over the past decades, a lot of research studies on the QoS of real-time database have been conducted (Amirijoo, Hansson, & Son, 2006; Kang, Oh, & Son, 2007a; Kang, Son, & Stankovic, 2004; Woochul, Son, & Stankovic, 2012). Traditional security mechanisms such as access control mechanisms (Bertino & Sandhu, 2005; Parmar, 2014) and policy enforcement mechanisms (Jabbour & Menasee, 2008; Jabbour & Menasee, 2009) are not sufficiently secure for database system, as the anomaly detection mechanisms are required to protect database system against the potential threats such as SQL injection and impersonation attacks (Kamra & Bertino, 2009; Srivastava, 2014). Thus, intrusion detection and prevention systems (IDPSs)

* Corresponding author.
E-mail addresses: zhaoxcszu@gmail.com (X. Zhao), qiuzhlin@szu.edu.cn (Q. Lin), jychen@szu.edu.cn (J. Chen), wangxm@szu.edu.cn (X. Wang), yujp@szu.edu.cn (J. Yu), mingz@szu.edu.cn (Z. Ming).

http://dx.doi.org/10.1016/j.eswa.2016.07.023
0957-4174/

Table 1
The contributions of the references regarding the improvement of security and QoS.

References	Contributions	Category
Amirijoo et al., 2006; Kang et al., 2004	Feedback control has been applied to real-time database to maintain data freshness for the timeliness of transactions in dynamic workloads.	the improvement of QoS
Woochul et al., 2012	By controlling both I/O and CPU resources, the proposed approach supports both the timeliness of transactions and the high data freshness in real-time database.	
Bertino & Sandhu, 2005; Jabbour & Menasee, 2008; Jabbour & Menasee, 2009; Parmar, 2014	Their researches illustrate that the traditional security mechanisms such as access control mechanisms and policy enforcement mechanisms are not sufficiently secure for database system.	the improvement of security
Kamra & Bertino, 2009; Srivastava, 2014	The anomaly detection mechanisms are used to protect database system against the potential threats such as SQL injection and impersonation attacks.	
Darwish et al., 2013; Rao et al., 2014; Saad et al., 2012	IDPSs have been extended to protect database system from malicious intrusions.	
Taneja et al., 2011	It illustrates that network security services in some application cases will consume resources and reduce the resource allocated to QoS.	the relationship between network security and QoS
Chen et al., 2009	The research shows the impact of security on QoS in communication network.	
Nieto & Lopez, 2014	A context-based parametric relationship model (CPRM) is provide to measure the security and QoS tradeoff in configurable environments.	
Alomari & Menasce, 2012	A single-objective optimization model based on the database platform is designed to optimize network security and QoS.	

have been used to complement the traditional security model in database system. IDPSs have been recently extended to protect database system from malicious intrusions (Darwish, Guirguis, & Ghozlan, 2013; Rao, Singh, Amin, & Sahu, 2014; Saad, Mahdi, & Zbakh, 2012).

However, the database system needs the security service and QoS simultaneously. Both network security and QoS consume computational resource and thus may affect the performance of application services. When high QoS is required, less available resources are provided to network security service. On the other hand, network security services are demanded to reach high level in some application cases, which will consume more resources and may greatly reduce the resource allocated to QoS (Taneja, Raman, & Gupta, 2011). Therefore, some researchers start to study the relationship between security and QoS in recent years (Chen, Hu, Zeng, & Zhang, 2009; Kashif, Madjid, Shi, & Sohail, 2013; Mostafa, Pal, & Hurley, 2014; Nieto & Lopez, 2014). A single-objective optimization model is designed in (Alomari & Menasce, 2012) based on the database platform, which uses intrusion detection system to guarantee the system security. A linear weighted method is used to convert the security and QoS as the global utility, and then a traditional climbing algorithm is performed to find out the combination of IDPSs configuration with maximum global utility. However, this linear relationship between security and QoS is not studied in detail and the users cannot simply select the IDPSs configuration according to either security or QoS. The main contributions of the above mentioned algorithms are clearly listed in Table 1.

Depending on the nature of the applications, their security requirements for the same user may be different. For example, trading online always needs high security requirements while watching video in internet only requires low security configuration. Moreover, different security requirements even for one application may be demanded by different users. For example, when users access a database, high security strength is asked for the user with root privilege while low security strength is provided for the user with visitor privilege. Due to the need of differentiated security, database system has to provide a set of optimal security solutions, which can satisfy the request of different security strength and maintain high QoS. Without any further information, these optimal solutions integrating the requirements of security and QoS are termed Pareto-optimal solutions (Bayon, Grau, Ruiz, & Suarez, 2012; Wang, Li, Yen, & Song, 2014), which indicate that no any other solution is better than them in both QoS and security. By this way, the database system has to find the representatives of Pareto-optimal solutions, which can be served as the available solutions for various requirements. Such that, users can select one Pareto-optimal solution to configure security mechanisms based on their preferences. However, even with the optimal settings of security and QoS in database system, real-time monitoring the required security and QoS parameters is a tremendous pressure for the system administrator who has lots of other work to monitor the performance parameters and run the optimal approach manually in a dynamic environment (Alomari & Menasce, 2012). To solve the aforementioned problem, autonomic system is a promising technique, as it is capable of self-management by self-configuring, self-optimizing, self-protecting and self-healing with feedback loops (Menasce & Kephart, 2007). Inspired by the autonomic computer system (Bennani & Menasce, 2005) designed by queuing networks models (Kleinrock, 1975; Menasce, 2004), it is also able to provide an automatic configuration for both security and QoS in database system.

Therefore, in this paper, an autonomic model for real-time database system is designed, which is aimed at optimizing QoS and security by dynamically changing the security configurations according to the requests from users. It is noted that, although the key indicators of QoS includes delay (the response time), jitter and packet loss rate, this paper mainly considers the relationship of the response time and network security in order to simplify the multi-objective model. The main reason to select the response time as the quantitative evaluation of QoS is mainly based on the facts that, relatively high response time is generally required in some application systems with real-time databases, such as maintenance management and expert systems. Moreover, the extra delays can also be easily captured by the uses and greatly affect the user experience when they use some resource-constrained terminals, such as networking terminals and handheld devices. After that, a classical multi-objective genetic algorithm (NSGA-II) is revised to get the Pareto-optimal sets of our model, as NSGA-II has demonstrated the effectiveness in solving many practical engineering problems (Martins, Carrano, Wanner, Takahashi, & Mateus, 2011; Metaxiotis & Liagkouras, 2012; Rubio-Largo, Vega-Rodriguez, Gomez-Pulido, & Sanchez-Peerez, 2012; Sengupta, Das, Nasir, Vasilakos, & Pedryc, 2012; Shaygan, Alimohammadi, Mansourian, & Govara, 2014). These Pareto-optimal solutions obtained by autonomic controller with NSGA-II can guarantee the security and the delay of the service within an acceptable range. Users can select one of

Pareto-optimal solutions according to their specific requirement or a default value will be automatically assigned based on the different request roles. The default value can refer to the historical data from the application system or the data from the training experiments. To clearly show the contributions of this paper, they are listed as follows:

(1) Different from the single-objective model to summarize the network security and QoS using a linear weighted method, our proposed approach adopts a multi-objective optimization model to simultaneously optimize the network security and QoS. This is the first attempt to build a multi-objective optimization model for optimizing the network security and QoS of databases. More available Pareto-optimal solutions considering the network security and QoS are provided and thus the users can select their preferences on either security or QoS.
(2) A classical multi-objective genetic algorithm (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002) is revised to optimize the proposed multi-objective model and obtain the Pareto-optimal set for the network security and QoS, which performs better and more efficiently than the climbing algorithm in (Alomari & Menasce, 2012). By this way, the database system can quickly respond to the tremendous requests for various users.
(3) By exploiting the intrinsic information from the proposed multi-objective model, a novel crossover approach is designed in the revised NSGA-II, which performs information exchange on each role. The experimental results validate that it is effective to strength the convergence speed and find the optimal solutions more rapidly.

At last, the performance of the revised NSGA-II is evaluated using the proposed multi-objective model to optimize the network security and QoS. Simulation results demonstrate that the revised NSGA-II performs better than the original NSGA-II with one-point and two-point crossover operators, and the obtained optimal set is able to approach the entire Pareto-optimal front under different workloads.

In the remainder of this paper, the multi-objective evaluation model and the quantitative functions for security and QoS are provided in Section 2. In Section 3, the details of controller architecture are introduced. Section 4 presents the details of the revised NSGA-II. At last, Section 5 gives the experimental results and the conclusions are outlined in Section 6.

2. Security and QoS evaluation model

2.1. Application environment

Usually, IDPSs can identify possible incidents, log information about them, attempt to block them and report them. Several response techniques are adopted in IDPSs, which involve attack stopping, firewall reconfiguring and attack content changing. IDPSs allow legal users and behaviors to access the system. Currently, there are two main detection techniques of IDPSs: statistical anomaly-based IDPS and signature-based IDPS. A statistical anomaly-based IDPS determines the normal network activity and alerts user when anomalous traffic is detected, while signature-based IDPS monitors packets in the network with pre-configured and pre-determined attack patterns known as signatures. The disadvantage of the former technique is false positives while the latter one is false negatives. False positives and false negatives have significant impact on the system accuracy, and thus have gained much attention when deploying IDPSs. Security mechanisms in IDPSs are often applied to specific attack types, whereas the applications in database are subject to a variety of attack types. Therefore, it is preferred to use a mix of IDPSs for enhancing the overall security strength. However, little research work focuses on how to raise the overall security strength by using a combination of different security mechanisms. On the other hand, increasing security strength also needs to pay more attention to the impact of IDPS mechanisms on system's QoS requirements, as IDPS mechanisms may evidently decrease QoS performance when they provide very high security service. Especially, when a mix of IDPSs is used that may consume too much computational resources, it induces a negative impact on system's QoS performance.

In our model, a mix of IDPSs is combined to protect the security of network application system, in which an integration of different and diverse mechanisms can provide higher security strength. In the typical setup, IDPSs locate between the web client and the application server, where IDPSs evaluate the requests from users before they are allowed to access the application server. Due to the different rules of IDPSs, the response time of different IDPSs for dealing with the same request is different. Therefore, the overall response time of the system depends on the workload intensity, the combination of the used security mechanisms and the overhead associated with each mechanism.

Data classification is utilized to deal with the huge amount of data in production environment. Due to the large number of users, it is impractical to create an access profile for each user. Therefore, the system would classify the users into roles, and define roles and their expected behaviors. The rule of role classification can be set by system administrator or modified when the system condition is changed. Also, it is required to classify the attack categories. This classification is performed based on the scenario, in which the attacks exist and the user is possible to be affected by such attacks. Then, the effectiveness of security mechanisms from IDPSs is defined for each attack category and the overhead is associated with each mechanism. In our environment, the number of used IDPSs and their detection rate are adopted to represent the effectiveness of mechanisms.

Fig. 1 shows an enterprise application environment for the evaluation model of security and QoS, in which the application server represents servers in a real-time database system. IDPSs are deployed at the outside edge of database system, which inspect all the incoming data from outside of the database system. The incoming requests from the terminals are classified into different roles, such as the system control commands sent by administrators, the analyzing data sent by employees, and the storing data sent by users. The system administrators send the control commands to the system, which require both high security and low delay in order to maintain the system's stability. The employees of enterprises achieve the historical data from real-time database in order to simulate and optimize the system, which pay more attention to security when compared to the time efficiency. The requirements of security and delay can be low when the users upload the files that can be shared with everyone. The controller is placed between the web client and the network application server, which is used to evaluate the requests and send them to IDPSs. Controller is run at a specific time interval in order to obtain the arrival rates of different roles, to identify the possible combination of IDPSs, to compute the security strength and response time for the current configuration using the proposed evaluation model, and to recommend the optimal solution that depends on user preference. After the execution of controller, the security policy will be configured depending on the selected profile.

2.2. Notation and definition

IDPSs are served as security mechanisms in this paper. Let N represent the number of security mechanisms. The number of roles

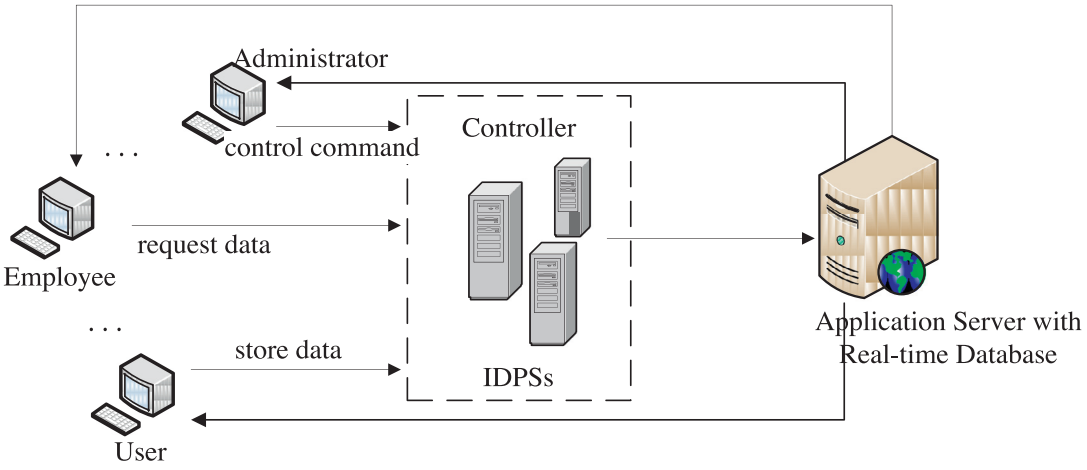

Fig. 1. Enterprise application environment with real-time database.

and the number of attack categories are denoted by R and A. Let $a_{r,j}$ be the attack likelihood of attack j to role r, $d_{i,j}$ be the detection rate of attack j by security mechanism i, and o_i be the overhead for mechanism i.

The controller policy ρ_r represents the assignment of security mechanisms for role r, which is represented by $\rho_r = (\varepsilon_{r,1}, \varepsilon_{r,2}, ..., \varepsilon_{r,N})$. It is noted that $\varepsilon_{r,i}$ set to 1 indicates that security mechanism i is employed for role r and otherwise $\varepsilon_{r,i}$ is set to 0. The overall system policy is then represented by the vector $\rho = (\rho_1, \rho_2, ..., \rho_R)$.

2.3. Quantitative function

The controller optimizes the evaluation functions of security and the response time (Alomari & Menasce, 2012), which express the performance of security strength and QoS. It is preferred to get the optimal results that the response time is as fast as possible while security strength is as strong as possible. Obviously, the above two objectives contradict with each other, as the higher security level will consume more computational resources and resultantly lengthen the response time. Considering the security strength, it is highly depending on the number of used IDPSs and their detection rates. The security strength is increased when more combinations of IDPSs are used to provide security services. The security function should satisfy the conditions that the overall security strength is at least greater than the maximum value of the used detection rate in the policy and it can be increased when additional mechanisms are further adopted. Thus, exponential averaging is employed here to compute the security strength for obtaining a more stable model. Therefore, the security strength for role r can be computed by

$$U_r^s(\overrightarrow{\rho_r}) = \sum_{j=1}^{A} a_{r,j}(\ln \sum_{i=1}^{N} e^{d_{i,j}\times 10} \times \varepsilon_{r,i})/10 \quad (1)$$

The total security utility can be represented as the weighted sum of all roles, as follows.

$$U_{total}^s(\overrightarrow{\rho}) = \sum_{\forall r} \varpi_r^s U_r^S(\overrightarrow{\rho_r}) \quad (2)$$

where ϖ_r^s is a weight factor of role r.

The average response time of IDPSs system for role r consists of the response time of IDPSs with T_{idps} and network applications with T_{nas}, which is obtained by

$$T_r = T_{idps} + T_{nas} \quad (3)$$

The total response time is the weighted sum of the response time utility for all roles and can be expressed by

$$T_{total} = \sum_{\forall r} \varpi_r^t T_r \quad (4)$$

where ϖ_r^t is a weight factor of role r.

The queuing networks (QN) model (Menasce, 2004) is adopted here to compute the QoS value for the policy. The requests from users are evaluated by the controller and then sent to IDPSs and the network application services. The IDPS mechanisms can work concurrently. However, the network application services must wait for the finish of IDPSs before it can work. The fork and join (Alomari & Menasce, 2013b) queues under open queuing networks model are adopted as the security mechanisms model.

Fig. 2 shows a network application system with fork and join QN model. The IDPSs are modeled as fork and join sub-networks, and the application server model is used with the performance model (Menasce, 2004). Requests from users are replicated and serviced in parallel on different queues, in which different service times are needed for different queues. The execution of the request from role consists of the controller evaluation and IDPSs execution that is driven by policy and network application services.

Suppose that the network application system in Fig. 2 is composed by N devices. The service demand law (Menasce, 2004) states that the utilization of device j by role r is

$$U_{j,r} = \lambda_r \times o_{j,r} \quad (5)$$

where λ_r is the arrival rate for role r and $o_{j,r}$ is the overhead of requests of role r at device j. The total utilization of device j is

$$U_j = \sum_{r=1}^{R} U_{j,r} \quad (6)$$

where R is the number of devices. Since the maximum value of total utilization is 100%, the value of U_j must be not larger than 1. Then, according to the arrival theorem and Little's Law (Menasce, 2004), the average response time of network application system for the request of role r can be represented as

$$T_{nas} = \sum_{j=1}^{N} \frac{o_{j,r}}{1 - U_j} \quad (7)$$

An approximation technique (Alomari & Menasce, 2013a) is adopted here to compute the average response time for fork and join queues with heterogeneous parallel servers in open queuing networks. Before performing the approximation, the security mechanisms are sorted by the rule:

$$o_{1,r} \times \varepsilon_{1,r} \geq o_{2,r} \times \varepsilon_{2,r} \geq ... \geq o_{i,r} \times \varepsilon_{i,r} \quad (8)$$

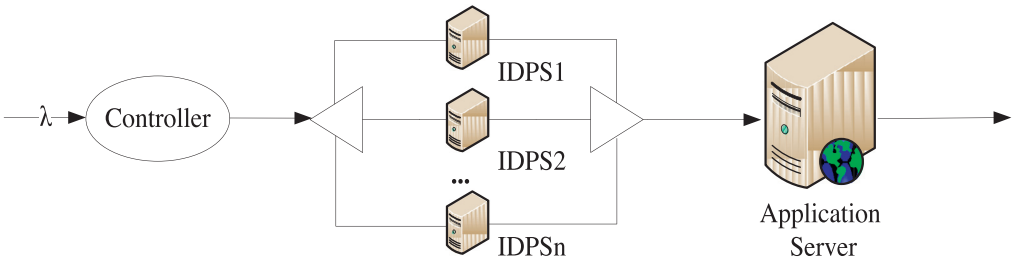

Fig. 2. Fork & Join queues QN model.

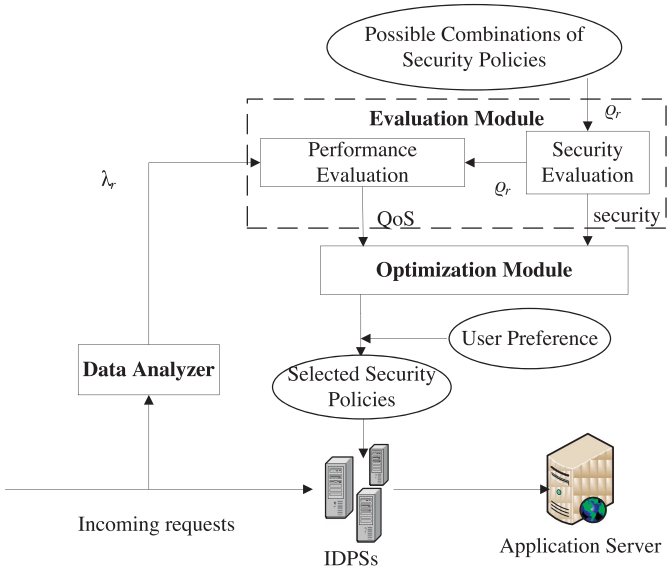

Fig. 3. Controller architecture.

where $o_{i,r}$ is the service demand of requests of role r at mechanism i, and $\varepsilon_{i,r}$ is set to 0 when security mechanism i is not used for role r. Otherwise, $\varepsilon_{i,r}$ is set to 1. Then, the average response time for role r with N heterogeneous parallel servers is

$$T_{idps} = \sum_{i=1}^{N} \frac{1}{i} \times \frac{o_{i,r}}{1 - U_i} \tag{9}$$

where U_i is the utilization of mechanism i by the requests of all roles. The total utilization of mechanisms is computed as

$$U_i = \sum_{r=1}^{R} \lambda_r \times o_{i,r} \times \varepsilon_{r,i} \tag{10}$$

3. Controller architecture

The controller searches the combinations of security policies to obtain the optimal solutions that simultaneously satisfy the security and QoS requirements from users. The complexity of the optimization problem depends on the space of all combinations of security mechanisms. For each role, different combinations of security mechanisms can be used. Therefore, the searching space is equal to 2^{R*N}, where R is the total number of roles and N is the total number of security mechanisms. The autonomic controller dynamically changes the policies according to the variety of requests from users. Fig. 3 shows the architecture of controller system that subjects to IDPSs. The controller is composed by three main parts: data analyzer, evaluation module and optimization module. The autonomic controller is run at regular interval, which is called control interval to optimize the results of security and response time functions.

The data analyzer obtains the workload intensity (rate of all role requests) λ_r of current system in control interval. Request rate λ_r of role r and security policy ρ from possible combinations of security policies are regarded as the input parameters for the evaluation module. The performance evaluation module uses the QN model to evaluate the response time according to the current policy. The output of performance evaluation module (response time) and security evaluation module (security strength) are treated as the input parameters for optimization module.

Due to the tremendous combinations of security mechanisms, it is impractical to find all optimal configurations with exhaustive search. Therefore, a state-of-the-art multi-objective genetic algorithm (NSGA-II) is revised in order to find a set of approximated optimal configurations, as it has been widely used to solve many practical engineering problems (Huang, Buckley, & Kechadi, 2010; Ronay et al., 2013). The advantages of NSGA-II include a fast non-dominated sorting (Deb et al., 2002) that has less computational complexity, and elitism mechanism to prevent the loss of good solutions found in the evolutionary search. Thus, NSGA-II is revised in order to find the Pareto-optimal set of our proposed model that can well balance the security and QoS requirements. The detailed description of the revised NSGA-II is given in Section 4. After that, according to the user preference on the security requirement or the delay requirement, the controller selects one configuration from the Pareto-optimal set and then drives IDPSs to reconfigure the policies. Once the requests are detected, they will be sent to the application servers.

4. The revised NSGA-II algorithm

4.1. Optimal evaluation

To find out the Pareto-optimal solutions for security and QoS, the controller performs the revised NSGA-II at the beginning of each interval. At first, Eqs. (2) and (4) are used to compute the total estimated values of security and QoS, which are regarded as the two objective values in NSGA-II. The systemic parameters of NSGA-II, such as the population size, the generation times, the probabilities of crossover and mutation, are all pre-defined by system administrator. After the execution of revised NSGA-II, an approximated Pareto-optimal set is obtained with N solutions, in which all the solutions are optimal when considering both of security and QoS. Controller can select one from the approximated Pareto-optimal set depending on the user's security or QoS preference. To better understand the multi-objective optimization model, the definition of Pareto-optimality is given (Lin et al., 2016), where $f_1(x)$ and $f_2(x)$ are the objective values of security and QoS.

Definition 1 (Pareto-dominance)**.** A decision variable vector x is said to dominate another decision variable vector y (noted as $x \succ y$) if and only if

$$(\forall i \in \{1,2\} : f_i(x) \le f_i(y)) \wedge (\exists j \in \{1,2\} : f_j(x) < f_j(y)) \tag{11}$$

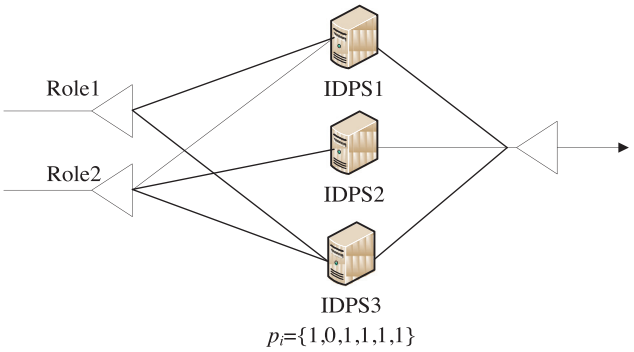

Fig. 4. Illustration of an individual representation.

Definition 2 (Pareto-optimal)**.** A solution x is said to be Pareto-optimal if and only if

$$\neg\exists y \in \Omega : y \succ x \tag{12}$$

Definition 3 (Pareto-optimal set)**.** The set PS includes all the Pareto-optimal solutions, as defined by

$$PS = \{x | \neg\exists y \in \Omega : y \succ x\} \tag{13}$$

Definition 4 (Pareto-optimal front)**.** The set PF includes the value of all the objective functions corresponding to the Pareto-optimal solutions in PS.

$$PF = \{F(x) = (f_1(x), f_2(x))^T | x \in PS\} \tag{14}$$

In order to tackle the optimization problem driven from the evaluation model, the individual (solution) used in NSGA-II is represented by the binary codes, indicating that which security mechanisms are adopted in IDPSs. It can be represented by $p_i = \{x_{1,1}, x_{1,2}, ..., x_{1,m}, ..., x_{r,1}, x_{r,2}, ..., x_{r,m}\}$, where r is the total number of roles and m is the total number of security mechanisms. Each dimension of individual $x_{i,j}$ is an integer that is set to 0 or 1, in which $x_{i,j} = 1$ means the role i uses security mechanism j while $x_{i,j} = 0$ indicates the role i doesn't run security mechanism j. Fig. 4 gives an example about the representation of an individual, where two roles are detected by three IDPSs. $p_i = \{1, 0, 1, 1, 1, 1\}$ indicates that the first role uses the first and the third IDPS, while the second role uses all the three IDPSs. A set of individuals compose the population, which is represented by $P = \{p_1, p_2, ..., p_N\}$ (N is the number of individuals).

4.2. The details of revised NSGA-II

The details of revised NSGA-II are presented as follows. Due to the difficulty of the optimization problem in our model, the classical NSGA-II is accordingly revised in order to better solve it. Fig. 5 shows the flowchart of revised NSGA-II, which includes the main components such as initialization, selection, crossover, mutation and archive. After the termination condition (i.e., the maximal generations MAX_Gen) is satisfied, the final population is exported as the approximated Pareto-optimal set. In the following subsections, the main components of our revised NSGA-II are respectively introduced.

4.2.1. Initialization

In this component, an initial population $P = \{p_1, p_2, ..., p_N\}$ with size N is randomly created and then the fitness values for each individual p_i ($i = 1, 2, ..., N$) including the security strength and response time are calculated. After that, the non-domination sorting approach (Deb et al., 2002) is performed on population P to figure out all the non-dominated fronts, where the first front means the non-dominated solutions in P while the second front is found to be non-dominated when deleting the solutions of the first front from P. By this way, the subsequent fronts can be found until all the solutions in P are marked with a front number. In our model, if the security strength of p_i is higher than that of p_j while the response time of p_i is less than that of p_j, it is said that p_i dominates p_j ($i, j \in [1, N]$). For each individual in population P, the smaller front number indicates the better non-domination relationship. To clearly show the initialization procedure, the pseudo-code of **initialization** is given in Fig. 6.

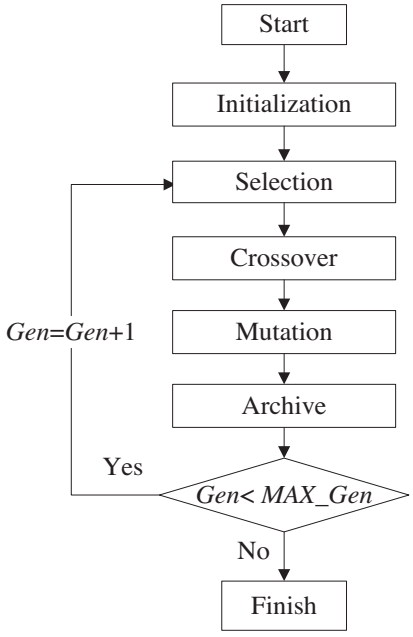

Fig. 5. The flowchart of revised NSGA-II.

4.2.2. Selection

In this procedure, N_s individuals are selected to compose an evolutionary population Q. Two individuals are randomly picked up from the parent population P, and then the one with smaller front number enters into Q as it has the better convergence in domination sense. If they have the same front number, one individual will be randomly selected. This step continues until the number of individuals in Q is equal to N_s. The pseudo-code of **selection** is described in Fig. 7.

4.2.3. Crossover and mutation

Traditionally, in binary code representation, one-point crossover and two-point crossover operators are two mostly used crossover operators in genetic algorithms. Assume that the two selected parents for crossover operator are represented by $p^1 = \{p^1_{1,1}, p^1_{1,2}, ..., p^1_{1,m}, ..., p^1_{r,1}, p^1_{r,2}, ..., p^1_{r,m}\}$ and $p^2 = \{p^2_{1,1}, p^2_{1,2}, ..., p^2_{1,m}, ..., p^2_{r,1}, p^2_{r,2}, ..., p^2_{r,m}\}$. A crossover probability p_c in [0, 1] decides the execution of crossover operator on p^1 and p^2. When performing the one-point crossover operator, an integer η is generated randomly in $[1, (r \times m)]$, where r is the total number of roles and m is the total number of security mechanisms. Then, for the role i with j security mechanisms ($i \in [1, r]$and $j \in [1, m]$) in p^1 and p^2, if the value of $(i \times j)$ is greater than η, the values of $p^1_{i,j}$ and $p^2_{i,j}$ will be swapped and resultantly two new child individuals q^1 and q^2 are obtained. An example with $r = 3, m = 3, \eta = 4$ for operating one-point crossover

Algorithm 1: Initialization

```
1  for i = 1 to N
2      generate an individual p_i randomly
3      evaluate the fitness values of p_i
4      add p_i to the population P
5  end for
6  sort P with the non-domination sorting approach
```

Fig. 6. The pseudo-code of initialization.

Algorithm 2: Selection

```
1  for i = 1 to N_s
2      picked up two individuals from population P randomly
3      if their front numbers are the same
4          add one of them to the population Q randomly
5      else
6          add the one with smaller front number to the population Q
7      end if
8  end for
```

Fig. 7. The pseudo-code of selection.

$$p^1=(0,1,0,1,\mathbf{1,1,0,1,0}) \quad q^1=(0,1,0,1,\mathbf{0,1,1,0,0})$$
$$p^2=(1,1,0,1,\mathbf{0,1,1,0,0}) \xrightarrow[\eta=4]{\text{Crossover}} q^2=(1,1,0,1,\mathbf{1,1,0,1,0})$$

Fig. 8. Illustration of one-point crossover.

$$p^1=(0,1,0,\mathbf{1,1,1},0,1,0) \quad q^1=(0,1,0,\mathbf{1,0,1},0,1,0)$$
$$p^2=(1,1,0,\mathbf{1,0,1},1,0,0) \xrightarrow[\eta_1=6,\eta_2=4]{\text{Crossover}} q^2=(1,1,0,\mathbf{1,1,1},1,0,0)$$

Fig. 9. Illustration of two-point crossover.

is illustrated in Fig. 8, where the genes marked with bold font are exchanged.

When executing two-point crossover operator, two distinct random integers η_1 and η_2 are produced in [1, $(r \times m)$] and let $\eta_1 > \eta_2$ by swapping their values. Then, for the role i with j security mechanisms ($i \in [1, r]$and $j \in [1, m]$) in p^1 and p^2, if the value of $(i \times j)$ is not greater than η_1 and not smaller than η_2, the values of $p^1_{i,j}$ and $p^2_{i,j}$ will be swapped and resultantly two new child individuals q^1 and q^2 are obtained. An example with $r = 3, m = 3, \eta_1 = 6, \eta_2 = 4$ for operating two-point crossover is illustrated in Fig. 9, in which the genes identified with bold font are switched.

However, the traditional one-point and two-point crossover operators are not well suitable for solving our proposed model as they don't fully exploit the prior knowledge in our model. As introduced above, one individual is built by mapping one combination of security mechanisms. Thus, the combinations of security mechanisms for each request role are independent. A role-based crossover operator is designed to exchange the combination of security mechanism according to the request roles. When executing our proposed crossover operator, for each role r, an integer η_r is sampled randomly in [1, m], which indicates that the role r is selected to exchange beginning from its η_r security mechanism. Therefore, for the role r with j security mechanisms ($j \in [1, m]$) in p^1 and p^2, if the value of j is not less than η_r, the values of $p^1_{r,j}$ and $p^2_{r,j}$ will be exchanged and resultantly two new child individuals q^1 and q^2 are obtained. An example with $r = 3, m = 3, \eta_1 = 2, \eta_2 = 1, \eta_3 = 3$ for running our role-based crossover is illustrated in Fig. 10, in which the genes identified with bold font are swapped.

After that, the bitwise mutation is adopted in **mutation**, where a mutation probability p_m controls the execution of mutation. For each child $q = \{q_{1,1}, q_{1,2}, ..., q_{1,m}, ..., q_{r,1}, q_{r,2}, ..., q_{r,m}\}$ produced by crossover operator, a random integer η is generated in [1, $(r \times m)$], and then the ηth variable of q will be changed. For example, if its value is 1, it will be changed to 0 after mutation.

4.2.4. Archive

The archive procedure determines the evolved population for the next generation, which preserves the individuals with smaller front numbers as they usually have better convergence. In **archive**, the original population P, the offspring popula-

$$p^1 = (0,\mathbf{1},\mathbf{0},\mathbf{1},\mathbf{1},\mathbf{1},0,1,\mathbf{0}) \qquad q^1 = (0,\mathbf{1},\mathbf{0},\mathbf{1},\mathbf{0},\mathbf{1},0,1,\mathbf{0})$$
$$p^2 = (1,\mathbf{1},\mathbf{0},\mathbf{1},\mathbf{0},\mathbf{1},1,0,\mathbf{0}) \xrightarrow[\eta_1 = 2,\eta_2 = 1,\eta_3 = 3]{\text{Crossover}} q^2 = (1,\mathbf{1},\mathbf{0},\mathbf{1},\mathbf{1},\mathbf{1},1,0,\mathbf{0})$$

Fig. 10. Illustration of role-based crossover.

Algorithm 5: the revised NSGA-II

Initialization (Algorithm 1);
Gen =1;
if *Gen* < *MAX_Gen*
  Selection (Algorithm 2);
  i=1;
  while $i \le N_s$
    role-based Crossover is performed on p_i and p_{i+1} to get q_i and q_{i+1};
    mutation is executed on q_i and q_{i+1} to get q_i' and q_{i+1}';
    add child q_i' and q_{i+1}' to the population *Q'*;
    i=*i*+2;
  end while
  Archive(Algorithm 4)
  Gen = *Gen*+1
end if
sort *P* with the non-domination sorting approach
output non-dominated solutions in population *P*

Fig. 11. The pseudo-code of the revised NSGA-II.

tion *Q'* that is produced by crossover and mutation and the random population *Q''*, are combined to form a union population *F*. A random population *Q''* has *M* diverse individuals randomly generated. Each diverse individual as represented by $d = \{d_{1,1}, d_{1,2}, ..., d_{1,m}, ..., d_{r,1}, d_{r,2}, ..., d_{r,m}\}$ (*r* is the total number of roles and *m* is the total number of security mechanisms) is randomly generated as follows.

$$d_{i,j} = \begin{cases} 0 & \text{if } rand < \mu \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

where μ is a predefined real value in [0, 1] and *rand* is a uniformly random real number in [0, 1]. For each individual of population *F*, their objective values in QoS and security are calculated and then the fast non-dominated sorting approach is performed on population *F* to get the front numbers for each solution. Then, *P* is set as an empty set, and the individuals with the first front number are selected to the next generation population *P*. If the size of *P* is less than *N*, the individuals with second front number will be chosen. By this way, the individuals with smaller front numbers will be gradually selected into *P* until the size of *P* is equal to *N*. It is noted that only parts of individuals in the last front will be chosen in order to let *P* exactly have *N* individuals.

4.2.5. The complete algorithm

The above subsections have introduced the procedures of initialization, selection, crossover, mutation and archive, which compose the main components of revised NSGA-II. The pseudo-code of the complete algorithm is described in Fig. 11. After **initialization**, the original population *P* is created and the generation time *Gen* is set to 1. Then, the loop of evolutionary progress is repeated until the generation time *Gen* reaches the maximum times *MAX_Gen*. During the evolutionary progress, **selection** as described in **Algorithm 1** is performed first and population *Q* is obtained. From the beginning of the first individuals in population *Q*, two neighboring individuals in population *Q* are performed by **crossover**, and are replaced by their two child individuals. Then, these new child individuals in population *Q* execute **mutation** and form the child population *Q'* . Finally, **archive** is performed and the new evolved population *P* is obtained. The generation time *Gen* is increased by 1 and the above evolutionary loop will be terminated when *Gen* reaches *MAX_Gen*. At the end of algorithm, the non-dominated solutions in the external archive are exported as the final result.

5. Experimental results

5.1. Experimental setting

In this section, the experiments are conducted to verify the validity of the controller system. The proposed NSGA-II algorithm and the greedy hill climbing algorithm are implemented using C++ language in VC++ 6.0 platform. Their source codes are all run on an Inter(R) Core(TM) i5-3230 M CPU machine, 2.6 GHz, 4GB memory with Windows 7 operating system. In our experiments, we run the optimal approach with two size problems: one with 3 different request roles and 8 different IDPSs, and the other with 3 different request roles and 10 different IDPSs. Detection rates of IDPS and the likelihood of attack for roles can be estimated using the suitable data and historical information. For NSGA-II, the crossover probability is 0.9 while the mutation probability is 1/*n*,

Table 2
Role parameter.

Parameter	Role 1	Role 2	Role 3
ϖ_r^s	0.4	0.4	0.2
ϖ_r^t	0.3	0.4	0.3

where n is the number of the binary string length. The evolution population size N is 100, the size M of random population Q' ' is 20, the archive size N_s is 80, and the maximal generation time *MAX_Gen* is 500. The weight values for each role are shown in Table 2 which can be set by administrator depending on actual conditions. The final solution results are obtained by picking out all the non-dominated solutions.

5.2. Performance metric

To evaluate the performance of NSGA-II algorithm, the inverted generational distance (*IGD*) (Li & Zhang, 2009; Zhu et al., 2016) is adopted. Assume that the approximated set obtained by the revised NSGA-II algorithm is S, while the true Pareto-optimal set is P, which is found by the exhaustive searching approach. For example, in Section 5.4, it takes almost 3 h to obtain the exact values of P using the exhaustive search. The *IGD* result can be obtained, as calculated by

$$IGD(S,P) = \frac{\sum_{x\in P} dis(x,S)}{|P|} \tag{16}$$

where $dis(x, S)$ represents the nearest Euclidean distance from solution x in P to the solutions of S, while $|P|$ indicates the number of solutions in P. Assuming that $m=(m_1, m_2,..., m_i)$ and $n=(n_1, n_2,..., n_i)$ are two points in the i-dimensional space, the Euclidean distance $dis(m,n)$ between m to n is given as follows.

$$dis(m,n) = \sqrt{(m_1-n_1)^2 + (m_2-n_2)^2 + ... + (m_i-n_i)^2} \tag{17}$$

In our paper, the security and response time of a solution are represented by a two-dimensional point.

Generally, a lower value of the *IGD* metric indicates that the obtained set S is closer to the true Pareto-optimal set P and more uniformly distributed along the true Pareto-optimal front.

5.3. The advantages of multi-objective model

The revised NSGA-II algorithm is run at 3 different request rates. These three different request rates $\lambda_1=\{6,8,11\}$, $\lambda_2=\{12,15,13\}$ and $\lambda_3=\{6,5,4\}$ respectively represent the general, high and low workloads. The three values in request rate sets λ represent the request rates of three different roles. Therefore, the experimental studies can effectively investigate the practical performance of NSGA-II algorithm in different request rates.

In (Alomari & Menasce, 2012), the network security and QoS are aggregated into a global utility of system using a linear weighting method and then a classical greedy hill climbing algorithm is adopted to search the optimal solution with the maximum global utility. However, this approach can't give a set of Pareto-optimal solutions and it is difficult to select the weight vectors to determine the relative importance of security and QoS. Our proposed multi-objective model can well solve the above problems and the uses may determine their preferences either on security or QoS. To illustrate the advantage of our multi-objective model, the proposed approach in (Alomari, F. & Menasce, 2012) is operated by 50,000 times at three different request rates sets λ_1, λ_2 and λ_3. Tables 3 and 4 give the optimal solutions obtained by the single-objective model using different weight value sets with request rates sets λ_1, and λ_3 at different problem size, where different weight vectors are used to determine the importance of security and QoS, such as $\varpi_1=\{0.2,0.8\}$, $\varpi_2=\{0.3,0.7\}$, $\varpi_3=\{0.4,0.6\}$, $\varpi_4=\{0.5,0.5\}$, $\varpi_5=\{0.6,0.4\}$, $\varpi_6=\{0.7,0.3\}$ and $\varpi_7=\{0.8,0.2\}$. As observed from Tables 3 and 4, the single-objective model can only obtain an optimal solution under different weight vectors. Generally, with the importance of security in weight vector is increased, the optimal value of security strength is also enhanced. Another shortcoming is that the single-objective model may obtain the same optimal result even using different weight vectors, such as (ϖ_3, ϖ_4) and (ϖ_5, ϖ_6, ϖ_7) in Table 3, (ϖ_1, ϖ_2) and (ϖ_3, ϖ_4) in Table 4. Moreover, the single-objective model cannot solve the problem at request rates sets λ_3 as it is easy to find a solution that makes the utilization U_i of $IDPS_i$ larger than 1 at the high workload. However, this case is not applicable in practical use.

Fig. 12(a) and (b) illustrate the comparison results between our multi-objective model using the revised NSGA-II that use role-based crossover operator and the single-objective model (Alomari & Menasce, 2012) using the hill climbing algorithm (Hill climbing) with 8 IDPSs and the request rates sets λ_1 and λ_3. As observed in Fig. 12(a) and (b), the hill climbing algorithm only finds several optimal solutions, while our revised NSGA-II can explore much more available optimal solutions. The solutions obtained by single-objective model at different weight value sets are the same with or dominated by that found in our model. Fig. 12(c) and (d) further illustrate the comparison results with 10 IDPSs and the request rates sets λ_1 and λ_3. These results are similar with that in Fig. 12(a) and (b). These experimental results validate that our multi-objective model is more effective than the single-objective model.

Table 5 shows the execution times for the compared algorithms under 3 request rates sets λ_1, λ_2 and λ_3, which are the average values of 30 execution times. About 1.4 s is required to run the NSGA-II algorithms, while the hill climbing search seems much faster. However, the hill climbing algorithm can only find one sub-optimal solution; whereas, the NSGA-II algorithms can find more than 60 combinations of sub-optimal solutions, which can meet the different requirements of the users regarding the security and QoS. Generally, the workload of system or the roles under user requests do not change a lot in a short time, so that the proposed optimization approach only runs periodically at a regular time interval. The execution times of our NSGA-II will not present negative effect to the system.

5.4. The comparison of our revised NSGA-II with original NSGA-II

To study the enhancement of role-based crossover operator, our revised NSGA-II is further compared with the original NSGA-II algorithm with one-point and two-point crossover approaches under the same parameter settings. All the compared algorithms are run by 30 times with 8 IDPSs and 10 IDPSs at three different request rates sets λ_1, λ_2 and λ_3. Fig. 13 shows the evolutionary curves of *IGD* with 8 IDPSs and three different request rates sets λ_1, λ_2 and λ_3 using the three NSGA-II algorithms, in which NSGA-II, NSGA-1 and NSGA-2 respectively represent the revised NSGA-II with role-based crossover operator, the original NSGA-II with one-point crossover and two-point crossover operators. In all cases of workload, it is clearly observed that the *IGD* results of our revised NSGA-II reduce faster and the trend of evolution is smoother, which indicate that our revised NSGA-II has faster convergence speed than the original one. Especially, in low or general workloads, the advantage of our approach is more evident.

Fig. 14 illustrates the Pareto-optimal sets obtained by the three compared algorithms with 8IDPSs and three different request rates sets λ_1, λ_2 and λ_3, where NSGA-II, NSGA-1 and NSGA-2 respectively represent the revised NSGA-II with role-based crossover operator, the original NSGA-II with one-point crossover

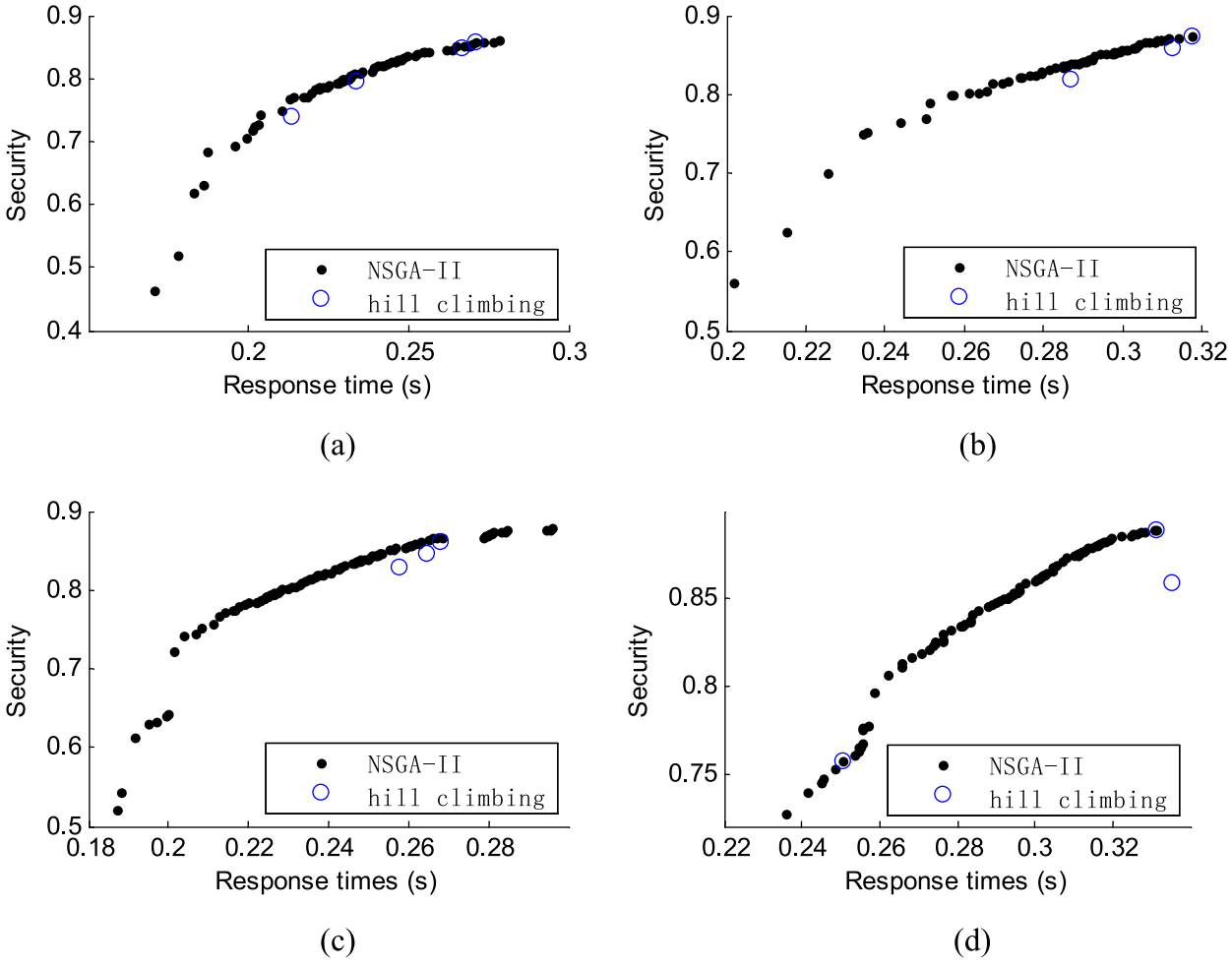

Fig. 12. Comparison results with 8 IDPSs at (a) λ_1 (b) λ_3 and with 10 IDPSs at (c) λ_1 (d) λ_3.

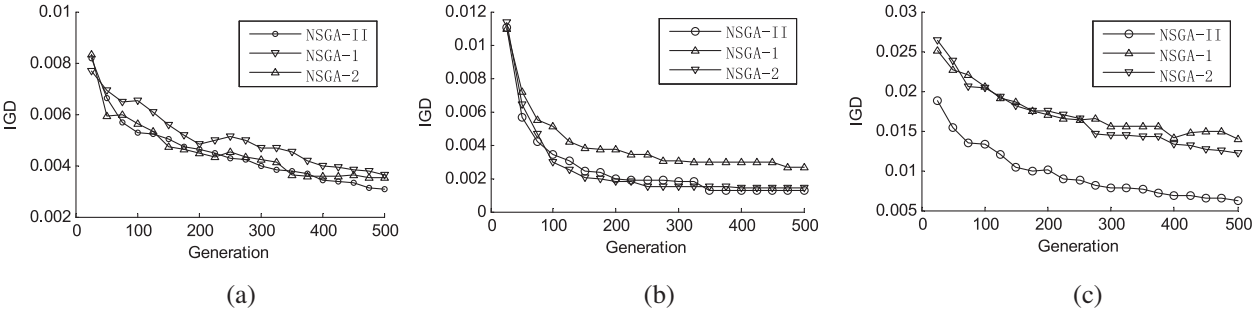

Fig. 13. Evolutionary curves with 8 IDPSs and (a) λ_1, (b) λ_2 and (c) λ_3.

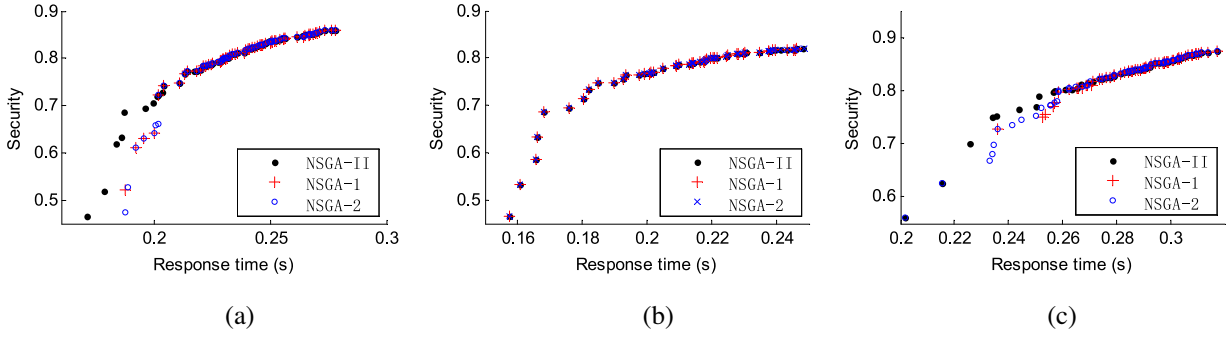

Fig. 14. Final Results with 8 IDPSs and (a) λ_1, (b) λ_2 and (c) λ_3.

Table 3
Results with different weight value sets for 3 roles and 8 IDPSs.

	ϖ_1		ϖ_2		ϖ_3		ϖ_4	
	security	delay	security	delay	security	delay	security	delay
λ_1	0.739041	0.213123	0.797296	0.233339	0.848868	0.266636	0.848868	0.266636
λ_3	0.819526	0.286701	0.860096	0.312148	0.860096	0.312148	0.860096	0.312148
	ϖ_5		ϖ_6		ϖ_7			
	security	delay	security	delay	security	delay		
λ_1	0.857626	0.271037	0.857626	0.271037	0.857626	0.271037		
λ_3	0.860096	0.312148	0.873403	0.31732	0.873403	0.31732		

Table 4
Results with different weight value sets for 3 roles and 10 IDPSs.

	ϖ_1		ϖ_2		ϖ_3		ϖ_4	
	security	delay	security	delay	security	delay	security	delay
λ_1	0.829517	0.257505	0.829517	0.257505	0.847137	0.264368	0.847137	0.264368
λ_3	0.757392	0.250306	0.757392	0.250306	0.757392	0.250306	0.757392	0.250306
	ϖ_5		ϖ_6		ϖ_7			
	security	delay	security	delay	security	delay		
λ_1	0.862688	0.26812	0.862688	0.26812	0.862688	0.26812		
λ_3	0.859674	0.334973	0.889605	0.33126	0.889605	0.33126		

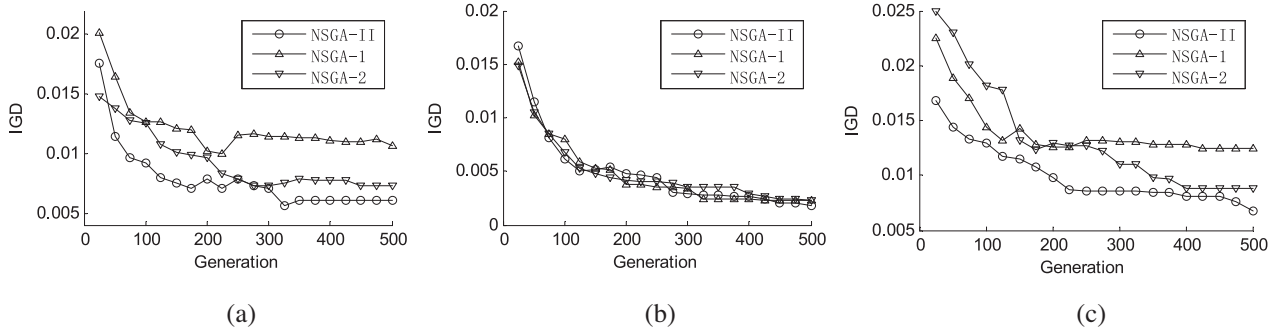

Fig. 15. Evolutionary curves with 10 IDPSs and (a) λ_1, (b) λ_2 and (c) λ_3.

Table 5
The execution times for different algorithms (seconds).

	λ_1	λ_2	λ_3
NSGA-II	1.42	1.312	1.504
Hill climbing	0.073	0.065	0.075

Table 6
The p-values of Wilcoxon's rank sum test for NSGA-1, NSGA-2 vs NSGA-II.

	NSGA-1 vs NSGA-II	NSGA-2 vs NSGA-II
8 IDPSs with λ_1	9.94E-03	5.79E-04
8 IDPSs with λ_2	2.78E-11	8.40E-03
8 IDPSs with λ_3	4.42E-06	3.26E-07
10 IDPSs with λ_1	3.50E-06	8.40E-03
10 IDPSs with λ_2	8.82E-01	7.35E-01
10 IDPSs with λ_3	1.44E-04	2.80E-03

and two-point crossover operators. Based on the observation from Fig. 14(a)–(c), the solutions obtained by our NSGA-II almost dominate those achieved by the original NGSA-II with two traditional crossover operators. At low or general workload, our approach can find the better solutions in low security scenario; whereas, at high workload, our approach illustrates the similar performance with NSGA-1 and NSGA-2. This is reasonable as it is easy to find the Pareto-optimal solutions at high workload as the number of available combinations of IDPSs is greatly reduced. Therefore, NSGA-1 and NSGA-2 perform very similarly with our approach at high workload. Moreover, with the increase of total requests, the *IGD* results will be enlarged, which means it is more difficult to find all the true Pareto-optimal solutions and the total response time is also lengthened.

Regarding the experiments with 10 IDPSs, Fig. 15 shows the evolutionary curves of *IGD* with 10 IDPSs and three different request rates sets λ_1, λ_2 and λ_3 using the three NSGA-II algorithms. Similar to the *IGD* results with 8 IDPSs, our revised NSGA-II also performs significantly better, especially in low workloads. Fig. 16 also shows the Pareto-optimal sets obtained by NSGA-II, NSGA-1 and NSGA-2 with 10 IDPSs and three different request rates sets λ_1, λ_2 and λ_3. Also, at low or general workload, our approach can find the better solutions in low security scenario; whereas, at high workload, our approach illustrates the similar performance with NSGA-1 and NSGA-2.

We run the Wilcoxon's rank sum test for the sets of *IGD* obtained by NSGA-II, NSGA-1 and NSGA-2, at two IDPSs sizes and three request rates sets λ_1, λ_2 and λ_3. The size of each set of *IGD* is 30. Generally, when the p-value of Wilcoxon's rank sum test for two sets of data is less than 0.05, the two data sets are not statistically similar. All the p-values are collected in Table 6. Except 10 IDPSs and request rates sets λ_2, other p-values from the Wilcoxon's rank sum test are far less than 0.05. This also can be observed from Fig. 16(b) that the *IGD* results for all three algorithms are very similar at 10 IDPSs and request rates sets λ_2. This is reasonable as it is easy to find the Pareto-optimal solutions at high workload.

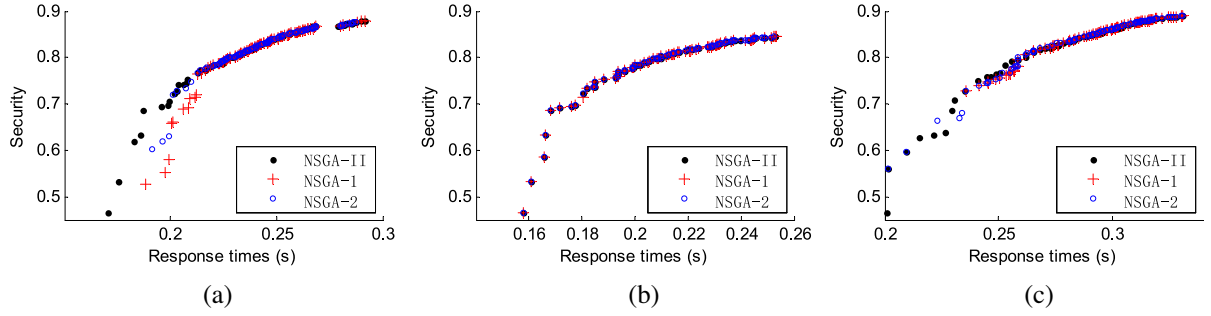

Fig. 16. Final Results with 10 IDPSs and (a) λ_1, (b) λ_2 and (c) λ_3.

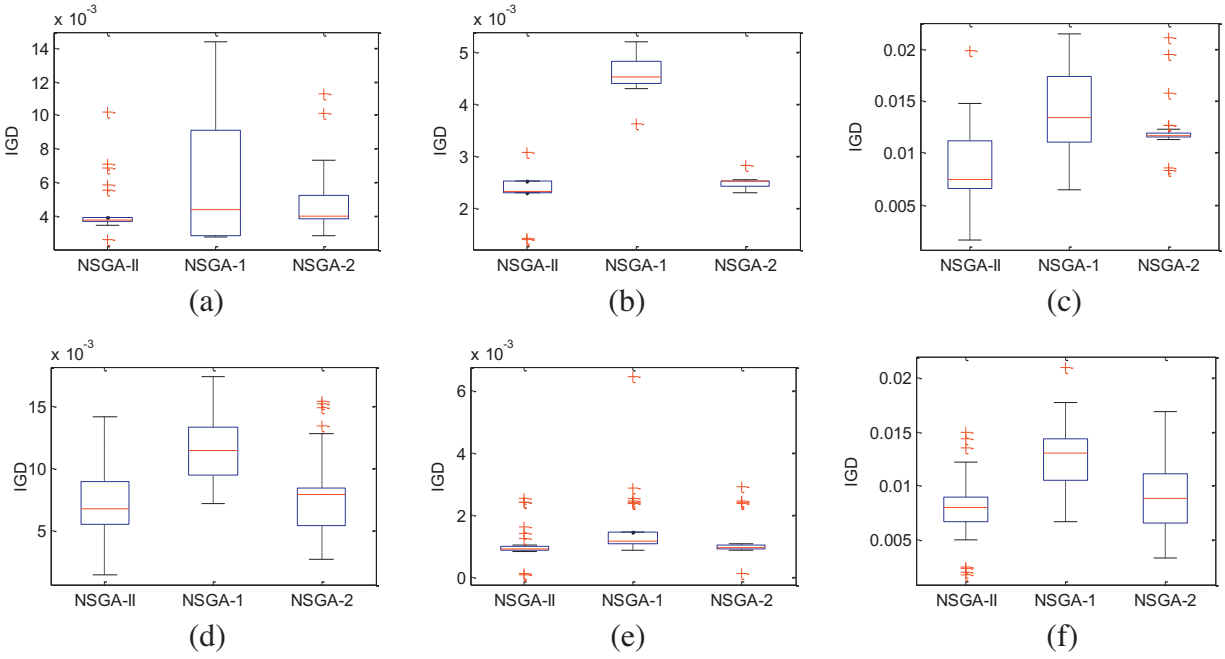

Fig. 17. The boxplots of *IGD* sets of three compared algorithms with 8 IDPSs at (a) λ_1, (b) λ_2, (c) λ_3 and 10 IDPSs at (a) λ_1, (b) λ_2, (c) λ_3.

Fig. 17(a)–(f) further illustrate the boxplots of *IGD* sets obtained by the three compared algorithms at three different request rates sets and two IDPSs sizes. The size of each set of *IGD* is 30. As we can see, at all cases of workload and IDPSs size, the median values of our NSGA-II in boxplot are less than that of NSGA-1 and NSGA-2. Especially, in low workloads, the advantage of our approach is more evident.

6 Conclusions

In this paper, an autonomic multi-objective optimization model is proposed, which can integrate both security and QoS under the available computational resource. In order to optimize the proposed multi-objective model, a multi-objective genetic algorithm NSGA-II is revised to obtain the Pareto-optimal solutions, in which a novel role-based crossover approach is designed. As the security requirements for different roles are independent, the proposed role-based crossover approach also performs information exchange for each role independently. Simulation results demonstrate that the revised NSGA-II performs better than the original NSGA-II with traditional crossover operators, and the obtained optimal set is able to approach the entire Pareto-optimal front under different workloads. These obtained Pareto-optimal security policies solutions not only can meet the different security requirement of the user, but also provide the optimal QoS.

In our future work, other objectives will be further considered in our multi-objective optimization model. For example, in cloud-computing environment, where the computational resources can be easily increased under demand, it still needs to balance the security and QoS as the users need to pay for the extra computational resources. The accounting for the computational resources can be one more potential objective in this application scenario. Moreover, other superior nature-inspired optimization algorithms are also investigated to optimize our proposed model.

Acknowledgements

This work was supported by the National Nature Science Foundation of China under Grants 61402291, Technology Planning of Guangdong under Grant 2014B010118005, Seed Funding from Scientific and Technical Innovation Council of Shenzhen Government under Grant 0000012528, Foundation for Distinguished Young Talents in Higher Education of Guangdong under Grant 2014KQNCX129, Natural Science Foundation of SZU under Grant 201531, and in part by the Science and Technology Projects of Shenzhen under Grant JCYJ20140418095735608.

Reference

Alomari, F., & Menasce, D. (2012). An autonomic framework for integrating security and quality of service support in databases. In *2012 IEEE sixth international conference on software security and reliability (SERE)* (pp. 51–60).

Alomari, F., & Menasce, D. (2013). Efficient response time approximations for multiclass fork and join queues in open and closed queuing networks. *IEEE Transactions on Parallel and Distributed Systems, 25*(6), 1437–1446.

Alomari, F., & Menasce, D. (2013). Self-protecting and self-optimizing database systems: Implementation and experimental evaluation. In *Proceedings of the 2013 ACM cloud and autonomic computing conference, Article No. 18*.

Al-Sayid, N., & Aldlaeen, D. (2013). Database security threats: A survey study. In *2013 5th International conference on computer science and information technology (CSIT)* (pp. 60–64).

Amirijoo, M., Hansson, J., & Son, S. (2006). Specification and management of QoS in real-time databases supporting imprecise computations. *IEEE Transactions on Computers, 55*(3), 304–319.

Andres, G., Jose, H., Ernesto, R., & Alfredo, M. (2013). Indexing and retrieving in fingerprint databases under structural distortions. *Expert Systems with Applications, 40*(8), 2858–2871.

Bayon, L., Grau, J., Ruiz, M., & Suarez, P. (2012). The exact solution of the environmental/economic dispatch problem. *IEEE Transactions on Power Systems, 27*(2), 723–731.

Bennani, M., & Menasce, D. (2005). Resource allocation for autonomic data centers using analytic performance models. In *Second international conference on autonomic computing* (pp. 229–240).

Bertino, E., & Sandhu, R. (2005). Database security - concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing, 2*(1), 2–19.

Chen, J., Hu, C., Zeng, H., & Zhang, J. (2009). Impact of security on QoS in communication network. In *International conference on networks security, wireless communications and trusted computing: 2* (pp. 40–43).

Darwish, S., Guirguis, S., & Ghozlan, M. (2013). Intrusion detection in role administrated database: Transaction-based approach. In *International conference on computer engineering and systems (ICCES)* (pp. 73–79).

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Hababeh, I., Khalil, I., & Khreishah, A. (2015). Designing high performance web-based computing services to promote telemedicine database management system. *IEEE Transactions on Services Computing, 8*(1), 47–64.

Huang, B., Buckley, B., & Kechadi, T. (2010). Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. *Expert Systems with Applications, 37*(5), 3638–3646.

Jabbour, G., & Menasee, D. (2008). Policy-based enforcement of database security configuration through autonomic capabilities. In *International conference on autonomic and autonomous systems* (pp. 188–197).

Jabbour, G., & Menasee, D. (2009). The insider threat security architecture: A framework for an integrated, inseparable, and uninterrupted self-protection mechanism. In *International conference on computational science and engineering: 3* (pp. 244–251).

Kashif, K., Madjid, M., Shi, Q., & Sohail, A. (2013). Component-based security system (COMSEC) with QoS for wireless sensor networks. *Security and Communication Networks, 6*(4), 461–472.

Kamra, A., & Bertino, E. (2009). Survey of machine learning methods for database security. In *Machine learning in cyber trust* (pp. 53–71). Springer.

Kang, K., Oh, J., & Son, S. (2007a). Chronos: Feedback control of a real database system performance. In *Proceedings 28th IEEE international conference on real-time systems symposium (RTSS)* (pp. 267–276).

Kang, K., Son, S., & Stankovic, J. (2004). Managing deadline miss ratio and sensor data freshness in real-time databases. *IEEE Transactions on Knowledge and Data Engineering, 16*(10), 1200–1216.

Kleinrock, L. (1975). *Queueing systems*. New York, USA: Wiley.

Laura, C., Jorge, H., & Viviana, E. (2006). Real time database systems. In *Encyclopedia of database technologies and applications* (pp. 524–530). Hershey, Pa.: Idea Group Referene.

Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation, 13*(2), 284–302.

Lin, Q., Chen, J., Zhan, Z., Chen, W., Coello Coello, C. A., Yin, Y., et al. (2016). A hybrid evolutionary immune algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation* in press. doi:10.1109/TEVC.2015.2512930.

Martins, F., Carrano, E., Wanner, E., Takahashi, R., & Mateus, G. (2011). A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. *IEEE Sensors Journal, 11*(3), 545–554.

Menasce, D. (2004). Performance by design computer capacity planning by example. Upper Saddle River, NJ, USA: Prentice Hall PTR.

Menasce, D., & Kephart, J. (2007). Guest editors' introduction: Autonomic computing. *IEEE Internet Computing*, 18–21.

Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective evolutionary algorithms for Portfolio management: A comprehensive literature review. *Expert Systems with Applications, 39*(14), 11685–11698.

Mostafa, H., Pal, P., & Hurley, P. (2014). Message passing for distributed QoS-security tradeoffs. *The Computer Journal, 57*(6), 840–855.

Nieto, A., & Lopez, J. (2014). A context-based parametric relationship model (CPRM) to measure the security and QoS tradeoff in configurable environments. In *2014 IEEE international conference on communications (ICC)* (pp. 755–760).

Parmar, J. (2014). Data security, intrusion detection, database access control, policy creation and anomaly response systems-A review. In *2014 International conference on advances in engineering and technology research (ICAETR)* (pp. 1–6).

Poolsappasit, N., Dewri, R., & Ray, I. (2012). Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing, 9*(1), 61–74.

Rao, U., Singh, N., Amin, A., & Sahu, K. (2014). Enhancing detection rate in database intrusion detection system. In *Science and information conference (SAI)* (pp. 556–563).

Ronay, A., Li, Y., Valeria, V., Enrico, Z., Enrique, L., & Carlos, M. (2013). NSGA-II–trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Expert Systems with Applications, 40*(4), 1205–1212.

Rubio-Largo, A., Vega-Rodriguez, M., Gomez-Pulido, J., & Sanchez-Peerez, J. (2012). A comparative study on multiobjective swarm intelligence for the routing and wavelength assignment problem. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 42*(6), 1644–1655.

Saad, E., Mahdi, K., & Zbakh, M. (2012). Cloud computing architectures based IDS. In *2012 International conference on complex systems* (pp. 1–6).

Sengupta, S., Das, S., Nasir, M., Vasilakos, A. V., & Pedryc, W. (2012). An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 42*(6), 1093–1102.

Shaygan, M., Alimohammadi, A., Mansourian, A., & Govara, Z. (2014). Spatial multi-objective optimization approach for land use allocation using NSGA-II. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7*(3), 906–916.

Srivastava, M. (2014). Algorithm to prevent back end database against SQL injection attacks. In *2014 International conference on computing for sustainable global development (INDIACom)* (pp. 754–757).

Taneja, N., Raman, B., & Gupta, I. (2011). Chaos based partial encryption of spiht compressed images. *International Journal of Wavelets Multiresolution and Information Processing, 9*(2), 317–331.

Tang, L., Li, T., Jiang, Y., & Chen, Z. (2014). Dynamic query forms for database queries. *IEEE Transactions on Knowledge and Data Engineering, 26*(9), 2166–2178.

Wang, Y., Li, H., Yen, G., & Song, W. (2014). MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. *IEEE Transactions on Cybernetics, 45*(4), 830–843.

Woochul, K., Son, S., & Stankovic, J. (2012). Design, implementation, and evaluation of a QoS-aware real-time embedded database. *IEEE Transactions on Computers, 61*(1), 45–49.

Zhu, Q., Lin, Q., Du, Z., Liang, Z., Wang, W., Zhu, Z., Chen, J., Huang, P., & Ming, Z. (2016). A novel adaptive hybrid crossover operator for multiobjective evolutionary algorithm. *Information Sciences, 345*, 177–198.