Yadigar N. Imamverdiyev
Institute of Information Technology of ANAS, Baku, Azerbaijan
yadigar@lan.ab.az
DOI: 10.25045/jpit.v09.i2.09

A MODEL FOR OPTIMAL PLANNING OF INFORMATION SECURITY INCIDENT RESPONSE OPERATIONS

A quick and adequate response to handling of information security incidents is critical for ensuring business continuity. To handle such incidents, special CERT commands are required, but the cost of maintaining them is a burden for most organizations, and they prefer to use the services of special CERT service providers. This study proposes a model for the optimal distribution of information security incident response operations between CERT groups; the model is formulated as an optimization problem, and differential evolution algorithm is developed to solve it.

Keywords: *information security, incident response, incident handling, incident management, CERT, CSIRT, scheduling, differential evolution.*

Introduction

Information security tools are not perfect, thus, there are dynamic changes in the field of information technology, and it is not possible to allocate the necessary resources to ensure information security. Unfortunately, for these reasons, full information security and information security breaches cannot be provided and information security incidents occur. The more the level of informatization of society increases, the more the level of financial losses and risks increases due to these incidents. Therefore, organizations have to respond to information security incidentsfast and sufficiently. To gain competitive advantage, it is also important not only to defend, but also to react effectively to information security incidents. Frequently, organization can take advantage of incidents rather than suffering loss by eliminatingthe incidents and ensuring high quality and effective response and control [1].

The first information security incident handling experience has shown that special teams - CERT (*Computer Emergency Response Team*) teams should be organized to perform this work [2]. The term of CERT is officially registered in the United States and the copyright belongs to Carnegie Mellon University. In Europe, the term CSIRT is most commonly used (*Computer Security and Incident Response Team*). At present, many CSIRTsare operating in a number of countries and its various organizational models are available [3]. The number of information security incidents is rapidly increasing, and maintenance of a large number of highly-qualified specialists in CSIRT requires substantial funds, and as a result, the essential problem for all CSIRTs is to balance the workload with limited human resources [4]. Therefore, planning an incident handling optimization ischallenging for CSIRT.

Large-scale information security incidents cover several security domains. Such incidents can blow the information systems of critical infrastructures and ultimately threaten the lives, property, economy, and even the national security [5, 6]. The rapid identification of large-scale incidents, information exchange, investigation and coordinated response and elimination of consequences can often greatly reduce the damage caused by such malicious actions. Several CERT teams are involved in the elimination of these incidents, and the function of the national CERT includestheir coordination. At the same time, the national CERT shall solve the problem of optimal planning.

For many organizations, information security measures become impossible to be realized in practice. It’s due to the lack of resources, personnel, expertise, and technology to address constantly-changing security issues.

Information security management services (e.g. *Managed Security Services, MSS*) can ensure security requirements of such organizations. MSS Providers (MSSP) offer a wide range of security services, including information security incidents handling. [7] discusses the different

types of services, the benefits and risks of using this service, and emphasizes the importance of selecting service providers, and evaluates the performance of these services. One of the most demanded services of MSSP is information security incident handling services.

For the aforementioned reasons, solution of the issue of operational distribution of information security incidents among real-time CERT-groups is urgent. This study proposesan approach for modeling the processes of information security incidents handling and developsthe Differential Evolution (DE) algorithm for its solution.

Processes of information security incidentshandling

The process of incidents handling requires a number of consecutive implementations (procedures). The sequence of these implementationsis determined by the incidents handlingrules in each organization [8]. The sequence of these implementationsmay vary in different organizations. This may be due to differences both in the organization and the terminology. The point is that the terms as *"incident handling", "incident response"* and *"incident management"* are often used as synonyms. However, they differsignificantly and should be reviewed for the identification of a typical set of procedures for incidents handling.

Incidents handling involves the identification of incidents (detecting and analyzing events, incidents, and alarms), systematization (prioritizing the incidents), analysis (what happened, how the damage is, what threats may be, what steps should be taken to prevent and restore) and responsesto incident (scheduling, coordination and implementation, dissemination of information, feedback and taking lesson) [1].

Incidents management refers not only to the incidents handling and responding to them, but also preventive measures. These measures may include the management of vulnerabilities, artifacts' management, user training, and increasingthe level of awareness.

ISO/IEC 27035:2011 offers a process model for managing information security incidents, which consists of the following five key stages [9]:

1. **Scheduling and preparation** - an incident management policy is developed and a competent team is structured to respond to information security incidents;
2. **Disclosure and awareness**– detectingincidents and informing about incidents;
3. **Evaluation and decision-making** - incidents are assessed and decisions are made their handling. For example, the vulnerabilities can be eliminated and business-processes can be thoroughly restored, or although the elimination of the consequences of the incident is delayed, the evidences of cyber-crimecan be collected;
4. **Response**– responding to the incident after expert analysis and post-incident restoration;
5. **Revealing lesson** - changes to improve the management processesof information security incidents and risk.

The procedures of responding to the incidents should be identified for the solution of the problem stated in the article. The following most commonly used procedures for CSIRT practice can be distinguished to respond to information security incidents [10-12]:

- **Identification (and registration)of incident**: Information about the incident is received or the incident is detected by any investigation or tool. The verification of the information about the incident and its belonging to CSIRT service area is checked.
- **Triage classification**: The incident is classified, its priority is defined, and ticketing is enabled for processing.
- **Collecting the evidence of the incident** and its processing.
- **Incident containment**. Necessary measures are taken to prevent the dissemination of the incident to other systems.
- **Incident eradication**. All exploited vulnerabilities are detected and eliminated. Malicious software and other unwanted components are deleted.
- **Recovery**. The system is recovered to the state prior to the incident.

- **Lesson learned from the incident**: occurred incident is analyzed and recommendations are developed to prevent similar incidents in the future.

Incident response procedures may vary depending on the type of incident, its criticality, possible damage, manager's attitude and so on.

Review of related works

The high cost of response to information security incidents encourages organizations to think about the expediency of maintaining their CSIRT team. In addition, organizations are not enthusiastic to share information about their security with other organizations. They hope that experts can help them defend from cyber threats without damaging their reputation. To meet these needs of organizations, a co-ordination model that supports response to security incidents in organizational architecture is proposed [13]. Additionally, the model also supports the function of presenting and submitting digital evidence to competent authorities during real-time monitoring and incident investigation.

Another study [14] identifies a number of significant and systematic shortcomings in response to incidents in Australian financial institutions. It shows that the response teams are experiencing significant knowledge; however organizations do not sufficiently use this experience to improve information security management processes. The article offers a number of recommendations and safety learning models in this regard.

It is difficult to automate the processes of information security incidents response, and it is mainly implemented by the human being taking critical advantage from the processes and technologies. In modern conditions, incident response is compounded by a number of reasons: the control over the computing environment is lost due to cloud computing and autosourcing services, the complexity of attacks increases, and security costs of organizations are insufficient. There is no alternative to information security incidents response, thus technology shall be projected to support people in this regard and maximize the chance to successfully fulfill their critical security functions.

Starting with the first CSIRTs, they encountered constant problems related to the work load, the quality of their services, and the range of users they serve. Responding to low priority and high priority incidents causes various problems [4, 16]. Low priority incident requests are exponentially increasing, which is several time more that the limited CSIRT resources. Responding to high priority incidents, long-term instability is observed in workloads and service quality, and the CSIRT's reputation gradually decreases within its users.

From the academic point of view, the issue can be formulated as a limited resource planning issue, which is referred to as an *Incident Response Scheduling Problem (IRSP)* in this article. The IRSP issue can also be classed as a *Resource Constrained Project Scheduling Problem (RCPSP)*, which simultaneously solves planning and scheduling issues [17]. This article offers an optimization approach to address the IRSP issue within the directive period of the incidents processing and at minimal cost. The main contribution of this article is the development of a modeling method, which allows real-time solutions. Simultaneous scheduling problem solution allows administrators real-time scheduling and implementing incident handling in accordance with the changing circumstances.

The RCPSP issue can be summarized as follows [18]. The time horizon [0, T], n number of jobs (implementations), $i = 1, ..., n$, and renewable resource – r, and $k = 1, ..., r$ is given. Resource can be a physical object, human, or computing resource (processor, memory). At any time moment $t = 0, ..., T$, a constant unit R_k of the k-th resource becomes available. The i-th job is executed within the time unit p_i, and during this period, the unit p_{ik} of the k-th resource is required. All data is assumed to be integers.

The purpose is to define the start time of $S_i \in \{0,1, \dots, T\}$ for the jobs $i = 1, ..., n$, so that the restrictions on resources and jobs are meet and the purpose function is given an optimal value.

Different purpose functions can be taken, for example, makespan $C_{max} := \max_i\{C_i\}$ may be required to be minimal, where $C_i := (S_i + p_i)$ is the completion time. The vector $S = (S_i)_{i=1}^n$ determines the project plan.

Note that RCPSP is a more general model; "open shop", "job shop", and "flow shop" are the special cases of RCPSP. In addition, the problems of classic commivoyager, the various options of the popular "bag" issue, the (linear, two-dimensional) packaging of containers, and the issue of compiling the training schedules in educational institutions can be expressed and solved as a matter of RCPSP. All of these show that RCPSP is applied in many areas, and therefore, numerous articles have published recently in this area [18]. Its application areas may include scheduling in manufacturing processes, tasks in multiprocessor, technical maintenance at airports, maintenance of aircrafts, sports competitions, audit boards and others [19].

The RCPSP combiner is a matter of optimization and is included in the NP-challenging issues [17]. There are three basic solution methods, namely precise methods, heuristic methods, and meta-evolutionary approaches to its solution. A good overview of these methods can be found in [20, 21]. Different variants of genetic algorithm [22-24], colony intellect method [25, 26], ant colony [27], bee colony methods [28-30], and DE algorithm [31, 32] have been proposed for solving the RCPSP problem.

This article uses the DE algorithm for solving the stated problem. DE algorithm was proposed by R.Storn and K.Price in 1997 [33], a powerful heuristic approach that combines classic crossover, mutation and selection operators in a simple way.

Problem statement

It is assumed that the CERT service provider is request to handle several security information incidents. The incidents can be received from different organizations (security domains). The CERT provider coordination center has to optimally distribute these jobs within own specialized incident response teams (CERT-groups) by certain criteria taking into account some restrictions. The CERT group can consist of only one person. Note that, for each incident, the priority of the incident is determined based on certain criteria (e.g. critical accidents) [34]. The prioritization of the incident determines the extent to which it will be operatively handled and what resources will be attracted for it.

The following restrictions can be considered:

- The incident handling (response) consists of several procedures;
- There procedures are followed in sequence, some procedures can be started only after all the preceding procedures are completed;
- Each incident responding procedure can be performed by only one response group;
- The response team cannot perform several handling procedures simultaneously;
- Certain incident responding procedures can be performed simultaneously;
- Some costs may be required when the response team moves from one incident handling to another (from one geographical location to another);
- Deadline can be assigned for the incident handling.
- Each response group can perform any incident procedure.

Optimization model for problem solving

Assume that the set of incidents J_1, J_2, …, J_n shall be handled by response groups R_1, R_2, …, R_m. Handling the incident J_1 consists of the procedure (i = 1, ..., n). It is assumed that the incidents do not depend on each other and there is no consistency relation between various incidents. The procedures of the same incident create chains for consistency: $O_{1j} \to O_{2j} \to \cdots \to O_{n_i,j}, i = 1, \dots, n.$. Each incidence J_1 is coordinated with the response time d_i and w_i (penalty coefficient for criticality or delay).

It is assumed that scheduling is divided into the time intervals of equal length called horizon periods (e.g. clocks), and the processing times are discrete versions of a period. Once the procedure is started, it cannot be stopped, i.e. no preemption is permitted. All response groups are available at time $t = 0$ and any incident handling can be started.

Each procedure can be handled by only one response group. If the j-th procedure is performed by the response team R_k, its processing time is t_{jk}. The response team R_k is available (vacant) during non-intersecting intervals $[s_k^v, l_k^v], v = 1, \dots, V_k$, where $l_k^v \le s_k^{v+1}, v = 1, \dots, V_k - 1$. In addition, the total working time of R_k can be limited byH_k^- from below and by H_k^+- from above, where $H_k^- \le H_k^+ (k = 1, \dots, m)$.

Table 1 illustrates a list of procedures for 3 incidents and the handling time of the procedures for 4 response groups. As it is seen from the table, for example, responding to the incident J_1 requires consistent implementation of the procedures O_{11}, O_{12} and O_{13}. The numerals at the intersection of the rows and columns of the table show the implementation period of respective procedure performed by the corresponding CERT-group R_1, R_2, R_3 and R_4 with the selected time units (e.g., hours).

Table 1.

Handling periods of procedures for three incidents and four response groups

Incidents	Procedures	Handling periods of procedures			
		R_1	R_2	R_3	R_4
J_1	O_{11}	1	3	4	1
	O_{12}	3	8	2	1
	O_{13}	3	5	4	7
J_2	O_{21}	4	1	1	4
	O_{22}	2	3	9	3
	O_{23}	9	1	2	2
J_3	O_{31}	8	6	2	5
	O_{32}	4	5	8	1

The scheme for the minimum processing time criteria for their CERT-groups based on the sequence of procedures may be as follows:

S={(O_{11}, R_1), (O_{12}, R_4), (O_{13}, R_1), (O_{21}, R_2), (O_{22}, R_2), (O_{23}, R_1), (O_{31}, R_3), (O_{32}, R_4)}

= {(O_{11}, R_1: 0-1), (O_{12}, R_4: 1-2), (O_{13}, R_1: 2-5), (O_{21}, R_2: 0-1), (O_{22}, R_2: 1-4),

(O_{23}, R_3: 4-6), (O_{31}, R_3: 1-3), (O_{32}, R_4: 3-4)}.

According to the scheme, for example, the procedure O_{11} is used by the CERT group R_1 shall be implemented within time interval [0; 1], the procedure O_{12} - by the CERT group R_4 within time interval [1; 2], the procedure O_{13} - by the CERT group R_1 within time interval [2; 5]. Thus, two CERT groups are involved in handling the incident J_1: the procedures O_{11} and O_{13} are implemented by the CERT group R_1, whereas the procedure O_{12} by the CERT group R_4.

Typically, Gantt diagram is used to visualize this type of schemes [18]. The Gantt diagram is shown in Figure 1 according to the scheme given above.

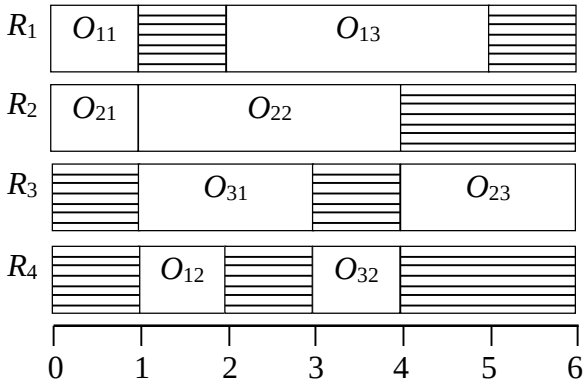

Figure 1. Diagram of handling scheme for three incidents and four response groups

As it is seen from the diagram, the CERT-group R_1 is vacant at intervals [1; 2] and [5; 6], the CERT-group R_2 – at [4; 6], the CERT-group R_3 – at [0; 1] and [3; 4], and the CERT-group R_4 – at [0; 1], [2; 3] and [4; 6].

Let's find the total handling duration of the incidents. Assume that completion time of the procedure O_{ij} is denoted by t_{ij}^F. The total handling duration can be defined by finding respective moments t_{ij}^F from Figure 1 for the abovementioned scheme and choosing the maximal one out of them. Thus, the total duration of the incidents will be 6 periods:

$$t = max\{t_{11}^F, t_{12}^F, t_{13}^F, t_{21}^F, t_{22}^F, t_{23}^F, t_{31}^F, t_{32}^F\} = max\{1,\ 2,\ 5,\ 1,\ 4,\ 6,\ 3,\ 4\} = 6.$$

Both the newly inserted and above mentioned signs are systematized as follows:

n - number of incidents;

m - number of response groups (CERT-groups);

n_i -total number of incident response procedures;

N -total number of response procedures, $N = \sum_{i=1}^{n} n_i$;

O_{ij} – j-th response procedure of the incidence i;

p_{ijk} - handling time of the procedure O_{ij} by the k-th CERT-group;

t_{ijk} – start time of the procedure O_{ij} by the k-th CERT-group;

t_{ij}^F – completion the procedure O_{ij};

i, h - index of incidents, $i, h = 1, 2, ..., n$;

k - index of response groups, where $k = 1, 2, ..., m$;

j, g - index of response procedures where $j, g = 1, 2, ..., n_i$;

d_i - directive response period of the i-th incidents;

T_i - delay time of response to the i-th incident;

w_i - weight of the i-th incident (degree of criticality or delay penalty coefficient);

W_k - total time spent on incidents handling by the k-th CERT-group;

$$x_{ijk} = \begin{cases} 1, & \text{if the } k-\text{th reponse group is assigned to the procedure } O_{ij}, \\ 0, & \text{otherwise} \end{cases}$$

The total time W_k spent on incidents handling by the k-th CERT-group is summarized by the abovementioned signs:

$$W_k = \sum_{i=1}^{n} \sum_{j=1}^{n_j} p_{ijk} x_{ijk}, \quad (1)$$

The response delay time T_i of the i-th incident is defined as follows:

$$T_i = max\left(t_{i,n_i}^F - d_i, 0\right), \quad (2)$$

Several criteria are often to be taken into account when scheduling an incident. Unquestionably, first of all, it is necessary to minimize the total time spent on the incidents. However, the workload should be allocated among CERT-groups so that no CERT-group will be overloaded. Moreover, the handling of critical incidents is also required to be performed within the directive period. Taking this into account, the article outlines the following criteria for minimizing the incidents:

1) Total time period spent on the incidents handling;
2) Maximum delay period of handling, taking into account the incident criticality;
3) Maximum time period spent on the incidents handling by the CERT-group.

Using the above signs, these criteria can be summarized as follows:

$$min\, F_1 = \max_{1\le i\le n}\left\{\max_{1\le j\le n_i}\left\{t_{ij}^F\right\}\right\}, \tag{3}$$

$$\min F_2 = \max_{1\le i\le n}\{w_i T_i\}, \tag{4}$$

$$min\, F_3 = \max_{1\le k\le m}\{W_k\}. \tag{5}$$

The model includes the following restrictions:

$$t_{ij}^F - t_{i,j-1}^F \ge p_{ijk}x_{ijk}, \qquad j = 2,\dots,n_i, \forall i,k \tag{6}$$

$$\left[(t_{hg}^F - t_{ij}^F - t_{hgk})x_{hgk}x_{ijk} \ge 0\right] \vee \left[(t_{ij}^F - t_{hg}^F - t_{ijk})x_{hgk}x_{ijk} \ge 0\right], \forall (i,j),(h,g),k \tag{7}$$

$$\sum_{k=1}^{m} x_{ijk} = 1, \quad \forall i,j. \tag{8}$$

Condition (6) provides restrictions on the sequence of the procedures. Condition (7) indicates that each CERT-group can handle only one procedure at a random moment. Condition (8) indicates that only one responding group can be selected for handling of each procedure.

In this study, the simplest approach is taken to addressing multi-criteria optimization [35]. Thus, the general purpose function is defined as a weighted sum of the above mentioned purpose functions, giving the same weight to each one:

$$F = \frac{1}{3}F_1 + \frac{1}{3}F_2 + \frac{1}{3}F_3. \tag{9}$$

Solution algorithm for optimization

Describing solution with vectors

There are two sub-issues in the optimization issue: the issue of assigning the procedures to CERT-groups and the issue of the sequence of procedures. Therefore, it would be expedient to show the solution with two vectors [36]. One of the vectors shows the permutation of the procedures, while the other describes the destination of the procedures to the CERT-groups. Both vectors are *N*-dimensional, where *N* denotes the total number of procedures. Suggested descriptive method ensures that the solution meets the condition of sequence. Note that these two vectors can be combined in one vector.

Permutation vector of procedures. Permutation is used to indicate the sequence of procedures. Procedures related to the same incident are indicated by the same number in the vector. This descriptive method of the procedures is illustrated in Figure 2.

O_{31}	O_{11}	O_{21}	O_{12}	O_{32}	O_{22}	O_{23}	O_{13}
3	1	2	1	3	2	2	1

Figure 2. Permutation vector of procedures

For instance, the numeral 3, which is the first element of the vector, is referred to the incident 3, furthermore, it is found in the incident 3 for the first time, therefore, it is referred to the first procedure of the incident 3, i.e., to O_{31}. Similarly, the numeral 3, which is the fifth element of the vector, is found in the incident 3 for the second time, therefore, it indicates the second procedure the incident 3, i.e., O_{32}.

Destination vector of CERT-groups. This vector element indicates the CERT-group assigned to the appropriate procedure. Destination vector is illustrated in Figure 3. For example, O_{11}, written above the first element of the vector, specifies that the first procedure of the incident 1 will be performed by CERT-group 1. Similarly, O_{12} and O_{13}indicate that the second procedure of the incident will be performed by CERT group 4, whereas of the third procedure will be implemented by the CERT group 1.

Generation of initial population

The initial population affects the performance of the algorithm. To improve the proposed algorithm, the initial population by Kacem and co-authors' approach is used [37]. In this approach, the initial population is generated by DestinattionRule1 and AssigmentRule2. DestinattionRule1 focuses on minimizing the handling time, and determines the sequence of procedures and CERT-groups. AssigmentRule2 provides the diversity of populations.

Used algorithm randomly generates the sequence of procedures and CERT-groups. Once the destinattions are determined, the sequence of procedures in CERT-groups is adjusted. This is achieved through three different methods: random (incidents are randomly selected), mostly remaining work and mostly remaining procedures are selected. The proposed algorithm generates 40% of the initial population using DestinattionRule1, and 60% - using DestinattionRule2. The sequence of the procedures is regulated by the abovementioned three dispatching rules consecutively.

Discrete DE algorithm

DE is a stochastic, population-based optimization algorithm. The concept of DE algorithm is derived from the extensive evolution algorithms, which includes genetic algorithms. Combining existing solutions and several mechanisms, such as mutation, crossing, and selection, it is used to obtain new solutions or optimal solution or at least to meet the conditions of the problem. DE is proposed for uninterrupted value optimization [38]. [39] suggests DE algorithm for binary variables. The options for the integer optimization issues of DE are also suggested [40, 41].

The discrete DE (DDE) algorithm proposed in [42] is modified to address the issue discussed in this article. Assume that the individuals of population are described as *2N*-dimensional vectors $x_i, \forall i \in \{1, \dots, N_p\}$, where *N* is the total number of incidents, while N_p is the population volume.

Mutation. Initially, a mutation operation is applied. The following equations can be used for the mutant population:

$$V_i^t = P_m \otimes F(x_i^{t-1}), \tag{10}$$

$$V_i^t = P_m \otimes F(x_a^{t-1}), \tag{11}$$

$$V_i^t = P_m \otimes G\left(x_g^{t-1}\right). \tag{12}$$

Where V_i^t is the *i*-th mutant individual of the population in the *t*-th iteration, where x_i^{t-1} is the *i*-th mutant individual of the population in the $(t-1)$-th iteration; x_a^{t-1}- an individual randomly selected from the population in the $(t-1)$-th iteration; x_g^{t-1}- the best global solution in the $(t-1)$-th iteration. *G* is a mutation operator, and P_m - mutation probability. $\otimes$ is a condition

operator, when the probability left on its left side is provided by certain condition, the mutation operator on its right side is enabled.

Assume that the equation (10) is used for the mutant population. A regularly distributed random number *r* is selected from [0; 1]. If $r < P_m$, then the operator $V_i^t = F(x_i^{t-1})$ is used to generate mutant individual in the *t*-th iteration. Otherwise, the mutant individual in the *t*-th iteration is adopted as $V_i^t = x_i^{t-1}$.

Crossover. Afterwards, an experimental individual is generated, for which a mutated individual V_i^t is crossed with the target individual $x_i^{t-1}$from the current population by the following rule:

$$y_i^t = P_c \otimes CR(x_i^{t-1}, V_i^t), \tag{13}$$

where *CR* is a crossover operator, and P_c - crossover probability. If $r < P_c$ for regularly distributed numeral $r \in (0,1)$, then crossover operator *CR* is applied to generate an experimental individual: $y_i^t = CR(x_i^{t-1}, V_i^t)$. Otherwise, an experimental individual is selected as $y_i^t = V_i^t$.

Selection. Mutation and crossover processes are followed by selection procedure. The compliance value of the experimental individual is calculated and compared to the compliance value of the target individual. The most appropriate one out of these two individuals are included in the following population:

$$x_i^t = \begin{cases} y_i^t, if\ F(y_i^t) \le F(x_i^{t-1}) \\ x_i^{t-1}, \text{otherwise} \end{cases} \tag{14}$$

By applying the above procedures to all individuals of the population, a new population is acquired, and this is repeated until the predetermined duration criteria are provided. The individual of the best last generation is taken as a solution of the problem.

The above-mentioned DDE algorithm uses the following operators derived from genetic algorithms such as mutation operator and crossover operator [43].

Mutation operator. A random procedure is chosen to mutate the sequence of procedures; the very preceding procedure and the very subsequent procedure are selected. Assume that their indexes are *a* and *b* accordingly. The selected procedure is placed within the range (*a, b*). This mutation algorithm maintains a sequence limitation and, therefore, ensures the solution to be within the range of possible solutions. This strategy is offered to mutate the destination vector of CERT-groups. The procedures implemented by the CERT-group with the minimum load are sought out of the CERT-group, which has the maximum work load. One of these procedures is assigned to a CERT-group with the minimum work load.

Crossover operator. The first part of the solution vector performs the crossover operator POX (*Precedence preserving order-based crossover*) for the sequence of procedures [44], and the second part uses the random multipoint crossover operator [45]. A useful derivative can be generated this way, which provides restriction conditions.

Conclusion

The complexity and negative consequences of cyber-attacks are constantly growing, and in this regard, rapid response to information security incidents is of particular importance for the permanence of business processes. The proposed model enabled the optimal scheduling the work of CERT-groups to eliminate the incidents as soon as possible taking into account handling priorities. Future studies include algorithms based on different modifications of optimization modeling and other evolution methods for their solution taking into account different features (e.g. the incidents received by CERT service at different times and their prevention with minimized costs).

References

1. Alguliyev R.M., Imamverdiyev Y.N. Information security incidents. Baku: "Information Technologies" publishing house, 2015, p. 219.
2. Cichonski P., Millar T., Grance T., andScarfoneK. Computer security incident handling guide.NIST Special Publication800-61, 2012, 147 p.
3. West-Brown M.J., Stikvoort D., and Kossakowski K.-P. Handbook for Computer Security Incident Response Teams (CSIRTs). CMU/SEI-2003-HB-002. 2003, 223 p.
4. Wiik J., Gonzalez J.J., Davidsen P.I., and Kossakowski K.P. Chronic workload problems in CSIRTs / Proc. of the 27th International Conference of the System Dynamics Society, 2009, pp.1–19.
5. Osorno M., Millar T., Rager D. Coordinated cybersecurity incident handling: Roles, processes, and coordination networks for crosscutting incidents / Proc. of the 16th ICCRTS "Collective C2 in Multinational Civil-Military Operations", 2011, pp.1–12.
6. Tøndel I.A., Line M.B., Jaatun M.G. Information security incident management: Current practice as reported in the literature // Computers & Security, 2014, vol.45, pp.42–57.
7. Deshpande D. Managed security services: An emerging solution to security / Proc. 2nd Annual Conference on Information Security Curriculum Development, 2005, pp.107–111.
8. Alberts C., DorofeeA., KillcreceG., and ZajicekR. R. M. Defining incident management processes for CSIRTs: A work in progress. Carnegie Mellon Software Engineering Institute, 2004, 249 p.
9. ISO/IEC 27035:2011 Information technology – Security techniques – Information security incident management. 2011, 78 p.
10. Mitropoulos S., Patsos D., Douligeris C. On incident handling and response: A state-of-the-art approach // Computers & Security, 2006, vol.25, pp.351–370.
11. Hidayah N., Rahman A., Kim K., Choo R.A survey of information security incident handling in the cloud // Computers & Security, vol.49, 2015, pp.45–69.
12. KnowlesW., PrinceD., HutchisonD., DissoJ. F. P., JonesK.A survey of cyber security management in industrial control systems // International Journal of Critical Infrastructure Protection, 2015, vol.9, pp.52–80.
13. Jeong K., Park J., Kim M., Noh B. A security coordination model for an inter-organizational information incidents response supporting forensic process / Fourth International Conference on Networked Computing and Advanced Information Management, 2008, vol.2, pp.143–148.
14. Atif A., Maynard S.B., and Shanks G. A case analysis of information systems and security incident responses // International Journal of Information Management, 2015, vol.35, no.6, pp.717–723.
15. Schneier B. The future of incident response // IEEE Security & Privacy, 2014, vol.12, no.5, pp.96–96.
16. Kuypers M. A., Maillart T., and Paté-Cornell E. An empirical analysis of cyber security incidents at a large organization. Working Paper. 2016, 22 p.
17. Brucker P., Drexl A., Möhring R., Neumann K., andPesch E. Resource-constrained project scheduling: Notation, classification, models, and methods // European Journal of Operational Research, 1999, vol.112, no.1, pp.3-41.
18. Brucker P., and Knust S., Complex Scheduling. GOR-Publications, 2012, 352 p.
19. Artigues C., Demassey S., andNeron E. (Eds.) Resource-constrained project scheduling: models, algorithms, extensions and applications. John Wiley & Sons. 2008, 288 p.
20. Kolisch R., Hartmann S. Experimental investigation of heuristics for resource constrained project scheduling: an update //European Journal of Operational Research, 2006, vol.174, no.1, pp.23–37.

21. Habibi F., Barzinpour F., and Sadjadi S. Resource-constrained project scheduling problem: review of past and recent developments // Journal of Project Management, 2018, vol.3, no.2, pp.55–88.
22. Alcaraz J., Maroto C., and Ruiz R. Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms // Journal of the Operational Research Society, 2003, vol.54, no.6, pp.614–626.
23. Valls V., Ballestini F., Quintanilla S. A hybrid genetic algorithm for the resource constrained project scheduling problem // European Journal of Operational Research, 2008, vol.185, no.2, pp.495–508.
24. Gen M., Gao J., and Lin L. Multistage-based genetic algorithm for flexible job-shop scheduling problem // Intelligent and Evolutionary Systems, 2009, pp.183–196.
25. Koulinas G., Kotsikas L., andAnagnostopoulos K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem // Information Sciences, 2014, vol.277, pp.680–693.
26. Tang D., Dai M., Salido M. A., andGiret A. Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization // Computers in Industry, 2016, vol.81, pp.82–95.
27. Myszkowski P. B., Skowroński, M. E., Olech, Ł. P., andOślizło K. Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem // Soft Computing, 2015, vol.19, no.12, pp.3599–3619.
28. Li J. Q., Pan Q. K., andGao K. Z. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems // The International Journal of Advanced Manufacturing Technology, 2011, vol.55, no.9, pp.1159–1169.
29. Akbari R., Zeighami V., and Ziarati K. Artificial bee colony for resource constrained project scheduling problem // International Journal of Industrial Engineering Computations,2011, vol.2, no.1, pp.45–60.
30. Gao K. Z., Suganthan P. N., Pan Q. K., Chua T. J., Chong C. S., andCai T. X. An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time // Expert Systems with Applications, 2016, vol.65, pp.52–67.
31. Damak J., Jarboui B., Siarry P., Loukil T. Differential evolution for solving multi-mode resource-constrained project scheduling problems //Computers & Operations Research, 2009, vol.36, no.9, pp.2653–2659.
32. Afshar-Nadjafi, B., Karimi H., Rahimi A., andKhalili S. Project scheduling with limited resources using an efficient differential evolution algorithm // Journal of King Saud University-Engineering Sciences, 2015, vol.27, no.2, pp.176–184.
33. Storn R., Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces //Journal of Global Optimization, 1997, vol.11, no.4, pp.341–354.
34. Imamverdiyev Y.N. An information security incident prioritization method / Proc. of the 7th International Conference on Application of Information and Communication Technologies, 2013, pp.183–187.
35. Hsu T., Dupas R., Jolly D., & Goncalves G. Evaluation of mutation heuristics for the solving of multiobjective flexible job shop by an evolutionary algorithm / Proc. of the IEEE International Conference on Systems, Man and Cybernetics, 2002, vol.5, pp.655–660.
36. Shao X., Liu W., Liu Q., & Zhang C. Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem // The International Journal of Advanced Manufacturing Technology,2013, vol. 67, no.9–12, pp.2885–2901.
37. Kacem I., Hammadi S., & Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems // IEEE Transactions on Systems, Man, and Cybernetics, Part C,2002, vol.32, no.1, pp.1–13.

38. Das S., &Suganthan P. N. Differential evolution: A survey of the state-of-the-art // IEEE Transactions on Evolutionary Computation, 2011, vol.15, no.1, pp.4–31.
39. Alguliev R. M., Aliguliyev R. M., &Hajirahimova M. S. Quadratic Boolean programming model and binary differential evolution algorithm for text summarization // Problems of Information Technology, 2012, no.2, pp.20–29.
40. Deng C., Liang C. Y., Zhao B., Yang Y., & Deng A. Y. Structure-encoding differential evolution for integer programming // Journal of Software, 2011, vol.6, no.1, pp.140–147.
41. Li H.,& Zhang L. A discrete hybrid differential evolution algorithm for solving integer programming problems // Engineering Optimization, 2014, vol.46, no.9, pp.1238–1268.
42. Pan Q. K., Tasgetiren M. F., & Liang Y. C. A discrete differential evolution algorithm for the permutation flowshop scheduling problem // Computers & Industrial Engineering, 2008, vol.55, no.4, pp.795–816.
43. Pezzella F., Morganti G., Ciaschetti G. A genetic algorithmfor the flexible job-shop scheduling problem // Computers & Operations Research, 2008, vol.35, no.10, pp.3202–3212.
44. Shi G. A genetic algorithm applied to a classic job-shop scheduling problem // International Journal of Systems Science, 1997, vol.28, no.1, pp.25–32.
45. Elgendy A. R., Mohammed H., &Elhakeem A. Optimizing dynamic flexible job shop scheduling problem based on genetic algorithm // International Journal of Current Engineering and Technology, 2017, vol.7, pp.368–373.