https://doi.org/10.52326/ic-ecco.2021/SEC.01

Prioritization of Cybersecurity Measures

Ion Bolun
Technical University of Moldova; 168, Stefan cel Mare Bd., MD-2004, Chisinau, Republic of Moldova;
ion.bolun@isa.utm.md, https://utm.md/

Abstract. **The considerable losses caused by the low level of cyber security of companies, institutions and so on and the limited financial resources available imply the need to prioritize the implementation of measures to counter cyber attacks. For this purpose, the respective optimization problem is formulated as a Boolean mathematical programming one. At large dimensions, the use of known methods of solving the problem requires a large volume of calculations. That's why, a simple rule for approximately solving the problem is obtained. By computer simulation, it is shown that the error of the solution when using this rule decreases considerably with the increase of the number of cybersecurity measures, more accurate – the more detailed such measures are. In order to reduce the error of solutions, three other simple algorithms are also proposed. The latter of these algorithms is more detailed and allows to reduce to a greater extent the solutions' error.**

Keywords: algorithm; entity; cybersecurity means; ordering rule; optimization problem.

I. Introduction

Information is a strategic resource. Many parts of it are confidential (personal data, commercial secret, state secret, e-transactions, etc.). Unauthorized access to such information, but also massive and targeted misinformation of the population, especially through Internet, leads to considerable losses, slowing the pace of economic growth and of advancing the well-being of the population.

For example, IT frauds cause losses of 0.5-5% of the total expenditure of public institutions [1]. Cybersecurity Ventures expects global cybercrime costs to grow by 15 percent per year over the next five years, reaching $10.5 trillion USD annually by 2025 [2]. This is about 9,3% of the global GDP estimate of 113,5 trillion USD done by International Monetary Fund [3].

At the same time, global spending on cybersecurity products and services for defending against cybercrime is projected to exceed $1 trillion cumulatively over the five-year period from 2017 to 2021 [4]. Part of it, the global information security market, is forecast to grow at a five-year CAGR of 8.5% to reach $170.4 billion in 2022 [5].

So, the global losses caused by the low security of cyberspace are considerably higher than the costs of implementing and maintaining those means of cybersecurity. Anyway, both are considerable and it is important the rational use of available resource for the efficient security of informatics infrastructures at all levels, especially of the critical ones. In this paper, some aspects of the prioritization of cybersecurity measures are systemized and discussed, taking into account the available financial resources.

II. Essence of the Problem

They are considered n entities (cybersecurity measures, companies, subdivisions, components of informatics infrastructure, etc.) in terms of improving their cybersecurity. For the latter, in this paper, the term "informatics security" (i-security) usually is used. Each entity $i \in G$, $|G| = n$, is characterized by:

- a_i – annual losses caused by the insufficient degree of i-security within entity i;
- b_i – annual costs with the implementation and maintenance of needed degree of i-security within entity i. It is considered that at a such degree of i-security there is $a_i = 0$;
- C – financial resources available for the implementation and maintenance of i-security means.

Obviously, the entities for which $a_i \le b_i$ ($c_i = a_i - b_i \le 0$) and also those for which $b_i > C$ is not rational to include in set G. So, one has $a_i > b_i$ and $b_i \le C$.

It is necessary to determine the optimal set $B \subseteq G$ of entities within which to ensure the necessary degree of i-security by minimizing the summary annual losses and costs involved

$$I = \sum_{i \in A} a_i + \sum_{k \in B} b_k \to \min \quad (1)$$

at

$$\sum_{k \in B} b_k \le C, \quad (2)$$

$$A \cup B = G, \quad (3)$$

where A is the set of entities with insufficient degree of i-security, and B is the set of i-secure entities.

The problem {(1)-(3)} is one of mathematical programming. Taking into account its peculiarities, this problem can be formulated in another form. Let x_i is a Boolean variable that takes the value 0 if $i \in A$ and the value 1 if $i \in B$. Then the summary annual losses and

https://doi.org/10.52326/ic-ecco.2021/SEC.01

costs with i-security for entity i can be determined as $(1 - x_i)a_i + x_i b_i$, $i \in G$, and relations (1) and (2) – as

$$I = \sum_{i \in G}[(1 - x_i)a_i + x_i b_i] \to \min \quad (4)$$

at

$$\sum_{i \in G} x_i b_i \le C, \quad (5)$$

Thus, the problem {(1)-(3)} can be replaced by the {(4), (5)} one. The latter is a Boolean mathematical programming problem with the unknowns x_i, $i \in G$.

To solve the problem {(4), (5)}, the respective well-known methods/computer applications can be used. But, first, at large dimension of the problem they need a considerable volume of calculations and, second, not always the respective means are available. Therefore, sometimes may be sufficient another, simplistic, approach.

III. The Main Rule to Problem Solving

Let's consider the problem in the form {(1)-(3)}.

Statement 1. It is preferable, in the meaning of (1), to ensure the needed degree of i-security within entity i than to ensure it within entity j, if at equal other conditions occurs $a_i / b_i > a_j / b_j$, that is

$$i \succ j, \text{ if } a_i / b_i > a_j / b_j. \quad (6)$$

Indeed, let's examine the case when sizes a_l, b_l, $l \in G$ are natural numbers. If these sizes are financial data, for example losses and costs in euro cents or dollar cents, then they are natural numbers.

Let $c_s = a_s - b_s$, $s \in G$ and M is the greatest common factor of the natural numbers b_i and b_j, and $z_i = b_i / M$, $z_j = b_j / M$. So, entity i can be considered as consisting of z_i unities to each of which corresponds (conventionally) the value M of annual costs for the implementation of necessary degree of i-security and, similarly, entity j can be seen as consisting of z_j unities to each of which corresponds (conventionally) the value M of annual costs for the implementation of necessary degree of i-security.

Then, because of $M = b_i / z_i = b_j / z_j$, if $c_i / z_i > c_j / z_j$ and other conditions are equal, it is preferable to implement the i-security means within entity i than to implement them within entity j. By replacing z_i and z_j in this inequality, one has $c_l / z_i = Mc_i / b_i > c_j / z_j = Mc_j / b_j$, that is $Mc_i / b_i > Mc_j / b_j$, so $c_i / b_i > c_j / b_j$ or $a_i / b_i > a_j / b_j$. ▼

Now, using the same approach, one can observe that the rule (6) is adequate also for real positive values of sizes a_l, b_l, $l \in G$, too. ■

IV. A Simple Approximate Solution to the Problem

The first step, when initiating the works on i-securing a set of entities, is to determine the parameters a_i and b_i values for each entity i. The second one, knowing the available in this purpose financial resources C, is to exclude from the project entities for which $a_i \le b_i$ ($c_i = a_i - b_i \le 0$) and also those for which $b_i > C$. So, one has to solve the problem (1)-(3).

Further, it is easy to observe that conditions (6) are transitive, that is if $a_i / b_i > a_j / b_j$ and $a_j / b_j > a_k / b_k$, then $a_i / b_i > a_k / b_k$. Thus, the third step is to order and renumber the entities $i \in G$ according to the rule

$$i \succ i + 1, \text{ if } a_i / b_i > a_{i+1} / b_{i+1}, \; i = \overline{1, n}. \quad (7)$$

The fourth step is to include in set B the first k entities of set G, where $k := \max\{i \mid \sum_{j=1}^{i} b_j \le C\}$.

Usually, the probability that takes place the equality $\sum_{i=1}^{k} b_i = C$ is very small. At the same time, in majority of cases the value of the difference $C - \sum_{i=1}^{k} b_i > 0$ may be acceptable, because the remained financial resources $C - \sum_{i=1}^{k} b_i$ can be used in other projects. In these cases, the ordering and renumbering of entities $i \in G$ according to rule (7) and then determining the set B, i.e. of such an approximate solution, is sufficient. The described above procedure of four steps, further as algorithm A_1 is addressed. In order to quantitatively estimate the dependence on various factors of the relative error of solutions, obtained when applying such a procedure, the computer simulation is performed. In calculations, the following initial data were used: simple size 10^4; $C = 1000$ units; $n = \{10, 15, 20, 30, 40, 50, 60, 80, 100\}$ entities; the value of parameters a_i, b_i, $i = \overline{1, n}$ are generated stochastically at the uniform distribution in intervals described below.

For each pair $\{a_i, b_i\}$, the value of size $b_i \in (0; C/d]$ is generated, where $d \in \{2, 3, 4, 5, 6, 7, 10, 15, 20, 25, 30, 40, 50\}$ and additionally $d < n$, because at $d = n$ the relation $\sum_{i=1}^{k} b_i < C$ always takes place; then, the value of size $a_i \in (b_i; 20\, b_i]$ is generated. In the latter interval, the value 20 is obtained, taking into account (very approximately) the value of the ratio 6.0 trln USD (in 2021) / 0.2 trln USD (in 2021) = 30 (see section I); the value 30 is reduced to 20, because many of the areas are not yet covered by cybersecurity measures.

The relative deviation δ (in %), equal to the average of the values $100\left(C - \sum_{i=1}^{k} b_i\right)/C$, usually can be used to compare the solutions. When calculating δ, cases for which $\sum_{i=1}^{n} b_i < C$ are not taken into account. At the same time, it is important to know the number of cases used when calculating δ. For this purpose, the relative frequency ρ is used, where ρ = 100 × (number of cases used when calculating δ) /10^4, %.

Some results of calculations, obtained using the SIMSEC application, are shown in Figures 1 and 2.

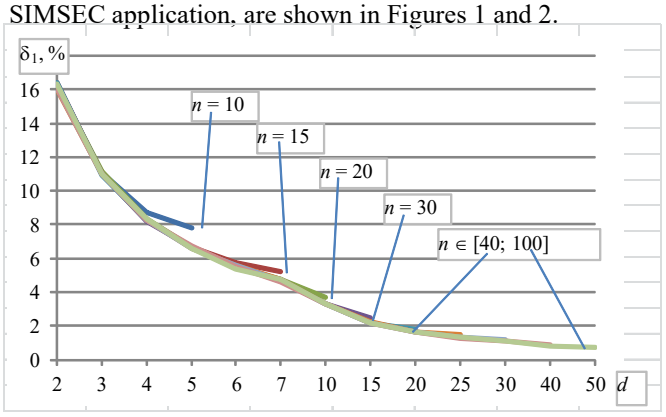

Figure 1. Dependence of error δ_1 on n and d at algorithm A_1.

https://doi.org/10.52326/ic-ecco.2021/SEC.01

From Figure 1, it can be seen that the deviation δ_1 of solutions, obtained by algorithm A_1, practically does not depend on the value of n, but decreases rapidly with the increase of d, becoming lower than 2% at about $d \geq 15$ and lower than 1% at $d \geq 40$. The maximum value of δ_1, equal to approx. 16%, is at $d = 2$, and the minimum value of δ_1, from the specified above alternatives of initial data, is of 0.70% at $d = 50$.

As for the frequency ρ (see Figure 2), it increases with the increase of n at (approx.) $n/3 < d < 2n/3$, but decreases rapidly with the increase of d, becoming 0 at approx. $d > 2n/3$. The maximum value of ρ, equal to 100 %, is at (approx.) $d < n/3$. Also, it takes place ρ ≥ 50% at $d \leq n/2$. Therefore, from the performed other calculations they are taken into account only cases in which $d \leq n/2$.

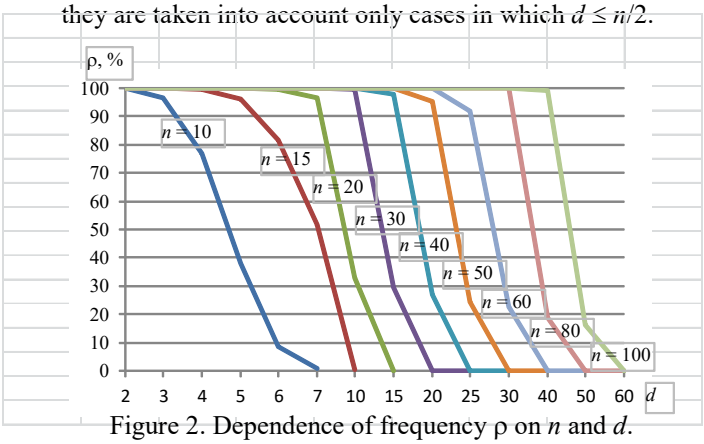

Figure 2. Dependence of frequency ρ on n and d.

Based on the results of performed calculations, it can be said that, in most cases, the use of the simplistic approach described above may be acceptable. In order to reduce the δ deviation at the given value of n, it is necessary to increase the parameter d value, that is to reduce the upper limit for the b_i, $i = \overline{1,n}$ values. Another way is to increase the value of the number n of entities and also that of the parameter d (see Figures 1 and 2).

V. Reducing the Solutions' Error

If $\sum_{i=1}^{k} b_i \neq C$, the question appear: how large can be the difference $C - \sum_{i=1}^{k} b_i$? The answer is formalized by the requirement $C - \sum_{i=1}^{k} b_i \leq \varepsilon$. Thus, instead of requirement (2), the following one has to be used

$$\sum_{k \in B} b_k \leq C \leq \sum_{k \in B} b_k + \varepsilon, \qquad (8)$$

and the problem {(1)-(3)} takes the form of {(1), (3), (8)}. Here, it should be noted that if the value of ε is too small, the solution may not exist.

In many cases, the value of $\delta = C - \sum_{i=1}^{k} b_i$ can be reduced by checking and, if convenient, including in set B of some entities from set $A = G/B$ so that $C - \sum_{i \in B} b_i \geq 0$. One such simple algorithm (A_2) is the following.

1. Initial data: C; N; a_i, b_i, $i \in G$; $n = |G|$, where $a_i > b_i$, $b_i \leq C$, $i \in G$.
2. $Z := \sum_{i \in G} b_i$. If $Z \leq C$, then $B_{opt} := G$, $I_{opt} := Z$, $Y := Z$, $\delta := 0$ and go to Step 14.
3. Determining the primary solution according to the A_1 algorithm of Section IV.
   3.1. Ordering and renumbering of entities $i \in G$ according to the rule (7).
   3.2. $k := \max\{i \mid Z \leq C\}$, where $Z = \sum_{j=1}^{i} b_j$ (the first part of restriction (8) is followed). $Z := \sum_{j=1}^{k} b_j$, $B := \{1, 2, \ldots, k\}$, $D := \sum_{i=k+1}^{n} a_i$ and $I := Z + D$. $I_{opt} := I$, $B_{opt} := B$, $\delta := C - Z$.
4. Reducing the value of δ by adding to set B of some entities from set A.
   4.1. If $k \geq n - 1$, the value of δ will not be reduced. Go to Step 5.
   4.2. $i := k + 2$.
   4.3. If $b_i \leq \delta$, then $B_{opt} := B_{opt} \cup i$, $Z := Z + b_i$, $D := D - a_i$, $I_{opt} := Z + D$ and $\delta := C - Z$.
   4.4. If $i < n$, then $i := i + 1$ and go to Step 4.3.
5. The solution is: B_{opt}, $A_{opt} := G \backslash B_{opt}$, I_{opt} and δ. Stop.

It should be mentioned that if at Step 4 of the A_2 algorithm new entities were added to set B (from set A), then the value of I_{opt} was reduced, too. At the same time, because for entities added to set B from set A are not followed the requirements of rule (7), it may be that the δ value increases. So, when the A_2 algorithm is applied, the value of I_{opt} can be reduced, and that of δ can increase (but rarely) compared to those obtained by the A_1 algorithm.

In order to comparatively analyze the A_1 and A_2 algorithms, for the latter were performed calculations at same initial data as for the A_1 one. Some of the obtained results are shown in Figure 3.

Figure 3. Dependence of δ_2 on n and d at algorithm A_2.

Unlike the case of algorithm A_1, the deviation of solutions obtained when applying the A_2 algorithm significantly depends on the value of the number n of entities, especially at relatively small its values (for example, $n \leq 30$). At the same time, regardless of the value of n, it decreases on d at small their values and increases on d at large their values. At specified above alternatives of initial data, the maximum δ_2 value of 6.21% is at $\{n = 10, d = 5\}$, and the minimum δ_2 value of 0.06% is at $\{n = 100, d = 20\}$ and $\{n = 100, d = 25\}$.

Comparing data in Figures 2 and 3, it can be seen that $\delta_2 < \delta_1$. In more details, the quantitative comparative

https://doi.org/10.52326/ic-ecco.2021/SEC.01

analysis of δ_1 and δ_2 deviations can be done based on $m = \delta_1/\delta_2$ ratio, some values of which are shown in Figure 4.

According to Figure 4, the value of m is decreasing on d and is increasing on n. The maximum m value of 44.22 times is at $\{n = 100, d = 2\}$, and the minimum m value of 1.26 times is at $\{n = 10, d = 5\}$. Thus, in majority of cases, the δ_2 value is considerably smaller than the δ_1 one.

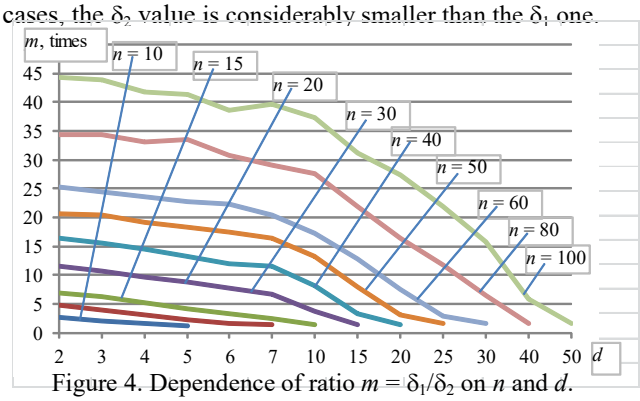

Figure 4. Dependence of ratio $m = \delta_1/\delta_2$ on n and d.

VI. An Improved Approach to the Problem Solving

Finally, if $C - \sum_{i=1}^{k} b_i > \varepsilon$, it may be reasonable, in the meaning of criterion (1), to change by places some entities of set B with some entities of set A. If to consider all possible such alternatives, it can be obtained the optimal solution. But at relatively large values of n, many calculations may be necessary. That is why in this purpose are used, as mentioned above, special well-known methods/computer applications for solving problems of Boolean mathematical programming. At the same time, because of $\varepsilon > 0$, it can be sufficient to consider only few of such alternatives.

The A_3 algorithm for roughly solving the problem {(1), (3), (8)} is described below. It checks the opportunity to replace a number of up to N, preponderantly last, entities of set B by some, preponderantly first, entities of set A. The accuracy of its solution does not depend on ε, but on the N value: the higher the value of N, the more accurate is the solution.

1. Initial data: C; N; a_i, b_i, $i \in G$; $n = |G|$, where $a_i > b_i$ and $b_i \le C$, $i \in G$.
2. $Z := \sum_{i \in G} b_i$. If $Z \le C$, then $B_{opt} := G$, $I_{opt} := Z$, $Y := Z$, $\delta := 0$ and go to Step 14.
3. Determining the primary solution according to the algorithm A_2 of Section V.
   3.1. Ordering and renumbering of entities $i \in G$ according to the rule (7).
   3.2. $k := \max\{i \mid Z \le C\}$, where $Z = \sum_{j=1}^{i} b_j$ (the first part of restriction (8) is followed). $Z := \sum_{j=1}^{k} b_j$, $B := \{1, 2, \ldots, k\}$, $D := \sum_{i=k+1}^{n} a_i$ and $I := Z + D$. $I_{opt} := I$, $B_{opt} := B$ and $\delta := C - Z$.
   3.3. If $k \ge n - 1$, then go to Step 4.
   3.4. $i := k + 2$, $Z_1 := Z$ and $D_1 := D$.
   3.5. If $b_i \le \delta$, then $B_{opt} := B_{opt} \cup i$, $Z_1 := Z_1 + b_i$, $D_1 := D_1 - a_i$ and $I_{opt} := Z_1 + D_1$ and $\delta := C - Z_1$.
   3.6. If $i < n$, then $i := i + 1$ and go to Step 3.5.
4. $N := \min\{N, k\}$. Gradual replacement of up to N, preponderantly last, entities of set B ($i = \overline{k - N + 1, k}$) with some of the first entities of set A, $k + 1 \le i \le \min\{k + N, n\}$ (Steps 5-12). $t := 1$, where t is the number of entities to be moved from B to A at the current value of v (see Step 5).
5. $v := k$, where v is the index of the first entity from the up to N entities of set B to be moved to set A. $w := 0$, where w is the number of entities already moved from B to A at the current value of v.
6. $E := B$, $X := Z$ and $R := D$. $l := v$, where l is the index of the current entity to be moved from B to A.
7. $E := E \setminus l$, $X := X - b_l$, $R := R + a_l$ and $w := w + 1$.
8. Identifying the entities from set A to replace the entity l in set E. $g := 0$, where g is a constant that specifies the difference between r and the index of the second entity (after the r one) to be moved from A to B. $h := 1$, where h is a variable that takes the value 0 or 1.
   8.1. $r := k + 1$, where r is the index of the first entity from those of set A to be moved to set B.
   8.2. $u := r$, where u is the index of the current entity to be moved from A to B. $H := E$, $Y := X$ and $V := R$.
   8.3. Calculations for the current set H. $Y := Y + b_u$. If $C - Y < 0$ (the first part of (8) is not followed), then go to Step 8.6 to modify the value of r.
   8.4. The first part of restriction (8) is followed. $H := H \cup u$, $V := V - a_u$ and $I := Y + V$. If $I < I_{opt}$, then $I_{opt} := I$, $B_{opt} := H$ and $\delta := C - Y$.
   8.5. If $u < \min\{n, k + N\} - gh$, then $u := u + gh + 1$, $h := 0$ and go to Step 8.2.
   8.6. If $r < \min\{n, k + N\} - gh$, then $r := r + 1$, $h := 1$ and go to Step 8.2.
   8.7. If $g < \min\{N, n - k\} - 2$, then $g := g + 1$, $h := 1$ and go to Step 8.1.
9. Moving the next entity from E to A. If $l = k - N + 1$, then go to Step 11 to modify the value of v.
10. If $w < t$, then $l := l - 1$ and go to Step 7.
11. If $v > \max\{1, k - N + 1\}$, then $v := v - 1$ and go to Step 6.
12. If $t < N$, then $t := t + 1$ and go to Step 5.
13. The solution is: B_{opt}, $A_{opt} := G \setminus B_{opt}$, I_{opt} and δ. Stop.

Some results of calculations by the A_3 algorithm using the SIMSEC application are shown in Figure 5. Initial data are the same as in calculations by the A_1 and A_2 algorithms and, additionally, $N = 10$.

Data in Figure 5 show that for some values of d the dependence of the δ deviation on N is decreasing, and for

https://doi.org/10.52326/ic-ecco.2021/SEC.01

the others - is increasing for small and is decreasing for large values of N. Taken into account that $2 \le d \le n/2$ and based on these and other calculations, it is identified that the dependence in question is decreasing at $10 \le n \le 15$, $\{n = 20, d \ge 4\}$, $\{n = 30, d \ge 10\}$, $\{n = 40, d \ge 15\}$, $\{n = 50, d \ge 20\}$, $\{n = 60, d \ge 25\}$, $\{n = 80, d \ge 30\}$ and $\{n = 100, d \ge 40\}$, and is increasing at $\{n = 20, d \le 3\}$, $\{n = 30, d \le 7\}$, $\{n = 40, d \le 10\}$, $\{n = 50, d \le 15\}$, $\{n = 60, d \le 20\}$, $\{n = 80, d \le 25\}$ and $\{n = 100, d \le 30\}$. At the same time, the dependence of costs I on N are always decreasing, regardless of the value of d.

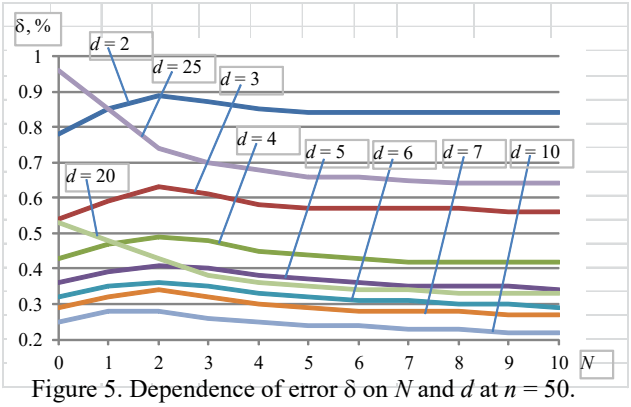

Figure 5. Dependence of error δ on N and d at $n = 50$.

VII. More Detailed Formulation of the Problem

The problem {(1)-(3)}, as well as the {(4), (5)} one not always is convenient. In practice, there are many cases when the i-security measures are opportune to be applied to entities gradually. Such an approach is closer to reality, more flexible and, at the same time, allows more efficient use of financial resources. In this case, it will be considered that each entity $i \in G$, $| G | = n$, is characterized by:

- a_{ij} – annual losses caused to entity i ($i = \overline{1,n}$) if within it there is implemented the i-security measure j ($j = \overline{0,m_i}$). If $j = 0$, that is if there is no implemented any i-security measures, one has annual losses a_{i0};
- b_{ij} – annual costs with the implementation and maintenance within entity i of i-security measure j (i-security activity j). Evidently, $b_{i0} = 0$;
- x_{ij} – a Boolean variable that takes the value 1, if the i-security measure j is implemented within the entity i and the value 0 – otherwise;
- C – financial resources available for the implementation and maintenance of i-security measures within the n entities.

As in Case 1 (see Sections II-VI), the entities for which $a_{i0} - a_{ij} \le b_{ij}$ ($c_{ij} = a_{i0} - a_{ij} - b_{ij} \le 0$) and also those for which $b_{ij} > C$ is not rational to be included in set G. So, one has $a_{i0} - a_{ij} > b_{ij}$ and $b_{ij} \le C$, $i = \overline{1,n}$, $j = \overline{1,m_i}$.

It is necessary to determine the i-security measures $j = \overline{0,m_i}$, $i = \overline{1,n}$ (the value of variables x_{ij}, $i = \overline{1,n}$, $j = \overline{0,m_i}$), which implementation within the n entities minimize the summary annual losses and costs with i-security

$$I = \sum_{i \in G} \sum_{j=\overline{0,m_i}} [(1 - x_{ij}) a_{ij} + x_{ij} b_{ij}] \to \min \quad (9)$$

at

$$\sum_{i \in G} \sum_{j=\overline{0,m_i}} x_{ij} b_{ij} \le C. \quad (10)$$

It should be noted that in problem {(9), (10)}, the i-security measure j for entity i ($j\{i,j\}$) may differ from the i-security measure j for entity r ($j\{r,j\}$).

The problem {(9), (10)} is one of Boolean mathematical programming with unknowns x_{ij}, $i = \overline{1,n}$, $j = \overline{0,m_i}$ and can be solved by respective well-known methods/ computer applications. A simplistic approximate approach, similar to that for the problem {(1), (3), (8)}, is described below.

Statement 2. Within entity i, it is preferable to implement the i-security measure $p(i,p)$ than the i-security measure $s(i,s)$, if at equal other conditions occurs $a_{ip} / b_{ip} > a_{is} / b_{is}$, that is

$$p(i,p) \succ s(i,s), \text{ if } a_{ip} / b_{ip} > a_{is} / b_{is}. \quad (11)$$

The relevancy of this statement can be easily shown similarly as that for Statement 1, if to consider a_{ir} instead of a_i and b_{ir} instead of b_i. ■

Statement 3. It is preferable to implement the i-security measure $p(i,p)$ within entity i, than to implement the i-security measure $s(r,s)$ within entity r, if at equal other conditions occurs $a_{ip} / b_{ip} > a_{rs} / b_{rs}$, that is

$$p(i,p) \succ s(r,s), \text{ if } a_{ip} / b_{ip} > a_{rs} / b_{rs}. \quad (12)$$

The relevancy of this statement can be easily shown similarly as for Statement 1, if to consider a_{kj} instead of a_k and b_{kj} instead of b_k. ■

As in Case 1, in order to implement the simplistic approximate approach, the restriction (10) has to be substituted by

$$\sum_{i \in G} \sum_{j=\overline{0,m_i}} x_{ij} b_{ij} \le C \le \sum_{i \in G} \sum_{j=\overline{0,m_i}} x_{ij} b_{ij} + \varepsilon. \quad (13)$$

The A_4 algorithm for roughly solving the problem {(9), (13)} is described below. The accuracy of its solution, as in the case of A_3 algorithm, does not depend on ε but on the value of N: the higher the value of N, the greater the accuracy of the solution.

1. Initial data: C; N; a_{ij}, b_{ij}, $i \in G$, $= \overline{0,m_i}$; $n = | G |$ and $a_{i0} - a_{ij} > b_{ij}$, $b_{ij} \le C$, $i \in G$, $j = \overline{1,m_i}$.
2. $Z := \sum_{i \in G} \sum_{j=\overline{1,m_i}} b_{ij}$. If $Z \le C$, then: $x_{ij} := 1$, $i \in G$, $j = \overline{1,m_i}$; $D := \sum_{i \in G} \sum_{j=\overline{1,m_i}} a_{ij}$ $I_{\text{opt}} := Z + D$, $Y := Z$ and go to Step 15.
3. For each entity $i \in G$, ordering and renumbering the i-security means $j = \overline{1,m_i}$ according to rule (11).
4. Ordering and numbering in a series $K = \{k(i,j)\}$, $k = \overline{1,|K|}\}$, the i-security measures $j = \overline{1,m_i}$ for all entities $i \in G$ according to rule (12). Here $|K| = \sum_{i \in G} m_i$.
5. $k := \max\{s \mid Z \le C\}$, where $Z = \sum_{p=1}^{s} b_p$, where b_p is the value of b_{ij} that corresponds to $p(i,j)$. $Z := \sum_{p=1}^{k} b_p$, $B := \{1, 2, \ldots, k\}$, $D := \sum_{p \in K} a_p$, where a_p is the value of a_{ij} that corresponds to $p(i,j)$ and $I := Z + D$. $I_{\text{opt}} := I$, $B_{\text{opt}} := B$, $N := \min\{N, k\}$.

https://doi.org/10.52326/ic-ecco.2021/SEC.01

6. Gradual replacement of up to N, preponderantly last, i-security measures of set B ($p = \overline{k-N+1,k}$, N) with some of the first i-security measures of set A, $k+1 \le p \le \min\{k+N, n\}$ (Steps 7-14).
7. $v := k$, where v is the index of the first i-security measure from the up to N, preponderantly last, i-security means of set B to be moved to set A. $t := 1$, $w := 0$, where t is the number of i-security measures to be moved from B to A at the current value of v, and w is the number of i-security measures already moved from B to A at the current value of v.
8. $E := B$, $X := Z$ and $R := D$. $l := v$, where l is the index of the current i-security measure to be moved from B to A.
9. $E := E \setminus l$, $X := X - b_l$, $R := R + a_l$ and $w := w + 1$.
10. Identifying the i-security measures from set A to replace the i-security measure l in set E. $r := k + 1$, where r is the index of the first i-security measure from, preponderantly last, i-security measures of set A to be moved to set B. $g := 0$, where g is a constant that specifies the difference between r and the index of the second i-security measure (after the r one) to be moved from A to B. $h := 1$, where h is a variable that takes the value 0 or 1.
    10.1. $u := r$, where u is the index of the current i-security measure to be moved from A to B. $H := E$, $Y := X$ and $V := R$.
    10.2. Calculations for the current set H. $Y := Y + b_u$. If $C - Y < 0$ (the first part of restriction (13) is not followed), then go to Step 10.5 to modify the value of r.
    10.3. The first part of restriction (13) is followed. $H := H \cup u$, $V := V - a_u$ and $I := Y + V$. If $I < I_{opt}$, then $I_{opt} := I$, $B_{opt} := H$, $A_{opt} := K \setminus B_{opt}$ and $\delta := C - Y$.
    10.4. If $u < \min\{n, k+N\} - gh$, then $u := u + gh + 1$, $h := 0$ and go to Step 10.2.
    10.5. If $r < \min\{n, k+N\} - gh$, then $r := r + 1$, $g := g + 1$, $h := 1$ and go to Step 10.1.
11. Moving the next i-security measure from E to A. If $l = k - N + 1$, then go to Step 13 to modify the value of v.
12. If $w < t$, then $l := l - 1$ and go to Step 8.
13. If $v > \max\{1, k - N + 1\}$, then $v := v - 1$, $t := t + 1$ and go to Step 7.
14. If $t < N$, then $t := t + 1$ and go to Step 6.
15. The solution is: B_{opt}, $A_{opt} := K \setminus B_{opt}$, I_{opt} and $\delta := C - Y$. Based on $p(i,j) = \overline{1,|K|}$, determining B_{iopt}, A_{iopt}, $i \in G$; a_{ij}, b_{ij}, $i \in G$, $j = \overline{0,m_i}$; Z_{iopt}, D_{iopt} and $I_{iopt} := Z_{iopt} + D_{iopt}$, $i \in G$. Stop.

It should be noted that the A_4 algorithm has many similarities to the A_3 one. If in the case of algorithm A_3 (and of the respective optimization problem {(1), (3), (8)}) the security object (system of objects) is detailed to the level of entities – a single level, then in the case of algorithm A_4 (and of the respective optimization problem {(9), (13)}) the security object (system of objects) is detailed to the level of entities, and each entity, in their turn, to the level of cybersecurity measures. But in essence, there are no radical differences between them. At the same time, in the case of a large object (system of objects) the use of the A_4 algorithm is more convenient, including in terms of reducing the error of solutions.

VIII. Conclusions

Two cases of i-securing the informatics infrastructure are examined: Case 1 - one level entities (Sections II-VI) and Case 2 - two level entities (Section VII). For each of them, the respective optimization problem is formulated; they can be solved using the computer applications for Boolean mathematical programming. However, the use of such an application may require considerable calculations or it may not be available at the moment.

That is why some approaches for the approximate solution of the problem are also examined. To this end, a rule for prioritizing the explored cybersecurity measures is obtained. Based on this rule, four algorithms (A_1-A_4) for roughly solving the problem are proposed. They are numerated in the order of the reducing of deviation δ. The A_1-A_3 algorithms are for Case 1, and the A_4 algorithm is for Case 2.

For the quantitative estimation of the solution error, the computer simulation was performed using the SIMSEC application developed for this purpose.

The results of calculations confirm the veracity of the proposed algorithms. At the same time, the value of deviation δ_2, obtained when using the A_2 algorithm, usually is considerably smaller than the deviation δ_1, obtained when using the A_1 algorithm. Also, the solution error decreases rapidly with the lowering of the upper limit for the values of quantities b_i, $i = \overline{1,n}$. Thus, the more detailed the cybersecurity measures in initial data, the lower the error of the solution.

References

[1] *Guide to Understanding the Total Impact of Fraud* (February 2020), International Public Sector Fraud Forum, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/866608/2377_The_Impact_of_Fraud_AW__4_.pdf (accessed 04.08.2021).

[2] S. Morgan. Cyberwarfare In The C-Suite. *Cybercrime Magazine*, Nov. 13, 2020.

[3] S. Morgan. Cybersecurity Ventures' 2019 Cybersecurity Market Report. *Cybercrime Magazine*, Jun. 10, 2019.

[4] *International Monetary Fund: World Economic Outlook Database* (October 2020). (https://knoema.com/tbocwag/gdp-forecast-by-country-statistics-from-imf-2021-2025?country=World, accessed 04.08.2021)

[5] *Forecast Analysis: Information Security, Worldwide, 2Q18 Update*. Gartner Research, Sept. 4, 2018 (https://www.gartner.com/en/documents/3889055, accessed 04.08.2021).