DOI 10.15276/imms.v12.no1-2.65
УДК 681.5.015:004.681

Informatics and Mathematical Methods in Simulation
Vol. 12 (2022), No. 1-2, pp. 65-72

МІНІМІЗАЦІЯ ПОМИЛОК ПРИ ФУНКЦІОНУВАННІ У СИСТЕМАХ УПРАВЛІННЯ КІБЕРБЕЗПЕКОЮ

М.В. Капустян[1], В.А. Кудінов[2], В.О.Хорошко[3]

[1]Хмельницький національний університет, вул. Інститутська, 11, Хмельницький, 29000, Україна; email: kapustian.mariia@gmail.com

[2]Національна академія внутрішніх справ, пл. Солом'янська, 1, Київ-ДСП, 03035, Україна; e-mail: kudinov_va@ukr.net

[3]Національний авіаційний університет, просп. Любомира Гузера, 1, Київ, 03058, Україна; e-mail: professor_va@ukr.net

У статті розглянуто властивості векторної цільової функції для мінімізації інтегрованої квадратичної оцінки помилки системи управління кіберзахистом. На відміну від традиційного завдання функції інтегральних квадратичних оцінок (ІКО) тільки в області стійких систем управління, пропонована векторна цільова функція визначається в усьому просторі варійованих параметрів. Запропоновано покроковий підхід встановлення достатньої кількості обмежень для переходу в область визначення ІКО і раціональний механізм його реалізації у вигляді пріоритетної оптимізації векторної цільової функції з використанням операції порівняння її значень. Розроблено алгоритми модифікованих методів Хука-Дживса і Нелдера-Міда для оптимізації векторних функцій. Основна відмінність запропонованих модифікованих методів оптимізації векторних функцій від методів мінімізації скалярних функцій полягає у використанні бінарних операцій порівняння значень векторних функцій. Застосування методів оптимізації векторної цільової функції дозволяє в єдиному обчислювальному процесі перейти з будь-якої початкової точки пошуку допустимих параметрів системи управління кіберзахистом інформаційних ресурсів в область стійкості системи управління і знайти в ній мінімум ІКО.

Ключові слова: інтегральні квадратичні оцінки (ІКО), оптимізація векторної цільової функції, системи управління кіберзахистом, стійкі системи управління.

Вступ

Управління кібербезпекою є невід'ємною частиною управління будь-якою системою чи об'єктом, оскільки для них інформація стала таким самим ресурсом, як матеріальні, фінансові, трудові та інші ресурси. Тому до управління кібербезпекою використовуються процесний, системний та ситуаційний підходи, як і до будь-якої іншої системи управління. Однак, як з'ясувалося, особливості самої інформації та інформаційних технологій потребують зовсім інших методів, засобів і заходів для їх захисту. Отже управління кібербезпекою є тією сферою, необхідність теоретичного та практичного вивчення якої стала очевидною. Перші дослідження в цій області почалися наприкінці 1980-х рр., а до кінця 1990-х рр. з'явилися перші національні й міжнародні стандарти (ISO/IEC 17799). Однак стандарти в області управління кібербезпекою не вирішують усіх проблем. Навпаки, завдання управління кібербезпекою ускладнюються разом із усе більш інтенсивним використанням інформаційних комп'ютерних технологій практично в усіх сферах людської діяльності. Це обумовлює необхідність узагальнення досвіду управління кібербезпекою, розробки теорії, вирішення практичних аспектів управління кібербезпекою.

Необхідно відзначити й те, що лише кілька років назад, говорячи про управління кібербезпекою, мали на увазі лише рівень великої організації. Останнім часом усе частіше мова йде про управління безпекою регіональних і національних інформаційних інфраструктур, усе частіше виникає потреба в забезпеченні та управлінні кібербезпекою на середніх, малих підприємствах та в некомерційних організаціях.

Таким чином, забезпечення інформаційної безпеки є необхідною умовою функціонування будь-якої компанії, а створення політики безпеки - одна з перших вимог до організації інформаційної безпеки підприємства.

У зв'язку з розширенням сфери використання інформаційних систем й їхнього ускладнення, проблема забезпечення інформаційної безпеки загострюється. Безпеку вже неможливо забезпечити одним лише набором технічних засобів і підтримувати тільки силами підрозділу безпеки. Відсутність регулярної оцінки інформаційних ризиків, недостатня інформованість співробітників про правила роботи із інформацією, що захищається, і дотриманні режиму інформаційної безпеки, відсутність формалізованої класифікації інформації зі ступеня її критичності й уявлень про те, скільки коштують інформаційні активи, - все це може звести нанівець зусилля компанії по забезпеченню інформаційної безпеки. З підвищенням ролі інформаційних систем у підтримці основних бізнесів-процесів компанії до цих проблем додаються питання забезпечення безперервності функціонування бізнесу в критичних ситуаціях. Ігнорування проблем інформаційної безпеки, практика "латанія дір" сьогодні можуть обійтися компанії дуже дорого. Вирішити ці питання можна шляхом побудови ефективної системи управління кібербезпекою (СУКБ).

Завданнями СУКБ є систематизація процесів забезпечення ІБ, розміщення пріоритетів компанії в області ІБ, досягнення адекватності системи інформаційної безпеки існуючим ризикам, досягнення її "прозорості". Останнє особливо важливо, тому що дозволяє чітко визначити, як взаємозалежні процеси й підсистеми інформаційної безпеки, хто за них відповідає, які фінансові й людські ресурси необхідні для їхнього забезпечення й т.д. Створення СУКБ дозволяє також забезпечити ефективне відстеження змін, внесених у систему інформаційної безпеки, відслідковувати процеси вироблення та виконання політики безпеки, ефективно управляти системою в критичних ситуаціях.

У цілому, процес управління безпекою відповідає за планування, виконання, контроль і технічне обслуговування всієї інфраструктури безпеки. Організація цього процесу ускладнюється також тією обставиною, що забезпечення інформаційної безпеки пов'язане з виникненням помилок як не навмисних, так і навмисних, які впливають на захищеність інформаційних ресурсів. Тому необхідно оцінювати ризики, які виникають при експлуатації систем управління та мінімізувати їх.

У світовій практиці є розроблені моделі систем управління ІБ, наприклад, "Information Security Management Maturity Model" (ISM3 від ISECOM) або "The Systems Security Engineering Capability Maturity Model" або стандарт NIST SP800-33. Існує також ряд міжнародних стандартів. Однак пряме використання цих моделей і стандартів утруднене. Вони або занадто конкретизовані, а в будь-якій організації, як правило, уже існує певна система процесів, ролей, організаційно-розпорядницьких документів інформаційної безпекою, які необхідно інтегрувати в систему управління кібербезпекою. При цьому не визначаються пріоритети, так званої "ваги директив", які звичайно застосовуються в стандартах аудита. Або, навпаки, рекомендації носять занадто загальний характер. Наприклад, стандарти містять або набір контрольних директив, або загальний підхід до систем менеджменту, тобто визначають, що потрібно зробити, але не визначають, як це зробити.

Завдання визначення оптимальних показників якості систем управління кіберзахистом інформаційних ресурсів автоматизованих систем є однією з найважливіших проблем проектування комплексних систем захисту інформації. Це зумовлено складністю подібних систем, наявністю безлічі варійованих параметрів, складністю обчислення показників якості. Такі показники задаються у вигляді інтегральних квадратичних оцінок (ІКО) помилки системи управління захистом [1]. Проблема полягає в тому, що існуючі чисельні методи їх мінімізації такі, як методи Хука-Дживса і Нелдера-Міда [3; 8-10], орієнтовані на єдиний (скалярний) параметр, що впливає на ІКО, тоді як в реальних системах таких параметрів може бути декілька і в сукупності вони утворюють векторний параметр. Здійснювати оптимізацію одночасно за всіма параметрами дуже складно, тому перспективним вважається покроковий підхід [6; 11], коли оптимізація виконується послідовно за параметрами. При цьому повинні неухильно задовольнятися обмеження на стійкість системи шляхом оптимізації векторної цільової функції, алгоритми обчислення якої розроблені в частині 1 статті [1].

Метою роботи є розробка методів покрокового підходу до мінімізації інтегральної квадратичної оцінки помилки системи управління кіберзахистом з областю визначення, обмеженій умовами стійкості.

Основна частина

Нехай передавальна функція системи управління залежить від вектору варійованих параметрів $x \in R^p$:

$$w(x,s) = \frac{\beta(x,s)}{\alpha(x,s)};\ \alpha(x,s) = \sum_{i=0}^{n} \alpha_i(x) s^{n-i};\ \beta(x,s) = \sum_{i=0}^{m} \beta_i(x) s^{m-i};\ \alpha_n = \beta_m. \tag{1}$$

Їй відповідає перехідна функція $y(x,t): y(x,0)=0;\ y(x,\infty)=1$. Похибка управління визначається як зважена сума відхилення $z(x,t) = y(x,\infty) - y(x,t)$ та її похідних $z_t^{(k)}(x,t)$:

$$e(x,t) = \sum_{k=o}^{l} w_k \cdot z_t^{(l-k)}(x,t), \tag{2}$$

де $w_k,\ k=\overline{0,l};\ l \le n-m;\ w_0 = 1$ – вагові коефіцієнти.

Як показано в [11], інтегральна квадратична оцінка (ІКО) похибки управління системою обчислюється за формулою:

$$I(x) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{\delta(x,s)\cdot\delta(x,-s)}{\alpha(x,s)\cdot\alpha(x,-s)}\, ds, \tag{3}$$

де $\delta(x,s) = \left[\alpha(x,s) - \beta(x,s)\cdot w(s)\right]/s;\ w(s) = \sum_{k=0}^{l} w_k \cdot s^{l-k}$.

ІКО визначається в області стійкості $D \subset R^p$, яка з використанням елементів першого стовпця визначника Рауса $\rho_k = \alpha_k^k,\ k = \overline{0,n}$, задається умовами

$$\alpha_i(x) > 0,\ i = \overline{0,n};\ \rho_k(x) > 0,\ k = \overline{2,n-1}. \tag{4}$$

Цим нерівностям відповідають області виконання обмежень

$$\Omega_1 = \left\{x \middle| \alpha_i(x) > 0,\ i = \overline{0,n}\right\};\ \Omega_k = \left\{x \middle| \rho_k(x) > 0\right\},\ k = \overline{2,n-1}. \tag{5}$$

Складемо з них систему квазідопустимих областей $D_1 = \Omega_1,\ D_k = D_{k-1} \bigcap \Omega_k,\ k = \overline{2,n-1}$, з яких за допомогою різниць множин побудуємо області рівнів обмежень

$$H_0 = R^p \setminus D_1;\ H_k = D_k \setminus D_{k+1},\ k = \overline{1,n-1};\ H_{n-1} = D_{n-1}. \tag{6}$$

Ступінь порушення першої групи нерівностей (відповідно до [1]) зобразимо штрафною функцією

$$P(x) = \sum_{i=0}^{n} \max\left\{-\alpha_i(x), 0\right\}. \tag{7}$$

Для оцінки ступеня порушення усіх умов стійкості системи управління в [11] сформована векторна цільова функція:

$$F(x) = \begin{cases} (0; P(x)), & x \in H_0; \\ (k; -\rho_{k+1}(x)), & x \in H_k,\ k = \overline{1,n-2}; \\ (n-1; I(x)), & x \in H_{n-1}; \end{cases} \tag{8}$$

і розроблений алгоритм її обчислення. Властивості цієї функції зумовлені властивостями областей D_k, які утворюють систему вкладених областей: $D_{n-1} \subseteq D_{n-2} \subseteq \ldots \subseteq D_1 \subseteq R^p$. Оскільки $D_k \subset \Omega_j$, $j = \overline{1,k}$, то чим більше значення індексу квазідопустимої області k, тим більше виконано в ній умов (обґрунтовані у [1]). У області D_{n-1} мають бути виконані усі умови стійкості. Області рівнів обмежень (відповідно до [1]) мають властивості $H_k \subset D_k, k = \overline{1, n-1}; H_j \bigcap H_k = \varnothing, j \neq k; D = H_{n-1}$, тому області рівнів ділять увесь простір варійованих параметрів на n областей: $R^p = H_0 \bigcup H_1 \bigcup \ldots \bigcup H_{n-1}$.

Перша складова (8) – функція рівня – відображає приналежність її аргументів певної області рівня і являє собою лічильник числа виконаних обмежень, вона кусочно-постійна і може набувати значень від 0 до $n-1$. Друга складова – функція штрафу – представляє поза області стійкості штрафну функцію порушеного обмеження. У області стійкості ця складова співпадає з ІКО. Векторна функція (8) враховує особливості завдання оптимізації – ієрархію обмежень в процесі обчислення ІКО.

Розробимо покроковий підхід мінімізації ІКО за допомогою функції (8), яка враховує умови стійкості (4). Нехай $x \in H_0$, тоді в силу (6) $\exists i \in \left\{\overline{0,n}\right\}, \alpha_i(x) \leq 0$. У області H_0 мірою цього порушення обмеження є функція штрафу (42), яка убуває у напрямі межі наступної області H_1, в якій має місце нерівність $\rho_2(x) \leq 0$. Мірою цього порушення обмеження служить функція штрафу $-\rho_2(x)$, що убуває у напрямі межі області H_2 і т.д. У будь-якій непорожній області рівня $H_k, k = \overline{1, n-2}$ мірою порушення обмеження є відповідна функція штрафу $-\rho_{k+1}(x)$. Крок переходу з області H_k в область H_{k+1} полягає в мінімізації функції штрафу (8) в області H_k. При такій мінімізації чергове обмеження задовольняється, функція (8) зростає і починається наступний крок методу. Враховуючи обмежену кількість областей рівнів, перехід з будь-якої початкової точки в область стійкості виконається не більше ніж за $n-1$ кроків.

Для реалізації покрокового підходу оптимізуватимемо векторну функцію $F(x)$ з урахуванням пріоритету її складових: першу складову, що має пріоритет, необхідно максимізувати, а другу складову – мінімізувати. Завдання оптимізації векторної цільової функції (8) з урахуванням пріоритету її складових позначимо $vecopt\ F(x)$. Для визначення кращих значень векторної функції застосуємо операцію порівняння її значень. Оскільки першу проекцію векторної функції (8), що має пріоритет, необхідно збільшувати, а другу – зменшувати, то два значення векторної цільової функції $U = (U_1, U_2)$ і $V = (V_1, V_2)$ порівняємо бінарною операцією «краще» $\prec$:

$$U \prec V = \begin{cases} 1, & якщо\ (U_1 > V_1) \vee (U_1 = V_1) \wedge (U_2 < V) \\ 0, & якщо\ (U_1 < V_1) \vee (U_1 = V_1) \wedge (U_2 \geq V) \end{cases} \quad (9)$$

Сенс цієї операції полягає в тому, що при різних перших складових векторної функції (8) в двох точках кращою вважається та, в якій перша складова, що має пріоритет, набуває більшого значення. При рівних значеннях першої складової кращою вважається та точка, в якій друга складова набуває меншого значення. Операцію (9) можна реалізувати за наступним алгоритмом.

Алгоритм 1. Порівняння значень векторної функції. Вхідні параметри: U, V – значення векторної функції. Вихідний параметр: B – бульовий результат порівняння.

Крок 1. Покласти $B := 0$.

Крок 2. Якщо $U_1 > V_1$, покласти $B := 1$ і перейти до кроку 5.

Крок 3. Якщо $U_1 < V_1$, перейти до кроку 5.

Крок 4. Якщо $U_2 < V_2$, покласти $B := 1$.

Крок 5. Кінець.

Для кількісної реалізації покрокового підходу модифікуємо прямі методи безумовної мінімізації функцій (методи Хука-Дживса і Нелдера-Міда) [8-10]. Зобразимо алгоритм модифікованого методу Хука-Дживса для оптимізації векторних функцій.

Алгоритм 2. Вхідні параметри: x – початкова точка, ε – допустима погрішність, $F(x)$ – векторна функція, $R(x, F_x)$ – функція досліджуючого пошуку. Вихідні параметри: x, F_x – краща точка пошуку і значення векторної функції в ній.

Крок 1. Покласти $F_x := F(x); \delta := 2$ (δ – крок досліджуючого пошуку).

Крок 2. Якщо $\delta < \varepsilon$, перейти до кроку 10.

Крок 3. Покласти $\delta := \delta/2$.

Крок 4. Покласти $(y, F_y) := R(x, F_x)$.

Крок 5. Якщо $(F_y \prec F_x)$, перейти до кроку 2.

Крок 6. Вирахувати $z := 2y - x; F_z := F(z)$.

Крок 7. Покласти $x := y; F_x := F_y$.

Крок 8. Вирахувати $(y, F_y) := R(z, F_x)$.

Крок 9. Якщо $(F_y \prec F_x)$, перейти до кроку 6, інакше – до кроку 2.

Крок 10. Кінець.

Тут (x, F_x) – краща точка попередньої інтеграції і значення функції в ній, (y, F_y) – краща точка досліджуючого пошуку і значення функції в ній, (z, F_z) – точка пошуку за зразком і значення функції в ній. На кроці 1 обчислюється значення цільової функції в початковій точці та ініціалізується крок досліджуючого пошуку. На кроці 2 перевіряється критерій зупину. Кроки 3-9 реалізують ітераційний цикл методу. На кроці 3 зменшується крок досліджуючого пошуку, а на кроці 4 здійснюється досліджуючий пошук типу 1. На кроці 5 порівнюються значення векторної функції в кінцевій точці досліджуючого пошуку типу 1 і базовій точці. На кроці 6 виконують пошук за зразком. На кроці 7 замінюється базова точка. На кроці 8 здійснюється досліджуючий пошук типу 2. На кроці 9 порівнюються значення векторної функції в кінцевій точці досліджуючого пошуку типу 2 і в новій базовій точці. Модифікований метод Хука-Дживса відрізняється від методу мінімізації скалярної цільової функції кроками 5 і 9, на яких замість операції «менше» застосовується операція «краще» (9). Модифікований досліджуючий пошук, що реалізовує функцію $R(x, F_x)$, представимо алгоритмом.

Алгоритм 3. Вхідні параметри: (x, F_x) – базова точка, δ – крок пошуку, $F(x)$ – векторна функція. Вихідні параметри: (y, F_y) – краща точка пошуку.

Крок 1. Покласти $n := \dim x; y := x; F_y := F_x; u := y; i := 1$.

Крок 2. Вирахувати $u_i := y_i + \delta; F_u := F(u)$.

Крок 3. Якщо $(F_u \prec F_y)$, покласти $y_i := u_i; F_y := F_u$ і перейти до кроку 6.

Крок 4. Вирахувати $u_i := y_i - \delta; F_u := F(u)$.

Крок 5. Якщо $(F_u \prec F_y)$, покласти $y_i := u_i; F_y := F_u$, інакше $- u_i := y_i$.

Крок 6. Якщо $i < n$, покласти $i := i + 1$ і перейти до кроку 2.

Крок 7. Кінець.

У цьому алгоритмі n – розмірність вектору змінних, i – номер змінної. На кроці 1 ініціалізються параметри досліджуючого пошуку. Кроки 2-6 складають ітераційний цикл методу. На кроці 2 виконується досліджуваний пошук в позитивному напрямі. На кроці 3 порівнюються значення векторної функції в поточній точці пошуку і в кращій точці. Після невдалого пошуку в позитивному напрямі кроком 4 виконується досліджуючий пошук в негативному напрямі. На кроці 5 порівнюються значення векторної функції в поточній точці пошуку і в кращій точці. На кроці 6 перевіряється критерій зупину. Операція «краще» для значень векторної функції в цьому алгоритмі застосовується на кроках 3 і 5.

Пошук мінімуму функції методом Нелдера-Міда залежить від результату порівняння значень цільової функції в різних точках пошуку при визначенні кращої або гіршої вершини, розтягуванні або стискуванні багатогранника [6; 8-10]. Для застосування цього методу в оптимізації векторних цільових функцій порівняння значень скалярної цільової функції замінимо порівнянням значень векторних функцій (9). Модифікований метод Нелдера-Міда представимо наступним алгоритмом.

Алгоритм 4. Вхідні параметри: x – початкова точка, $F(x)$ – векторна функція, ε – допустима погрішність. Вихідні параметри: (x, F_x) – краща точка пошуку і значення векторної функції в ній.

Крок 1. Покласти $n := \dim x; m := n+1; \delta := 1; j := 1; P_m := x$.

Крок 2. Вирахувати $F_m := F(x)$.

Крок 3. Покласти $P_j := x; P_{ij} := x_j + \delta$.

Крок 4. Вирахувати $F_j := F(P_j)$.

Крок 5. Якщо $j < n$, покласти $j := j+1$ і перейти до кроку 3.

Крок 6. Якщо $\delta < \varepsilon$, перейти до кроку 29.

Крок 7. Покласти $l := m; F_x := F_m; h := m; F_w := F_m; j := 1$.

Крок 8. Якщо $F_j \prec F_x$, покласти $F_x := F_j; l := j$.

Крок 9. Якщо $F_w \prec F_j$, покласти $F_w := F_j; h := j$.

Крок 10. Якщо $j < n$, покласти $j := j+1$ і перейти до кроку 8.

Крок 11. Покласти $x := P_l; w := P_h$.

Крок 12. Вирахувати $\delta := \max\left|P_{ij} - x_i\right|$.

Крок 13. Вирахувати $c := (\sum_{j=1}^{m} P_j - w)\big/n$.

Крок 14. Вирахувати $y := 2c - w; F_y := F(y)$.

Крок 15. Якщо $F_y \prec F_x$, перейти до кроку 16, інакше – до кроку 19.

Крок 16. Вирахувати $z := 2y - c; F_x := F(z)$.

Крок 17. Якщо $F_z \prec F_y$, покласти $P_h := z; F_h := F_z$, інакше – $P_h := y; F_h := F_y$.

Крок 18. Перейти до кроку 6.

Крок 19. Покласти $j := 1$.

Крок 20. Якщо $j \neq h$ і $F_y \prec F_j$, покласти $P_h := y; F_h := F_y$ і перейти до кроку 6.

Крок 21. Якщо $j < m$, покласти $j := j+1$ і перейти до кроку 20.

Крок 22. Якщо $F_y \prec F_w$, покласти $w := y; F_w := F_y$.

Крок 23. Вирахувати $z := 0,5(w + c); F_z := F(z)$.

Крок 24. Якщо $F_z \prec F_w$, покласти $P_h := z; F_h := F_z$ і перейти до кроку 6.

Крок 25. Покласти $j := 1$.

Крок 26. Якщо $j \neq l$, вирахувати $P_j := 0,5(P_j + x); F_j := F(P_j)$.

Крок 27. Якщо $j < m$, покласти $j := j+1$ і перейти до кроку 26.

Крок 28. Перейти до кроку 26.

Крок 29. Кінець.

У цьому алгоритмі m означає кількість вершин багатогранника; δ – його розмір; P_j, F_j – вершина з номером j і значення векторної функції в ній; x, l, F_x – краща вершина, її індекс і значення векторної функції в ній; w, h, F_w – гірша вершина, її індекс і значення векторної функції в ній; c – центр тяжіння вершин за винятком гіршої; y і F_y – точка віддзеркалення і значення векторної функції в ній; z і F_z – точка розтягування або стискування і значення векторної функції в ній. Кроки 1 і 2 ініціалізують параметри та формують початкову вершину. Кроки 3-5 формують початковий багатогранник. Кроки 6-28 складають ітераційний цикл методу. На кроці 6 перевіряється критерій зупину. Кроки 7-11 визначають кращу і гіршу вершини. Крок 12 обчислює розмір багатогранника. На кроці 13 обчислюється центр вершин багатогранника, за винятком гіршої вершини. На кроках 14-21 виконуються віддзеркалення і розтягування багатогранника, а на кроках 22-28 – стискання і редукція багатогранника.

Висновки

1. У статті розглянуто властивості векторної цільової функції для мінімізації інтегрованої квадратичної оцінки помилки системи управління кіберзахистом. На відміну від традиційного завдання функції ІКО тільки в області стійких систем управління, пропонована векторна цільова функція визначається в усьому просторі варійованих параметрів.

2. Запропоновано покроковий підхід встановлення достатньої кількості обмежень для переходу в область визначення ІКО і раціональний механізм його реалізації у вигляді пріоритетної оптимізації векторної цільової функції з використанням операції порівняння її значень.

3. Розроблено алгоритми модифікованих методів Хука-Дживса і Нелдера-Міда для оптимізації векторних функцій. Основна відмінність запропонованих модифікованих методів оптимізації векторних функцій від методів мінімізації скалярних функцій полягає у використанні бінарних операцій порівняння значень векторних функцій.

4. Застосування методів оптимізації векторної цільової функції дозволяє в єдиному обчислювальному процесі перейти з будь-якої початкової точки пошуку допустимих параметрів системи управління кіберзахистом інформаційних ресурсів в область стійкості системи управління і знайти в ній мінімум ІКО.

Список літератури

1. Капустян М.В., Пархуць Л.Т., Хорошко В.О. Аналіз методів складання оптимальних розкладів роботи складних систем. *Інформація та математичні методи в моделюванні*. 2012. Т.2, №1, С. 467-57.

2. Лапко А.П., Крохов С.В., Ченцов С.И., Фельдман Д.А. Обучающие системы обработки информации и принятия решений. Новосибирск: Наука, 1996. 284 с.

3. Современна теория систем управления / Под ред. К.Т. Леондеса. М: Мир, 2000. 521 с.

4. Fuller A. The replacement of saturation constrains by energy constrains in control optimization theory. *International Journal of Control*. 1997. № 3. P. 201-277.

5. Головкин Б.А. Расчёт характеристик и планирования процессов. М.: Радио и связь, 2003. 274 с.

6. Кудинов В.А., Пархуць Л.Т., Хорошко В. А. Методика системного проектирования корпоративных сетей. *Вісник ДУІКТ*. 2005.Т.3. №3-4. С. 184-186.

7. Острем К. Ю. Введение в стохастическую теорию управления. М.: Мир, 1973. 322 с.

8. Химмельблау Д. Прикладное нелинейное программирование. М.: Наука, 2005. 536 с.
9. Хеллман О. Введение в теорию оптимального поиска. М.: Радио и связь, 2003. 284 с.
10. Хоменюк В. Б. Элементы теории многоцелевой оптимизации. М.: Наука, 1983. 254 с.
11. Майника Э. Алгоритмы оптимизации на сетях и графах. М.: Мир, 1981. 324 с.

MINIMIZATION OF ERRORS DURING FUNCTIONING IN CYBER SECURITY MANAGEMENT SYSTEMS

M.V. Kapustian[1], V.A. Kudinov[2], V.O. Khoroshko[3]

[1]Khmelnytskyi National University, Instytutska Str., 11, Khmelnytskyi, 29000, Ukraine; email: kapustian.mariia@gmail.com
[2]National Aviation University, Lubomir Huser Ave., 1, Kyiv, 03058, Ukraine; e-mail: professor_va@ukr.net
[3]National Academy of Internal Affairs, Solomyanska Sq., 1, Kyiv-DSP, 03035, Ukraine; e-mail: kudinov_va@ukr.net

The article considers the properties of a vector objective function to minimize the integrated quadratic error estimation of the cybersecurity management system. In contrast to the traditional task of the integral quadratic estimation (IQE) function only in the field of stable control systems, the proposed vector objective function is determined in the whole space of variable parameters. A step-by-step approach of establishing a sufficient number of constraints for the transition to the scope of IQE and a rational mechanism for its implementation in the form of priority optimization of the vector objective function using the operation of comparing its values.

Algorithms of modified Hooke-Jeeves and Nelder-Mead methods for optimization of vector functions are developed. The main difference between the proposed modified methods for optimizing vector functions from methods for minimizing scalar functions is the use of binary operations to compare the values of vector functions. The use of methods for optimizing the vector objective function allows in a single computational process to move from any starting point of search for acceptable parameters of the cybersecurity management system of information resources to the area of management system stability and find a minimum of IQE.

Keywords: integral quadratic estimation (IQE), vector objective function optimization, cybersecurity management system, stable control systems.