

УДК 004.942

О.О. КРЯЖИЧ, О.В. КОВАЛЕНКО

ЗАБЕЗПЕЧЕННЯ АЛГОРИТМІЧНОГО ТА РЕСУРСНОГО БАЛАНСУ ПРИ ВИРІШЕННІ СИТУАЦІЙНИХ ЗАДАЧ

***Анотація.** В роботі досліджена необхідність оптимізації програмних і апаратних засобів для забезпечення життєздатності інформаційних технологій. Наведено математичне обґрунтування балансового методу оптимізації. Методика оптимізації розглянута як для апаратних засобів, що задіяні в системі виконання одного завдання, так і на прикладі програми на мові Python.*

***Ключові слова:** життєздатність, оптимізація, алгоритм, баланс, час.*

Вступ

Важливе місце для підвищення життєздатності інформаційних технологій займають методи оптимізації їх програмних, апаратних і інформаційних засобів. Таке підвищення життєздатності може бути отримане на основі використання принципу балансу. Цей принцип є подальшим розвитком принципу узгодження пропускних здібностей різних частин системи, приведеного в роботі В. М. Глушкова [1], і принципу оптимізації обчислювальних засобів за критерієм частоти їх використання, приведеного в роботі Майерса [2]. Досягнення балансу складових частин інформаційної технології дозволяє не лише підвищити її продуктивність, але і раціональніше використовувати ресурси системи.

За сучасного розвитку засобів обчислювальної техніки питання раціонального використання ресурсів системи повинно було б відійти на другий план. Проте черговість виконання завдань при використанні кластерних обчислень, виконання операцій за допомогою хмарних технологій, використання Wi-Fi для роботи на відстані, підключення віддалених експертів та спеціалістів, що працюють на різній за характеристиками та технологіями обчислювальної техніці для вирішення окремих задач знову підняли питання оптимізації.

Актуальність досліджуваної теми підтверджується також необхідністю запуску окремих, іноді – різноспрямованих алгоритмів, для вирішення задач ситуаційного управління. Так, наприклад, учасники антитерористичної операції потребували планшетів, обладнаних програмами, що дозволяють з одного боку отримувати дані з квадрокоптерів, знімки дистанційного зондування Землі, обробляти їх, а з іншого – передавати оброблені дані іншим фахівцям і службам з певним рівнем надійності до захисту даних та гарантованою можливістю зчитування цих даних на техніці, що є у тих фахівців. Такі ж самі задачі стоять і перед фахівцями Державної служби України з надзвичайних ситуацій, спеціалістів медицини катастроф та багатьох інших.

Мета роботи – розглянути можливості забезпечення алгоритмічного та ресурсного балансу при вирішенні ситуаційних задач.

Завдання роботи:

- дослідити основну проблематику та особливості оптимізації алгоритмів та обчислювальних ресурсів для вирішення задач ситуаційного управління;
- представити балансовий метод як варіант оптимізації алгоритмів та наявних ресурсів для вирішення різноспрямованих задач;
- розглянути спрощений приклад, що демонструє роботу представленої методики.

Питання оптимізації алгоритмів та апаратних засобів не є новими – вони активно досліджуються як окремими вченими, так і організаціями, наприклад, Інститутом кібернетики ім. В.М. Глушкова НАН України, з проведенням конференцій, симпозіумів, шкіл. Серед найбільш відомих вітчизняних дослідників за зазначеною тематикою можна назвати В.М. Глушкова [1], І.В. Сергієнка [3], закордонних – А. Ахо, Дж. Хопкрофта [4]. Активно розглядаються питання оптимізації алгоритмів в роботах В.Є. Снитюка [5], І.В. Стеценко [6], Г.С. Теслера [7] та інш.

Слід зазначити, що питання оптимізації алгоритмів при вирішенні ситуаційних задач з метою адаптації алгоритмів для використання на різних платформах та техніці з різними характеристиками, раніше змістовно не розглядалося.

1. Проблематика дослідження та особливості оптимізації

Можливість підвищення життєздатності [8] інформаційної технології ґрунтується на тому, що частота використання команд, даних, адрес, операційних пристроїв, каналів зв'язку, місткостей пам'яті і т. д. розподілена по діапазону можливих значень у край нерівномірно. Тому реальна життєздатність інформаційної технології визначається найбільш вузьким місцем в системі, тобто, може бути описана за законом Амдала. Одним з підходом для виявлення «вузьких місць» в інформаційній технології та здійснення на цій основі оптимізації технічних, програмних і інших засобів являється використання критерію простого функціонального балансування різними ресурсами. Виявлення вузьких місць в системі означає знаходження ресурсів, обмеженість яких істотно впливає на ефективне використання інформаційної технології.

Ідейні викладки подібної оптимізації наведені ще фон Нейманом, який висловив наступне положення [9]: «Ми хотіли б ввести до машини у вигляді електронних схем тільки такі логічні структури, які або потрібні для функціонування повноцінної системи, або дуже зручні, оскільки часто використовуються». Для успішного виконання такої оптимізації потрібна детальна інформація про досліджувані процеси і параметри. Ця інформація може бути отримана двома способами: статичними або динамічними вимірами частоти цієї інформації, що найбільше зустрічається серед досліджуваних параметрів.

Результат аналізу цієї інформації дозволяє:

- перепланувати проходження завдання (для кращого завантаження наявних ресурсів, для керування черговістю вирішення задач при кластерних обчисленнях);

- визначити найбільш «вузькі місця» в інформаційній технології (для визначення її життєздатності, балансування за критеріями продуктивності і ресурсозберігання);
- визначити життєздатність інформаційної технології при різних варіантах конфігурації системи;
- оптимізувати конфігурацію обчислювальних засобів для конкретного застосування окремої інформаційної технології;
- визначити міру сумісності двох і більше процесів;
- організувати ефективне функціонування паралельно працюючих процесів;
- удосконалити існуючу систему команд шляхом введення нових команд і виключення рідко використовуваних;
- оптимізувати структуру полів команд (кодів операції, адрес, посилань на операнди, даних);
- оптимізувати архітектуру системи під певну інформаційну технологію;
- оптимізувати пропускну здібності каналів зв'язку;
- оптимізувати надійність компонентів обчислювальної системи як єдиного цілого;
- оптимізувати кількість рівнів, об'єм і час обміну між різними рівнями і видами пам'яті;
- оптимізувати пропускну спроможність каналів зв'язку усередині і поза обчислювальною системою;
- оптимізувати організацію обчислювальних процесів.

Необхідно пам'ятати, що існує сильна кореляція між статичними і динамічними частотними характеристиками програм [9]. Це дозволяє вибрати в якості основи оптимізації цих статичних або динамічних вимірів і отримати при цьому вигоду відразу за двома критеріями – часовою або алгоритмічною складністю, наприклад, часу виконання програм і завантаженням наявних ресурсів, на яких використовують інформаційну технологію. Скорочення часу виконання програм і об'єму займаної ними пам'яті на різних обчислювальних пристроях з різними операційними системами (персональному комп'ютері, ноутбукі або планшеті, що працюють відповідно на Linux, Windows та Android) можна добитися за рахунок введення в систему нових спеціальних команд для виконання функцій, що часто зустрічаються.

Ще одним підходом до підвищення життєздатності інформаційної технології є використання кодів змінної довжини, обернено пропорційній частоті кодів, що використовуються найчастіше (коди операції і інших параметрів команди). Такий підхід використовується в алгоритмах стискування інформації і використання змінного формату команд.

Критерій балансування ресурсами широко використовується в економіці і управлінні підприємствами [10] і виходить з положення, що доцільно усувати «вузькі місця» шляхом перерозподілу ресурсів. Так, виходячи з економічних і інших чинників, немає сенсу домагатися пікової продуктивності окремих груп пристроїв і пропускну спроможності каналів, якщо середні показники значно менше цього рівня. В цьому випадку більше виправдано здійснити перерозподіл ресурсів з метою підвищення не пікової, а середньої продуктивності системи на заданому класі завдань, що у підсумку забезпечить певний рівень надійності виконання завдань та життєздатність

інформаційної технології при значних коливаннях навантаження на пристрої та канали зв'язку.

Зазначені питання постають у наступній ситуації. Припустимо, відбувається вирішення надзвичайної ситуації, що виникла на небезпечному об'єкті. Задіяні різні за підпорядкуванням сили рятувальників, що оснащені різноманітною за характеристикам комп'ютерною та телекомунікаційною технікою. Координація відбувається з оперативного штабу, який оснащено сучасною технікою з використанням інформаційної технології для ситуаційного рішення поставлених задач. Оперативна система обчислювальної техніки штабу – Windows 7.0. Декілька віддалених фахівців, що забезпечують роботи баз даних, працюють під операційною системою Linux за допомогою швидкісного Інтернету. Оперативні групи рятувальників, командири яких оснащені бюджетними планшетами з системою Android, версіями 2.0 – 4.0, повинні отримувати оперативну інформацію через Wi-Fi про загальний стан небезпечного об'єкта, а також звертатися до баз даних для коригування власних дій в межах своєї, більш вузької задачі. Тож можна з високою долею вірогідності стверджувати, що один і той же алгоритм вирішення окремої задачі може виконуватися декілька мілісекунд на техніці, що розташована у штабі, і декілька хвилин, а то й десятків хвилин, на техніці командирів оперативних груп, що є неприпустимим при координації рятувальної операції у надзвичайних умовах. У зв'язку з зазначеним можна припустити, що одночасно слід не лише оптимізувати алгоритми виконання окремих задач, а й збалансувати взаємодію різної техніки за критерієм продуктивності.

2. Балансовий метод оптимізації

Розглянемо приклад використання критерію балансу при програмно-апаратній оптимізації для забезпечення життєздатності інформаційних технологій за умов максималізації навантаження на апаратну частину та засоби комунікацій у обмежений період часу.

Для цього введемо в розгляд оцінку продуктивності залежно від складності обчислень і числа операцій.

Позначимо через v_1, v_2, \dots, v_n – швидкодію операційних пристроїв, що реалізують різні види операцій $1, 2, \dots, n$, а через g_1, g_2, \dots, g_n – складність обчислень, що характеризується числом операцій типу $1, 2, \dots, n$, що виконуються при рішенні задачі. При цьому передбачається, що пристрої можуть працювати паралельно в часі. Тоді верхня оцінка максимальної продуктивності обчислювальних засобів системи буде визначено співвідношенням:

$$\lambda \leq \min(v_1/g_1, v_2/g_2, \dots, v_n/g_n). \quad (1)$$

Таким чином, найменш продуктивна на заданому класі завдань група v_i/g_i за законом Амдала і визначає реальну продуктивність всього програмно-апаратного комплексу, що задіяний під вирішення задачі. Виникає завдання оптимізації, яке ґрунтується на критерії балансу:

$$v_1/g_1 = v_2/g_2 = \dots = v_n/g_n, \quad (2)$$

або

$$\varphi_1(v_1/g_1) = \varphi_2(v_2/g_2) = \dots = \varphi_n(v_n/g_n). \quad (3)$$

Завдання оптимізації для алгоритму може бути вирішене за допомогою критерію балансу з використанням частот тих команд, що зустрічаються найчастіше.

Відповідно до операційного методу дослідження систем, усі вирішувани на обчислювальних машинах завдання розділяються на класи залежно від набору використовуваних операцій і частотної функції розподілу цих операцій [11]. Для визначення частотних функцій розподілу операцій цього класу завдань слід розглянути наступні вектори: типових операцій, характеристичний, частотний, еталонний частотний і еталонний часовий.

Визначимо ефективність обчислювального процесу при рішенні заданого класу завдань у вигляді наступної моделі:

$$B = \sum_{i=1}^n f_i t_i = \sum_{i=1}^n b_i, \quad (4)$$

де n – кількість досліджуваних алгоритмів;

t_i – середній час проходження i -го алгоритму в системі з декількох обчислювальних засобів;

f_i – частота зустрічі i -го алгоритму при вирішенні заданого класу завдань;

b_i – оцінка вкладу i -го алгоритму в ефективність обчислювального процесу.

З виразу (4) випливає $f_i = B/t_i$, тобто за інших рівних умов для підвищення ефективності обчислювального процесу в системі з декількох обчислювальних засобів слід прагнути до зменшення часу проходження алгоритму, який має максимальну частоту задіяння у процесі обчислень.

Але це вірно в основному для спеціалізованих інформаційних технологій, які націлені на вирішення конкретного класу завдань. У загальному випадку відповідно до принципу балансу слід ввести міру ефективності вибору і реалізації i -го алгоритму:

$$\mu_i = f_i t_i - \frac{1}{n} \sum_{i=1}^n f_i t_i = f_i t_i - B/n. \quad (5)$$

Величина μ_i показує вплив i -го алгоритму на середню ефективність системи обчислювальних та опосередковано характеризує життєздатність інформаційної технології в цілому. Тож у цьому випадку для детального

опису життєздатності інформаційної технології доцільно разом з μ_i дослідити міру «рівномірності» ефективності обчислювального процесу на конкретному класі завдань, яка визначається величиною середнього квадратичного відхилення (зміщеного або незміщеного):

$$\sigma^n = \left(\sum_{i=1}^n (f_i t_i - B/n)^2 / n \right)^{1/2} = \left(\sum_{i=1}^n (f_i t_i - B/n)^2 / (n-1) \right)^{1/2} \quad (6)$$

разом з коефіцієнтом «рівномірності» ефективності $\psi = \sigma n / B$.

На основі цих критеріїв можна порівнювати як окремі обчислювальні засоби, так і оцінювати їх самостійно за критерієм проходження конкретного алгоритму при вирішенні задач на цих обчислювальних засобах.

Аналогічно наведеному вище можна оптимізувати і інші параметри (ресурси) обчислювальних засобів, що призначені для виконання задач за допомогою певної інформаційної технології.

Приймемо, що $q_i = F(t_i)$ – залежність між апаратними витратами і швидкодією i -го алгоритму. В цьому випадку похідна $\partial q_i / \partial t_i$ визначає крутизну цієї залежності. Введемо в розгляд коефіцієнт вирівнювання ефективності обчислювального засобу для i -го алгоритму (блоку алгоритму):

$$\theta_i = \mu_i / \frac{\partial q_i}{\partial t_i}. \quad (7)$$

Якщо функція $q_i = F(t_i)$ розподілена за експоненціальним законом, то

$$\theta = \mu_i e^{-\lambda_i t_i} / \lambda_i Q_i, \quad (8)$$

де Q_i – константа витрат, тобто, $q_i = Q_i e^{-\lambda_i t_i}$, оптимізаційний процес доцільно почати з максимального Q_i .

Розглянемо процес оптимізації групи стандартних алгоритмів. Серед цієї групи можна виділити базисні операції – алгоритми, які не містять інших частин, наприклад, арифметичні операції. Можна вважати, що ця група базисних операцій реалізована оптимально і може виступити основою для оптимізації інших алгоритмів за критерієм балансу. Таким чином, маємо два вектори: базисних операцій $O = O(O_1, O_2, \dots, O_n)$ і стандартних алгоритмів $\Phi = \Phi(\Phi_1, \Phi_2, \dots, \Phi_n)$ з частотами тієї складової алгоритму, що зустрічається найчастіше $f_i, i=1, 1+n$.

Тоді внесок групи стандартних алгоритмів в загальну ефективність обчислювального процесу складає

$$B_{ca} = \sum_{i=1}^n f_i t_i, \quad (9)$$

а внесок базисних операцій

$$B_o = \sum_{i=1}^l f_{i+n} t_{i+n}, \quad (10)$$

де t_i – середній час виконання алгоритму або операції.

Згідно з критерієм рівномірної ефективності (балансу) вклади в загальну ефективність усіх алгоритмів мають бути рівні. У представленому випадку $B_{ca}/n = B_o/n$, отже, орієнтовна ефективність для групи стандартних алгоритмів може бути представлена наступним чином:

$$\overline{B_{ca}} = B_o / f_i l, \quad i = \overline{1, n}. \quad (11)$$

Природна вимога задовольняти критерію балансу окремих алгоритмів і усередині групи алгоритмів, що оптимізується, може бути реалізована шляхом задоволення умови

$$\forall (f_i t_i = f_j t_j, \quad i, j = \overline{1, n}) \quad (12)$$

Враховуючи B_{ca} і моделі (4) та (12), можна визначити орієнтовний час виконання стандартних алгоритмів

$$\overline{t_i} = B_o / f_i m, \quad i = \overline{1, n}. \quad (13)$$

З урахуванням часу t_i можна обчислити величини θ_i, μ_i і досягти мінімізації θ_1 , що набуває досить великих значень з метою наближення до мінімального значення коефіцієнта балансу ψ .

Поява від'ємних θ_i показує, що є можливість зменшити швидкість реалізації i -го алгоритму на окремих обчислювальних засобах і забезпечити на цій основі перерозподіл ресурсів для рівного доступу до виконання обчислювальних завдань.

У загальному випадку залежність (5) може бути складнішою

$$\forall (\varphi_i(f_i t_i) = \varphi_j(f_j t_j), \quad i, j = \overline{1, n}) \quad (14)$$

де φ_i – функціональні залежності, що відбивають закономірні зв'язки між групами параметрів алгоритмів, технічних засобів і т. інш.

Для отримання стійких оцінок у виразах (5) і (6) можна використовувати медіанні оцінки. В цьому випадку

$$\overline{\mu_i} = f_i t_i - \text{med} U_i, \quad \sigma_i = \text{med} |U_i - \text{med} U_i| / A \quad (15)$$

де A – константа, що відповідає заданому закону розподілу.

В якості максимальної ефективності системи команд щодо реалізації алгоритмів на різних обчислювальних засобах може служити рівність середньої вірогідності правильного виконання команд для усієї сукупності команд конкретної обраної мови програмування, записане у вигляді:

$$A(P_i = P_j, \overline{i, j} = 1, \overline{r}), \text{ тобто, } A(\lambda_i m_i t_i = \lambda_j m_j t_j, i, j = \overline{1, r}). \quad (16)$$

де t_i – час реалізації i -ої команди; λ_i – інтенсивність відмов окремих обчислювальних засобів, що реалізують i -у команду; m_i – середня частота використання i -ої команди в програмах даного класу завдань; P_i – середня вірогідність безвідмовної роботи обчислювальних засобів, що реалізують i -ту команду; r – потужність системи команд, тобто сукупність типів команд.

Застосування запропонованої методики можна проілюструвати на прикладі розподілу інформаційних навантажень в межах персонального комп'ютеру між рівнями інтерпретації мови програмування.

Питання про організацію структури рівнів пристрою управління і зв'язків між рівнями дуже важливе, оскільки він значною мірою визначає ефективність обчислювального процесу і вартість пристрою управління. Необхідно знайти розумну відповідність між інформаційними об'ємами матриць пам'яті, що реалізують рівні управління, швидкодією цих матриць і частотою звернення до них.

Припустимо, в досліджуваному комп'ютері є n взаємодіючих блоків $S_i, i=1, n$ з часом звернення до них відповідно τ_i . В результаті аналізу роботи цих блоків при проходженні через комп'ютер потоку завдань можна визначити частоти f_i звернення до цих блоків. Тоді загальний ефективний час машини складе:

$$t_{Ei} = \sum_{i=1}^n f_i \tau_i = \sum_{i=1}^n t_{\Delta i}, \quad (17)$$

де t_{Ei} – ефективний час роботи i -го блоку.

Оцінимо вклад кожного блоку i в ефективність обчислювального процесу:

$$P_i = t_{Ei} / t_E = f_i \tau_i / \sum_{j=1}^n f_j \tau_j. \quad (18)$$

Природно, необхідно враховувати швидкодію того блоку, який має максимальну величину P_j , оскільки в цьому випадку витрати вкладені найефективніше. Отже, комп'ютер буде в деякому розумінні якнайкраще

спроєктований при рівних величинах P_i блоків, тобто, коли час звернення до блоків розподілений обернено пропорційно до частоти звернення до них.

Аналіз роботи операторного і мікрокомандного рівнів пристроїв управління обчислювальної техніки, яка широко застосовується в Україні, вказує на те, що частота звернення до матриці пам'яті оперативного рівня (f_0) може бути приблизно в 8 – 10 разів менше частоти звернення до матриці нижнього рівня, т. е. $f_m = 10f_0$. Тоді загальний ефективний час роботи обчислювальної техніки складатиме:

$$t_E = f_0\tau_0 + f_m\tau_m + \Delta t_E, \quad (19)$$

де Δt_E – ефективний час роботи інших блоків машини.

Підставивши $f_m = 10f_0$ в рівняння (19), отримаємо:

$$t_E = f_0\tau_0 + 10f_0\tau_m + \Delta t_E. \quad (20)$$

Реалізована на мові Python програма-приклад з формування рядка-поліндрома показує різницю у часі для виконання завдань навіть за умов використання сучасної обчислювальної техніки (рис. 1).

```

1 import sys
2 ryadok = sys.argv[1]
3 ryadok = ryadok.replace(" ", "")
4 ryadok = ryadok.lower()
5 L = len(ryadok)
6 i = 0
7 for bukva in ryadok:
8     if ryadok[0] == ryadok[L-1]:
9         i = i+1
10 if i == L:
11     print 'YES'
12 else:
13     print 'NOT'
14
15

```

а) час виконання 26 сек.

```

1 import sys
2 s = sys.argv[1]
3 flag = True
4 s_cleaned = s.lower().replace(' ', '')
5 # збережемо довжину рядка(оптимізація)
6 s_length = len(s_cleaned)
7 d == s_cleaned[::-1]
8 for i in range(s_length / 2):
9     if s_cleaned[i] != s_cleaned[s_length-1-i]:
10         flag = False
11
12 if flag:
13     print 'YES'
14 else:
15     print 'NO'

```

б) час виконання 5 сек.

Рисунок 1 – Приклад програми з неоптимізованим (а) та оптимізованим (б) алгоритмом

Тобто, якщо, що в роботі використовується сучасний комп'ютер і час звернення до матриць пам'яті усіх рівнів управління однаковий, то з наведеного рівняння (20) видно, що, зменшивши удвічі час звернення до мікропрограмної матриці, можна підвищити ефективність роботи окремого обчислювального засобу в 5 разів в порівнянні із зміною швидкодії матриці.

Висновки

В роботі наведено один з аспектів забезпечення життєздатності інформаційної технології. Він не є базисним, але є змістовним – від найслабшої ланки апаратних засобів, що працює в рамках виконання завдань за конкретною подією, залежить вірогідність ефективного виконання всієї роботи.

Представлена методика дозволяє знаходити «вузькі місця» при застосуванні різної за характеристиками техніки та на основі балансового методу перерозподіляти апаратні ресурси та оптимізувати відповідно програмні засоби з метою підвищення життєздатності інформаційної технології при реалізації конкретних завдань. Описаний спосіб розподілу інформаційних навантажень між рівнями інтерпретації дозволяє оптимізувати набір технічних засобів та алгоритми, що задіяні для виконання завдань, для забезпечення необхідної швидкодії. Слід зазначити, що при виборі швидкодії окремих засобів обчислювальної техніки, що буде задіяна у обробці інформації для виконання конкретних завдань, можна враховувати також вартість вкладених витрат і отримувану економію при зміні швидкодії на основі вимог до ціни і швидкості обчислення для усієї системи. В якості критерію можна використовувати ціну швидкодії обчислювальної машини $\delta = \kappa \Delta T / \Delta C$, де κ – ваговий коефіцієнт, ΔT і ΔC – приріст швидкодії і вартості відповідно.

Наведений у статті аналіз може бути виконаний для різних ієрархій пам'яті (жорсткі диски, флеш-пам'ять, кеш-пам'ять і т. інш.).

СПИСОК ЛІТЕРАТУРИ

1. Глушков В.М. Кибернетика. Вопросы теории и практики – М.: Наука, 1986. – 488с.
2. Майерс С. Эффективное использование C++. 55 верных советов улучшить структуру и код ваших программ. – М.: ДМК-Пресс, 2006. – 300 с.
3. Сергієнко І.В. Елементи загальної теорії оптимальних алгоритмів та суміжні питання / І. В. Сергієнко, В. К. Задірака, О. М. Литвин; Нац. акад. наук України, Ін-т кібернетики ім. В. М. Глушкова. – К.: Наукова думка, 2012. – 404 с.
4. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979. – 536 с.
5. Снитюк В.Е., Быченко А.А., Джулай А.Н. Эволюционные технологии принятия решений при пожаротушении. – Черкассы, Маклаут, 2008. – 268 с.
6. Стеценко І.В. Моделювання систем: навч. посіб. [Текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. Ун-т. – Черкаси: ЧДТУ, 2010. – 399 с.
7. Теслер Г.С. Концепция построения гарантоспособных вычислительных систем / Г.С. Теслер // Математичні машини і системи. – 2006. – №1. – С. 134-145.
8. Кряжич О.О. Забезпечення життєздатності інформаційних технологій управління техногенною безпекою при їх адаптації // Математичне моделювання в економіці. – 2014. – №1. – С. 33 – 39.
9. Нейман Дж. Теория самовоспроизводящихся автоматов. – М.: Мир, 1971. – 384 с.
10. Ведута Н.И. Экономическая кибернетика. Очерки по вопросам теории. Минск: Наука и техника, 1971. – 318 с.
11. Довгий С.О., Бідюк П.І., Трофимчук О.М., Савенков О.І. Методи прогнозування в системах підтримки прийняття рішень. – К.: Азимут-Україна, 2011. – 608 с.

Стаття надійшла до редакції 09.02.2015