

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 519.6 : 004.021

М. М. БАЙАС, В. М. ДУБОВОЙ

Винницький національний технічний університет, Вінниця

КООРДИНАЦІЯ РЕШЕНИЙ О РАСПРЕДЕЛЕНИИ РЕСУРСОВ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

Анотація. Ефективна система координації та управління ресурсами є важливим чинником стабільного розвитку виробничих компаній. Незважаючи на застосування різних механізмів, спрямованих на досягнення балансу у виробничих системах, необхідність вдосконалення методів розв'язання завдань координації та управління ресурсами все ще існує. У цій статті представлено можливість застосування генетичних алгоритмів для розв'язання проблем координації та управління ресурсами в розподілених системах. У дослідженні доведено, що генетичні алгоритми можуть бути ефективно інтегровані в системи підтримки прийняття рішень для координації ресурсів та управління ними. Розроблено метод розв'язання завдань розподілу ресурсів і синхронізації технологічних процесів на прикладі виробництва молочної продукції.

Ключові слова: генетичні алгоритми, розподіл ресурсів, координація локальних систем управління розподіленими об'єктами.

Анотация. Эффективная система координации и управления ресурсами является важным фактором стабильного развития производственных компаний. Несмотря на применение различных механизмов, направленных на достижение баланса в производственных системах, необходимость совершенствования методов решения задач координации и управления ресурсами все еще существует. В этой статье представлена возможность применения генетических алгоритмов для решения проблем координации и управления ресурсами в распределенных системах. Исследование показывает, что генетические алгоритмы могут быть эффективно интегрированы в системы поддержки принятия решений для координации ресурсов и управления ими. Разработан метод решения задачи распределения ресурсов и синхронизации технологических процессов на примере производства молочной продукции.

Ключевые слова: генетические алгоритмы, распределение ресурсов, координация локальных систем управления распределенными объектами.

Abstract. An effective system of coordination and resource management is an important factor for stable development of manufacturing companies. Despite the use of various mechanisms aimed at achieving a balance in production systems, the need to improve methods for solving problems of coordination and resource management still exists. This paper presents the possibility of using genetic algorithms to solve the problems of coordination and resource management in distributed systems. The study shows that genetic algorithms can be effectively integrated into a decision support system for the coordination and management of resources. We developed a method for solving the problem of resource allocation and synchronization process on the example of dairy production.

Key words: genetic algorithms, resource allocation, coordination of local control systems distributed objects.

Введение

Пространственное распределение технологических объектов сложных систем вызывает трудности в управлении. Изолированная работа объектов сложных систем часто приводит к многочисленным проблемам и порождает высокую степень неопределенности при принятии управленческих решений. Объекты могут функционировать независимо друг от друга, иметь различные цели и ограничения, различные параметры производительности и эффективности, но для достижения общей цели необходимо организовать их сложную взаимосвязь [1, 2].

В данном контексте **проблема** координации решений при управлении производственными системами является особенно **актуальной**. Постановке задач координационного управления сложными системами и методам их решения уделяется много внимания со стороны украинских и зарубежных исследователей, о чем свидетельствует большое количество публикаций, появившихся в последние десятилетия. Фундаментальная систематизация задач координации сделана в [3-5]. Рассматриваются преимущественно итерационные и неитерационные детерминированные алгоритмы координации. Однако разнообразие задач координации (критериев, ограничений, структур систем и т.п.), большая размерность задач (количество координируемых параметров), наличие неопределенности при оценивании состояния координируемых процессов обуславливают необходимость дальнейших исследований. В частности, в опубликованных работах не рассмотрена задача распределения ресурсов во взаимосвязи с задачей синхронизации параллельных технологических процессов.

Целью работы является разработка метода координации, под которой понимается процесс принятия правильного решения по распределению ресурсов и синхронизации процессов управления технологическими линиями для достижения общих целей производства.

Метод решения

Рассмотрим **задачу** координации локальных решений на примере молочного завода, выпускающего три вида молочной продукции: молоко, кефир и сметану. Основным ресурсом, используемым заводом для производства пастеризованного молока, кефира и сметаны, является сырое молоко. Необходимо распределить поставляемый ресурс между тремя производственными линиями. Поскольку все три линии используют один комплекс упаковочно-складского оборудования, то необходимо согласование моментов окончания производственных циклов для предотвращения простоев.

На заводе действует централизованная иерархическая система координации [9]. Схема системы показана на рис. 1

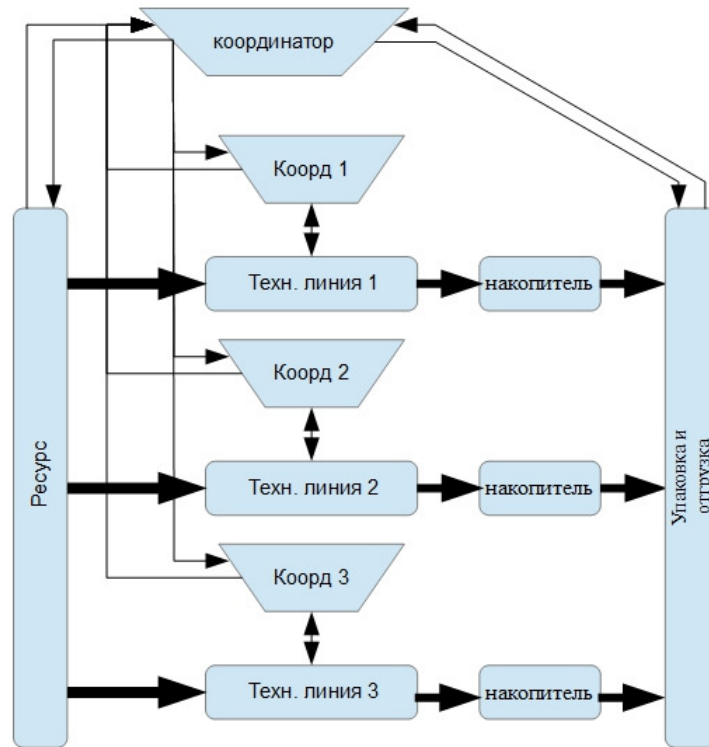


Рисунок 1 – Схема системы координации технологических линий молокозавода

Целью предприятия является максимизация прибыли, полученной от производства $n=3$ видов молочных продуктов:

$$F = \sum_{i=1}^n M_i (C_i - z_i) \rightarrow \max, \quad (1)$$

где M_i – количество произведенной продукции, C_i – цена произведенного продукта, z_i – удельные затраты производства,

$$z_i = \frac{Z_{ПОi} T_{\max}}{M_i} + Z_{ПЕРi}, \quad (2)$$

где $Z_{поi}$ – постоянные затраты за единицу времени, $Z_{перi}$ – переменные затраты, T_{\max} – длительность технологического процесса.

С учетом (2) критерий оптимизации процесса принимает вид

$$F = \sum_{i=1}^n M_i \left(C_i - \frac{Z_{ПОi} T_{\max} + M_i Z_{ПЕРi}}{M_i} \right) \rightarrow \max \quad (3)$$

Время начала и завершения операций тесно связано с объемом перерабатываемого сырья (ресурса). Для расчета T_{\max} рассмотрим график производства для трех линий, показанный на рис. 2.

Состояние $p_i(t)$ каждой производственной

$$p_i(t) = p_{oi} [H(t - t_{oi}) - H(t - t_{fi})] \quad (4)$$

где p_{oi} производительность i -й линии, t_{oi} – время начала i -й операции линии, t_{fi} – время окончания операции i -й линии, t_{pi} – длительность технологического процесса i -й линии, $H(t)$ – **Функция** Хевисайда

$$H(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases} \quad (5)$$

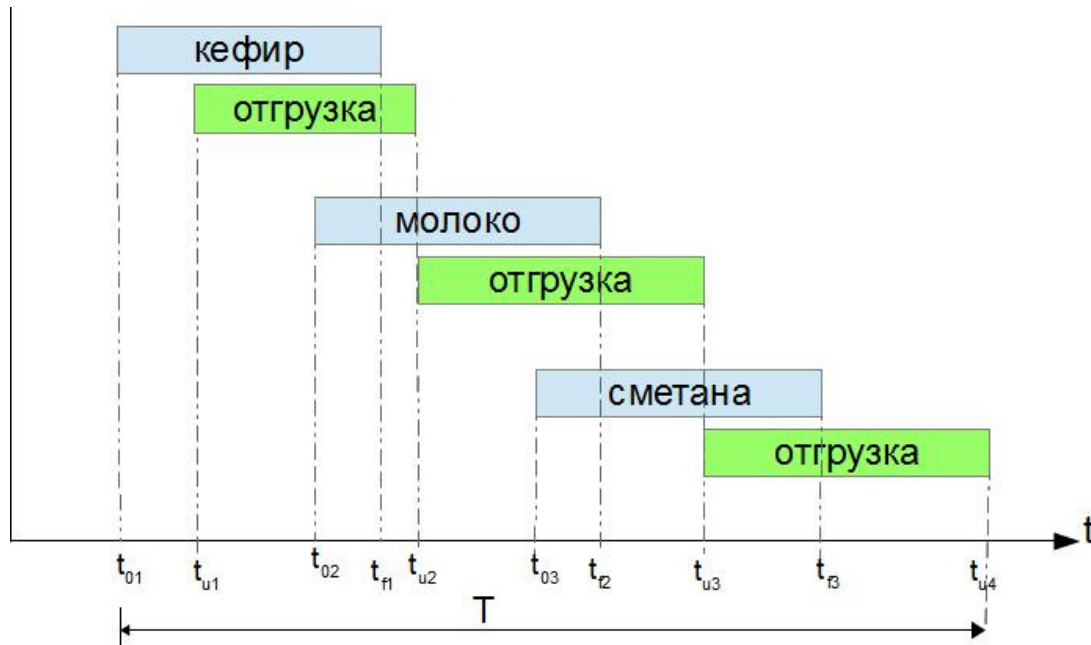


Рисунок 2 – Ленточный график производственного процесса

Продукция всех технологических линий поступает на один участок упаковки и отгрузки. В связи с этим продукция линии собирается в накопителе пока участок упаковки и отгрузки занят другой продукцией. Между временем производства, временем отгрузки и размерами накопителей выполняются соотношения:

$$t_{fi} = t_{oi} + t_{pi} \quad (6)$$

$$t_{li} = \frac{A \max_i}{P_{oi}} \quad (7)$$

$$t_{ui} = \frac{A \max_i}{u_{oi} - P_{oi}} \quad (8)$$

$$T_i = t_{li} + t_{ui} \quad (9)$$

$$n = \left[\frac{t_{fi} - t_{oi}}{T_i} \right] \quad (10)$$

где t_{li} – длительность заполнения накопителя, $A \max_i$ – объем накопителя, A – объем, который накапливается в накопитель, t_{ui} – длительность отгрузки накопителя, n – количество циклов, u_{oi} – производительность процесс отгрузки, T_i – длительность цикла, $u_i(t)$ – состояние отгрузки.

$$u_i(t) = u_{oi} \left[\sum_0^n H\left(t - t_{oi} - t_{li} - nT_i + \frac{t_{ui}}{2}\right) - H\left(t - t_{oi} - t_{li} - nT_i - \frac{t_{ui}}{2}\right) + k \left[H(t - t_{fi}) - H\left(t - p_{oi} \left(\frac{t_{fi} - nT_i}{u_{oi}}\right)\right) \right] + (1 - k) \left[H(t - nT_i + t_{li}) - H\left(t - \left(t_{fi} + \frac{A_{max} - (u_{oi} - p_{oi})(t_{fi} - (nt_{li} + t_{li}))}{u_{oi}}\right)\right) \right] \right] \quad (11)$$

$$k = \begin{cases} 1, & t_f < nT_i + t_{li} \\ 0, & t_f \geq nT_i + t_{li} \end{cases} \quad (12)$$

$$A_i(t) = (p_i(t) + u_i(t))t \quad (13)$$

Цикл производства показан на рис.3.

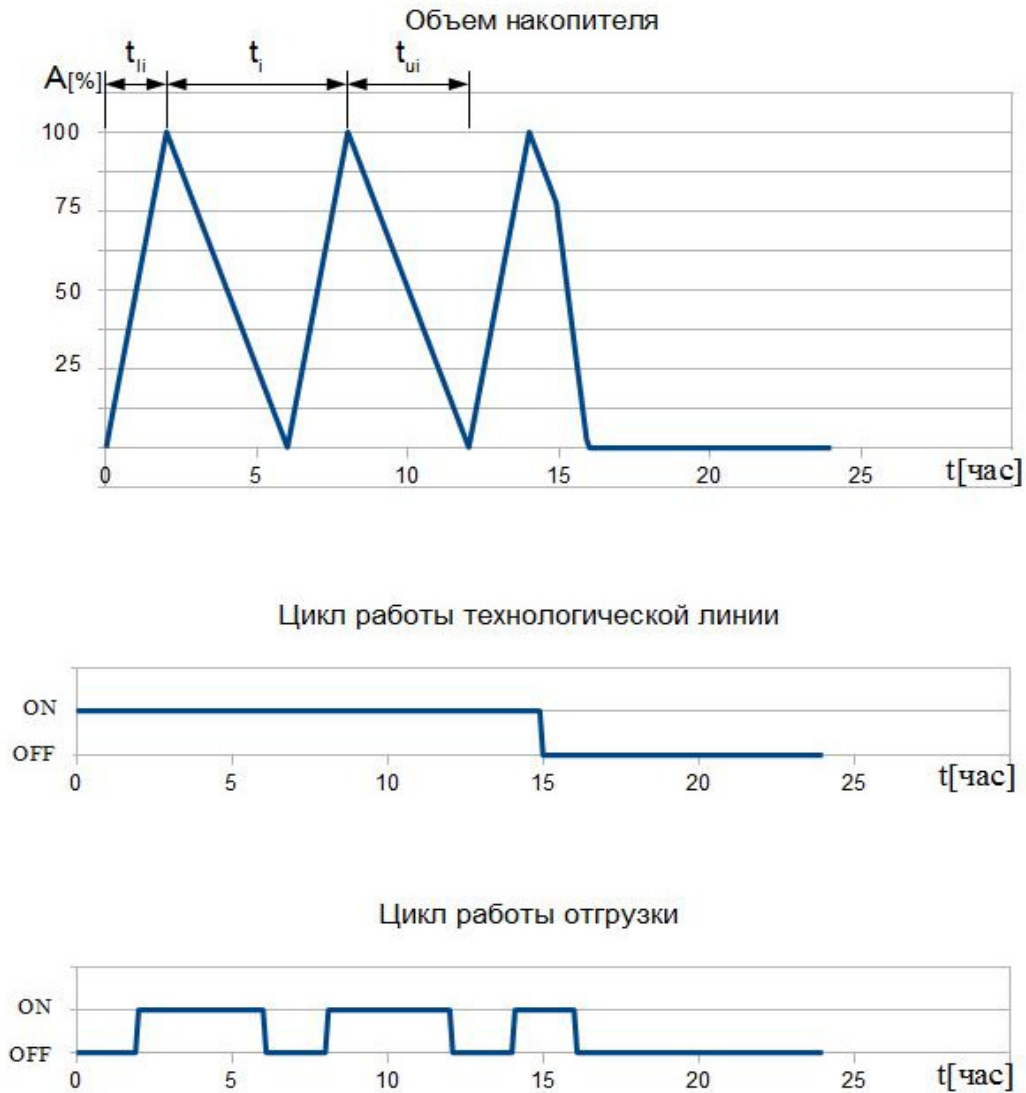


Рисунок 3 – Цикл производства трех видов продукции

Таким образом, для $n=3$ целевая функция координации зависит от 6 переменных: $F(M_1, M_2, M_3, t_{01}, t_{02}, t_{03})$.

В данной задаче учтем следующие ограничения:

- количество производимой продукции не должно превышать спрос

$$D_i \geq M_i, \tag{14}$$

где D_i – величина спроса на производимую продукцию;

- общее количество продукции соответствует количеству ресурса с учетом коэффициентов выхода

$$P_{\Sigma} = \sum_{i=1}^n \frac{M_i}{\eta_i} \tag{15}$$

- время начала работы линии

$$0 \leq t_{0i} \leq T_{\max} - T_i$$

Задачу оптимальной координации будем решать при исходных данных, основные из которых представлены в табл.1 [9].

Таблица 1. Характеристика производимой продукции

№	Наименование продукции	Объем накопителя (тонна)	Цена, [грн./л]	Постоянные затраты, грн/ месяц	Переменные затраты грн/тонна	Производительность (тонна/ч)	
						процесса	отгрузки
1	Кефир	0,16	11,00	121115,50	13476	0,080	0,50
2	Сметана	0,10	17,80	122797,00	43274	0,036	0,50
3	Молоко	1,00	10,50	490706,50	10719	0,486	0,55

Значение целевой функции находится в результате численного решения системы уравнений (3)-(13). Данная задача оптимизации характеризуется нелинейностью и полимодальностью целевой функции, наличием ограничений, большим количеством параметров оптимизации. Среди наиболее подходящих методов решения задачи можно выделить генетические алгоритмы [1, 2, 6, 7].

Генетический алгоритм оперирует конечным множеством решений \square , именуемым популяцией \square [8]. Новые решения получаются в результате различных комбинаций частей решений популяции при помощи операторов отбора, кроссовера и мутации. Примеры выполнения этих операций для рассматриваемой задачи показаны на рис. 4.

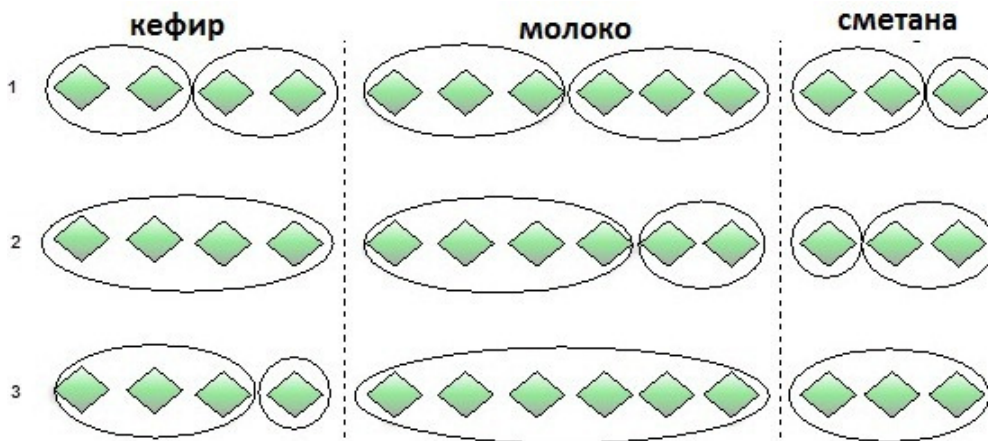


Рисунок 4 – Варианты распределения ресурса

Новые решения позиционируются в популяции в соответствии с их положением на поверхности оптимизируемой функции. Последовательность работы генетического алгоритма выглядит следующим образом.

1. На первом этапе формируется начальная популяция, т.е. массив решений некоторым образом заполняется определенными значениями. В частности, эти значения могут генерироваться случайным образом. $P = X_1, X_2, \dots, X_n$. X_i – элементы особой популяции, но для данной задачи распределения ресурсов с учетом приведенных выше ограничений более эффективной будет неслучайная начальная популяция вида:

$$P = \begin{bmatrix} M_1 & M_2 & M_3 & t_{01} & t_{02} & t_{03} \\ P_{\Sigma} & 0 & 0 & 1 & 0 & 0 \\ 0 & P_{\Sigma} & 0 & 0 & 1 & 0 \\ 0 & 0 & P_{\Sigma} & 0 & 0 & 1 \\ \frac{P_{\Sigma}}{3} & \frac{P_{\Sigma}}{3} & \frac{P_{\Sigma}}{3} & 1 & 1 & 1 \\ \frac{P_{\Sigma}}{2} & \frac{P_{\Sigma}}{2} & 0 & 1 & 3 & 2 \\ \frac{P_{\Sigma}}{2} & 0 & \frac{P_{\Sigma}}{2} & 1 & 2 & 3 \\ 0 & \frac{P_{\Sigma}}{2} & \frac{P_{\Sigma}}{2} & 2 & 1 & 3 \\ \frac{P_{\Sigma}}{2} & \frac{P_{\Sigma}}{3} & \frac{P_{\Sigma}}{6} & 2 & 3 & 1 \\ \frac{P_{\Sigma}}{2} & \frac{P_{\Sigma}}{6} & \frac{P_{\Sigma}}{3} & 3 & 1 & 2 \\ \frac{P_{\Sigma}}{6} & \frac{P_{\Sigma}}{2} & \frac{P_{\Sigma}}{3} & 3 & 2 & 1 \end{bmatrix}$$

2. На втором шаге вычисляется приспособленность каждой особи популяции по ее признакам с использованием критерия (3). Чем лучше значение, принимаемое целевой функцией и обеспечиваемое особью, тем выше ее приспособленность.

3. Особи ранжируются по критерию приспособленности. Из особей с самым высоким уровнем приспособленности (для популяции из 13 особей уровень приспособленности выше среднего будут иметь 6 особей) составляются родительские пары. Общее число пар $C_6^2 = \frac{6!}{2!(6-2)!} = 15$, что обеспечивает рост популяции.

4. Выполняется операция кроссовера, которая состоит в случайном выборе точки на хромосоме (вектор хромосом $X_i = C_1, C_2, \dots, C_m$), относительно которой будет производиться скрещивание. В нашем примере на рис. 4. при скрещивании часть новой особи до этой точки формируется за счет одного родителя, а часть после этой точки – за счет другого родителя.

5. Новые особи мутируют с вероятностью не менее $\frac{1}{N}$, где N – размер популяции. Это обеспечивает изменчивость популяции.

6. Отбрасываются особи, не удовлетворяющие ограничения (2) и (3).

7. Вычисляется приспособляемость новых особей, после чего эти они заменяют в популяции предыдущие наименее приспособленные особи. Очевидно, что если приспособляемость какой-либо новой особи ниже, чем приспособляемость старых, то эта особь не включается в популяцию.

8. Если самое лучшее решение в популяции не признается удовлетворительным, то снова возвращаемся к третьему шагу описанного алгоритма. Если же решение нас устраивает, то алгоритм завершает свою работу.

В табл. 2 представлены результаты выполнения программы, когда число особей в популяции остается постоянным и равняется десяти, а число поколений увеличивается с каждым шагом.

Таблица 2. Результаты выполнения программы для фиксированного размера популяции.

Количества поколений	M1	M2	M3	Последовательность	Целевая функция
10	1	5	12	312	53663
50	1	5	12	312	51100
100	2	5	11	132	39338
200	2	5	11	231	42981
300	3	5	10	231	43595
400	1	5	12	231	37480
500	2	5	11	312	50516

В табл. 3 представлены результаты выполнения программы, где уже число особей в популяции увеличивается с каждым шагом, а число поколений остается постоянным и равняется десяти.

Таблица 3. Результаты выполнения программы для фиксированного числа поколений

Размер популяции	M1	M2	M3	Последовательность	Целевая функция
10	2	5	11	231	45677
50	1	5	12	321	54537
100	2	4	12	321	53253
200	2	3	13	321	53835
300	1	5	12	312	54261
400	2	5	11	312	55330
500	1	5	12	312	54577

На рис. 5 представлен пример влияния количества поколений на целевую функцию.

На рис. 6 представлен пример влияния размера популяции на целевую функцию.

В первом случае мы видим, что при фиксированном размере популяции коэффициент вариации остается практически неизменным. Во втором случае, при неизменном количестве поколений коэффициент вариации уменьшается.

Для анализа эффективности применения генетического алгоритма для решения поставленной задачи координации в качестве базы сравнения был использован алгоритм случайного поиска [10]. В качестве критерий оценки были использованы следующие: а) число итераций, необходимое для достижения значения допуска err ; б) коэффициент вариации CV на тысячу прогонов алгоритма для обеспечения надежности алгоритма; в) значение критерия оптимизации для данной задачи. Результаты сравнения показаны в табл. 4.

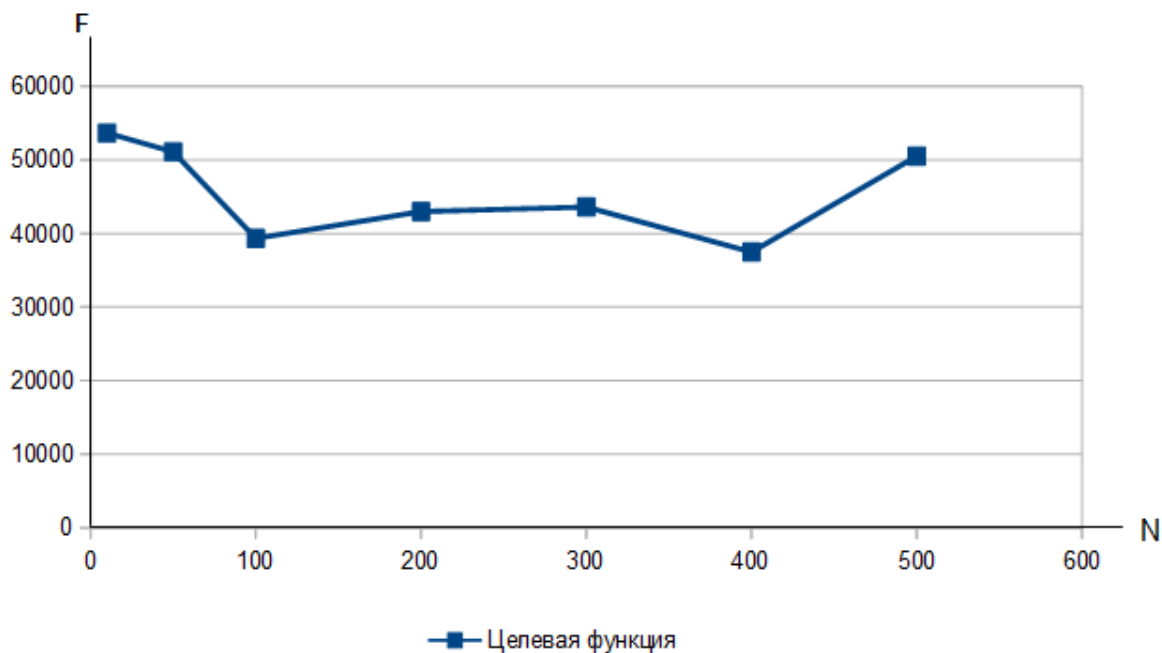


Рисунок 5 – Зависимость целевой функции от количества поколений

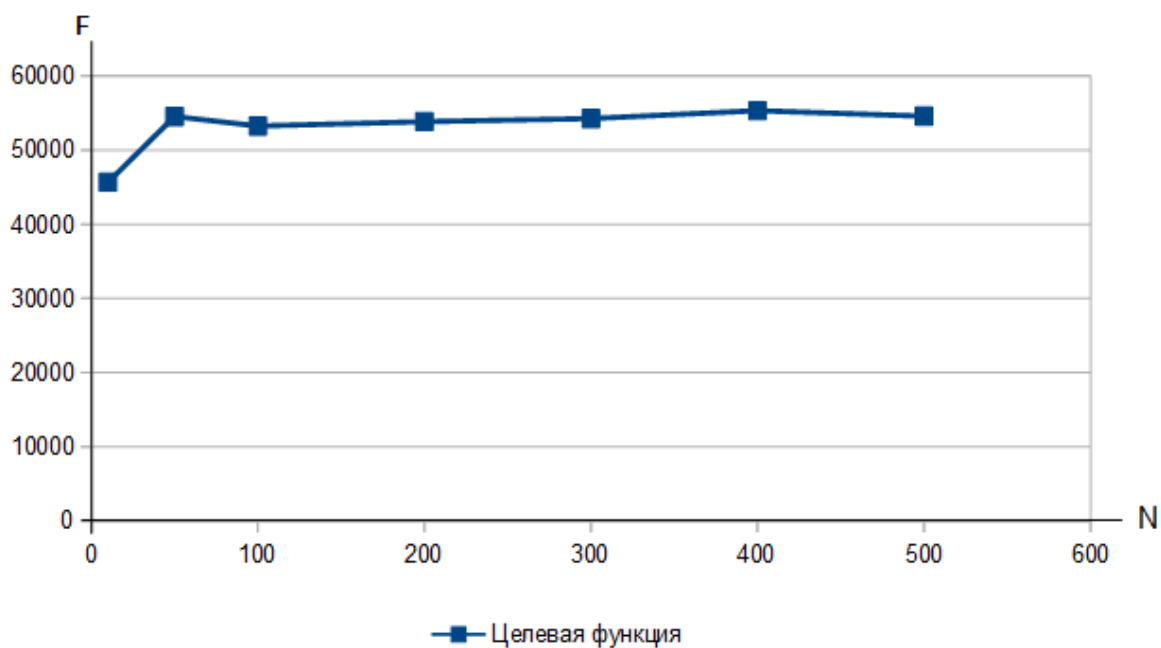


Рисунок 6 – Зависимость целевой функции от размера популяции

Таблица 4. Сравнение выполнения используемых алгоритмов

Метод	Количество итераций		CV на 1000 запусков	Значение критерия оптимизации для данной задачи
	ferr < 3.0%	ferr < 0.5%		
Генетический алгоритм	500	1000	0.036	53070
Случайный поиск	1000	3000	0.062	51230

Заключение

Предложенное решение задачи координации распределения ресурсов и синхронизации работ предприятия с использованием генетических алгоритмов дает лучшее приближение к оптимальному значению при меньшем количестве итераций по сравнению с методом случайного поиска. Это позволяет получить выигрыш во времени принятия решений и тем самым сократить простой оборудования

Разработанная модель координации деятельности молочного завода на базе генетического алгоритма оптимизации может быть адаптирована для других технологических процессов, требующих взаимосвязанных распределения ресурсов и синхронизации.

В рассмотренном примере использовался только один ресурс, который распределялся между технологическими линиями. В дальнейшем планируется увеличить количество ресурсов, подлежащих распределению.

Благодарности

Эта работа была выполнена при финансовой поддержке гранта "Министерство высшего образования, науки, технологий и инноваций "SENESCYT" Эквадор, и спонсируется университетом Santa Elena "UPSE", Эквадор.

Список литературы

1. Marcia Bayas Sampedro. Efficient Resources Allocation in Technological Processes Using Genetic Algorithm / Marcia Bayas Sampedro and V.M. Dubovoy / Middle-East Journal of Scientific Research 14 (1): 01-04, 2013 ISSN 1990-9233 © IDOSI Publications, 2013 DOI: 10.5829/idosi.mejsr.2013.14.1.16313.
2. Bayas M. M. Efficient Resources Allocation in Technological Processes Using an Approximate algorithm based on Random Walk /. Bayas M. M. and Dubovoy V. M. / International Journal of Engineering and Technology. Vol. 5 № 5 Oct-Nov 2013.
3. Месарович М. Теория иерархических многоуровневых систем / Месарович М., Мако Д., Така-хара И. – М.: Мир, 1973. – 344 с.
4. Ладанюк О. А. Автоматизированное управление взаимосвязанными подсистемами технологических комплексов пищевых производств / Ладанюк О. А. – К., 1996. – 176 с.
5. Системна задача координації в технологічних комплексах неперервного типу / Ладанюк А. П., Шумигай Д. А., Бойко Р. О. – Режим доступу : http://dspace.nuft.edu.ua/jspui/bitstream/123456789/4444/1/Sh_3.pdf.
6. Панченко Т.В. Генетические алгоритмы / Панченко Т.В.; под ред. Ю. Ю. Тарасевича. – Астрахань: Астраханский университет, 2007. – 87 с.
7. Niu D. An Efficient Distributed Algorithm for Resource Allocation in Large- Scale Coupled Systems/ Niu D. and Li B. / 2013.
8. Mitchell M. An introduction to genetic algorithms / Mitchell M. – Cambridge, MA: MIT Press., 1996
9. Технология и техника переработки молока / Молочный портал. – Режим доступу : <http://molokoportal.ru/category/tehnologiya-i-tehnika-pererabotki-moloka/>
10. Ghate A. Adaptive search with stochastic acceptance probabilities for global optimization / Archis Ghate. Robert L. Smith / Operatios Research Letters. Vol. 36 № 3 May 2008 . – 285-290 p.

Сведения об авторах

Байас Марсия – аспирантка кафедры КСУ, Винницкий национальный технический университет, Хмельницкое шоссе, 95, г. Винница, 21021.

Дубовой Владимир Михайлович – д.т.н., профессор, заведующий кафедры КСУ, Винницкий национальный технический университет, Хмельницкое шоссе, 95, г. Винница, 21021, тел. 59-81-57.

УДК 004.056

Ю. В. БАРИШЕВ, В. А. КАПЛУН

Вінницький національний технічний університет, м. Вінниця

МЕТОД АВТЕНТИФІКАЦІЇ ВІДДАЛЕНИХ КОРИСТУВАЧІВ ДЛЯ МЕРЕЖЕВИХ СЕРВІСІВ

Анотація: В даній статті представлено аналіз методів авторизації користувачів мережесервісів. Запропоновано метод захисту мережесервісів, який передбачає розробку правил політики інформаційної безпеки не лише для користувачів, але й для робочих станцій, з яких цим користувачам дозволяється отримувати доступ до конфіденційної інформації. Таким чином для отримання доступу до інформаційного ресурсу необхідна як успішна автентифікація користувача, так і успішна автентифікація робочої станції перед початком сеансів обміну даними. Крім того метод передбачає шифрування даних, що передаються, на ключі, який залежить від параметрів автентифікації та даних, надісланих протягом сеансу зв'язку.

Ключові слова: автентифікація, гешування, комп'ютерна мережа, пароль, параметри робочої станції.

Аннотация: В данной статье представлено анализ методов авторизации пользователей сетевых сервисов. Предложено метод защиты сетевых сервисов, который предполагает разработку правил политики информационной безопасности не только для пользователей, но и для рабочих станций, с которых этим пользователям позволяет получать доступ к конфиденциальной информации. Таким образом для получения доступа к информационному ресурсу необходима как успешная аутентификация пользователя, так и успешная аутентификация рабочей станции перед началом сеансов обмена данными. Кроме того метод предполагает шифрование передающихся данных на ключе, который зависит от параметров аутентификации и данных, переданных во время сеанса связи.

Ключевые слова: аутентификация, хеширование, компьютерная сеть, пароль, параметры рабочей станции.

Abstract: Network services users authentication methods analyses are performed in the article. The network services protection method, which is supposed information protection rules development for both users and workstations, which usage for getting access to confidential data by users is permitted. Thereafter to get the access to an informational resource both the user authentication and the workstation authentication should be passed before information exchange begins. Moreover the method is supposed to implement transferred data encryption with usage of the key, that depends on authentication parameters and data, which is transferred during the communication session.

Key words: authentication, hashing, computer network, password, workstation parameters.

Вступ

Комплексна природа процесу захисту інформації породжує необхідність аналізу багатьох загроз, які різні за своєю природою. І часто користь від використання потужних методів захисту інформації таких, як криптографічні методи, зводиться нанівець внаслідок впливу людського фактору [1]. Наприклад, використання протоколів автентифікації користувачів при обміні даними між ними та файловими серверами не може гарантувати захист конфіденційності інформації, що передається, на стороні отримувача цієї інформації, тобто користувача мережевого сервісу. Водночас використання мережесервісів дозволяє підприємствам пришвидшити обробку інформації. Останнє набуває особливої актуальності для підприємств, підрозділи яких територіально розподілені, однак потребують постійної взаємодії під час виробничого процесу. У зв'язку з тим, що витік інформації може відбуватися внаслідок візуального зняття інформації з моніторів комп'ютерів, фізичного втручання зловмисників в роботу працівників підприємства, які мають права на обробку конфіденційної інформації, тому важливо в політиці інформаційної безпеки обмежувати і способи обробки інформації, і робочі місця, з яких вона може бути здійснена. Саме тому актуальною є розробка методу комплексного захисту конфіденційності інформації для користувачів мережесервісів, який би дозволяв обмежувати доступ авторизованих користувачів до даних, якщо він відбувається з незахищених робочих місць.

Мета досліджень

Виходячи з вищевказаного, метою даного дослідження визначено покращення захисту конфіденційності інформації, що надається мережевими сервісами, шляхом розробки методу та програмного засобу авторизації користувачів, який дозволить обмежити кількість робочих станцій, з яких можна здійснити авторизацію, та шифрувати дані, що передаються між робочою станцією та сервером.

Завдання досліджень

Для досягнення мети необхідно розв'язати такі задачі:

- проаналізувати відомі методи автентифікації користувачів;
- розробити метод автентифікації на основі гешування та прив'язки до параметрів робочої станції;
- розробити метод шифрування даних, що передаються;
- реалізувати дані методи у вигляді програмного засобу.

Аналіз відомих методів авторизації користувачів мережесервісів

Один з найбільш розповсюджених методів забезпечення авторизації користувачів у комп'ютерних системах базується на використанні ними пароля, що належить лише одному користувачу [2-4]. Під час автентифікації користувач вводить свій ідентифікатор і пароль. Еталонна пара логін-пароль зберігається в спеціальній базі [2, 5]. Процедура автентифікації в цьому випадку передбачає такі етапи [2, 4, 5]:

- користувач запитує доступ до системи і вводить особистий ідентифікатор та пароль;
- введені унікальні дані надходять на сервер, де порівнюються з еталонними;

- у випадку збігу даних з еталонними, автентифікація визнається успішною, при відмінності – користувачу пропонується знову ввести ідентифікатор та пароль.

Введений користувачем пароль може передаватися мережею таким чином [4-6]:

- у незашифрованому вигляді, на основі протоколу пароліного автентифікації;
- з використанням шифрування;
- з використанням гешування.

В останніх двох випадках дані, що вводились користувачем, передаються мережею у захищеному вигляді. З точки зору максимізації захищеності, при зберіганні й передаванні паролів доцільніше використовувати гешування. Це пояснюється тим, що, перехопивши зашифрований пароль, зломисник може зробити висновок про його довжину, що спростить йому реалізацію атаки підбору пароля. Крім того шифрування даних невеликої довжини (1-3 блоки даних) вразливе до атак збільшення довжини повідомлення – падіння (padding) [7]. Водночас геш-значення, отримане внаслідок виконання процесу гешування, завжди має сталу довжину незалежно від довжини вхідних даних гешування, в даному випадку – пароля, а тому не може бути об'єктом таких атак [3, 6, 8]. Саме тому більш стійкими до атак є методи, що передбачають зберігання на стороні сервера саме геш-значення пароля.

На рис. 1 наведено схему авторизації користувачів, що використовує збереження паролів у вигляді їх геш-значень [2, 5].

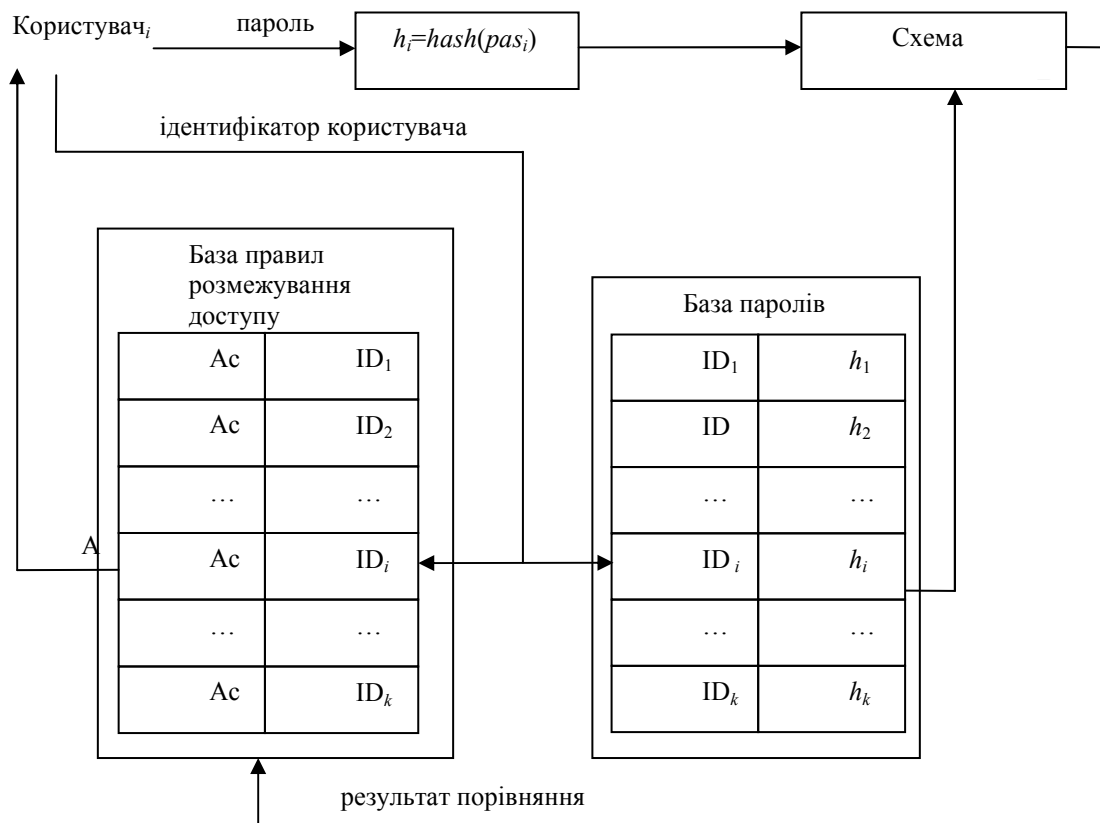


Рисунок 1 – Схема авторизації користувача з використанням паролів

де h_i – геш-значення пароля i -го користувача; Access _{i} – набір прав доступу для i -го користувача; $\text{hash}(\cdot)$ – функція гешування;

З рис. 1 видно, що пароль використовуватиметься користувачем і при наступній авторизації, тобто він є багаторазовим. Використання багаторазових паролів має низку недоліків. По-перше, сам пароль, або його геш-значення зберігаються на сервері. Відповідно, отримавши доступ до них, зломисник може, надсилаючи їх на сервер, імітувати користувачів та здобувати їх права доступу. По-друге, часто суб'єкт повинен запам'ятовувати пароль, тому розв'язання задачі вибору стійкого паролю покладається на нього. В такому випадку правильність розв'язання цієї задачі адміністратору з інформаційної безпеки складно перевірити, а зломисник може отримати пароль, застосувавши навички соціальної інженерії [1, 2, 9].

При захисті програм від несанкціонованого копіювання, тобто автентифікації комп'ютерних систем, застосовуються методи, які передбачають внесення до програмного коду блоків, які впливатимуть на процес виконання програми залежно від параметрів комп'ютерної системи, на якій виконується даний код. Зазвичай дані блоки коду виконують перевірку автентичності комп'ютерної системи шляхом порівняння її поточних характеристик з еталонними [8, 9].

Методи автентифікації комп'ютерних систем попри схожість їх загальних алгоритмів, відрізняються параметрами комп'ютерної системи, за якими відбувається автентифікація цієї системи. Прив'язка відбувається на основі таких характеристик комп'ютерної системи [8, 9]: серійний номер жорсткого диску; дата створення та контрольна сума BIOS; версії та властивості операційних систем; вміст системних файлів; продуктивність апаратури; наявність додаткових пристроїв тощо. В певних випадках для надання унікальності кожному з сеансів автентифікації використовується криптографічна сіль [10].

Отже, процедури автентифікації комп'ютерної системи та її користувачів у відомих методах відбуваються незалежно одне від одного. Для протидії цьому використовуються криптографічні алгоритми, зокрема гешування та шифрування [2, 3, 6]. Шифрування даних, що передаються, в мережевих сервісах відбувається за алгоритмами симетричного або асиметричного шифрування [2, 3, 6]. Однак зашифровані дані автентифікації матимуть довжину, яка залежить від довжини вхідних даних. Саме тому доцільніше використовувати гешування, оскільки воно передбачає отримання вихідного значення фіксованої довжини незалежно від довжини вхідних даних.

Відомі методи автентифікації користувачів передбачають, що успішно авторизований, користувач не буде виконувати обробку інформації із застосуванням незахищеної робочої станції. З урахуванням, що причиною інцидентів у галузі інформаційної безпеки в переважній більшості випадків є людський фактор [1, 11] такі міркування – занадто оптимістичні, а тому не можуть мати місце при розробці розподілених інформаційних систем, в яких обробляється інформація цінна не лише для користувачів цієї системи.

Метод організації захищеного зв'язку користувачів

З розглянутого вище впливає, що при реалізації доступу до віддалених сервісів автентифікація користувача та системи, з якої відбувається спроба доступу повинна відбуватись одночасно, адже в цьому випадку адміністратор системи захисту інформації може переконатися в захищеності процесів обробки інформації на робочих станціях та заблокувати шляхи витоку інформації з них. Для цього авторами пропонується метод організації захищеного доступу користувачів до мережевих сервісів, який передбачає такі дії:

1. Введення ідентифікатора користувача до робочої станції.
2. Пересилання ідентифікатора користувача та ідентифікатора робочої станції до сервера.
3. Обчислення геш-значення на стороні робочої станції:

$$h_{ij} = f(h_0, pas_i \parallel params_j),$$

де pas_i – пароль i -го користувача ($i = \overline{1, k}$) мережі;

$params_j$ – параметри j -ї робочої ($j = \overline{1, n}$) станції;

$f(\cdot)$ – функція гешування.

4. Надсилання отриманого геш-значення від робочої станції до сервера та його порівняння зі значенням, що зберігається на сервері.

5. Якщо геш-значення збігаються, то сервер надає користувачеві відповідні права доступу до мережевого сервісу, інакше зв'язок розривається.

6. Обмін між сервером і робочою станцією конфіденційними даними відбувається у зашифрованому вигляді, причому при першому сеансі обміну даними, як ключ шифрування використовується геш-значення пароля користувача h_i , що формалізується так:

$$data_1^* = E_{h_i}(data_1).$$

Очевидно, що при цьому розшифрування відбувається на тому ж ключі. Це можливо внаслідок того, що і авторизована сторона, і сервер при коректному завершенні кроку 5 мають підтвердження того, що пароль був введений користувачем правильно.

7. Кожен наступний сеанс обміну інформацією буде зашифровуватися з використанням ключа, який визначається шляхом гешування даних, пересланих під час попереднього сеансу обміну даними:

$$h_i^{(w)} = f(h_i^{(w)}, data_{w-1}),$$

де $h_i^{(0)} = h_i$.

Відповідно розшифрування під час w -го сеансу обміну даними між сервером та робочою станцією пропонується виконувати відповідно до такої формули:

$$data = D_{h_i}(data^*).$$

На рис. 2 зображено схему авторизації користувача і робочої станції, що пропонується у даному методі. Вигляд бази даних автентифікації, наведеної на рис. 2, спрощено для більшої наочності методу.

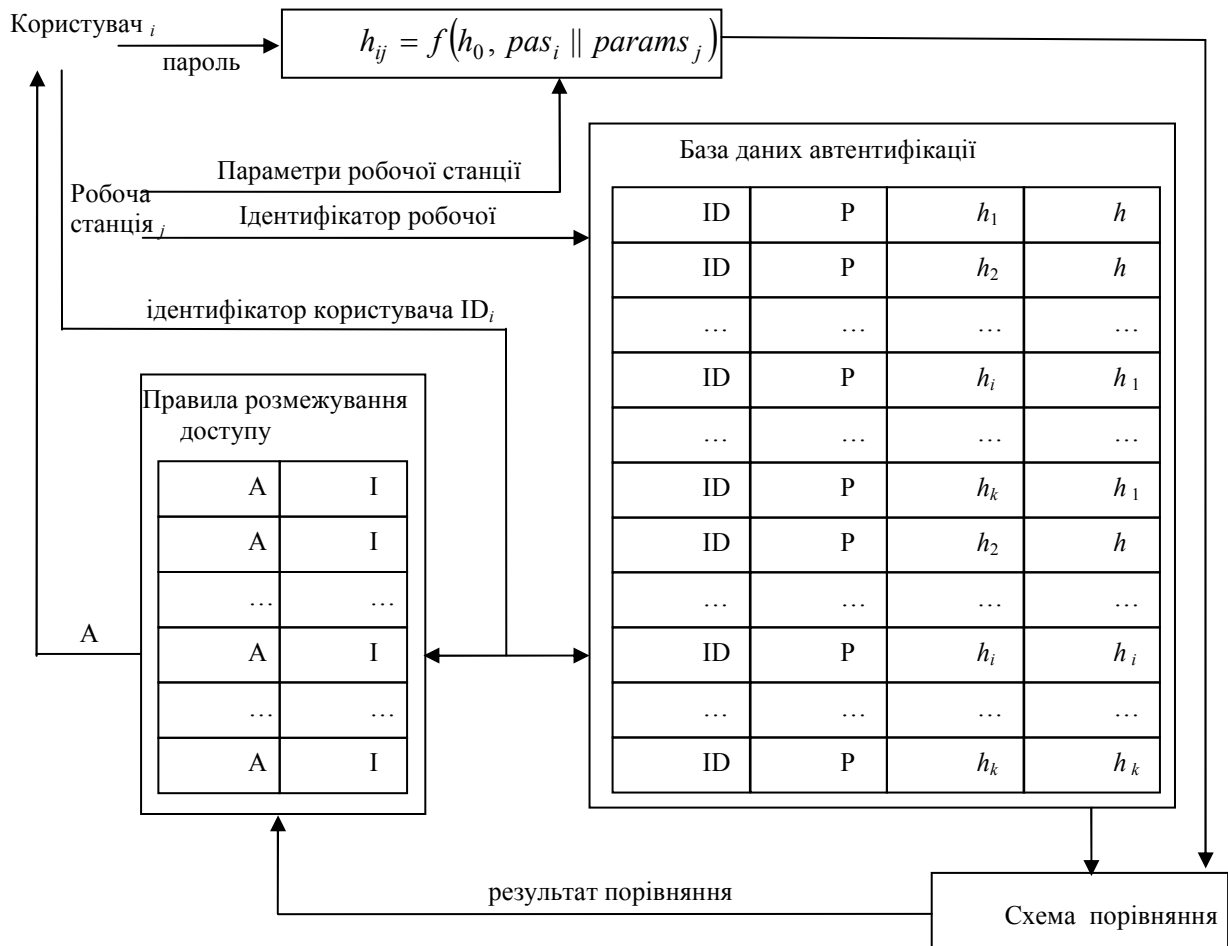


Рисунок 2 – Модифікована схема авторизації користувача з використанням паролів

З рис. 2 видно, що для успішної реалізації кроку 6 запропонованого методу в базі даних автентифікації передбачено наявність геш-значення пароля i -го користувача h_i , хоча власне під час автентифікації це геш-значення не використовується, а натомість використовується геш-значення h_{ij} .

Внаслідок впровадження методу кількість даних, якими обмінюються сторони під час авторизації, фактично не змінюється, оскільки ідентифікатор робочої станції повідомляється серверу ще на етапі створення зв'язку між ним та робочою станцією, відповідно в переважній більшості випадків ідентифікатор робочої станції буде відомим серверу ще до початку застосування методу. Відповідно в цих випадках автори пропонують спростити крок 2 методу та не пересилати ідентифікатор робочої станції. Дана модифікація дозволить не лише заощадити ресурси каналів зв'язку, але й ускладнити зловмиснику злам у випадку, коли він має можливість спостерігати за даними, що передаються між сторонами. У випадках, коли ідентифікатор робочої станції необхідно буде пересилати, кількість даних, що передаються також зросте незначно, оскільки для надання чотирьом мільярдам робочих станцій унікальних ідентифікаторів достатньо чотирьох байтів.

Висновки

Описаний вище метод дозволяє приховати від користувачів та від зловмисника, який відслідковує дані, які передаються, спосіб захисту, тим самим ускладнивши свій злам. З метою уникнення атак, пов'язаних із соціальним інжинірингом та візуальним зняттям інформації, запропонований метод, крім автентифікації користувача, передбачає автентифікацію робочої станції за її унікальними параметрами. Перевагою даного методу є те, що розмежування прав доступу користувачів відбувається не лише організаційними засобами, які можна обійти, але й криптографічними, на злам яких зловмисникам необхідно буде витратити набагато більше часу.

Список літератури

1. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире / Брюс Шнайер. – СПб.: Питер, 2003. – 368 с.
2. Защита информации в телекоммуникационных системах / Г. Ф. Коначович, В. П. Климчук, С. М. Паук, В. Г. Потапов. – К.: "МК-Пресс", 2005. – 288 с.
3. Информационная безопасность открытых систем: Учебник для вузов. В 2-х томах. Том 2. – Средства защиты в сетях / С. В. Запечников, Н. Г. Милославская, А. И. Толстой, Д. В. Ушаков. – М.: Горячая линия-Телеком, 2008. – 558 с.
4. Гайворонський М. В. Безпека інформаційно-комунікаційних систем. / М. В. Гайворонський, О. М. Новіков. – К.: Видавнич група BHV, 2009. – 608 с.
5. Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам. Учебное пособие для вузов. / А. А. Афанасьев, Л. Т. Веденев, А. А. Воронцов и др.; Под ред. А. А. Шелупанова, С. Л. Груздева, Ю. С. Нахаева. – М.: Горячая линия-Телеком, 2009. – 552 с.
6. Петров А. А. Компьютерная безопасность. Криптографические методы защиты / А. А. Петров. – М.: ДМК, 2000. – 448 с.
7. Vaudenay S. Security Flaws Induced by CBC Padding Applications to SSL, IPSEC, WTLS... / Serge Vaudenay. – 2002. – 12 p. – Режим доступу до статті: http://infoscience.epfl.ch/record/52417/files/IC_TECH_REPORT_200150.pdf
8. Дудатьев А. В. Захист програмного забезпечення. Навчальний посібник. Частина 1. / А. В. Дудатьев, В. А. Каплун, В. П. Семеренко. – Вінниця: ВНТУ, 2005. – 140 с.
9. Казарин О. В. Теория и практика защиты программ. / О. В. Казарин – М.: МГУЛ, 2004. – 450 с.
10. Biham E. A Framework for Iterative Hash Functions: HAIFA. / Eli Biham, Orr Dunkelman // NIST second cryptographic hash workshop. – 2006. – 9 с. – Режим доступу до статті: http://csrc.nist.gov/groups/ST/hash/documents/DUNKELMAN_NIST3.pdf
11. Просис К. Расследование компьютерных преступлений / Крис Просис, Кевин Мандиа. – М.: Лори, 2012 – 416 с.

Відомості про авторів

Баришев Юрій Володимирович – к. т. н., старший викладач кафедри захисту інформації Вінницького національного технічного університету, Хмельницьке шосе, 95, м. Вінниця, 21021, службовий телефон (0432) 598 380; e-mail: yuriy.baryshev@gmail.com.

Каплун Валентина Аполінаріївна – старший викладач кафедри захисту інформації Вінницького національного технічного університету, Хмельницьке шосе, 95, м. Вінниця, 21021, службовий телефон (0432) 598 380.

УДК 62 50 658 21

Т.Н. БОРОВСКАЯ, И.С. КОЛЕСНИК, В.А. СЕВЕРИЛОВ, И.В. ШУЛЬГАН

Винницкий национальный технический университет, г. Винница

ОПТИМАЛЬНОЕ АГРЕГИРОВАНИЕ ИНТЕГРИРОВАННЫХ СИСТЕМ «ПРОИЗВОДСТВО-РАЗВИТИЕ»

Аннотация. Рассматривается задача замены интегрированной системы «производство – развитие производства» эквивалентным оптимальным элементом по функции «затраты – выпуск». Критерий оптимальности – приращение темпа выпуска, переменная управления – распределение входного ресурса между затратами производства и затратами развития – повышения эффективности производства и увеличения производственных мощностей. Решение данной задачи расширяет алгебру оптимального агрегирования структур на системы с параметрическими связями между элементами. Практическое значение разработки – эффективное управление современными динамичными производствами. Разработаны рабочие модели, приведены примеры исследований.

Ключові слова: функция производства, функция развития, оптимальное агрегирование, бинарный оператор, метод оптимального агрегирования.

Анотация. Розглядається задача заміни інтегрованої системи «виробництво - розвиток виробництва» еквівалентним оптимальним елементом по функції «витрати - випуск». Критерій оптимальності - приріст темпу випуску, змінна управління - розподіл вхідного ресурсу між витратами виробництва і витратами розвитку - підвищення ефективності виробництва і збільшення виробничих потужностей. Рішення даної задачі розширює алгебру оптимального агрегування структур на системи з параметричними зв'язками між елементами. Практичне значення розробки - ефективне управління сучасними динамічними виробництвами. Розроблено робочі моделі, наведено приклади досліджень.

Ключевые слова: функция производства, функция развития, оптимальное агрегирование, бинарный оператор, метод оптимального агрегирования...

Abstract. The problem of replacing an integrated system of "production - production development" equivalent element for optimal function "input - output". Optimality criterion - the increment rate of release, control variable - the input resource allocation between expenses of production and expenses for development - improve production efficiency and increase production capacity. Solution of this problem extends algebra optimal aggregation structures to systems with parametric links between elements. The practical significance of the development - efficient management of modern dynamic productions. Developed working models are examples of research.

Key words: Production function, function development, optimal aggregation, binary operator, method of the optimal aggregation.

Введение

Сегодня стабильное развитие предприятия любой отрасли базируется на интеллектуальных рабочих местах, высокотехнологическом производстве и эффективной продукции. Эта база должна непрерывно развиваться для выживания в глобальном окружении. Наличие наработанного конструкторско-технологического потенциала и эффективной структуры кадров недостаточно без решения задач оперативного оптимального распределения ресурсов между элементами производственной системы, и между производством и развитием производства для каждого элемента. Задача данной работы – разработка и исследование рабочих математических моделей для нового класса объектов – интегрированных систем «производство-развитие».

Актуальность работы обусловлена тем, что в этой области практика уверенно опережает теорию. Проведен анализ литературы в области функционирования и развития передовых производств – «Мерседес», «Тойота», «Аэробус», «ОАК» и др. в аспекте быстрого и постоянного обновления производства и продуктов производства. Проанализированы пути развития производств у таких стран как Турция, Индия, Бразилия, Китай за счёт комплексных процессов закупки эффективной продукции, оборудования и технологий её производства, строительства соответствующих предприятий и подготовки собственных кадров всех уровней. Специфическая особенность таких соглашений – продавец комплекса обязуется выручку инвестировать в развитие этого производства. Последнее означает, что возврат инвестиций продавца зависит от успешности проекта у покупателя. Отсутствует достоверная информация об использовании и эффективности каких либо программных систем оптимального управления интегрированными системами производства и развития. Не найдено по результатам поиска аналогов и прототипов, использующих методологию оптимального агрегирования. Единственный аналог относится к оптимизации больших баз данных [1].

Концепции производственные системы иерархичны. Производственные системы и их элементы рассматриваются как «технологические преобразователи» ресурсов в продукт, характеризующиеся зависимостью «затраты – выпуск» - функцией производства (ФП). Полагаем ФП ограниченными нестрогим монотонными и нестрогим положительными.

Рациональные производственные системы постулат совместности. В рациональной производственной системе выделение дополнительного ресурса любой подсистеме, по крайней мере, не ухудшает показатель выхода всей системы [2]. В рациональной производственной системе возможно устойчивое состояние, когда одновременно достигают максимума критерии всех элементов и системы в целом [2-4].

Принцип оптимальности. Оптимальное распределение ресурса между некоторой парой элементов производственной системы зависит только от величины ресурса выделенного для этих элементов [5]. Принцип оптимальности, на котором основан метод динамического программирования Беллмана,

формулюється як незалежність оптимальної траєкторії із деякої точки фазового простору від предисторії [6]. Принцип оптимальності – основа для доведення можливості декомпозиції багатовимірної задачі оптимізації в послідовність одновимірних задач.

Методологія оптимального агрегування інтегрує в себе: агрегування моделі виробничої системи в еквівалентний елемент, субоптимізацію розподілу ресурсу в підсистемах, і алгебру виробничих функцій, подібну алгебрі передаточних функцій для лінійних динамічних систем [5, 7, 8].

Постановка задачі

На рис. 1 представлена конкретизація постановки задачі. В верхній частині - логіка дослідження: виробничий елемент; виробничий елемент з підсистемою розвитку і окремими ресурсними входами; ті ж елементи з загальним ресурсним входом і блоком розподілу ресурсу; оптимальний еквівалентний виробничий елемент – мета розробки. В розробленій алгебрі оптимального агрегування паралельних, послідовних і кільцевих структур [5, 7, 8] цей еквівалентний елемент повинен бути сумісним з виробничими елементами без підсистем розвитку. Математична особливість задачі агрегування систем «виробництво – розвиток» - наявність крім ресурсних зв'язей, параметричної зв'язи між підсистемами «розвиток» і «виробництво».

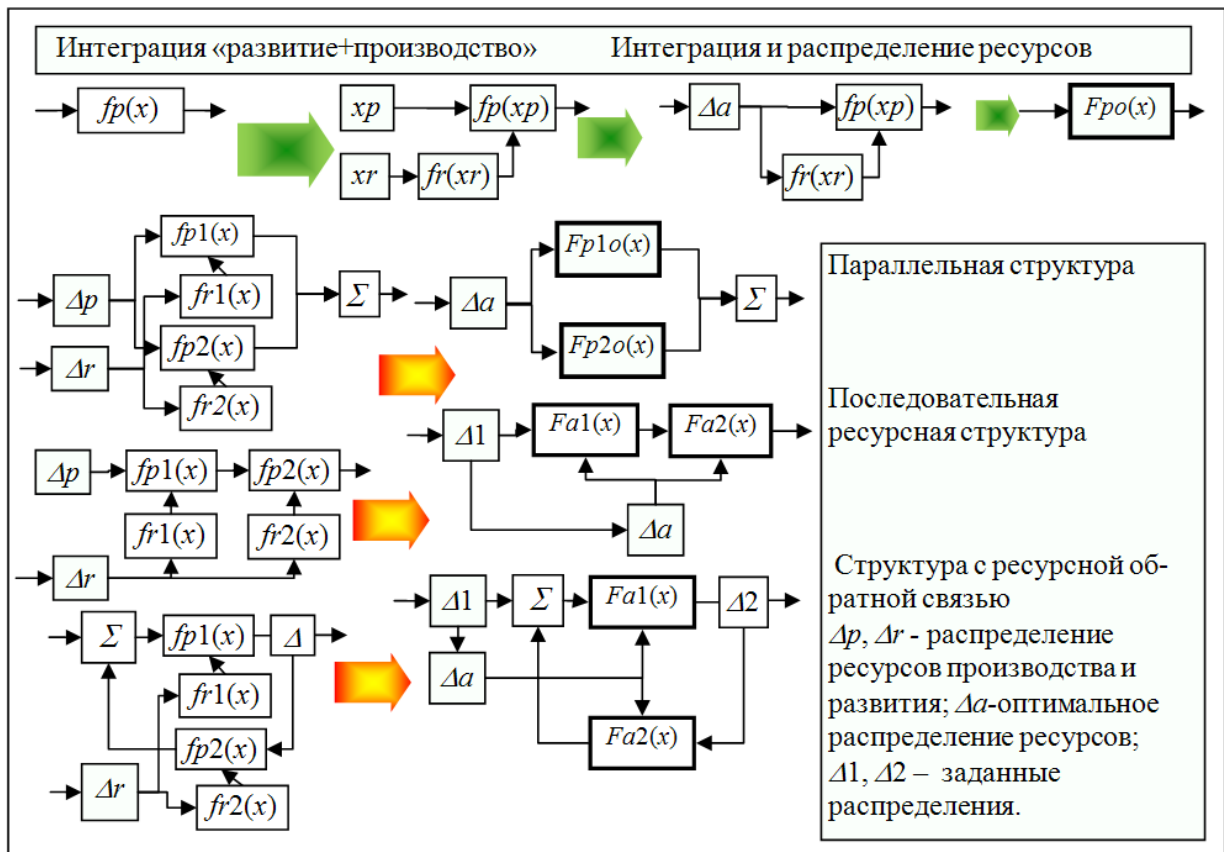


Рисунок 1 – Преобразование типовых структур ПС на базе интегрированной модели «производство – развитие»

Отсутствие прямых прототипов, внешних и собственных, обусловило разносторонний контроль каждого шага конструирования математической модели. На рис. 2 представлены отображения распределений некоторого «кванта ресурса» в приращение производства продукции при распределениях «всё в развитие», «всё в производство» и распределение в некоторой пропорции – это первый шаг контроля.

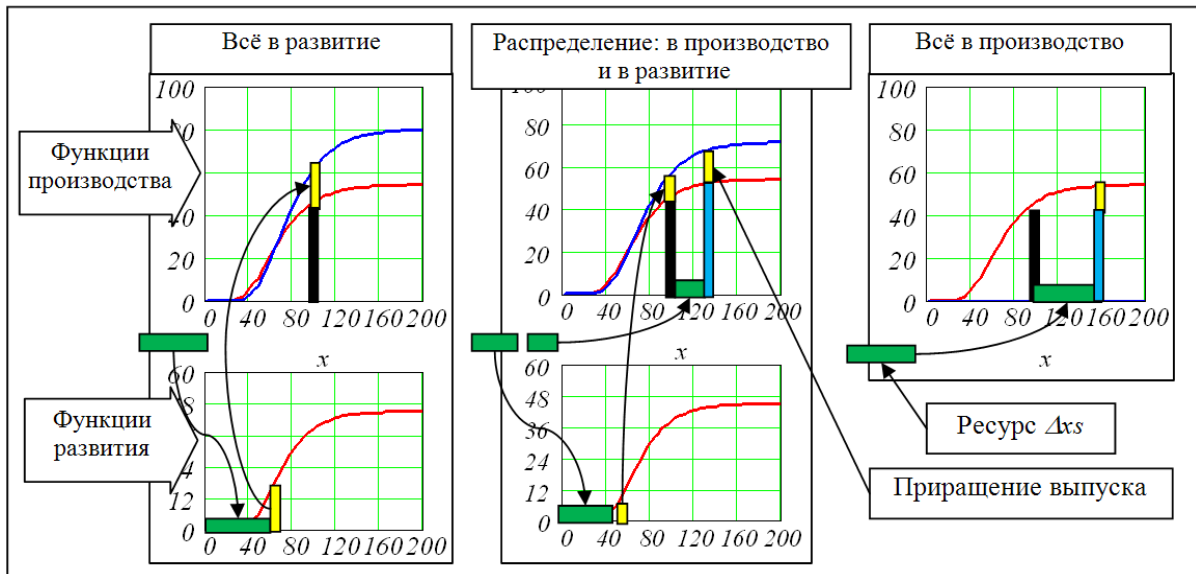


Рисунок 2 - Распределения ресурса в интегрированной системе «производство – развитие»

Информационная технология конструирования математических моделей

Поставленная задача, не может быть решена в рамках существующих стандартов научных исследований и, в частности, стандартов научных публикаций. Поэтому, по необходимости и естественным путём была создана «рациональная технология», гарантирующая получение результатов при конечных затратах. Для сравнения представим технологии в табл. 1, помня, что это не альтернативы, а технологии с различными объектами и возможностями.

Таблица 1 – Сравнительный анализ технологий конструирования математических моделей

	Технология, обусловленная ограничениями : структуры научных публикаций, формулы, ..	Технология, обусловленная возможностями математических пакетов и спецификой задач
1	Анализ авторитетных и новых аналогов и прототипов	Отслеживание релевантных публикаций и анализ перспективных направлений
2	Формулирование целей и задач исследования по улучшению аналогов	Формирование рабочей модели в среде пакета. Декомпозиция модели, контроль, вычислительные эксперименты
3	Разработка «бумажной» математической модели. Аналитическое исследование	Разработка сервисных модулей для стандартных видов анализа.
4	Выбор пакета класса «AnyShell». Вычислительные эксперименты в среде демопримера.	Исследования на «виртуальной реальности». Коррекция и развитие абстрактных математических моделей
5	Подготовка отчётов и публикаций	Подготовка «слайдовых» отчётов и публикаций (без трансляции формул в стандарт)

Существенное отличие технологий в том, что в традиционной, по сути, рассматриваются модели-аппроксимации статистических данных, а в предлагаемой – модели воспроизводящие «порождающие механизмы» поведения. Этот подход был предложен и применён Дж. Форрестером [9, 10].

Для существенно новых задач в предлагаемой технологии абстрактные математические модели обычно появляются на последних шагах исследования последовательности уточняемых имитационных моделей.

Математическую модель оптимального агрегирования систем «производство – развитие» разделяем на субмодели, соответствующие реальным подсистемам:

- модели (функции) производства и развития;
- модели отображения затрат развития в изменение функции производства;
- модели процессов функционирования и развития интегрированной системы;
- модели критерия оптимальности агрегированной системы;
- модели оптимизации распределения ресурса между производством и развитием;
- модели системы оптимальных эквивалентных функций системы для интервала начальных состояний системы.

Множественное число «модели» означает, что в процессе конструирования новых моделей для новых задач естественно возникают альтернативы, обусловленные:

- неполнотой знаний о «порождающих» механизмах» объекта;
- желанием повысить вычислительную эффективность модели;
- выбором цели и, соответственно критерия оптимизации;
- выбором математического описания процессов, например: дифференциальные, интегральные, разностные уравнения;
- специализацией модели на отрасли: агропроизводства, металлургии, электроники.

В данной работе представлена модель-фрейм, первая версия – открытая для изменений и дающая возможность накопления знаний на «виртуальной реальности».

Разделим перечисленные выше субмодели на два уровня:

- модели первого уровня, отображающие связь между величинами, для которых возможны прямые измерения;
- модели второго уровня, – отображающие связь между величинами, получаемыми в результате производных измерений.

Модели первого уровня

На рис. 3 представлен пример базовой библиотеки математических моделей производственных функций [5].

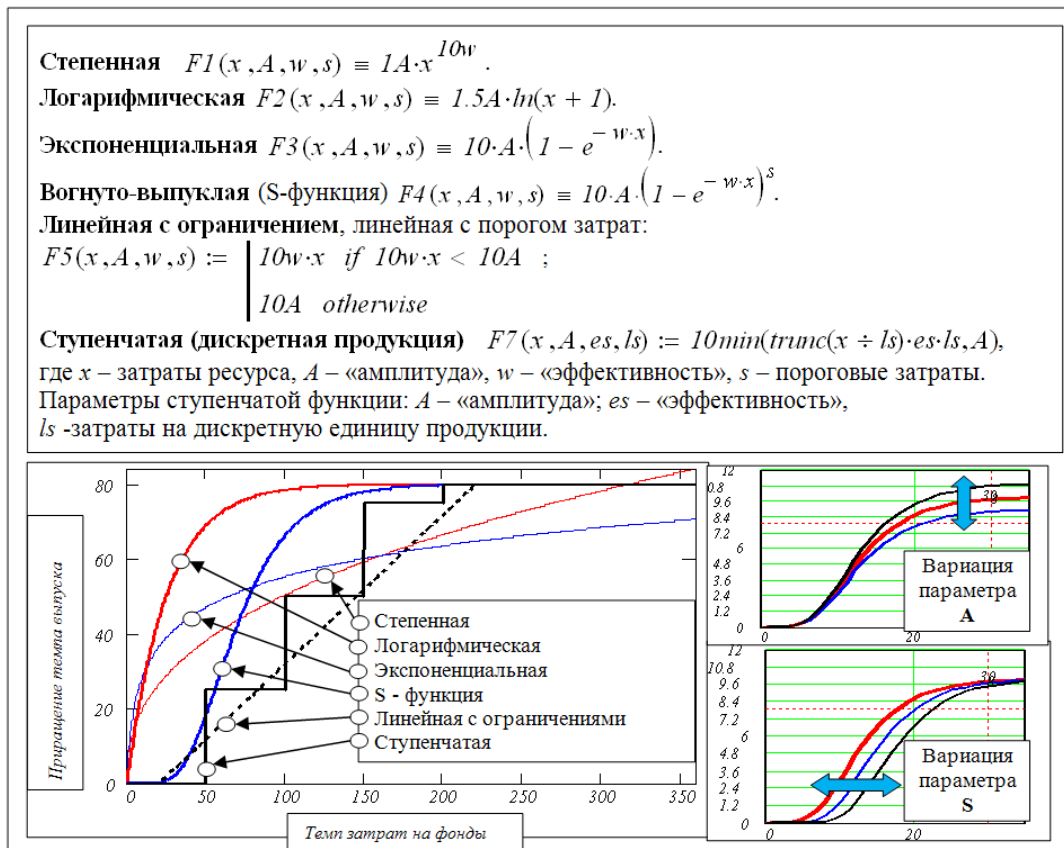


Рисунок 3 - Базовые модели функций производства. Пример

Ограничения на вид функций производства: нестрого монотонно возрастающие, нестрого положительные, ограниченные. В этом примере представлены и классические выпуклые модели, на которых построены фундаментальные направления – теория открытого управления, классическая теория рынка, теория портфеля ценных бумаг и актуальные для практики – невыпуклые, негладкие, разрывные. Эти модели сведены к трёхпараметрической форме. На рис. 4 представлен пример отображения затрат развития в параметры функции производства. Аналогами для разработки таких моделей была последняя глава в монографии Форрестера [10], обобщения экологических моделей развития («башни моделей») у М. Пешеля [11].

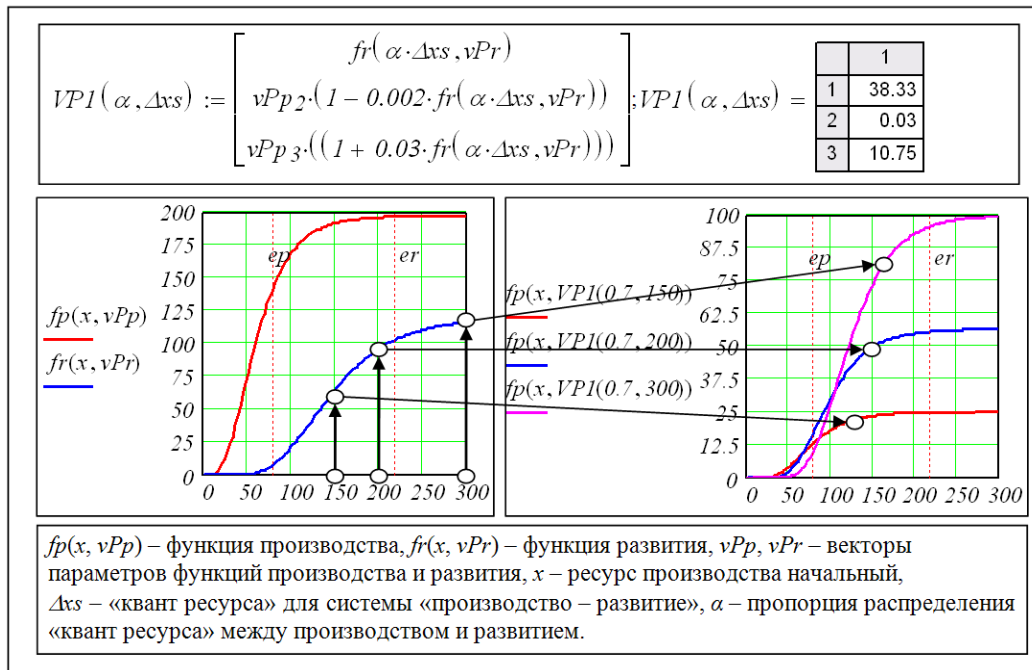


Рисунок 4 - Отображение затрат развития в функцию производства. Пример

Модели второго уровня

В среде математического пакета модели второго уровня формируются как функции пользователя. Понятия функции в математике и программировании различны. Множество функций пользователя включает в себя подмножество функций классической математики. Возникает конкретный вопрос – можно ли расширить множество функций математики за счёт программных модулей «функций пользователя»? Создатели пакета Mathcad назвали его «живой математикой». Если рассматривать подмножество функций пользователя в некоторой программной платформе – таких, что берут и возвращают числовые данные, то это, несомненно, математические функции, такие же, как экспонента и логарифм. Функции пользователя могут порождать некоторые интересные для теории и полезные для практики алгебры, подобные алгебрам чисел, матриц, передаточных функций, распределений вероятностей. Для таких нечётких и метаморфозных объектов, как функции производства (ФП), построена алгебра с оператором оптимального агрегирования, позволяющего «складывать» и «перемножать» ФП [5]. Сравним классические функции с функциями пользователя данной работы. Запишем общеизвестную функцию синус как параметризованную функцию пользователя:

$$SIN(x, vps) = SIN[x, (A \ \omega \ \phi)] = A \cdot \sin(\omega \cdot x + \phi).$$

Определим функции пользователя для элементарных операций алгебры чисел:

$$SUM(a, b) = a + b; \quad MUL(a, b) = a \cdot b.$$

Запишем выражение с бинарным оператором оптимального агрегирования, который берёт пару функций и возвращает оптимальную, по критерию суммарного производства, эквивалентную, по зависимости «вход – выход», функцию для системы из параллельно работающих элементов с ФП $f1$ и $f2$:

$$Fops(x) = f2op(f1, f2).$$

На рис. 5 представлен пример представления выбранного критерия оптимальности системы «производство - развитие» функцией пользователя четырёх переменных. Исследуется характер зависимости критерия от пропорции распределения ресурса для начальных значений затрат ресурса производства: 40, 60, 120, 250. На каждом графике даны зависимости для величин кванта ресурса 70, 150, 200. Видим, что оптимальное распределение меняется в интервале $0 < \alpha < 1$ и что возможны

монопольные распределения «всё в производство», «всё в развитие». Особенность этой функции – параметрическая связь между функциями развития и производства.

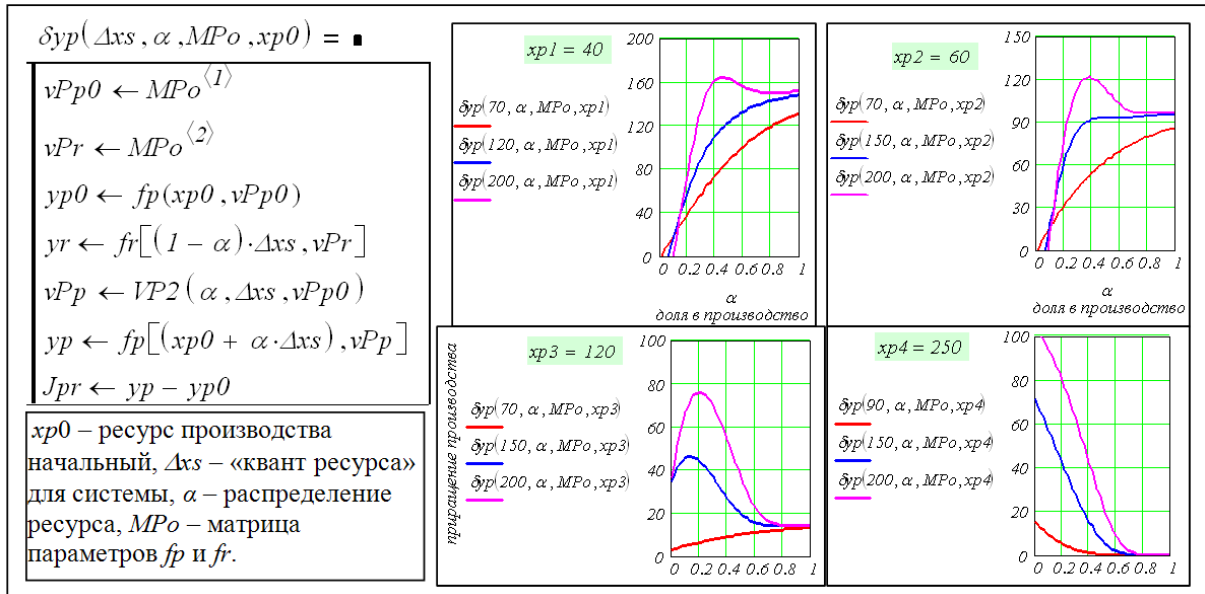


Рисунок 5 - Анализ критериальной функции системы «производство – развитие». Пример

Для исследования системы оптимальных зависимостей критерия от начального состояния $(\Delta x_s, xp_0)$ делаем модуль, который для каждой точки $(\Delta x_s, xp_0)$ находит значение α_{opt} , дающее критерию максимум (рис. 6).

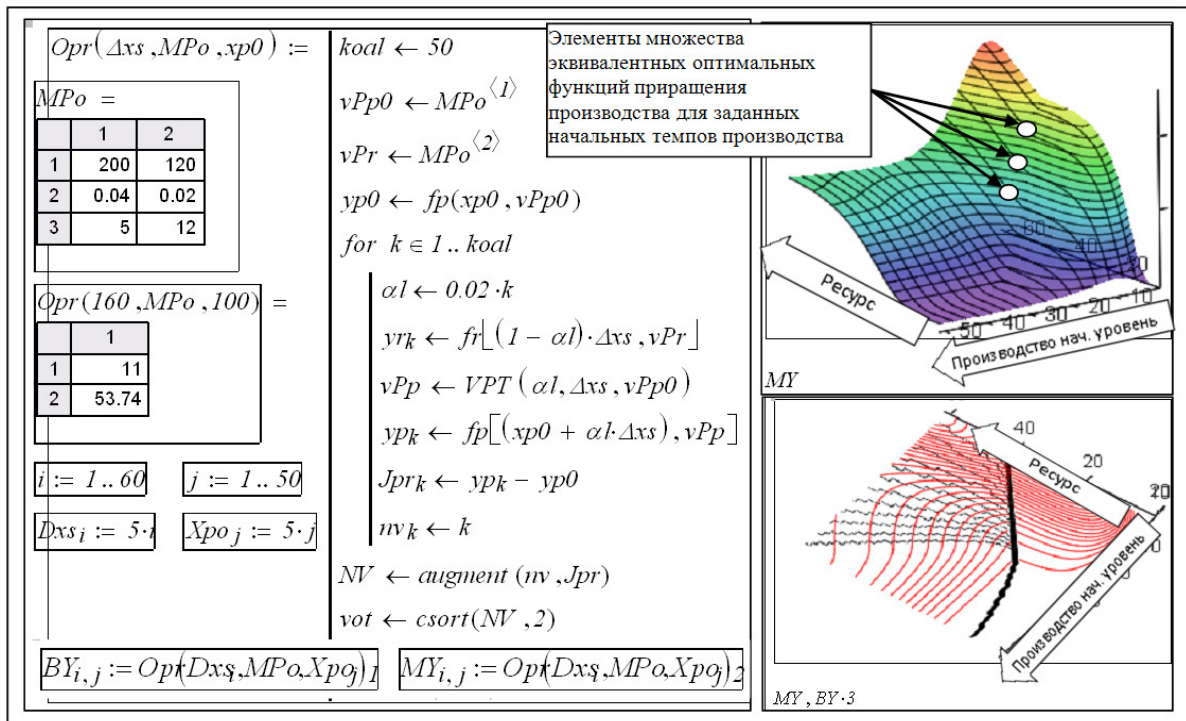


Рисунок 6 - Анализ критериальной функции системы «производство - развитие. Пример

Для производственных систем доказана и реализована возможность декомпозиции многомерной задачи нелинейного программирования в систему одномерных задач оптимизации [5]. Это позволяет использовать безотказный метод прямого перебора на базе векторизации вычислений. В тексте программы (см. рис. 6) нахождение экстремума выполняется в последних строках. На верхнем графике

представлена MU – дискретизованная зависимость максимального значения критерия от величин начального производства и кванта ресурса. На нижнем графике совместно выведены функция MU и функция BY – значения оптимального распределения кванта ресурса между производством и развитием. Трёхмерный график повернут в ракурс, позволяющий сравнить линии уровня этих функций. На графике видна линия разрыва оптимальных распределений – переход от распределения «всё в производство» к распределениям с ненулевой частью «в развитие».

Для исследования оптимальных эквивалентных функций и соответствующих функций оптимального распределения кванта ресурса при заданном значении начального темпа производства делаем программный модуль, который вычисляет зависимость критерия от $(\Delta x_s, \alpha)$ – кванта ресурса и его распределения при фиксированных значениях параметров MPO и XPO – матриц параметров системы и начального темпа производства.

На рис. 7 представлен текст версии программного модуля. В левой части приведены: - выход модуля – вектор из двух матриц; - «распаковка» выхода, где $Kripr(\cdot)$ – матрица значений критериальной функции; - матрица значений годографа максимумов критериальной функции. Ниже приведены фрагменты этих матриц с размерностями 60×50 и 60×2 .

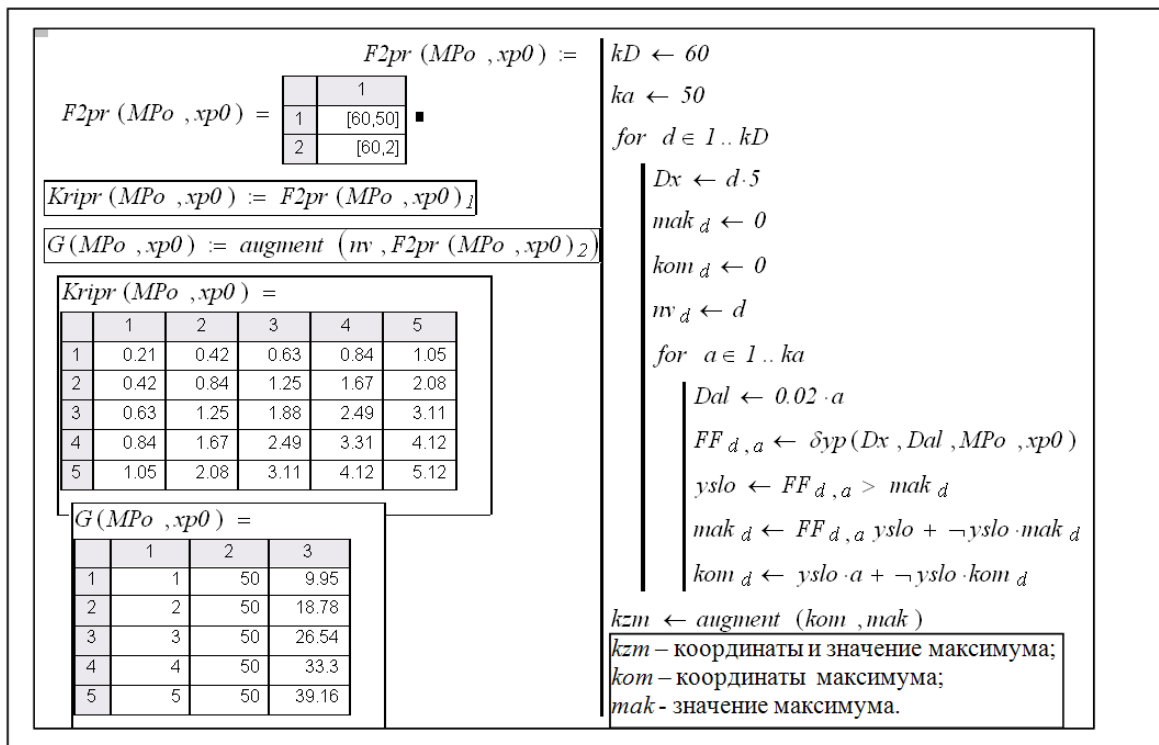


Рисунок 7 - Модуль определения оптимальной эквивалентной ФП при заданном начальном темпе ресурса производства. Пример

Результаты моделирования приведены на рис. 8. В верхнем ряду – графики критериальной функции для двух значений начального темпа производства $xp01 = 66$, $xp02 = 128$. На эти графики наложены годографы максимумов. Видим разрывность годографов. Интерпретация разрывов: при определённом значении кванта ресурса скачком меняется его распределение. В среднем ряду – проекции «вид с боку» для тех же графиков. Годографы в этой проекции – непрерывные монотонные, кусочно-гладкие зависимости оптимального приращения ФП в зависимости от величины кванта ресурса.

Нижняя пара графиков – эквивалентные оптимальные функции приращения производства для двух заданных значений начального темпа производства и требуемые оптимальные распределения ресурса.

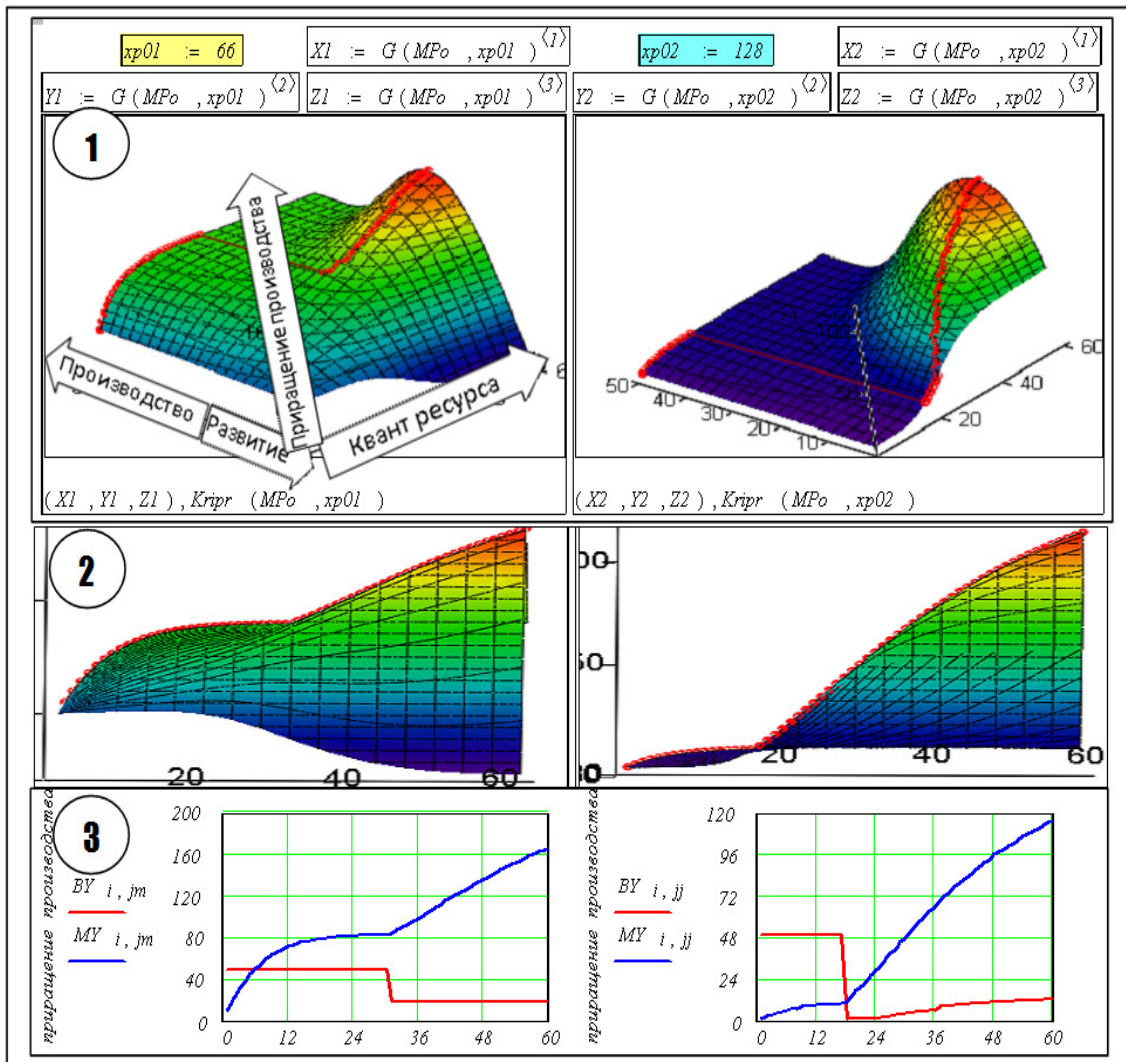


Рисунок 8 - Анализ модели оптимального агрегирования системы «производство – развитие»

Последний шаг в поставленной задаче (см. рис.1) - агрегирование параллельной структуры, элементы которой – оптимально агрегированные системы «производство - развитие».

На рис. 9 представлены результаты оптимального агрегирования двух и трёх элементов класса «производство – развитие».

В верхнем ряду представлены задачи «для математика» и для «практика». «Математик» решает две задачи – эквивалентное преобразование системы из четырёх элементов одним и оптимальное распределение ресурсов между элементами. Вместо одной задачи оптимизации функции четырёх переменных, он решает четыре одномерных задачи. Приведенная структура – это рабочая формула решения поставленной задачи, где $F2opr$ и $f2exo$ – бинарные операторы оптимального агрегирования неоднородных – «производство- развитие» и однородных «производство – производство» элементов. «Практик» должен распределить, желательно оптимально, ресурс в производственной системе. Обычно это делается сверху вниз по уровням иерархии. Метод оптимального агрегирования даёт решение для каждого элемента.

Во втором ряду представлены рабочие формулы агрегирования для систем из двух и трёх элементов и результаты вычислений по этим формулам – матрицы, первый столбец которых – значения ФП для дискретной сетки значений ресурса, другие столбцы - распределения ресурса по производственным элементам.

В третьем ряду – графики, построенные по данным этих матриц. На них представлены примеры вывода результатов.

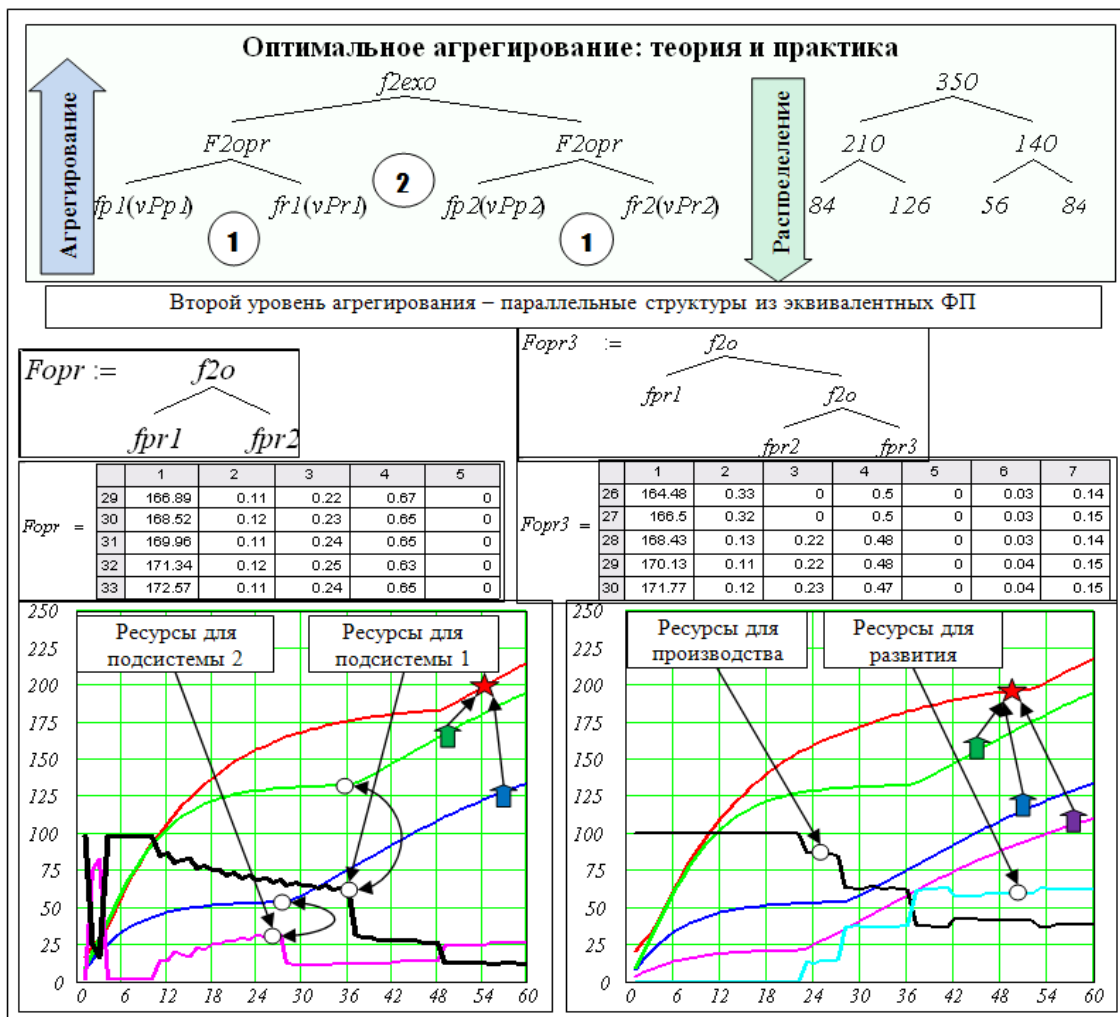


Рисунок 9 - Оптимальное агрегирование второго уровня. Примеры

На левом графике выведено распределение ресурса между двумя подсистемами класса «производство – развитие», на правом – распределение ресурса между статьями затрат «производство» и «развитие».

Видим сложный характер функций оптимального распределения ресурса и порождённый этими распределениями простой характер оптимальных эквивалентных ФП (ОЭФП): кусочно-гладких и выпуклых с одной точкой разрыва производных. Это свойство ОЭФП имеет место для систем из двух и трех элементов класса «производство – развитие». Исследования в этом направлении – темы следующих публикаций.

Выводы

На основании анализа процессов функционирования и развития современных высокотехнологических производств и существующих методов интеграции функциональных подсистем сформулирована задача оптимального управления интегрированными производственными системами.

Для решения задачи выбраны ресурсный подход и методология оптимального агрегирования. Разработана базовая рабочая модель оптимально агрегированной системы для нового класса производственных элементов «производство – развитие», пригодная для получения новых научных результатов и практического применения в качестве подсистемы АСПР.

Разработанная модель является эффективным средством для проведения научных исследований, пригодна для создания подсистемы управления процессами производства и развития предприятия. Оптимизационная задача для системы «производство – развитие» имеет множество тестуев развития – изменения технологий производства, концентрации или распределения производственных мощностей и др. Разработанная модель – параметризованная и модульная, поэтому она является эффективной основой для создания новых моделей интегрированных производств.

Список литературы

1. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Knowledge-based programs. *Distributed Computing*, 10(4):199–225, 1997.
2. Месарович М. Д. Теория иерархических многоуровневых систем / М. Д. Месарович, З. Мако, М. Такахара – М.: Мир, 1973. – 310 с.
3. Опойцев В. И. Равновесие и устойчивость в моделях коллективного поведения / В. И. Опойцев – М.: Наука, 1977. – 346 с.
4. Бурков В. Н. Большие системы: моделирование организационных механизмов / В. Н. Бурков – М.: Наука, 1989. – 246 с.
5. Боровська Т. М. Метод оптимального агрегування в оптимізаційних задачах: монографія / Т. М. Боровська, І. С. Колесник, В. А. Северілов. – Вінниця: УНІВЕРСУМ–Вінниця, 2009. – 229 с. – ISBN 978–966–641–285–3.
6. Беллман Р. Процессы регулирования с адаптацией / Р. Беллман – М.: Наука, 1964. – 317 с.
7. Боровська Т. М. Моделювання та оптимізація у менеджменті: навч. посіб. для студ. ВНЗ / Т. М. Боровська, В. А. Северілов, С. П. Бадьора, І. С. Колесник. – Вінниця: УНІВЕРСУМ – Вінниця, 2009. – 145 с. – ISBN 978–966–641–287–7.
8. Боровська Т. М. Моделювання задач управління інвестиціями: навч. посіб. для студ. ВНЗ / Т. М. Боровська, В. А. Северілов, С. П. Бадьора, І. С. Колесник. – Вінниця: ВНТУ, 2009. – 178 с. – ISBN 978–966–641–311–9.
9. Форрестер Дж. Динамика города: пер. с англ. / Дж. Форрестер – М.: Прогресс, 1974. – 276 с.
10. Форрестер Дж. Основы кибернетики предприятия: пер. с англ. / Дж. Форрестер – М.: Прогресс, 1971. – 340 с.
11. Пешель М. Моделирование сигналов и систем / М. Пешель – М.: Мир, 1981. – 286 с.

Сведения об авторах

Боровская Таиса – к.т.н., доц., доцент кафедры КСУ, Винницкий национальный технический университет, (0432)598222, Винница, Хмельницкое шоссе, 95.

Колесник Ирина – к.т.н., доц., доцент кафедры ВТ, Винницкий национальный технический университет, (0432)598379, Винница, Хмельницкое шоссе, 95.

Северілов Виктор – к.т.н., доцент, Винница.

Шульган Инна – студентка каф. КСУ, Винницкий национальный технический университет, (0432)598222, Винница, Хмельницкое шоссе, 95.

УДК 355.01:510.21

А.В. ДУДАТЬСВ

Вінницький національний технічний університет, м. Вінниця

ТЕОРЕТИЧНІ АСПЕКТИ ТА ТЕХНОЛОГІЇ КЕРОВАНОГО ХАОСУ ДЛЯ РЕАЛІЗАЦІЇ КОМПЛЕКСНОГО ІНФОРМАЦІЙНОГО ЗАХИСТУ СОЦІОТЕХНІЧНИХ СИСТЕМ

Анотація. Для побудови ефективної комплексної системи захисту інформації сучасної соціотехнічної системи необхідно вирішити дві задачі: захист власних інформаційних ресурсів та захист від інформаційного впливу конкурентів. Досвід останніх років і подій показує, що ефективність застосування інформаційної зброї у сучасних умовах інформатизації суспільства достатньо велика і за своїми кількісними показниками може бути порівняна зі зброєю масового знищення.

У статті розглянута проблема інформаційного протистояння ймовірних конкурентів, наведена аксіоматика комплексної інформаційної безпеки соціотехнічних систем на всіх етапах їх життєдіяльності. Сформульовані і доведені теореми, які формалізують поведінку двох суб'єктів інформаційної взаємодії за певними сценаріями. Розглянуті теоретичні питання у подальшому дозволять розвинути загальну теорію комплексної інформаційної безпеки та побудувати ефективну комплексну систему захисту інформації (КСЗІ).

Ключові слова: інформаційна війна, інформаційна зброя, керований хаос, комплексна система захисту інформації.

Аннотация. Для построения эффективной комплексной системы защиты информации современных социотехнических систем необходимо решить две задачи: защита собственных информационных ресурсов и защита от информационного влияния конкурентов. Опыт последних лет и событий показывает, что эффективность применения информационного оружия в современных условиях информатизации общества достаточно велика и по своим количественным показателям сопоставима с оружием массового поражения.

В статье рассмотрена проблема информационного противоборства потенциальных конкурентов, приведена аксиоматика комплексной информационной безопасности социотехнических систем на всех этапах её жизнедеятельности. Сформулированы и доказаны теоремы, которые формализуют поведение двух субъектов информационного взаимодействия по определённым сценариям. Рассмотренные вопросы позволят в дальнейшем развить общую теорию комплексной информационной безопасности и построить эффективную комплексную систему защиты информации (КСЗИ).

Ключевые слова: информационная война, информационное оружие, управляемый хаос, комплексная система защиты информации.

Annotation. To build an effective integrated systems information protection contemporary socio-technical systems it is necessary to solve two problems: protection of own information resources and protection informationog impact of competitors. The experience of recent years shows that the efficiency of use of information weapons in modern conditions of Informatization of the society is quite high and their quantitative indicators comparable to weapons of mass destruction.

The article considers the problem of information confrontation potential competitors, privedena axiomatics of complex safety sociotechnical systems at all stages of its life. Formulated and proved theorems which formalize the behavior of two subjects of information interaction in certain scenarios. Considered are the questions will allow to further develop the General theory of complex information security and to build effective integrated security system informatsii.

Key words: information warfare, information weapon, controlled chaos, complex system of information protection.

Вступ

Будь-який еволюційний процес розвитку складної системи, у тому числі сучасної соціотехнічної системи (СТС), супроводжується зміною протилежних станів – хаосу та порядку. При цьому стан хаосу супроводжується зміною або загибеллю структури системи, а стан впорядкованості системи — організацією або самоорганізацією її структури. Застосування технологій керованого хаосу, ефективність яких фахівцями порівнюється зі зброєю масового знищення, в останні десятиріччя набувають все більшого розповсюдження. Головною метою однієї із конкуруючих або конфліктуючих сторін, що використовує технології керованого хаосу, як це не парадоксально, для активного елемента інформаційної взаємодії, є створення дискретного процесу розвитку опонента і таким чином унеможливлення його суб'єктивного розвитку. Реалізація цієї мети надає можливість зробити процес інформаційного протистояння максимально керованим з боку активного елемента і ефективної реалізації спеціальних інформаційних операцій. Трансформуючи цю мету на життєдіяльність сучасних СТС, які є об'єктами інформаційної взаємодії і функціонують у конкурентному середовищі, головну мету можна сформулювати як завоювання лідерства на певному сегменті ринку шляхом дискредитації або знищення конкурентів. Зрозуміло, що досягнення цієї мети реалізується за допомогою спеціальних засобів та заходів ведення інформаційних війн.

Задача забезпечення комплексної інформаційної безпеки сучасних соціотехнічних систем складається із вирішення двох завдань: захисту своїх власних інформаційних ресурсів і захисту від інформаційного впливу конкурентів. З урахуванням того, що неналежне забезпечення тієї чи іншої складової комплексної інформаційної безпеки СТС може призвести до значних втрат, тобто ризику виникнення небажаних подій у СТС можуть бути значними і навіть критичними щодо подальшого існування СТС і різних інфраструктур, які забезпечують життєдіяльність СТС. Оскільки об'єкти або СТС критичного застосування є пріоритетами ймовірних конкурентів та кіберзлочинців, то стає зрозумілою актуальність забезпечення комплексної інформаційної безпеки саме з урахуванням вищесформульованих двох задач. Наприклад, критичними системами загальнодержавного масштабу є системи, які відносяться до оборонної, енергетичної, транспортної або хімічної галузей. Несанкціоноване втручання в роботу інформаційних управляючих систем, які використовуються у

вищезгаданих галузях, може привести до зупинки роботи цілих підприємств, соціальних і екологічних катастроф і навіть “паралічу” держави в цілому.

Актуальність

Забезпечення комплексної інформаційної безпеки сучасних СТС є неперервною за своїм змістом задачею, рішення якої дозволить забезпечити необхідний або достатній рівень захищеності інформаційних ресурсів об’єкта захисту. З урахуванням сучасних тенденцій життєвого циклу складних систем, у тому числі СТС, що характеризується використанням ефективних технологій інформаційних війн, актуальним є подальший розвиток загальної теорії комплексної інформаційної безпеки СТС, що дозволить у подальшому реалізувати ефективні механізми захисту щодо зовнішнього інформаційного впливу.

Мета

Метою даної роботи є подальший розвиток загальної теорії комплексної інформаційної безпеки соціотехнічних систем, що дозволить практично реалізувати ефективні технології і механізми захисту і забезпечити достатній рівень захищеності в умовах інформаційної війни.

Постановка задачі

1. Провести аналіз умов функціонування соціотехнічних систем.
2. Сформулювати теоретичний базис для розвитку загальної теорії комплексної інформаційної безпеки соціотехнічних систем в умовах конкурентного інформаційного середовища.

Основна частина

Повний цикл життєдіяльності СТС супроводжується взаємним впливом складових системи “людина - технологічне середовище”. Будь-яка система має свою морфологію, поведінку, самоповедінку, що породжує функціональну діяльність, відповідно до цільових функцій. Опис систем можна виконувати у декількох напрямках: функціональному, морфологічному, інформаційному тощо. Оскільки ми розглядаємо питання забезпечення комплексної інформаційної безпеки, то доцільно запропонувати підхід щодо опису системи як об’єкта захисту на всіх етапах його життєдіяльності. Отже, для кожного етапу життєдіяльності СТС були запропоновані такі аксіоми: [1]

Аксіома 1. На етапі постановки задачі комплексна безпека визначається умовами експлуатації майбутньої системи і не може бути меншою допустимого рівня.

Аксіома 2. На етапі проектування рівень безпеки забезпечується тривіальними методами, засобами та заходами, які забезпечують необхідний рівень.

Аксіома 3. На етапі реалізації системи рівень безпеки забезпечується реалізацією методів, засобів та заходів, передбачених на етапі проектування.

Аксіома 4. На етапі експлуатації безпека системи забезпечується в 2 етапи:
етап оцінювання рівня поточної безпеки;
етап забезпечення необхідного рівня безпеки.

Аксіома 5. Оцінювання рівня безпеки відбувається у конкурентному середовищі, яке включає інші антагоністичні системи.

Аксіома 6. Етап експлуатації системи може супроводжуватись її знищенням антагоністичними системами або знищенням антагоністичних систем.

Аксіома 7. Етап експлуатації системи супроводжується зміною її структури або зв’язків між елементами, що впливає на рівень її безпеки.

Аксіома 8. Кожна система має закінчення життєвого циклу.

Сформульовані аксіоми дозволяють продовжити розвиток загальної теорії комплексної інформаційної безпеки соціотехнічних систем саме з урахуванням функціонування СТС у конкурентному середовищі і проведенні спеціальних інформаційних операцій. Процес циклічності СТС представлений на рисунку 1.

З рис. 1 видно, що життєвий цикл розвитку СТС може відбуватися у двох напрямках: перший напрямок характеризує циклічність розвитку СТС, а другий напрямок демонструє можливий шлях, який закінчується знищенням системи. З точки зору конкуруючої сторони стан комплексної безпеки СТС за першим напрямком розвитку може задовольняти конкурента лише тимчасово, а другий напрямок є оптимальним, оскільки завершується знищенням конкуруючої системи.

Керований хаос за своєю природою – це технології інформаційної війни. Технології керованого хаосу (КХ) – це новий слабоконтрольований у наш час вид інформаційної зброї для встановлення контролю над відповідними об’єктами. Використання технологій КХ спрямоване на рішення двох основних задач: зменшення чисельності конкурентів, які представляють небезпеку та послаблення або знищення конкурентів. Коли інформаційна зброя знаходиться тільки у одного з протидіючих опонентів (суб’єкт активний), то зрозуміло, що стан іншого суб’єкта (пасивного) в більшості випадків є приреченим. Будь-який процес розвитку об’єкта супроводжується зміною фаз стану цього об’єкта: стан

порядку і хаосу. Ці стани не дискретні у часі, а з'єднуються між собою перехідними процесами, які, власне, і завершуються або станом впорядкованості, або хаосу.

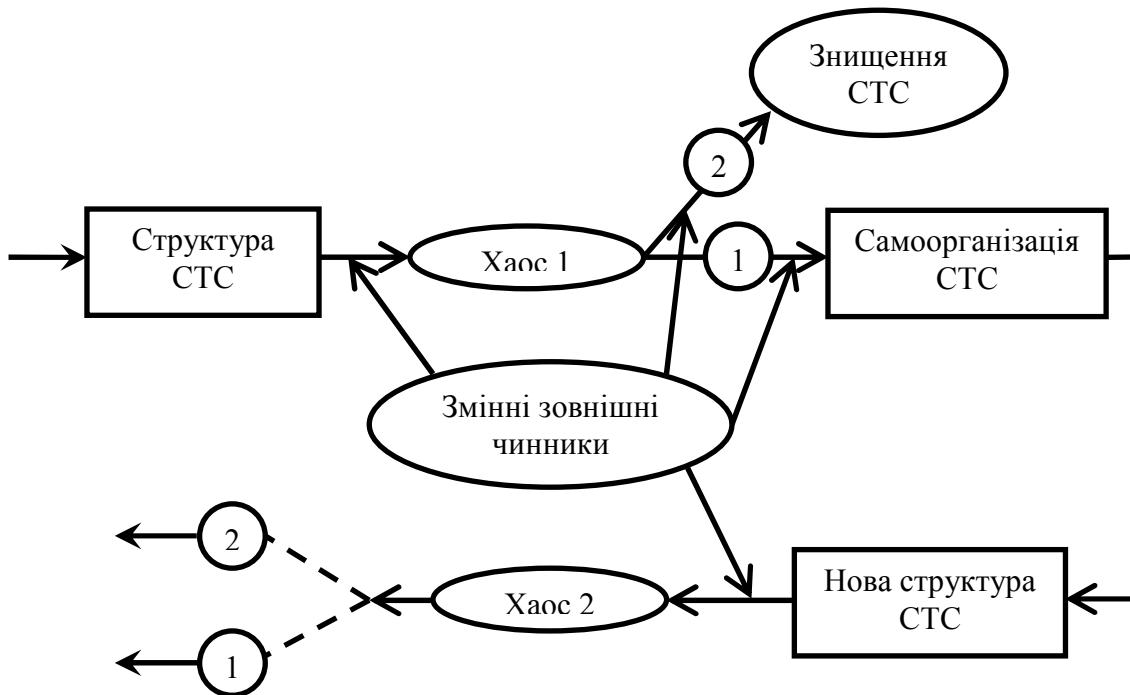


Рис. 1 – Життєвий цикл СТС

Природа хаосу може бути подвійна. Це хаос, викликаний еволюційним або суб'єктивним розвитком системи. Цей процес супроводжується зміною як структури в цілому, так і кількістю елементів та зв'язками між ними. Закінчується цей процес адаптацією системи до внутрішніх і зовнішніх змін і створенням стійкої структури. Хаос може бути викликаний також виключно зовнішнім впливом з боку конкурентів (зовнішнє середовище) з використанням спеціальних технологій ведення інформаційної війни. В будь-якому випадку, незалежно від природи хаосу, зрозуміло, що система є вразливою у режимі перехідного процесу. Саме тому спеціальні операції, які можуть бути проведені з боку конкуруючих об'єктів, є найбільш ефективними під час перехідного стану, або під час вже сформованого, або керованого хаосу. Які шляхи реалізації керованого хаосу? Ці шляхи можна узагальнити під однією назвою – «інформаційно-соціотехнічні» та ідентифікувати таким чином: нав'язування «потрібних реформ» у системі, підтримка і стимулювання запитів на штучне підвищення стандартів, знищення відповідних цінностей з метою дискредитації корпоративних секретів тощо. Реалізується керований хаос за допомогою трьох основних механізмів проведення спеціальних інформаційних операцій: механізмів реалізації пропаганди, агітації та інформаційного протистояння. Відповідно, з урахуванням можливих станів об'єктів інформаційної взаємодії (активний-пасивний) потрібно використовувати найбільш ефективні механізми проведення спеціальних інформаційних операцій. [2,3]

Виходячи з вищенаведеного і використовуючи загальносистемні визначення, сформулюємо теорему.

Теорема 1. Якщо об'єкти (суб'єкти) інформаційної протидії O_1 і O_2 взаємодіють за сценарієм «активний - пасивний», то результат проведення спеціальних інформаційних операцій для активного об'єкта може бути задовільним. $SYS_1 = \{STR_1, E_1, Z_1, P_1\}$

Доведення: Нехай об'єкт O_1 описується множиною $SYS_1 = \{STR_1, E_1, Z_1, P_1\}$, а об'єкт O_2 описується відповідно множиною $SYS_2 = \{STR_2, E_2, Z_2, P_2\}$, де STR_1, STR_2 відповідно структури для O_1 і O_2 , E_1, E_2 , відповідно кількості елементів для O_1 і O_2 , Z_1, Z_2 відповідно зв'язки для Z_1 і Z_2 , P_1 і P_2 – поведінка для O_1 і O_2 . З урахуванням того, що O_1 – пасивний і виступає лише у ролі об'єкта, а O_2 – активний об'єкт взаємодії і його можна ідентифікувати, як суб'єкта інформаційної взаємодії, тобто він проводить спеціальні операції – *OPER* над об'єктом O_1 , можемо навести:

$$SYS_1 = OPER(STR'_1, E'_1, Z'_1, P'_1),$$

де STR'_1, E'_1, Z'_1, P'_1 - змінні структури, елементи, зв'язки між елементами і поведінка об'єкта O_1 . Зміни, які відбуваються у пасивного елемента, а саме об'єкта інформаційної взаємодії можуть привести до зниження необхідного рівня комплексного інформаційного захисту, або навіть до знищення об'єкта, що задовольняє активний елемент, або суб'єкт інформаційної взаємодії.

Теорема 2. Якщо об'єкти (суб'єкти) інформаційної протидії O_1 і O_2 взаємодіють за сценарієм «активний-активний», то результат проведення спеціальних інформаційних операцій з обох боків не може бути задовільним.

Доведення: Нехай об'єкт (суб'єкт) O_1 описується множиною $SYS_1 = \{STR_1, E_1, Z_1, P_1\}$ а об'єкт O_2 (суб'єкт) описується відповідно множиною $SYS_2 = \{STR_2, E_2, Z_2, P_2\}$, де STR_1, STR_2 відповідно структури для O_1 і O_2 , E_1, E_2 , відповідно кількості елементів для O_1 і O_2 , Z_1, Z_2 відповідно зв'язки для Z_1 і Z_2 , P_1 і P_2 – поведінка для O_1 і O_2 . З урахуванням того, що O_1 – активний елемент і виступає у ролі об'єкта і суб'єкта інформаційних відносин одночасно, і O_2 – також активний елемент інформаційної взаємодії, тобто вони проводять спеціальні інформаційні операції – $OPER$ по відношенню один до одного і комплексно захищають власні інформаційні ресурси, можемо навести:

$$SYS'_1 = OPER'_2(STR'_1, E'_1, Z'_1, P'_1),$$

$$SYS'_2 = OPER'_1(STR'_2, E'_2, Z'_2, P'_2),$$

де SYS'_1 і SYS'_2 представляють змінні структури об'єктів захисту O_1 і O_2 . Це означає, що під дією спеціальних операцій, які використовували як суб'єкт O_1 , так і суб'єкт O_2 , відбулися відповідні структурні зміни як у SYS_1 , так і у SYS_2 , які не відповідають оптимальному стану елементів інформаційної протидії як об'єктів захисту.

Наведені теореми демонструють можливі шляхи розвитку сучасних СТС, які перебувають у стані інформаційної протидії і створюють теоретичне підґрунтя для розробки математичних моделей інформаційних війн. Очевидно, що ефективність інформаційного впливу визначається кількістю об'єктів, що змінили свій стан у тому напрямку, який необхідний для об'єкта впливу. Аналіз моделей інформаційних війн надасть можливість зробити попередній аналіз ризиків відносно ефективності застосування джерел та механізмів проведення спеціальних інформаційних операцій і проведення ранжування відносно важливості ймовірних інформаційних спеціальних операцій і відповідно слабких місць у системі комплексного інформаційного захисту. Іншими словами, результати аналізу дозволяють визначити найбільш небезпечні як джерела, так і механізми проведення інформаційних операцій. Це дає можливість мінімізувати можливі втрати, побудувати ефективний комплексний захист інформаційних ресурсів для об'єкта захисту шляхом використання оптимального складу засобів та заходів щодо реалізації комплексного захисту. **Приклад**

Запропоновані теоретичні аспекти загальної теорії комплексної інформаційної безпеки розглянемо на прикладі двох сценаріїв поведінки взаємодіючих об'єктів (суб'єктів): 1-й сценарій описується ситуаційною моделлю «активний-пасивний» і другий сценарій описується ситуаційною моделлю «активний-активний». Загальним для двох сценаріїв поведінки є нав'язування певних стандартів: технологічних, інформаційних, економічних, соціальних тощо. Нав'язування стандартів або моделей поведінки спрямовується, в першу чергу, до використання і доступу до різноманітних ресурсів, які для об'єкта впливу можуть бути і критичними, а це означає, що для першого типу взаємовідносин інформаційний вплив може бути оптимальним для активного об'єкта. Для другого сценарію поведінки результат проведення спеціальних інформаційних операцій може бути не оптимальним або незадовільним для обох суб'єктів інформаційної взаємодії, оскільки активні дії відбуваються з двох боків. В якості прикладу можна навести відомі ситуації з реалізацією готової продукції: якщо немає вибору – то, що продається, те і купується. Зрозуміло, що під поняттям «продукція» слід розуміти все, що має соціальний попит, у тому числі і інформаційний продукт. Альтернативний варіант – вибір у покупця є і підприємство, продукція якого не реалізується (воно переможене) або поступово завершує свою життєдіяльність, або прикладає певні зусилля для своєї самоорганізації з метою подальшого виживання.

Висновки

Запропоновані теоретичні аспекти загальної теорії комплексної інформаційної безпеки сучасних соціотехнічних систем дозволяють у певній мірі формалізувати можливі варіанти або сценарії проведення інформаційного протиборства конкуруючих сторін, що у подальшому надасть можливість організувати ефективний захист безпосередньо власних інформаційних ресурсів та реалізувати захист від інформаційного впливу конкурента.

Література

1. Дудатьєв А.В. Аксиоматика комплексної безпеки соціотехнічних систем / А.В. Дудатьєв // Інформаційні технології та комп'ютерна інженерія. – 2013. -№1(26). – С. 22-25.
2. Цыганов В.В. Интеллектуальные механизмы информационных войн / В.В. Цыганов, С.Н. Бухарин, В.В. Васин // Проблемы управления. – М., 2007. – №1. С. 25-30.
3. Дудатьєв А.В. Інформаційна безпека соціотехнічних систем в умовах інформаційної війни / А.В. Дудатьєв // Інформаційні технології та комп'ютерна інженерія. – 2011. -№3(22). – С. 75-79. .

Відомості про авторів

Дудатьєв Андрій Веніамінович – к.т.н., доцент кафедри захисту інформації, Вінницький національний технічний університет, вул.Хмельницьке шосе 95, м. Вінниця, Україна.

УДК 004.89+519.216.3+656.2

Т. О. САВЧУК, А. В. КОЗАЧУК

Вінницький національний технічний університет

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ СТАНУ ТЕХНОГЕННОЇ СИТУАЦІЇ

Анотація. В статті описані складові інформаційної технології прогнозування стану техногенної ситуації, проведено аналіз результатів використання цієї технології для прогнозування кількості втрачених пакетів у локальній мережі, а також проведено аналіз адекватності математичної моделі, що входить до складу запропонованої інформаційної технології.

Ключові слова: інформаційна технологія, імітаційне моделювання, багатовимірне представлення даних.

Аннотация. В статье описаны составляющие информационной технологии прогнозирования состояния техногенной ситуации, проведен анализ результатов использования этой технологии для прогнозирования количества потерянных пакетов в локальной сети, а также проведен анализ адекватности математической модели, входящей в состав предложенной информационной технологии.

Ключевые слова: информационная технология, имитационное моделирование, многомерное представление данных.

Abstract. The paper describes the components of information technology of forecasting of technological situation and analysis of results of usage of this technology to predict the number of lost packets on the local network, as well as an analysis of the adequacy of mathematical model, which is a part of the proposed information technology.

Key words: information technology, simulation, multi-dimensional data representation.

Вступ

Прогнозування стану техногенної ситуації є комплексним процесом, що складається з декількох етапів на яких відбувається уточнення типу техногенної ситуації, що визначає метод прогнозування стану техногенної ситуації та формування подальших рекомендацій. Перераховані етапи з'єднуються між собою інформаційними потоками, що містять вхідні та вихідні дані кожного з етапів прогнозування стану техногенної ситуації. Для систематизації процесу прогнозування стану техногенної ситуації етапи цього процесу доцільно об'єднати у технологічний ланцюжок, що дозволяє розглядати їх як інформаційну технологію [1]

Розробка інформаційної технології прогнозування стану техногенної ситуації

У відповідності з класичним визначенням [2, 3] інформаційної технології, інформаційну технологію прогнозування стану техногенної ситуації можна представити як сукупність процесів визначення значимих характеристик техногенної ситуації та пошуку кореляції між ними, внесення початкових даних про стан техногенної ситуації, вибору метода прогнозування стану техногенної ситуації та безпосереднього прогнозування її стану за допомогою моделювання гіперкубу станів техногенної ситуації або інших методів інтелектуального аналізу даних, формування рекомендацій щодо дій стосовно техногенної ситуації і програмного забезпечення прогнозування стану техногенної ситуації, об'єднані в технологічний ланцюжок з метою підвищення ефективності прогнозування стану техногенної ситуації.

Інформаційна технологія прогнозування стану техногенної ситуації складається з наступних етапів:

1. Попередня обробка інформації про техногенну ситуацію.

- a. Визначення набору значимих характеристик техногенної ситуації – виділення властивостей техногенної ситуації, можливі значення яких є відомим та таких, що можуть бути використаними для прогнозування стану техногенної ситуації.
- b. Внесення початкових даних про стан техногенної ситуації – впорядкування наявної інформації про можливі стани техногенної ситуації та внесення її у сховище даних.
- c. Пошук кореляції між характеристиками техногенної ситуації – необхідно для збільшення швидкодії алгоритмів прогнозування за рахунок зменшення розмірів структур даних.
- d. Вибір метода прогнозування стану техногенної ситуації – застосування критеріїв доцільності використання багатовимірного підходу до прогнозування стану техногенної ситуації. На основі вибору виконується один або декілька перелічених кроків:
- e. Побудова та прогнозування стану гіперкубу
- f. Підготовка даних для проведення навчання алгоритмів «Дерева прийняття рішень», «Нейронна мережа», «Наївний Баєс»

2. Прогнозування стану техногенної ситуації

1. Прогнозування за допомогою гіперкубу станів техногенної ситуації
2. Прогнозування за допомогою дерев прийняття рішень та нейронних мереж
3. Формування рекомендацій щодо дій стосовно техногенної ситуації.

Основні етапи інформаційної технології прогнозування стану техногенної ситуації зображена на рис. 1.

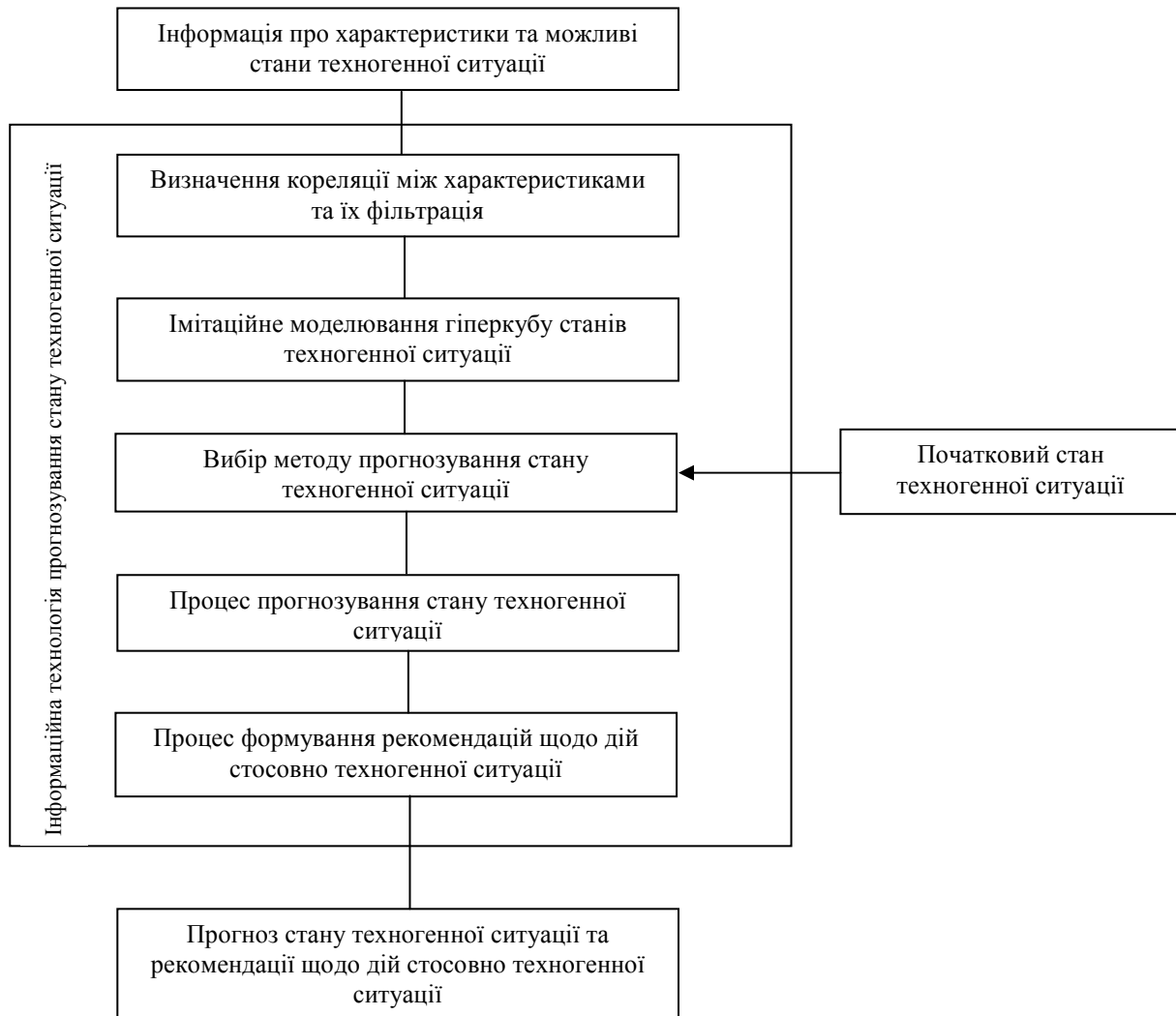


Рисунок 1 – Основні етапи інформаційної технології прогнозування стану техногенної ситуації

Аналіз результатів застосування

Дослідимо результати застосування інформаційної технології прогнозування стану техногенної ситуації для прогнозування відсотку втрачених кадрів маршрутизатором D-Link Dir300, що працює в режимі комутатора у ТОВ «СДМ Україна». Кількість втрачених кадрів залежить від таких характеристик використання комутатора як середній розмір пакета, розподілення трафіку між портами комутатора, обсяг пам'яті черги комутатора, внутрішня архітектура комутатора та максимальний розмір адресної таблиці. При цьому під час роботи комутатора можуть змінюватися такі характеристики як середній розмір кадру та трафік між портами комутатора. Модель D-Link Dir300 має чотири порти. З кожного порту може відбуватися передача інформації на будь-який з трьох інших. Таким чином, можна виділити $4 \cdot 3 = 12$ характеристик, що описують техногенну ситуації. Ще однією характеристикою є середній розмір кадру [4, 5]. Обчислення проводиться на комп'ютері з двох ядерним процесором з частотою кожного ядра 2.4 ГГц та обсягом оперативної пам'яті 4 ГБ.

Таким чином, позначивши за tr_{ij} трафік між портами i та j , можна визначити вектор характеристик техногенної ситуації W :

$$W = \{\text{середній розмір кадру; } t_{12}; t_{13}; t_{14}; t_{21}; t_{23}; t_{24}; t_{31}; t_{32}; t_{34}; t_{41}; t_{42}; t_{43}; \text{ відсоток втрачених кадрів}\}.$$

Потужність вектору W рівна 14:

$$n = 14.$$

Цільовою характеристикою техногенної ситуації, значення якої прогнозується є w_{14} – відсоток втрачених кадрів:

$$w_m = w_{14}.$$

Вектор V характеристик техногенної ситуації, що використовуються при прогнозуванні значення цільової характеристики має потужність $n-1 = 13$ та складатиметься з таких характеристик використання комутатора:

$$V = \{\text{середній розмір кадра; } t_{12}; t_{13}; t_{14}; t_{21}; t_{23}; t_{24}; t_{31}; t_{32}; t_{34}; t_{41}; t_{42}; t_{43}\}$$

Для досягнення прийнятної складності обчислень при прогнозуванні відсотку втрачених комутатором кадрів за допомогою інформаційної технології прогнозування стану техногенної ситуації, можливі діапазони значень характеристик техногенної ситуації повинні бути розбиті на групи, що описують можливі значення характеристики.

Розмір кадра було розбито на дві групи: малий (64 – 256 байт) та великий (257 – 1518 байт).

Інформацію про трафік між портами у чотирьохпортового комутатора, що використовує протокол Fast Ethernet розбито на групи в залежності від кількості інформації, що передається в одному напрямі між парою портів: слабкий (0 – 33 Мб/с), середній (34 – 66 Мб/с) та сильний (67 – 100 Мб/с).

Таким чином, гіперкуб станів техногенної ситуації, що використовується для прогнозування відсотку втрачених комутатором кадрів, містить $3^{12} * 2 = 1062882$ комірок. Така кількість комірок дозволяє проводити моделювання стану гіперкуба на персональному комп'ютері.

Для використання методу прогнозування стану техногенної ситуації за допомогою гіперкуба станів техногенної ситуації необхідно також визначити коефіцієнти c_i , що описують вплив характеристики w_i на цільову характеристику v_m . Комутатор працює у напівдуплексному режимі, при якому кожен порт може подавати інформацію або на вхід або на вихід в конкретний момент часу, що означає однаковий вплив як вхідного так і вихідного трафіка порта на кількість втрачених кадрів. Враховуючи рівнозначність всіх портів комутатора, вплив кожного з портів на кількість втрачених кадрів є рівним і не залежним від напрямку передачі даних. Розмір кадру впливає на кількість втрачених кадрів, визначаючи співвідношення корисної і службової інформації у трафіку. Загальний вплив розміру кадрів суттєво менший за вплив завантаженості каналів. Враховуючи вищезазначені твердження, можна ввести наступні значення коефіцієнтів c_i :

$$c_1 = 10; c_2 = c_3 = c_4 = \dots = c_{13} = 50.$$

Враховуючи симетричність внутрішньої будови комутатора та той факт, що кількість втрачених кадрів залежить в від сумарного потоку інформації, що надходить на кожен з портів комутатора, можна ввести альтернативний набір характеристик V_1 та розрахувати їх значення:

$$V_1 = \{\text{середній розмір кадра; } t_{вх1}; t_{вх2}; t_{вх3}; t_{вх4}; t_{вих1}; t_{вих2}; t_{вих3}; t_{вих4}\},$$

де $t_{вхi}$ – сумарний трафік, що надходить на i -й порт з інших портів комутатора, $t_{вихi}$ – сумарний трафік, що надходить на i -й порт комутатора ззовні. Значення характеристик вектору V_1 можуть бути розраховані наступним чином:

$$v(t_{вхi}) = v(t_{1i}) + v(t_{2i}) + v(t_{3i}) + v(t_{4i}), \tag{4.1}$$

$$v(t_{вихi}) = v(t_{i1}) + v(t_{i2}) + v(t_{i3}) + v(t_{i4}).$$

Вектор характеристик V_1 має меншу потужність у порівнянні з вектором V і при цьому містить таку ж кількість корисної інформації про техногенну ситуацію, що досягнуто за допомогою використання додаткових знань про предметну область, а саме – знання про факт того, що кількість пакетів, що втрачаються на порту залежить від сумарного трафіку через цей порт.

Враховуючи факт, що через особливості роботи комутатора $t_{ii} = 0$ та виходячи із співвідношення 4.1, значення величин $t_{вхi}$, можуть змінюватись в діапазоні 0 – 300 Мб/с [6], варто змінити діапазони групування цих величин на наступні: слабкий (0 – 100 Мб/с), середній (101 – 200 Мб/с) та сильний (201 – 300 Мб/с).

Варто зазначити, що внутрішній трафік маршрутизатора може перевищувати зовнішній у $m-1$ разів, де m – кількість портів комутатора. Таке збільшення трафіка зумовлене відсутністю у таблицях комутації MAC-адрес, які зазначені як адреси отримувачів пакетів, що проходять через комутатор [53]. При звичайних умовах експлуатації внутрішній трафік комутатора незначно перевищує зовнішній, це означає, що значна кількість комірок гіперкубу станів техногенної ситуації містить комірки, що з малою імовірністю будуть зустрічатися при нормальних умовах експлуатації гіперкуба. Кількість таких комірок складає 9664. Також існують повністю невизначені комірки, значення яких не може бути отримане експериментально, кількість таких комірок – 846. Загальна кількість комірок гіперкубу станів техногенної ситуації – 13122.

Для застосування вектору V_1 необхідно також ввести нові значення коефіцієнтів відстані c_{1i} . Так як природа характеристик вектору V_1 така ж, як і у вектору V , значення коефіцієнтів c_{1i} змінювати не потрібно, зміниться тільки їх кількість:

$$c_{1i} = c_i, i \in [1;9].$$

Розрахуємо значення критеріїв доцільності використання багатовимірного підходу. Експериментальним шляхом було отримано початкове значення 7442 комірок гіперкубу, з них 2610 комірок описують стани комутатора, що можуть виникнути при повсякденному його використанні.

Значення базового критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації рівне

$$Q_6 = \frac{N_{u6}}{N} = \frac{7442}{13122} = 0.567.$$

Для розрахунку значення зваженого критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації потрібно визначити величину N_{uz} ,

$$N_{uz} = \sum_{k_1=1}^{L_1} \sum_{k_2=1}^{L_2} \dots \sum_{k_{n-1}=1}^{L_{n-1}} W_{k_1, k_2, \dots, k_{n-1}} \alpha_{k_1, k_2, \dots, k_{n-1}}.$$

Прийнявши значення коефіцієнта w за 1 для комірок, що описують стан комутатора, що може бути досягнутий за нормальних умов експлуатації, та за 0.05 для комірок, що описують стан комутатора, що малоймовірно може бути досягнутий за нормальних умов експлуатації, отримуємо:

$$N_{uz} = 1 * 260 + (7442 - 2610) * 0.05 = 2851,6.$$

Відповідне значення критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації рівне

$$Q_3 = \frac{N_{uz}}{N} = \frac{2851,6}{13122} = 0,217.$$

Значення крок часу прогнозування Δt було обрано як 1 с. Таке значення кроку забезпечує достатню швидкодю та прийнятну точність прогнозування. Кількість кроків прогнозування P , таким чином, складає $250/1 = 250$ с.

Розрахунок значення зваженого критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації з урахуванням кількості кроків моделювання Q_k є недоцільним, так як кількість кроків моделювання – 250 перевищує діаметр гіперкубу. За таких умов

$$Q_k = Q_3 = 0,217.$$

Конкретні умови експлуатації, для яких проводиться прогнозування заздалегідь невідомі, тому розрахувати значення гармонічного критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації Q_7 неможливо.

Отже, на основі алгоритму вибору методу прогнозування стану техногенної ситуації слід використовувати зважений критерій доцільності використання багатовимірного підходу до прогнозування техногенної ситуації. Експертним шляхом було визначено мінімальне допустиме значення зваженого критерія доцільності використання багатовимірного підходу до прогнозування техногенної ситуації на рівні 0.15. Таким чином, для прогнозування кількості втрачених комутатором пакетів слід використовувати імітаційне моделювання гіперкубу станів техногенної ситуації.

Для визначення кількості втрачених комутатором пакетів також було застосовано метод прогнозування стану техногенної ситуації за допомогою регресійних дерев прийняття рішень. Для реалізації прогнозу було використана платформа Microsoft SQL Server Analysis Services. Експериментальні дані були записані в таблицю реляційної бази даних, кожен стовпець якої відображав одну характеристику техногенної ситуації. За допомогою DMX-запитів були створені багатовимірна модель зберігання даних (англ. Mining Structure) та модель аналізу даних (англ. Mining model), що базується на реалізації алгоритму дерев прийняття рішень під назвою «Microsoft Decision Trees Algorithm». Так як кількість характеристик техногенної ситуації – 9 не є значною, додаткових заходів щодо регулювання глибини дерева прийняття рішень не проводилось. Значення параметрів алгоритму за замовченням не змінювались.

Для наведених вище умов експлуатації було експериментально визначено кількість втрачених пакетів. Порівняння результатів прогнозування з двома методами наведено в табл. 1.

Таблиця 1 – Порівняння результатів прогнозування з результатом інформаційної технології прогнозування стану техногенної ситуації

Час, с	Кількість втрачених пакетів за 1/10 с		
	Експериментальні дані	Імітаційне моделювання	Дерева прийняття рішень
1	2	3	4
10	548	537,1624761	532,69
20	540	547,7793894	540,38
30	526	554,6942891	548,07
40	570	559,1159696	555,76

Продовження таблиці 1

1	2	3	4
50	551	561,8707638	563,45
60	560	563,5215559	571,14
70	569	564,4501175	573,82
80	560	564,913996	573,08
90	554	565,085767	572,34
100	561	565,0800752	571,6
110	560	564,9722274	570,86
120	564	564,8109515	570,12
130	567	564,6271275	569,38
140	560	564,4397463	568,64
150	553	564,2599593	567,9
160	551	564,0938191	567,16
170	539	563,9441233	566,42
180	567	563,8116459	565,68
190	562	563,6959522	564,94
200	563	563,5959314	564,2
210	570	563,5101382	563,46
220	565	563,4370081	562,72
230	565	563,3749874	561,98
240	534	563,3226078	561,24
250	550	563,2785255	560,5

Графічна інтерпретація отриманих результатів порівняння представлена на рис. 2.

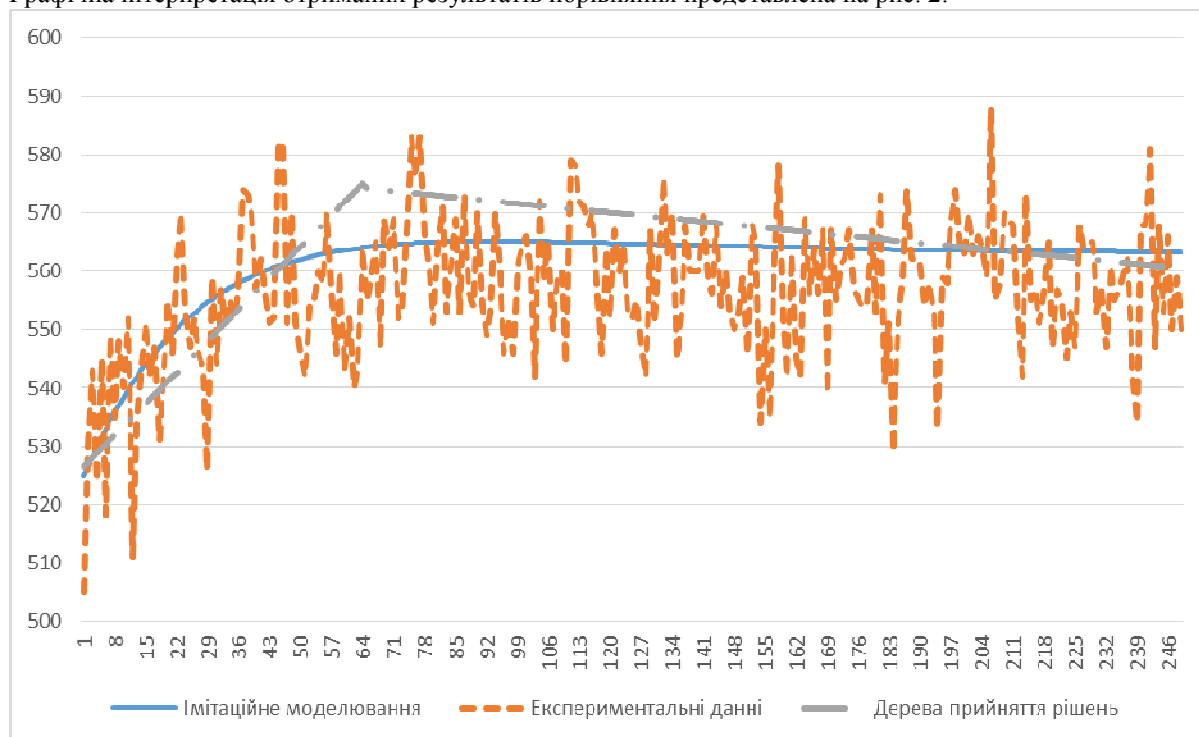


Рисунок 2 – Порівняння результатів прогнозування кількості втрачених пакетів, отриманих різними методами

На основі даних з таблиці 1 можна розрахувати середнє квадратичне відхилення результатів, отриманих методами імітаційного моделювання гіперкубу станів техногенної ситуації та регресійних дерев прийняття рішень. Значення середнього квадратичного відхилення для різних горизонтів прогнозування наведені у таблиці 2.

Таблиця 2 – Середнє квадратичне відхилення при прогнозуванні кількості втрачених пакетів

Кількість кроків прогнозування	СКВ результатів імітаційного моделювання	СКВ результатів дерев прийняття рішень
50	11,77	12,46
100	11,66	14,72
250	11,65	13,7

Таким чином, точність результату, отриманого за допомогою імітаційного моделювання гіперкубу старів техногенної ситуації вища від 7% (для коротких проміжків часу) до 21% (для середніх та довгих проміжків часу).

Оцінка адекватності моделі

Оцінимо адекватність моделі гіперкубу станів техногенної ситуації як ступінь відповідності моделі реальному об'єкту або процесу, для опису якого вона будується [8].

Модель є насиченою, так як потужність множини експериментальних даних $N=250$ перевищує кількість параметрів моделі $m=9$, що робить визначення адекватності моделі гіперкубу станів техногенної ситуації можливим.

Перевірка адекватності математичної моделі даними експерименту проводиться на основі зіставлення дисперсії відтворюваності середнього значення функції відгуку $s_2(y)$ і дисперсії адекватності. Оцінка дисперсії адекватності при $N > m$ характеризує відхилення між результатами спостережень і значеннями, формованими по функції відгуку.

Визначимо дисперсію адекватності використовуючи співвідношення

$$\sigma_a^2 = \frac{1}{N-m} \sum_{u=1}^N (y_u - y_u^*)^2,$$

де y_u – експериментальне значення кількості втрачених пакетів,

y_u^* – прогноз кількості втрачених пакетів отриманих за допомогою гіперкуба станів техногенної ситуації [9]. Підставивши дані з таблиці 4.1 в наведене співвідношення отримаємо

$$\sigma_a^2 = \frac{1}{250-9} 33671,27 = 139,71$$

Визначимо дисперсію експериментальних даних:

$$\sigma_y^2 = \frac{1}{N} \sum_{u=1}^N (y_u - \bar{y})^2,$$

де \bar{y} – математичне сподівання експериментальних даних.

$$\bar{y} = \frac{1}{N} \sum_{u=1}^N y_u = 556,58$$

$$\sigma_y^2 = \frac{1}{250} 37164,56 = 148,65$$

На основі величин σ_y^2 та σ_a^2 розрахуємо експериментальне значення F-критерія Фішера [10]:

$$F = \text{MAX} \left(\frac{\sigma_a^2}{\sigma_y^2}; \frac{\sigma_y^2}{\sigma_a^2} \right) = \text{MAX}(0,9398; 1,0639) = 1,0639.$$

Модель гіперкубу станів техногенної ситуації можна вважати адекватною, якщо значення F менше за критичне, що визначається за таблицями F-розподілу. Для рівня значимості $p=0,05$ критичне значення критерія Фішера $F_{кр}$ складає 3,18. Так як $F < F_{кр}$, модель гіперкубу станів техногенної ситуації можна вважати адекватною.

Висновки

1. Розроблено інформаційну технологію прогнозування стану техногенної ситуації, що дозволяє на основі інформації про можливі стани та початкові параметри техногенної ситуації отримати прогноз її розвитку та рекомендації про подальші дії щодо техногенної ситуації.

2. Проведено оцінку адекватності математичної моделі процесу розвитку техногенної ситуації на основі застосування F-критерія Фішера. Для рівня значимості $p=0,05$ розраховане значення критерія Фішера менше за критичне, з чого можна зробити висновок про те, що модель – адекватна.

3. Досліджено результати впровадження інформаційної технології прогнозування стану техногенної ситуації на ТОВ «СДМ Україна». Показано, що кількість втрачених пакетів на 1/10 секунди може бути зпрогнозована з середнім квадратичним відхиленням менше 15.

Список літератури

1. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы // М.: Форум, – 2011г. – 544с. – ISBN 978-5-8199-0449-7.
2. Лихачева Г.Н. Информационные технологии: Учебно-практическое пособие / Г.Н. Лихачева, М.С. Гаспариан. // М.: Изд. центр ЕАОИ. – 2007. – 189 с. – ISBN 978-5-374-00032-0.
3. Корнеев И.К. Информационные технологии. Учебник / И.К. Корнеев, Г. И. Ксандопуло, В.А. Адамович. // Проспект, 2009. – 224с. - ISBN: 978-5-482-01401-1.
4. Ашигалиев Д.У. Анализ требований к производительности коммутаторов / Ашигалиев Д.У., Амиргалиев Е.Н., Сейлова Н.А.// г. Алматы, Республика Казахстан. Научный журнал МО и Н Поиск 2010-10-16 2010, №1
5. Устранение проблем интерфейса и неполадок портов коммутатора // Cisco Systems, Inc. Режим доступа: http://www.cisco.com/cisco/web/support/RU/10/105/105430_53.pdf
6. Kuan-Ta C. Toward an Understanding of the Processing Delay of Peer-to-Peer Relay Nodes/ Kuan-Ta Chen, Jing-Kai Lou// Proceedings of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2008. Режим доступа: http://mmnet.iis.sinica.edu.tw/pub/chen08_relay.pdf
7. T. Ferrari QoS and Multiprotocol Label Switching Experiments for the Design of an ATM-Based National Network/ Tiziana Ferrari, Antonia Ghiselli, Cristina Vistoli// INET/IGC 2004 "Strengthening the Net: Building an Open and Trusted Internet"
8. Шаригін О. А. Розробка підходу до перевірки адекватності моделі прийняття рішень з нечіткими параметрами / О. А. Шаригін // Оптико-електронні інформаційно-енергетичні технології. – 2012. – № 1(23). – С. 59– 61. – ISSN 1681-7893.
9. Основы теории надежности и диагностика. Новгородский государственный университет имени Ярослава Мудрого. Режим доступа: <http://www.novsu.ru/npe/files/um/1128/umk/OTND/index.htm>
10. Красовский Г.И. Планирование эксперимента/ Красовский Г.И., Филаретов Г.Ф. // Минск: Изд-во БГУ им. В.И. Ленина, 1982. – 302 с.

Відомості про авторів

Савчук Тамара Олександрівна – к.т.н., проф., кафедри Комп’ютерних наук ВНТУ, м. Вінниця, Хмельницьке шосе 95.

Козачук Андрій Валерійович – аспірант кафедри Комп’ютерних наук ВНТУ. м. Вінниця, Хмельницьке шосе 95.

КОМП'ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

УДК 681.518.3

О.Ф. ЄНІКЄЄВ

Української державної академії залізничного транспорту, м. Харків

КОМП'ЮТЕРНА СИСТЕМА ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ АЛМАЗНОГО ШЛІФУВАННЯ В УМОВАХ НЕПОВНОЇ ІНФОРМАЦІЇ

Анотація. На основі методів безпосереднього цифрового та покоординатного управління, ієрархічного принципу та непрямих вимірів амплітуди мікронерівностей розроблено концепцію побудови комп'ютерної системи для підвищення ефективності алмазного шліфування в умовах неповної інформації. Запропоновано метод та апаратні засоби для компенсації кінематичної похибки первинних перетворювачів миттєвої швидкості. Розроблено інформаційні технології обробки сигналу миттєвої швидкості, а також поділу вкладів подач до сигналу девіацій

Ключові слова: концепція, архітектура, структурно-логічна організація, методи підвищення точності.

Аннотация. На основе методов непосредственного цифрового и покоординатного управления, иерархического принципа и косвенных измерений амплитуды микронеровностей разработана концепция построения компьютерной системы для повышения эффективности алмазного шлифования в условиях неполной информации. Предложен метод и аппаратные средства для компенсации кинематической погрешности первичных преобразователей мгновенной скорости. Разработаны метод и информационные технологии обработки сигнала мгновенной скорости, а также разделения вкладов подач в сигнал девиаций

Ключевые слова: концепция, архитектура, структурно-логическая организация, методы повышения точности.

Abstract. The article is devoted to the explanation of the new concept of the efficiency improvement of diamond grinding process and building a computer system on its basis. The architecture of this multiprocessor system is developed using the hierarchical principle, the methods of direct digital deviation control, indirect finished surface quality control and roughness feedback signal in the form of grinding wheel rate deviations.

Key word: concept, architecture, structurally logical organizations, methods of improving the accuracy.

Вступ

Впровадження до машинобудівного виробництва України сучасних інформаційних та енергозберігаючих технологій базується на розробці технічних засобів для автоматизації технологічних процесів й, зокрема, алмазного шліфування. Застосування відомих систем програмного завдання режимів шліфування орієнтовано на скорочення тривалості технологічного циклу обробки деталі при умові отримання заданої якості її обробленої поверхні. Відсутність вхідної інформації про стан поточних мікронерівностей поверхні деталі, яку оброблює шліфувальний верстат, є недоліком цих комп'ютерних систем.

Метою статті є розробка концепції побудови комп'ютерної системи для підвищення ефективності процесу алмазного шліфування на основі непрямих вимірювань мікронерівностей. Досягнення поставленої мети забезпечується розв'язуванням таких задач:

реалізація непрямого методу для вимірювань мікронерівностей та розробка інформаційної технології оцінювання прогнозованої амплітуди;

розробка структурно-логічної організації та інформаційного забезпечення системи;

реалізація інформаційно-вимірювального пристрою;

аналіз ефективності комп'ютерної системи.

Автором пропонується ідея використання девіацій швидкості обертання шліфувального круга в якості вхідного сигналу для оцінювання мікронерівностей [1]. Вхідну інформацію комп'ютерна система отримує прямими вимірюваннями миттєвої швидкості. Інформаційна технологія обробки сигналу швидкості складає такі обчислювальні процедури: виділення сигналу девіацій та подання його у вигляді обмеженого ряду Фур'є, розрахунок діючого значення та його порівняння з еталоном. Еталонні рівні девіацій визначено на основі можливих квалітетів чистоти та внесено до банку даних комп'ютерної системи. Тому, девіації є джерелом непрямой інформації про фактичні мікронерівності поверхні деталі, що оброблюється верстатом, а також сигналом, який містить в собі інформацію про відхилення прогнозованого квалітету чистоти від встановленого програмою обробки партії деталей. На основі цієї ідеї, методів безпосереднього цифрового та покоординатного управління, ієрархічного принципу розроблено концепцію побудови комп'ютерної системи для підвищення ефективності алмазного шліфування (рис. 1).

На рис. 1 позначено наступне: $\Delta\omega$ – вимірювальний сигнал, $\bar{U} = (u_1, u_2, \dots, u_i)^T$ – вектор завдання параметрів процесу алмазного шліфування, $\bar{X} = (S_{non}, S_{np}, f_{kp})^T$ – вектор поточних параметрів, $\bar{\xi} = (\xi_1, \xi_2, \dots, \xi_k)^T$ – вектор випадкових завад, R_z – амплітуда мікронерівностей,

$\bar{Q} = (\Delta_{эм}, S_{non,онм}, S_{np,онм}, f_{кр,онм})^T$ – вектор уставок. На основі обробки даних непрямих вимірювань мікронерівностей пристрій цифрової обробки сигналів задає поперечну та повздовжню подачі, а також кутову швидкість обертання шліфувального круга.

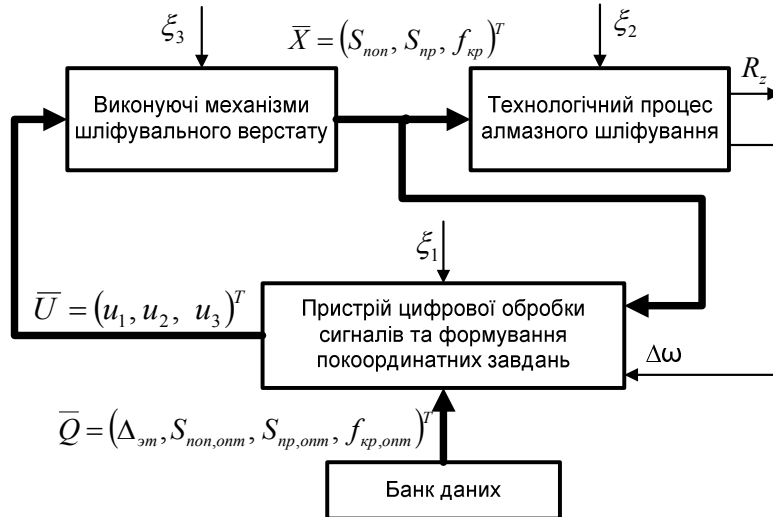


Рисунок 1 – Концепція побудови комп’ютерної системи для підвищення ефективності алмазного шліфування на основі непрямих вимірювань мікронерівностей

При розробці архітектури розподіленої комп’ютерної системи (рис. 2) знайшли застосування мікроконтролери. Її перший рівень реалізує традиційну технологію програмного завдання подачі шліфувального круга, а також середньої швидкості обертання алмазного інструменту. Реалізацію каналів системи проведено на основі вимог по швидкодії обробки даних. Завдання оптимальних режимів шліфування виконується пристроями (1 і 2) під час пауз між прямим та зворотнім ходом алмазного інструменту з урахуванням результату вимірювань девіацій. Якщо діюче значення сигналу девіацій перевищує еталонний рівень, то пристрій 4 цифрової обробки сигналу миттєвої швидкості формує на пристрої 1 та 2 сигнали зміни величин подачі.

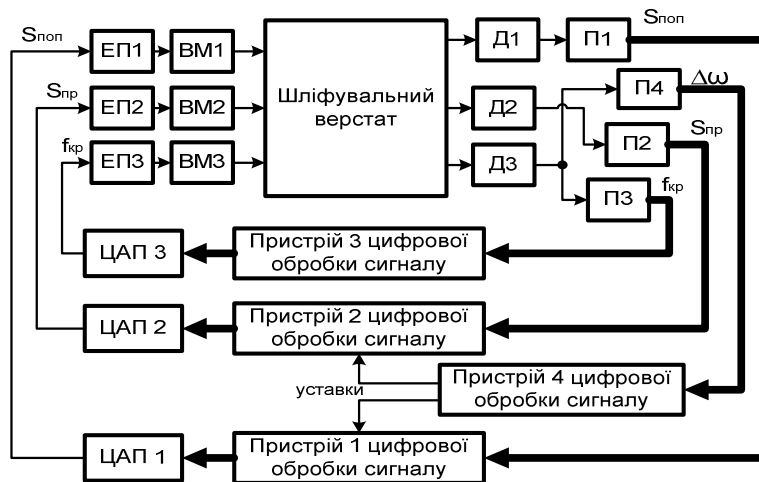


Рисунок 2 – Архітектура дворівневої розподіленої комп’ютерної системи для підвищення ефективності алмазного шліфування

Вибір періоду дискретизації інформації

Для розробки критерію задаємося значенням похибки відновлення аналогового сигналу за дискретними відліками. Сигнали розподіленої комп’ютерної системи подаємо у вигляді обмеженого ряду Фур’є, який має десять гармонійних складових

$$v(t) = \sum_{i=1}^n A_{2i-1} \sin[(2i-1)\omega t] \quad (1)$$

Втрати інформації при дискретизації за часом гармонійного сигналу оцінюються величиною динамічної похибки, яка отримана у такому вигляді

$$\delta_{дин} = 1 - \frac{\pi}{m} \sum_{k=0}^{0.5m-1} \sin\left(k \frac{2\pi}{m}\right), \quad (2)$$

де m – кількість інтервалів дискретизації.

Відносна похибка дискретизації сигналу (1) визначена з урахуванням внесків гармонійних складових

$$\delta_{сум} = \sqrt{\frac{\sum_{i=1}^n A_{2i-1}^2}{\sum_{i=1}^n A_{2i-1}^2} \delta_{(2i-1)дин}^2} \quad (3)$$

За допомогою цього виразу побудовано графік (рис. 3), який використано автором для визначення кількості інтервалів дискретизації сигналів розподіленої комп'ютерної системи. При обраній похибці відновлення апаратно-програмними засобами вхідного сигналу період його дискретизації визначається шляхом поділу часу прямої ходи шліфувального круга на встановлену кількість інтервалів. Швидкодія апаратно-програмних засобів розподіленої комп'ютерної системи повинна перевищувати це значення періоду дискретизації.

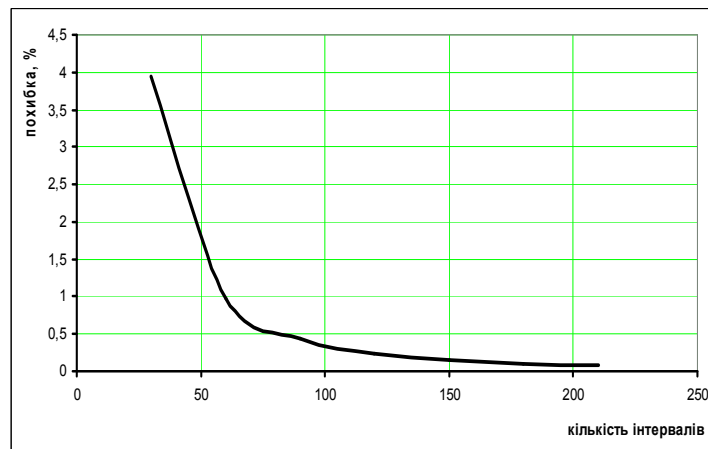


Рисунок 3 – Графік похибки дискретизації

Моделювання процесу алмазного шліфування

Вимоги до метрологічних характеристик пристрою для вимірювань миттєвої швидкості обертання шліфувального круга сформулюємо на основі аналізу вихідного математичної моделі процесу алмазного шліфування. При її розробці в якості вхідних сигналів використано подачі алмазного інструменту. Вихідним сигналом моделі є сила різання матеріалу заготівлі. З урахуванням динаміки процесу різання матеріалу заготівлі, пружних деформацій кінематичної схеми шліфувального верстату та лінійного зносу алмазного інструменту отримано вираз для передатної функції, який пов'язує поперечну подачу із радіальною складовою сили різання [1]

$$W_1(p) = \frac{K_{рез,y} (1 - e^{-pt_1})}{1 + K_{рез,y} (1 - e^{-pt_1}) \left[pK_y \left(\frac{p^2}{\Omega_m^2} + \frac{2\beta_m p}{\Omega_m} + 1 \right) + \frac{K_{ук}}{p\tau_{кр}} (1 - e^{-p\tau_{кр}}) \right]}, \quad (4)$$

де $K_{рез,y}$ – коефіцієнт передачі процесу різання матеріалу заготовки по координаті y (S_{non}), K_y – статична жорсткість конструкції верстату, β_m – коефіцієнт демпфування верстату, $K_{ук}$ – коефіцієнт зносу шліфувального круга, $\tau_{кр}$ – час обертуту круга, t_1 – час прямої або зворотної ходи круга, Ω_m – резонансна частота конструкції шліфувального верстату.

З урахуванням динаміки процесу різання матеріалу заготовки та пружних деформацій кінематичної схеми верстату отримано вираз для передатної функції, який пов’язує поздовжню подачу із тангенціальною складовою сили різання [1]

$$W_2(p) = \frac{K_x K_{рез,x} (1 - e^{-pt_1})}{p [K_x + K_{рез,x} (1 - e^{-pt_1})]} \quad (5)$$

де $K_{рез,x}$ – коефіцієнт передачі процесу різання матеріалу заготовки по координаті x (S_{np}), K_x – статична жорсткість конструкції верстату.

Вхідний сигнал з урахуванням прямої та зворотної ходи шліфувального круга подано таким виразом (для поздовжньої подачі подання аналогічне)

$$S_{non}(t) = \begin{cases} 0, & \text{якщо } -\pi < \Omega t < -\pi + \alpha, -\alpha < \Omega t < \alpha, \pi - \alpha < \Omega t < \pi, \\ S_{non}, & \text{якщо } \alpha < \Omega t < \pi\alpha, \\ -S_{non}, & \text{якщо } t\pi + \alpha < \Omega t < t\alpha, \alpha > 0, 0 < \alpha < 0.5\pi. \end{cases} \quad (6)$$

Радіальна та тангенціальна складові сили різання матеріалу заготовки ортогональні за впливом на шліфувальний круг. На підставі лінійності перетворення Лапласа та з урахуванням теореми додавання отримано таке рівняння для їхнього підсумовування

$$F(p) = \sqrt{W_1^2(p)S_{non}^2(p) + W_2^2(p)S_{np}^2(p)} \quad (7)$$

Сила різання матеріалу заготовки є джерелом крутильних коливань. Вал із шліфувальним кругом подано у вигляді механічної системи з одним ступенем волі. Рух круга описується диференціальним рівнянням другого порядку

$$J_{кр} \varphi''(t) + \beta_1 \varphi'(t) + \gamma_1 \varphi(t) = R_{\text{ед}} F(t) \quad (8)$$

де $J_{\text{ед}}$ – момент інерції круга, β_1 – коефіцієнт демпфування, γ_1 – крутильна жорсткість валу.

Після виконання процедури нормалізації та перетворень отримана передатна функція, яка пов’язує девіації та силу різання матеріалу заготовки

$$W_3(p) = \frac{R_{кр} t_1 \gamma_1 p}{J_{кр} t_1 p^2 + \beta_1 t_1 p + \gamma_1} \quad (9)$$

За виразами (4), (5), (7) та (9) складена структурна схема процесу алмазного шліфування (рис. 4). Імітаційним моделюванням отримано графік девіацій швидкості обертання шліфувального круга у межах його одного обертуту (рис. 5). Встановлено, що амплітуда девіацій не перевищує 0.05 амплітуди сигналу миттєвої швидкості. Тому процедура вимірювань сигналу девіацій потребує розробки високоточних методів та відповідних технічних засобів.

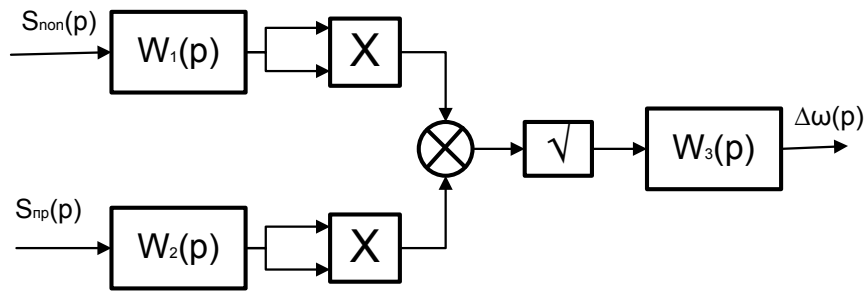


Рисунок 4 – Структурна схема процесу алмазного шліфування

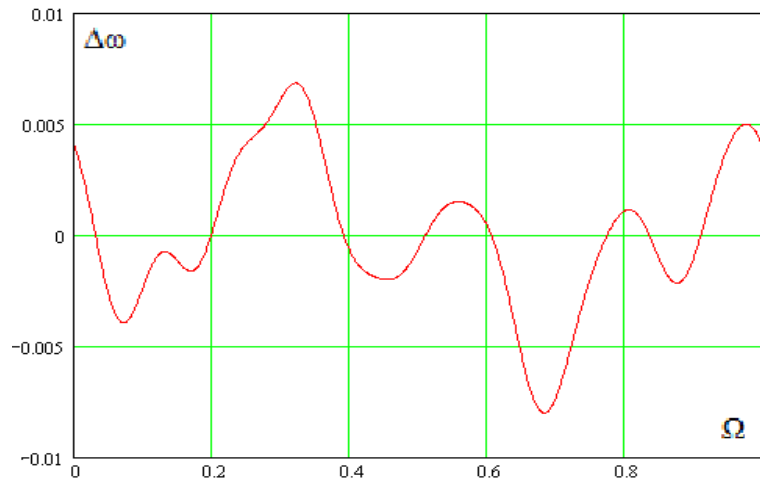


Рисунок 5 – Вихідний сигнал моделі процесу алмазного шліфування

Інформаційно-вимірювальний пристрій

Його розробку почато з аналізу складових похибки первинного перетворювача:

- похибка відновлення аналогового сигналу за дискретними відліками, яка визначається для його частотного подання таким чином

$$\delta_{\epsilon} = \frac{t_0}{8} \sqrt{0.1 \sum_{i=1}^{10} \Omega_i^4} 100\% ; \quad (10)$$

- кінематична похибка первинного перетворювача, яку визначено на прикладі досліджень допусків на виготовлення зубчастих коліс різного класу точності [2] та подано у табл. 1. При оцінюванні кінематичної похибки застосовано такий вираз

$$\delta_k = \frac{\Delta m}{m} 100\% , \quad (11)$$

де Δm – допуск на кінематичну похибку зубчастого колеса, m – модуль;

- динамічна похибка первинного перетворювача.

Таблиця 1 – Чисельні дані кінематичної похибки

Клас точності зубчастого колеса	4	5	6	7	8
$\delta_k, \%$	0.80	1.25	2.00	2.75	4.00
Похибка вимірювань девіацій, %	16.0	25.0	40.0	55.0	80.0

Данні табл. 1 дозволяють зробити висновок: проблема прецизійних вимірювань девіацій полягає в розробці технічних засобів для зменшення величини кінематичної похибки виготовлення первинних перетворювачів. Один з можливих шляхів розв'язування цієї задачі є виконання паспортизації

первинного перетворювача. У результаті цього отримуємо поправки, які враховує вимірювальний перетворювач при визначенні сигналу девіацій. До складу вимірювального перетворювача також входить пристрій синхронізації, який має відповідний первинний перетворювач.

Для компенсації кінематичної похибки автор пропонує метод багатоканальних вимірювань інтервалів часу, які формуються одною рисою первинного перетворювача та відповідають повному оберту його вала. При такому методі вимірювань миттєвої швидкості кінематична похибка виготовлення первинного перетворювача не впливає на тривалість сформованих інтервалів часу. Данні табл. 1 та цей висновок доводять коректність запропонованого методу.

Динамічна похибка первинного перетворювача являє собою зрушення за часом дискретних відліків часової реалізації девіацій, які виникають як наслідок кінематичної. Проведено оцінювання динамічної похибки зрушень за часом дискретних відліків сигналу девіацій. Абсолютне значення динамічної похибки визначається наступним чином

$$\Delta_{дин} = \frac{1}{z} \sum_{i=0}^{z-1} [\Delta\omega(i\Delta t + \sigma_{см}) - \Delta\omega(i\Delta t)], \quad (12)$$

де z – кількість рисок первинного перетворювача.

З точністю до величин другого порядку меншості відносно динамічної похибки первинного перетворювача підсумок замінюємо інтегралом

$$\Delta_{дин} = \frac{1}{T} \int_0^T [\Delta\omega(\Omega t + \Omega\sigma_{см}) - \Delta\omega(\Omega t)] dt, \quad (13)$$

де $\sigma_{\tilde{n}i}$ – середнє квадратичне відхилення часового зсуву дискретних відліків, яке визначено на підставі допусків виготовлення первинного перетворювача.

Для гармонійного сигналу вираз (13) після математичних перетворень набуває наступного вигляду

$$\Delta_{дин,i} = \frac{2A_i}{\pi} \sin(\varphi_i - 0.5i\Omega\sigma_{см}) \sin(i\Omega\sigma_{см}). \quad (14)$$

Відносну похибку гармонійного сигналу отримано у такому вигляді

$$\delta_i = \frac{2}{\pi} \sin(\varphi_i - 0.5i\Omega\sigma_{см}) \sin(i\Omega\sigma_{см}). \quad (15)$$

Динамічну похибку первинного перетворювача визначимо з урахуванням внесків гармонійних складових до сигналу девіацій

$$\delta_{дин} = \sqrt{\frac{\sum_{i=1}^{10} (A_i \delta_i)^2}{\sum_{i=1}^{10} A_i^2}}. \quad (16)$$

Результати розрахунків динамічної похибки первинного перетворювача подано у табл. 2.

Таблиця 2 – Чисельні дані динамічної похибки

Клас точності первинного перетворювача	4	5	6	7	8
Похибка вимірювань девіацій, %	3.1	4.8	7.4	9.9	14.1

Сутність методу, який пропонується автором для апаратної компенсації кінематичної похибки первинного перетворювача, полягає у наступному. Вихідний сигнал первинного перетворювача за допомогою лічильника та дешифратора перетворюється у декілька імпульсних послідовностей, які відповідають моментам проходження біля чутливого елемента датчика однієї риски та подаються на вхід

відповідного каналу для вимірювань інтервалів часу. Кількість цих каналів у складі пристрою визначається кількістю рисок первинного перетворювача. Апаратну реалізацію каналів для вимірювань проведено на основі методу дискретизації за часом. Усунення взаємних накладань вихідних сигналів каналів при їхньому поєднанні у сигнал вимірювальної інформації для пристрою цифрової обробки виконується за допомогою лічильників. Об'єм останніх та частота взірцевого генератора обираються таким чином, щоб переповнення лічильника виконувалося за час трохи менший ніж середній період імпульсної послідовності. При цьому із вимірювальної інформації кожного каналу виключається калібрований за тривалістю проміжок часу. Поєднання вихідних сигналів каналів в сигнал вимірювальної інформації виконується за допомогою схеми АБО.

На підставі цього методу розроблено оригінальний пристрій для вимірювань миттєвої швидкості обертання шліфувального круга (рис. 6), який описано в роботі [3].

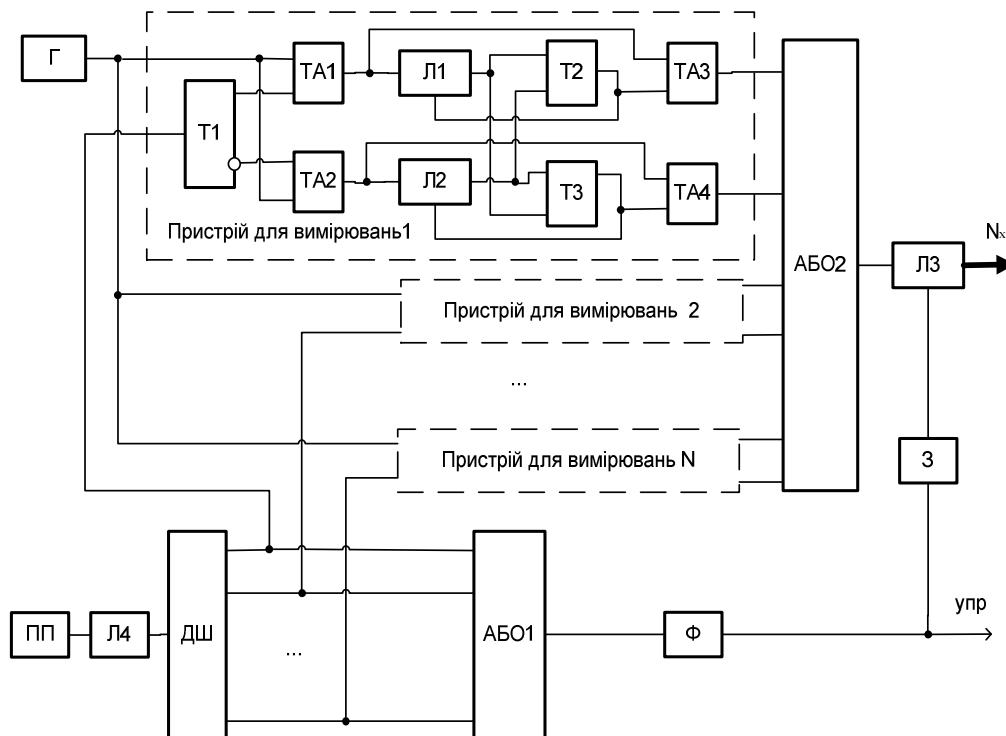


Рисунок 6 – Пристрій компенсації кінематичної похибки

На рис. 6 позначено: ПП – первинний перетворювач; ДШ – дешифратор; Г – взірцевий генератор; ТА1, ..., ТА4 – логічні схеми ТА; Nx – вихідний код; N – кількість каналів для вимірювань; АБО1, АБО2 – логічні схеми АБО; Ф – формувач імпульсів; Т1, ..., Т3 – тригери; Л1, ..., Л4 – лічильники імпульсів; 3 – лінія затримки. Пристрій для вимірювань має два канали, які вимірюють тривалість імпульсів прямого та інверсного виходів Т1. Сигнал вимірювальної інформації для пристрою 5 формує схема АБО2. Лічильник Л3 перетворює цей сигнал у цифровий код. Блок Ф формує сигнал запису коду у регістр пристрою 5 та за допомогою блоку 3 переводить лічильник Л3 до нульового стану.

Інформаційна технологія обробки сигналу девіацій

Поділ внесків подач до часової реалізації девіацій швидкості обертання круга розроблено на основі його частотного подання. Імітаційним моделюванням отримано графіки внесків подач до сигналу девіацій (рис. 7). Ці графіки отримано з моделі процесу алмазного шліфування таким завданням подач:

$S_{non}(t) = 0$, $S_{np}(t) = 0$. Визначено вагові коефіцієнти поділу внесків подач до сигналу девіацій

$$\gamma_{non} = \frac{\Delta\omega_{non}}{\Delta\omega} = 0.81 \quad \text{та} \quad \gamma_{np} = \frac{\Delta\omega_{np}}{\Delta\omega} = 0.586.$$

На основі допускового контролю розроблено алгоритм формування пристроєм 4 сигналів для відповідних пристроїв 1 та 2 цифрової обробки сигналів:

- часову реалізацію сигналу девіацій після нормалізації його параметрів подаємо обмеженим рядом Фур'є;

- визначаємо діюче значення сигналу девіацій;
- нормовані значення подач одержуємо шляхом множення діючого значення сигналу девіацій на відповідний ваговий коефіцієнт;
- аналізуємо банк даних про еталонні рівні амплітуд девіацій й, якщо буде потреба, виконуємо коректування величин подач.

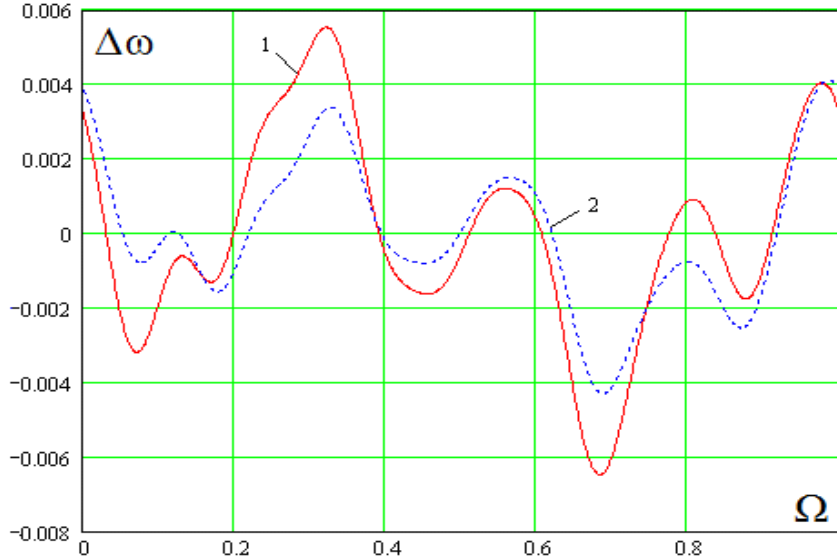


Рисунок 7 – Графіки внесків подач до сигналу девіацій:
1 – поперечної подачі, 2 – поздовжньої

Алгоритми завдання подач розроблено з урахуванням обмежень, які накладає умова отримання прогнозованих мікронерівностей поверхні деталі. Ці обмеження закладено в наступний алгоритм програмного завдання пристроєм 1 величини поперечної подачі шліфувального круга:

- програма обробки партії деталей задає припустиме значення мікронерівностей R_z поверхні деталі, що оброблюється шліфувальним верстатом;
- пристрій 1 цифрової обробки сигналу на підставі аналізу банку даних обирає Q_1 та задає чорновий цикл обробки поверхні заготівлі $S_{i\ddot{u}i} = Q_1 S_{i\ddot{u}i, m}$;

• він періодично після виконання шліфувальним кругом одного проходу обчислює поточний припуск обробки $h = h_i - k S_{i\ddot{u}i}$, де $k = 0, 1, 2, \dots$;

якщо $h_i - k S_{i\ddot{u}i} \leq h_k$, то пристрій 1 задає чистовий цикл обробки поверхні заготівлі у відповідності з виразом

$$S_{non} = Q_1 \left[S_{non, k} + (S_{non, m} - S_{non, k}) \frac{h}{h_k} \right].$$

На основі алгоритму програмного завдання поперечної подачі круга пристроєм 1, а також такої моделі виконуючого механізму

$$U(t) = b_1 \left[e(t) + T_{\partial 1} \frac{de(t)}{dt} + T_{\partial 2}^2 \frac{d^2 e(t)}{dt^2} + T_{\partial 3}^3 \frac{d^3 e(t)}{dt^3} \right], \quad (17)$$

після перетворень отримано позиційний алгоритм її завдання в наступному вигляді

$$U_n = U_{n-1} + b_1 \left(1 + \frac{T_{\partial 1}}{T_0} \right) e_n - b_1 \left(1 + \frac{2T_{\partial 1}}{T_0} \right) e_{n-1} + \frac{b_1 T_{\partial 1}}{T_0} e_{n-2}. \quad (18)$$

В основу розробки алгоритму завдання поздовжньої подачі шліфувального круга покладено те, що вона за своєю сутністю є швидкість руху виконуючого механізму. У результаті математичних перетворень швидкісний алгоритм завдання цієї подачі отримано в такому вигляді

$$U_n = U_{n-1} + K_p \left(\frac{T_0}{2T_n} + \frac{T_\partial}{T_0} \right) e_n + K_p \left(\frac{T_0}{2T_n} - \frac{2T_\partial}{T_0} \right) e_{n-1} + K_p \frac{T_\partial}{T_0} e_{n-2} \quad (19)$$

Функціонування проблемно-орієнтованої розподіленої комп'ютерної системи на основі розробленого програмного забезпечення організовано за допомогою середовища реального часу із перериваннями.

Аналіз ефективності системи

Канали першого рівня розподіленої комп'ютерної системи оперують з трьома вхідними сигналами. Причому час обробки даних подач та швидкості співвідносяться наступним чином: $t_{non} > t_{np} > t_{кр}$. Загальний час обробки даних цими апаратно-програмними засобами можна визначити за допомогою такого виразу

$$T_\Sigma = T_{кр} + \Delta T_{np} + \Delta T_{non} \quad (20)$$

Складова $T_{кр}$ визначається наступним чином [4]

$$T_{кр} = 0.5(T_{\max} + T_{\min}) \quad (21)$$

Складова ΔT_{np} визначається за допомогою такого виразу [4]

$$\Delta T_{np} = \int_{T_{\min}}^{T_{\max}} p(t_{кр}) \int_{T_{кр}}^{T_{\max}} (T_{np} - T_{кр}) p(t_{np}) dt_{np} dt_{кр} \quad (22)$$

При рівномірній щільності розподілу маємо

$$\Delta T_{np} = \frac{T_{\max} \left(\frac{T_{np}}{T_{кр}} - \frac{T_{\min}}{T_{\max}} \right)^3}{6 \frac{T_{np}}{T_{кр}} \left(1 - \frac{T_{\min}}{T_{\max}} \right)^2} \quad (23)$$

Складова ΔT_{non} визначається за допомогою наступного виразу [4]

$$\Delta T_{non} = \int_{T_{\min}}^{T_{\max}} p(t_{кр}) \int_{T_{кр}}^{T_{\max}} p(t_{np}) \int_{T_{np}}^{T_{\max}} (T_{non} - T_{np}) p(t_{non}) dt_{non} dt_{np} dt_{кр} \quad (24)$$

При рівномірній щільності розподілу маємо

$$\Delta T_{non} = \frac{T_{\max} \left(\frac{T_{non}}{T_{кр}} - \frac{T_{\min}}{T_{\max}} \right)^4}{12 \frac{T_{np}}{T_{кр}} \frac{T_{non}}{T_{кр}} \left(1 - \frac{T_{\min}}{T_{\max}} \right)^3} \quad (25)$$

Підсумувавши час виконання кожної команди керуючої підпрограми, яка задає швидкість обертання алмазного інструменту, отримуємо Ткр. Загальний час реалізації каналами першого рівня розподіленої комп'ютерної системи операцій по завданню режимів шліфування склав 0.066 с. Цей час задовольняє вимогам розподіленої комп'ютерної системи по швидкодії обробки даних.

Висновок. У статті запропонована концепція побудови розподіленої комп'ютерної системи, яка відрізняється дворівневою структурно-логічною організацією, використанням непрямих вимірювань мікронерівностей, а також покоординатним завданням програмних рухів виконуючих механізмів верстату по підвищенню ефективності процесу алмазного шліфування. Уперше запропоновано використати сигнал девіацій швидкості обертання шліфувального круга для непрямого оцінювання прогнозованих мікронерівностей поверхні деталі. Архітектура проблемно-орієнтованої розподіленої комп'ютерної системи реалізує комбінований принцип, який відрізняється застосуванням сигналів девіацій, подач та швидкості обертання алмазного інструменту. Розроблено інформаційні технології цифрової обробки вхідних сигналів та база даних комп'ютерної системи.

Вдосконалено математичну модель процесу алмазного шліфування, яка відрізняється двома координатами, використанням подач алмазного інструменту у якості вхідних сигналів та девіацій у якості вихідного. На основі частотного подання сигналів уперше розроблено метод та інформаційну технологію поділу внесків подач шліфувального круга до сигналу девіацій. Уперше запропоновано метод вимірювань миттєвої швидкості обертання шліфувального круга, який відрізняється застосування апаратних засобів компенсації кінематичної похибки первинного перетворювача. Розроблено інформаційно-вимірювальний пристрій та технологію цифрової обробки сигналу миттєвої швидкості. Встановлено ефективність розподіленої комп'ютерної системи по завданню оптимальних режимів алмазного шліфування.

Список літератури:

1. Еникеев А.Ф. Оптимальное управление технологическим процессом алмазного шлифования. – Краматорск: ДГМА, 2001. – 160 с.
2. Допуски и посадки. Справочник. / [Гл. ред. Мягков В.Д.] – Л.: Машиностроение, Часть 2, 1978. – с. 545–1032.
3. А.с. № 1538679 (СССР). Устройство для контроля неравномерности вращения вала / Марченко Б.Г., Борисенко А.Н., Еникеев А.Ф.
4. Сбор данных в управляющих вычислительных системах / Краус М., Кучбах Э., Вошни О.-Г. –М.: Мир, 1987. – 294 с.

Інформація про автора:

Єнікєєв Олександр Фанілович, кандидат технічних наук, доцент, доцент кафедри автоматизованих систем електричного транспорту Української державної академії залізничного транспорту.

УДК 004.054

Д.В. ФЕДАСЮК, В.С. ЯКОВИНА, П.В. СЕРДЮК, О.О. НИТРЕБИЧ

Національний університет «Львівська політехніка», м.Львів

МЕТОД ПОБУДОВИ СЦЕНАРІЇВ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНАЛІЗУ ЙОГО ЗМІННИХ

Анотація. Підвищення надійності програмного продукту є надзвичайно важливою та актуальною науковою задачею. Основним засобом вирішення даної проблеми є тестування програмного забезпечення (ПЗ), що вимагає значних затрат ресурсів, адже у більшості випадків проводиться вручну та повинно охоплювати усі сценарії використання і потоки управління та даних програмного продукту. У цій роботі описано метод автоматизованої побудови сценаріїв тестування на основі розробленої моделі поведінки ПЗ з урахуванням його змінних, що дозволяє скоротити фінансові, часові та людські ресурси та забезпечує рівномірне покриття коду.

Ключові слова: тестування на основі моделі, сценарій тестування, програмна відмова, модель поведінки ПЗ.

Аннотация. Повышение надежности программного продукта является чрезвычайно важной и актуальной научной задачей. Основным средством решения данной проблемы является тестирование программного обеспечения (ПО), которое требует значительных затрат ресурсов, ведь в большинстве случаев производится вручную и должно охватывать все сценарии использования и потоки управления и данных программного продукта. В этой работе реализован метод автоматизированного построения сценариев тестирования на основе разработанной модели поведения ПО с учетом его переменных, что позволяет сократить финансовые, временные и человеческие ресурсы и обеспечивает равномерное покрытие кода.

Ключевые слова: тестирования на основе модели, сценарий тестирования, программный отказ, модель поведения ПО.

Abstract. Improving the software reliability is very important and actual scientific task. The primary tool for solving this problem is software testing, which requires significant resources because it's mostly manual and should cover all usage scenarios and control and data flows of software. In this paper the new method for automated test cases construction is implemented. It is based on developed software usage model with consideration of its variables, therefore allowing to reduce financial, temporal and human resources and provide a uniform code coverage.

Key words: model-based testing, test case, software failure, software usage model.

Вступ

На сьогодні тестування є одним з найбільш важливих засобів перевірки надійності програмного забезпечення. Однак, зазвичай це ручне тестування, що є трудомістким процесом без ефективної автоматизації, яке потребує багато часу та фінансових ресурсів, але не здатне забезпечити виявлення усіх відмов ПЗ. Отже, важливим завданням у сфері програмної інженерії є автоматизація процесу тестування, а саме ефективна побудова сценаріїв тестування, що дозволить скоротити часові та економічні затратити.

Однією з найновіших технологій для автоматизованої побудови сценаріїв тестування програмного забезпечення є тестування на основі моделі програмного продукту [1-3] – тестування ПЗ, в якому тестові сценарії частково або повністю будуються з моделі, яка описує деякі аспекти (частіше функціональні) тестованої системи. Моделі програмної системи можуть відображати її поведінку або використовуватися для створення тестових стратегій чи середовища тестування.

Існує ряд моделей програмного забезпечення, що використовуються засобами автоматизованої побудови тестових сценаріїв. У загальному вони поділяються на чотири великі групи, які у свою чергу під час моделювання відповідно використовують [4]: скінченні автомати, граф станів, UML діаграми та Марковські ланцюги. Окрім того існують моделі на основі дерева прийняття рішень, таблиць рішень та графіків, але вони є новими і не достатньо дослідженими [4].

Граф станів, який ще іноді називають "потік тестування" використовується для моделювання програмних систем [5, 6] та побудови тестових сценаріїв та відображає реальне використання таких систем, і, таким чином, є більш точним, ніж використання скінченних автоматів. Перевагами моделей даного класу є можливість їхнього застосування для багатопотокових програм, але все-таки такі моделі важко спроектувати та реалізувати для складних програмних систем.

Широкого застосування набуло тестування програмної системи на основі UML діаграм [7-10]. Зазвичай автоматизоване генерування тестових сценаріїв відбувається на основі діаграми станів [8], прецедентів [9], послідовності [10]. Основними перевагами такого підходу є використання для складних систем, а можливість моделювання паралельного програмного забезпечення.

Моделі на основі скінченних автоматів є відомими динамічними моделями для автоматизованої побудови тестів [11, 12]. Формально скінченні автомати, що зображують програмне забезпечення, є сукупністю п'яти параметрів (I, S, T, F, L) , де I – множина вхідних параметрів ПЗ, S – множина всіх станів системи, T – це функція, яка визначає, чи відбувається перехід, коли системи з деякого стану, F – множина станів системи, які є кінцевими (з них система завершує свою роботу) та L – стани, з яких система починає свою роботу. Основними недоліками використання даних моделей є те, що їх легко спроектувати й підтримувати лише для систем із малим рівнем складності, неможливість використання даного класу моделей для паралельних ПЗ та систем з нескінченною кількістю станів.

Ще однією великою групою моделей для автоматизованої побудови тестових сценаріїв є моделі на основі дискретного Марковського ланцюга [13, 14]. За структурою Марковські ланцюги подібні до скінченних автоматів, але очевидно перевагою використання такого підходу є не лише генерація тестів,

але й аналіз відмов для оцінки та прогнозування надійності програмного продукту. Сценарії тестування отримуються через послідовні переходи між компонентами у відповідності до ймовірностей передачі контролю. Генерування тестових сценаріїв може бути автоматизоване за допомогою хорошого генератора випадкових чисел на будь-якій мові програмування високого рівня.

Таким чином для адекватного аналізу надійності ПЗ важливими задачами є удосконалення та розробка нових моделей на основі Марковського ланцюга, за допомогою яких можна здійснювати автоматизоване генерування тестів з урахуванням взаємозалежності переходів від однієї компоненти до іншої.

У даній роботі на основі розробленої авторами моделі поведінки ПЗ, що базується на Марковських ланцюгах [15], представлено метод автоматизованої побудови набору сценаріїв тестування, який враховує залежність виконання компонент та забезпечує рівномірне покриття коду, що дозволяє підвищити ефективність процесу тестування та зменшити його затрати. Окрім того даний метод можна застосувати для моделювання процесу тестування, що дасть змогу визначити вплив характеристик програмної системи на її надійність.

Актуальність

В останні роки складність комп'ютерних програм, що виконують важливі та відповідальні функції, швидко зростає, тому проблема підвищення надійності програмного забезпечення стає все більш актуальною науковою задачею. Тестування програмного продукту є основним засобом аналізу надійності програмних систем, яке ще до тепер часто є ручним, тому важливо розробляти нові методи автоматизованої побудови сценаріїв тестування, що значно підвищить ефективність процесу тестування. Окрім того, під час побудови тестових сценаріїв важливим є аналіз змінних програмного продукту, адже багато моделей оцінки надійності ПЗ використовують його метрики, які в свою базується на змінних ПЗ.

Мета

Метою даної роботи є розроблення методу побудови сценаріїв тестування ПЗ стратегіями «чорної» та «білої» скриньки на основі моделі поведінки програмного продукту, що базується на аналізі його змінних.

Модель поведінки програмного продукту на основі його змінних

Модель поведінки ПЗ, що є вхідним параметром для автоматизованої побудови тестів, зображується у вигляді орієнтованого графу $G = \{S, P\}$, де $S = \{S^0, S^1, \dots, S^n\}$ – множина методів програми S^i , P – множина переходів між відповідними методами [15].

Якщо позначити:

- V – множина усіх змінних програмного забезпечення;
- V^i – множина $\{v, E\}$, де v – змінна програмного продукту, а $E = \{E_{cor}, E_{incor}\}$ – множина відповідних правильних E_{cor} та неправильних E_{incor} класів еквівалентності такої змінної [16];
- $V_{used}^i = \{V^i, i = \overline{1, m}\}$ – множина змінних, які використовуються у методі S^i ; Вважаємо, що змінні та їхні значення є основним джерелом помило програмного забезпечення;
- $V_{change}^i = V_{change_user}^i \cup V_{change_program}^i$ – список змінних, що змінюються у методі S^i . Ця множина є об'єднанням множини змінних, що можуть змінюватись користувачем $V_{change_user}^i = \{V^i, i = \overline{1, l}\}$ (змінні, які можна протестувати за допомогою стратегії «чорної» скриньки), а також множини змінних, що можуть бути змінені лише внутрішньою логікою програми $V_{change_program}^i = \{V^i, i = \overline{l+1, k}\}$ (тестуються лише «білою» скринькою, k – кількість змінних, що змінюються у методі S^i);
- $V_{error}^i = \{V, E_{incor}\}$ – список змінних та відповідних неправильних класів еквівалентності, що можуть викликати помилку в програмній системі.

Тоді кожен вузол графу S^i є набором відповідних $\{V_{used}^i, V_{change}^i, V_{error}^i\}$.

Як відомо сьогодні існує дві найбільш важливі техніки, на основі яких здійснюють побудову тестових сценаріїв, а також проводять аналіз надійності та якості програмного продукту [17]:

1. Стратегія «чорної скриньки», яка розглядає системні характеристики програм, ігноруючи їхню внутрішню логічну структуру.

2. Стратегія «білої скриньки», яка передбачає детальне дослідження внутрішньої логіки і структури коду, для якого повинна бути відома повна інформація про вихідний код ПЗ.

Унаслідок врахування усіх змінних і відповідних класів еквівалентності, розроблена модель використання програмного продукту більш адекватно описує поведінку програмного забезпечення та дає можливість побудувати сценарії тестування ПЗ стратегіями як "чорної", так і "білої" скриньки.

Розглянемо побудову сценаріїв тестування для кожної з цих стратегій.

Побудова тестових сценаріїв методом «чорної» скриньки

Вважатимемо, що кожен сценарій тестування складається з декількох кроків $T^k = \{T_k^0, T_k^1, \dots, T_k^n\}$, де кожен крок сценарію $T_j^k = (S_j^k, C_j^k)$ визначається парою: методом S_j^k та множиною змінних C_j^k з відповідними класами еквівалентності, що точно змінюються в цьому методі. Змінні набувають своїх значень на кожному кроці тесту внаслідок введення даних користувачем, зчитуванням з бази даних та ін., через що можуть виникнути помилки під час виконання даного сценарію тестування.

Варто зауважити, що у відповідності до реального процесу тестування, набори тестів будуються ітераційно, крок за кроком виконуючи їх на програмному забезпеченні та враховуючи виявлені помилки під час побудови наступного набору тестів.

Введемо деякі позначення: $Er^j(S^i)$ – число тестів, під час яких відбулась відмова, і які включають метод S^i на j -тій ітерації. $M(S^i)$ – множина номерів усіх методів, що мають переходи із компоненти S^i ; $P(S^i)$ – множина всіх методів, у які можна передати контроль з методу S^i ; N – число всіх вершин графу поведінки програмної системи; TS – набір усіх сценаріїв тестування; $coverage(S^i)$ – кількість входжень в метод S^i .

Алгоритм автоматизованої побудови тестових сценаріїв складається із двох основних частин: початкова генерація тестів без інформації про відмови та решти K наборів тестів, що її включають.

Крок1. *Ініціалізація*. Вибір мінімального числа покриття всіх компонент ПЗ α та середньої довжини тестового сценарію β з досвіду тестувальника. Рекомендованим діапазоном початкових значень цих параметрів може бути $\beta \in \overline{5,7}$, $\alpha \in \overline{5,10}$. Установлення $TS = \{\}$, номер ітерації тестування $m=0$, номер тесту $k=0$.

Крок2. Побудова сценарію тестування T_k під час початкової генерації набору тестів.

1. $j=0$ – номер кроку такого тесту;
2. Випадковим чином обирається початковий крок з ймовірністю $p_k^j = 1/N$;
3. З ймовірністю $p_k^{j+1} = 1/|P(S^i)|$ вибирається наступний крок T_k^{j+1} у k -ому сценарії;
- $j++$;
4. Якщо $j \leq \beta$, тоді перехід на Крок2.3;
5. Додавання побудованого тесту до набору: $TS = TS \cup \{T_k\}$;

Крок3. Перевірка покриття усіх компонент. Якщо $coverage(S^i) < \alpha, i = \overline{1, N}$, тоді $k++$, перехід на Крок2.

Крок4. Виконання усіх побудованих тестів та обчислення $Er^m(S^i), i = \overline{1, N}$. Якщо $Er^m(S^i) = \emptyset, i = \overline{1, N}$, тоді Крок9, інакше $m++$;

Крок5. *Побудова m -го набору тестів ($m \geq 1$). $TS = \{\}$, $k=0$.*

Крок6. Побудова T_k сценарію тестування для m -го набору тестів.

1. $j=0$ – номер кроку такого тесту;

2. Випадковим чином обирається номер першого кроку, що буде відповідно першим кроком T_k^j тесту, з ймовірністю $p_k^j = \max_{i=1..N} \frac{Er^{m-1}(S^i)}{\sum_{l=1}^N Er^{m-1}(S^l)}$;
3. З ймовірністю $p_k^{j+1} = \max_{i \in M(S^j)} \frac{Er^{m-1}(S^i)}{\sum_{l \in M(S^j)} Er^{m-1}(S^l)}$ обирається наступний крок у T_k^{j+1} сценарії; $j++$;
4. Якщо $j \leq \beta$, тоді перехід на Крок6.3;
5. Додавання побудованого тесту до набору: $TS = TS \cup \{T_k\}$.

Крок7. Перевірка покриття усіх компонент. Якщо $\text{coverage}(S^i) < \alpha, i = \overline{1, N}$, тоді $k++$, перехід на Крок6.

Step8. Перехід на Крок4.

Step9. Кінець.

Побудовані послідовності методів виконуються при різних вхідних даних таким чином, щоб кожний вхідний параметр набував принаймні одного значення з кожного класу еквівалентності. Класи еквівалентності у методах чорної скриньки будуються не з аналізу коду, а відповідно до досвіду тестувальника. При цьому вхідні дані можуть і не набути усіх можливих значень і деякі відмови будуть пропущені, що є недоліком даної стратегії тестування.

Наприклад, для того, щоб повністю покрити код ПЗ, необхідно для кожного метода S^i здійснити $\|v_1\| \cdot \|v_2\| \cdot \dots \cdot \|v_n\|$ перевірок, де $\|v_j\|$ – кількість класів еквівалентності змінної $v_j \in S^i (j = \overline{1..n})$. Але через обмеження часових та людських ресурсів не можливо зробити таку велику кількість перевірок. Окрім того не всі перевірки відповідають реальній логіці програми (певних значень змінних не можливо досягнути під час використання програми), тому існує задача покриття коду тестами так, щоб повністю перебрати усі можливі набори змінних, допустимих внутрішньою логікою, або, якщо таких випадків дуже багато, то перебрати усі кортежі класів еквівалентності змінних довжиною до деякого степеня m , який визначається співвідношенням вимог до якості ПЗ та затратами на процес тестування. Стандартно ($m=1$) обмежуються перебором усіх класів еквівалентності для кожної змінної у методі $S^i - \|v_1\| + \|v_2\| + \dots + \|v_n\|$. Для $m=2$ необхідно перевірити всі пари наборів класів еквівалентності у методі $S^i - \sum_{k,l=1}^n \|v_k\| \cdot \|v_l\|$ і т.д. Тобто метод чорної скриньки може не повністю перебрати усі можливі випадки значень змінних, що є суттєвим недоліком даного підходу.

Отже, набори тестів, побудовані на кожній ітерації методом «чорної скриньки», можуть бути використані під час тестування, що у свою чергу дозволяє пришвидшити даний процес та зменшити фінансові та людські витрати. Окрім того, така генерація сценаріїв забезпечує рівномірне покриття коду, що покращує ефективність тестування, та відповідно підвищує якість ПЗ. Також завдяки рівномірному покриттю коду та використанню класів еквівалентності, ймовірність виявлення відмови зростає під час тестування програми, що в результаті збільшує надійність ПЗ.

Побудова тестових сценаріїв методом «білої» скриньки

Першою перевагою стратегії «білої» скриньки, яку неважко реалізувати, – це аналіз змінних, які використовуються та змінюються у методах. Відкинувши змінні, що не використовуються, можна зменшити перебір можливих V_{used}^i - підмножини змінних множини S^i , яку може використати метод S^i (змінні, що дійсно використовує цей метод). Також можна чітко визначити V_{change}^i - множину, яку може змінювати відповідний метод завдяки аналізу коду. Очевидно, що множна змінних методу S^i рівна $V_{used}^i \cup V_{change}^i$. Перетин цих змінних не обов'язково порожній.

Якщо розглянути тестовий сценарій, який складається лише з одного кроку $T=(S, C)$, то множина змінних, яка покривається даним сценарієм рівна $\alpha(T)=V_{used} \cap C$. Значення інших змінних можна покрити тільки тестовими сценаріями, які матимуть два або більше кроків. Множина змінних методу S^i , яка покривається сценарієм тестування, що містить два кроки $T = \{T^0, T^1\} = \{(S^0, C^0), (S^1, C^1)\}$ та включає два послідовні методи S^0 та S^1 має вигляд: $\alpha(T) = V_{used}^1 \cap (C^1 \cup C^0)$. У загальному випадку для тестового сценарію $T = \{T^0, T^1, \dots, T^N\} = \{(S^0, C^0), (S^1, C^1), \dots, (S^N, C^N)\}$ множина змінних, яку він покриває та впливає на процеси останнього методу S^N дорівнює: $\alpha(T) = V_{used}^N \cap (C^N \cup C^{N-1} \cup \dots \cup C^1 \cup C^0)$. Для того, щоб повністю покрити метод S^N різними значеннями змінних, необхідно, щоб $\alpha(T) = V_{used}^N$.

Зауважимо також, що сценарій тестування покриває множини змінних також і попередніх методів. Наприклад, для проміжного методу S^m ($m < N$) цього сценарію T множина покриття рівна:

$$\alpha(T, S^m) = V_{used}^m \cap (C^1 \cup C^2 \cup \dots \cup C^{m-1} \cup C^m). \quad (1)$$

Для останнього вузла: $\alpha(T, S^N) = \alpha(T)$.

Очевидно, що чим більше кроків у сценарії, тим більше можливостей для задання різних значень змінних. З іншої сторони, зі збільшення кількості кроків у сценаріях тестування зменшується ймовірність того, що тестовий сценарій буде пройдений до кінця, оскільки він може бути зупинений через виникнення відмови. Для зменшення трудоемності процесу тестування та підвищення його ефективності необхідно вибрати такий набір сценаріїв та кроків у ньому, які би повністю покривали значення усіх змінних методів та мали найменшу кількість кроків. Варто також зауважити, що у програмному забезпеченні буде багато змінних, які набувають своїх значень лише в одному методі (на одній формі, у конфігурації, і т.д), тому під час побудови тестових сценаріїв необхідно враховувати цей факт. Введемо позначення: $c(v_i)$ – кількість методів, у яких змінна може змінитись. Окрім того позначимо максимальну довжину тестового сценарію для даного ПЗ через β_{max} . Тоді міру покриття PVC (значення параметра покриття, яке вимагає, щоб змінні методу набували усіх своїх значень) визначатимемо як:

$$E(TS) = \sum_{i=1}^N \sum_{v_i \in V_{used}^i} \tilde{\delta}(v_i, \bigcup_{T \in TS} \alpha(T, S^i)), \quad (2)$$

$$\text{де } \tilde{\delta}(v, S) = \begin{cases} 0, & v \notin S; \\ 1, & v \in S. \end{cases}$$

Також ведемо поняття, яке буде визначати складність виконання тестів. Оскільки з точки зору тестувальника складність виконання переходу до наступного кроку є значно меншою у більшості випадків, ніж задання даних, то вимірювати складність будемо як кількість даних, що вносить користувач під час побудови тесту.

$$CM(TS) = \sum_{i=1}^{\|TS\|} \sum_{j=1}^{\|T_i\|} C_i^j \quad (3)$$

Таким чином можна формалізувати задачу побудови оптимального набору сценаріїв тестування ПЗ, що має такий вигляд:

$$CM(TS) \rightarrow \min$$

$$\text{з обмеженнями } E(TS) = E_{\max}(TS) = \sum_{i=1}^N \|V_{used}^i\|. \quad (4)$$

Алгоритм знаходження оптимальних покриттів є алгоритмом перебору з відсіканням гілок, які є надлишковими згідно міри покриття $E(TS)$ та складності виконання $CM(TS)$.

Крок1. *Ініціалізація*. Побудова початкового набору тестових сценаріїв, що складаються лише з одного кроку $TS = \{(S^0, V_{change}^0), (S^1, V_{change}^1), \dots, (S^N, V_{change}^N)\}$. На початковому кроці тестового

сценарію користувач буде змінювати усі можливі параметри, адже вони впливають на всі подальші кроки.

Крок 2. *Знаходження повного покриття PVC.* Множини тестових сценаріїв, побудованих на попередньому кроці, розширимо таким чином: для сценарію $T = ((S^0, C^0), (S^1, C^1), \dots, (S^K, C^K))$ з множини TS знаходимо усі методи S^{K+1} , які суміжні методу S^K і додаємо новий крок (S^{K+1}, C^{K+1}) до сценарію T , де $C^{K+1} = V_{change}^{K+1} \setminus \bigcup_{i=1}^K C^i$. Додаємо новий сценарій до множини TS , якщо при цьому зростає $E(TS)$. Якщо $E(TS) = E_{\max}(TS)$, то переходимо на Крок3, інакше повторюємо Крок2.

Крок3. *Мінімізація складності виконання сценарію тестування.* Для кожного сценарію з множини TS здійснюємо перевірку: якщо під час видалення його з множини TS величина $E(TS)$ не змінюється, то видаляємо його, інакше перевіряємо кожен крок (S^k, C^k) , видаляючи з множини C^k ті змінні, які не впливають на $E(TS)$. Це робиться для мінімізації $CM(TS)$.

Крок4. *Перебір класів еквівалентності змінних.* Побудовані тестові сценарії з множини TS виконуються при різних вхідних даних таким чином, щоб кожний вхідний параметр набував принаймні одного значення з кожного класу еквівалентності. Для підвищення ефективності тестування необхідно проаналізувати взаємозв’язані змінні та перебирати усі кортежі відповідних значень класів еквівалентності, що є значно складнішою задачею. Іншим підходом є більша дискретизація вузлів графу поведінки ПЗ до рівня програмних виразів та операторів управління, що дасть змогу глибоко проаналізувати покриття та взаємний вплив змінних.

Крок5. *Ітераційне виконання тестових сценаріїв.* Після побудови та виконання тестових сценаріїв знаходимо множину методів S_{err} , у яких виникли помилки та відповідний набір класів еквівалентності змінних V_{err} . Якщо $S_{err} = \emptyset$, то закінчуємо алгоритм, інакше, починаючи з Кроку1, будуємо нову множину тестових сценаріїв TS з новим покриттям:

$$E(TS) = \sum_{S_i \in S_{err}} \sum_{v_i \in V_{used}^i \cap V_{err}^i} \tilde{\delta}(v_i, \bigcup_{T \in TS} \alpha(T, S^i)).$$

Отже, наведений алгоритм дозволяє будувати сценарії тестування ПЗ з максимальним покриттям коду при мінімальній складності виконання сценарію тестування, що підвищує ефективність процесу тестування та виправлення дефектів програмного продукту, що своєю чергою, дає змогу підвищити його показники надійності.

Експериментальні результати

На основі розробленого методу побудови сценаріїв тестування ПЗ було реалізовано програмний продукт, вхідним параметром якого є модель поведінки ПЗ. Було проведено чисельні експерименти, в яких задавались різні характеристики моделі поведінки ПЗ для умовного програмного продукту (кількість методів; кількість змінних з відповідними класами еквівалентності, що використовуються та змінюються методами; ймовірності переходів між методами та інформація про помилки). Для ілюстрації адекватності методу побудови сценаріїв тестування для такої моделі використання було побудовано набори тестових сценаріїв для двох стратегій тестування, які потім використовувались для дослідження залежності кількості відмов ПЗ від різних його характеристик. Результати моделювання поведінки кількості відмов в залежності від кількості тестових сценаріїв з урахуванням різних стратегій наведені на рис.1.

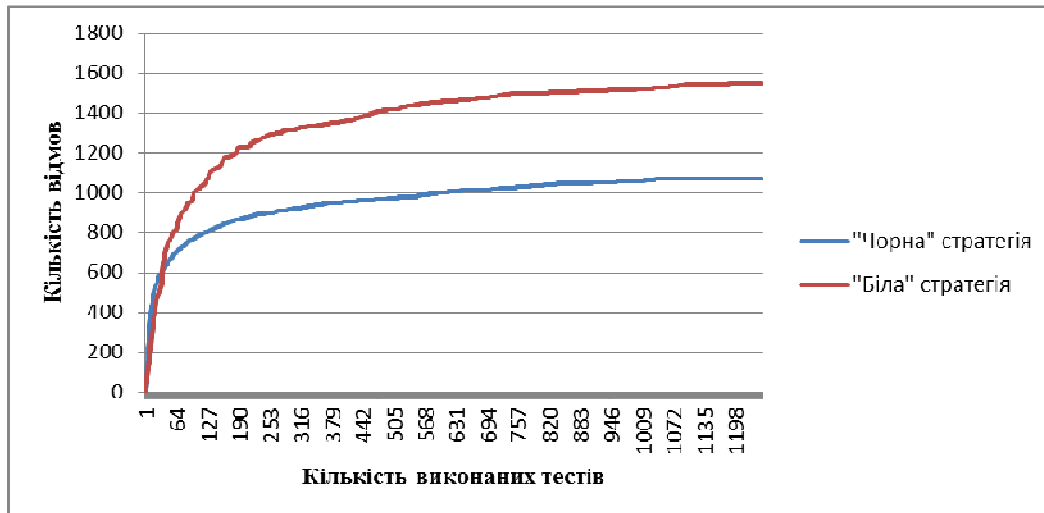


Рисунок 1 – Графік залежності кількості пройдених тестів та кількості зафіксованих відмов двома стратегіями тестування

Як видно з рис.1 під час тестування стратегією «білої» скриньки було знайдено більше відмов ПЗ, ніж під час тестування стратегією «чорної скриньки». Цей факт, а також якісний вигляд залежності відповідає відомим даним тестування реальних програмних продуктів [18, 19], адже тестування стратегією «білої» скриньки враховує особливості архітектури програмного продукту та особливості його помилок.

Графік залежності кількості відмов, що були зафіксовані під час процесу тестування, від кількості методів програмного забезпечення, (рис. 2) ілюструє лінійний зв'язок між ними для усіх стратегій тестування. Такий самий вигляд залежності кількості відмов від об'єму ПЗ був описаний в [19], що підтверджує адекватність запропонованого методу.

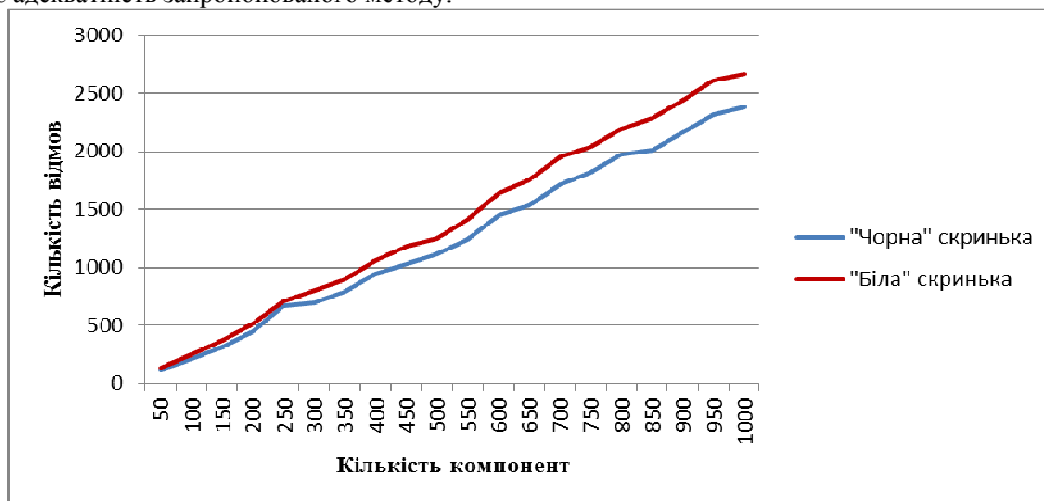


Рисунок 2 – Графік залежності кількості відмов, знайдених двома стратегіями тестування та кількості методів ПЗ

Окрім того проведено дослідження кількості виявлених відмов від середньої кількості змінних, що використовуються у методах програмного продукту (рис. 3). Кількість змінних, що використовуються методом ПЗ, вимірюється відсотком від загальної кількості змінних. Як видно з графіку зі збільшенням кількості таких змінних кількість відмов під час тестування стратегією «білої» скриньки дещо зростає, що обумовлене можливістю аналізувати усі змінні та їхні значення під час побудови тестів, а, отже, виявляти більше відмов. Під час тестування стратегією «чорної» скриньки із ростом кількості змінних, що використовуються методом ПЗ, кількість відмов зменшується. Це пояснюється тим, що стратегія «чорної» скриньки не дає можливості проаналізувати внутрішні змінні програмного продукту, які впливають на перебіг виконання програми, а лише ті, які може змінювати користувач (кількість таких змінних залишається незмінною), тому ефективність цієї стратегії погіршується з ростом середньої кількості змінних, що використовуються методом.

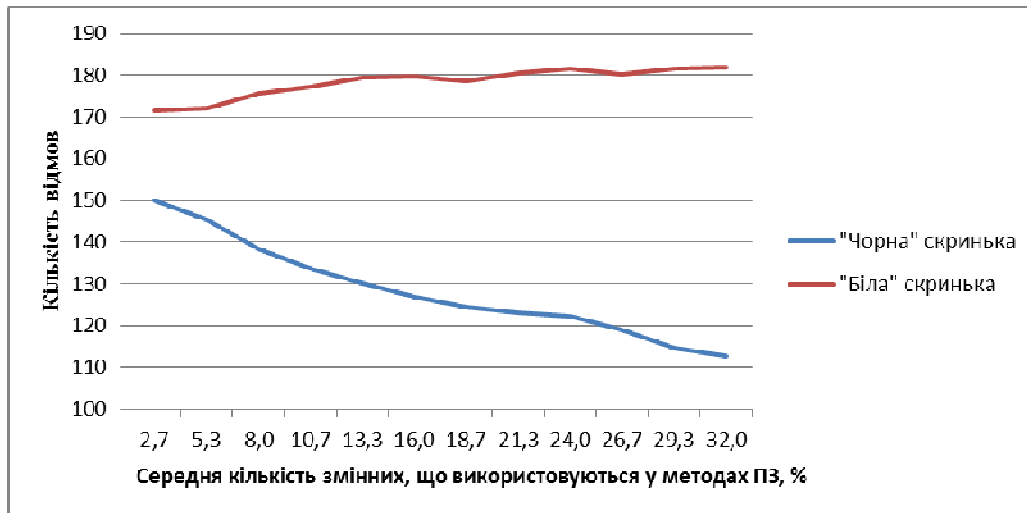


Рисунок 3 – Графік залежності кількості відмов від середньої кількості змінних, що використовуються у методах ПЗ, для двох стратегій тестування.

Висновки

У цій роботі розроблено метод автоматизованої побудови сценаріїв тестування програмного забезпечення, входним параметром якого є модель поведінки програмного забезпечення на основі його змінних [15], який може бути використаний як для стратегії «чорної», так і «білої» скриньки. Даний метод забезпечує рівномірне покриття коду та підвищує ефективність процесу тестування, зменшуючи часові, фінансові та людські ресурси. Метод побудови тестів реалізований у вигляді програмного продукту, який крім автоматизованої побудови тестових сценаріїв дозволяє змоделювати поведінку ПЗ та моделювати виконання побудованих тестових сценаріїв. З використанням реалізованого ПЗ проведено дослідження впливу характеристик процесу тестування (кількості виконаних тестів) і програмного продукту (кількість компонент ПЗ та середня кількість змінних, що використовуються у методах ПЗ) на кількість виявлених на етапі тестування відмов. Отримані графіки залежностей кількості відмов якісно співпадають з відомими літературними даними, що підтверджує адекватність розробленого методу побудови тестових сценаріїв.

Список літератури

1. Broy M. Model Based Testing of Reactive Systems. / M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, A. Pretschner. – LNCS 3472, Springer. – 2005. – 659 p.
2. Blackburn M. Why Model-Based Test Automation is Different and What You Should Know to Get Started. / M. Blackburn, R. Busser, A. Nauman // In International Conference on Practical Software Quality. – 2004. – p. 87-90.
3. Legeard B. Controlling Test Case Explosion in Test Generation from B Formal Models. / B. Legeard, F. Peureux, M. Utting // The Journal of Software Testing, Verification and Reliability. – 2004. – no. 14(2). – pp. 81–103.
4. Ebrahim Shamsoddin-Motlagh. A Review of Automatic Test Cases Generation. / Ebrahim Shamsoddin-Motlagh // International Journal of Computer Applications. – 2012. – no. 57(13). – pp. 25-29.
5. Belli F. A Holistic Approach to Testing of Interactive Systems using Statecharts. / F. Belli, C.J. Budnik, A. Hollman // Proceedings of 2nd South-East European Workshop on Formal Methods (SEEFM 05), South-Eastern European Research Center SEERC. – 2005. – pp. 1–15.
6. Hyoung Seok Hong. Automatic Test Generation from Statecharts Using Model Checking / Insup Lee, Oleg Sokolsky, Sung Deok Cha // In Proceedings of FATES'01, Workshop on Formal Approaches to Testing of Software. – 2001.
7. Hu Y.T. Automatic Black-Box Method-Level Test Case Generation Based on Constraint Logic Programming. / Y.T. Hu, N.W. Lin // Computer Symposium (ICS). – 2010. – pp. 977-982.
8. Samuel P. (2008). Automatic test case generation using unified modeling language (UML) state diagrams. / P. Samuel, R. Mall, A. Bothra // The Institution of Engineering and Technology, IET Softw. – 2008. no. 2. – pp. 79–93.
9. Sarma M. Automatic Test Case Generation from UML Models. / M. Sarma, R. Mall // 10th International Conference on Information Technology. – 2007. – pp. 196-201.
10. Sarma M. (2007). Automatic Test Case Generation from UML Sequence Diagrams. / M. Sarma, D. Kundu, R. Mall // 15th International Conference on Advanced Computing and Communications. – 2007. – pp. 60-65.

11. Santiago V. An Environment for Automated Test Case Generation from Statechart-based and Finite State Machine-based Behavioral Models. / V. Santiago, N. Vijaykumar, D. Guimaraes //IEEE International Conference on Software Testing Verification and Validation Workshop (ICSTW'08). – 2008. – pp.63-72.
12. Susumu Fujiwara. Test Selection Based on Finite State Models. / Susumu Fujiwara, Gregor Bochmann, Ferhat Khendek, Mokhtar Amalou, Abderrazak Ghedamsi // IEEE Transactions on Software Engineering. – 1991. – no. 6(17). – pp. 591–603.
13. Whittaker J. A Markov Chain Model for Statistical Software Testing. / J. A. Whittaker, M. G. Thomason // Software Engineering, IEEE Transactions. – 1994. – pp. 812-824.
14. Winfried Dulz. Matelo - Statistical Usage Testing by Annotated Sequence Diagrams, Markov Chains and TTCN-3 / Winfried Dulz, Fenhua Zhen // Third International Conference On Quality Software. – 2003.
15. Fedasyuk D.V. Variables state-based software usage model. / D.V. Fedasyuk, V.S. Yakovyna, P.V. Serdyuk, O.O. Nytrebych // ECONTECHMOD. – 2014 (in publishing).
16. Степанченко И.В. Методы тестирования программного обеспечения: Учебное пособие. / И.В. Степанченко - Волгоград: ВолгГТУ. – 2006. - 74 с.
17. Mohd Ehmer Khan. Different Forms of Software Testing Techniques for Finding Errors. / Mohd Ehmer Khan // International Journal of Computer Science Issues. – 2010. V. 7. – pp. 11-16.
18. Shaik M. Software reliability Growth model with bass diffusion test- effort function and analysis of software release policy. / M. Shaik, R. Shaheda // International Journal of Computer Theory and Engineering. – 2011. – V.3. – pp. 671-680.
19. El Emam K. A methodology for validating software product metrics. / El K. Emam // Tech. rep. NCR/ERC-1076, National Research Council of Canada, Ottawa, Ontario, Canada. – 2000.

Відомості про авторів

Федасюк Дмитро Васильович – професор, доктор технічних наук, завідувач кафедри програмного забезпечення національного університету «Львівська політехніка».

Яковина Віталій Степанович – доцент, кандидат фізико-математичних наук, доцент кафедри програмного забезпечення національного університету «Львівська політехніка».

Сердюк Павло Віталійович – доцент, кандидат технічних наук, доцент кафедри програмного забезпечення національного університету «Львівська політехніка».

Нитребич Оксана Олександрівна – аспірант кафедри програмного забезпечення національного університету «Львівська політехніка».

УДК 519.876.5

М.П. ДИВАК, А.В. ПУКАС, І.С. ОЛІЙНИК

Тернопільський національний економічний університет

ОСОБЛИВОСТІ КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ МЕТОДУ ЛОКАЛІЗАЦІЇ ПАРАМЕТРІВ ІНТЕРВАЛЬНИХ МОДЕЛЕЙ ІЗ ВИДІЛЕННЯМ «НАСИЧЕНОГО БЛОКУ»

Анотація. Параметрична ідентифікація є одним із етапів побудови моделей статичних систем. Одним із найбільш ефективних методів параметричної ідентифікації на основі даних з інтервальною невизначеністю є метод локалізації (апроксимації) області параметрів інтервальних моделей із виділенням «насиченого блоку» у базовій інтервальній системі лінійних алгебраїчних рівнянь (ИСЛАР). Метод має ряд переваг у порівнянні з існуючими, зокрема він є найкращим, за критерієм простоти оперування отриманим коридором інтервальних моделей для задач прогнозування та за критерієм мінімізації обчислювальної складності знаходження множинних оцінок параметрів моделей. Проте не дослідженими залишаються умови застосування методу. У статті показано ефективність комп'ютерної реалізації методу та досліджено умови, за яких відомі алгоритми його реалізації не дають можливість отримати розв'язок.

Ключові слова: інтервальна модель, метод локалізації параметрів інтервальних моделей, «насичений блок» ИСЛАР.

Аннотация. Параметрическая идентификация является одним из этапов построения моделей статических систем. Одним из наиболее эффективных методов параметрической идентификации на основе данных с интервальной неопределенностью есть метод локализации (аппроксимации) области параметров интервальных моделей с выделением «насыщенного блока» в базовой интервальной системе линейных алгебраических уравнений (ИСЛАУ). Метод имеет ряд преимуществ в сравнении с существующими, в частности он является наилучшим, по критерию простоты управления полученным коридором интервальных моделей для задач прогнозирования и по критерию минимизации вычислительной сложности нахождения множественных оценок параметров моделей. Однако не исследованными остаются условия применения метода. В статье показана эффективность компьютерной реализации метода и исследовано условия, за которых известные алгоритмы его реализации не дают возможность получить решение.

Ключевые слова: интервальная модель, метод локализации параметров интервальных моделей, «насыщенный блок» ИСЛАУ.

Abstract. Parameter identification is one from the stages of construction the models of the static systems. One from the most effective methods of parameter identification based on data with an interval uncertainty there is a method of localization (approximation) the area of interval model parameters with selection the «saturated block» in the basic interval system of linear algebraic equations (ISLAE). This method has some advantages in comparison with existing methods. In particular it is the best by the criterion of simplicity of operation by corridor of interval models for the prediction tasks and by the criterion of minimization the computing complexity of finding the multiple estimations of models parameters. However the conditions of method's application are not research. There is the efficiency of computer realization the method and the conditions at which the known algorithms of its realization do not enable to get the solution are research in the article.

Keywords: interval model, method of localization the interval model parameters, «saturated block» of ISLAE.

Вступ

В теорії математичного моделювання особливе місце займає клас моделей у вигляді «чорної скриньки» з інтервальною невизначеністю на «вході» чи на «виході». Під інтервальною невизначеністю тут розуміємо встановлений замість точного числового значення змінної деякий числовий інтервал можливих значень цієї змінної. Така невизначеність може бути пов'язана із похибками вимірювань, похибками заокруглень в процесі імітаційного експерименту чи суб'єктивно встановленими інтервалами можливих значень змінних в процесі опитування експертів [1]. Чим ширший зазначений числовий інтервал, тим більша невизначеність у встановленні значення деякої змінної. Таким чином результати експерименту, за якими будують математичну модель, представляють у вигляді числових інтервалів – інтервальних даних [2], а математичну модель, яку побудовано за результатами аналізу інтервальних даних називають інтервальною моделлю. Особливістю побудованих інтервальних моделей є можливість відображення гарантованого коридору, в межах якого прогнозують значення «виходу». При побудові інтервальних моделей дослідникам доводиться розв'язувати задачі структурної та параметричної ідентифікації. Вказана проблематика описана у працях Бакана Г.М., Вошиніна О.П., Грановського В.А., Дивака М.П., Кунцевича В.М., Куржанського А.Б., Личака М.М., Шарого С.П.

При структурній ідентифікації знаходять структуру математичної моделі, а при параметричній – налаштовують її параметри, тобто шукають їх оцінки. У випадку застосування аналізу інтервальних даних отримують не точкові, а множинні оцінки параметрів моделі, які відповідно дають можливість побудувати гарантований коридор інтервальних моделей [3]. Властивості інтервальних моделей – як математичного об'єкта в достатній мірі досліджено в монографії [4]. Зокрема, в зазначеній праці встановлено, що найкращим, за критерієм простоти оперування коридором інтервальних моделей для задач прогнозування та за критерієм мінімізації обчислювальної складності знаходження множинних оцінок параметрів моделей, є метод локалізації (знаходження гарантованої області) параметрів інтервальних моделей із виділенням «насиченого блоку» [4].

Зазначений метод дає можливість отримати множинні оцінки параметрів інтервальних моделей у вигляді області, яка геометрично є паралелограмом (для двовимірного випадку – паралелограм) із відомим центром симетрії та відомими аналітичними виразами для вписаного чи описаного багатовимірного

еліпсоїда. Саме ці властивості множинного розв'язку задачі параметричної ідентифікації на основі інтервальних даних забезпечують простоту оперування коридором інтервальних моделей для задач прогнозування. Іншою перевагою методу локалізації параметрів інтервальних моделей із виділенням «насиченого блоку» є наявність простої обчислювальної схеми з високим ступенем паралелізму, яку описано за допомогою обчислювального графу [4].

Враховуючи зазначені переваги цього методу локалізації, розроблено його численні комп'ютерні реалізації для розв'язування задач моделювання та прогнозування в енергетиці, екології, медицині та технологічних галузях. Проте подальшого детального аналізу вимагають випадки, коли результати комп'ютерної реалізації методу є не прогнозованими, зокрема пов'язані із відсутністю розв'язку інтервальної системи лінійних алгебричних рівнянь (ІСЛАР), яка є базовою математичною задачею методу.

Метою даної праці є дослідження та встановлення умов виникнення вищезазначеної ситуації в процесі комп'ютерної реалізації методу.

Особливості побудови інтервальних моделей

Залежність між «виходом» та «входами» статичної системи описуємо у вигляді алгебричного лінійного за невідомими параметрами рівняння [5]:

$$y_0 = \beta_1 \cdot \varphi_1(\bar{x}) + \dots + \beta_m \cdot \varphi_m(\bar{x}), \quad (1)$$

де y_0 – істинне невідоме значення «виходу» системи; $\bar{x} \in R^m$ – вектор вхідних змінних; $\vec{\beta} = (\beta_1, \dots, \beta_m)^T$ – вектор невідомих параметрів, $\vec{\varphi}^T(\bar{x}) = (\varphi_1(\bar{x}), \dots, \varphi_m(\bar{x}))^T$ – вектор відомих базисних функцій.

Результати експерименту представляємо матрицею значень вхідних змінних та вектором інтервальних значень вихідної змінної y .

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & & \vdots \\ x_{i1} & \dots & x_{in} \\ \vdots & & \vdots \\ x_{N1} & \dots & x_{Nn} \end{pmatrix}; [\vec{Y}] = \begin{pmatrix} [y_1^-; y_1^+] \\ \vdots \\ [y_i^-; y_i^+] \\ \vdots \\ [y_N^-; y_N^+] \end{pmatrix}; \quad (2)$$

Умовами для знаходження вектора $\vec{\beta}$ невідомих параметрів є:

$$y_i^- \leq y_{oi} \leq y_i^+, \quad (3)$$

звідси отримуємо інтервальну систему лінійних алгебричних рівнянь (ІСЛАР) [5]:

$$\begin{cases} y_1^- \leq b_1 \varphi_1(\bar{x}_1) + \dots + b_m \varphi_m(\bar{x}_1) \leq y_1^+ \\ \vdots \\ y_i^- \leq b_1 \varphi_1(\bar{x}_i) + \dots + b_m \varphi_m(\bar{x}_i) \leq y_i^+ \\ \vdots \\ y_N^- \leq b_1 \varphi_1(\bar{x}_N) + \dots + b_m \varphi_m(\bar{x}_N) \leq y_N^+ \end{cases} \quad (4)$$

Множину розв'язків отриманої ІСЛАР (4) отримаємо у такому вигляді Ω :

$$\Omega = \left\{ \vec{b} \in R^m \mid \vec{Y}^- \leq F \cdot \vec{b} \leq \vec{Y}^+ \right\}, \quad (5)$$

де $F = \{\phi_j(\bar{x}_i), i = 1, \dots, N, j = 1, \dots, m\}$ - відома матриця значень базисних функцій, $\bar{Y}^- = \{y_i^-, i = 1, \dots, N\}$ та $\bar{Y}^+ = \{y_i^+, i = 1, \dots, N\}$ - вектори, складені з верхніх та нижніх меж інтервалів $[y_i^-, y_i^+]$. Відомо, що геометрично область розв’язків ІСЛАР (4) є многогранником [3].

Довільний вектор параметрів $\bar{b} \in \Omega$ ІСЛАР (4) породжує модель $\hat{y}(\bar{x}) = \bar{\phi}^T(\bar{x}) \cdot \bar{b}$, яку надалі називатимемо інтервальною моделлю [6] (статичної системи). Разом з тим множина розв’язків Ω породжує множину рівнозначних (з точки зору наявної інтервальної невизначеності) інтервальних моделей, кожна з яких задовольняє умовам задачі. При цьому всі інтервальні моделі належать множині (коридору):

$$[\hat{y}(x)] = [\hat{y}^-(x); \hat{y}^+(x)], \tag{6}$$

де $\hat{y}^-(\bar{x}) = \min_{\bar{b} \in \Omega} (\bar{\phi}^T(\bar{x}) \cdot \bar{b})$ та $\hat{y}^+(\bar{x}) = \max_{\bar{b} \in \Omega} (\bar{\phi}^T(\bar{x}) \cdot \bar{b})$ - нижня та верхня межі функціонального коридору.

Якщо кількість рівнянь в ІСЛАР дорівнює кількості невідомих параметрів, то область (5) – паралелопоп. Виділену із ІСЛАР (4) ІСЛАР із кількістю інтервальних рівнянь, яка співпадає із кількістю невідомих параметрів називають «насиченим блоком» [4]. Надалі матрицю значень базисних функцій для насиченого блоку будемо позначати за F_m , а область розв’язків такої системи за Ω_m .

Область розв’язків «насиченого блоку» має ряд важливих властивостей:

1. Для насиченого блоку, тобто у випадку співпадіння кількості спостережень N в експерименті з кількістю невідомих параметрів m , матриця F_m ІСЛАР (4) – квадратна ($m \times m$) і якщо вона не вироджена, то існує аналітична залежність для обчислення кожної вершини паралелопопа [4]:

$$\bar{b}_s = F_m^{-1} \cdot \bar{Y}_s \tag{7}$$

де \bar{Y}_s - вектор, складений з межових значень інтервалів $[y_i^-, y_i^+]$, наприклад, він може мати такий вигляд $\bar{Y}_s = (y_1^+, y_2^-, y_3^+, \dots, y_N^-)$. Слід зауважити, що таких вершин є $R = 2^m$ [4].

2. Центр симетрії паралелопопа $\bar{\bar{b}}$ визначаються за формулою [4]:

$$\bar{\bar{b}} = \frac{1}{2^m} \sum_{s=1}^{2^m} \bar{b}_s = F_m^{-1} \cdot \bar{\bar{Y}} \tag{8}$$

де $\bar{\bar{Y}}$ - вектор є середнім арифметичним усіх векторів складених з межових значень, а його компоненти – середні інтервальні значення $\bar{y}_i = (y_i^+ + y_i^-) / 2, i = 1, \dots, m$, тобто

$$\bar{\bar{Y}} = \frac{1}{2^m} \cdot \left(\sum_{s=1}^{2^m} \bar{Y}_s \right) = (\bar{y}_1, \dots, \bar{y}_m)^T \tag{9}$$

3. Об’єм отриманого паралелопопа обчислюють за формулою:

$$V_{\Omega_m} = 4^m \cdot \left(\prod_{i=1}^m (y_i^+ - y_i^-)^2 \right) \cdot \det(F_m \cdot F_m^T)^{-1} \tag{10}$$

Представлення області параметрів у вигляді паралелопопа, тобто «насиченого блоку», є зручним, оскільки суттєво спрощує використання множинних оцінок параметрів. Зокрема, область у вигляді паралелопопа, в свою чергу, можна представити у вигляді описаного або вписаного багатовимірних

еліпсоїдів, спираючись на твердження, наведені у праці [4]. Проте, якщо в ІСЛАР (3) $N > m$, то побудова вищезгаданого представлення області параметрів суттєвим чином ускладнюється. Разом з тим у праці [4] наведено метод локалізації параметрів інтервальних моделей із виділенням «насиченого блоку», що дає можливість апроксимувати область розв'язків ІСЛАР з $N > m$ у вигляді паралелотопа.

Таким чином, основною математичною задачею при побудові інтервальних моделей є задача оцінювання множини розв'язків ІСЛАР, яку доцільно розв'язувати із застосуванням методу локалізації параметрів інтервальних моделей із виділенням «насиченого блоку». Проте у всіх реалізаціях методу припускають, що ІСЛАР є сумісною. Таке припущення не завжди відповідає практичним застосуванням. Встановлення умов, за яких метод локалізації не забезпечує розв'язку задачі є метою подальших досліджень в даній праці.

Метод локалізації параметрів інтервальних моделей із виділенням «насиченого блоку»

Розглянемо суть методу локалізації. На першому етапі реалізації методу вибираємо із ІСЛАР (4) m рівнянь, які утворюють також сумісну систему. Розв'язком цієї системи є область Ω_m , яка геометрично має вигляд паралелотопа з вершинами (7).

Способи формування (вибору) цих рівнянь описано в праці [6]. Проте слід зауважити, що на практиці не існує строго формального підходу до вибору «насиченого блоку», який би апіорі гарантував задані властивості локалізаційної області, зокрема її мінімальні розміри. У подальших дослідженнях припустимо, що насичений блок є вибрано у той чи інший спосіб. Інтервальні рівняння, що входять в сформовану систему будемо називати базовими.

Тепер поставимо умову, щоб об'єм локалізаційної області параметрів моделей був мінімальним:

$$V_{\Omega_m} \longrightarrow \min, \quad (11)$$

з урахуванням решти $N-m$ - інтервальних рівнянь ІСЛАР (4) та за умови включення

$$\Omega \subseteq \Omega_m. \quad (12)$$

Спираючись на формулу (10), отримуємо задачу локалізації у вигляді математичної задачі оптимізації [7]

$$4^m \cdot \left(\prod_{i=1}^m (y_i^+ - y_i^-)^2 \right) \cdot \det(F_m \cdot F_m^T)^{-1} \xrightarrow{F_m} \min \quad (13)$$

за умов (12).

При відомій фіксованій матриці F_m задача локалізації (13), (12) стає еквівалентною таким задачам:

$$y_i^+ \longrightarrow \min, \quad y_i^- \longrightarrow \max, \quad \forall i = 1, \dots, m, \quad \Omega \subseteq \Omega_m \quad (14)$$

Для розв'язування задачі локалізації (14), у праці [4] запропоновано ітераційну обчислювальну схему. Позначимо за $y_i^-(k+1)$ нижнє та за $y_i^+(k+1)$ - верхнє інтервальні значення правих частин кожного з базових m рівнянь на $k+1$ ітерації і обчислюватимемо їх за такими формулами:

$$y_i^-(k+1) = y_i^-(k) + \delta_i^-(k+1), \quad y_i^+(k+1) = y_i^+(k) - \delta_i^+(k+1), \quad i = 1, \dots, m. \quad (15)$$

Тоді, спираючись на вираз (14), значення $\delta_i^-(k+1) \geq 0$ та $\delta_i^+(k+1) \geq 0$ у формулах (15) необхідно максимізувати, виходячи із умови включень

$$\Omega \subseteq \{ \Omega_m(k) \cap \check{\Omega}(k+1) \} \subseteq \Omega_m(k+1), \quad (16)$$

де $\Omega_m(k+1)$ - m - вимірний паралелотоп, отриманий на $k+1$ - й ітерації; $\check{\Omega}(k+1)$ - «гіперсмуга», яка визначається $k+1$ рівнянням ($k=1, \dots, N-m$) із тих, що залишились в ІСЛАР (4) після вибору m базових рівнянь.

У праці [4] отримано аналітичні співвідношення для розрахунку значень $\delta_i^-(k+1)$ та $\delta_i^+(k+1)$ для наступної ітерації. Розглянемо коротко суть міркувань, на основі яких отримані зазначені співвідношення.

Позначимо за $\vec{b}_s(k)$ ($s = 1, \dots, 2^m$) вершину паралелограма $\Omega_m(k)$ і перепишемо формулу (7) для обчислення координат вершин на k -тій ітерації

$$\vec{b}_s(k) = F_m^{-1} \cdot \vec{Y}_s(k), \quad (17)$$

де $\vec{Y}_s(k)$, - вектор, складений із комбінацій нижніх $y_i^-(k)$ та верхніх $y_i^+(k)$ інтервальних значень правих частин кожного з базових m рівнянь, які обчислюємо за рекурентними формулами (15).

У праці [4] показано, якщо вершини $\vec{b}_s(k)$ належать одній грані паралелограма, то відповідні вектори $\vec{Y}_s(k)$ в формулі (17) мають хоча б по одній спільній компоненті $y_i^-(k)$ чи $y_i^+(k)$. Це означає, що обчислення меж $y_i^-(k)$ та $y_i^+(k)$ в базових рівняннях на k -тому кроці за рекурентними схемами (15) та за умов максимізації $\delta_i^-(k+1) \geq 0$ і $\delta_i^+(k+1) \geq 0$, графічно відображається переміщенням відповідних граней паралелограма $\Omega_m(k)$ в сторону зменшення його розмірів, але при цьому має виконуватися умова (16).

З метою визначення відстані, на яку потрібно переміщувати грань, щоб забезпечити умову включення (16), для кожної вершини $\vec{b}_s(k)$ уведено скалярні функції, які характеризують відстань між вершиною і відповідною межею “гіперсмуги” $\tilde{\Omega}(k+1)$

$$L_s(k) = y_{k+1}^- - \vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{b}_s(k), \quad (18)$$

$$L'_s(k) = \vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{b}'_s(k) - y_{k+1}^+ = -L_s(k) - \Delta_{k+1}, \quad (19)$$

де \vec{x}_{k+1} - вектор вхідних змінних у $k+1$ спостереженні, який визначає $k+1$ рівняння в ІСЛАР (4); y_{k+1}^- , y_{k+1}^+ - нижнє та верхнє інтервальні значення вихідної змінної для $k+1$ інтервального рівняння; $\Delta_{k+1} = y_{k+1}^+ - y_{k+1}^-$.

В подальшому обчислювальна схема методу ґрунтується на аналізі значень функцій $L_s(k)$, $L'_s(k)$, в особливості їх знаків.

Такий аналіз, проведений у праці [4], дозволяє виявити розміщення вершин гіперпаралелепіеда відносно “гіперсмуги”, утвореної $k+1$ інтервальним рівнянням. Проте проведені дослідження комп’ютерної реалізації методу показали, що неврахована можлива несумісність ІСЛАР (4). Тому розглянемо таку можливість при подальшому удосконаленні обчислювальної схеми.

Як показано на рис. 1а) для випадку $m = 2$, вершини \vec{b}_1 , \vec{b}_2 грані з номером $i = 1$ розміщені з одного боку “смуги” і відповідно $L_s(k) > 0 \quad \forall s = 1, 2$; одна вершина \vec{b}_3 грані з номером $i' = 1$ належить “смугі”, а інша \vec{b}_4 знаходиться на її межі, тому $L_3(k) < 0$, а $L'_4(k) = 0$.

На рис. 1б) показано, що вершини \vec{b}_2 , \vec{b}_3 грані з номером $i' = 2$ розміщені по обидва боки “смуги” і, відповідно, $L_2(k) < 0$, $L_3(k) > 0$; обидві вершини \vec{b}_3 , \vec{b}_4 грані з номером $i' = 1$ знаходяться вище “смуги” (з одного боку “смуги”), тому $L'_s(k) > 0 \quad \forall s = 3, 4$.

Разом з тим, наведений рисунок 1 не ілюструє випадок коли ІСЛАР (4) є несумісною, що не забезпечує отримання вимог щодо наявності розв’язку задачі локалізації. На рис.2 наведено випадки, коли зазначена система є несумісною.

Як показано на рис. 2а) для випадку $m=2$, усі вершини $(\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4)$ паралелограма розміщені з одного боку "смуги" і відповідно $L_s(k) > 0 \quad \forall s = 1, \dots, 4$. Проте ІСЛАР, графічний розв'язок якої представлено на рис.2 у вигляді паралелограма та «смуги», яка його не перетинає, є несумісна.

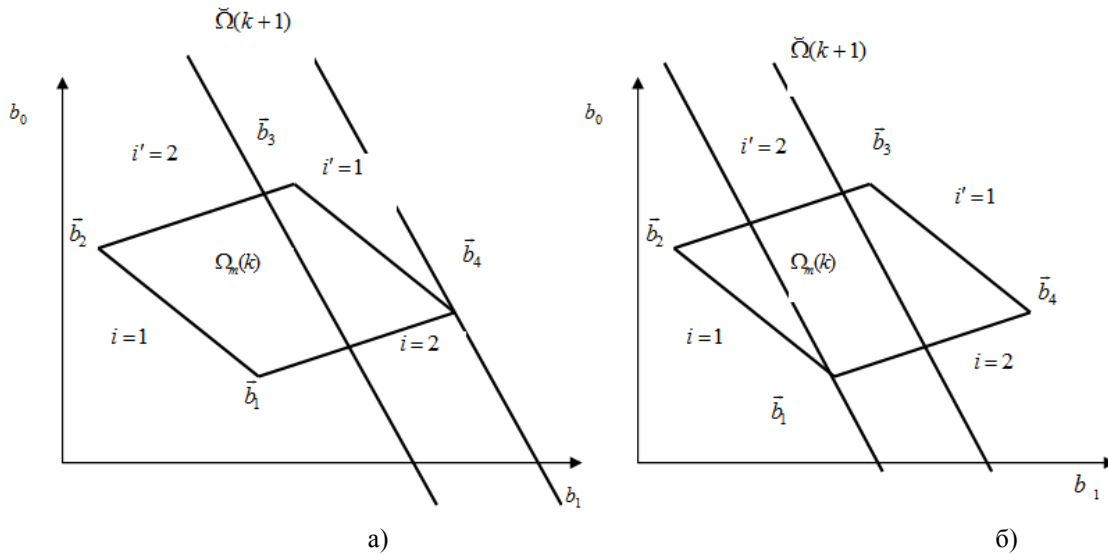


Рисунок 1 – Розміщення вершин множини локалізації $\Omega_m(k)$ стосовно "смуги" $\tilde{\Omega}(k+1)$

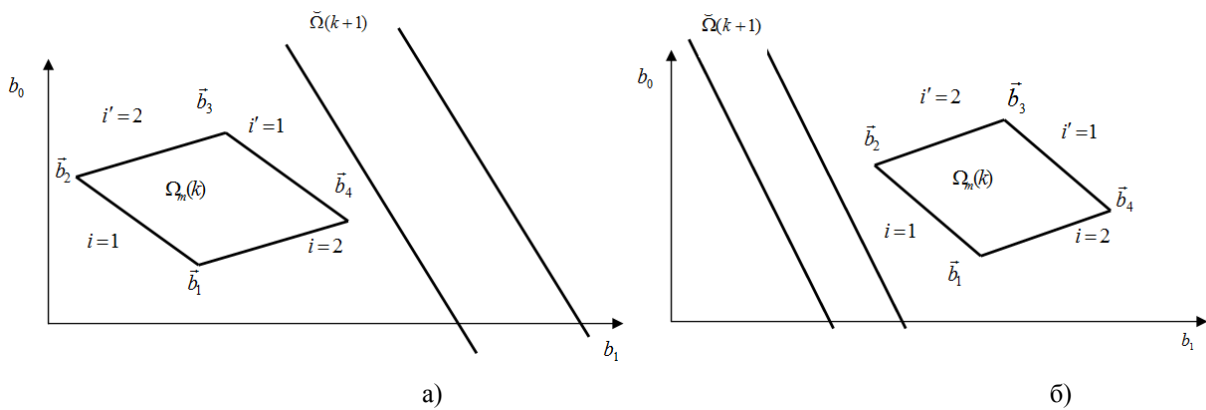


Рисунок 2 - Ілюстрація до встановлення несумісності ІСЛАР

З іншого боку на рис. 2б) усі вершини $(\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4)$ паралелограма розміщені з іншого боку "смуги" і відповідно $L'_s(k) > 0 \quad \forall s = 1, \dots, 4$. Проте ІСЛАР як і у попередньому випадку є несумісною.

Зазначені випадки не передбачено в комп'ютерних реалізаціях методу локалізації параметрів, що неодноразово призводило до непередбачуваних результатів цих реалізацій.

Проведений аналіз, дозволяє сформулювати критерій інформативності $k+1$ рівняння: якщо для усіх вершин певної i -ї грані виконуються умови

$$L_s(k) > 0, \quad \forall s = 1, \dots, 2^{m-1} \quad (20)$$

та одночасно

$$\exists L_s(k) < 0, \quad s = 1, \dots, 2^m, \quad (21)$$

то $k + 1$ інтервальне рівняння є інформативним і ця грань переміщується (в сторону зменшення розмірів паралелотопа) до моменту дотику її найближчої вершини до межі "гіперсмуги". Звідси на k -тій ітерації для цієї грані справедливим є такий вираз:

$$\min_{s=1, \dots, 2^{m-1}} \{L_s(k+1)\} = 0 \quad (22)$$

Зауважимо, що переміщення i -ї грані паралелотопа у випадку виконання умов (20), (21) означає переміщення нижньої межі $y_i^-(k+1)$ інтервалу $[y_i^-(k+1), y_i^+(k+1)]$. Користуючись зробленим зауваженням, знайдемо рекурентну формулу для $L_s(k+1)$. Для цього підставимо у формулу (17) замість i -ї компоненти $y_i^-(k+1)$ вектора $\vec{Y}_s(k+1)$ її значення $y_i^-(k) + \delta_i^-(k+1)$ з виразу (15), отримаємо таку рекурентну формулу:

$$\vec{b}_s(k+1) = \vec{b}_s(k) + \vec{f}_i \cdot \delta_i^-(k+1), \quad (23)$$

де \vec{f}_i - i -тий стовпець матриці F_m^{-1} .

З урахуванням (18), розрахункова формула для значень $L_s(k+1)$ $k+1$ -ї ітерації матиме такий вигляд:

$$L_s(k+1) = L_s(k) - \vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i \cdot \delta_i^-(k+1) \quad (24)$$

Із виразу (24) бачимо, що при додатному значенні $\delta_i^-(k+1)$, одночасне виконання умов (20) та (22) можливе тільки у випадку $\vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i > 0$. З врахуванням цього прирівняємо праву частину рівняння (24) до нуля відповідно до умови (20) та із урахуванням умов (21) після нескладних перетворень отримаємо співвідношення для розрахунку значення $\delta_i^-(k+1)$

$$\delta_i^-(k+1) = \begin{cases} \min_{s=1, \dots, 2^{m-1}} \{L_s(k) / |\vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i|\}, \\ \text{якщо } (L_s(k) > 0, s = 1, \dots, 2^{m-1}) \wedge (\vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i \neq 0) \wedge (\exists L_s(k) < 0, s = 1, \dots, 2^m) \\ 0, \text{ якщо } L_s(k) \leq 0 \\ \text{"розв'язок ІСЛАР відсутній"} \text{ якщо } (L_s(k) > 0, \forall s = 1, \dots, 2^m) \end{cases} \quad (25)$$

Дотримуючись вище викладених міркувань отримаємо вираз для розрахунку $\delta_i^+(k+1)$

$$\delta_i^+(k+1) = \begin{cases} \min_{s=1, \dots, 2^{m-1}} \{L'_s(k) / |\vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i|\}, \\ \text{якщо } (L'_s(k) > 0, s = 1, \dots, 2^{m-1}) \wedge (\vec{\phi}^T(\vec{x}_{k+1}) \cdot \vec{f}_i \neq 0) \wedge (\exists L'_s(k) < 0, s = 1, \dots, 2^m) \\ 0, \text{ якщо } L'_s(k) \leq 0 \\ \text{"розв'язок ІСЛАР відсутній"} \text{ якщо } (L'_s(k) > 0, \forall s = 1, \dots, 2^m) \end{cases} \quad (26)$$

Отже для реалізації однієї ітерації, запропонованого локалізаційного методу необхідно виконати таку послідовність обчислень [4]:

1. Розрахувати значення скалярних функцій $L_s(k)$ та $L'_s(k)$ для усіх вершин паралелотопа.

2. Обчислити $\delta_i^-(k+1)$ та $\delta_i^+(k+1)$, відповідно, за формулами (25) та (26).

3. Обчислити межі інтервалу $[y_i^-(k+1); y_i^+(k+1)]$ за формулою (15).

Слід зауважити, що для обчислень за формулами (25) та (26) необхідне попереднє групування усіх вершин $s = 1, \dots, 2^m$ до відповідних граней $i = 1, \dots, m$. Інша особливість, яку слід враховувати при реалізації методу є порядок вибору рівнянь із ІСЛАР, які не задають конфігурацію локалізаційної області.

Комп'ютерна реалізація методу локалізації параметрів інтервальних моделей із виділенням «насиченого блоку»

Для апробації алгоритму реалізації методу використаємо дані із прикладу побудови моделі прогнозування потужності малої гідроелектростанції (МГЕС), розглянутого у праці [8]. Метою такого розгляду є ілюстрація уведеної модифікації алгоритму, щодо можливої несумісності ІСЛАР.

Для локалізації параметрів цієї моделі у праці [8] використовували підхід, що ґрунтується на знаходженні інтервальних оцінок параметрів. Тобто метод ґрунтується на знаходженні локалізаційної області у вигляді прямокутного паралелепіпеда. Як відомо, такий метод дає можливість отримати тільки «грубу» оцінку області параметрів і відповідно забезпечує низьку точність інтервальних моделей. Тому для розв'язування зазначеної задачі використаємо метод локалізації параметрів із виділенням «насиченого блоку».

У праці [8] описана модель прогнозування потужності МГЕС:

$$y(\vec{x}) = b_1 \cdot x_1 + b_2 \cdot x_1 x_3 + b_3 \cdot \sin(x_3) + b_4 \cdot x_1 \cdot x_2^2 \quad (27)$$

де $y(\vec{x})$ - згенерована електроенергія, x_1 - реактивна потужність, x_2 - напір на МГЕС, x_3 - рівень води на гідропості.

У табл. 1 представлено вихідні дані генерованої електроенергії МГЕС та фактори впливу на обсяги згенерованої електроенергії [2].

Розглянемо алгоритм обчислень. Програмний модуль для розв'язування зазначеної задачі створено в середовищі Mathcad.

Крок 1. Вибір початкових даних та формування «насиченого блоку» ІСЛАР згідно структури (27):

$$\begin{cases} 1087,2 \leq 182,5 \cdot b_1 + 1186,25 \cdot b_2 + 0,215 \cdot b_3 + 3861,7 \cdot b_4 \leq 1211,28; \\ 1105,32 \leq 182,6 \cdot b_1 + 1059,08 \cdot b_2 - 0,465 \cdot b_3 + 3863,816 \cdot b_4 \leq 1231,468; \\ 1069,08 \leq 182,7 \cdot b_1 + 1004,85 \cdot b_2 - 0,706 \cdot b_3 + 4035,843 \cdot b_4 \leq 1191,092; \\ 1069,08 \leq 182,7 \cdot b_1 + 908,019 \cdot b_2 - 0,967 \cdot b_3 + 4035,843 \cdot b_4 \leq 1191,092. \end{cases}$$

Таблиця 1 - Зведені дані про щоденну генеровану електроенергію МГЕС

№	Реактивна потужність	Напір	Рівень води на гідропості	Згенерована електроенергія (кВт/доба)
i	x_{1i}	x_{2i}	x_{3i}	$[y^-(x_i); y^+(x_i)]$
1	182,5	4,6	6,5	[1087,2; 1211,28]
2	182,6	4,6	5,8	[1105,32; 1231,468]
3	182,7	4,7	5,5	[1069,08; 1191,092]
4	182,7	4,7	4,97	[1069,08; 1191,092]
5	182,7	4,7	5,45	[1087,2; 1211,28]
6	182,7	4,6	6,8	[906; 1009,4]
7	182,9	4,7	7,5	[1069,08; 1191,092]
8	183	4,7	11,9	[1087,2; 1211,28]
9	183,1	4,7	14,9	[1087,2; 1211,28]
10	183,1	4,7	12,5	[1105,32; 1231,468]
11	183,1	4,7	9,8	[1123,44; 1251,656]
12	183,1	4,55	10,4	[1087,2; 1211,28]
13	183,1	4,65	15,5	[1087,2; 1211,28]

продовження табл. 1

14	183,1	4,6	13,7	[1141,56; 1271,844]
15	184,6	4,65	13,7	[1105,32; 1231,468]
16	184,6	4,7	14,9	[1159,68; 1292,032]
17	184,6	4,7	17	[1159,68; 1292,032]
18	184,6	4,7	14	[1159,68; 1292,032]
19	184,6	4,7	12,8	[1159,68; 1292,032]
20	184,7	4,7	12,5	[1141,56; 1271,844]

Оскільки структура досліджуваної моделі містить чотири невідомі параметри, то матриця «насиченого блоку» має розмірність 4x4. Дані для цієї матриці, тобто базові рівняння, обирали, спираючись на міркування щодо забезпечення максимальної кількості ітерацій обчислювального процесу з метою його ґрунтовного дослідження. Для цього використали дані перших чотирьох спостережень із табл. 1, що утворюють «насичений блок».

Крок 2. Формування матриці F та обчислення F^{-1} . Ілюстрація - рис. 3.

Крок 3. Групування векторів $\vec{Y}_s(k)$, ($s = 2^m = 16$), використовуючи інтервальні значення $y_i^-(k)$ та $y_i^+(k)$ кожного із m базових інтервальних рівнянь «насиченого блоку». Ілюстрація - рис. 4.

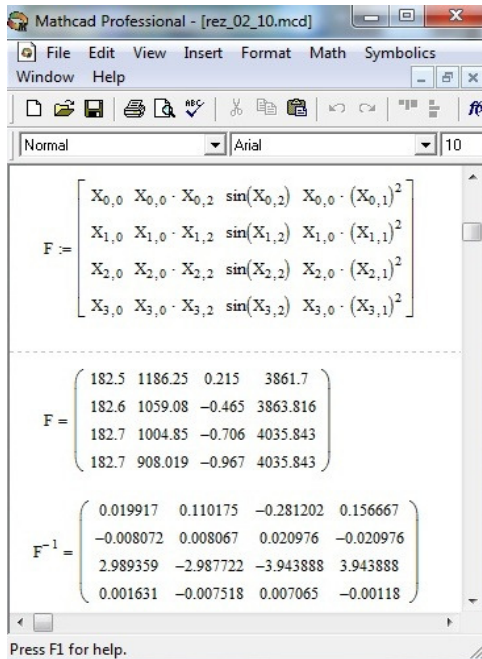


Рисунок 3 – Ілюстрація процедури формування матриці F та обчислення F^{-1}

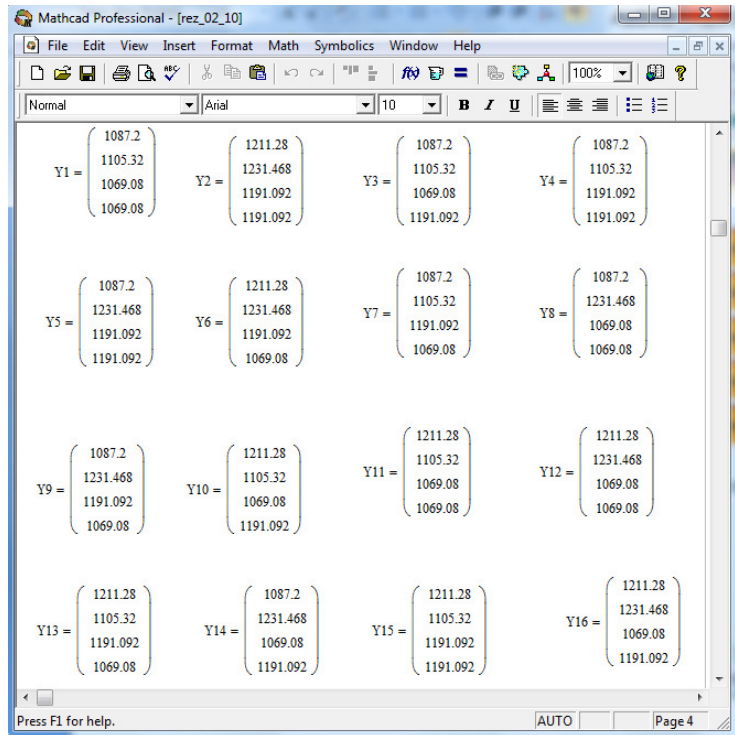


Рисунок 4 – Ілюстрація процедури формування вектора $\vec{Y}_s(k)$, ($s = 2^m = 16$)

Крок 4. Обчислення координат вершин паралелопа $\Omega_m(k)$ на $k=0$ ітерації за формулою (17). Ілюстрація – рис. 5.

Даний етап обчислень передбачений формулами (18), (19), однак для детальнішого опису процесу реалізації алгоритму продемонстровано процедуру знаходження вершин як окрему процедуру обчислень.

Крок 5. Уведення додаткового рівняння до «насиченого блоку» на $k = 1$ ітерації та здійснення обчислень за формулами (18), (19). Ілюстрація – рис. 6.

Додаткове рівняння, уведене на $k = 1$ ітерації:

$$1087,2 \leq 182,7 \cdot b_1 + 995,715 \cdot b_2 - 0,74 \cdot b_3 + 4035,843 \cdot b_4 \leq 1211,28$$

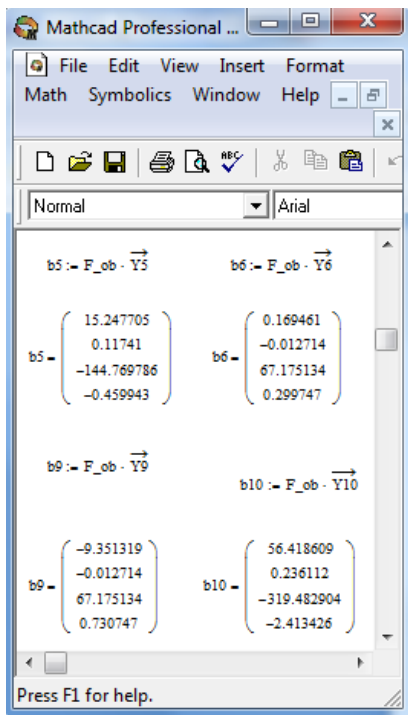


Рисунок 5 – Ілюстрація фрагменту процедури обчислення вершин паралелопа $\Omega_m(k)$

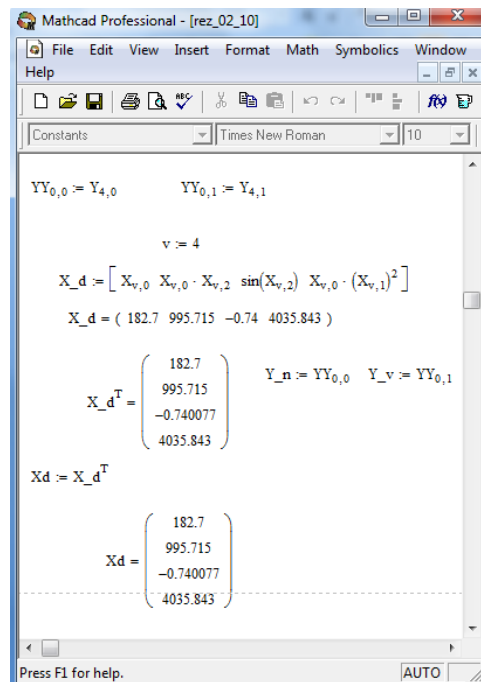


Рисунок.6 – Ілюстрація етапу введення додатково-го рівняння до «насиченого блоку»

На рис. 6 $\vec{\phi}^T(\vec{x}_{k+1})$ (формули (18), (19)) позначено як «Xd», а значення y_{k+1}^- та y_{k+1}^+ «YY0,0» та «YY0,1» відповідно.

Крок 6. Обчислення відстаней $L_s(k)$ та $L'_s(k)$ між вершинами паралелопа та відповідною межею «гіперсмуги» $\tilde{\Omega}(k+1)$. Ілюстрація - рис. 7.

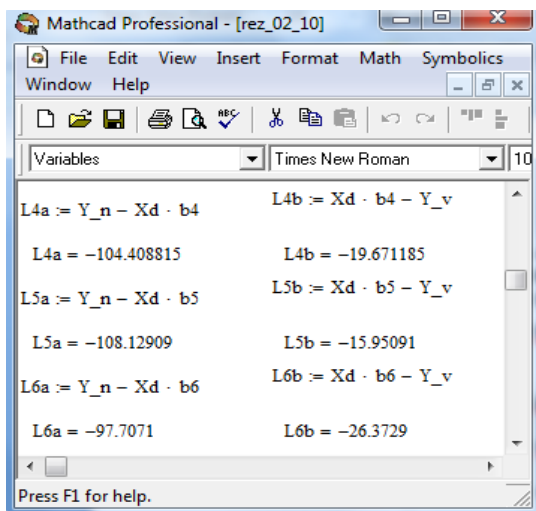


Рисунок 7 – Ілюстрація фрагменту обчислень $L_s(k)$ та $L'_s(k)$, $k=1$.

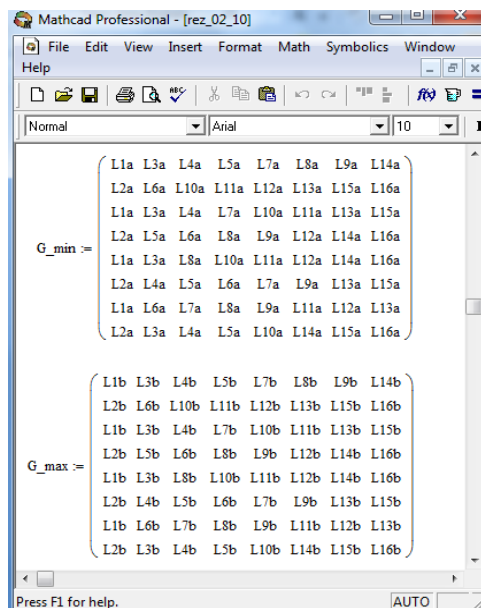


Рисунок 8 – Ілюстрація групування вершин паралелопа по гранях

Крок 7. Перевірка умов (20), (21) для усіх вершин, шляхом групування їх по гранях. Групування здійснено, спираючись на твердження, що якщо вершини $\vec{b}_s(k)$ належать одній грані паралелепіпеда, то відповідні вектори $\vec{Y}_s(k)$ в формулі (17) мають хоча б по одній спільній компоненті $y_i^-(k)$ чи $y_i^+(k)$.

Для зручності візуального представлення обчислення уведено матриці G_{\min} та G_{\max} . Кожен рядок зазначених матриць відображає відстані відповідних вершин, розміщених на одній грані до межі «гіперсмуги». Причому матриця G_{\min} відображає значення відстані від вершин до нижньої межі «гіперсмуги», а матриця G_{\max} – до верхньої межі «гіперсмуги» відповідно (рис. 8).

Крок 8. Аналіз знаків $L_s(k)$ та $L'_s(k)$ та перевірка «інформативності» додаткового інтервального рівняння. Ілюстрація – рис.9.

На $k = 1$ ітерації встановили, що, враховуючи умови (20), (21), уведене в систему на $k = 1$ ітерації інтервальне рівняння є «неінформативним» і переміщення граней не здійснювали, а продовжили обчислення згідно формул (18), (19) на $k = 2$ ітерації, тобто перейшли на крок 5.

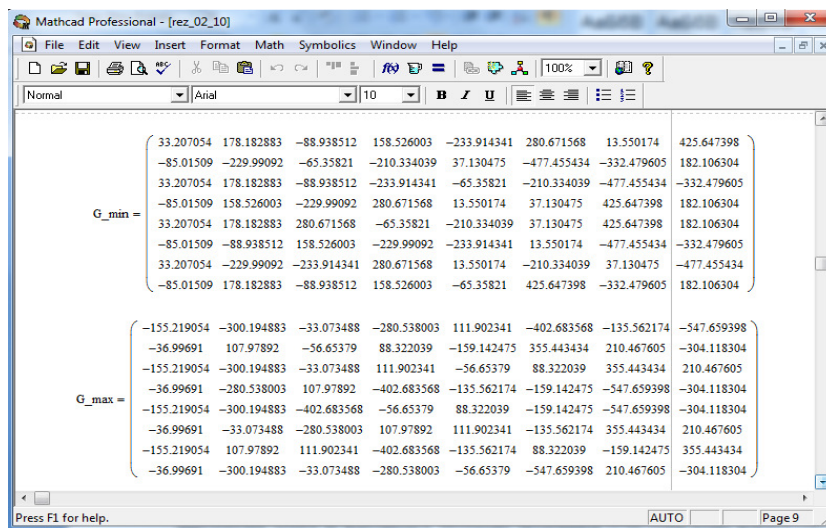


Рисунок 9 – Ілюстрація процедури перевірки «інформативності» рівняння, введеного в ІСЛАР на $k = 1$ ітерації

Крок 5. Уведення додаткового рівняння до «насиченого блоку» на $k = 2$ ітерації та здійснення обчислень за формулами (18), (19).

Додаткове рівняння, уведене на $k = 2$ ітерації:

$$906 \leq 182,7 \cdot b_1 + 1242,36 \cdot b_2 + 0,494 \cdot b_3 + 3865,932 \cdot b_4 \leq 1009,4$$

Згідно формул (18) та (19) обчислили відстань між вершинами та відповідними межами «гіперсмуги», утвореної додатковим рівнянням (рис. 10).

Перевіряючи виконання умов (25) та (26) на $k=2$ ітерації, отримали повідомлення про несумісність системи, адже згідно умови (26):

$$\text{"розв'язок ІСЛАР відсутній" якщо } (L'_s(k) > 0, \forall s = 1, \dots, 2^m).$$

Іншими словами додане обмеження, задане інтервальним рівнянням, утворило «смугу», верхня межа якої не має жодної спільної точки з областю розв'язків, отриманою на попередній ітерації.

Встановлений факт означає, що на основі рівняння (27) та наведених даних з таблиці 1 неможливо побудувати інтервальну модель генерованої в МГЕС електроенергії. Хоча у праці [8] зазначена модель була побудована із використанням методу локалізації області розв'язків ІСЛАР у вигляді багатовимірного прямокутного паралелепіпеда. Можна припустити, оскільки зазначений метод локалізації багаторазово використовує симплекс-процедури (багаторазове розв'язування задачі лінійного програмування), то це призводить до накопичення похибок в процесі обчислення, що, як результат, породжує хибні розв'язки. Модифікований у даній праці метод локалізації із виділенням «насиченого блоку» має в цьому сенсі перевагу, оскільки обчислювальна процедура його реалізації не призводить до накопичення похибок заокруглень.

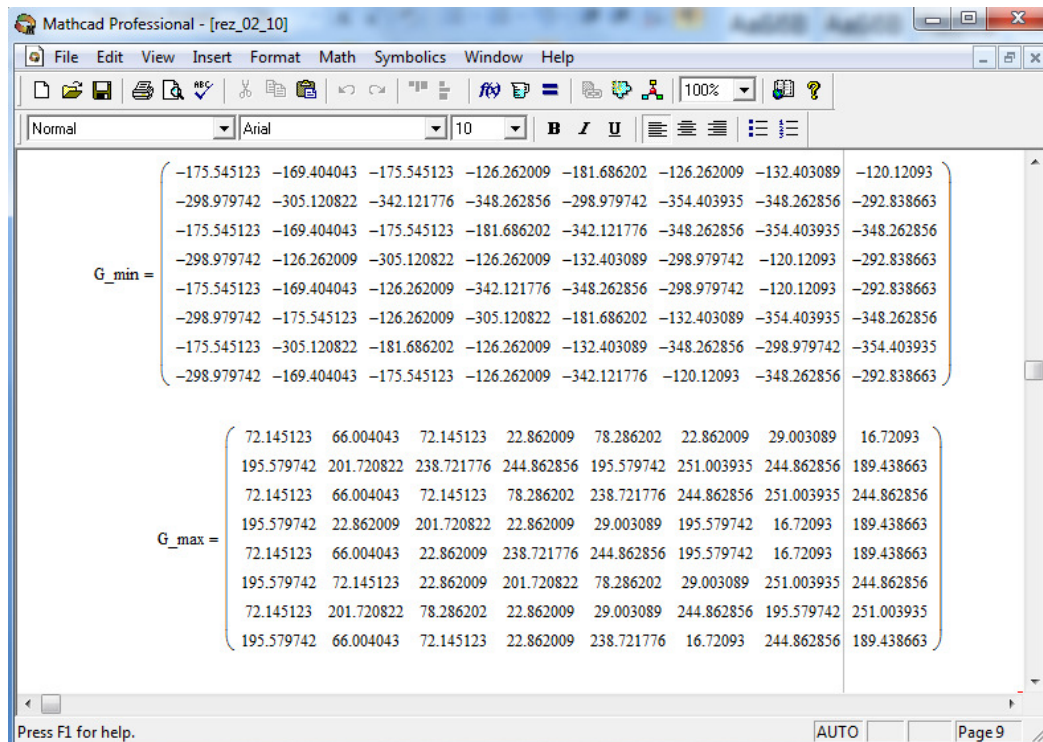


Рисунок 10 – Ілюстрація результатів обчислення відстаней між вершинами та межами «гіперсмуги» на $k=2$ ітерації

Слід зауважити, що при реалізації алгоритму без використання зазначеної умови на основі існуючої версії алгоритму у вище зазначеному випадку отримували зациклення, причина якого не була очевидною. Уведення розглянутої в статті модифікації методу та алгоритму забезпечило усунення зациклення та забезпечило надійність та відмово стійкість розробленої програмної реалізації.

Висновки

1. Розглянуто метод ідентифікації області параметрів інтервальних моделей із виділенням «насиченого блоку» у базовій інтервальній системі лінійних алгебричних рівнянь, яка побудована на основі даних з інтервальною невизначеністю. Встановлено, що основною математичною задачею при реалізації цього методу є задача оцінювання множини розв'язків ІСЛАР.

2. Встановлено умови застосування методу, зокрема побудовано стійкі кількісні критерії сумісності базової ІСЛАР і на цій основі модифіковано базові формули реалізації методу та базовий алгоритм.

3. Із застосуванням комп'ютерної реалізації методу на прикладі побудови математичної моделі для прогнозування генерованої електроенергії на МГЕС показано позитивний ефект від проведених досліджень та удосконалень методу. Зокрема, для удосконаленого алгоритму забезпечено прогнозованість результату та відсутність зациклювань у рекурентній схемі.

Список використаних джерел

1. Вошинин А. П. Оптимизация в условиях неопределенности / А. П. Вошинин, Г. Р. Сотиров. – М.: МЭИ – София: Техника, 1989. – 224 с.
2. Шарый С. П. Интервальные алгебраические задачи и их численное решение // Дис. доктора физ.-математ. наук / С. П. Шарый. – Новосибирск: Ин-т вычисл. Технологий СО РАН, 2000. – 322 с.
3. Дивак М. П. Ідентифікація моделей об'єктів в умовах інтервальної невизначеності на основі методів аналізу інтервальних даних / М. П. Дивак, П. Г. Стахів // Пр. міжн. конф. з управління «АВТОМАТИКА-2000», Львів 11–15 вересня 2000: в 7-ми томах. – Львів: Держ. НДІ інформ. інфраструкт., 2000. – Т. 2. – С. 90–97.
4. Дивак М. П. Задачі математичного моделювання статичних систем з інтервальними даними: монографія / за ред. М. П. Дивака. – Тернопіль: Економічна думка, 2011. – 216 с.
5. Бочков А. Ф. Интервальный анализ как альтернатива регрессионному анализу / А. Ф. Бочков, А. П. Вошинин, Г. Р. Сотиров // Заводская лаборатория. – 1990. – №7. – С. 76 – 81.

6. Воцинин А. П. Планирование оптимального насыщенного эксперимента в задачах построения интервальных моделей / А. П. Воцинин, М. П. Дывак // Заводская лаборатория. – 1993. – №1. – С. 56–59.
7. Дывак М. П. Метод локалізації гарантованих оцінок в задачах параметричної ідентифікації // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2000. – №4. – С.12–17.
8. Дывак М.П. Інтервальна модель для прогнозування потужності малої гідро-електростанції «Топольки» / М.П. Дывак, В. І. Манжула, Ю. П. Франко // Енергетика та електрифікація.- 2008.- №11 (303). - С. 21-29.

Відомості про авторів

Дывак Микола Петрович – д.т.н., професор, завідувач кафедри комп'ютерних наук, декан факультету комп'ютерних інформаційних технологій ТНЕУ.

Пукас Андрій Васильович – к.т.н., доцент кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій ТНЕУ.

Олійник Ірина Степанівна – аспірант кафедри комп'ютерних наук факультету комп'ютерних інформаційних технологій ТНЕУ.