

## Посилання на статтю

Назимко В.В. Автоматизоване прийняття рішень у ході оперативного управління проектом / В.В. Назимко, Є.О. Юшков, С.А. Зінченко // Управління проектами та розвиток виробництва: Зб.наук.пр. – Луганськ: вид-во СНУ ім. В.Даля, 2013 - №3(47). - С. 26-35. - Режим доступу: <http://www.pmdp.org.ua/>

УДК 658.012.23

**В.В. Назимко, Є.О. Юшков, С.А. Зінченко**

### **АВТОМАТИЗОВАНЕ ПРИЙНЯТТЯ РІШЕНЬ У ХОДІ ОПЕРАТИВНОГО УПРАВЛІННЯ ПРОЕКТОМ**

Розроблено метод автоматизованого оперативного управління проектом в умовах браку часу, збурень і шумів на основі порівняння вхідного вектору і інформативної частини правил, що застосовуються для вибору команд управління, які є близькими до оптимальних. Обґрунтовано критерій оптимальності управління на основі врахування релевантності правил. Рис. 1, дж. 16.

Ключові слова: оперативне управління проектом, правила, оптимальна команда, релевантність.

**В.В. Назимко, Е.А. Юшков, С.А.Зинченко**

### **АВТОМАТИЗИРОВАННОЕ ПРИНЯТИЕ РЕШЕНИЙ В ХОДЕ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ПРОЕКТОМ**

Разработан метод автоматизированного оперативного управления проектом в условиях нехватки времени, возмущений и шумов на основе сравнения входного вектора и информативной части правил, применяемых для выбора команд управления, которые близки к оптимальным. Обоснован критерий оптимальности управления на основе учета релевантности правил. Рис. 1, ист. 16.

**V.V. Nazimko, E.A. Yushkov, S.A. Zinchenko**

### **AUTOMATED DECISION MAKING DURING PROMPT PROJECT MANAGEMENT**

Method for automated prompt management of a project under time pressure, disturbances and information noise is developed on the base of comparing the input vector and informative part of the rules that are close to optimal and used for selection of managerial teams. An optimality criterion of management considering the roles relevance is proposed.

JEL D81

## **ВСТУП**

**Загальна постановка проблеми.** У ході оперативного управління проектом ймовірність прийняття помилкового рішення менеджером проекту або членом його команди підвищується зі скороченням часу. Отже виникає необхідність автоматизованого прийняття рішень. Але така задача наштовхується на складності, пов'язані з тим, що вхідні дані, які використовуються у ході управління проектом зазвичай поступають у синтаксичній формі, а команди мають вербальний вигляд. Натомість

автоматизовані системи призначені для маніпулювання числовими даними. Підхід до вирішення вказаної проблеми обговорюється у даній статті.

**Аналіз результатів сучасних досліджень та виділення невирішеної частини проблеми.** Автоматизовані системи управління широко застосовуються у промисловості й інноваційних технологіях для забезпечення автоматизованого управління складними системами [1]. Характерно, що вхідні дані, які використовуються для вироблення команд управління і самі сигнали управління мають зазвичай числову форму. Приклади застосування автоматизованих систем прийняття групових рішень, у яких фігурують символічні дані відомі у галузі освіти для автоматизації навчального процесу [2]. Проте застосування систем прийняття оптимальних рішень у практиці відоме більше у промисловості для автоматизованого управління установками, які забезпечують числову вхідну інформацію, і керуються також сигналами, що мають цифрову форму.

Проектування автоматизованих систем прийняття оперативних рішень, що мають нечислове вираження є трудоємним і потребує багато колективних зусиль програмістів, спеціалістів автоматики, а також фахівців-технологів. Значна трудомісткість проектування систем притаманна навіть при умовах застосування потужного об'єктно-орієнтованого підходу. Так зазначений колектив спеціалістів повинен створити інформаційну модель процесу, що автоматизується, ідентифікувати об'єкти, їх атрибути та зв'язки.

Має бути розроблена модель станів та подій з указівкою взаємних зв'язків, та створені моделі процесів з діаграмами потоків даних. Вказані моделі реалізуються через один з алгоритмів, наприклад кінцевих автоматів з указанням переходів та їх ймовірностей. Останнім часом набувають популярності Байєсовські та штучні нейронні мережі. Досить зазначити, що пропуск або втрата кількох а то й одного суттєвого елемента (наприклад можливого зв'язку між об'єктами або переходу між станами) досить для того, щоб вся модель працювала неадекватно і давала суттєві помилки у невизначеній множині значень вхідних факторів.

Сучасні інформаційні системи управління komponуються з ряду окремих підсистем, наприклад текстових процесорів, електронних таблиць, систем управління базами даних, систем бухгалтерського обліку, управління запасами, комунікаціями і насамкінець управління проектами. Отже помилка у прийнятті оперативного рішення у ході управління проектом може звести нанівець всю роботу суміжних систем.

Глибину причину складності автоматизації управління проектом у концентрованій формі висвічує проф. Рач В.А., який зауважує, що «...управление проектами и программами... располагается на стыке естественных и гуманитарных наук...» [3], а тому методологія досліджень і інструментарій так званих наук сильної версії (точних наук) поступається методологіям наук слабкої версії тобто гуманітарних наук. Разом з тим враховуючи потужні результати і напрацювання точних наук у вирішенні проблем автоматизації, доцільно продовжувати пошуки і можливості використання досягнутого потенціалу у новій галузі знань, якою є управління проектами. Отже **метою дослідження** є пошуки можливості автоматизації прийняття рішень у ході оперативного управління проектом.

#### **ОСНОВНІ ОТРИМАНІ РЕЗУЛЬТАТИ**

**Викладення основного матеріалу дослідження.** Обґрунтування алгоритму для вибору оптимальної команди управління. Перш за все зупинимось на уточненні терміну «оперативне» управління. На сьогодні цей термін використовується зазвичай у якісному контексті [4] і не має строгого визначення. Тому ми будемо притримуватись чіткого визначення терміну «оперативне» управління за аналогією [5]. Під оперативним управлінням проектом будемо

розуміти таке, що здійснюється в умовах браку часу на прийняття строго оптимального рішення:

$$T_c = f(t, k_1, \dots, k_n) \leq T_0, \quad (1)$$

де  $T_c$  – поточний ліміт часу, який доступний для прийняття оперативного вирішення чергового завдання проекту,  $f$  – функція, яка залежить від поточного часу  $t$ , а також параметрів проекту  $k_1, \dots, k_n$ ;  $T_0$  – час, необхідний для прийняття оптимального рішення.

Це означає, що рішення, які оперативно приймаються у ході виконання проекту будуть близькі до оптимальних у межах певного довірчого інтервалу. Для задоволення цієї умови буде обґрунтована цільова функція і розроблено спеціальний алгоритм пошуку рішень або команд управління проектом, близьких до оптимальних.

Історія розвитку систем підтримки прийняття рішень (СППР) вичерпно викладена Пауером [6]. СППР виникли в результаті злиття управлінських інформаційних систем і систем управління базами даних. Для аналізу і виробок пропозицій в СППР використовуються різні методи [7]. Найчастіше застосовують інформаційний пошук, інтелектуальний аналіз даних, пошук знань у базах даних, міркування на основі прецедентів, імітаційне моделювання, еволюційні обчислення і генетичні алгоритми, нейронні мережі, ситуаційний аналіз, когнітивне моделювання та ін.

Аналіз свідчить про велику кількість накопичених ідей, підходів і алгоритмів які можуть бути кандидатами для включення в автоматичну систему вибору оптимальної команди управління проектом. Поточне рішення або команда оперативного управління є реакцією на певні вхідні сигнали. Складність прийняття рішення пов'язана не тільки з браком часу, але й з тим, що у ході реалізації проекту зазвичай виникає необхідність реагування на вектор вхідних сигналів. Причому розмірність вектора весь час змінюється, а до корисних сигналів підмішуються випадкові збурення і шуми. Фактично оперативне управління проектом у реальності зазвичай здійснюється в динамічному оточенні в умовах параметричної і структурної невизначеності.

Останнім часом для оперативного управління в таких умовах набуває популярності використання бази знань, основою яких є правила [8]. Правила створюються експертами з використанням сучасних методів. Системи підтримки прийняття рішень на основі використання готових перевірених правил гарантують гнучкість, а також оперативність або моторність (agility) управління [9], роблячи його ефективним в умовах випадкових збурень і перешкод.

Правило вибору команди управління проектом складається з двох частин: інформативної і власне команди. Інформативна частина правила пояснює умови прийняття команди управління. Механізм вибору оптимальної команди пролягає через співставлення вхідного вектору з інформаційною частиною правила, обидва з яких виражені у лінгвістичній формі. Наприклад типове правило для проекту швидкісного буріння виглядає наступним чином: *«якщо метаноемність вугілля висока, швидкість буріння середня, навантаження на лаву середнє, ймовірність раптових викидів середня то необхідно взяти помірні кредити»*. Інформаційна частина правила помічена курсивом. Найбільш близьким до оптимального буде таке правило, інформаційна частина котрого якомога краще співпадає з поточним вхідним вектором, який має наприклад такий вигляд: *«метаноемність вугілля висока, швидкість буріння середня, навантаження на лаву високе ймовірність раптових викидів середня»*.

Можна іти шляхом використання форм і зарезервувати стандартні поля і навіть ідентифікатори для заповнення цих полів у ході формування вхідних

векторів. Проте в такому випадку втрачається гнучкість системи підтримки прийняття рішень і її оперативність. Більш того, у реальних умовах невизначеності і наявності шумів та випадкових збурень достовірність прийняття правильних рішень буде проблематичною. Саме тому ми притримуємось ідеології, яка звертається до використання символічних змінних вільного формату для компоновки вхідних векторів [3].

Широкий спектр засобів порівняння символічних строк охоплює діапазон від строгих чисельних методів теорії автоматів до систем і алгоритмів обробки семантичних структур і баз даних а також маніпулювання логічними поняттями. Оскільки управління проектами здійснюється головним чином за допомогою вербальних команд, ми зміщуємо увагу на алгоритми, що маніпулюють семантичними об'єктами.

Такі програмні засоби як стандартні процедури обробки строкових змінних (наприклад пошук фрагменту в символічній строці) не можуть самостійно вирішити таку складну проблему як вибір оптимальної команди управління, хоча як допоміжні засоби вони заслуговують на увагу.

Серед програмних засобів логічного маніпулювання поняттями слід нагадати засіб логічного програмування ПРОЛОГ [10], який заснований на обчисленні предикатів, математичні основи якого закладені так званою логікою першого порядку [11]. Враховуючи стохастичний характер оточуючого середовища, в якому реалізуються сучасні проекти, слід зробити висновок про те, що ПРОЛОГ не має значної перспективи, оскільки його правила і предикати не враховують випадкової варіації і нечітких меж.

Останнім часом набуває популярності система WordNet, який є електронним тезаурусом, що побудований у вигляді мережі симантичних наборів або симсетів. Система започаткована дослідницькою групою Міллера з Принстонського університету для англійської мови [12]. На даний час аналоги системи розбудовуються для японської, російської, французької, естонської і багатьох інших мов. Причому кожні два роки збирається міжнародна конференція (у 2014 році буде уже шоста) для обговорення проблем розвитку національних WordNet-ів.

Спочатку система розвивалась як інструмент для лінгвістів, проте деякий час поспіль стала більш популярна у програмістів і спеціалістів з інформаційних технологій. Система використовується у довідникових мережах, та як інструмент швидкого і ефективного пошуку інформації. Важливо, що теоретичні основи системи WordNet запозичені з психології. Фактично систему намагаються навчити мислити як це робить людина. Тому база знань для функціонування системи побудована на так званих симсетах. Мережа симсетів є деревом, у якому основні поняття виступають як предки, що породжують величезні розмаїття потомків.

Слова англійської мови були розділені на чотири великі групи: іменники, дієслова, прикметники і прислівники. Потомки та предки у симсетах пов'язані складними стосунками. Основними є гіперонім (breakfast → meal), гіпонім (meal → lunch), меронім: (table → leg), антонім (leader → follower). Окрім того використовуються лексичні, контекстні і інші зв'язки між словами у середині симсетів. Одні і ті ж слова можуть бути у різних симсетах. Саме гіпонімія є основою для формування синтаксичних мереж.

У англійській мові виділено 25 симсетів серед іменників, що охоплюють десятки а то і сотні тисяч широко вживаних слів. Так російський суспільно політичний тезаурус, побудований по типу WordNet містить 33 тисячі понять, 87 тисяч термінів 4500 з яких є багатозначними. Загальна кількість описаних відношень між поняттями у симсетах становить 130 тисяч, а число

співвідношень, що отримані за допомогою логічних властивостей системи перевищує 700 тисяч.

Сьогодні уже очевидно, що WordNet може зайняти провідне місце серед систем автоматизованого маніпулювання текстами широкого вжитку. Проте автори і розробники системи усвідомлюють, що для надійного функціонування у вирішенні спеціалізованих задач мають бути розроблені детальні мережі симсетів з врахуванням спеціалізованих термінів. Наприклад навіть у авіаційній галузі є багато напрямків (проектування двигунів, систем автоматики, літальних апаратів, систем безпеки і багато інших), для яких потрібні окремі бази знань, основою яких є сотні і тисячі симсетів. Щоб скласти такі мережі симсетів, потрібні досвідчені експерти і значний час.

Ще один серйозний недолік, який поки притаманний системі, пов'язаний з тим, що співвідношення, які заведені у бази знань у вигляді симсетів не завжди спрацьовують надійно, що призводить до неоднозначних висновків, а часто і до принципових помилок. Справа в тому, що співвідношення між поняттями є неоднозначними і інколи залежать від контексту або ситуації. Саме тому часто трапляються випадки, коли неоднозначність взаємних відносин між поняттями-членами симсетів породжує хибні висновки системи. Це є особливо серйозним недоліком, коли справа йдеться про життя людей, що задіяні у проекті, чи можливість великих фінансових, матеріальних втрат, або негативних наслідків для оточуючого середовища.

Отже незважаючи на перспективність у майбутньому на даний час говорити про готовність WordNet для автоматизованого управління проектом зарано. Більш принадливим методом для реалізації кількісного порівняння двох семантичних об'єктів є алгоритм обчислення так званого коефіцієнту косинусної подібності або коефіцієнту Отсукі-Охави [13]. Цей коефіцієнт обчислюється як відстань між двома векторами згідно наступної залежності:

$$K = \frac{n(A \cap B)}{\sqrt{n(A)n(B)}}, \quad (2)$$

де у чисельнику стоїть величина, пропорційна кількості суспільних ознак об'єктів А і В, що порівнюються, а у знаменнику середнє геометричне цих об'єктів.

У випадку з вибором оптимальної команди управління проектом виникає складна задача векторизації вхідної інформації та інформаційної частини правила. Тобто треба перевести семантичну форму вхідного вектора і правила у кількісну. Останнім часом з'являються публікації, в яких описані спроби вирішити вказану складну задачу [14]. Для оцифрування семантичних структур (наприклад речення, команди, правила) автори пропонують використовувати щільність концептних вузлів семантичної мережі симсетів. Так чим нижче знаходяться слова у симсеті, тим тісніше зв'язок між ними і тим більше їх подібність. Тому подібність обліковують як коефіцієнт Отсукі-Охави, використовуючи числові значення щільності як направляючі косинуси багатовимірного простору.

Такий підхід є принадливим, проте він не гарантує високої надійності результатів співставлення текстів. Згідно [14] ефективність такого алгоритму не перевищує 80% і коливається у межах від 60% до 70%. Якщо для використання такого алгоритму у пошукових системах вказаної ефективності достатньо, то при управлінні проектом, що пов'язаний з небезпечними наслідками (наприклад небезпечними умовами підземного вуглевидобутку) необхідно гарантувати більшу надійність.

Тому звернемося до інших математичних алгоритмів знаходження відстані між двома векторами. Претендентом могла б бути відстань Кульбака – Лейблера (інформаційна дивергенція, або відносна ентропія) в теорії інформації – це несиметрична міра віддаленості один від одного двох імовірнісних розподілів [15].

У даному випадку маємо два вектори (вхідний і інформаційна частина правила), які скомпоновані з двох дискретних випадкових величин  $X$ ,  $Y$ , які приймають значення в одній множині алфавіту  $R$ , і їх розподіли задаються функціями ймовірності  $p$  та  $q$  відповідно. Тоді відстань Кульбака – Лейблера  $D_{KL}$  визначається наступною формулою:

$$D_{KL}(p, q) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}. \quad (3)$$

Для того, щоб алгоритм Кульбака-Лейблера працював адекватно, треба, щоб одні й ті ж об'єкти описувались однаковими термінами, причому атрибути об'єктів стояли на відповідних місцях. Такі вимоги вже близькі до використання форм і зарезервованих стандартних полів, а це крок назад.

З врахуванням проведеного аналізу значної принадності набуває метод Вагнера-Фішера, який є продуктом динамічного програмування, започаткованого Беллманом [16]. Метод дає змогу знайти мінімально можливу відстань Левенштейна між двома символічними послідовностями (векторами). Ця відстань визначається як сума вставок, видалень, та перестановок символів (наприклад літер алфавіту) у двох векторах або послідовностях у процесі їх порівняння. Оскільки існує велика кількість можливих комбінацій вказаних операцій для приведення однієї послідовності до іншої, використовується метод динамічного програмування Вагнера-Фішера, який знаходить таку комбінацію згаданих операцій (перестановок, видалень та вставок), яка є мінімальною. Кожна операція має свою вагу, у найпростішому випадку одиницю. Отже відстань Левенштейна визначається як сума ваг мінімально можливої кількості вказаних операцій.

Алгоритм вибору оптимального правила для вибору команди оперативного управління проектом працює наступним чином (рис. 1). На вхід системи автоматизованого вибору команди управління періодично поступає вхідний вектор, який характеризує поточний стан проекту. Вхідний вектор компонується щодобово, позмінно з полів відповідних баз даних, що накопичують оперативну інформацію про стан основних блоків проекту: бюджету, поставок, ризиків, якості і т.д. Вектор проходить через фільтр алфавіту. Оскільки сам алгоритм вибору оптимального правила є достатньо простий, практика свідчить про необхідність накладення певних обмежень на зміст інформації, з якої компонується вхідний вектор.

По-перше об'єкти повинні мати одну й ту ж назву. Оскільки проекти реалізуються у конкретних галузях, це обмеження не є критичним. Синоніми і антоніми трапляються зазвичай у побутовій мові широкого вжитку і рідко застосовуються у прикладних галузях. Наприклад такі слова як «конвеєр», «поверх», «сосна (як матеріал деревини)», «цемент» і інші терміни рідко мають заміників і у професійній лексиці є стабільними. По-друге Важливо, щоб у вхідному векторі не застосовувалась приставка «не», оскільки вона змінює смисл на протилежний а різниця між векторами (реченнями) з приставкою і без неї невелика. Такі обмеження не є принциповими і легко враховуються користувачами системи, а фільтр попереджає користувача про видалення певних слів.

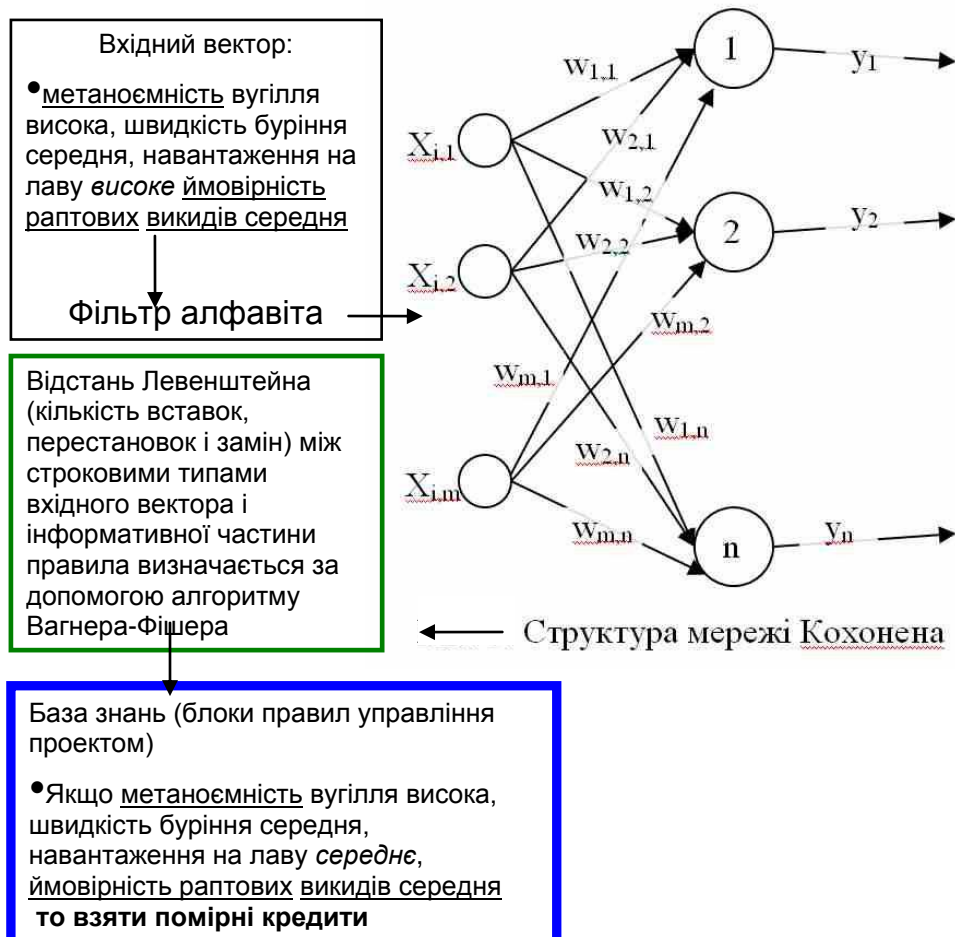


Рис. 1. Схема вибору правила, близького до оптимального

Отже вхідний вектор потрапляє на класифікатор у вигляді штучної мережі Кохонена. Класифікатор сортує вхідні вектори і визначає блок проекту, до якого відноситься певна група правил. Це підвищує швидкість автоматизованого прийняття рішення. Далі очищений вектор подається на блок порівняння з інформаційною частиною правил, що належать до вибраного блоку управління проектом. Правила складаються експертами заздалегідь і накопичуються у базі правил. Після порівняння система вибирає те правило, інформаційна частина якого найбільш близька до вхідного вектору. Команда такого правила і є найбільш близькою до оптимальної.

Може так статися, що ні в одному з існуючих правил немає інформаційної частини, яка була б близькою до вхідного вектору. У цьому випадку усі відстані між вхідним вектором і інформаційними частинами правил нормуються, і максимальна відстань Левенштейна приймається за одиницю або 100%. Заздалегідь в систему вводиться гранична відносна відстань, яка вважається критичною, наприклад 30%. Тоді у разі відсутності правила, інформаційна частина якого б була ближче ніж 30%, вектор вважається суттєво новим і система видає завдання для складання правила, яке б адекватно реагувало на

новий вектор. Таким чином здійснюється поповнення бази знань новими правилами.

Окрім введення нових правил можуть уточнюватися існуючі у разі коли вхідний вектор і інформаційна частина правила мають деякі відмінності. Для уточнення команди у такому випадку використовується алгоритм нечіткої логіки. Саме цим прийомом підтримується ідеологія управління проектом інструментарієм наук слабкої версії [3]. Незважаючи на це, практика свідчить про те, що точність управління з застосування фуззи логіки підвищується у порівнянні з використанням чітких величин вхідних сигналів і команд управління.

Насамкінець треба наголосити, що розроблена система автоматизованого вибору оптимального рішення або команди оперативного управління проектом працює у режимі радника. Остаточний вибір команди приймається менеджером проекту або членом його команди. Проте досвід показує, що у ході промислових випробувань системи довіра членів команди до правил зростає з накопиченням досвіду і бази знань, а ефективність оперативного управління збільшується у два-три рази.

Суттєвою перевагою розробленої системи є те, що близькі до оптимальної команди видаються на монітор разом з інформацією про вхідний вектор та поточними обмеженнями проекту. Всі рішення з інформацією про умови, у яких вони були прийняті протоколюються і архівуються. Це по-перше дає змогу провести аналіз оперативного управління проектом, а по-друге документально перевірити усі дії команди у випадках аварій і катастроф.

Система може почати працювати навіть з пустої бази знань. Правила накопичуються шляхом реєстрації усіх команд, які були видані по ходу освоєння проекту, починаючи з передінвестиційної фази. Правила, що викликаються рідко або неефективні відбраковуються з часом, а ті що використовуються часто вносять активний вклад у використання інформаційного активу проекту.

Для вибраковки неефективних правил і гарантування, що вибрані команди управління проектом будуть близькі до оптимальних у певних межах довірчого інтервалу запропоновано наступний критерій оптимальності оперативного управління:

$$J(\theta) = \Sigma(r(t) - y(t))^2 + \rho(u(t-1))^2, \quad (4)$$

де  $\theta$  – вектор параметрів, що настроюються у проекті (наприклад, величина вироблення продукту проекту, його якість, собівартість);  $r(t)$  – вхідний сигнал, у якості якого використовують фактичне поточне значення регульованої величини;  $y(t)$  – вихід системи управління проектом (очікуване значення вихідної величини, або, як кажуть, уставка);  $\rho(u(t-1)) \geq 0$  – функція штрафу;  $u$  – сигнал управління проектом. Поточне значення штрафної функції визначається згідно залежності:

$$\rho = 1/(-\Sigma r_i(t) \log_2 r_i(t)), \quad (5)$$

де  $r(t)$  – релевантність правила у базі даних; символ  $\log_2$  використовується для визначення двоїчного логарифму. Суму знаходимо за всіма попередніми дискретними кроками часу, що минув з початку реалізації проекту.

Як бачимо, за основу для визначення штрафної функції використана формула Шеннона, проте замість ймовірності застосована релевантність правила, яка є потужнішою з точки зору інформативності. Так релевантність не тільки пропорційна частоті (ймовірності) виклику правила, але, й враховує унікальність правила та його розмір (енергію, або вартість реалізації). Таким чином за допомогою цільової функції здійснюється тиск у бік більш ефективних правил, які забезпечують вибір команд управління, які є ближче до оптимальних. Таким чином відбраковка неефективних правил здійснюється за допомогою



порогового значення, а збільшення оптимальності команд управління регулюється зменшенням величини порогового значення критичної величини відстані Левенштейна.

Випробування системи автоматизованого прийняття рішень у промислових умовах під час управління проектом вуглевидобутку у 10 південній лаві ш/у Покровське засвідчило її ефективність. Протягом двадцяти діб випробувань ефективність системи підвищилась в 1,8 рази, а довіра користувачів до системи збільшувалась з накопиченням правил. Застосування системи дало змогу підвищити середній добовий вуглевидобуток з очисного вибою на 17%.

#### **ОБГОВОРЕННЯ**

**Подальшими дослідженнями** планується доповнити систему стохастичними моделями оперативного аналізу проектних ризиків.

**Висновки.** Виконано математичну формалізацію правил і законів оперативного управління проектом в умовах параметричної та структурної невизначеності, яка реалізована оптимальним регулятором, побудованим на нейронній мережі Кохонена, блоці порівняння вхідного вектору з інформаційною частиною правила, а також моделі нечіткої логіки, що дозволило розробити метод автоматизованого вибору команд управління проектом.

Вдосконалено оптимальний регулятор, який знаходить відстань Левенштейна між вхідним вектором і всіма правилами вибору альтернативних варіантів (команд) управління проектом за допомогою алгоритму Вагнера-Фішера.

Обґрунтовано векторний критерій ефективності оперативного управління проектом в умовах браку часу і ресурсних обмежень, величина якого пропорційна квадрату різниці фактичного сигналу системи і сигналу управління, а також обернено пропорційна добутку релевантності нечіткого правила, на підставі якого вибирається команда оперативного управління на двійковий логарифм цієї релевантності.

#### **ЛІТЕРАТУРА**

1. Boyer, S.A. (2010). SCADA Supervisory Control and Data Acquisition. USA: ISA – International Society of Automation. p. 179.
2. Петух, А.М. Автоматизована система підтримки групових рішень / Петух А.М., Войтко В.В., Кузьмін Є.В., Кузьміна Н.Ф. // Вісник Вінницького політехнічного інституту. – 2009. – №1. – С. 76-79.
3. Рач, В.А. Методологический инструментарий научного исследования в управлении проектами / В.А.Рач / Управление проектами та розвитком виробництва: Зб. наук. пр. – Луганськ: вид-во СНУ ім. В.Дала, 2012. – №4(44). – С.5-13.
4. Рудакова, А.В. Использование фрактальных свойств больших распределенных систем для решения задач оперативного управления / А.В.Рудакова, Н.В.Сарафаникова // Системные технологии, 2012. – 4(81). – С.128-135.
5. Методы классической и современной теории автоматического управления: учебник в 5-ти тт.; 2-е изд., перераб. и доп. Т.5: Синтез регуляторов систем автоматического управления / Под ред. К.А. Пупкова и Н.Д. Егупова. – М.: Издательство МГТУ им. Н.Э. Баумана, 2004. – 616 с.
6. Power, D.J. (2003). A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web. – <http://DSSResources.COM/history/dsshistory.html>, version 2.8, May 31, 2003.
7. Power, D.J.(2004). Specifying an Expanded Framework for Classifying and Describing Decision Support Systems, Communications of the Association for Information Systems, Vol. 13, Article 13, February 2004, p. 158-166.
8. Apostolou, D., Mentzas, G., Stojanovic, L., Thoenssen, B., & Lobo, T. P. (2010). A collaborative decision framework for managing changes in e-Government services. Government Information Quarterly, 28(1), p. 101-116.

9. Gong, Y. and Janssen, M. (2011). From policy implementation to business process management: Principles for creating flexibility and agility. *Government Information Quarterly*, 29 (Supplement 1), S61-S71.
10. Стирлинг, Л. Искусство программирования на языке ПРОЛОГ / Л. Стирлинг, Э. Шапиро. – М.: Мир, 1990. – 215 с.
11. Гильберт, Д. Основы теоретической логики / Д. Гильберт, В. Аккерман. – М., 1947. – 18 с.
12. Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller K. (1990). WordNet: An online lexical database. *Int. J. Lexicograph.* 3(4), p. 235-244.
13. Choi S.S., Cha S.H., Tappert C. (2010). A survey of binary similarity and distance measures // *J. on Systemics, Cybernetics and Informatics.* 1(8), p. 43-48.
14. Zhu G. Liu L. (2013). Automatic Organization of Programming Resources by Neural Computing // *Journal of software*, 6 (8), p. 1471-1478.
15. Kullback S., Leibler R.A. (1951). On information and sufficiency // *The Annals of Mathematical Statistics.* 1(22), p. 79-86.
16. Navarro, G. (2001). A guided tour to approximate string matching. *ACM Computing Surveys* 33 (1), p. 31-88.

Рецензент статті  
д.т.н., проф. Маєвський В.С.

Стаття надійшла до редакції  
17.08.2013 р.