

3.4 Результати оцінки ефективності алгоритмів рішення задачі цілочисельного лінійного програмування з булевими змінними на основі рангового підходу

3.4.1 Оцінка ефективності складних систем

Існує багато підходів щодо класифікації ефективності, а різноманітність визначень поняття “ефективність” свідчить про поліморфний характер даного визначення, так як кожне з трактувань характеризує термін “ефективність” з різних аспектів, важливість яких формується у відповідності до об’єкта та цілей дослідження.

Таким чином, як визначають автори роботи [265] “ефективність – це найбільш загальна, визначальна властивість будь-якої цілеспрямованої діяльності, яка з пізнавальної (гносеологічної) точки зору розкривається через категорію мети та об’єктивно виражається ступенем досягнення мети з урахуванням витрат ресурсів та часу”.

У техніці ефективність пов’язують із поняттям операції, під якою розуміється упорядкована сукупність взаємозалежних дій, вкладених у досягнення певної мети. “Ефективність операції – є ступінь відповідності реального (фактичного або очікуваного) результату операції необхідному (бажаному) або, іншими словами, ступінь досягнення мети операції” [265, 266].

Якість складної системи проявляється повною мірою лише у її функціонуванні, тобто використанням за призначенням. Тому, найбільш об’єктивна оцінка якості системи може бути отримана ефективністю її цільового застосування [265].

Ефективність системи не можна вивести повністю лише з властивостей системи, необхідно враховувати також властивості надсистеми. Як зазначається у роботі [266], ефективність неможливо спостерігати безпосередньо, її можна лише виміряти непрямыми методами.

Терміну “ефективність” відповідають два англійські терміни: “effectiveness” та “efficiency”. Перший має на увазі здатність встановлювати

бажані цілі та здатність досягати цих цілей. Сутність другого терміну полягає у співвіднесенні отриманою вигодою із ресурсами, витраченими на отримання цієї вигоди [271].

В наш час, у низці робіт йдеться розмова про теорію ефективності (наприклад, [266-268]). При цьому, в одних джерелах зазначається, що ця теорія ще не оформилася у самостійну науку [268]. А в інших, показана думка, що ця теорія цілком сформована: “об’єктом вивчення теорії ефективності є цілеспрямовані дії – операції, а предметом – закономірності, які пов’язують ефективність операції з якістю системи ..., умовами та способами її використання в операції” [265]. Теорію ефективності називають також “інструментом дослідження операцій” [267]. На Заході розвивається аналогічний напрямок – Efficiency and Productivity Analysis (аналіз ефективності та продуктивності) [270-271].

Важливим поняттям теорії ефективності є *показник ефективності*. Ефективність, як і будь-яка властивість системи, має певну інтенсивність свого прояву. Міру інтенсивності прояву ефективності називають показником ефективності W [265]. Є й розгорнуте визначення [268]: “показник ефективності великої системи – це кількісна характеристика кінцевого результату її функціонування та розвитку протягом обумовленого періоду у порівнянні з цільовим нормативом та витратою ресурсів при заданих характеристиках стану системи та впливу зовнішнього середовища, а також при заданому векторі управління”. Такий показник ефективності пропонується називати узагальненим показником ефективності великої системи.

Автори дослідження [269] відзначають, що показники ефективності структури системи управління мають відповідати чотирьом відомим вимогам, а саме, вони повинні:

- відповідати цілям і завданням процесу управління;
- мати зрозуміле фізичне поняття;
- бути чутливими до змін умов обстановки, до вироблених і реалізованих системою рішень;

– бути зручними для визначення та використання.

У найбільш загальній формі узагальнений показник ефективності великої системи пропонується будувати як деяку функцію або функціонал [268]:

$$W = \Phi(Y_K, Y_H, U_K, U_H), \quad (1)$$

де Y_K – можливий або фактично досягнутий корисний ефект (кінцевий результат) функціонування та розвитку системи; Y_H – цільовий корисний ефект (необхідний кінцевий результат) функціонування та розвитку системи; U_K – можливі або фактичні витрати кількості праці (дійсного та минулого) для отримання Y_K ; U_H – мінімальні необхідні витрати кількості праці (дійсного та минулого) для отримання Y_H .

Якщо (Y_K, U_K) розглядаються як можливі величини, то йдеться розмова про прогнозування ефективності, а у разі, коли (Y_K, U_K) фактично отримані, то показник W відображатиме фактично досягнуту ефективність за деякий період функціонування системи.

Для ухвалення рішення про досягнення визначеної мети необхідний *критерій ефективності*, тобто правило, яке дозволяє зіставляти стратегії, що характеризуються різним ступенем досягнення мети, та здійснювати спрямований вибір стратегій з множини допустимих. Критерій ефективності вводиться на основі певної концепції раціонального поведінки (вироблення рішень): придатності, оптимізації та адаптивізації [266].

1. Концепції придатності. Раціональною визнається будь-яка стратегія i , для якої значення показника ефективності W не нижче деякого необхідного рівня W^{TP} :

$$W(u) \geq W^{TP}, u \in U, \quad (2)$$

де U – множина допустимих стратегій.

Якщо показник ефективності векторний, то ця умова записується для кожного приватного показника W_i , що входить до складу показника W :

$$W_i(u) \geq W_i^{TP}, \forall i. \quad (3)$$

2. Концепція оптимізації. Раціональними вважаються ті стратегії $u \in U$, які дають максимальний ефект:

$$W(u^*) = \max_{u \in U} W(u). \quad (4)$$

Рішенням може бути множина рівноцінних оптимальних стратегій $U^* \subset U$. При цьому, показник ефективності $W(u)$ – скаляр.

3. Концепція адаптивізації. При цій концепції стратегія u може змінюватися у процесі функціонування системи. При цьому, до поняття стратегії включаються також параметри системи та її структура. Зміни проводяться на основі не тільки апріорної інформації, але також поточної та прогнозованої інформації. Множина допустимих стратегій та мета системи також можуть змінюватися. У рамках такої концепції раціональною вважається така адаптивна стратегія $u(t)$ з множини $U(t, \tau)$, яка забезпечує, наприклад, виконання наступної умови:

$$U_t(u^*(t), \tau) \geq W_i^{TP}(u(t), \tau), u(t) \cup U(t, \tau), \quad (5)$$

де t – час, τ – попередження прогнозу.

Запис τW_t означає, що показник ефективності може змінюватись у часі [266].

Найважливішим правилом під час виборів критерію ефективності є узгодженість мети операції з обраним критерієм.

Отже, критерій ефективності не можна побудувати без використання нормативів, а побудови показників ефективності наявність нормативів перестав бути обов'язковим [268].

З пояснень видно, що поняття “ефективність” і “мета” тісно пов'язані. Кількісне вивчення складних систем проводиться із залученням поняття мети. Використовується навіть термін цілеспрямована система. Системи діляться на прості і складні залежно від відсутності або присутності у тому поведінці акту рішення. Прості системи успішно вивчаються фізикою без використання поняття мети [269].

Складність системи було запропоновано формалізувати так [264]. Нехай існує Y елементів системи, для кожного з яких реалізується K рівноймовірних станів. У цьому випадку оцінка складності системи провадиться за формулою:

$$C = \log K^Y = Y \log K = -\log P_k, \quad (6)$$

де $P_k = \frac{1}{K}$ – ймовірність стану елемента.

Для нерівноймовірних станів, коли ймовірності станів елементів різні ($P_k \neq \text{const}$), складність визначається наступним чином:

$$C = -Y \sum_{k=1}^K P_k \log P_k = YH, \quad (7)$$

де H – інформаційна ентропія системи.

У додатку до технічних систем дослідження ефективності операції є завданням:

- оцінювання (вимірювання) ефективності операції;
- аналізу ефективності операції;
- оптимального синтезу ефективної операції.

Перші два завдання часто об'єднуються під загальною назвою *прямої* задачі, а третя називається зворотною [267].

Етапи вирішення прямої задачі (задача аналізу ефективності операцій):

- виявлення та формулювання цілей операції;
- обґрунтування показників ефективності операції;
- обґрунтування критерію оцінювання ефективності операції;
- розробка математичної моделі операції;
- обчислення показників ефективності операції;
- оцінювання ефективності операції;
- дослідження впливу експлуатаційно-технічних характеристик цілеспрямованої технічної системи ефективність операції;
- аналіз чутливості показників ефективності до експлуатаційно-технічних характеристик цілеспрямованої технічної системи та відбір значимих факторів.

Етапи вирішення зворотної задачі (задача синтезу ефективної операції), як інженерно-технічної, так і організаційної:

- виявлення та формулювання цілей операції. Відбір значних керованих чинників;
- обґрунтування показників ефективності операції та критерію її

оцінювання;

- побудова математичної моделі операції;
- випробування моделі операції та визначення її оптимальних характеристик; обґрунтування вимог до структури цілеспрямованої технічної системи (структурний синтез);
- обґрунтування вимог до експлуатаційно-технічних характеристик цілеспрямованої технічної системи (параметричний синтез);
- обґрунтування вимог щодо організації процесу функціонування системи (алгоритмічний синтез).

Наведений порядок дослідження ефективності може бути застосований не лише до складних технічних систем, а й до систем в інших предметних галузях.

Підходи до дослідження ефективності у різних сферах відрізняються великою різноманітністю. Однак існують і загальніші підходи, що застосовуються для оцінки ефективності широких класів складних об'єктів.

3.4.2 Показники ефективності алгоритмів

Основні поняття і визначення, які пов'язані із зображенням алгоритмів у вигляді графів і їх паралельних форм, а також взаємозв'язок між паралельними алгоритмами та архітектурою паралельних обчислювальних структур (ПОС) для їх реалізації, надані у роботі [269]. Для коректного порівняння алгоритмів, що розробляються, з відомими необхідно визначити основні показники їх ефективності, що дозволить провести порівняльний аналіз.

Найбільш широко вживаним показником ефективності роботи алгоритму є відрізок часу, що витрачається алгоритмом для вирішення поставленого завдання. Проте, такий підхід не однозначно визначає ефективність алгоритмів, оскільки цей показник залежить від типу електронно-обчислювальної машини (ЕОМ), на якій виконується алгоритм, від вибору мов програмування тощо.

При аналізі алгоритмів час їх роботи має термін числа елементарних кроків (арифметичних операцій, порівнянь тощо), необхідних для виконання цього

алгоритму на гіпотетичній ЕОМ. Іншими словами, виконання будь-якої операції вимагає одну одиницю часу.

Проте, число кроків, що виконуються алгоритмом, не однакове для різного розміру входу початкових даних. Для усунення такої неоднозначності у поведінці алгоритму при переході від одного входу до іншого можливо розглянути усі входи даного розміру n разом та визначити складність алгоритму для входу як число кроків алгоритму, у гіршому випадку за усіма цими входами.

У комбінаторних задачах оптимізації входом є комбінаційний об'єкт: граф, множина цілих чисел, сімейство кінцевих множин тощо. Щоб увести цей вхід у обчислювальній системі для вирішення, необхідно яким-небудь чином закодувати або зобразити у вигляді послідовності символів над деяким фіксованим алфавітом таким, як двійковий алфавіт. Кодування комбінаторних об'єктів можна провести будь-яким з багатьох відомих способів [269]. Оскільки вхід алгоритму зображується у вигляді послідовності або ланцюжка символів над деяким фіксованим алфавітом. Можливо не деталізувати способи кодування комбінаторних об'єктів. Визначимо розмір входу алгоритму як довжину цієї послідовності, тобто число символів у неї. При аналізі алгоритмів звичайно цікавить швидкість зростання складності алгоритму, яку описують за допомогою наступних формалізмів.

Визначення. Нехай $f(n)$ і $g(n)$ – функції, які визначені на множині цілих позитивних чисел та приймають позитивні дійсні значення.

1. Позначимо:

$$f(n) = O(g(n)), \quad (8)$$

якщо існує така константа $c > 0$, що $f(n) \leq c \times g(n)$ для достатньо великих n .

2. Позначимо:

$$f(n) = \Omega(g(n)), \quad (9)$$

якщо існує така константа $c > 0$, що $f(n) \geq c \times g(n)$ для достатньо великих n .

3. Позначимо:

$$f(n) = \Theta(g(n)), \quad (10)$$

якщо існує такі константи $c > 0$ $c' > 0$, що $c \times g(n) \leq f(n) \leq c' \times g(n)$ для достатньо

великих n .

Замість (10) можна записати:

$$f(n) \bigcup_{\cap} g(n). \quad (11)$$

Відношення (11) є відношенням еквівалентності. Клас еквівалентності цього відношення, що містить $f(n)$ тобто множину усіх функцій $g(n)$, таких, що $f(n) = W(g(n))$, називають швидкістю росту $f(n)$ [275].

Завдяки введеному поняттю, швидкість зростання складності алгоритму, як показано [275], можна оцінити зверху, використовуючи вираз типу “вимагає часу $O(n^3)$ ”.

Таким чином, часова складність алгоритму відображає потрібні для нього витрати часу. Функція, яка вхідній довжині n ставить у відповідність максимальний (за усіма індивідуальними завданнями довжини n) час, не буде повністю визначена до тих пір, доки не зафіксована схема кодування та не вибрано обчислювальний пристрій, що визначає час роботи.

Один і той же алгоритм A можна виконувати за різний час на різних структурах. Тому, при переході від однієї структури до іншої часто користуються поняттям складності, місткості обчислювальної системи, що характеризує зростання апаратних витрат при такому переході.

Апаратні витрати можуть задаватися у числі елементарних процесорних елементів, числі зв'язків у системі, обсязі пам'яті тощо. При цьому, використовується вираз “потрібно $O(n^3)$ елементів пам'яті”.

При аналізі складності алгоритмів оцінка у як найгіршому випадку не завжди об'єктивно несе інформацію про час, який витрачається алгоритмом, на рішення задачі, оскільки гірші випадки при рішенні конкретної задачі можуть бути рідкісними.

Тому, у роботі для кожного алгоритму теоретично визначена складність у як найгіршому випадку, а при експериментальному моделюванні роботи алгоритмів проводяться оцінки їх складності у середньому, що дозволяє об'єктивніше оцінити можливості алгоритму.

Таким чином, для проведення експериментального дослідження алгоритмів рішення задачі цілочисельного лінійного програмування (ЦЛП) з булевими змінними (БЗ) (розділ 3) та їх порівняння з відомими, будуть вибрані наступні показники:

1. $N_{оп}$ – число елементарних операцій (ЕО) складання та порівняння, що виконуються алгоритмами.
2. Об'єм пам'яті (V) для реалізації конкретного алгоритму.
3. Для оцінки якості наближених алгоритмів вибрана абсолютна δf і відносна Δf погрішності:

$$\delta f = |f(\bar{x}) - f(\bar{x}^{opt})|, \quad (12)$$

$$\Delta f = \frac{|f(\bar{x}) - f(\bar{x}^{opt})|}{f(\bar{x}^{opt})}, \quad (13)$$

де $f(\bar{x})$ – цільова функція, яка визначена на деякій множині M ; $\bar{x}$ – допустиме рішення задачі, що є наближенням; $\bar{x}^{opt}$ – оптимальне рішення задачі.

Для об'єктивного порівняння розроблених алгоритмів з відомими, а також, з урахуванням їх специфіки, додатково будемо використовувати наступні показники якості:

- $\bar{r}$ – середній ранг шляхів, відповідних оптимальному рішенню тестової задачі, який визначається числом одиниць у векторі $\vec{x}$;
- K_n – кількість неточних рішень від загального числа тестових задач, що вирішуються наближеними алгоритмами, виражені у %;
- коефіцієнт виграшу K_e у швидкодії розроблених алгоритмів щодо алгоритму, з яким здійснюється порівняння, дорівнює:

$$K_e = \frac{\overline{N_{on}}}{N_{on}^P}, \quad (14)$$

де $\overline{N_{on}}$ – число N_{on} відомим еталонним алгоритмом; N_{on}^P – число N_{on} розробленого алгоритму.

Такі показники, як зайнятий обсяг пам'яті V , δf , Δf , $\bar{r}$, K_n та K_e утворюють групу $\{\Phi_k\}$ -показників, не залежних від часу виконання алгоритму. А тимчасові

показники: число N_{on} і K_e – утворюють групу $\{\Theta_m\}$ -показників.

Використання показників функціональної групи дозволяє оцінити ефективність застосування правил відсікання $\{L_w, K_i\}$ неперспективних варіантів в узагальненій процедурі A_0 , а група тимчасових показників дозволяє порівняти розроблені алгоритми з відомими..

3.4.3 Експериментальне дослідження алгоритмів

З метою перевірки правильності функціонування алгоритмів, заснованих на принципі оптимізації по напрямку та правил розроблена програмна модель реалізації наближених алгоритмів.

Подання бінарного дерева рішень у вигляді графа $D\Delta$ обумовило необхідність збереження шляхів усього лише двох рангів: поточного r , на основі якого будуються шляхи наступного рангу та наступного $r=r+1$, що утворюється в результаті такої побудови.

На початковому етапі проводиться встановлення початкових значень змінних, які необхідні для організації збереження векторів у пам'яті ЕОМ, а також розмірність вхідних параметрів для алгоритму (m, n) . Після цього, задаються коефіцієнти $\{c_j\}$, $\{a_{ij}\}$, $\{b_{ij}\}$ у функціоналі й обмеженнях.

Наступні кроки алгоритму підпорядковані побудові шляхів першого рангу. По завершенню побудови шляхів першого рангу в утворених множинах виділяється шлях із максимальною довжиною по вагах функціоналу, значення якого присвоюється глобальному екстремуму.

Далі проводиться відсікання по обраному правилу $\{L_w\}$ безперспективних векторів поточного рангу r . На основі шляхів, що залишилися, з урахуванням принципу оптимізації по напрямку, поняття коридору формуються шляхи наступного $(r+1)$ -го рангу, що задовольняють властивості v .

Кількість сформованих векторів з однієї множини в іншу позначено як z . Значення числа z залежить від обраного правила $\{L_w\}$.

При цьому, якщо утвориться шлях із великим значенням по вагам

функціоналу, ніж розмір глобального екстремуму, то останній обновляється.

Побудова шляху такого рангу потребує виконання наступних операцій:

- збільшити розмір функціонала на вагу ребра, що з'єднує вершину j поточного рангу r та вершину p рангу $r=r+1$;
- збільшити довжину шляху msj вершині j по вагах обмежень $da(msj)$ на вагу $a1p$ ребра, що з'єднує вершину j і вершину p ;
- здійснити перевірку виконання властивості v ;
- при задоволенні властивості v зробити коригування ідентифікатора шляху msj із додаванням до нього номера p . Під ідентифікатором шляху розуміється список номерів вершин графа $D\Delta$, через які проходить цей шлях.

У пам'яті ЕОМ зберігаються шляхи двох рангів: поточного r і наступного $(r+1)$, причому, коли побудова других завершена, шляхи поточного рангу можна видалити з пам'яті.

У випадках, якщо узагальнена процедура побудувала шлях рангу $r=n$ або усі множини такого рангу порожні ($s=0$), то алгоритм закінчує роботу і виводить на друк ідентифікатор оптимального шляху та значення цільової функції для вектора, що відповідає цьому шляху.

Метою експериментальних досліджень розроблених наближених алгоритмів було виявити наступне:

- чи потребує алгоритм менше обчислень значень цільової функції у порівнянні з іншими?;
- чи потребує даний алгоритм менше машинного часу?;
- чи потребує алгоритм менше машинної пам'яті для своєї реалізації?;
- як сходиться і яку подає інформацію про обчислювальний процес?;
- на скільки алгоритм стійкий до помилок обчислень?;
- яку забезпечує точність рішень?;
- як залежить отримана точність рішення задачі за даним алгоритмом від машинного часу, що відпускається на його роботу?;
- як залежить ефективність наближених алгоритмів від засобу

сортування коефіцієнтів при функціоналі та обмеженні?

Зрозуміло, що з'ясування перерахованих питань необхідно і при аналізі результатів машинного експерименту, отриманих на тестових задачах. Вибір тестових задач проводився таким чином, щоб вони задовольняли наступним вимогам:

- у наборі задач необхідно передбачити значну кількість різноманітних тестових задач, у яких не повторюються ті самі закономірності;
- у тестових задачах повинні бути усі труднощі, що можуть зустрітися при використанні алгоритму у принципі;
- задачі повинні включати достатньо багато простих проблем, щоб можна було реалістично порівняти алгоритми у звичайних ситуаціях;
- тестові задачі повинні відбивати типові риси та особливості класу розв'язуваних задач.

Приступимо до розгляду експериментального дослідження наближених алгоритмів. У ході рішення тестових задач генерувались по рівномірному закону розподілу за допомогою датчика випадкових чисел коефіцієнти у функціоналі в діапазоні $[1 \div 100]$ та в обмеженнях у діапазоні $[1 \div 50]$.

Вибір інших діапазонів для функціоналу змінив лише його абсолютне значення, але у середньому не вплинув на параметри алгоритмів. Зміна діапазону у обмеженнях впливає лише на ранг шляху (ранг шляху – кількість вершин графа $D\Delta$, що утворюють цей шлях).

Структура дослідження кожного алгоритму полягала у наступному. Вирішувалося 100 тестових задач із заданими вхідними параметрами n обраним алгоритмом із різноманітним сортуванням коефіцієнтів при функціоналі та обмеженні. У ході рішення обчислювалися показники ефективності алгоритмів.

Результати рішення задачі наближеним алгоритмом порівнювалися з результатом її рішення точним алгоритмом, описаним у [264-265]. Отримані значення подані на графіках (рисунках 1, 2).

Як показали результати експериментального дослідження, кількісні значення обраних показників істотно залежать від рангу одержуваного рішення,

що визначає число одиниць у оптимальному рішенні.

Так, діапазон зміни r_{cp} можна умовно розбити на три умовно виділені зони:

1 зона – $r_{cp}=[0 \div n/3]$; 2 зона – $r_{cp}=[n/3 \div 2n/3]$; 3 зона – $r_{cp}=[2n/3 \div n]$ (рис. 3).

У першій зоні алгоритм знаходить рішення швидко, оскільки за рахунок зондування дуже ефективно відсікаються гілки дерева рішень, що відповідають одиничним розгалуженням.

Аналогічним чином пояснюється і швидке одержання рішення у 3 зоні, вектори у якій складаються із великого числа одиниць.

Прояв усієї експоненціальної складності алгоритму відбувається саме у другій зоні.

Для наближених алгоритмів зростання кількості ЕО зі збільшенням r_{cp} визначається числом локальних областей W , що необхідно опрацювати.

Таким чином, для об'єктивного порівняння алгоритмів необхідно вказувати, до якої зони вони належать, тобто який відсоток одиниць (або нулів) містить оптимальне рішення. Відповідно до розподілу на зони побудовані таблиці залежності точності наближених рішень для другої зони від способу сортування коефіцієнтів.

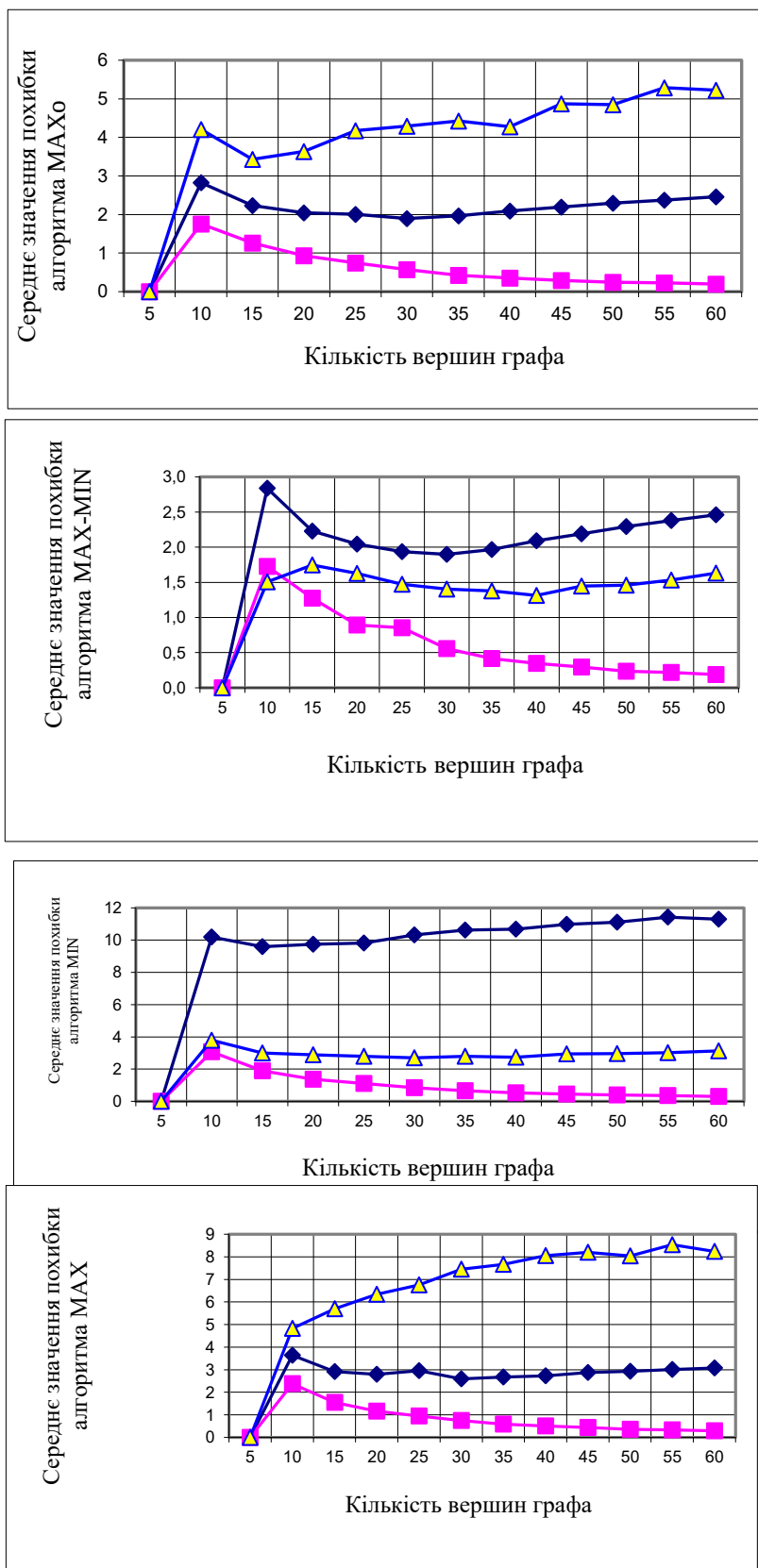


Рисунок 1 – Оцінка ефективності алгоритмів

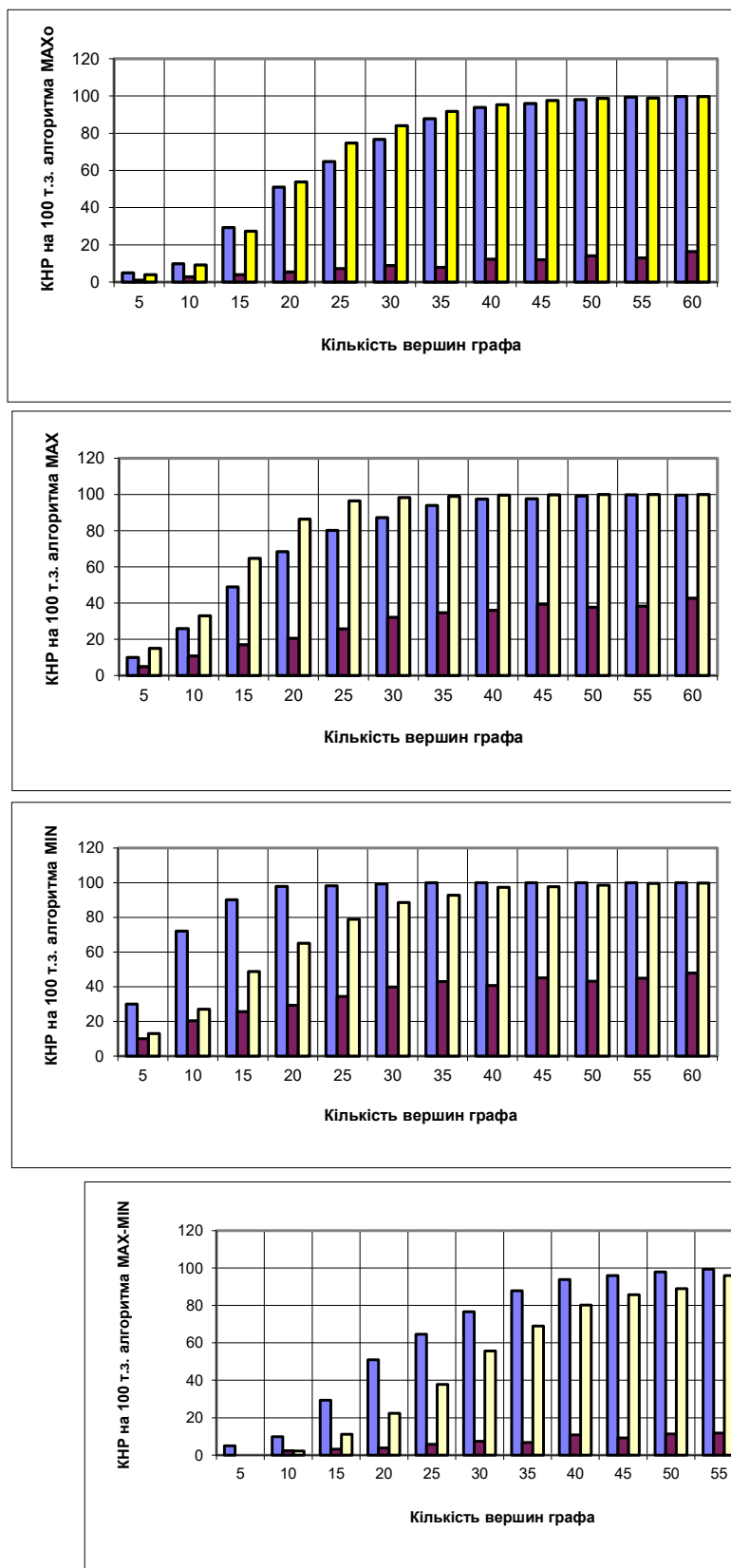


Рисунок 2 – Оцінка ефективності алгоритмів



Рисунок 3 – Кількість елементарних операцій точного алгоритму

З аналізу отриманих графіків та таблиць можна зробити висновок, що найкращою точністю володіє алгоритм MAX-MIN, похибка якого при $n > 50$ не перевищує 0,5%. Найбільше ефективним сортуванням є сортування у порядку зменшення відношень коефіцієнтів при функціоналі до відповідних коефіцієнтів у обмеженнях [277-280].

3.4.4 Результати імітаційного моделювання задачі оптимізації

На даний час методів перевірки, що гарантують повну адекватність отриманої моделі реальному процесу, немає. Існують лише методи, які дозволяють виявити неадекватність моделей у тім чи іншому ступені.

Оцінити точність моделі – це значить знайти розбіжність результатів, отриманих після реалізації цієї моделі з дійсними результатами. Зовсім ясно, що якість виконання кожного з етапів моделювання впливає на підсумкову точність моделі.

Як показано у відомій літературі, для перевірки адекватності моделі можуть використовуватися наступні методи:

- порівняння результатів моделювання на розробленій моделі з реальними результатами (тестування моделі на реальних даних);

- порівняння отриманих на розробленій моделі результатів з даними, отриманих на апробованих моделях;
- перевірка моделі на наборах параметрів, для яких результат моделюемого процесу відомий заздалегідь;
- перевірка вірогідності вихідної даних, розмірності та масштабування параметрів у рівняннях моделі;
- перевірка коректності моделі при виродженні умов моделювання;
- метод зворотного переходу.

Основними вимогами, що визначають придатність моделі до практичного застосування, є вимоги ймовірності, оперативності та контрольованості результатів.

Вимога оперативності моделювання встановлюється виходячи з розташовуваного штабом часу на збір, підготовку, введення інформації у ЕОМ, рішення задачі та аналіз результатів.

На практиці увесь цикл моделювання на етапах безпосередньої підготовки та ведення операції не повинний перевищувати 30 хв., а з обліком убудованих у модель баз даних – 1-3 хв. При підборі параметрів рішення та моделюванні декількох варіантів можливе збільшення циклу моделювання на 50-70%.

Так величина часу T_p і часу T розрахунків на практиці залежать від безлічі непередбачених умов та реально є випадковими. Оцінимо показник оперативності кожної моделі розподілу як ймовірність своєчасного одержання результатів моделювання:

$$P = 1 - e^{-\frac{T_p}{T}}. \quad (15)$$

Використовуючи середнє значення часу циклу моделювання та T_p часу (3 хв.), знайдемо значення показника P оперативності (табл. 1).

TECHNICAL AND AGRICULTURAL SCIENCES IN MODERN REALITIES: PROBLEMS, PROSPECTS AND SOLUTIONS

Таблиця 1. – Оцінка показника оперативності

Характеристики	АЛГОРИТМИ								
	Точні		Наближені						
	1	2	1	2	3	4	5	6	7
КВГ	30	30	30	30	30	30	30	30	30
СЧР (с.)	23.86	5.13	0.0	0.25	0.11	0.27	0.56	*	5.13
ПО	0.999	1	1	1	1	1	1	1	1
КВГ	50	50	50	50	50	50	50	50	50
СЧР (с.)	2100.4	7.56	0.05	1.12	0.75	1.05	2.01	*	7.26
ПО	0.082	1	1	1	1	1	1	1	1
КВГ	100	100	100	100	100	100	100	100	100
СЧР (с.)	-	9.39	0.14	3.21	2.46	3.17	7.35	*	9.3
ПО	-	0.999	1	1	1	1	1	1	1
КВГ	150	150	150	150	150	150	150	150	150
СЧР (с.)	-	61.94	0.31	14.17	9.76	13.30	47.9	*	44.90
ПО	-	0.945	1	0.99	0.999	0.999	0.976	1	0.98
КВГ	200	200	200	200	200	200	200	200	200
СЧР (с.)	-	310.81	0.73	46.04	32.40	45.53	89.3	*	45.2
ПО	-	0.439	1	0.97	0.996	0.98	0.866	1	0.98
КВГ	250	250	250	250	250	250	250	250	250
СЧР (с.)	-	1560.8	1.54	115.68	91.68	111.8	228.1	*	1.5
ПО	-	0.109	1	0.78	0.859	0.8	0.545	1	1
КВГ	300	300	300	300	300	300	300	300	300
СЧР (с.)	-	2700.3	2.1	171.15	158.3	165.7	310.5	*	2.4
ПО	-	0.064	1	0.650	0.679	0.662	0.439	1	1
КВГ	400	400	400	400	400	400	400	400	400
СЧР (с.)	-	-	2.27	222.9	193.1	211.9	532.9	*	2.5
ПО	-	-	1	0.554	0.606	0.572	0.286	1	1

Алгоритми для дослідження були обрані:

- точні алгоритми:

1 – алгоритм Балаша;

2 – багатоступінний алгоритм на основі рангового підходу;

- наближені алгоритми:

1 – алгоритми з часовою складністю $O(n)$;

2 – алгоритми з часовою складністю $O(n^2)$ при сортуванні по обмеженню;

3 – алгоритми з часовою складністю $O(n^2)$ при сортуванні по функціоналу;

4 – алгоритми з часовою складністю $O(n^2)$ при сортуванні по відношенню;

5 – алгоритми з часовою складністю $O(n^3)$;

6 – рівнобіжні алгоритми;

7 – алгоритм, який забезпечує максимальну точність обчислень при припустимих часових і ресурсних витратах.

Результати розрахунків, що представлені у таблиці 1 та рисунку 3, показують що при рівні оперативності $P \geq 0,9$ в даний час забезпечення розрахунками етапу оптимального планування можливо тільки алгоритмами з тимчасовою складністю $O(n)$ та алгоритм, що забезпечує задану точність обчислень при припустимих часових і ресурсних витратах. Застосування точних алгоритмів можливо при невеликій розмірності задачі розподілу – до 250 вершин графу, алгоритмів з часовою складністю $O(n^2)$ до 400.

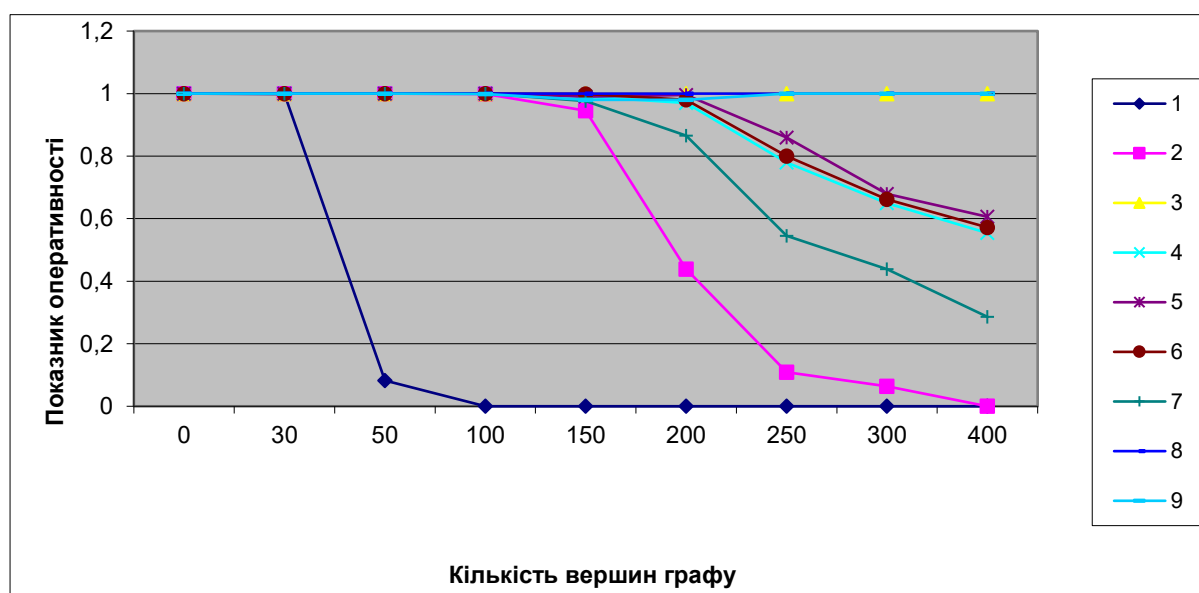


Рисунок 4 – Залежність показника оперативності від розмірності задачі

Висновки:

1. Порівняльний аналіз розроблених алгоритмів з відомими за вибраними показниками ефективності показав, що їх часова складність істотним чином залежить від рангу оптимального рішення, який може належати одній з трьох умовно виділених зон. Тому, об'єктивне порівняння алгоритмів можливо лише

для рішень, що належать одній і тій же зоні.

2. З порівняння рішень за зонами видно, що найбільший виграш розроблені алгоритми дають у другій зоні, де число припустимих рішень експоненціально. Це є важливою перевагою у порівнянні з відомими.

3. Дослідження погрішності наближених алгоритмів показало, що із збільшенням розмірності вирішуваної тестової задачі, вона стабілізується і для різних стратегій відсікання лежить у межах від 1 до 10%.

4. Властивість стабілізації погрішності дозволила побудувати ефективні точні багато етапні алгоритми рішення ЦЛП з БЗ. Якщо точність попереднього етапу лежить у інтервалі до 2%, то часова складність таких алгоритмів у середньому з довірчою вірогідністю 0,95 не перевищує $O(c^{mn})$, де $c=\text{const}$ при $n < 100$.