УДК 519.7, 519.8
DOI: 10.20535/SRIT.2308-8893.2016.2.08

ОПТИМІЗАЦІЯ ПЛАНУВАННЯ РОЗПОДІЛУ ЗАВДАНЬ І ТРАНСПОРТУВАННЯ ПАКЕТІВ ДАНИХ У РОЗПОДІЛЕНІЙ ОБЧИСЛЮВАЛЬНІЙ МЕРЕЖІ

Р.П. КРАСНЮК, Г.Г. ЦЕГЕЛИК

Розглянуто задачі об'ємно-календарного планування розподілу завдань і транспортування пакетів даних у розподіленій обчислювальній мережі. Виконано математичну постановку, сформульовано ефективні обчислювальні алгоритми та отримано наближені розв'язки цих задач. Показано ефективність запропонованих обчислювальних алгоритмів щодо побудови наближеного розв'язку одно- та багатокритеріальних задач оптимізації на основі порівняльного аналізу застосування цих алгоритмів на тестових прикладах. Зроблено висновок про обчислювальну ефективність запропонованих алгоритмів зі збільшенням розмірностей задач.

ВСТУП

Інтеграція інформаційних та обчислювальних ресурсів в єдине середовище та організація ефективного доступу до них є одним з основних напрямів розвитку сучасних інформаційних технологій. Першочерговою стає проблема ефективного використання обчислювальних ресурсів кожного вузла мережі для вирішення складних наукових, виробничих і технологічних завдань. Тому одним з актуальних завдань сьогодення є ефективне керування обчислювальними ресурсами у розподіленому середовищі. Зростання кількості ресурсних центрів, що становлять розподілену інфраструктуру, за відсутності або низької ефективності підсистеми планування, яка забезпечує керування потоком задач, не тільки знижує продуктивність використання усієї розподіленої інфраструктури, але й може зробити беззмістовним її створення. Крім того, розподілені системи характеризуються динамічним розвитком, що унеможливлює вирішення завдань продуктивного керування ресурсами у статичному середовищі без використання методів та підходів, які розроблялися для динамічних задач.

Упровадження більш дійових алгоритмів керування розподіленим середовищем у безпосередньо вже створеній розподіленій інфраструктурі ускладнено і зумовлено додатковими витратами та недостатнім завантаженням ресурсних центрів, а через масштабність ресурсного середовища взагалі неможливе. Тому актуальним завданням є створення систем моделювання розподіленої комп'ютерної інфраструктури й ефективних моделей та методів оптимізації, які дозволяють адекватно оцінити поведінку мережі за змінюваних умов і на підставі цього покращити стратегії керування потоками завдань.

Мета роботи — дослідити проблему планування розподілу завдань і транспортування пакетів даних у розподіленій обчислювальній мережі. Для досягнення поставленої мети вирішено такі завдання:

© *Р.П. Краснюк, Г.Г. Цегелик*, 2016

– сформульовано та досліджено математичні моделі оптимального планування розподілу завдань і транспортування пакетів даних;

– розроблено ефективні обчислювальні алгоритми пошуку наближеного розв'язку одно- та двокритеріальних оптимізаційних задач;

– досліджено ефективність сформульованих алгоритмів за результатами числових експериментів на тестових прикладах та порівняно з точними розв'язками.

ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Актуальність досліджень оптимізації розподілу завдань у комп'ютерних мережах зумовила появу значної кількості публікацій, присвячених цій тематиці. Зокрема, у праці [1] побудовано та досліджено математичні моделі розподілу потоків обмежених ресурсів у енергетичних системах. Розроблено методи розв'язання задач розподілу потоків, що базуються на поєднанні екстремальних підходів до розрахунку мереж. Оптимізація упорядкування передавання повідомлень у вузлах комп'ютерних мереж з урахуванням динаміки трафіку досліджено у праці [2]. Для розв'язання задачі запропоновано критерій максимальної тривалості доставлення пакета, що мінімізується. З використанням імітаційної моделі вузла мережі проведено числові експерименти оптимізації упорядкування передавання повідомлень у вузлах мережі для різної кількості черг. Числові методи отримання парето-оптимальних точок, що ґрунтуються на зведенні багатокритеріальних задач до «скаляризованих» задач оптимізації зі спеціальними цільовими функціями, розглянуто у праці [3]. Сформовано алгоритм, згідно з яким вихідна задача зі знаходження ефективної точки зводиться до послідовного розв'язання задач квадратичного програмування.

У фундаментальній праці [4] проаналізовано ефективні архітектури систем розподілених обчислень, наведено моделі та технології їх функціонування. Моделюванню та оптимізації доступу до інформаційних файлів баз даних для одно- та багатопроцесорних систем присвячено працю [5], у якій розглянуто групу нових та ефективних математичних моделей оптимізації розподілу завдань у системах розподілених обчислень. У монографії [6] висвітлюється питання моделювання бізнес-процесів за допомогою розподілених систем обчислень. Досліджено методи прийняття рішень щодо оптимізації за заданим критерієм.

ЗАДАЧА КАЛЕНДАРНОГО ПЛАНУВАННЯ РОЗПОДІЛУ ЗАВДАНЬ В ОБЧИСЛЮВАЛЬНІЙ МЕРЕЖІ

Розглянемо обчислювальну систему, що складається з деякої множини вузлів, у якій розподіляються пакети завдань, які необхідно виконати за період планування. Відома максимальна кількість завдань, яка може бути виконана у вузлах на кожному кроці планування; мінімально допустима кількість завдань, яка має бути виконана у вузлах за кожним пакетом; максимальний обсяг завдань, який може бути виконаний у вузлах на кожному кроці планування; мінімально допустима і максимально необхідна кількість завдань, що повинна бути виконана у вузлах для кожного пакета. Необхідно визначити на заданий період планування програму виконання пакетів завдань, що забезпечує ефективне функціонування обчислювальної мережі таку, яка задовольняє обмеження можливих обсягів роботи.

Нехай I — множина вузлів обчислювальної мережі, J — множина пакетів, T — множина кроків планування. Позначимо через A_t максимальну сумарну кількість завдань, які можуть бути виконані в обчислювальній мережі за всіма пакетами на кроці планування t; B_j — мінімально допустиму кількість завдань, які мають бути виконані у вузлі за пакетом j протягом усього часу планування; C_{it} — максимальну кількість завдань, які можуть бути виконані у вузлі i на кроці планування t; D^-_{ij}, D^+_{ij} — мінімально допустиму і максимально необхідну кількість завдань, які повинні бути виконані у вузлі i для пакета j, $i \in I$, $j \in J$, $t \in T$.

Позначимо через x_{ijt} цілочислову кількість завдань, які виконуються на кроці t для пакета j у вузлі i, $i \in I$, $j \in J$, $t \in T$. Тоді математична модель календарного планування складається з такої системи обмежень:

$$\sum_{i\in I}\sum_{j\in J} x_{ijt} \le A_t, \quad t \in T, \tag{1}$$

(загальна кількість задач на кожному кроці планування не повинна перевищувати максимально допустиму кількість);

$$\sum_{i\in I}\sum_{j\in J} x_{ijt} \ge B_j, \quad j \in J, \tag{2}$$

(загальна кількість задач за кожним із пакетів має бути не меншою за мінімальну допустиму кількість);

$$\sum_{j\in j} x_{ijt} \le C_{it}, \quad i \in I, \quad t \in T, \tag{3}$$

(загальна кількість задач для кожного вузла на кожному кроці планування не може перевищувати максимально допустиму кількість);

$$D^-_{ij} \le \sum_{t\in T} x_{ijt} \le D^+_{ij}, \quad i \in I, \quad j \in J, \tag{4}$$

(обмеження мінімально допустимої та максимально необхідної кількості завдань у кожному вузлі за кожним пакетом);

$$0 \le x_{ijt}, \ i \in I, \ \ j \in J, \ \ t \in T, \tag{5}$$

(природні обмеження на змінні).

Критерії оптимальності можуть залежати від різних показників необхідної програми виконання завдань, наприклад, від кількості виконуваних завдань за пакетами та кількості завдань у вузлах обчислювальної мережі.

Тоді досліджувана задача календарного планування полягатиме у визначенні такої програми виконання завдань, для якої виконуються обмеження (1)–(5) і набувають екстремальних значень критерії:

$$f_j\left(\sum_{i\in I}\sum_{j\in J}x_{ijt}, B_j\right), \; j\in J, \quad \Theta_{it}\left(\sum_{j\in J}x_{ijt}, C_{it}\right), \quad i\in I, \quad t\in T,$$

що визначають умови ефективного виконання завдань за пакетами та ефективності функціонування вузлів обчислювальної мережі.

ОПТИМІЗАЦІЙНА ЗАДАЧА ПЛАНУВАННЯ ТРАНСПОРТУВАННЯ ПАКЕТІВ ЗАВДАНЬ

Розглядається складна розподілена обчислювальна система, основними елементами якої є:

- вузли, які генерують пакети завдань;
- вузли, які ретранслюють пакети завдань;
- комунікаційні зв'язки між вузлами обчислювальної системи.

Схема функціонування процесу транспортування пакетів завдань включає в себе формування пакетів у вузлах, що обмежують обсяги ресурсів пам'яті; транспортування пакетів за різними комунікаційними зв'язками, які обмежують пропускну здатність та проходження пакетів завдань через ретрансляційні вузли, кожен з яких, у свою чергу, має обмежену «потужність» (пропускну здатність) ретрансляції.

Актуальною для подібних систем є така задача планування: за заданих обмежень обсягів пакетів, які формуються, обмежень пропускної здатності комунікаційних ліній зв'язку та відомих «потужностей» ретрансляційних вузлів необхідно на заданий період планування за «стандартних» умов визначити максимально можливі обсяги транспортування пакетів у наявній системі за мінімальних видатків на її обслуговування. У постановці цієї задачі передбачаються «стандартними» умови як умови безаварійної роботи, за яких довільно задані для елементів системи характеристики можуть бути досягнуті.

Нехай $i=\overline{1,m}$ — номери ретранслюючих вузлів; $j=\overline{1,q_{ik}}$ — номери комунікаційних зв'язків, які сполучають вузол з номером i з вузлом з номером k; $i=\overline{1,m}$, $k=\overline{1,m}$; W_{jik} — максимально можлива пропускна здатність лінії зв'язку з номером j, що сполучає вузол i з вузлом k; $j=\overline{1,q_{ik}}$, $i=\overline{1,m}$, $k=\overline{1,m}$; G_{jik} — максимальна «потужність» ретрансляції вузла i, що обслуговує j-у лінію зв'язку, яка сполучає ретранслюючі вузли з номерами i та k; $j=\overline{1,q_{ik}}$, $i=\overline{1,m}$, $k=\overline{1,m}$; Q_i — пропускна «потужність» вузла з номером i; $i=\overline{1,m}$; c_{ijk} — витрати на траспортування одиниці пакета даних у вузлі i, що обслуговує j-у комунікаційну лінію від i-го до k-го вузла; $j=\overline{1,q_{ik}}$, $i=\overline{1,m}$, $k=\overline{1,m}$; V_i — обсяг пакетів даних, який може надходити у вузол з номером i, $V_i \ge 0$, $i=\overline{1,m}$. У випадку, якщо вузол здійснює тільки ретрансляцію пакетів, то $V_i = 0$, $i=\overline{1,m}$. Будемо вважати, що пропускна здатність комунікаційних ліній зв'язку та «потужність» ретранслюючих вузлів вимірюються в одних і тих самих одиницях.

Позначимо через x_{ijk} обсяг даних, який буде переданий по комунікаційній лінії зв'язку з номером j від вузла i до вузла k, $j=\overline{1,q_{ik}}$, $i=\overline{1,m}$, $k=\overline{1,m}$. Тоді можна сформувати обмеження математичної моделі:

$$\sum_{k=1}^{m}\sum_{j=1}^{q_{ik}} x_{jik} \le Q_i\,,\ \ i=\overline{1,m} \tag{6}$$

(обсяг даних, що передається від вузла з номером i, не повинен перевищувати його «потужності»);

$$\sum_{k=1}^{m}\sum_{j=1}^{q_{ik}} x_{jik} = V_i + \sum_{k=1}^{m}\sum_{j=1}^{q_{ik}} x_{jik}\,,\ i=\overline{1,m} \tag{7}$$

(рівняння балансу — обсяг даних, який передається від ретрансляційного вузла з номером i, дорівнює обсягу даних, сформованого у вузлі i, і той обсяг даних, що надходить транзитом у вузол з номером i);

$$x_{jik} \le \min(G_{jik}, W_{jik})\,,\ j=\overline{1,q_{ik}},\ i=\overline{1,m},\ k=\overline{1,m} \tag{8}$$

(обсяг даних, який передається комунікаційною лінією зв'язку j, не повинен перевищувати максимальну «потужність» вузла, що обслуговує цю лінію зв'язку, і пропускну здатність j-ї комунікаційної лінії зв'язку, що сполучає i-й та k-й вузли);

$$x_{jik} \ge 0\,,\ j=\overline{1,q_{ik}}\,,\ i=\overline{1,m}\,,\ k=\overline{1,m} \tag{9}$$

(природні умови на змінні).

Наведена загальна математична модель планування транспортування пакетів даних (6)–(9) становить систему лінійних обмежень транспортного типу. Критерії оптимальності задачі планування можна сформулювати як

$$F(X) = \min_{i=\overline{1,m}}\left(\sum_{k=1}^{m}\sum_{j=1}^{q_{ik}} x_{jik}\right) \to \max\,, \tag{10}$$

(сумарний обсяг даних, що передається у розподіленій обчислювальній мережі, має бути максимально можливим);

$$Q(X) = \sum_{i=1}^{m}\sum_{k=1}^{m}\sum_{j=1}^{q_{ik}} c_{jik} x_{jik} \to \min \tag{11}$$

(сумарні видатки на транспортування пакетів даних мають бути мінімальними).

Задача (6)–(11) є двокритеріальною задачею оптимального планування за критеріями максимізації обсягу транспортування пакетів даних та мінімізації видатків на їх транспортування.

ФОРМУВАННЯ ОБЧИСЛЮВАЛЬНИХ АЛГОРИТМІВ

Застосування точних методів до побудови розв'язків задач оптимізації може ускладнюватися для розв'язання задач великої розмірності через значні часові витрати. Загалом задовільні для практичного використання результати можна отримати, застосувавши евристичні алгоритми, такі як «жадібні» та генетичні алгоритми, що дають наближений розв'язок поставленої задачі [7, 8].

Відповідно до «жадібного» методу розв'язок задачі розмірності n шукаємо у вигляді вектора $X=[x_1,...,x_n]$. Початковий розв'язок може бути будь-яким, наприклад нульовим: $X=[0,...,0]$. На кожному кроці вибирається x_i з умов задач: зокрема для задачі (1)–(5) — це умова ефективного виконання завдань за пакетами, для задачі (6)–(9) — умови (10)–(11). Крім того, значення x_i має бути таким, щоб розв'язок задачі задовольняв додаткові обмеження, які накладаються при формулюванні оптимізаційної задачі.

Точність розв'язку задач з використанням «жадібного» алгоритму оцінюється за формулою

$$W=\frac{Z_a\cdot 100}{Z_g}, \tag{12}$$

де Z_a — екстремальне значення цільової функції, знайдене з використанням наближеного методу; Z_g — екстремальне значення цільової функції, отримане точним методом.

Оцінку точності (табл. 1) отримано для оптимізаційних задач (1)–(5) і (6)–(11), розв'язок яких знайдено з використанням «жадібного» алгоритму. Оцінка розрахована як середнє значення за ста розв'язками поставлених задач оптимізації, вхідні дані для яких задавалися випадковим чином.

Таблиця 1. Оцінка точності використання «жадібного» алгоритму в задачах планування розподілу завдань і транспортування пакетів даних

Задача	Середній коефіцієнт точності W,%	Кількість задач, розв'язків для яких не знайдено
(1)–(5)	87,0	12
(6)–(11)	71,5	15

Іншим ефективним методом побудови наближеного розв'язку є використання генетичних алгоритмів [8]. Вхідними даними для реалізованих у роботі генетичних алгоритмів є два початкові розв'язки, батьківські хромосоми, $X=[x_1,...,x_n]$ та $Y=[y_1,...,y_n]$; X і Y можуть визначатися випадково чи бути результатом попереднього розв'язку задачі іншим оптимізаційним методом, наприклад, з використанням «жадібних» алгоритмів. Необхідна умова для X і Y — задоволення додаткових обмежень, які накладаються при формулюванні оптимізаційної задачі.

Після задання X і Y відбувається схрещування, тобто взаємний обмін елементами (генами) векторів X і Y, що містяться на однакових позиціях.

У реалізованому в роботі програмному додатку обмінюються два елементи. Елементи для обміну вибираються випадково. Наприклад, за розмірності задачі, яка дорівнює трьом, із батьківських хромосом $X=[x_1, x_2, x_3]$ і $Y=[y_1, y_2, y_3]$ після визначення випадковим чином генів, що підлягають взаємному обміну, зокрема першого і третього, проявляють дві дочірні хромосоми: $YX1=[y_1, x_2, y_3]$ і $YX2=[x_1, y_2, x_3]$.

Після схрещування відбувається мутація хромосом, тобто заміна значення генів, які обираються випадково. У реалізованих у роботі генетичних алгоритмах до мутації схильні дочірні і батьківські хромосоми і змінюється значення єдиного гена. Імовірність того, що мутація відбудеться, у реалізованих алгоритмах можна змінювати, задаючи значення відповідного коефіцієнта. Для розглянутого прикладу, після обрання випадковим чином генів, які можуть бути змінені в усіх хромосомах, наприклад, перший ген у X, другий ген в $YX1$, третій ген в $YX2$, а Y не піддалася мутації, отримуємо такий набір хромосом:

батьківські: $X=[\overline{x_1}, x_2, x_3]$ та $Y=[y_1, y_2, y_3]$;

дочірні: $YX1=[y_1, \overline{x_2}, y_3]$ та $YX2=[x_1, y_2, \overline{x_3}]$,

коли верхні риски позначають логічне заперечення. Після мутації відбувається селекція, тобто з чотирьох розв'язків X, Y, $YX1$, $YX2$ залишаються тільки два, для яких значення *fitness*-функції є максимальним. *Fitness*-функція може задаватися довільно [8].

Після селекції генетичний алгоритм запускається з початку: два отримані за селекції розв'язки використовуються як батьківські хромосоми, оскільки генетичні алгоритми належать до типу рекурсивних. Процес може повторюватися задану кількість разів (поколінь) або зупинятися з настанням певної події, наприклад, отримання розв'язку, що забезпечує додаткові обмеження оптимального значення цільової функції.

Для числового експерименту обрано побудову наближеного розв'язку задачі (6)– (11) з використанням генетичного алгоритму, коли при схрещуванні обмінюються два гени; мутації застосовувалися як до дочірніх, так і до батьківських хромосом; за випадкової мутації відбувалася зміна одного гена і ймовірність мутації хромосом обрано рівною 0,5. Для порівняння *fitness*-функція $F(X)$ визначалася чотирма залежностями: 1) $F(X)=$ $=\dfrac{1000^{f_1(X)}}{f_2(X)}$; 2) $F(X)=\dfrac{e^{f_1(X)}}{f_2(X)}$; 3) $F(X)=\dfrac{f_1(X)}{f_2(X)}$; 4) $F(X)=\dfrac{f_1(X)}{e^{f_2(X)}}$, коли X — розв'язок оптимізаційної задачі («хромосоми»); $f_1(X)$ і $f_2(X)$ — цільові функції у формулах (10), (11).

Умовою припинення обчислень у реалізації генетичного алгоритму стало отримання в наборі хромосом, що пройшли селекцію, при цьому кількість поколінь має бути більшою за тисячу. Із досягненням кількості поколінь десять тисяч та за відсутності в наборі хромосом, що пройшли селекцію, алгоритм припиняє виконання з видачею повідомлення про відсутність розв'язку. Для чотирьох, наведених вище варіантів *fitness*-функції $F(X)$, виконано оцінку точності розрахунку за формулою (12) (табл. 2). Оцінку розраховано як середнє значення тисячі розв'язків поставленої задачі оптимізації, вхідні дані для якої задавалися випадково.

Таблиця 2. Оцінка точності використання генетичного алгоритму в задачі (6)–(11) оптимізації транспортування пакетів даних у розподіленій обчислювальній мережі

Fitness-функція $F(x)$	Середній коефіцієнт точності $W,\%$	Кількість задач, розв'язків для яких не знайдено за десятьма тисячами поколінь
1	76,1	5
2	77,3	22
3	68,6	10
4	82,7	137

Ще одним ефективним методом побудови наближеного розв'язку двокритеріальної оптимізаційної задачі є застосування методу мінімального відхилення від ідеальної точки. Цей метод є різновидом загального методу згортання цільової функції [9], але істотно відрізняється від нього за характером інтерпритації отриманого результату. Основна ідея методу полягає в попередньому відшуканні ідеальної точки задачі багатокритеріальної оптимізації і розв'язанні деякої нової задачі однокритеріальної оптимізації. У цьому випадку як нову задачу оптимізації розглядають задачу мінімізації відхилення від знайденої ідеальної точки в деякій визначеній метриці [9]. Отриманий розв'язок беруть за остаточний розв'язок вихідної задачі багатокритеріальної оптимізації. Ідеальною точкою задачі (6)–(11) багатокритеріальної оптимізації є сукупність значень $x_{ijk} = x_{ijk}^*$, для якої виконується умова

$$f_l(x_{ijk}^*) = \min_{x_{ijk}} f_l(x_{ijk}) = f_l^*,\ l = 1,2\ ;\ i = 1,2,...,m\ ;\ j = 1,2,...,q_{ik}\ ;\ k = 1,2,...,m\ .$$

Як метрику, що використовується для розрахунку відхилення від ідеальної точки, вибирають метрику

$$\rho_s(f_l(x_{ijk}), f_l^*) = \left[\sum_{l=1}^{2} \left| f_l(x_{ijk}) - f_l^* \right|^s \right]^{\frac{1}{s}},\ s = 1,2,...$$

Коли $s = 1$, отримаємо

$$\rho_1(f_l(x_{ijk}), f_l^*) = \sum_{l=1}^{2} \left| f_l(x_{ijk}) - f_l^* \right| .$$

За умови $s = 2$, отримаємо евклідову метрику:

$$\rho_2(f_l(x_{ijk}), f_l^*) = \left[\sum_{l=1}^{2} \left| f_l(x_{ijk}) - f_l^* \right|^2 \right]^{\frac{1}{2}}, \qquad (13)$$

а за умови $s = \infty$ — рівномірну метрику

$$\rho_\infty(f_l(x_{ijk}), f_l^*) = \max_l \left| f_l(f_l(x_{ijk}) - f_l^* \right|,\ l = 1,2 .$$

Як наслідок розв'язок багатокритеріальної задачі можна звести до розв'язку однокритеріальної задачі оптимізації

$$\rho_s(f_l(x_{ijk}), f_l^*) \to \min_{x_{ijk}} \tag{14}$$

за умов (6)–(9).

Алгоритм методу мінімального відхилення від ідеальної точки, що орієнтований на розв'язок задачі багатокритеріальної оптимізації у постановці (14), (6)–(9), має ітерактивний характер і полягає у виконанні таких кроків:

Крок 1. *Попереднє знаходження ідеальної точки.* Одним з можливих методів (наприклад, «жадібним» чи генетичним алгоритмом) розв'язати сукупність однокритеріальних задач оптимізації (6)–(11). Будуть розраховані оптимальні значення f_l^* для кожної цільової функції.

Крок 2. *Формування нової цільової функції.* Як нову функцію цілі розглядати, наприклад, вираз (13).

Крок 3. *Розв'язок нової оптимізаційної задачі.* Одним з методів (як і раніше, «жадібним» чи генетичним) побудувати розв'язок нової однокритеріальної задачі оптимізації (14), (6)–(9), тобто знайти сукупність значень $x_{ijk} = x_{ijk}^*$, для яких обчислити значення функції цілі $f_l(x_{ijk}^*) = f_l^{**}$. Скориставшись формулою для метрики, аналогічною до тієї, яка використовувалася для формування нової цільової функції на кроці 2, сформулюємо умову припинення обчислень. Наприклад, для випадку $s = 2$ отримаємо

$$\varphi(f_l^*, f_l^{**}) = \left[\sum_{l=1}^{2} \left|f_l^* - f_l^{**}\right|^2\right]^{\frac{1}{2}} < \varepsilon, \tag{15}$$

де ε — похибка обчислень. Якщо умова (15) виконується, то результатом розв'язання багатокритеріальної задачі є сукупність значень $x_{ijk} = x_{ijk}^*$, а оптимальні значення цільових функцій становлять $f_l(x_{ijk}^*) = f_l^{**}$. У випадку порушення умови (15) переприсвоюються $f_l^* = f_l^{**}$, $l = 1, 2$ і відбувається повернення до кроку 2.

Для трьох варіантів метрик, наведених вище, виконано оцінку середньої кількості кроків розрахунку розв'язку задачі (14), (6)–(9) за ста варіантами вхідних даних, які задавалися випадково (табл. 3).

Таблиця 3. Оцінка кількості кроків у методі відхилення від ідеальної точки за побудови розв'язку оптимізаційної задачі (6)–(11)

Степінь метрики	**Середнє значення кількості кроків алгоритму за умов $\varepsilon = 10^{-5}$, $m = 10$**
$s = 1$	50
$s = 2$	27
$s = \infty$	34

Для побудови розв'язків однокритеріальних оптимізаційних задач використовувалися модифіковані «жадібні» та генетичні алгоритми. Як показав аналіз отриманих результатів, ефективним варіантом є використання «жадібного» алгоритму внаслідок меншої кількості ітерацій для досягнення необхідної точності наближеного розв'язку. У середньому кількість ітерацій з використанням «жадібного» алгоритму складала 30, коли розрахунок генетичним алгоритмом відповідно становив 42 ітерації за умов $\varepsilon = 10^{-5}$, $m = 10$ та вибору метрики (13).

ВИСНОВКИ

Переважна більшість практичних задач оптимального керування у розподілених обчислювальних мережах не мають аналітичного розв'язку у формі розрахункових формул. Тому стає актуальним формування, дослідження та вибір обчислювальних методів для практичного розв'язування задач багатокритеріальної оптимізації. На вибір методів і засобів впливає характер математичної моделі та математичні властивості множини допустимих альтернатив. У першому випадку клас задач, до якого належить розглядувана математична модель, як правило, визначає вибір методу та алгоритму розв'язання відповідної задачі оптимізації. У другому випадку така характеристика множини допустимих альтернатив, як, наприклад, розмірність вихідних даних, істотно впливає на можливість отримання точного чи наближеного розв'язку.

Виходячи з аналізу досліджених у роботі математичних моделей оптимізації планування розподілу завдань і транспортування пакетів даних у розподіленій обчислювальній мережі та розуміння великої розмірності вихідних даних, що випливає з розгляду архітектури розподіленої мережі, виконано побудову, аналіз, адаптацію та практичну перевірку наближених методів розв'язання задач оптимізації: «жадібних», генетичних та відхилення від ідеальної точки алгоритмів. Застосування цих алгоритмів показало ефективність побудови наближеного розв'язку. Наведено відповідні алгоритми та виконано порівняльний аналіз ефективності застосування методів у тестових прикладах, вихідні дані для яких задавались випадково, оцінено обчислювальну точність алгоритмів відносно точних розв'язків задач. У випадку розрахунків за алгоритмом методу відхилення від ідеальної точки проведено низку числових експериментів за умови вибору метрики, що дозволило зробити висновок про обчислювальну ефективність евклідової метрики зі збільшенням розмірності задач.

Проведений аналіз дозволяє використовувати адаптовані алгоритми як самостійно для отримання розв'язків із задовільною для практичного застосування точністю результатів, так і як основу для розроблення комплексних алгоритмів у комбінації з іншими методами.

Сформульовані та досліджені математичні моделі оптимізації розподілу ресурсів та адаптовані до цих моделей наближені методи розв'язання задач оптимізації можуть бути покладені в основу створення програмних комплексів керування розподіленою комп'ютерною інфраструктурою, що і є предметом подальших досліджень.

ЛІТЕРАТУРА

1. *Кірік О.Є.* Розподіл ресурсів у розподільчих системах з оптимальним перерозподілом навантаження постачальників продукту / О.Є. Кірік // Системні дослідження та інформаційні технології. — 2013. — № 4. — С. 38–51.
2. *Пустовойтов П.Е.* Оптимизация порядка передачи сообщений в узлах компьютерных сетей с учетом динамики трафика / П.Е. Пустовойтов, Л.Г. Раскин // Системні дослідження та інформаційні технології. — 2013. — № 3. — С. 53–57.
3. *Александрова В.М.* Деякі методи знаходження ефективних точок багатокритеріальної задачі оптимізації / В.М. Александрова, Л.О. Соболенко // Системні дослідження та інформаційні технології. — 2014. — № 4. — С. 100–110.
4. *Куссуль Н.Н.* Grid-системы для задач иследования Земли. Архитектура, модели и технологи / Н.Н. Куссуль, А.Ю. Шелестов . — К.: Наук. думка. — 2008. — 452 с.
5. *Цегелик Г.Г.* Моделювання та оптимізація доступу до інформації файлів баз даних для однопроцесорних та багатопроцесорних систем / Г.Г. Цегелик. — Львів: ЛНУ імені Івана Франка. — 2010. — 192 с.
6. *Томашевський О.М.* Інформаційні технології та моделювання бізнес-процесів: навч. посіб. / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук. — К.: Вид-во «Центр учбової літератури». — 2012. — 296 с.
7. *Ахо А.В.* Структуры даных и алгоритмы / А.В. Ахо, Д.Э. Хонкрофт, Д.Д. Ульман. — М.: Издат. дом «Вильямс», 2003. — 426 с.
8. *Сигал И.Х.* Введение в прикладное дискретное программирование: модели и вычислительные методы / И.Х. Сигал, А.П. Иванова. — М.: Физматлит, 2002. — 320 с.
9. *Новиков Ф.А.* Дискретная математика для программистов / Ф.А. Новиков. — СПб.: Питер, 2011. — 526 с.

Надійшла 20.09.15