DOI 10.15276/imms.v13.no1-2.162
УДК 519.6:004.93

Informatics and Mathematical Methods in Simulation
Vol.13 (2023), No. 1-2, pp.162-172

КОМБІНОВАНІ АЛГОРИТМИ ВИЗНАЧЕННЯ ПОЧАТКОВОГО РІШЕННЯ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ

Б. І. Юхименко[1], Н.П. Волкова[2], Ю.Ю. Козіна[3]

Національний університет «Одеська політехніка»,
Одеса, 65044, пр. Шевченко, 1,
e-mails: biruteyu@gmail.com[1], volkova.n.p@op.edu.ua[2], yuliyakc21@gmail.com[3]

Проблему вирішення завдань дискретної оптимізації повністю не вирішено. Безліч публікацій, наукових розробок, алгоритмів та програмних продуктів не дає можливості перевести математичний апарат розв'язання задач дискретної оптимізації до класу P складності обчислень. Усі аналітичні та комбінаторні алгоритми вирішення задач лінійної та нелінійної оптимізації є NP складними. У зв'язку з цим будь-які розробки, щодо підвищення ефективності роботи алгоритмів, залишаються затребуваними та актуальними. У цій роботі запропоновано використовувати детерміновані та ймовірнісні прийоми формування пріоритетної черги компонент вектора рішень з метою присвоєння їм позитивних значень. Послідовне формування варіанта рішень можна використовувати як отримання наближеного рішення чи рекордного значення цільової функції, що у точних алгоритмах як вихідне рішення, підлягає поліпшенню. У роботі наведено способи формування пріоритетної черги конкретизації компонентів вектора рішень. Основу детермінованих методів становить ідея жадібного алгоритму. Місце розташування у черзі визначається величиною відповідної компоненти вектора вартості. Облік величини нев'язки у системі обмежень підвищує пріоритетність компоненти. За таким засобом модифікується другий детермінований спосіб. Імовірнісна оцінка пріоритетів ґрунтується на ідеях алгоритмів мурашиної колонії та імітації відпалу. Розмір ймовірності визначає значимість компоненти – претендента на позитивне значення. Наведено числовий приклад невеликої розмірності задачі про ранець, що демонструє отримання наближеного розв’язання.

Ключові слова: дискретна оптимізація, рекорд, пріоритетна черга, метод гілок та меж.

Вступ. Дискретна оптимізація використовується у різних галузях діяльності. Завдання планування, транспортування вантажів, логістики та багато інших приводяться до моделей цілочисельного, частково цілочисельного, лінійного та нелінійного математичного програмування, до моделей з булевими змінними. Методи реалізації цих моделей часто не відповідають сучасним вимогам. Вирішення багатьох прикладних завдань, що мають велику розмірність, не призводить до оптимального результату. Вони відстають в оперативності і у деяких випадках не дають очікуваного рішення. З математичних позицій методи дискретної оптимізації належать до класу NP складності. Тому, завдання їх модифікації, залучення комбінаторних прийомів, що збільшують швидкість збіжності, є актуальним. Мотивація у сучасній дискретній оптимізації – як перейти з класу NP складності до класу P.

Останнім часом, для вирішення багатьох комбінаторних завдань, задач цілочисельного лінійного програмування використовуються прийоми, що імітують поведінку не інтелектуальних істот. Популярними стали алгоритми мурашиної колонії [1]. Вони ефективно використовуються для вирішення задачі про комівояжера, розподілу виробничих замовлень, розміщення продуктивних сил та ін. Характерні моменти поведінки мурах як масовість, інформативність, прямий та зворотний зв'язок, дозволяє розробляти наближені ймовірні алгоритми. Інтерпретація та формалізація поведінки мурах є основою створення алгоритмічних прийомів, що залежать від низки параметрів. Число мурах-агентів виконують однакову роботу – паралельне рішення одного і того ж завдання, рівень інформативності – кількість феромонів, переданих особинам колонії як повідомлень, зміна ситуації – випаровування феромонів – визначає величину ймовірності прийнятого рішення. Для кожного конкретного випадку, конкретної задачі та її структурних особливостей є індивідуальний вибір значень параметрів. Це не є детермінований процес. Неможливо теоретично довести ні кінцівку, ні точність одержуваного рішення. Це недолік таких алгоритмів, які імітують поведінку живих істот чи процесів які є у природі.

Поліпшення ефективності роботи перебірних – комбінаторних алгоритмів може вестись різними шляхами. Оскільки методика розв'язання оптимізаційних завдань передбачає отримання вихідного рішення, питання їх визначення доречні й у дискретній оптимізації. Чим точніше вихідне рішення, тим швидше отримуємо оптимальне рішення під час використання точних алгоритмів. Крім того, вони можуть використовуватись як деякі наближені рішення. При вирішенні практичних завдань, що мають великі розмірності та не відрізняються точністю вихідної інформації, оптимальне рішення не може бути потрібне. Крім того, значення цільової функції вихідного рішення використовується як рекордне під час роботи комбінаторних алгоритмів.

Мета роботи. Метою роботи є пошук комбінованих алгоритмів визначення початкового розв'язання задач дискретної оптимізації для підвищення оперативності отримання рішення. У цій роботі пропонується використовувати деякі детерміновані та ймовірнісні прийоми як засоби отримання пріоритетної черги конкретизації значень компонент вектора рішень. Отриманий варіант розв'язання оптимізаційної задачі може використовуватися як наближене рішення. Значення цільової функції розглядається як рекордне значення при відсіюванні неперспективних підмножин варіантів на базі методу гілок та меж чи інших комбінаторних алгоритмів. Пропонуються засоби визначення початкових варіантів рішень без великих обчислювальних складнощів отримання рішення, яке може бути вихідним для точних алгоритмів, а також наближеним, якщо особлива точність не потрібна.

Основна частина. Основним комбінаторним методом вирішення завдань дискретної оптимізації є метод гілок та кордонів. Бездоказова його збіжність за рахунок кінцівки безлічі варіантів, гнучкість і відкритість дозволяє вводити блоки, що модифікують та що прискорюють процес отримання оптимального рішення. Наявність процедури оцінювання множини варіантів, розбиття множини на підмножини, а також ознаки оптимальності, дає можливість повний перебір варіантів призвести до часткового перебору, за рахунок відсіювання неперспективних підмножин. Насамперед, це алгоритмізація різних способів отримання оцінок (кордонів) множини (підмножин) варіантів розв'язання задачі. Найчастіше використовується ідея розширення – звуження їхньої області визначеності [2].

Якщо Z' оптимальне значення цільової функції, що отримане на розширеній множині варіантів G', то воно може використовуватися як оцінка вихідної множини G. Позначимо її через $\xi(G)$.

При розв'язанні задач цілочисельного лінійного програмування (ЦЛП) розширення безлічі варіантів проводиться шляхом приведення завдання до відповідної задачі лінійного програмування (ЛП), тобто. відкидаються вимога цілісності компонентів вектора рішень.

У роботі [3] наведено два способи розширення безлічі варіантів для вирішення задачі про багатомірний ранець методом гілок та меж. Дано експериментальну оцінку їх ефективності.

Дійсно, якщо визначається максимальне значення цільової функції Z, де

$$Z = \max \sum_{j=1}^{n} c_j x_j \tag{1}$$

на множині варіантів G, що задається обмеженнями

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = \overline{1,m} \tag{2}$$

$$x_j \in \{0,1\}, \quad j = \overline{1,n} \tag{3}$$

то замінив вимогу (3) на (3`) отримаємо множину варіантів G', що задається обмеженнями (2)-(3'), де

$$x_j \in [0,1], \quad j = \overline{1,n} \tag{3'}$$

Маємо задачу лінійного програмування

$$Z' = \max \sum_{j=1}^{n} c_j x_j$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = \overline{1,m},$$

$$x_j \in [0,1], \quad j = \overline{1,n}.$$

Завжди $G \subseteq G'$ і отже, $Z' \ge Z$. Оптимальне значення Z' є оцінкою множини цілочисельних варіантів тобто $Z' = \xi(G)$.

Більш спрощений спосіб отримання оцінок полягає у вирішенні одновимірних m не цілочисельних завдань про ранець, а саме

$$Z_i' = \max \sum_{j=1}^{n} c_j x_j$$

при обмеженнях $\sum_{j=1}^{n} a_{ij} x_j \le b_i$, $x_j \in [0,1], \quad j = \overline{1,n}$, складових по черзі підмножини G', $i = \overline{1,m}$. Оцінка безлічі цілих варіантів рішення визначається як:

$$\xi(G) = \min Z_i', \ i = \overline{1,m}.$$

Очевидно, що для всіх $i = \overline{1,m}$ справедливе співвідношення $G \subseteq G'$.

Якщо складність обчислень визначається кількістю виконуваних операцій, другий спосіб значно менш складний. Отримання оптимальних не цілочисельних рішень одномірних завдань про ранець методом Данцига [4] простий, не містить жодних ітераційних процесів. Цього сказати про перший спосіб не можна. Визначення оцінки $\xi(G)$ вимагає використання симплекс-методу, що відрізняється своєю обчислювальною складністю. Якщо складність обчислень визначається числом ітерацій, виконаних до отримання оптимального рішення, краще використовувати перший спосіб. Значення цільової функції Z' на розширеному множині менше відхилиться від Z на вихідному. Кількість вершин дерева рішень, що переглядаються, значно зменшиться. Процедура розбиття безлічі варіантів на підмножини є досліджуваною і модифікованою частиною методу. Відомі різні підходи поділу безлічі варіантів на підмножини. Від цього залежить пошук перспективного, що містить оптимальний варіант підмножини однієї з перших робіт за методом гілок і кордонів є алгоритм Ленд і Дойг [5]. Пропонується безліч варіантів ділити на два підмножини. Параметром поділу є нецілочисельна компонента x_k в оптимальному варіанті розв'язання відповідної задачі ЛП. Мотивація така – виключити з розгляду варіанти, у яких зустрічатимуться нецілочисельні значення компоненти x_k навколо її оптимального значення. Безліч варіантів ділиться на два підмножини шляхом доповнення до основних обмежень таке $x_k \le [x_k]$ чи таке $x_k \ge [x_k + 1]$ де [a] – ціла частина від a . Такий спосіб розбиття, як і весь алгоритм, не мав успіху. Предком методу гілок та меж вважається алгоритм Літла та ін. [6] для вирішення задачі про комівояжер. Параметром поділу є пара міст, яку доцільно включити до циклу об'їзду. Пари оцінюються величиною «втрат» у значення цільової функції, якщо не будуть включені до циклу. Перевага надається парі з найбільшою кількістю втрат. Існує безліч модифікацій методу для вирішення задачі про комівояжер: точні [7], наближені [8], імовірнісні – наближені [1], з симетричною матрицею відстаней [4], обчислювальні алгоритми для вирішення задач великої розмірності [9] та багато інших. Ідея послідовної побудови рішення [10] внесло деяке впорядкування в процедуру розбиття безлічі варіантів на підмножини. На кожному етапі розбиття конкретизуються можливі значення компоненти вектора рішень. Множина, що розбивається, ділиться на частини, кожна з яких містить варіанти, що відрізняються за значенням змінної, що визначається. Проблема полягає у визначенні пріоритетної черги компонент, згідно з якою вони будуть конкретизовані. Від цього залежить ступінь наближення до оптимального одержуваного рішення [11]. Відомі також алгоритми як жадібний [12], генетичний [13], мурашиної колонії [14], багато комбінованих (наприклад [15]), що пропонують різні способи формування пріоритетної черги. Багато хто з них використовується для отримання наближеного рішення, рекордного значення цільової функції або гарного початкового варіанта рішення, що підлягає подальшому покращенню. У методі гілок та меж рекордне значення використовується як засіб відсіювання варіантів.

Наближені алгоритми комбінаторної оптимізації є двоетапними. На першому етапі якимось засобом визначається варіант рішення, але в другому – його поліпшення. Поліпшення – це зміна значень параметрів, спеціальні прийоми, застосування методик іншого способу, або, з так званої ідеї двоїстості. У задачах з булевими змінними деяким компонентам, що мають значення «1», присвоюється значення «0» і навпаки. Що ж до першого етапу, то допустиме рішення виходить або детермінованим способом, або з використанням елементів випадкового пошуку. Сама процедура отримання варіанта рішення розглядається як присвоєння позитивних значень компонентам вектора рішень згідно з їхньою пріоритетною чергою, яка визначається з урахуванням структурних особливостей завдання, або пропорційно величині ймовірності, що обчислюється в залежності від ряду параметрів.

У роботі запропонована така схема класифікації наближених алгоритмів комбінаторної оптимізації (рис. 1).

Нижче наводиться короткий опис підходів формування пріоритетної черги компонент вектора рішень для задач у постановці (1)-(2)-(3).

Позначення:

– вектор рішень $X = \{x_j\}, \quad j = \overline{1,n}$;

– множина індексів конкретизованих компонент $I_x = \{j / x_j = 1\}$;

– множина індексів компонент вектора рішень, яким можна присвоїти значення «1»: $V_j = \left\{ \bar{j} \in I_x / (b_i - \sum_{j \in I_x} a_{ij}) \geq 0 \right\}$, $\forall i$;

– пріоритетна черга індексів компонент вектора рішень $S_j = \{j_1, j_2, \dots, j_k\}$.

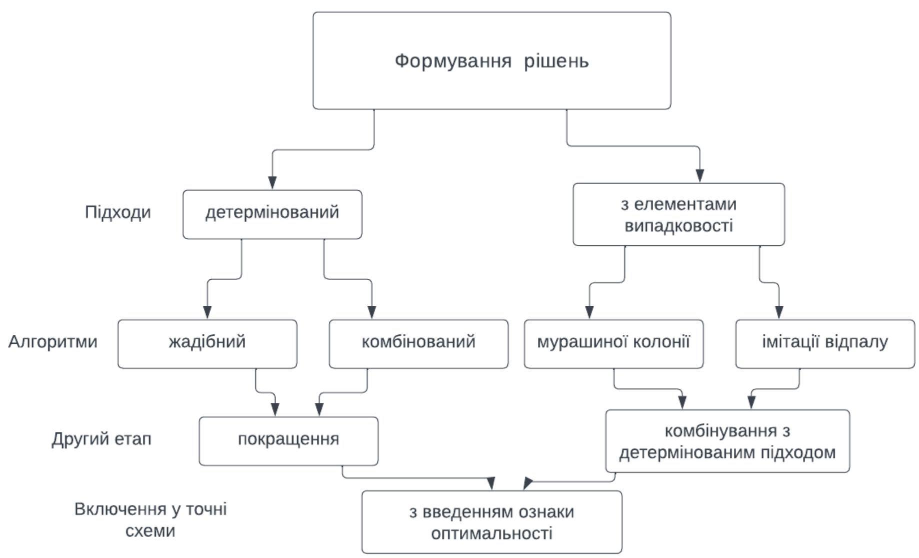

Рис.1. Класифікація наближених алгоритмів комбінаторної оптимізації

При детермінованому підході пріоритетна черга формується лише з урахуванням числових даних, які входять у математичну модель. До них належить

$$C = \{c_j\}_{j=\overline{1,n}};\ A = a_{ij},\ i = \overline{1,m};\ j = \overline{1,n};$$

та

$$B = \{b_i\}\ i = \overline{1,m}.$$

Ідея жадібного алгоритму передбачає облік лише компонент вектора цін як величин, які вносять «внесок» значення цільової функції.

Якщо $c_{j_1} \geq c_{j_2}, \dots, \geq c_{j_k}$, то пріоритетна черга $S_j = \{j_1, j_2, \dots, j_k\}$.

Комбінований спосіб поєднує два моменти: ідею жадібного алгоритму та врахування величини нев'язки у системі обмежень при конкретизації чергової компоненти x_j вектора рішень. Визначається «вагова» оцінка q_j для $x_j / j \in V_j$ таким чином отримаємо:

$$q_j = c_j \sum_{i=1}^{m} (b_i - a_{ij}).$$

Ранжування в порядку зростання величин q_j визначає пріоритетну чергу S_j .

При недетермінованому підході пріоритетність компонентів визначається величиною ймовірності. Чим більша ця величина, тим більший шанс вибору відповідної компоненти. Сам вибір, зазвичай, здійснюється шляхом генерації випадкової величини, рівномірно розподіленої в інтервалі (0,1). Інтервал ділиться на частини пропорційно до величин ймовірностей. Попадання значення випадкової величини у відповідну частину інтервалу визначає номер компоненти. Відкритим залишається питання про обчислення величин ймовірностей, що надають пріоритетність компонентам. Один із способів це імітація поведінки живих неінтелектуальних істот чи природних явищ. Формалізація такого недетермінованого природного процесу призводить до набуття числових значень. У роботі [1] описано, як поведінка особин мурашиної колонії подає ідею створення алгоритмів розв'язання задач комбінаторної оптимізації. Багаточисельність жителів колонії, кожен із яких виконує самостійно свою роботу, досягаючи загальної мети, сприймається як елемент масивності. У імовірнісних алгоритмах це представляється як набір статистичних даних про отримувані рішення. Вирішується те саме завдання багатьма агентами (ніби мурахами), число яких представляється параметром у роботі алгоритму.

Інший дуже важливий момент це наявність передачі між особами колонії. При пересуванні відкладається слід через звані феромони. Чим більше їх конкретизація, тим більше мурах піде цим шляхом. Формально – кількість керує процесом та забезпечує прийняття рішень. Природно, кількість феромонів може збільшуватися, але з часом і зменшаться - випаровування. Результат рішення визначається їх кількістю. У мурашиних алгоритмах це є двома параметрами α та β. Випаровування феромонів залежить від відстані між точками пересування. Чим вона більша, тим значніше зменшується їх кількість. Говорять, зменшується інформативність про стан справ. Цей компонент виявляється у алгоритмах через структурні особливості розв'язуваної задачі. З метою отримання ймовірнісної оцінки компоненти вектора рішень необхідно дати формальну оцінку накопичення інформативності про неї (феромонів) з урахуванням зворотного інформаційного зв'язку (випаровування феромонів) через структурні особливості завдання, що розв'язується.

Позначимо:

α – рівень інформативності;

β – рівень зворотнього інформаційного зв’язку;

a_j – інформація про *j*-у компоненту вектора рішень (наявність віртуальних феромонів) $j = \overline{1,n}$;

r – кількість паралельно розв'язуваних завдань (мурах колонії).

Тоді величина ймовірності визначається за формулою

$$P_j^l = \frac{a_j^\alpha Y(c)^\beta}{\sum_{j \in V_j^l} a_j^\alpha Y(c)^\beta} \quad j \in V_j^l;\ l = \overline{1,r} \tag{4}$$

де

$$Y(c)=\{c_j - \min c_j, \textit{якщо}\, c_j > \min c_j\}.$$

Величина ймовірності визначається для тих компонентів, які з структурних особливостей, можуть набувати значення «1».

Отримаємо r варіантів рішення $X^l = \{x_j^l\}$ $j=\overline{1,n}; l=\overline{1,r}$, де кожна x_j – а компонента приймає значення «1» з ймовірністю, що визначається за формулою (4).

Початкове значення a_j є однаковим для всіх $j=\overline{1,n}$ та всіх $l=\overline{1,r}$. Надалі при виконанні етапу поліпшення наближеного рішення значення будуть змінюватися з урахуванням накопиченого досвіду отримання безлічі варіантів рішення.

Врахування структурних особливостей задачі, що виражається через величину $Y(c)$, може здійснюватися індивідуально (див. напр. [16]). Використовуючи мотивацію генетичного алгоритму [12] «отримати добре потомство», визначаються вагові оцінки γ_j для кожної компоненти x_j вектора рішень, де

$$\gamma_j = \sum_{l=1}^{r} x_j^l / n,\ j \in V_j^l.$$

Покращення наближеного рішення розглядається як ітераційний процес. Якщо t номер ітерації, то при переході від ітерації t до ітерації $t+1$ змінюється значення a_j за формулою (5)

$$a_j(t+1) = (1-\rho)a_j(t) + \gamma_j \tag{5}$$

де ρ – регулюючий параметр.

Другий детермінований підхід формування вектора рішень, заснований на ідеї імітації відпалу [17]. Це формалізація фізичного процесу, пов'язаного із зміною внутрішньої енергії тіла, яке розігрівається до високої температури і далі починає повільно остигати. Внутрішня його енергія переходить із одного стану до іншого. Потрібно знайти точку з мінімальною внутрішньою енергією. Дослідження того, що відбувається, дало можливість створити алгоритм. Оптимізаційна задача розглядається як фізична система, а допустиме рішення та значення цільової функції як стан тіла та його внутрішня енергія. У такій аналогії процес відпалу представляє процес знаходження рішення з мінімальним значенням цільової функції.

Досить простий у обчислювальному сенсі алгоритм імітації відпалу використовувався для розв'язання задачі про комівояжера, найпростішого завдання розміщення та інших прикладних завдань [18]. Крім того, доведена його асимптотична збіжність [17]. З математичних позицій алгоритм належить до класу алгоритмів локальної оптимізації. Оптимальне рішення (локальний оптимум) шукається на околиці локального оптимуму. При вирішенні оптимізаційних завдань з булевими змінними поняття «околиця» сприймається як сукупність рішень, у яких значення однієї з компонентів замінюється протилежним («1» на «0» чи навпаки).

Алгоритм імітації відпалу безпосередньо не встановлює пріоритетну чергу конкретизації компонентів вектора рішень. В околиці деякого допустимого рішення випадковим чином на основі ймовірнісної оцінки здійснюється пошук локального оптимуму. Величина ймовірності залежить від температур t та напряму, у якому ведеться пошук. Хай визначена початкова температура t_0. Використовуя будь-який спосіб формування рішення, отримано варіант X та значення цільової функції $Z(X)$. На околиці X за допомогою деякого оператора Ψ визначено варіант X' та $Z(X')$.

Обчислюється відхилення $\Delta = Z(X') - Z(X)$, якщо $\Delta > 0$, то рахується що стан X' краще, ніж X з ймовірністю $p = e^{-\Delta / t}$.

Перехід у новий стан також має випадковий характер. Генерується випадкова величина ξ рівномірно розподілена в інтервалі (0,1). Якщо $\xi > p$, то стан, що описує X зберігається, в іншому випадку переходимо в стан X' тобто $X = X'$ і $Z(X) = Z(X')$; зменшується температура на деякий $\Delta t\,(t = t - \Delta t)$ та повторюється ітераційний процес пошуку нового локального оптимуму. Процес керується або досягненням певної температури t, або досягненням достатнього значення цільової функції.

Приклад. Розв'язується багатовимірне завдання про ранець з використанням ряду модифікуючих блоків методу гілок і кордонів, кожен з яких певною мірою покращує ефективність його роботи. Числові дані завдання запозичили з [4]. Маємо задачу у постановці:

$$Z = \max(6x_1 + 5x_2 + 2x_3 + 3x_4 + 4x_5 + 5x_6 + 4x_7 + 8x_8 + 7x_9 + 3x_{10})$$

при обмеженнях

$$\begin{cases} 2x_1 + 2x_2 + 2x_3 + x_4 + 2x_5 + 2x_6 + 2x_7 + 3x_8 + 3x_9 + x_{10} \le 10 \\ x_1 + 3x_2 + 3x_3 + 2x_4 + 2x_5 + 2x_6 + x_7 + 4x_8 + 2x_9 + 2x_{10} \le 7 \\ 4x_1 + x_2 + 2x_3 + 2x_4 + 2x_5 + 3x_6 + 3x_7 + 3x_8 + 2x_9 + x_{10} \le 8 \end{cases}$$

$$x_j \in \{0,1\}, \quad j = \overline{1,10}.$$

Оцінка множини варіантів $\xi(G)$ розраховується використовую процедуру його розширення. Було вирішено три не цілочисельних одновимірних задач про ранець методом Данцига [4]. Величини $\lambda_j^1 = \dfrac{c_j}{a_{ij}}, j = \overline{1,n}$ ранжуються у порядку їх не спадання, на основі чого визначаються послідовності присвоєння значень «1» відповідним компонентам відповідного вектора рішень. Маємо:

Таблиця 1.

Результати розрахунків оцінок множин варіантів

i	Послідовність	Варіант рішення	Z_i^*	
1	1 4 10 8 2 6 9 5 7 3	1 1 0 1 0 ½ 0 1 0 1	27,5	$\xi(G)$=24
2	1 7 9 6 8 5 2 4 10 3	1 0 0 0 0 1 1 ¼ 1 0	24,0	
3	2 9 10 8 5 6 1 4 7 3	0 1 0 0 ½ 0 0 1 1 1	25,0	

Початкове рекордне значення цільової функції R визначалося згідно з пріоритетною чергою компонент вектора рішень. Розглядалися два способи: перший спосіб – пріоритетність компоненти визначалася величиною відповідної компоненти вектора вартості; другий – з урахуванням величини нев'язки у системі обмежень.

Таблиця 2.

Результати розрахунків пріоритетності компоненти

Спосіб	Пріоритетна послідовність	Варіант рішення	R
1	8 9 1 2 6 5 7 4 10 3	0 0 0 0 0 0 1 1 1 0	19
2	9 1 2 6 8 5 7 10 4 3	1 1 0 0 0 0 0 0 1 0	18

Оптимальне значення цільової функції Z^* знаходиться в інтервалі $19 \le Z^* \le 24$.

Процес отримання оптимального рішення та значення Z^* наведено на рисунку 2. Дерево рішень будувалося шляхом послідовної конкретизації змінних вектора рішень згідно з пріоритетною чергою S, отриманою способом 1.Оцінки підмножин визначались алгоритмом Данцига. Підмножини представлені j чи $\overline{j}$ означає яка компонента та яким значенням конкретизується. Продовження гілки дерева відсікається якщо оцінка підмножини не більша за рекорд R або відповідне значення неприпустимо.

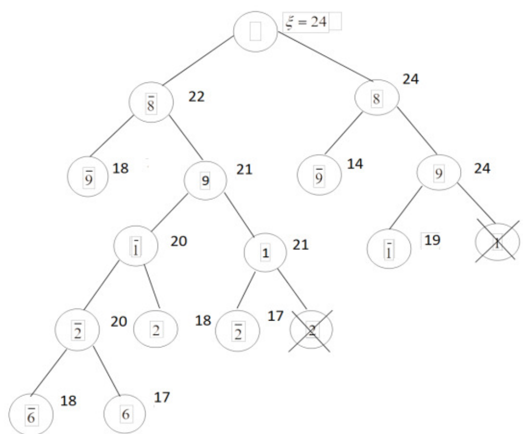

Рис.2. Дерево рішень

Рішення цього прикладу алгоритмом мурашиної колонії здійснювалося за таких умов. Початкове значення $\alpha_0 = 0{,}1$. Далі, після виконання кожної ітерації α визначалося згідно з наведеною формулою (5). Параметр $\beta = 0{,}1$ зберігався незмінним. Значення $\gamma_j \left(j = \overline{1,n}\right)$ визначалися на кожній ітерації. Функція $Y(c)$ (облік структурних особливостей задачі) обчислювалося як наведено за формулою 4, де V_j^i – множина індексів компонент вектора рішень претендентів на значення «1». Процедуру рішення здійснювало 6 мурах (агентів). Результат наступний: виконано 20 ітерацій, оптимальне рішення отримано двічі. На 11 ітерації п'ятою мурахою і на 17-ї ітерації третьою мурахою.

Висновки. Ефективність роботи наближених алгоритмів оцінюється ступенем наближеності одержуваного рішення до оптимального. Важливу роль також грає оперативність отримання прийнятного рішення. Пропонуються способи визначення початкових варіантів рішень, що дозволяють без великих обчислювальних складнощів отримати рішення, яке може бути вихідним для точних алгоритмів, а також наближеним, якщо особлива точність не потрібна.

Наприклад, у способі гілок та меж важливу роль відіграє наявність початкового рекордного значення цільової функції R. Кількість висячих вершин дерева рішень при роботі безпосередньо залежить від величини R і його близькості до оптимального значення цільової функції. Чим ближче до оптимального значення R, тим більше неперспективних підмножин буде виключено з розгляду. Розглянуті способи отримання рішення дає добре, інколи і оптимальне рішення. Тому будь-який алгоритм методу гілок та меж може бути модифікований способом отримання початкового варіанта рішення, значення цільової функції якого буде рекордом R. Такий позитивний момент надає суворості ознаки оптимальності методу. Крім того, спрощується формування R-близьких рішень.

Запропоновані способи визначення початкових варіантів рішень дозволили зменшити час на отримання рішення в 1,2 рази у порівнянні з класичними алгоритмами розв'язання задач дискретної оптимізації.

Список літератури

1. Штовба С.Д. Муравьиные алгоритмы. *Exponenta Pro. Математика в приложениях*. 2003. № 4. С.70-75. URL: http://surl.li/hbolu
2. Таха Х.А Введение в исследование операций. М.: Вильямс, 2007. 910 с.
3. Ткаленко О.Ю. К вопросу оценки сложности алгоритмов метода ветвей и границ. *Project, Program, Portfolio p3 Management*: Матеріали другої Міжнародної науково-практичної конференції. 2017. Т.1. С.88-92. URL: http://dspace.opu.ua/jspui/ handle/123456789/6992
4. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование. М.: Физматлит, 2007. 304 с.
5. Land A.H., Doig A.G. An automatic method of solving discrete programming problems. *Econometrica*. 1960. V. 28, № 3. P. 497-520. URL: https://doi.org/10.2307/1910129.
6. Литл Дж, Муртик К., Суини Д., Керел К. Алгоритм для решения задачи коммивояжера. *Экономика и математические методы*. 1965. Т.1. вып.1. С. 94-107. URL: http://surl.li/hboyh
7. Lai X., Hao J.K., Glover F., Lii Z. A two-phase tabu – evolutionary algorithm for the 0-1 multidimensional knapsack problem. *Information Sciences* 2018. V.436. P. 282-301. URL: https://doi.org/10.1016/j.ins.2018.01.026
8. Erlebach T., Kellerer H., Pferschy U. Approximating multiobjective knapsack problems. *Management Science*. 2002. 48 (12). P.1603-1612. https://doi.org/10.1287/mnsc.48.12.1603.445
9. Хачатуров В. Р., Веселовский В.Е., Злотов А.В. Комбинаторные методы и алгоритмы решения задач дискретной оптимизации большой размерности. М.: Наука , 2000. 360 с.
10. Михалевич В.С. Последовательные алгоритмы оптимизации и их применение. *Кибернетика*. 1965. №1. С. 45—55; №2. С. 85—88.
11. Юхименко Б.И., Козина Ю.Ю. Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования. Труды *Одесского политехнического университета*. 2005. Вып.2. С.199-204. URL: http://surl.li/hbphv
12. Дюбин Г.И., Корбут А.А. Жадные алгоритмы для задачи о ранце: поведение в среднем. *Сибирский журнал индустриальной математики*. 1999. №2 (24). С.68-98. URL: http://surl.li/hbpjl
13. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит. 2006. 250с.
14. Штовба С.Д. Муравьиные алгоритмы: теория и приложения. *Программирование*. 2005. №4. С.3-18. URL: http://surl.li/hbpkj
15. Юхименко Б.И., Волкова Н.П. Приближенные алгоритмы решения задач о многомерном ранце. *Дослідження в математиці і механіці*. 2017. Т.22 .

вып.2(30). С.104-115. URL: http://rmm-journal.onu.edu.ua/article/view/135745/pdf_39

16. Юхименко Б.И., Ткаленко О.Ю. Алгоритм муравьиной колонии для многомерной задачи о ранце. *Реєстрація, зберігання і обробка даних*. 2019. Т.21, №2. С.3-11. URL: http://drsp.ipri.kiev.ua/article/view/180014/184142
17. Lundy M., Mees A. Convergence of an annealing algorithm. *Math. Programing.* 1996. V.34. pp.111-124. URL: https://link.springer.com/article/10.1007/BF01582166
18. Леванова Т.В. Алгоритм муравьиной колонии и имитации отжига для задач о p-медиане. *Автоматика и телемеханика*. 2004. №3. С.80-88. URL: http://surl.li/hbpne

COMBINED ALGORITHMS FOR DETERMINING THE INITIAL SOLUTION OF DISCRETE OPTIMIZATION PROBLEMS

B. I. Yukhimenko[1], N.H. Volkova[2], Yu.Yu. Kozina[3]

National Odesa Polytechnic University,
Shevchenko Ave., 1, Odesa, 65044, Ukraine
e-mails: biruteyu@gmail.com[1], volkova.n.p@op.edu.ua[2], yuliyakc21@gmail.com[3]

The problem of solving the task of discrete optimization is not completely solved. Lack of publications, scientific developments, algorithms and software products do not give the ability to transfer the mathematical apparatus of solving discrete optimization problems to the class P of computational complexity. All analytical and combinatorial algorithms for solving problems of linear and non-linear optimization are NP completeness. There are developments to improve the efficiency of robotic algorithms, they are filled with requests and actual ones. In this paper, it is proposed to use deterministic and probabilistic methods for forming a priority queue of decision vector components in order to assign positive values to them. After the formation of the variant of the solution, it is possible to vectorize as if the approached solution is taken away from the record value of the goal function, which in exact algorithms is like a solution, which results in a polyp. In this paper has a method for forming a priority line for concretizing the components of the solution vector. The basis of deterministic methods is the idea of a greedy algorithm. The location in the queue is determined by the value of the corresponding component of the cost vector. The appearance of the value of non-visibility in the system of restrictions increases the priority of the component. Behind such a way, another method of determination is modified. Probability assessment probability of priorities is based on the ideas of algorithms in an ant colony and simulated annealing. The scope of probability indicates the significance of the component – a contender for a positive value. A numerical example of a small variability of the task about a knapsack has been introduced, which demonstrates the imitation of a nearby solution.

Keywords: Discrete optimization, record, priority queue, branch and bound method.