УДК 004.421

Н.О. Маслова, канд. техн. наук, доц.,
Н.В. Лисакова, магістрант,
Донецький національний технічний університет,
м. Красноармійськ, Украина
masgpp@list.ru

Застосування комбінованого алгоритму при рішенні багатовимірних задач

Робота присвячена опису алгоритму, що є комбінацією методу динамічного програмування та методу меж та гілок. Метод динамічного програмування суттєво спрощує задачі, але безпосереднє його використання, як правило, пов'язане з значними обчисленнями. Для уникнення цього застосовуються наближені методи динамічного програмування, апроксимація та дискретизація. В багатомірному випадку одним з сучасних та актуальних напрямів є розробка комбінованих алгоритмів.

Ключові слова: динамічне програмування, комбінований алгоритм, багатовимірна задача, ефективність.

Вступ

Термін «динамічне програмування» вперше пропонував Р. Беллман для опису методу рішення задачі, де відповідь можна отримати тільки після рішення попередньої задачі.

У наш час цей термін застосовується в теорії управляння та теорії обчислювальних систем й визначає метод рішення складних задач за допомогою розбиття їх на простіші крокові задачі. Слово «програмування» у цьому словосполученні розуміють як «прийняття рішень», «планування», «оптимізація», а слово «динамічне» вказує на суттєве значення часу та порядку виконання операцій в процесах і методах, що розглядаються. Й, таким чином, «динамічне програмування» - це напрямок, який присвячено теорії і методам розв'язання багатокрокових задач оптимального управління, а внесок Р. Беллмана в теорію динамічного програмування зафіксовано в понятті «рівняння Беллмана», та можливості представлення оптимізаційної задачі у вигляді рекурсії.

Метод динамічного програмування суттєво спрощує вихідні задачі, але безпосереднє його використання, як правило, пов'язане з значними обчисленнями. Для подолання цих труднощів розробляються наближені методи динамічного програмування, застосовується апроксимація та дискретизація. Для рішення дискретних оптимізаційних завдань найбільш часто застосовують методи динамічного програмування і метод меж та гілок. А в багатомірному випадку одним з сучасних та **актуальних** напрямів є розробка комбінованих алгоритмів [1,2].

Метою пропонованої статті є опис комбінованого алгоритму, в основі якого - метод динамічного програмування та метода меж та гілок; аналіз ефективності розробленого алгоритму; порівняння його із методом динамічного програмування та методом гілок та меж.

Опис

Загальна ідея в динамічному програмуванні (ДП) виглядає так: щоб знайти рішення багатокрокової задачі, потрібно розбити її на окремі частини (підзадачі), знайти рішення цих спрощених задач, після чого об'єднати рішення підзадач в одне спільне рішення. Іноді деякі з цих підзадач однакові. Підхід динамічного програмування полягає в тому, щоб вирішити кожну підзадачу тільки один раз, скоротивши тим самим кількість обчислень. Це особливо корисно у випадках, коли число повторюваних підзадач експоненціально велике.

Типовий алгоритм вирішення задач методом динамічного програмування [3]:

1. Описати схему побудови оптимального рішення.
2. Виписати рекурентне співвідношення, що зв'язує оптимальні значення параметра для підзадач.
3. Рухаючись знизу вгору, обчислити оптимальне значення параметра для підзадач.
4. Користуючись отриманою інформацією, побудувати оптимальне рішення

Підзадачі вирішуються розподілом їх на частки ще меншого розміру і т. д., поки не приходять до завдання, знаходження рішення для якого досить тривіально. Обчислювальна перевага такого підходу полягає в тому, що вирішуються одномірні оптимізаційні підзадачі замість великої n-мірної задачі. Обчислення виконуються рекурентно в тому сенсі, що оптимальне рішення однієї підзадачі використовується в якості вихідних даних для наступної.

Трудомісткість рішення задачі динамічного програмування визначається головним чином розмірністю задачі, обумовленої числом параметрів стану на кожному кроці (s) і числом змінних управління на даному кроці (r).

Ці числа взаємопов'язані й задачі, одномірні по s і багатомірні по r можуть бути перетворені в одномірні по r і багатомірні по s. Тому розмірність задачі можна умовно характеризувати, добутком (r*s).

Виділяють одно-, двох- та багатовимірні задачі динамічного програмування. Багатовимірне динамічне програмування відрізняється від одновимірного кількістю вимірів, тобто кількістю параметрів станів.

Одномірною є, наприклад, задача розподілу коштів, якщо вони виділяються конкретними порціями на кінцеву кількість періодів. До багатомірних задач динамічного програмування, рішення яких остаточно ще не знайдено, відносяться задача пошуку шляху в орієнтованому ациклічному графі, задача комівояжера, задача амортизації, задача розподілу ресурсів підприємства зі значною кількістю філій.

Головним недоліком динамічного програмування є зростання складності зі збільшенням розмірності задачі. А перевагою є можливість застосування на кожному кроці ефективних алгоритмів рішення одномірних задач.

На рисунку 1 відображено зростання обчислювальної складності методу при збільшенні кількості елементів системи.

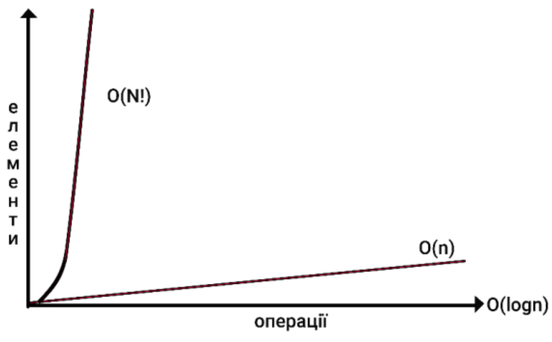

Рисунок 1 - Обчислювальна складність методу ДП

Таким чином, якщо багатомірна модель може бути розбита на кінцеву кількість одномірних задач й для цих задач будуть вибрані ефективні за часом (швидкодією) методи рішення, то загальний час рішення основної задачі також буде зменшено. Так, якщо для рішення одномірного блоку обрати метод повного перебору, то при m кроках рішення загальної задачі, отримуємо кількість операцій, що кратна m* O(N!), а у випадку застосування, наприклад, методу Балаша, ця складність становить $O(2^n)$.

Через той факт, що задачі динамічного програмування не втрачають своєї актуальності у теперішній час, існує багато інструментів для їх рішення. Однак через те, що кожна задача повинна мати індивідуальний підхід, бо у кожної конкретної задачі буде особлива цільова функція, то створити універсальний інструмент майже неможливо.

У вільному доступі у мережі інтернет можна знайти декілька онлайн калькуляторів, що допоможуть вирішити динамічні задачі наступних типів:

– задачу розподілення ресурсів;
– задачу розподілення інвестиції;
– задачу заміни обладнання;
– задачу Джонсона;
– задачу про рюкзак.

Хоча і існують звичайні інструменти для вирішення типових задач, вони підходять більше для навчання, ніж для використання при рішенні реальних задач, наприклад, на підприємстві. Тому, найчастіше для реальних задач створюються окремі програмні продукти або, застосовують спеціально розроблені алгоритми.

Метод гілок і меж – один з поширених методів дискретної оптимізації. Метод працює на дереві рішень та визначає принципи роботи конкретних алгоритмів пошуку розв'язків, тобто, є мета-алгоритмом. Для різних задач комбінаторної оптимізації створюють спеціалізовані алгоритми гілок та меж [4].

Ідею методу було вперше сформульовано A.H. Land та A.G. Doig (1960) в галузі дослідження операцій. R.J. Dakin (1965) розробив простий для впровадження алгоритм.

Результатом роботи алгоритму є знаходження максимуму функції на допустимій множині. При цьому множина може бути як дискретною, так і раціональною. В ході роботи алгоритму виконуються дві операції: розбиття вихідної множини на підмножини (гілки), та знаходження оцінок (меж). Існує оцінка множини згори та оцінка знизу. Оцінка згори – точка, що гарантовано не менша за максимум на заданій підмножині. Оцінка знизу – точка, що гарантовано не більша за максимум на заданій підмножині. Множина, що має найбільшу оцінку зверху зветься рекордною. На початку вся множина вважається рекордною. Далі:

1) рекордна множина розбивається на підмножини;
2) знаходять оцінки згори та знизу для нових підмножин;

3) визначають максимальну оцінку знизу серед усіх підмножин;
4) видаляють ті множини, у яких оцінка зверху менша за максимальну оцінку знизу;
5) знаходять максимальну оцінку згори серед усіх підмножин та вважають її рекордною;
6) якщо не досягнуто дискретності, або необхідної точності, повертаються до пункту 1.

Результатом роботи є значення між оцінкою згори та знизу для рекордної множини. Точністю є різниця між верхньою та нижньою оцінками, тобто для дискретних множин алгоритм завершений тоді, коли ці оцінки співпадають.

Метод використовується для вирішення деяких NP-повних задач. Швидкість алгоритму залежить від вигляду функції та способу визначення оцінок, але гарантовано не більше за повний перебір.

Алгоритм методу гілок і меж передбачає декомпозицію вихідної задачі лінійного програмування (ЗЛП) на послідовність завдань, що містять додаткові обмеження на змінні, які потім оптимізуються.

1. Процес починають з рішення задачі симплексним або графічним методом без урахування вимоги на цілочисельність змінних. Цю задачу називають ЗЛП-0. Якщо всі змінні оптимального плану цілі, то цей план також є оптимальними для задач цілочисельного програмування.

2. Якщо деяка змінна, не отримала цілочисельного значення, то проводиться розгалуження на дві нові завдання ЗЛП-1, ЗЛП-2. Одне із завдань ЗЛП-1 являє собою завдання ЗЛП-0, доповнену обмеженням $x_j \le [x_j]$ де $[x_j]$ - ціла частина числа x_j. Друга утворюється шляхом додавання до задачі ЗЛП-0 обмеження $x_j \le [x_j] + 1$. Слід зазначити, що вибір цілочисельної змінної може бути довільним, визначатися таким чином:

– за зростанням або спаданням індексів;
– змінна представляє важливе рішення прийняте в рамках даної задачі;
– коефіцієнт в цільовій функції при цій змінній істотно перевершує всі інші.

3. Завдання ЗЛП-1 і ЗЛП-2 вирішуються самостійно. Гілка закінчується, якщо область допустимих рішень є порожньою або її оптимальне рішення повністю цілочисельне. В іншому випадку виникає необхідність розгалуження з п.2, позначаючи наступні номери завдань ЗЛП в природному порядку ЗЛП-3, ЗЛП-4.

Процес рішення можна представити у вигляді дерева, в якому вершина ЗЛП-0 відповідає початковим планом вирішення завдання, а кожна з з'єднаних з нею гілкою вершин відповідає оптимальному плану наступного завдання.

Алгоритм пошуку рішення комбінованим методом зображено на рисунку 2.

На першому етапі завдання вирішується методом динамічного програмування окремо по кожному з обмежень. Послідовності, отримані в результаті рішення функціонального рівняння динамічного програмування, надалі використовуються для оцінки верхньої (нижньої) межі цільової функції. На другому етапі завдання вирішується методом гілок і меж. При використанні цього методу визначається спосіб розбиття всієї множини допустимих варіантів на підмножини, тобто спосіб побудови дерева можливих варіантів, і спосіб оцінки верхньої межі цільової функції.

Для реалізації алгоритмів було створено три класи, відповідно назвам методів. Діаграма класів зображена на рисунку 3.

Кожен клас має статичну функцію Calculate, що приймає на вхід двовимірний масив з вагами ребір графу. Повертає цей метод строку, яка містить у собі Гамільтонів цикл, що є рішенням задачі. Також кожен клас має метод GetMinimum для пошуку мінімуму у рядку або у стовпчику. Метод PrintMatrix є допоміжнім та використовується для виводу поточного стану алгоритму у консоль. Метод TransData також є допоміжним та служить для транспонування матриці даних.

Поле _chars містить у собі перелік буків, що використовуються для означення вершин графу. Константа M використовується для означення використаних або відкинутих ребр.

Окрім того, клас BranchAndBounds має додаткові методи GetDi та GetDj, що рахують константи приведення для методу гілок та меж.

Для розробки програмного продукту, що реалізує вищезазначені методи, було обрано мову програмування C# та платформу .NET Framework. Обрані інструменти дозволяють отримати гідний результат за короткий термін.

Для аналізу якості алгоритмів було створено тестовий програмний продукт та проведено ряд випробувань на різних наборах даних.

Створений продукт дає можливість використовувати автоматично генеровані дані для подальшого розрахунку або ввести дані у поля зліва (рисунок 4). Після натискання кнопки «Розрахувати» у правій частині вікна буде відображено результат роботи алгоритмів, та затрачений час.

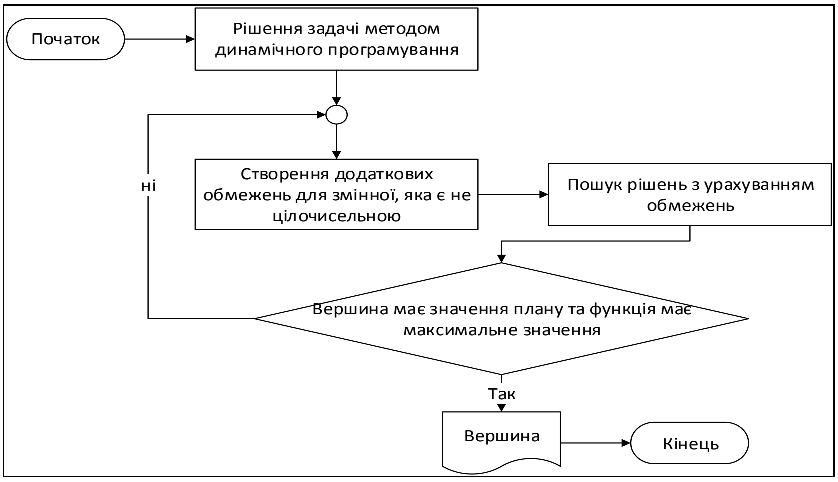

Рисунок 2 - Загальний вигляд комбінованого алгоритму

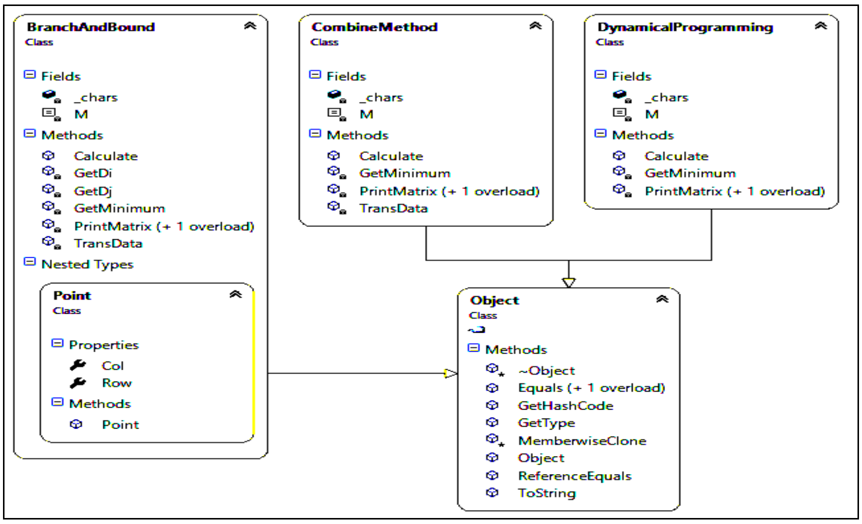

Рисунок 3 - Діаграма класів проекту

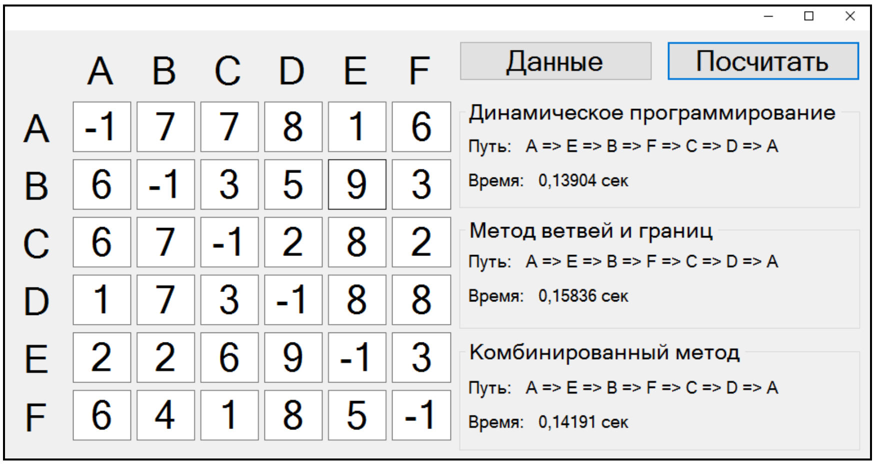

Рисунок 4 - Інтерфейс користувача

Для порівняння результатів роботи програми на різних алгоритмах та об'ємах даних було створено таблицю 1 (результати подані у секундах).

Таблиця 1 – Порівняння роботи алгоритмів

Метод	Кількість вузлів			
	6	20	50	100
Динамічне програму-вання	0,139	3,915	46,855	460,697
Метод гілок та меж	0,158	4,655	56,549	596,378
Комбінова-ний метод	0,142	4,155	49,487	488,368

Висновки

Основними напрямками комплексного застосування методів динамічного програмування і методів гілок і меж є:

– відсічення безперспективних варіантів за допомогою принципу оптимальності та оцінки меж рішення шляхом зняття умов цілочисельності;

– використання оптимальних послідовностей динамічного програмування

– відсічення безперспективних варіантів за допомогою принципу оптимальності та оцінки меж рішення за оптимальними послідовностями.

Комбінований метод з точки зору бистродії показав себе краще ніж звичайний метод гілок та меж, але він потребує трохи більшого часу, ніж метод динамічного програмування.

У подальшому для покращення результатів комбінованого алгоритму планується провести рефакторінг програмного коду, та оптимізацію викликів зовнішніх методів та зменшення кількості циклів там, де це можливо.

Список використаної літератури

1. http://www.nit7.artdesign.ru/sections/b/73.html
2. Маслова Н.А. Методы теории вычислений в решении задач управления технологическими процессами / Н.А. Маслова // Штучний інтелект. – 2007. – № 3. – С. 165–171.
3. Каліхман И.Л. Динамическое программирование в примерах и задачах: уч. пособ. / И.Л. Каліхман, М.А. Войтенко. – М.: Высш. школа, 1979. – 125 с. ил.
4. Михалевич В.С. Методы последовательной оптимизации и дискретных сетевых задач оптимального распределения ресурсов / В.С. Михалевич, А.И. Кукса. – М.: Наука. Главная редакция физико-математической литературы, 1983. – 208 с.

Надійшла до редакції 28.09.2015

Н.А. МАСЛОВА, Н.В. ЛЫСАКОВА
Донецкий национальный технический университет, г.Красноармейск, Украина

ПРИМЕНЕНИЕ КОМБИНИРОВАННОГО МЕТОДА ПРИ РЕШЕНИИ МНОГОМЕРНЫХ ЗАДАЧ

Работа посвящена описанию алгоритма, который является комбинацией метода динамического программирования и метода границ и ветвей. Метод динамического программирования существенно упрощает задачи, но непосредственное его использование, как правило, связано со значительными вычислениями. Для уменьшения объема вычислений применяются приближенные методы динамического программирования, аппроксимация и дискретизация. В многомерном случае одним из современных и актуальных направлений является разработка комбинированных алгоритмов.

Ключевые слова: динамическое программирование, комбинированный алгоритм, многомерная задача, эффективность.

N.A. MASLOVA, N.V. LYSAKOVA
Donetsk National Technical University, Krasnoarmiysk, Ukraine

APPLICATION OF THE COMBINED ALGORITHM IN SOLVING MULTI-DIMENSIONAL PROBLEMS

The work is devoted to description of the algorithm, which is a combination of dynamic programming and branch and bound method. Dynamic programming method greatly simplifies the task, but its immediate use, is usually associated with significant computing. To reduce the amount of computation approximate methods of dynamic programming, approximation and sampling are applied. In the multidimensional case one of the modern and current trends is the development of combined algorithms.

Keywords: dynamic programming techniques, combined algorithm, multidimensional task efficiency.