

ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ ЕВРИСТИЧНОГО ПОШУКУ ДЛЯ ЗАДАЧІ ПРО ТУРНІРИ

Вінницький національний технічний університет

Анотація

Досліджується застосування евристик для розв'язання задачі мінімізації відстаней, які мають подолати команди при проведенні двох кругового турніру. Досліджені евристики можуть бути використані для пошуку рішень і в інших прикладних задачах

Ключові слова: евристика, задача про турніри, алгоритми пошуку.

Abstract

Traveling Tournament Problem and how it can be solved with heuristic search is discussed. The Traveling Tournament Problem tries to find a schedule for a league that minimizes the overall traveling distance of all teams in this league. The schedule has to have the form of a double round-robin tournament.

Keywords: heuristics, a problem on tournaments, search algorithms.

Вступ

Задачі про кругові турніри відносяться до широко розповсюджених задач математичної оптимізації. Окрім теоретичного, дослідницького, такі задачі мають і практичне значення. Наприклад, в експертному методі попарних порівнянь відповіді експерта можна записати у вигляді турнірної таблиці. Так само, результати задачі про турніри можуть бути застосовані для розв'язання будь-яких задач, які можна подати турнірною таблицею [1].

Постановка задачі

Існує багато різних варіацій задачі про турніри, які відрізняються набором формальних обмежень, що робить їх особливо цікавими з точки зору дослідження застосування певних евристик. Одним з таких різновидів є складання розкладу турніру, який проводиться у два кола:

- кожна команда грає з кожною іншою командою двічі, один раз вдома і один раз на виїзді;
- жодна команда не грає з одним і тим же суперником дві гри поспіль;
- кожна команда має серію виїзних ігор (не більше трьох ігор поспіль вдома чи на виїзді), у ході якої вона не повертається додому.

Задано матрицю відстаней між усіма містами. Усі команди починають і закінчують грати у місті постійної дислокації команди. Мета полягає в тому, щоб мінімізувати сумарну відстань подорожив кожної команди протягом усього сезону. Якщо розклад складено успішно, загальні відстані подолані командами за весь турнір, можуть бути значно зменшені [2].

Розробка і аналіз евристик

Усе різноманіття алгоритмів для визначити маршруту, по якому можна потрапити з однієї точки в іншу, можна розбити на дві групи:

- алгоритми, що дозволяють визначити оптимальний шлях [3];
- алгоритми, що дозволяють знайти субоптимальний шлях [4].

Алгоритми першої групи потребують повного дослідження простору станів задачі. Найпростішим способом пошуку оптимального шляху є повний перебір всіх можливих маршрутів. Це гарантує знаходження оптимального шляху. Але на практиці частіше за все такий спосіб неможливо використати із-за надмірних накладних витрат, оскільки він потребує зберігання в пам'яті усієї досліджуваної області.

Прикладом алгоритмів пошуку субоптимальних шляхів є евристичні алгоритми, які на кожному кроці наближаються до кінцевої точки. Однак при пошуку одного з близьких до оптимального шляху слід враховувати, що спочатку важко точно передбачити, який саме варіант буде обраний. У цьому випадку можна використовувати різні алгоритми при виборі напрямку на кожному кроці.

Найбільш раціональним у такій ситуації здається розробка окремих алгоритмів для різних класів задач на базі алгоритмів визначення оптимального шляху із застосуванням евристик, що враховують специфіку даної проблемної області.

Для задачі, що розглядається, за базовий алгоритмом доцільно взяти класичний алгоритм пошуку в глибину.

Пошук в глибину має дуже скромні потреби в пам'яті. Він вимагає зберігання тільки єдиного шляху від кореня до листового вузла, поряд з рештою нерозгорнутими дочірніми вузлами для кожного вузла шляху. Після розгортання деякого вузла, він може бути видалений з пам'яті, якщо будуть повністю досліджені всі його нащадки.

Для простору станів з коефіцієнтом розгалуження b і максимальною глибиною m пошук в глибину вимагає зберігання тільки $bm+1$ вузлів. При організації пошуку в глибину з поверненнями, використовується ще менше пам'яті. При такому пошуку щоразу формується лише один нащадок. У кожному частково розгорнутому вузлі запам'ятовується інформація про те, який нащадок має бути сформований наступним. Це потребує лише $O(m)$ пам'яті, а не $O(bm)$. Ще більше пам'яті можна зекономити, якщо при формуванні нащадку безпосередньо модифікувати опис поточного стану, а не здійснюватися його попереднє копіювання. Але для цього треба мати можливість скасовувати кожну модифікацію при поверненні, виконуваному для формування наступного нащадку [5].

Недоліком пошуку в глибину є те, що при неправильному виборі він прямує вниз з дуже довгим (або навіть нескінченним) шляхом, хоча інший варіант міг би привести до рішення, що знаходиться повз кореня дерева пошуку. Цю проблему можна вирішити обмеживши допустиму межу глибини.

У пошуці у глибину можна використати ітеративне поглиблення, що дозволяє знайти найкращу межу глибини. Це досягається шляхом поступового збільшення межі (яка спочатку дорівнює 0, потім 1, потім 2 і т. д.) доти, поки не буде знайдено ціль. Така подія відбувається після того, як межа глибини досягає значення d , глибини самого поверхневого цільового вузла. Пошук за ітеративним поглибленням аналогічний пошуку в ширину в тому відношенні, що в ньому при кожній ітерації перед переходом на наступний рівень досліджується повний рівень нових вузлів.

Ефективним засобом штучного інтелекту для розв'язання класичних задач пошуку у просторі станів, оскільки поєднує в собі переваги алгоритмів пошуку в ширину і у глибину є алгоритм IDA* евристичного пошуку з ітеративним заглибленням який, вимагає мало пам'яті та гарантує знаходження оптимального шляху, якщо він існує [6]. IDA* намагається знайти рішення виконанням послідовних ітерацій пошуку у глибину. На кожній наступній глибині виконується процедура, яка з використанням межі f_{lim} на значеннях $f(n)$ оціночної функції вузлів, визначає які з вузлів слід розгорнути, породжуючи і досліджуючи їх нащадків. Межа зростає з кожною ітерацією.

Враховуючі, що у турнірній задачі всі рішення мають постійну глибину, за умов використання монотонної оціночної функції $f(n)$, кількість ітерацій можна зменшити застосувавши примусове поглиблення, яке за цих умов не впливає на гарантію оптимальності рішення [7]. При цьому кількість ітерацій може бути обмежена шляхом визначення цілого параметру $\lambda \geq 1$ і примушенням дерева пошуку на кожній ітерації виходити хоча б на λ рівнів глибше у дереві, ніж на попередній ітерації. Таким чином, вузли, у яких значення $f(n)$ перевершують межу для поточної ітерації, можуть тепер породжувати нащадків, поки їх глибина не виходить за межі глибини вузла, де межа дорівнює попередній ітерації плюс λ .

Висновки

Результати проведених досліджень довели ефективність використання запропонованих евристик для задачі мінімізації відстаней, які мають подолати команди при проведенні двох кругового турніру у порівнянні з класичними алгоритмами A^* та IDA*. Подальші дослідження полягатимуть у дослідженні евристики елітних шляхів [7]. Хоча досліджені евристики не дозволяють розв'язувати задачу кількості команд більшої за 10-12, вони складають основу для розробки евристичних алгоритмів для інших класів задач.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Заславский А.А. Задачи о турнирах / А.А.Заславский, Б.Р.Френкин, А.В.Шаповалов. – Изд.: МЦНМО. – 2017. - 112 с. - ISBN: 978-5-4439-0985-1
2. Uthus, D. C. DFS* and the Traveling Tournament Problem / D. C. Uthus, P. J. Riddle, H. W.Guesgen // Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. - Springer-Verlag.. - Volume 5547 of Lecture Notes in Computer Science, 2009. – P. 279 -293.
- 3 Месюра В.І. Основи проектування системи штучного інтелекту. Навчальний посібник . В.І. Месюра, Л.М. Ваховська. – Вінниця: УНІВЕРСУМ – Вінниця, 2000. – 96 С.
4. Арсенюк І. Р. Навчально-дослідницька система автоматичного керування радіокерованим візком І. Р. Арсенюк, В. І. Месюра, Ю. Л. Ляшенко // Вісник Вінницького політехнічного інституту, 2010. – №1. – С. 112-115
5. История компьютера. Поиск в глубину // URL: <http://chernykh.net/content/view/286/486>.
6. Korf R. E. Depth-first iterative-deepening: An optimal admissible tree search / R. E. Korf // Artificial Intelligence. – 27(1). – 1985. – P. 97-109.
7. Uthus D. C. Solving the traveling tournament problem with iterative-deepening A*/ D. C. Uthus, P. J. Riddle, H. W.Guesgen // Journal of Scheduling. – N.15. – 2011. – P. 601-614.

Малиновський Володимир Сергійович – студент групи 1 КН-13б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: voff.kelbas@yandex.ua

Месюра Володимир Іванович – к.т.н., доцент, професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

Volodymyr S. Malynovskii – Student of Department of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: voff.kelbas@yandex.ua

Volodymyr I. Mesyura – Cand. Sc. (Eng), Assistant Professor, Professor of the Computer Science Chair, Vinnytsia National Technical University, Vinnytsia.