Гріщенко Костянтин Сергійович

аспірант кафедри обчислювальної техніки

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

ORCID 0009-0008-9251-0222

Науковий керівник Писарчук Олексій Олександрович

доктор технічних наук, професор кафедри обчислювальної техніки

Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"

ORCID 0000-0001-5271-0248

**Метод розподілу замовлень на виробництво для BigData систем**

Актуальність

В умовах глобального дефіциту ресурсів та зростаючої популярності концепцій бережливого виробництва, з урахуванням швидкої трансформації потреб і цілей в промисловості, виникає необхідність створення ефективних математичних моделей і технологій для оптимального розподілу ресурсів. Підвищення продуктивності та точності розв’язання задач з великою розмірністю вхідних даних сприятиме розширенню можливостей інформаційних систем. Це, у свою чергу, дозволить впроваджувати автоматизовані підходи до управління ресурсами в компаніях та організаціях, де раніше такі рішення були недоступні через недостатню продуктивність існуючих технологій.

Вступ

Розподіл обмежених ресурсів є класом оптимізаційних задач, спрямованих на знаходження точки максимального або мінімального значення цільової функції в межах визначеного обмеженнями простору. Ці задачі мають ключове значення для управління складними системами. Виділення таких задач у самостійну наукову дисципліну — "Дослідження операцій" — було здійснено ще у 1951 році[3], що заклало фундамент для подальших досліджень у сфері математичного моделювання планування, логістики та розподілу ресурсів.

Сучасні підходи до оптимізації

1. Жадібні алгоритми, метод гілок і границь.
2. Методи цілочисельного лінійного та нелінійного програмування.
3. Методи динамічного програмування.
4. Алгоритми машинного навчання та навчання з підкріпленням.

Запропонований підхід

Ключовим обмеженням існуючих точних підходів є недостатня масштабованість для застосування в BigData системах. У цього явища є головна причина - експоненційний ріст часу розрахунку в залежності від вхідних даних. Спираючись на класичні математичні моделі[1] пропонується метод їх масштабування з ціллю зменшення кількості змінних, значення яких необхідно знайти.

Метод полягає в ітеративному розв’язку базової оптимізаційної моделі для виокремленої підмножини вхідних даних. Для управлінням розміром вхідних даних, що обробляються на кожній ітерації вводиться параметр BATCH\_SIZE.

Кроки алгоритму:

1. Відповідно до значення введеного параметру BATCH\_SIZE вибирається кількість замовлень з вхідного масиву
2. Формується модель програмування обмеженнями[4], для не розподілених замовлень вносяться змінні, для уже розподілених - константи
3. Знаходиться рішення доступними алгоритмами розв’язку цілочисельних лінійних задач, до прикладу Or-Tools
4. Знайдені значення змінних оптимального рішення зберігаються в окрему структуру
5. Якщо є нерозподілені замовлення повертаємось на крок 1.

Запропонований метод поєднує в собі переваги жадібного алгоритму і точного підходу[2]. Це дозволяє розв’язувати задачі великої розмірності за прийнятний час спираючись при цьому на критерії оптимізації, а не виключно на евристику. Збереження переваги в гнучкості моделювання і висока продуктивність відкривають можливість до застосування даного підходу в BigData системах. Крім того метод подібний за способом застосування і перевагами з звичним для цього типу систем підходом пакетної обробки даних.

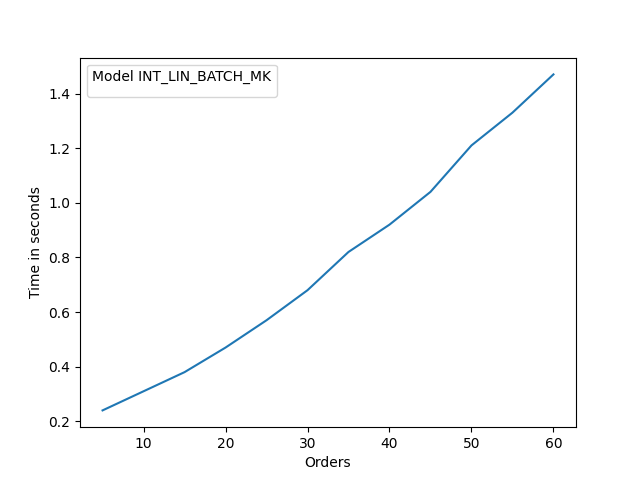
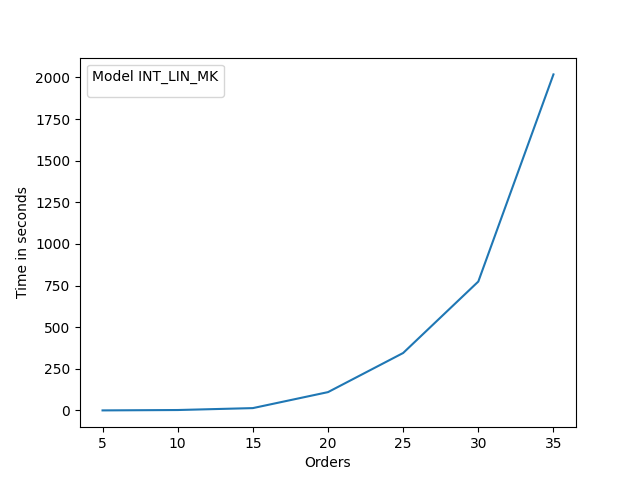


Рис. 1– Порівняно продуктивності базового методу і запропонованого

Часу розрахунку зростає лінійно в залежності від розміру вхідних даних що продемонстровано графіком. Демонструється прискорення понад 2000 раз відносно базового підходу для вхідних даних розмірністю в N = 35 замовлень і BATCH\_SIZE = 5, що пояснюється експоненційною складністю алгоритму застосованого на кроці 3. В описаному випадку базовий алгоритм викликається 7 раз , тоді як в базовому підході 1, але для всіх вхідних даних відразу. Негативною стороною є неможливість гарантувати оптимальність рішення, але ступінь відхилення оптимуму буде меншим ніж для жадібних алгоритмів.

Література

1. **Applegate D., Cook W.** A Computational Study of the Job-Shop Scheduling Problem // ORSA Journal on Computing. – 1991. – № 3(2). – С. 149–156. DOI: 10.1287/ijoc.3.2.149.
2. **Even G., Naor J. S., Rao S., Schieber B.** Divide-and-conquer approximation algorithms via spreading metrics // Journal of the ACM. – 2000. – № 47(4). – С. 585–616. DOI: [10.1145/347476.347478](https://doi.org/10.1145/347476.347478).
3. **Morse P. M., Kimball G. E.** Methods of Operations Research. – New York: Technology Press of MIT and John Wiley & Sons, 1951.
4. **Pysarchuk O., Baran D., Mironov Y., Pysarchuk I.** Algorithms of statistical anomalies clearing for data science applications // System Research and Information Technologies. – 2023. – № 1. – С. 78–84. DOI: 10.20535/SRIT.2308-8893.2023.1.06
5. **Wallace M.** Practical applications of constraint programming // Constraints. – 1996. – № 1. – С. 139–168. DOI: [10.1007/BF00143881](https://doi.org/10.1007/BF00143881).