UDK 656.078:004.7

Zora I.

PARALLEL OPTIMIZATION OF A MULTIDIMENSIONAL TENSOR MODEL OF PASSENGER FLOW WITH FUZZY PARAMETERS

Vinnytsia National Technical University

Анотація

У роботі розглянуто проблему експоненційного зростання обчислювальної складності при моделюванні наповнення транспортного засобу на міських маршрутах із використанням нечітких множин. Показано, що дискретизація універсумів змінних призводить до формування декартового добутку множин на кожному кроці розрахунку. Проведено порівняння циклічного та рекурсивного підходів до реалізації алгоритму та обґрунтовано неефективність рекурсії при великій кількості викликів. Запропоновано використання паралельної декомпозиції обчислень за незалежними підмножинами значень. Отримані результати підтверджують доцільність застосування паралельних обчислень для задач транспортного моделювання з нечіткими параметрами.

Ключові слова: програмна інженерія, нечітка логіка, паралельна декомпозиція, транспортні технології.

Abstract

The paper examines the problem of exponential growth in computational complexity arising in the modeling of vehicle occupancy on urban routes using fuzzy sets. It is shown that discretization of variable universes leads to the formation of Cartesian products of sets at each calculation step. A comparison between iterative and recursive approaches to algorithm implementation is conducted, and the inefficiency of recursion under a large number of function calls is substantiated. The use of parallel decomposition of computations across independent subsets of values is proposed. The obtained results confirm the feasibility and effectiveness of applying parallel computing techniques to transport modeling tasks with fuzzy parameters.

Keywords: software engineering, fuzzy logic, parallel decomposition, transport technologies.

In the process of modeling the organization of passenger transportation on urban routes, the calculation of vehicle occupancy at each stop can be performed using the following recurrent relation:

$$H_k = H_{k-1} + P_k - V_k,$$

where H_k is the number of passengers after the k-th stop, P_k is the number of passengers boarding, and V_k is the number of passengers alighting [1].

In the absence of objective input data required for calculations, fuzzy sets may be applied. The use of fuzzy sets leads to exponential growth in the volume of computations. If each of the variables H, P, and V is represented by a discretized universe containing four possible values, then the transition to the next stop is formed as a Cartesian product of the corresponding sets. The number of combinations at each step is therefore:

$$| H_k | = | H_{k-1} | \cdot 4 \cdot 4 = 16 \, | H_{k-1} |.$$

Given an initial universe size of $| H_0 | = 4$, after n stops:

$$| H_n | = 4 \cdot 16^n.$$

For 10 stops:

$$| H_{10} | = 4 \cdot 16^{10} \approx 4.4 \times 10^{12}.$$

Thus, the number of possible values grows exponentially with respect to the number of stops and proportionally to the dimensionality of the fuzzy universes.

The total number of computational operations over 10 steps is approximately:

$$\sum_{k=1}^{10} 16^k \approx 1.17 \times 10^{13}.$$

Even assuming modern processor performance at the level of 10^9simple arithmetic operations per second, the estimated execution time would be [2]:

$$\frac{1.17 \times 10^{13}}{10^9} \approx 11{,}700 \textit{ seconds} \approx 3.25 \textit{ hours}.$$

This estimation accounts only for arithmetic operations and does not include overhead associated with memory management, array allocation, and data structure handling, which further increases actual execution time.

Traditionally, such computations are implemented either using iterative (nested loop) methods or recursively (through traversal of a combination tree). The recursive approach is methodologically convenient because it naturally reflects branching during transitions between stops. However, each recursive call involves stack frame creation, parameter passing, and additional processor-level overhead. When dealing with billions or trillions of combinations, even an overhead of 20–50 nanoseconds per function call results in additional minutes or hours of execution time. Therefore, the recursive method is inefficient in problems characterized by exponential growth in the number of states.

The iterative method (nested or sequential loops) eliminates function call overhead and ensures better memory locality. However, it remains fully sequential. For the example with 10 stops and 4 values in each universe, the number of arithmetic operations exceeds 10^{13}, which at a performance of 10^9operations per second results in more than 3 hours of computation. Consequently, even an optimized sequential implementation requires significant execution time.

Analysis of the formation of the value set shows that computations for different elements of the set $H_{k-1}$are independent. For each h_i, 16 new values of the form $h_i + p_j - v_l$ are generated. This means that the computation can be decomposed into independent subtasks that can be executed in parallel.

Considering synchronization and scheduling overhead, the realistic acceleration coefficient is approximately 0.7–0.85 of the number of processor cores. Suppose the system has 8 physical processor cores. The theoretical maximum speedup under ideal load balancing is 8x, while the practical speedup is approximately 5–7x.

Thus, the estimated execution time for the considered example can be expressed as [3]:

$$T_{parallel} \approx \frac{T_{sequential}}{6} \approx \frac{3{,}25 \text{ hours}}{6} \approx 30\text{–}35 \text{ minutes}.$$

For systems with 16 cores, acceleration of 10–12x is possible, reducing execution time to 15–20 minutes.

Therefore, in the presence of exponential growth in the number of combinations, optimization should focus not only on algorithmic improvements but also on utilizing the capabilities of modern multicore processors. Decomposing tensor computations into independent parallel streams reduces the sequential iterative workload and provides a multiplicative reduction in processing time, which is particularly important for transport modeling tasks with fuzzy parameters and large combinatorial datasets.

REFERENCES

1. Zora, I., & Khoshaba, O. (2025). Use of fuzzy sets in calculating the passenger capacity utilisation rate in conditions where it is impossible to collect objective data. *Information Technologies and Computer Engineering*, 22(1), 115-123. https://doi.org/10.63341/vitce/1.2025.115

2. Williams, S., Waterman, A., & Patterson, D. 2009. Roofline: an insightful visual performance model for multicore architectures. *Commun. ACM* 52, 4 (April 2009), 65–76. https://doi.org/10.1145/1498765.1498785

3. Hennessy, J., & Patterson, D. Computer Architecture: A Quantitative Approach. 6th ed. Morgan Kaufmann, 2019., 1527.

Зьора Іван Євгенійович - аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: ivan.zora@gmail.com.

Науковий керівник: ***Хошаба Олександр Мирославович*** - к.т.н, доцент, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця.

Ivan Zora - Postgraduate Student of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: ivan.zora@gmail.com.

Research supervisor: ***Oleksandr Khoshaba*** - Ph.D., Associate Professor, Associate Professor of the Department of Software Engineering, Vinnytsia National Technical University, Vinnytsia.