The two-dimensional closest neighbor search problem solution using the cellular automata with locators [1]

D. I. Vasilev[2]

This article describes a cellular automaton with locators that solves the problem of finding the nearest neighbour. The problem is to find from a finite set of points the one closest to a predetermined "central"point. In contrast to the classical model of a cellular automaton, in the model under consideration, instantaneous transmission of signals through the ether at an arbitrary distance is allowed. It is shown that this possibility makes it possible to solve the problem in constant time, which is strikingly different from the one-dimensional case, where a logarithmic lower complexity estimate by the minimal distance is obtained.

Keywords: cellular automata, homogeneous structures, the closest neighbour search problem.

A *cellular automaton with locators* is a 8-tuple

$$\sigma = (\mathbb{Z}^k, E_n, V, E_q, +, L, \varphi, \psi)$$

where $\mathbb{Z}^k$ is the set of k-dimensional vectors with integer coordinates, $E_n = \{0, 1, \dots, n-1\}$, $V = (\alpha_1, \dots, \alpha_{h-1})$ is an ordered set of pairwise different nonzero vectors from $\mathbb{Z}^k$, $E_q = \{0, 1, \dots, q-1\}$, $+$ is a commutative semigroup operation defined on E_q, $L = (\nu_1, \dots, \nu_m)$ is an ordered set of pairwise different solid angles in $\mathbb{R}^k$ with a vertex at the origin, $\varphi : E_n^h \times E_q^m \to E_n$ is a function depending on the variables $x_0, x_1, \dots, x_{h-1}, z_1, \dots, z_m$ such that $\varphi(0, \dots, 0) = 0$, $\psi : E_n^h \times E_q^m \to E_q$ is a function that depends on the variables $x_0, x_1, \dots, x_{h-1}, z_1, \dots, z_m$. Here the variables $x_0, x_1, \dots, x_{h-1}$ take values from E_n and the variables $z_1, \dots, z_m$ take values from E_q. Elements of the set $\mathbb{Z}^k$ are called *cells* of the cellular automaton σ; elements of the set E_n are called *cell states* of the cellular automaton σ; the set V is called the *neighborhood pattern* of the cellular automaton σ; elements of the set E_q are called *broadcasting signals*; the set L is called the *locator pattern* of the cellular automaton σ; the function φ is called the *local transition function* of the automaton σ; the function ψ is called the *broadcasting function* of

---

[1] Originally published in *Intellektualnye Sistemy. Teoriya i prilogeniya* (2021) **25**, No. 4, 83-87 (in Russian).

[2] *Vasilev Denis Igorevich* — junior researcher, Lomonosov Moscow State University, Faculty of Mechanics and Mathematics, Chair of Mathematical Theory of Intelligent Systems, email: denis.vasilev.igor@gmail.com

the automaton σ. The state 0 is interpreted as *rest state* and the condition $\varphi(0, \dots, 0) = 0$ is interpreted as a condition for maintaining the rest state.

This definition was introduced by Gasanov E.E. [1] and improved by Kalachev G.V. [2].

Let's formulate the closest neighbour search problem on the line. Let the I be the initial state of a cellular automaton on $\mathbb{Z}^1$ which satisfies the following conditions: a) Any cell is on one of $\{q_S; q_{C_0}, *\}$ states; b) There is only one q_{C_0} cell; c) There is a finite and non-empty set of q_S cells.

We will define that a cellular automaton state I' is solution for the problem I if I' satisfies the following conditions: a) The q_{C_0} cell from I is in q_{CF} state in I'; b) The cell which is the closest to the q_{C_0} cell in I is in the q_{SE} state. If there are several closest cells then an arbitrary one must be chosen; c) The rest cells are in $*$ state.

We define that cellular automaton σ solves the closest neighbour search problem if it satisfies the following conditions: a) If the initial state I of the cellular automaton is a closest neighbour search problem then the automaton must end up in I' state which is solution for I; b) If the automaton takes state S which is solution for some closest neighbour search problem this state must be kept for all the next tacts.

Let's call *the general position of the closest neighbour search problem* a problem in which there is at least one cell in the state q_S on both sides of the cell in the state q_{C_0}.

In the article [3], the theorem was proved for the one-dimensional case:

Theorem 1. *There is a cellular automaton σ with 25 states and with the power of the broadcasting alphabet 12, which solves the problem of finding the closest neighbour in a time not exceeding $\log_2 s + 7$, where s is the distance from the central cell with the initial state q_{C_0} to its nearest neighbour with the initial state q_S.*

An article with a similar lower complexity estimate was sent to the editorial office of the Vestnik of the MSU journal:

Theorem 2. *For any cellular automaton with locators σ with the power of the broadcasting alphabet M and any general position of the nearest neighbour search problem I, $T_I^\sigma > \log_M(\frac{s}{5})$ is performed, where s is the distance from the cell in the state q_{C_0} to the nearest cell in the state q_S in the problem I, and T_I^σ is the number of the clock cycles for which the automaton σ solves the problem I.*

Thus, for the one-dimensional case of the nearest neighbour search problem, the order of complexity of the problem is obtained.

It turned out that for the dimension $n \geq 2$, similar estimates are incorrect, since for such problems it was possible to build an automaton with locators that solves them in constant time. Here is an example of such an automaton for the case of $n = 2$:

Theorem 3. *There is a cellular automaton σ with 15 states and with the power of the broadcasting alphabet* 40*, which solves the two-dimensional problem of finding the nearest neighbour in a time not exceeding* 13.

Consider a cellular automaton with locators σ with a set of locators consisting of locators from Fig. 1 and one expanded locator that reads the sum of the signals of all cells of the cellular space. Let's define the broadcasting alphabet as a subset $\{0;1\}^{20}$ and a semigroup operation of a component-by-component maximum on it. For convenience, we will denote the cell signals by one or more numbers - the numbers of non-zero positions in the ether signal. So the signal $(0,1,0,1,0,...,0,0,0)$ we will record as a pair of signals 2 and 4.

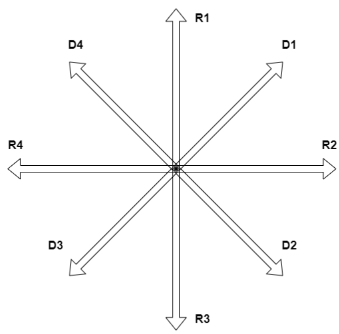

Fig. 1. Locators layout and names

Let's define a coordinate system on the cellular space with the center in the central cell. The central cell in the constructed automaton constantly sends a 1 signal to the ether. Cells that receive such a signal from the $R3$ locator will realize that they are on the upper coordinate semi-axis. Similarly, each cell can identify its location on the other three coordinate semi-axes. The cells located on the semi-axes constantly send a signal to the ether with the number of their semi-axis (upper — 2, right —3, lower —4 and left — 5). By these signals, each cell can recognize in which coordinate quarter it is located. For example, having received the signal 2 from the locator $R4$ and the signal 3 from the locator $R3$, it is possible to uniquely determine that the cell in question is in the first coordinate quarter. The idea of functioning of the constructed automaton is to project along the Manhattan circle all the points from the problem on one semi-axis, find the projection closest to the center, and then restore its prototype. For example, a point from the first quarter can send a special signal that is read only by the right semi-axis. A point from the right semi-axis, having received such a signal from the locator

$D4$, will understand that it is a projection of one of the points of the problem (Fig. 2 on the left). By repeating this iteration 4 times, you can project all the points onto the upper semi-axis (Fig. 2 on the right).

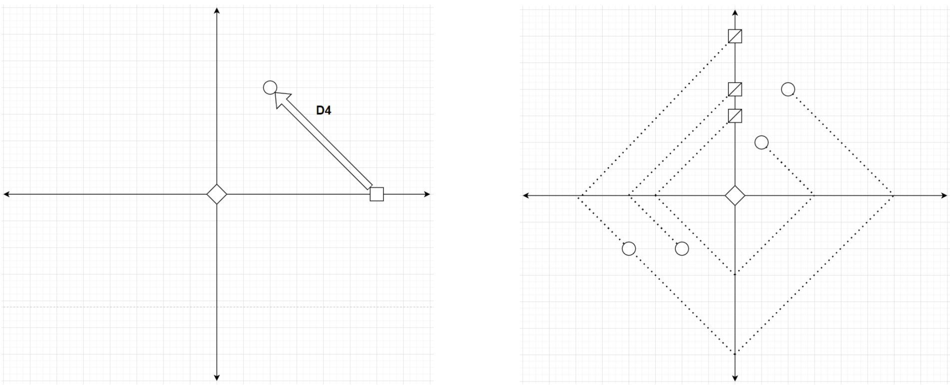

Fig. 2. On the left is an example of projecting a single point onto a semi-axis. On the right, the progress of projecting the problem onto the upper semi-axis.

To find the nearest neighbour on the upper semi-axis, it is enough for each candidate to broadcast a special signal, and after receiving such a signal from the $R3$ locator, he will withdraw (i.e., switch to the default state). After the nearest projection is found, it is enough to restore its prototype by the reverse course of the described algorithm.

References

[1] Gasanov E. E., "Cellular automata with locators", *Intelligent systems. Theory and Applications*, **20**:2 (2020), 121–133 (In Russian).

[2] Kalachev G. V., "Remarks on the definition of a cellular automaton with locators", *Intelligent systems. Theory and Applications*, **24**:4 (2020), 47–57 (In Russian).

[3] Vasilev D. I., "The closest neighbour problem solution using the cellular automata with locators model", *Intelligent systems. Theory and Applications*, **24**:3 (2020), 99–120 (In Russian).