

Research Article

Solution to Shortest Path Problem Using a Connective Probe Machine

Jiuyun Sun, Huanhe Dong , Yuan Kong , and Yong Fang 

College of Mathematics and Systems Science, Shandong University of Science and Technology, Qingdao 266590, China

Correspondence should be addressed to Yuan Kong; kongyuan@sdust.edu.cn and Yong Fang; fangyong@sdust.edu.cn

Received 23 June 2019; Revised 16 September 2019; Accepted 22 October 2019; Published 7 November 2019

Academic Editor: Haopeng Zhang

Copyright © 2019 Jiuyun Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the continuous urban scale expansion, traffic networks have become extremely complex. Finding an optimal route in the shortest time has become a difficult and important issue in traffic engineering study. In this study, a novel computing model, namely, probe machine, is used to solve this problem. Similar to previous studies, urban transport networks can be abstracted into maps, in which points representing places of origin, destinations, and other buildings constitute the data library and edges representing the road make up the probe library. The true solution can be obtained after one probe operation on the computing platform. And by comparing the solving process with Dijkstra's and Floyd's algorithms, the computing efficiency of the probe machine is clearly superior, although all three methods can solve the shortest path problem and obtain the same solution.

1. Introduction

Traffic engineering is the basic theory of the research and development of traffic engineering discipline. Its main purpose is to seek the transportation system planning, construction and management scheme with the largest travel efficiency, the least traffic accidents, the fastest traffic speed, the least transportation cost, and the lowest energy consumption. The rapid development of urban traffic and increase in the number of vehicles has made travelling more convenient for people in recent years. However, urban traffic is not smooth due to the numerous traffic network nodes and complex sections; some cities suffer from traffic congestion [1, 2]. Finding a suitable path for drivers is not only one of the solutions to promote the development of urban transportation [3–5], but also one of the main purposes in traffic engineering research. Therefore, we must address this issue by determining the shortest path. In traffic engineering, “shortest path” does not necessarily mean the shortest distance, but also the shortest time or the lowest cost.

The shortest path problem is a classical problem in graph theory, which has been applied in many fields [6]. Finding the path with the shortest distance is the most basic application of the shortest path problem, which is also a very

practical problem. The basic objective of the shortest path problem is to find the path, with the lowest weight, between two points where every edge in the graph has its own weight value. Many algorithms to solve the shortest path problem have been proposed in previous studies, such as Dijkstra's algorithm [7], Bellman–Ford algorithm [8], and Floyd's algorithm [9]. The most classical algorithm for solving such a problem is Dijkstra's algorithm. It can solve the shortest path problem from a given point to any point; however, it fails to solve the shortest path problem with a negative weight. Therefore, Floyd's algorithm and other algorithms were proposed to solve the above problem.

In 2016, Xu proposed a new computing model, i.e., the probe machine [10]. Probe machine is a completely parallel computing model that can simultaneously process multiple pairs of data. Parallel computing can speed up operations, expand the scale of processing problems, and facilitate the ability of algorithms to handle problems [11, 12]. Compared with Turing machines, probe machines can solve NP-complete problems after a probe operation, such as the Hamiltonian cycle problem [10], vertex problem [10], travelling salesman problem [13], working operation problem [14], and so on. The probes of a probe machine can be divided into connective probes and transitive probes. A connective probe can connect

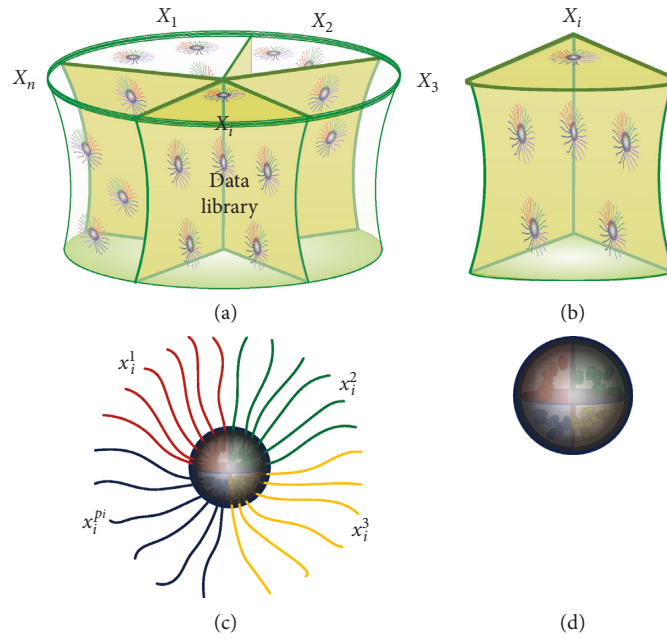


FIGURE 1: Data library [10].

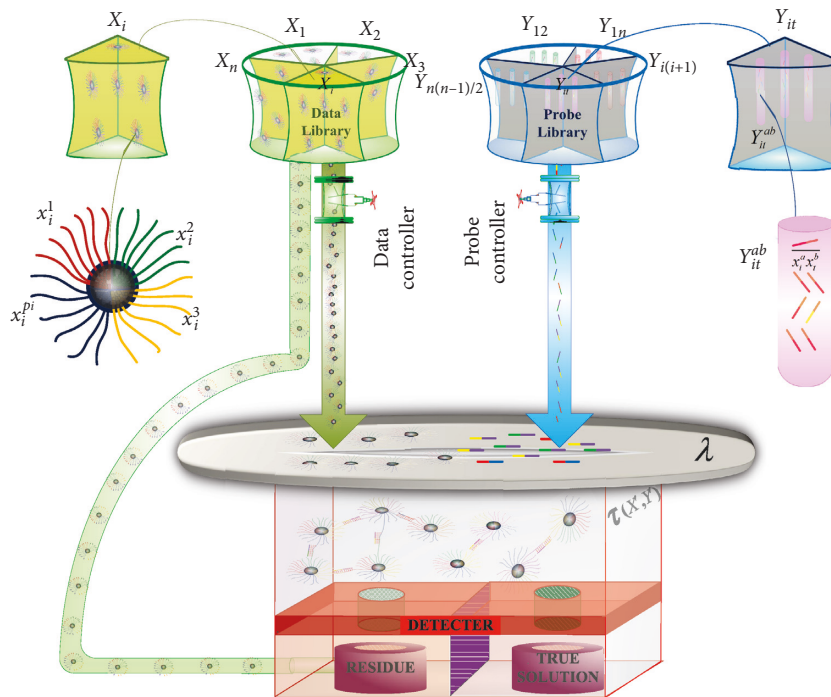


FIGURE 2: Connective probe machine [10].

Step 5: taking v_6 as the intermediary point and updating the matrix,

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 3 & 1 & 7 & 11 & 10 & 13 & 9 \\ \infty & 0 & \infty & 7 & 1 & 5 & 4 & 7 & 3 \\ \infty & 2 & 0 & 2 & 3 & 7 & 6 & 9 & 5 \\ \infty & \infty & \infty & 0 & \infty & 10 & 12 & \infty & \infty \\ \infty & \infty & \infty & 6 & 0 & 4 & 3 & 6 & 2 \\ \infty & \infty & \infty & \infty & \infty & 0 & 2 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 3 & 0 \end{pmatrix}. \quad (6)$$

Step 6: taking v_7 as the intermediary point and updating the matrix,

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 3 & 1 & 7 & 11 & 10 & 13 & 9 \\ \infty & 0 & \infty & 7 & 1 & 5 & 4 & 7 & 3 \\ \infty & 2 & 0 & 2 & 3 & 7 & 6 & 9 & 5 \\ \infty & \infty & \infty & 0 & \infty & 10 & 12 & 16 & \infty \\ \infty & \infty & \infty & 6 & 0 & 4 & 3 & 6 & 2 \\ \infty & \infty & \infty & \infty & \infty & 0 & 2 & 6 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 3 & 0 \end{pmatrix}. \quad (7)$$

Step 7: taking v_9 as the intermediary point and updating the matrix,

$$\mathbf{A} = \begin{pmatrix} 0 & 5 & 3 & 1 & 7 & 11 & 10 & 12 & 9 \\ \infty & 0 & \infty & 7 & 1 & 5 & 4 & 6 & 3 \\ \infty & 2 & 0 & 2 & 3 & 7 & 6 & 8 & 5 \\ \infty & \infty & \infty & 0 & \infty & 10 & 12 & 16 & \infty \\ \infty & \infty & \infty & 6 & 0 & 4 & 3 & 5 & 2 \\ \infty & \infty & \infty & \infty & \infty & 0 & 2 & 6 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 3 & 0 \end{pmatrix}. \quad (8)$$

Thus, the shortest path from v_1 to v_8 is $v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_9 \rightarrow v_8$.

3.3. Solving the Shortest Path Problem Using the Probe Machine. The probe machine solves the shortest path problem as follows.

3.3.1. Data Library Construction

$$X = \bigcup_{i=1}^9 E^2(v_i). \quad (9)$$

$E^2(v_i)$ is the data pool of v_i , and $E^2(v_i)$ is a set of two long paths centered around v_i . The two long paths are denoted as v_{ilj} , and i, l, j are different from each other. At the same time, because this problem is a directed graph, each path has one data fiber. The data library is as follows:

$$\begin{aligned} X &= \bigcup_{i=1}^9 E^2(v_i), \\ E^2(v_1) &= \emptyset, \\ E^2(v_2) &= \{x_{215}, x_{235}\}, \\ E^2(v_3) &= \{x_{312}, x_{314}\}, \\ E^2(v_4) &= \{x_{416}, x_{436}, x_{456}\}, \\ E^2(v_5) &= \{x_{524}, x_{526}, x_{527}, x_{528}, x_{529}\}, \\ E^2(v_6) &= \{x_{647}, x_{657}\}, \\ E^2(v_7) &= \{x_{768}, x_{758}\}, \\ E^2(v_8) &= \emptyset, \\ E^2(v_9) &= \{x_{958}\}. \end{aligned} \quad (10)$$

The data fibers are as follows:

$$\begin{aligned} \mathfrak{F}(x_{215}) &= x_{215}^5, \\ \mathfrak{F}(x_{235}) &= x_{235}^4, \\ \mathfrak{F}(x_{312}) &= x_{312}^2, \\ \mathfrak{F}(x_{314}) &= x_{314}^4, \\ \mathfrak{F}(x_{416}) &= x_{416}^6, \\ \mathfrak{F}(x_{436}) &= x_{436}^6, \\ \mathfrak{F}(x_{456}) &= x_{456}^6, \\ \mathfrak{F}(x_{524}) &= x_{524}^4, \\ \mathfrak{F}(x_{526}) &= x_{526}^6, \\ \mathfrak{F}(x_{527}) &= x_{527}^7, \\ \mathfrak{F}(x_{528}) &= x_{528}^8, \\ \mathfrak{F}(x_{529}) &= x_{529}^9, \\ \mathfrak{F}(x_{647}) &= x_{647}^7, \\ \mathfrak{F}(x_{657}) &= x_{657}^7, \\ \mathfrak{F}(x_{768}) &= x_{768}^8, \\ \mathfrak{F}(x_{758}) &= x_{758}^8, \\ \mathfrak{F}(x_{958}) &= x_{958}^8. \end{aligned} \quad (11)$$

3.3.2. Probe Library Construction. In order to avoid arbitrary data aggregation, if there is a probe between data v_{ilj} and v_{tab} , one of the following two conditions need to be met:

$$\begin{aligned}
(1) \quad & |\{i, l, j\} \cap \{t, a, b\}| = |\{l, j\} \cap \{a, b\}| = 1, \\
(2) \quad & t \in \{l, j\}, i \in \{a, b\}, |\{l, j\} \cap \{a, b\}| = 0.
\end{aligned} \tag{12}$$

At the same time, without loss of generality, because this is a directed graph, the direction of the probe needs to be same as the direction of the edge. The probe library established is as follows:

$$\begin{aligned}
Y &= \bigcup_{i=1, t=2}^{i=8, t=9} Y_{it}, \quad (i < t), \\
Y_{23} &= \{\overline{x_{235}^5 x_{312}^2}\}, \\
Y_{24} &= \{\overline{x_{215}^5 x_{456}^6}, \overline{x_{235}^5 x_{456}^6}\}, \\
Y_{25} &= \{\overline{x_{215}^5 x_{524}^4}, \overline{x_{215}^5 x_{526}^6}, \overline{x_{215}^5 x_{527}^7}, \overline{x_{215}^5 x_{528}^8}, \overline{x_{215}^5 x_{529}^9}, \overline{x_{235}^5 x_{524}^4}, \overline{x_{235}^5 x_{526}^6}, \overline{x_{235}^5 x_{527}^7}, \overline{x_{235}^5 x_{528}^8}, \overline{x_{235}^5 x_{529}^9}\}, \\
Y_{26} &= \{\overline{x_{215}^5 x_{657}^7}, \overline{x_{235}^5 x_{657}^7}\}, \\
Y_{27} &= \{\overline{x_{215}^5 x_{758}^8}, \overline{x_{235}^5 x_{758}^8}\}, \\
Y_{29} &= \{\overline{x_{215}^5 x_{958}^8}, \overline{x_{235}^5 x_{958}^8}\}, \\
Y_{34} &= \{\overline{x_{314}^4 x_{436}^6}\}, \\
Y_{35} &= \{\overline{x_{312}^2 x_{524}^4}, \overline{x_{312}^2 x_{526}^6}, \overline{x_{312}^2 x_{527}^7}, \overline{x_{312}^2 x_{528}^8}, \overline{x_{312}^2 x_{529}^9}\}, \\
Y_{36} &= \{\overline{x_{314}^4 x_{657}^7}\}, \\
Y_{45} &= \{\overline{x_{456}^6 x_{524}^4}\}, \\
Y_{46} &= \{\overline{x_{416}^6 x_{647}^7}, \overline{x_{436}^6 x_{647}^7}, \overline{x_{456}^6 x_{647}^7}\}, \\
Y_{47} &= \{\overline{x_{416}^6 x_{768}^8}, \overline{x_{436}^6 x_{768}^8}, \overline{x_{456}^6 x_{768}^8}\}, \\
Y_{56} &= \{\overline{x_{524}^4 x_{647}^7}, \overline{x_{526}^6 x_{657}^7}\}, \\
Y_{57} &= \{\overline{x_{526}^6 x_{768}^8}, \overline{x_{527}^7 x_{758}^8}\}, \\
Y_{59} &= \{\overline{x_{529}^9 x_{958}^8}\}, \\
Y_{67} &= \{\overline{x_{647}^7 x_{768}^8}, \overline{x_{657}^7 x_{768}^8}\}.
\end{aligned} \tag{13}$$

3.3.3. Probe Operation. The data controller and probe controller are used to extract the corresponding amount of data and probes from the data library and probe library and add them to the computing platform. After the probe operation, all possible solutions to the problem are obtained. Through the detector, the optimal solution is output as follows:

$$v_1 \longrightarrow v_3 \longrightarrow v_2 \longrightarrow v_5 \longrightarrow v_9 \longrightarrow v_8. \tag{14}$$

4. Results and Discussion

Although all three methods can solve the shortest path problem and obtain the same solution, there are some obvious differences in the solving process and computing efficiency. Dijkstra's algorithm is based on the breadth-first search algorithm, which calculates the shortest path from the starting point to all other points. The calculation step of Dijkstra's algorithm is to expand from the starting point to the layer by layer. Floyd's algorithm is based on dynamic programming algorithm to find the shortest path between any two points in the graph. The first step of Floyd's

algorithm is to find the adjacency matrix of the graph, and the second step is to update the adjacency matrix by taking each point as the intermediate point. And, the solution process of the probe machine is mainly divided into two steps: the first step is to build the data library and the probe library, and the second step is to get all possible solutions through the probe operation. It can be seen from the solution process that each step of Dijkstra's algorithm and the Floyd algorithm is part of solving the shortest path problem, but the real solution process of the probe machine is only probe operation. The purpose of the remaining steps of the probe machine is to prepare for the probe operation or to screen the solution calculated by the probe operation. This phenomenon should be attributed to the way of storing data. When the Dijkstra algorithm and the Floyd algorithm are used, the linear data placement mode and the sequential data processing mode are taken, so the solution process needs to be performed step by step; however, in the probe machine, a nonlinear data storage method is adopted, so the probe machine can simultaneously process multiple pairs of data. In other words, the probe machine is a completely parallel

computational model, which is the biggest difference between the probe machine and other computing models.

Moreover, Dijkstra's and Floyd's algorithms are only used to deal with problems in graph theory. The probe machine is a completely parallel computing model, which can also get the true solution after one probe operation when solving NP-complete problems. Although we only use the probe machine to deal with the shortest path problem in this paper, we can see the powerful computing power of the probe machine through the solving process.

5. Conclusion

In urban traffic planning, the shortest path problem is a very basic problem. In this study, a probe machine was used to solve this problem. It can solve complex traffic problems after a single probe operation, which improves the efficiency and shortens the time required for solving traffic planning problems. When dealing with actual traffic problems, the speed at which a probe machine solves them is much higher than the speed of other models. A probe machine has high computing power; however, it is still a new concept. It has several shortcomings and needs to be improved further to handle specific problems. For example, when dealing with complex traffic problems, complex data and probe libraries need to be constructed. The process of constructing these libraries cannot be completely replaced by a probe machine or computer. The more complex the problem to be solved is, the more difficult it becomes to construct data library and probe library. In the future, we will continue to study the probe machine and attempt to improve it.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant nos. 61602188 and 11975143), Shandong Provincial Natural Science Foundation (grant no. ZR2019QD018), and Scientific Research Foundation of Shandong University of Science and Technology for Recruited Talents (grant nos. 2017RCJJ068 and 2017RCJJ069).

References

- [1] A. Lozano and G. Storchi, "Shortest viable path algorithm in multimodal networks," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 3, pp. 225–241, 2001.
- [2] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Annals of Operations Research*, vol. 89, pp. 319–328, 1999.
- [3] L. Feng, Z. Chenghu, and W. Qing, "An optimum vehicular path algorithm for traffic network based on hierarchical spatial reasoning," *Geo-spatial Information Science*, vol. 3, no. 4, pp. 36–42, 2000.
- [4] K. Zhou, S.-W. He, R. Song, and L.-Y. Cheng, "Multiobjective optimization method of public transit networks based on travel behavior," *Journal of Highway and Transportation Research and Development (English Edition)*, vol. 9, no. 4, pp. 71–77, 2015.
- [5] L. Adacher, G. Oliva, and F. Pascucci, "Decentralized route guidance architectures with user preferences in urban transportation networks," *Procedia-Social and Behavioral Sciences*, vol. 111, pp. 1054–1062, 2014.
- [6] S. Pallottino, "Shortest-path methods: complexity, interrelations and new propositions," *Networks*, vol. 14, no. 2, pp. 257–267, 1984.
- [7] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [8] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.
- [9] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.
- [10] J. Xu, "Probe machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1405–1416, 2016.
- [11] C.-Y. Han, F.-Y. Zheng, T.-D. Guo, and G.-P. He, "Parallel algorithms for large-scale linearly constrained minimization problem," *Acta Mathematicae Applicatae Sinica, English Series*, vol. 30, no. 3, pp. 707–720, 2014.
- [12] B. Y. Guo, H. H. Dong, and Y. Fang, "Lump solutions and interaction solutions for the dimensionally reduced nonlinear evolution equation," *Complexity*, vol. 2019, Article ID 5765061, 9 pages, 2019.
- [13] S. Sha and Y. Chen, "Probe model for solving travelling salesman problem," *Journal of Huaibei Normal University (Natural Science)*, vol. 37, pp. 36–39, 2016.
- [14] J. Yang and Z. Yin, "The working operation problem based on probe machine model," in *Bio-Inspired Computing: Theories and Applications*, pp. 47–53, Springer, Singapore, 2016.
- [15] Y. Fang, H. Dong, Y. Hou, and Y. Kong, "Frobenius integrable decompositions of nonlinear evolution equations with modified term," *Applied Mathematics and Computation*, vol. 226, pp. 435–440, 2014.
- [16] Y. Lin, H. Dong, and Y. Fang, "N-soliton solutions for the NLS-like equation and perturbation theory based on the riemann-Hilbert problem," *Symmetry*, vol. 11, no. 6, p. 826, 2019.
- [17] T.-O. Ishdorj, A. Leporati, L. Pan, X. Zeng, and X. Zhang, "Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources," *Theoretical Computer Science*, vol. 411, no. 25, pp. 2345–2358, 2010.
- [18] B. Song, T. Song, and L. Pan, "Time-free solution to SAT problem by P systems with active membranes and standard cell division rules," *Natural Computing*, vol. 14, no. 4, pp. 673–681, 2015.
- [19] B. Song, M. J. Pérez-Jiménez, and L. Pan, "An efficient time-free solution to SAT problem by P systems with proteins on membranes," *Journal of Computer and System Sciences*, vol. 82, no. 6, pp. 1090–1099, 2016.
- [20] Y. Kong and Y. Fang, "On behavior of the correction equations in Jacobi-Davidson method," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5169362, 4 pages, 2019.
- [21] B. Song and Y. Kong, "Solution to PSPACE-complete problem using P systems with active membranes with time-freeness,"

- Mathematical Problems in Engineering*, vol. 2019, Article ID 5793234, 8 pages, 2019.
- [22] W. Y. Szeto, Y. Wu, and S. C. Ho, "An artificial bee colony algorithm for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 215, no. 1, pp. 126–135, 2011.
 - [23] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei, "A hybrid genetic algorithm for multidepot and periodic vehicle routing problems," *Operations Research*, vol. 60, no. 3, pp. 611–624, 2012.
 - [24] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.
 - [25] Z. Shao and A. Vesel, "Modeling the packing coloring problem of graphs," *Applied Mathematical Modelling*, vol. 39, no. 13, pp. 3588–3595, 2015.
 - [26] L. A. Levin and R. Venkatesan, "An average case NP-complete graph colouring problem," *Combinatorics, Probability and Computing*, vol. 27, no. 5, pp. 808–828, 2018.
 - [27] J.-Y. Chen, D. M.-H. Chiang, and R.-S. Guo, "Partner selection model for design chain collaboration," *International Journal of Production Research*, vol. 51, no. 4, pp. 1131–1145, 2013.
 - [28] D. Bienstock and G. Muñoz, "LP formulations for polynomial optimization problems," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1121–1150, 2018.
 - [29] T. Song, S. Pang, S. Hao, A. Rodríguez-Patón, and P. Zheng, "A parallel image skeletonizing method using spiking neural P systems with weights," *Neural Processing Letters*, vol. 50, no. 2, pp. 1485–1502, 2018.
 - [30] T. Song, X. Zeng, P. Zheng, M. Jiang, and A. Rodriguez-Paton, "A parallel workflow pattern modeling using spiking neural P systems with colored spikes," *IEEE Transactions on Nano-Bioscience*, vol. 17, no. 4, pp. 474–484, 2018.
 - [31] T. Song, X. Wang, X. Li, and P. Zheng, "A programming triangular DNA origami for doxorubicin loading and delivering to target ovarian cancer cells," *Oncotarget*, 2018.
 - [32] G. K. Wu, Y. Q. Liang, and S. H. Xu, "Numerical computational modeling of random rough sea surface based on JONSWAP spectrum and Donelan directional function," *Concurrency and Computation: Practice and Experience*, p. e5514, 2019.
 - [33] D. Ghosh, A. Singh, K. K. Shukla, and K. Manchanda, "Extended Karush-Kuhn-Tucker condition for constrained interval optimization problems and its application in support vector machines," *Information Sciences*, vol. 504, pp. 276–292, 2019.

