Supplementary Notes for: Resonance algorithm: an intuitive algorithm to find all shortest paths between two nodes

Yu Liu[1,+,*], Qiguang Lin[2,+], Binbin Hong[3], Yunru Peng[1], Daniel Hjerpe[4], and Xiaofeng Liu[2,5,*]

[1]International Academic Center of Complex Systems, Beijing Normal University, Zhuhai, 519087, China
[2]College of IoT Engineering, Hohai University, Changzhou, 213022, China
[3]Institute of Microscale Optoelectronics, Shenzhen University, Shenzhen, 518060, China
[4]Ericsson AB, Kista 16483, Sweden
[5]Jiangsu Key laboratory of Special Robotic Technologies, Changzhou, 213022, China
[*]yu.ernest.liu@bnu.edu.cn; xfliu@hhu.edu.cn
[+]these authors contributed equally to this work

S1. Exemplified map: a general case

The example we gave in the main text is an undirected and positively-weighted map. To represent the most general case, here we show another map with mixed edges (namely, with both undirected and direct edges) and positively-weighted edges, irrespective of loops (as shown in Fig. S1). In this case, there is only one shortest path: A-S-Q-P-L-M-B, which is also worked out by the algorithm (to test, run the code attached, where the file exampleMap1.mat is this map).

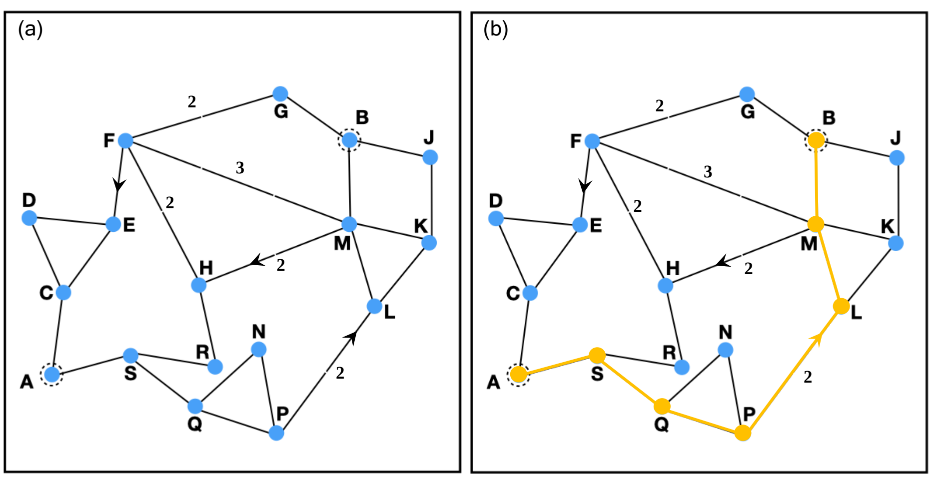

Figure S1. An exemplified map, to represent the most general case.

S2. Resonance algorithm's potential links to Fermat's principle and quantum mechanics

In RA, the process that each node sends signals to all of its neighbors when it receives one can be considered as that signals are experiencing every possible path. On the other hand, as long as the origin and the destination are determined, the shortest path (equivalently, the path that takes the least time) is automatically determined. Intriguingly, it seems like a metaphor of Fermat's principle[1] which says that the path taken by a ray (or other types of waves in general) is always the one that takes the least time.

Let us consider the gravity between Earth and the Moon. A naive way to think about it is that a gravitational field is first created by Earth; the Moon perceives this field around itself, and this surrounding gravitational field determines the Moon's movement. But, inspired by RA, we can also consider Earth and the Moon simultaneously. That is, they both create their own gravitational field that expands outwards in all directions at the speed of light; and when either field reaches the other object, the interaction forms. This mechanism guarantees that the interaction forms in the quickest way (indeed, natural interactions occur in the quickest way) since the gravitational fields are experiencing every possible path before the interaction forms, just as RA.

Now, from another perspective, let us consider the wave function of a free particle X in quantum mechanics[2], i.e., $\psi(\boldsymbol{r},t) = Ae^{i(\boldsymbol{k}\cdot\boldsymbol{r}-\omega t)}$ where $\boldsymbol{r}$ is X's position, t is the time, A is the amplitude of the wave function, $\boldsymbol{k}$ is the wave vector, namely the direction X travels, and ω is the angular frequency. We can see that the complex conjugate of $\psi(\boldsymbol{r},t)$ is $\psi^*(\boldsymbol{r},t) = Ae^{-i(\boldsymbol{k}\cdot\boldsymbol{r}-\omega t)} = Ae^{i(\boldsymbol{k}\cdot(-\boldsymbol{r})-\omega(-t))} = \psi(-\boldsymbol{r},-t)$ which is exactly the wave function of a free particle that travels in the opposite direction in reversed time (denoted this imaginary free particle as Y). Therefore, the absolute value of X's wave function can be written as (it is also the probability density that X is at $\boldsymbol{r}$)

$$|\psi(\boldsymbol{r},t)|^2 = \psi^*(\boldsymbol{r},t)\cdot\psi(\boldsymbol{r},t) = \psi(-\boldsymbol{r},-t)\cdot\psi(\boldsymbol{r},t)$$

which is the multiplication of Y's wave function and X's wave function, which is non-zero only when both of them are non-zero. This strongly resembles the "intersection" in RA where X refers to the forward process and Y refers to the time-reversed backward process. And yet, free particles indeed travel in the quickest way.

Are these arguments above suggesting that the resonance-algorithm-like process is underlying these natural phenomena? Indeed, these thoughts are by no means scientifically solid, but it is deserved to be said if it offers any inspiration, reflection, or chance of discussions.

S3. Extend Dijkstra's algorithm to give all possible shortest paths

The classic Dijkstra's algorithm only gives one shortest path, even if there are more equally-long shortest paths. Here we first illustrate this classic version and explain why it has this property; and then extend this classic version to give all possible shortest paths. Take the graph shown in Fig. S2 as an example, and look for shortest paths from node A to B. Initially, only the origin node A is in the set of "visited", denoted as set Ω. For each round, visited nodes' neighbors that are not included in Ω will be examined, and the one that has the shortest *tentative distance* will be added to Ω.

As shown in Fig. S2, in round i, node A's neighbors B, C, E, F and G are examined. We find that the distances from A to C, E and F are the smallest (namely, 1) but identical. Yet we should only put either C, E or F into Ω (usually the one with the smallest index, node C in this case). In round ii, A's and C's neighbors B, D, E, F and G are examined, and E enters Ω. In round iii, A's, C's and E's neighbors B, D, F and G are examined, and F enters Ω. Repeat, until all nodes are included in Ω; and then we have found the shortest path from the origin to any node. The reason why this algorithm gives only one shortest path is that the nodes in set Ω can only be accessed once. For example, after node D is included in Ω (accessed through node C), it cannot be accessed through node E. So, path A→E→D cannot be found, although it has the same length as A→C→D.

To solve this problem so that it gives all equally-long shortest paths, we extend the single parent node that is attached to each node, as in the classic algorithm, to a "parent nodes list"[3] (see codes at www.wuyichen.org/resalg). Furthermore, when a new node N enters Ω, we do not immediately go to the next round as before, but first examine each of N's neighbors that is the closest to N, denoted as node M (for convenience, denote M's current tentative distance as d, and the newly-calculated distance from the origin to M through N as d'): (1) if $d' < d$, M's tentative distance will be updated to d', and M's parent nodes list will be reset such that it only includes N; (2) if $d' = d$, N will be added to M's parent node list; (3) if $d' > d$, nothing happens.

For example, in round ii, when E enters Ω, we should then examine D (since D is closest to E among all of E's neighbors, namely D and B). Currently, C is D's only parent node. But now we find that the distance from the origin to D through E is equal to the current tentative distance of D (namely, the distance of the path A→C→D). So, we add node E into D's parent nodes list (which includes C and E now).

Continue this process until all nodes are included in Ω. At the end, all possible shortest paths can be obtained by recursively tracing the parent nodes list of each node.

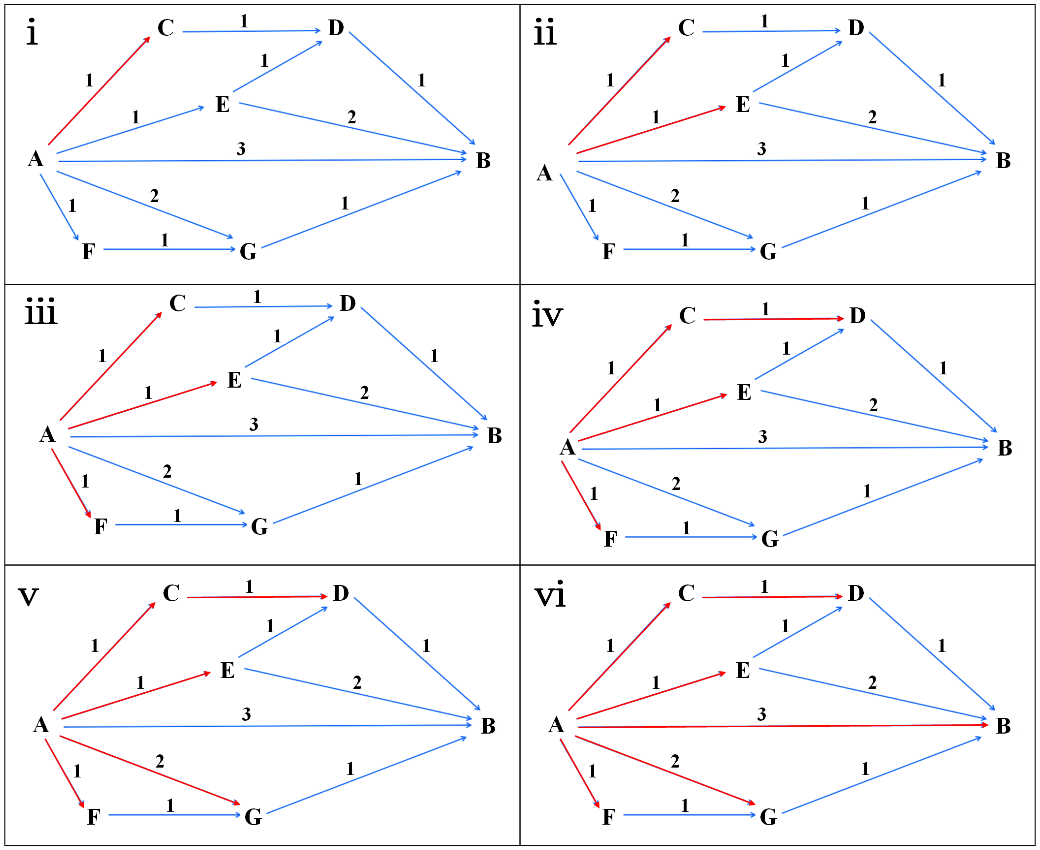

Figure S2. Illustration of the classic Dijkstra's algorithm.

S4. How the random maps are generated

The random graphs that are used to test the algorithms are generated as follows:

1. First generate n nodes marked from 1 to n.
2. Every potential connection between two nodes (n^2 connections in total) has 50 percent chance to be connected, and each connected edge is assigned with a random weight between 1 and 10.
3. Check if this generated graph is connected. If so, output this graph; Otherwise, discard it and repeat.

References

1. Ziggelaar, A. The sine law of refraction derived from the principle of Fermat—prior to Fermat? The theses of Wilhelm Boelmans S. J. in 1634. *Centaurus* **24**, 246–262 (1980).

2. Schlosshauer, M. Decoherence, the measurement problem, and interpretations of quantum mechanics. *Rev. Mod. Phys.* **76**, 1267–1305 (2005).

3. Wang, S. & Li, A. Multi-adjacent-vertexes and multi-shortest-paths problem of Dijkstra algorithm. *Comput. Sci.* **41**, 217–224 (2014).