Title

Exploring Endless Space

Journal

College Mathematics Journal, 54(3)

ISSN

0746-8342

Author

Aldous, David

Publication Date

2023-05-27

DOI

10.1080/07468342.2023.2201150

Exploring Endless Space

David Aldous

David Aldous (aldousdj@berkeley.edu) received his Ph.D. from Cambridge University in 1977, and held Professor positions at U.C. Berkeley from 1979 until retirement in 2018. His research has ranged across theoretical and applied probability. A central theme has been the study of large finite random structures, obtaining asymptotic behavior as the size tends to infinity via consideration of some suitable infinite random structure.

Games have long provided inspiration for mathematics – for instance the correspondence between Pascal and Fermat on the subject of settling fairly a wager on an unfinished game [**5**] is often regarded as the origin of mathematical probability. We discuss an example from a modern computer game. In the 4X genre [**14**], exemplified by the *Civilization* [**15**] series, there is an underlying map (different every time you play) of a fictional region, but you initially see only a neighborhood of a starting location. The first X is eXplore, by moving some kind of scout to learn more of the map. In the specific game *Endless Space 2 (ES2)* [**16**] you are initially compelled to explore via a "move to nearest unvisited location" command. How efficient is this rule, as a way of exploring the whole map? Questions like this have been studied intensively in the theory of algorithms, and are often difficult to answer. But in our particular setting there is a surprisingly simple worst-case analysis. In this article we revisit this known result, and comment on the less studied issue of average-case analysis.

Mathematical formulation of initial exploration in ES2

In discrete mathematics, a finite *graph* consists of a finite number of *vertices* (sometimes called *nodes*) and some *edges* (sometimes called *links*), where each edge indicates a relationship between two vertices, which are then called *adjacent* vertices. Visualizing a graph as a network of roads linking cities, we assume that each edge has some positive real length. We assume the graph is *connected*, in that for any two vertices there is a path of consecutive edges linking them. Such a path of edges has a *length*, the sum of edge-lengths, and the *distance* $d(v, v^*)$ between vertices is the length of the shortest path from v to v^*. For simplicity assume all such distances are distinct.

Consider such a graph G on n vertices. Fix a starting vertex v_0. There is some shortest path that *covers* (visits every vertex at least once) the graph; finding this "optimal" path is essentially[1] the famous *travelling salesman problem (TSP)* [**2**] which requires knowledge of the entire graph, and extensive computation, to solve.

In contrast there is a very simple and natural rule for finding a non-optimal cover path, the *nearest unvisited vertex (NUV)* rule.[2] The rule is simply

[1] TSP involves a *tour* returning to the initial vertex, but the distinction is not significant for our purposes.

[2] Confusingly usually called *nearest neighbor*, inconsistent with the usual terminology that neighbors are linked by a single edge.

- **move to the closest unvisited vertex**.

So in the NUV cover path, the vertices can be written $v_0, v_1, v_2, \ldots, v_{n-1}$ in order of first visit; symbolically

$$v_i = \arg \min_{v \notin \{v_0,\ldots,v_{i-1}\}} d(v_{i-1}, v), \quad 1 \le i \le n-1,$$

and this has length $L_{NUV}(G, v_0) = \sum_{i=1}^{n-1} d(v_{i-1}, v_i)$. How does this compare to the length $L_{opt}(G, v_0)$ of the optimal path?

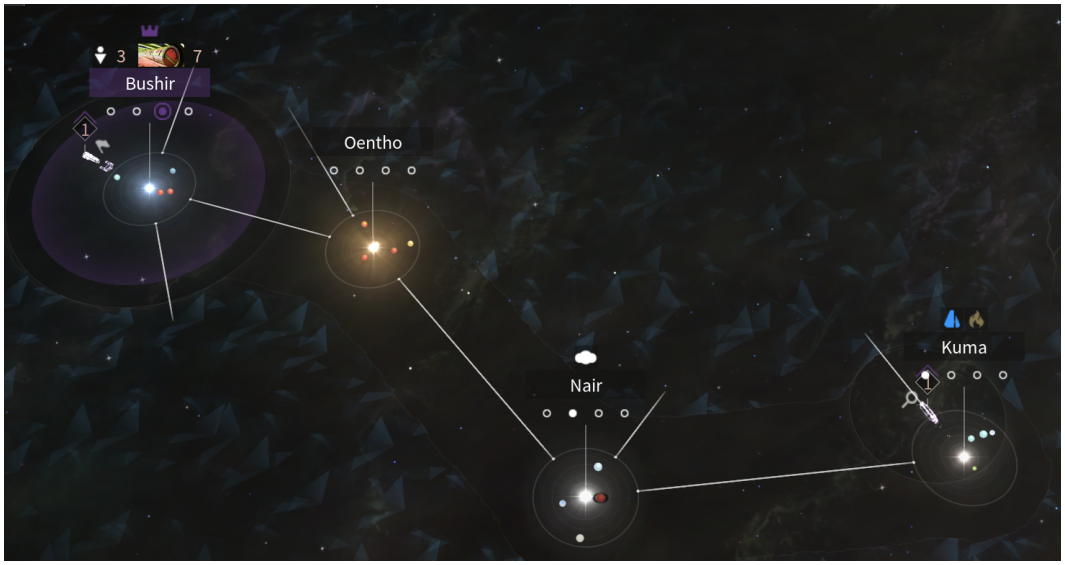

Figure 1. Screenshot of initial exploration in ES2.

Before starting the mathematical analysis, let us describe what you see when you start to play ES2. In the game, as shown in Figure 1, vertices are stars and edges are pathways (the slanted lines[3]) for your scout spaceship. Simplifying slightly[4], the ship moves at speed 1, and on encountering a star you see the initial part any new pathways from that star, and the ship starts moving along one new pathway, if possible, or otherwise starts to return along the arrival pathway.[5] In Figure 1 the ship took one of the three pathways from initial star Bushir, has visited 3 other stars, and is now leaving Kuma toward some yet-unseen star, with 4 un-explored pathways at previous stars.

In mathematical language, the computer has simulated an entire graph and a starting vertex. The player is initially shown the starting segments of the edges from that vertex. When you reach a new (previously unvisited) vertex, you see the number of edges there, and observe any edges that link to old (previously visited) vertices. So after visiting k distinct vertices (k steps) what you have seen (Figure 1) is the induced subgraph on those vertices, plus the starts of other edges leading away from that subgraph.

[3]The thin vertical lines are markers, not part of the graph.
[4]The game uses discrete turns in which ships move a constant distance.
[5]Later you can control the ship.

Note that in devising the algorithm for the NUV path, after k steps to find the next step, it is not necessary to consider the entire graph. One just needs to keep track of the induced subgraph (with edge-lengths) on those k vertices, plus the lengths of other edges leading away from that subgraph; that is enough information to determine the route to the next new vertex. In other words, what you see on the screen in the initial turns of the game is precisely the implementation of this natural "local" algorithm for generating the NUV walk.

The mathematical result

We now leave the game and consider the mathematics question raised earlier. It seems natural to measure the "efficiency" of the NUV scheme via the ratio $r(G, v_0) = L_{NUV}(G, v_0)/L_{opt}(G, v_0)$ of NUV path length to optimal path length. In the *analysis of algorithms* field [**4**] such questions are studied either in the average-case setting (discussed briefly later) or the worst-case setting. Here we consider the worst-case setting, where we study

$$a(n) := \text{ maximum of } r(G, v_0) \text{ over all } n\text{-vertex graphs } G \text{ and initial } v_0.$$

How does $a(n)$ vary with n, up to order of magnitude? In many contexts, questions like this are difficult. But in this context the solution is comparatively simple, and has been known for over 40 years. More precisely, it has been studied in the *tour context*, where the paths are required to return finally to the starting vertex. The citations below refer to the tour context, but their results carry over to our context with only minor changes to numerical constants.

Theorem 1 ([12]). *In the tour context on an n-vertex graph,*

$$\frac{1}{3}(\log_2(n+1)) + \frac{4}{9} \le a(n) \le \frac{1}{2}(\lceil \log_2 n \rceil + 1).$$

Here $\log_2$ means log to base 2.

For results like this, one needs a proof for the upper bound and an example for the lower bound. Both were given in [**12**], and other examples demonstrating the order $\log n$ lower bound can be found in [**7, 9, 8**]. Instead of repeating proofs of the sharpest known bounds, we will prove weaker results (5, 6) in ways which seem a little more intuitive.

Schemes other than NUV but still based only on the same "local" information have been studied – see the survey [**10**].

Proof of an upper bound

The NUV path makes its first visits to the distinct vertices in some order, which we write as $v_0, v_1, v_2, \ldots, v_{n-1}$. Say v_i has NUV-rank i and define $\Delta(v_i) = d(v_i, v_{i+1})$ and $\Delta(v_{n-1}) = 0$. So $\Delta(v)$ is the distance, when first visiting v, from v to the nearest unvisited vertex.

The argument[6] rests upon a simple observation, Lemma 1 below. Fix a vertex v^*

[6]Our proof is similar to the proof of Theorem 1 in [**12**], though they use a more precise but less intuitive analog of Lemma 1 .

and a real $L > 0$, and consider the set of vertices within distance L from v^*:

$$S(v^*, L) := \{v : d(v, v^*) \le L\}.$$

Lemma 1. *$\Delta(v) \le 2L$ for all $v \in S(v^*, L)$ except perhaps for the vertex $\bar{v}$ of highest NUV rank within $S(v^*, L)$.*

This holds because, when first visiting $v_i \in S(v^*, L)$ with $v_i \ne \bar{v}$, there is some first unvisited vertex v^o on the minimum-length path from v_i to $\bar{v}$ (maybe $v^o = \bar{v}$), and so

$$\Delta(v_i) \le d(v_i, v^o) \le d(v_i, \bar{v}) \le 2L$$

the final inequality using the triangle inequality via v^*.

Now consider the optimal cover path

$$v_0 = w(0), w(1), \ldots, w(n-1)$$

whose length is $L_{opt} = L_{opt}(G, v_0)$. Write $\lambda(w(i)) = \sum_{j=0}^{i-1} d(w(j), w(j+1))$ for the length of this path up to $w(i)$. Fix L and select vertices along the optimal path at distance L apart. Precisely, define $I(0) = 0$ and for $k \ge 0$ define

$$I(k+1) = \min\{i > I(k) : \lambda(w(i)) - \lambda(w(I(k))) > L\}$$

until no such i exists. This defines $I(k), 0 \le k \le N(L) - 1$ where the number $N(L)$ of defined values must satisfy $N(L) \le \lceil L_{opt}/L \rceil$. By definition, for each k all the vertices $w(i), I(k) \le i < I(k+1)$ are within distance L from $w(I(k))$, and so Lemma 1 implies that at most one[7] of those vertices w has $\Delta(w) > 2L$. So at most $N(L)$ vertices overall have $\Delta(w) > 2L$. Using this result for $L = mL_{opt}/n$ for integers $m \ge 1$,

$$\text{the number of vertices } w \text{ with } \tfrac{\Delta(w)}{2L_{opt}} > \tfrac{m}{n} \text{ is at most } \tfrac{n}{m} + 1. \tag{1}$$

Note also that *a priori* we have $\Delta(w) \le L_{opt}$. Finally, we have

$$L_{NUV} = \sum_w \Delta(w) \tag{2}$$

and there is now a standard (but slightly tedious in detail) calculation to bound this using (1). For the details we find it easiest to use probabilistic terminology and the identity that for a random variable $X \ge 0$,

$$\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X > x)\, dx. \tag{3}$$

Consider first $Y = \frac{\Delta(w)}{2L_{opt}} \le \frac{1}{2}$, where W denotes a uniform random vertex. So (2) says

$$\mathbb{E}[Y] = \tfrac{1}{n} \sum_w \tfrac{\Delta(w)}{2L_{opt}} = \tfrac{1}{2L_{opt}} \tfrac{L_{NUV}}{n} = \tfrac{r(G,v_0)}{2n}. \tag{4}$$

[7] This is the "key inequality" mentioned later.

And (1) says

$$\mathbb{P}(Y > \tfrac{m}{n}) \le \tfrac{1}{m} + \tfrac{1}{n},\ m = 1, 2, \ldots\ldots$$

To apply (3) we set $X = (Y - \frac{1}{n})^+ \le \frac{1}{2}$ so that $\mathbb{P}(X > x) \le \min(1, \frac{1}{nx} + \frac{1}{n})$ for all $x > 0$. Now

$$\mathbb{E}[X] = \int_0^\infty \mathbb{P}(X > x)\, dx \le \int_0^{1/2} \min(1, \tfrac{1}{nx} + \tfrac{1}{n})\, dx \le \tfrac{2}{n} + \tfrac{1}{n} \log n$$

and then

$$\mathbb{E}[Y] \le \tfrac{1}{n} + \mathbb{E}[X] \le \tfrac{3+\log n}{n}.$$

Combining with (4) we have $r(G, v_0) \le 2(3 + \log n)$ and so

$$a(n) \le 2(3 + \log n). \tag{5}$$

An example for a lower bound

The proof above might seem rather inefficient at first sight, and one might guess that some other proof would give a stronger result. On the other hand if one seeks to invent an example to show the order $\log n$ bound cannot be improved, then the conceptual issue is to find an example where some "key inequality" in the proof is (in order of magnitude) an equality. Following this line of thought, one might seek an example with the property

> (*) for various values of L with $1 \ll L \ll n$ there are distinguished vertices, separated by distance L all along the optimal path, such that the length of the NUV path from one distinguished vertex to the next is order L

corresponding to the "key inequality" footnoted in the proof. The example below implements that idea. We will construct a graph by adding segments in stages, and a vital aspect of the construction is that the NUV path at one stage starts out as the complete NUV path for the previous stage before traversing any added vertices.

Example. Start (stage 0) with a cycle of length m (for large m) with edges of length 1 (to follow the "distinct edge lengths" convention we should make small perturbations, but that doesn't affect the argument here). A section of the cycle is drawn as a line in Figure 2. Start from an initial vertex on the cycle. Clearly both L_{opt} and L_{NUV} equal $m - 1$; the paths go round the cycle.

For stage 1 of the construction, at every $d_1 = 4$'th vertex in the cycle, attach an edge of length $s_1 = 1.5$ leading to a new "stage 1" vertex. Now the optimal path goes up and down each new edge while going round the cycle, so L_{opt} is approximately[8] $m + 2 \times s_1 \times m/d_1 = \frac{7}{4}m$. But the NUV path first goes around the cycle and then must visit the stage 1 vertices in cyclic order by repeating a trip around the cycle. So its length L_{NUV} is now approximately $m + m + 2 \times s_1 \times m/d_1 = \frac{11}{4}m$. And the total number of vertices is $m + m/4$.

[8] There are some minor details omitted – if m is not exactly divisible by d_1, for instance – but these do not affect our asymptotic conclusion (6).

At stage 2, illustrated in Figure 2, at each $d_2 = 5$'th stage 1 vertex we create a new edge of length $s_2 = 7.5$ to a new "stage 2" vertex. Because the stage 1 vertices are distance 7 apart, the NUV path must first repeat the NUV path from the previous stage, and then visit the stage 2 vertices in cyclic order by using another trip around the cycle. The effect of the stage 2 augmentation is that L_{opt} increases by $2 \times s_2 \times \frac{m}{d_1 d_2}$ whereas L_{NUV} increases by that amount plus m.

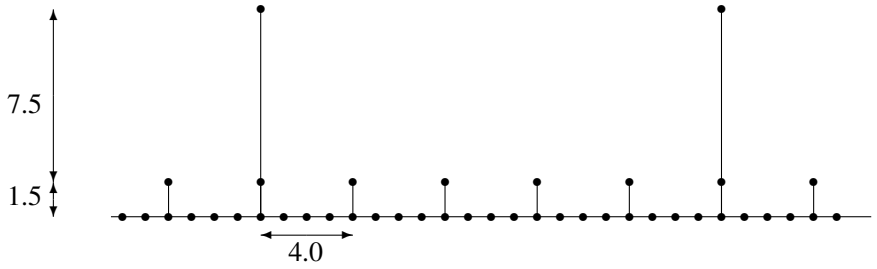

Figure 2. Stage 2 of the construction.

We can continue this construction, at each stage $i \geq 3$ attaching, at some proportion $1/d_i$ of stage $(i-1)$ vertices, a new edge of some length s_i to a new stage i vertex. We need to retain the "consistency" property, that the NUV path must first repeat the NUV path in the previous stage, and then visit the stage i vertices in cyclic order. The key point is that, for suitable choices of d_i and s_i and number of stages, we can arrange that the total number of vertices and the optimal path length L_{opt} remain $O(m)$ while the total NUV length L_{NUV} is not $O(m)$. So we are implementing the idea at (*), the distinguished vertices being the new vertices at a given stage.

To outline the details, choose integers $d_i \sim i^2$. The distance between consecutive stage-i vertices is $2\sum_{j=1}^{i} s_j + \prod_{j=1}^{i} d_j$ so we define

$$s_{i+1} = 2\sum_{j=1}^{i} s_j + \prod_{j=1}^{i} d_j + 1$$

and this ensures the consistency property. It is easy to check $s_{i+1} = (1 + o(1)) \prod_{j=1}^{i} d_j$ and so $s_{i+1} / \prod_{j=1}^{i+1} d_j \sim i^{-2}$. The extra length of the shortest covering path caused by this stage is $2m \times s_{i+1} / \prod_{j=1}^{i+1} d_j$ and so the total length L_{opt} of the shortest covering path remains $O(m)$. Similarly the total number of vertices $n(m)$ remains $O(m)$.

Now consistency implies that L_{NUV} increases by at least m for each stage, because another trip round the cycle is required. The number of stages permissible (that is, until distances become order m) is (up to an additive constant) the solution $\rho = \rho(m)$ of $\prod_{j=1}^{\rho} d_j = m$, that is of $(\rho!)^2 = m$, and this solution satisfies

$$\rho(m) \sim \frac{\log m^{1/2}}{\log \log m^{1/2}}.$$

So in terms of the number $n = n(m)$ of vertices, we have constructed graphs G_n for

which the ratio $r(G_n, v_0) = L_{NUV}(G - n, v_0)/L_{opt}(G_n, v_0)$ satisfies

$$r(G_n, v_0) \sim \frac{\log n}{2 \log \log n}, \tag{6}$$

which is close to the best known order, $\log n$.

Average-case analysis

Examples like ours, designed to demonstrate the possible $\log n$ ratio, are clearly artificial, and it seems plausible that for "typical" graph families the ratio is $O(1)$ as $n \to \infty$. This "typical" notion is formalized within *average-case analysis* by invoking some specific probability distribution on n-vertex graphs, though conceptually this merely rephrases the issue – what probability distributions generate "typical" graphs? There is a vast literature involving random graphs and complex networks [**3, 6, 11, 13**] but apparently very little study of NUV path length on random graphs. One approach is to scale edge-lengths so that the average distance to nearest neighbor is $O(1)$. Here the NUV length must be at least order n, so one can ask under what conditions the NUV length is indeed $O(n)$ rather than larger order. This question is studied in the recent technical research paper [**1**].

Returning finally to the topic of 4X games, even for other games in which the NUV strategy is not initially forced, it seems a reasonable strategy for acquiring territory. One can make a toy model of a generic game, in which players start at random vertices of a large graph, and move simultaneously at unit speed, gaining possession of vertices as they visit and thereafter forbidding others to visit such vertices. The goal is to capture as many vertices as you can. The NUV strategy is simple to define; how does it compare to more complex strategies that players implicitly use in games such as *Endless Space* and *Stellaris*? An interesting undergraduate research project is to study that question via simulation, for different random graph models and different strategies.

References

1. Aldous, D.J. (2022). The nearest unvisited vertex walk on random graphs. *Probab. Engineering Inform. Sci.*. 36:851–867.
2. Cook, W.J. (2012). *In Pursuit of the Traveling Salesman*. Princeton, NJ: Princeton University Press,
3. Coolen, A.C.C., Annibale, A., Roberts, E.S. (2017). *Generating Random Networks and Graphs*. Oxford U.K.: Oxford University Press.
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, c. (2009). *Introduction to Algorithms*, 3ed ed. Cambridge, MA: MIT Press.
5. Devlin, K. (2008). *The Unfinished Game*. New York: Basic Books.
6. Frieze, A., Karoński, M. (2016). *Introduction to Random Graphs*. Cambridge U.K.: Cambridge University Press.
7. Hougardy, S., Wilde, M. (2015). On the nearest neighbor rule for the metric traveling salesman problem. *Discrete Appl. Math.* 195:101–103.
8. Hurkens, C.A.J., Woeginger, G.J. (2004). On the nearest neighbor rule for the traveling salesman problem. *Oper. Res. Lett.* 32(1):1–4.
9. Johnson, D.S., Papadimitriou, C.H. (1985). Performance guarantees for heuristics. In: eds. Lawler, E.L, Lenstra, J.K, Rinnooy Kan, A. H. G., Shmoys, D.B. *The Traveling Salesman Problem*. Chichester: Wiley, pp. 145–180. Wiley.
10. Nicole Megow, N., Mehlhorn, K., Pascal Schweitzer, P. (2012). Online graph exploration: new results on old and new algorithms. *Theoret. Comput. Sci.* 463:62–72.
11. Newman, M. (2018). *Networks*, 2nd ed. Oxford: Oxford University Press.

12. Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.* 6(3):563–581.
13. van der Hofstad, R. (2017). *Random graphs and complex networks.* Cambridge: Cambridge University Press.
14. Wikipedia contributors. (2022). 4X. https://en.wikipedia.org/wiki/4X .
15. Wikipedia contributors. (2022). Civilization (series). https://en.wikipedia.org/wiki/Civilization (series).
16. Wikipedia contributors. (2022) Endless Space 2. https://en.wikipedia.org/wiki/Endless Space 2 .