Solving the 106 years old 3^k Points Problem with the Clockwise-algorithm

Marco Ripà

sPIqr Society, World Intelligence Network
Rome, Italy
e-mail: `marcokrt1984@yahoo.it`

Published online: 09 Jul. 2020 **Revised:** 18 Jul. 2020

Abstract. In this paper, we present the clockwise-algorithm that solves the extension in k-dimensions of the infamous nine-dot problem, the well known two-dimensional thinking outside the box puzzle. We describe a general strategy that constructively produces minimum length covering trails, for any $k \in \mathbb{N} - \{0\}$, solving the NP-complete $(3 \times 3 \times \cdots \times 3)$-points problem inside a $3 \times 3 \times \cdots \times 3$ hypercube. In particular, using our algorithm, we explicitly draw different covering trails of minimal length $h(k) = \frac{3^k-1}{2}$, for $k = 3, 4, 5$. Furthermore, we conjecture that, for every $k \geq 1$, it is possible to solve the 3^k-points problem with $h(k)$ lines starting from any of the 3^k nodes, except from the central one. Finally, we cover $3 \times 3 \times 3$ points with a tree of size 12.

Keywords: Nine dots puzzle, Nine-dot problem, Clockwise-algorithm, Thinking outside the box, Hypergraph, Lateral thinking, Link-length, Connectivity, Polygonal path, Optimization problem.

2010 Mathematics Subject Classification: Primary 05C85; Secondary 05C57, 68R10.

1 Introduction

The classic *nine dots puzzle* [9, 12] is the well known thinking outside the box challenge [3, 13], and it corresponds to the two-dimensional case of the general 3^k-points problem (assuming $k = 2$) [2, 5, 10, 15].

The statement of the 3^k-points problem is as follows:
"Given a finite set of 3^k points in $\mathbb{R}^k$, we need to visit all of them (at least once) with a polygonal path that has the minimum number of line segments $h(k)$, and we simply define the aforementioned line segments as *lines*. Let G_k be a $3 \times 3 \times \cdots \times 3$ grid in ${\mathbb{N}_0}^k$, we are asked to join all the points of G_k with a minimum (link) length covering trail $C := C(k)$ ($C(k)$ represents any trail consisting of $h(k)$ lines), without letting one single line of C go outside of a $3 \times 3 \times \cdots \times 3$ k-dimensional (hyper-)box (i.e., remaining inside a $4 \times 4 \times \cdots \times 4$ grid in $\mathbb{Z}^k$, which strictly contains G_k, and we call it *box*)".

It is trivial to note that the formulation of our problem is equivalent to asking:
"Which is the minimum number of turns $(h(k) - 1)$ in order to visit (at least once) all the points of the k-dimensional grid $G_k = \{(0, 1, 2) \times (0, 1, 2) \times \cdots \times (0, 1, 2)\}$ with a connected series of line segments (i.e., a possibly self-crossing polygonal chain allowed to turn at nodes and at Steiner points)?" [1, 16].

The goal of the present paper is to definitely solve the 3^k-points problem for any $k \in \mathbb{N} - \{0\}$.

We introduce a general algorithm, that we name as the *clockwise-algorithm*, which produces optimal covering trails for the 3^k-points problem. In particular, we show that $C(k)$ has $h(k) = \frac{3^k-1}{2}$ lines, answering to the most spontaneous 106 years old question which arose from the original Loyd's puzzle [12].

The aspect of the 3^k-points problem that most amazed us, when we eventually solved it, is the central role of Loyd's expected solution for the $k = 2$ case. In fact, the clockwise-algorithm, able to solve the main problem in a k-dimensional space, is the natural generalization of the classic solution of the nine dots puzzle.

2 Solving the 3^k-points problem

The stated 3^k-points optimization problem, especially for $k < 4$, appears to have concrete applications in manufacturing, drone routing, cognitive psychology, and integrated circuits (VLSI design). Many suboptimal bounds have been proved for the NP-complete [4] 3^k-points problem under additional constraints (such as limiting the solutions to Hamiltonian paths or considering only rectilinear spanning paths [2, 6, 10]), but (to the best of our knowledge) the $3^{k>3}$-points problem remains unsolved to the present day, and this paper provides its first exact solution so far [14].

2.1 A tight lower bound

Given the 3^k-points problem as introduced in Section 1, if we remove its constraint on the inside the box solutions, then we have that a lower bound is provided by Theorem 1.

Theorem 1. $\forall k \in \mathbb{N} - \{0\}$, $h(k) \geq \frac{3^k-1}{2}$.

Proof. If $k = 1$, then it is necessary to spend (at least) one line to join the 3 points.

Given $k = 2$, we already know that the nine-dot problem cannot be solved with less than 4 lines (see [8], assuming $n = 3$).

Let k be greater than 2. We invoke the proof of Theorem 1 in [14], substituting $n_i = 3$.

Thus, equation (4) of [14] can be rewritten as

$$h_l(3_1, 3_2, \dots, 3_k) = \left\lceil \frac{3^k-1}{2} \right\rceil, \tag{1}$$

which is an integer (since $3^k - 1$ is always even).

Therefore, $h(k) \geq h_l(3_1, 3_2, \dots, 3_k) = \frac{3^k-1}{2}$ for any (strictly positive) natural number k. □

It is redundant to point out that Theorem 1 provides also a valid lower bound for any 3^k-points (*arbitrary*) *box-constrained* problem. The purpose of Section 2.2 is to show that this bound matches $h(k)$ for every k.

2.2 The clockwise-algorithm

In order to introduce the clockwise-algorithm, let we begin from the trivial case $k = 1$. This means that we have to visit 3 collinear points with a single line, remaining inside a unidimensional box which is 3 units long.

One solution is shown in Figure 1.

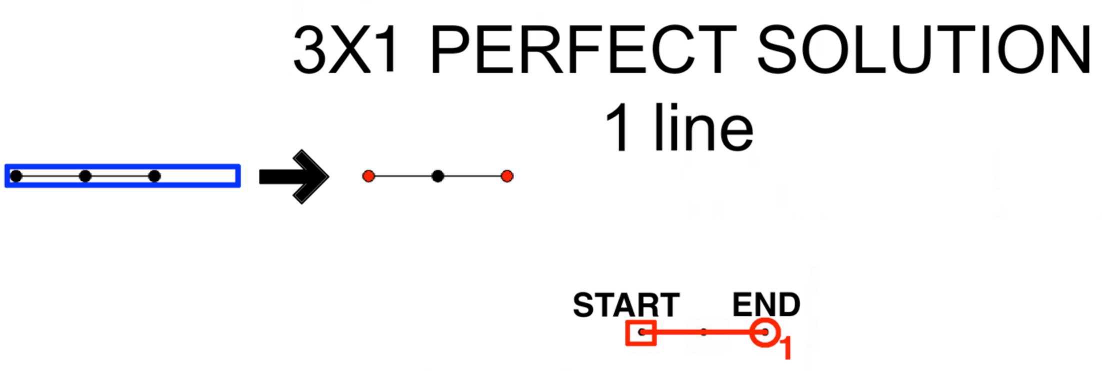

Figure 1. Solving the 3×1 puzzle inside the box (3 units of length), starting from one of the line segment endpoints. The puzzle is solvable with this $C(1)$ path starting from both the red points.

Considering the spanning path by Figure 1, it is easy to see that we cannot solve the 3^1-points problem starting from one point of G_1 if and only if this point is the central one.

Given $k = 2$, we are facing the classic nine dots puzzle considering a 3×3 box (9 units of area). The well-known Hamiltonian path shown in Figure 2 proves that we can solve the problem, without allowing any line to exit from the box, if we start from any node of G_2 except from the central one [8].

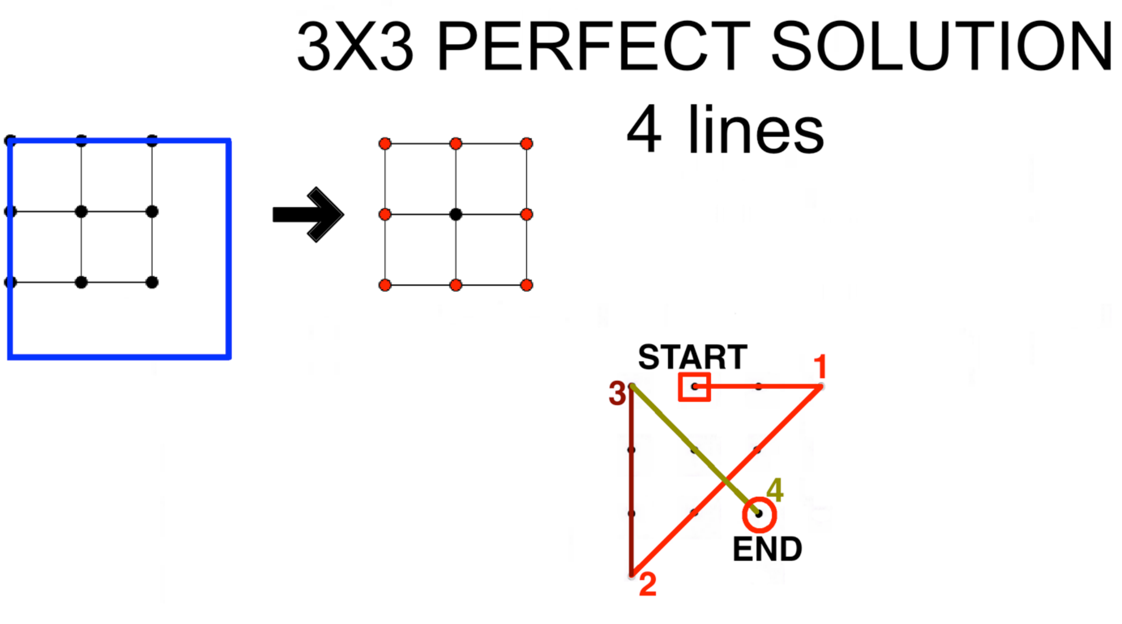

Figure 2. $C(2)$ is a path that consists of $h(2) = \frac{3^2-1}{2}$ lines. In order to solve the 3×3 puzzle with 4 lines starting from one node of G_2, it is necessary to avoid to start from the central point of the grid.

Looking carefully at $C(2)$, as shown in Figure 2, we note that line 1 includes $C(1)$ if we simply extend it by one unit backward. Thus, $C(1)$ and the first line of $C(2)$ are essentially the same trail, and so they are considering the clockwise-algorithm. Line 2 can be obtained from line 1 going backward when we apply a standard rotation of $\frac{\pi}{4}$ radians: we are just spinning around in a two-dimensional space, forgetting the $3^{2-1} - 1$ collinear points that will later be covered by the repetition of $C(1)$ following a different direction. We are now able to understand what line 3 really is: it is just a link between the repeated $C(2-1)$ trail backward and the final $C(2-1)$ trail following the new direction. In general, the aforementioned link corresponds to line $2 \cdot h(k-1)+1 = 3^{k-1}$ of any $C(k)$ generated by the clockwise-algorithm.

Definition 1. Let G_3 be the grid in ${\mathbb{N}_0}^3$ such that $G_3 = \{(0,1,2) \times (0,1,2) \times (0,1,2)\}$. We call "nodes" all the 27 points of G_3, as usual. In particular, we indicate the nodes $V_1 \equiv (0,0,0)$, $V_2 \equiv (2,0,0)$, $V_3 \equiv (0,2,0)$, $V_4 \equiv (0,0,2)$, $V_5 \equiv (2,2,0)$, $V_6 \equiv (2,0,2)$, $V_7 \equiv (0,2,2)$, $V_8 \equiv (2,2,2)$ as "vertices", we indicate the nodes $F_1 \equiv (1,1,0)$, $F_2 \equiv (1,0,1)$, $F_3 \equiv (0,1,1)$, $F_4 \equiv (2,1,1)$, $F_5 \equiv (1,2,1)$, $F_6 \equiv (1,1,2)$ as "face-centers", we call "center" the node $X_3 \equiv (1,1,1)$, and we indicate as "edges" the remaining 12 nodes of G_3.

Now, we are ready to describe the generalization of the original Loyd's covering trail to a higher number of dimensions. Given $k=3$, a minimum length covering trail has already been shown in [14], but this time we need to solve the problem inside a $3 \times 3 \times 3$ box. Our strategy is to follow the optimal two-dimensional covering trail (see Figure 2) swirling in one more dimension, according to the 3-steps scheme given by lines 1 to 3 of $C(2)$, and beginning from a congruent starting point.

Thus, if we take one vertex of G_3, while we rotate in the space at every turn (as observed for $k=2$), it is possible to repeat twice (forward and backward) the whole $C(2)$ or, alternatively (Figure 3), we can follow $\frac{8}{3}$ times the scheme provided by its lines 1 to 3. In both cases, at the end of the process, $3^{3-2} - \frac{1}{3}$ gyratories have been performed, so we spend the (3^{3-1})-th line to close the subtour ($C(3)$ can never be a cycle plus we avoided to extend its first line backwards, but we have already seen that this fact does not really matter), joining $3-1$ new points. In this way, we reach the *starting vertex* again, and the last $3^3 - 1$ unvisited nodes belong only to $G_{k-1} = G_2$ (choosing the right direction). Therefore, we can finally paste $C(2)$ (Figure 2) by extending one unit backward its first line (the new $(2 \cdot h(3-1)+2)$-th line) in order to visit all the 3^2 nodes of G_{3-1}.

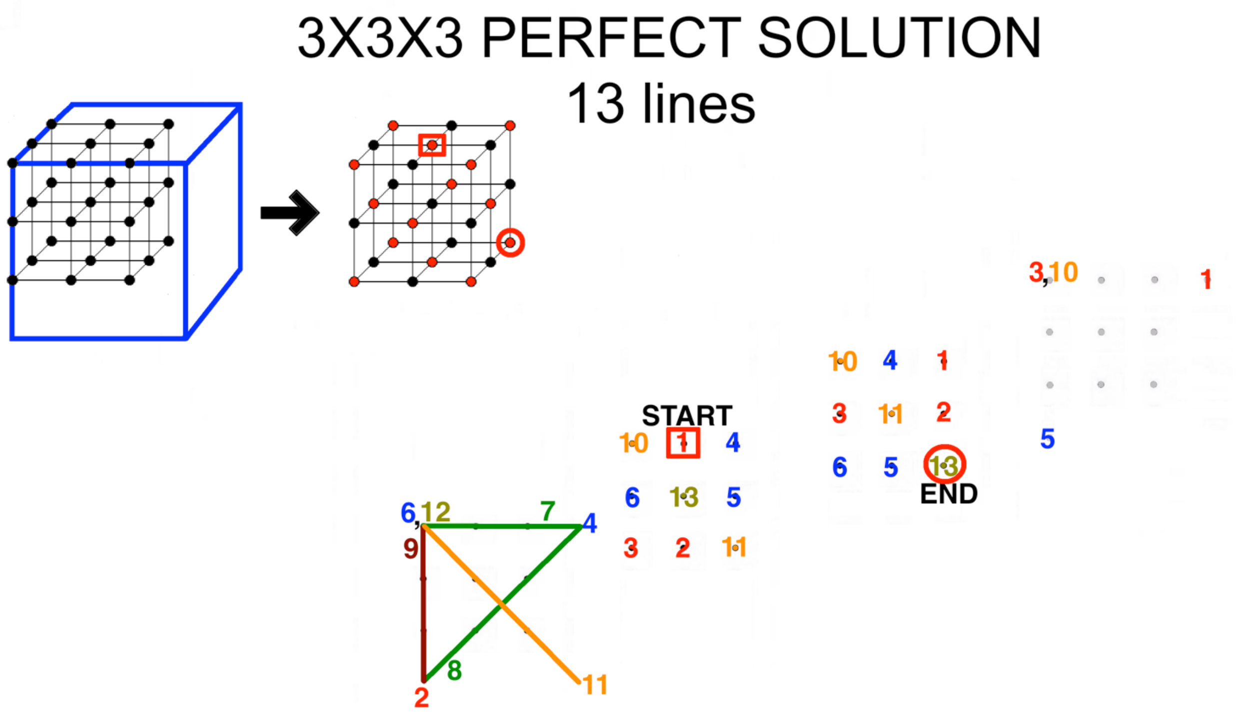

Figure 3. $C(3)$ solves the $3 \times 3 \times 3$ puzzle inside a $3 \times 3 \times 3$ box (27 cubic units of volume), starting from face-centers or vertices, thanks to the clockwise-algorithm.

Before moving on $k = 4$, we wish to prove that the 3^3-points problem is solvable starting from any node of G_3 if we exclude the center of the grid (as we have previously seen for $k \in \{1, 2\}$). This result immediately follows by symmetry when we combine the trails shown in Figures 3&4.

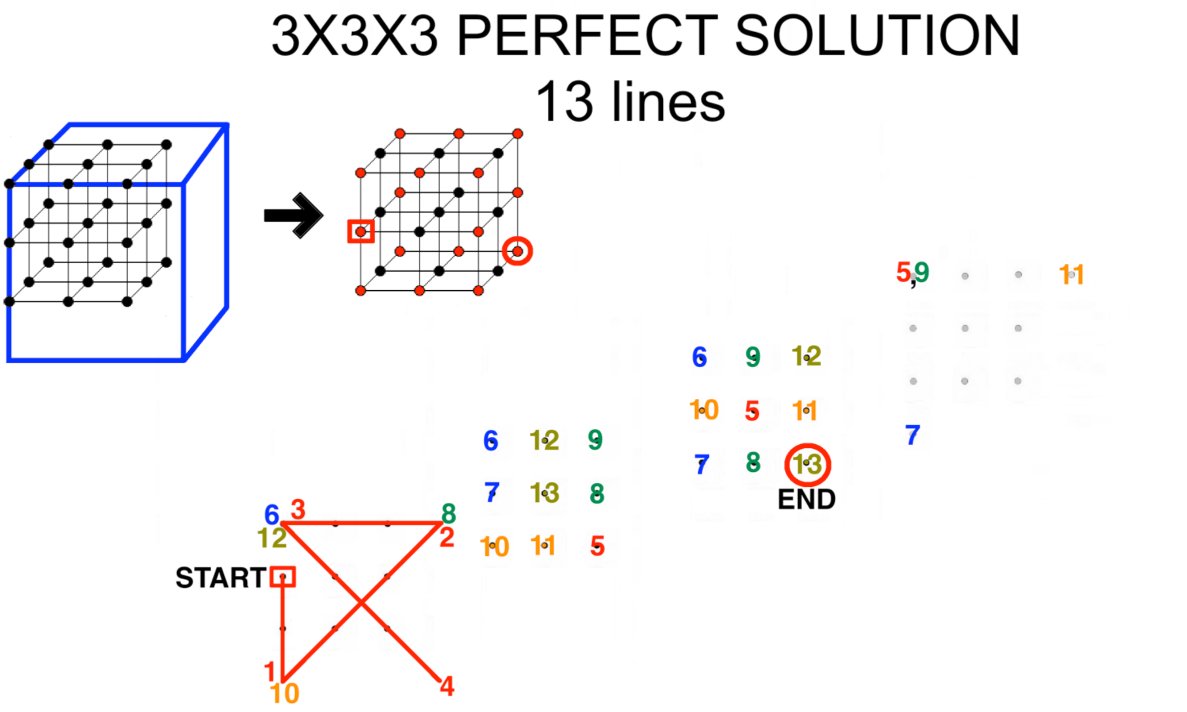

Figure 4. Solving the $3 \times 3 \times 3$ puzzle inside a $3 \times 3 \times 3$ box (27 cubic units of volume), starting from edges or vertices.

The number of solutions with $\frac{3^k-1}{2}$ lines increases as k grows. Moreover, if we remove the box constraint, we are able to find new minimal covering trails [14], including those that reproduce (on a given 3×3 subgrid of G_3) the endpoints by Figure 2, as shown in Figure 5.

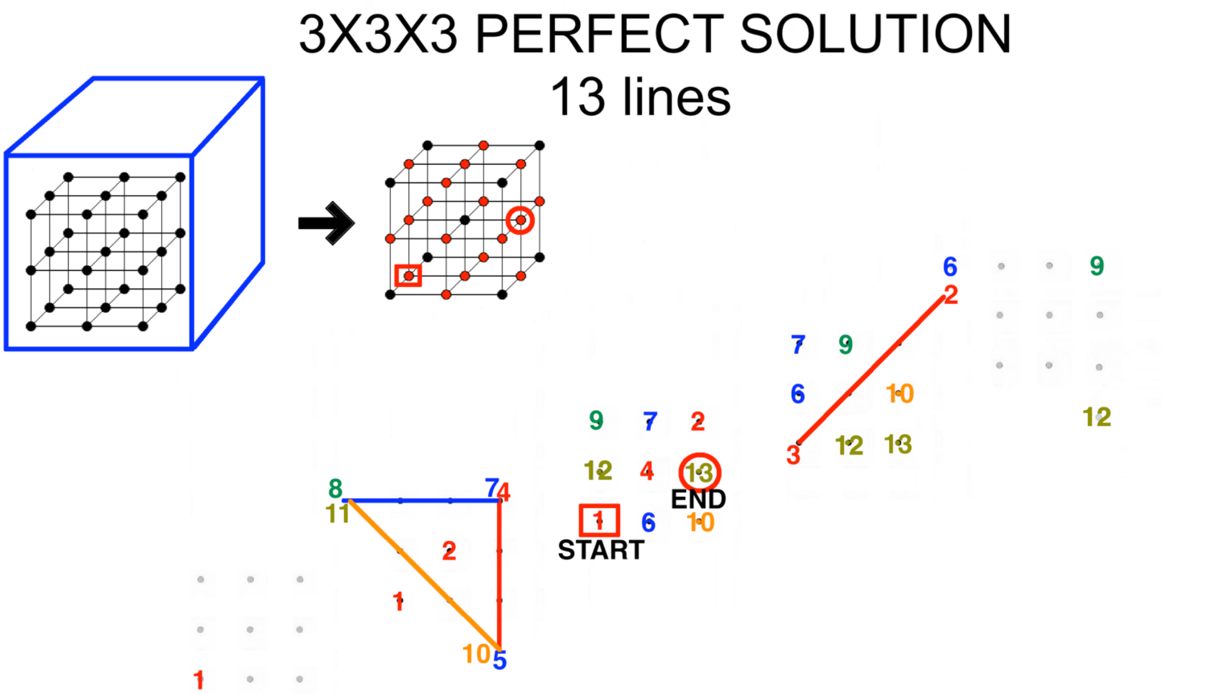

Figure 5. Solving the $3 \times 3 \times 3$ puzzle inside a $3 \times 3 \times 4$ box (36 cubic units of volume).

Finally, we present the solution of the 3^4-points problem. Two examples of minimum length covering trails generated by the clockwise-algorithm are given.

The method to find $C(4)$ is basically the same one that we have previously discussed for G_3. So, we utilize the standard pattern shown in Figure 3 as we used $C(2)$ in order to solve the 3^3-points problem. We apply $C(3)$ forward (while we spin around following the 3-steps gyratory as shown in Figure 6), then backward (Figure 7), subsequently we return to the starting vertex with line 27 (the $(2 \cdot h(4-1)+1)$-th link), and lastly we join the 3^3-1 unvisited nodes with $C(3)$ by simply extending backward its first line (corresponding to the 28-th link of $C(4)$ - see Figure 8).

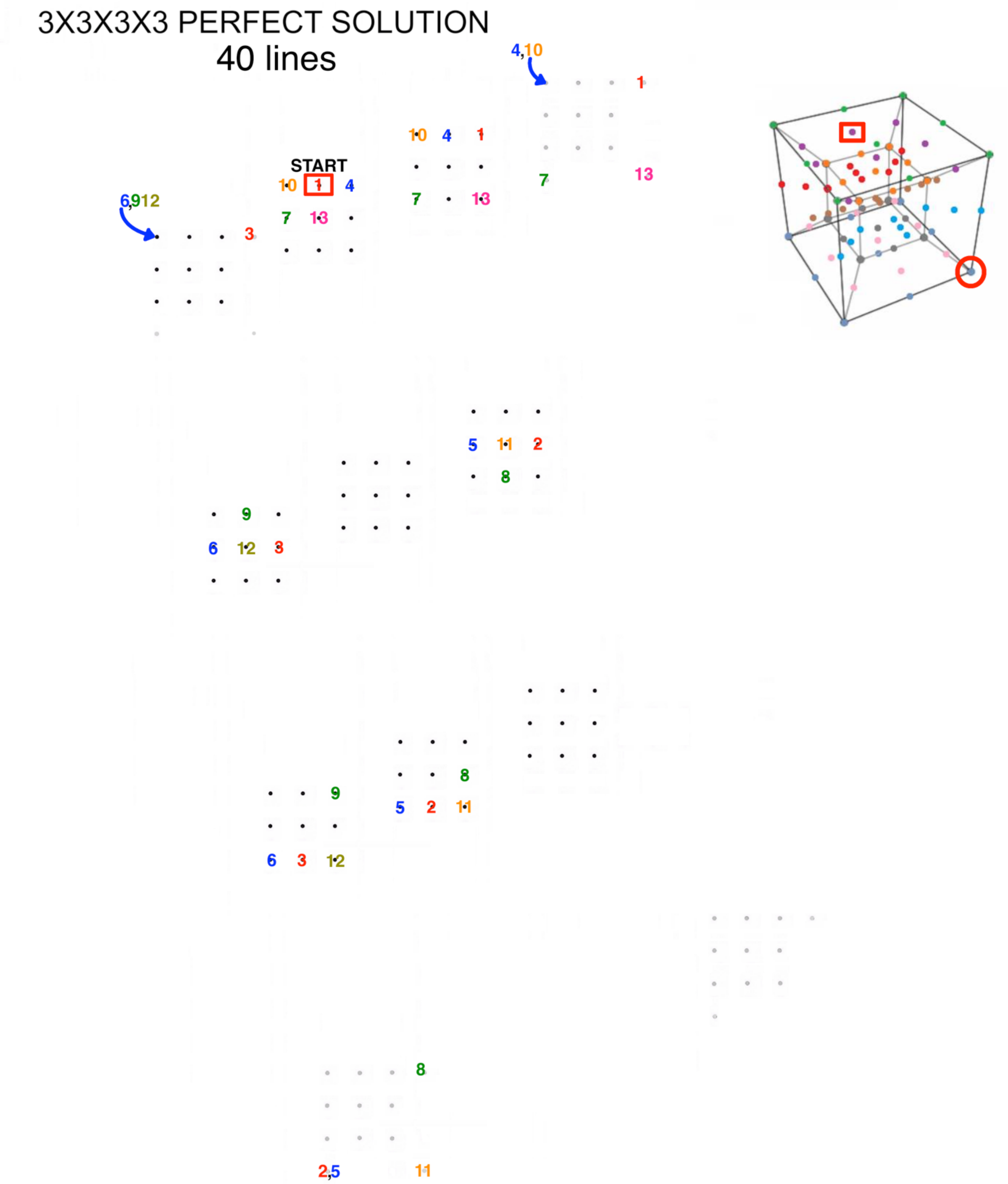

Figure 6. Lines 1 to 13 of $C(4)$ following $C(3)$, as shown in Figure 3.

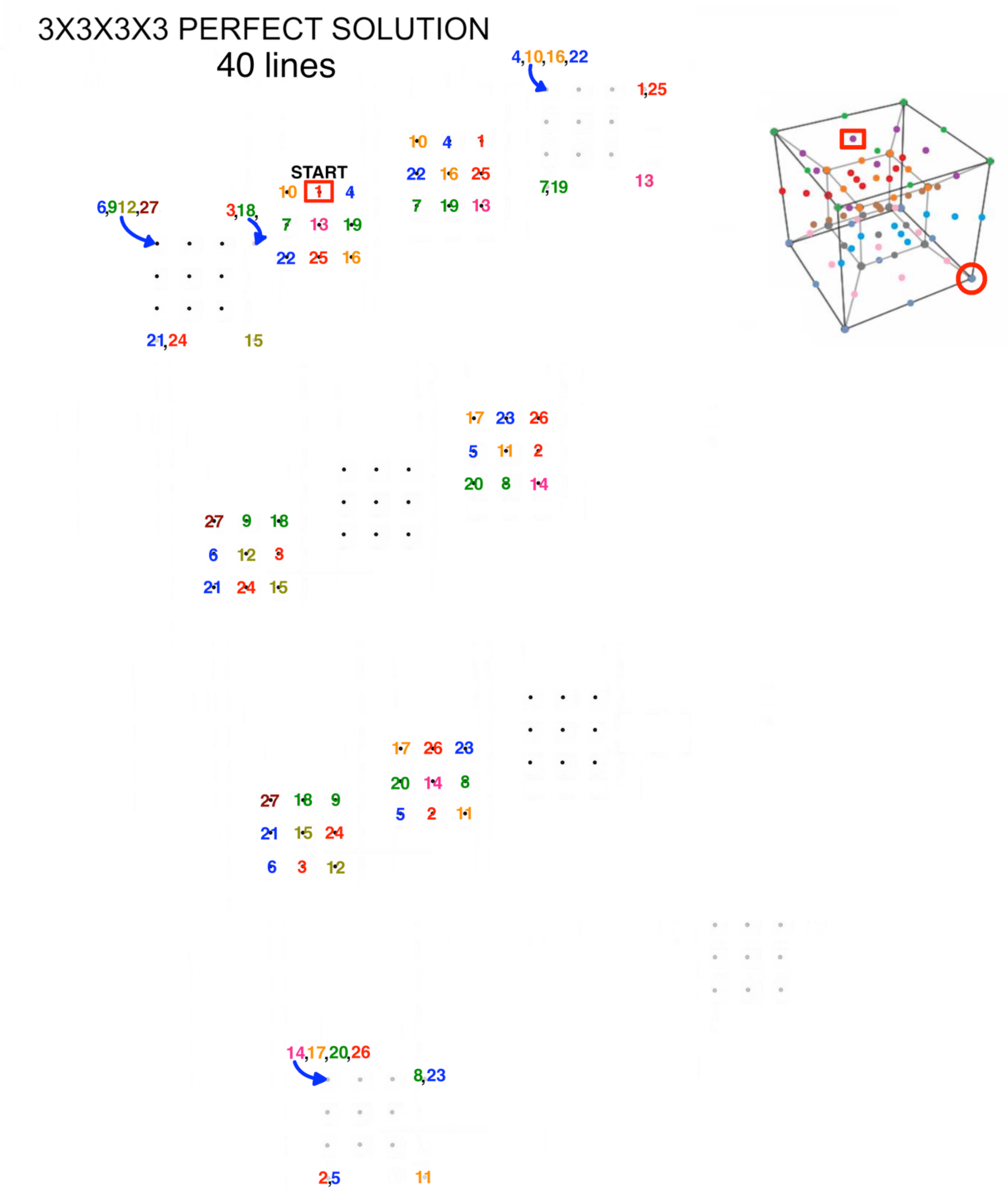

Figure 7. Lines 14 to 27 of $C(4)$ following $C(3)$ backward, the 27-th link to come back to the "starting point" is also included.

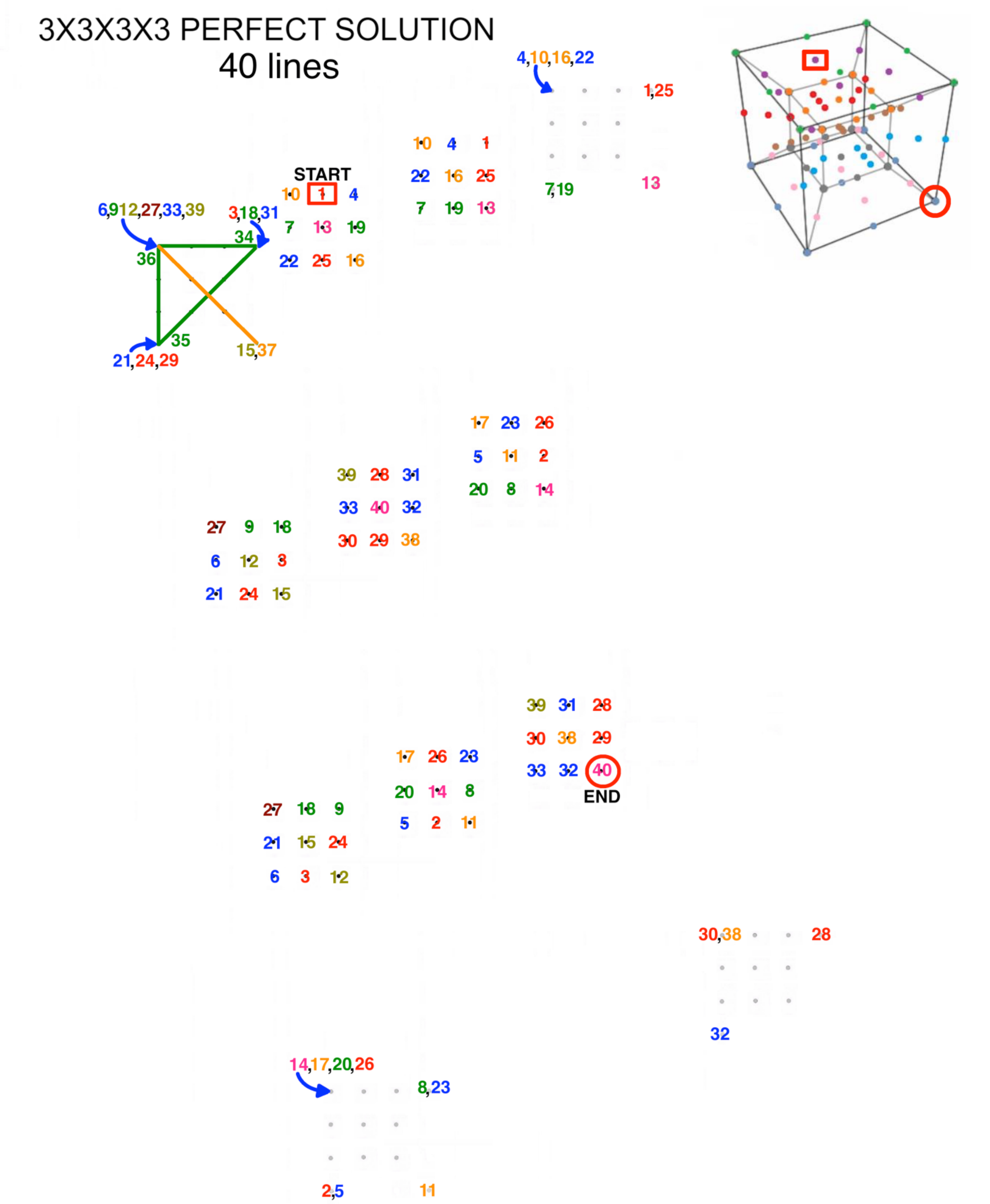

Figure 8. A minimum length covering trail that completely solves the $3 \times 3 \times 3 \times 3$ puzzle with 40 lines, inside a $3 \times 3 \times 3 \times 3$ box (hyper-volume 81 units⁴), thanks to the clockwise-algorithm applied to $C(3)$ from Figure 3.

The clockwise-algorithm reduces the complexity of the 3^k-points problem to the complexity of the 3^{k-1}-points one. A clear example is shown in Figure 9 .

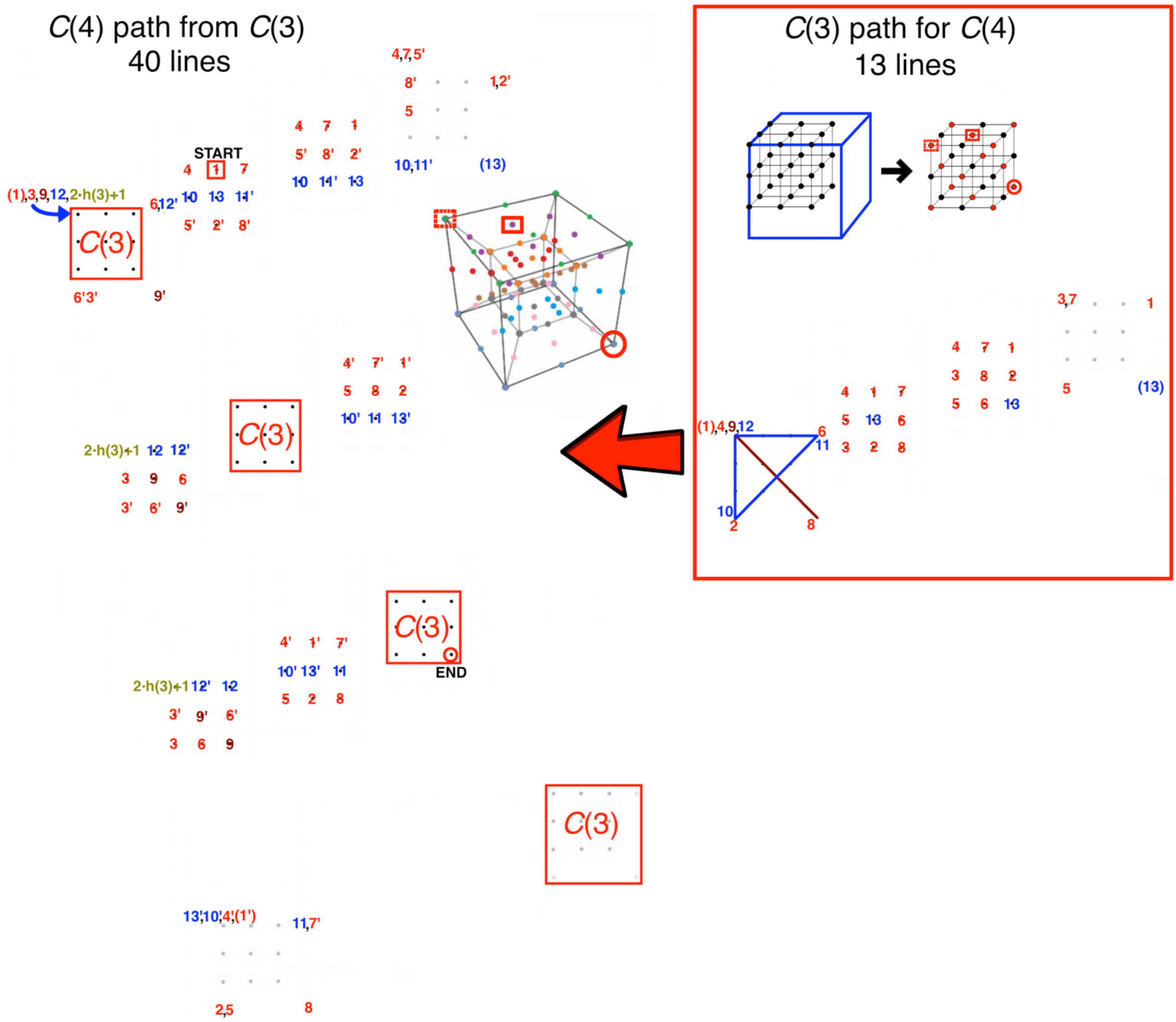

Figure 9. How the clockwise-algorithm concretely works: it takes a minimum length covering trail $C(3)$ as input, and returns $C(4)$. Lines 1-13 belong to the covering trail $C(3)$ (shown in the upper-right quadrant), line 13' follows line 13 and belongs to $C(3)$ backward. $C(3)$ backward ends with line 1': it is extended (by one unit) in order to be connected to the $(2 \cdot h(3^3) + 1)$-th link, and this allows $C(3)$ to be repeated one more time (joining the remaining 26 unvisited nodes).

Since the clockwise-algorithm takes $C(k-1)$ as input and returns $C(k)$ as its output, it can be applied to any $C(k)$ in order to produce some $C(k+1)$ consisting of $h(k+1) = 3 \cdot h(k) + 1$ lines. Thus, it is possible to shown by induction on k that the 3^k-points problem can be solved, inside a $3 \times 3 \times \cdots \times 3$ box of hyper-volume 3^k units^k, drawing optimal trails with $3 \cdot h(k-1) + 1$ lines (Figure 10).

Therefore, $\forall k \in \mathbb{N} - \{0\}$,

$$h(k+1) = 3 \cdot h(k) + 1 = \frac{3^{k+1}-1}{2}. \tag{2}$$

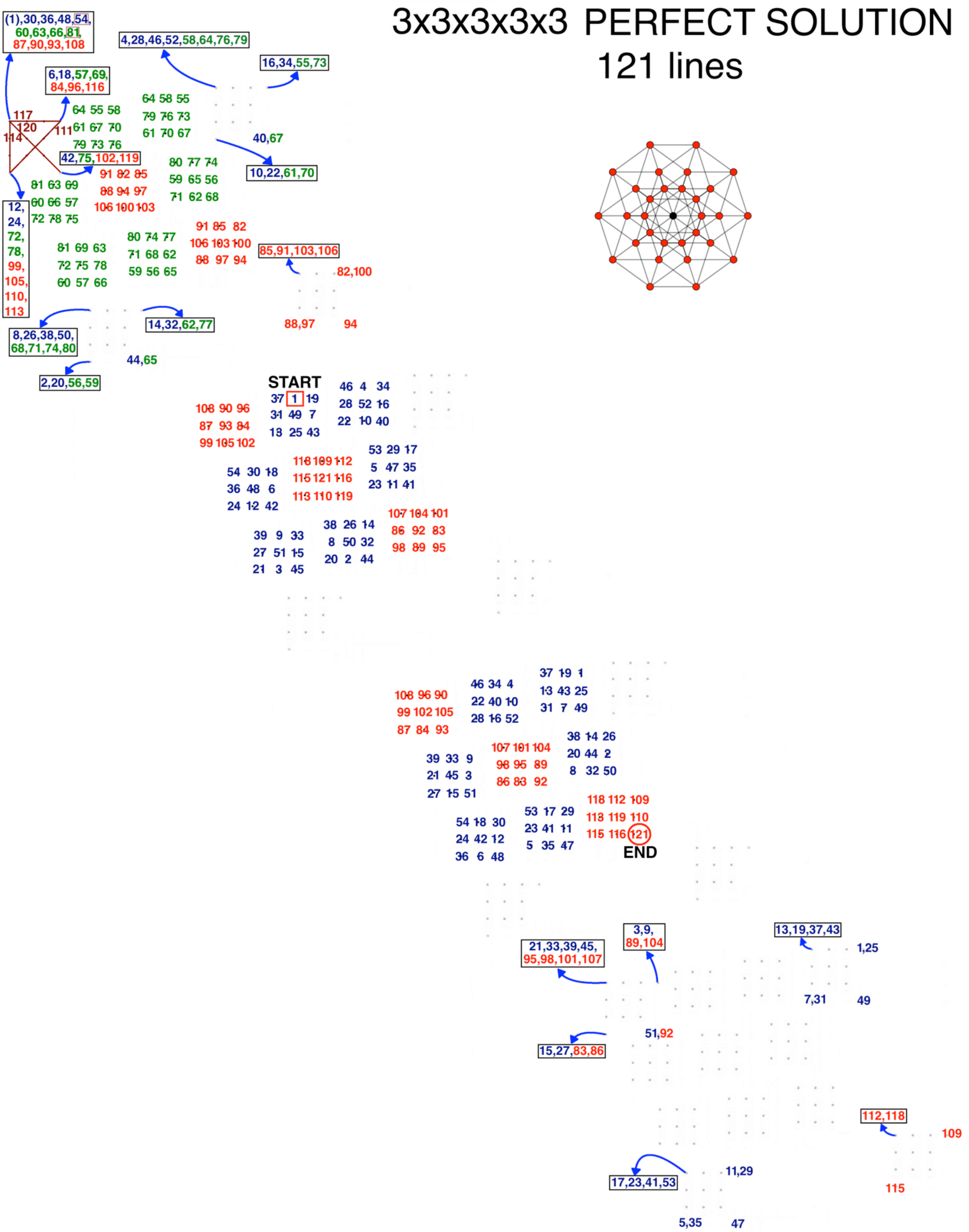

Figure 10. For any $k > 1$, the 3^k-points problem can be explicitly solved by the clockwise-algorithm ($k = 5$ in our example). A $C(k)$ with $\frac{3^k-1}{2}$ lines immediately follows from any valid $C(k-1)$, and this surely occurs if $C(k-1)$ has one of its endpoints in a vertex of G_{k-1}.

3 Covering 3^k-points by trees

Definition 2. We call a *tree* any acyclic connected arrangement of line segments (i.e., *edges* of the tree) which covers some of the nodes of G_k, and we denote as $T(k)$ any tree (drawn in $\mathbb{R}^k$) that covers all the points belonging to the k-dimensional grid G_k. More specifically, $T(k)$ represents a *covering tree* for G_k of size $t(k)$ (i.e., $T(k)$ has $t(k)$ edges).

In 2014, Dumitrescu and Tóth [7] shown the existence of an inside the box covering tree for G_k, $\forall k \in \mathbb{N} - \{0\}$, of size $t_u(k) = h(k) = \frac{3^k-1}{2}$ (e.g., the set of all the endpoints of the 13 edges of $t_u(3) \subset G_3$ - see Definition 1). It is not hard to prove that, when we take as a constraint our $3 \times 3 \times \cdots \times 3$ box (as usual), the upper bound $t_u(k)$ is not tight for every $k > 3$.

Lemma 1. Let box be the set of 4^k points such that $box := \{(-1,0,1,2) \times (-1,0,1,2) \times \cdots \times (-1,0,1,2)\} \subset \mathbb{Z}^k$. $\forall k \geq 4$, $\exists$ a covering tree $T(k)$ for G_k whose all its vertices belong to box $\wedge$ s.t. $T(k)$ has size $t(k) < h(k)$.

Proof. We invoke Theorem 1 to remember that $h(k) \geq \frac{3^k-1}{2}$. It follows that it is sufficient to provide a general strategy to cover G_k with a tree consisting of $\frac{3^k-1}{2} - c(k > 3)$ edges, for some $c(k > 3) \geq 1$. The tree in $\mathbb{R}^3$ shown in Figure 11, that covers $3^3 - 1$ nodes of G_3 with its 12 edges, also provides a valid upper bound for $t(4)$, since it is sufficient to clone twice the same pattern and spend one more link to join the remaining three collinear points belonging to each copy of G_3. So, we add 2 more lines (at most) to connect every duplicated tree (to the other two copies of itself) and to fix the aforementioned link (which joins the last 3 unvisited nodes of G_4), in order to create a covering tree of size 39.

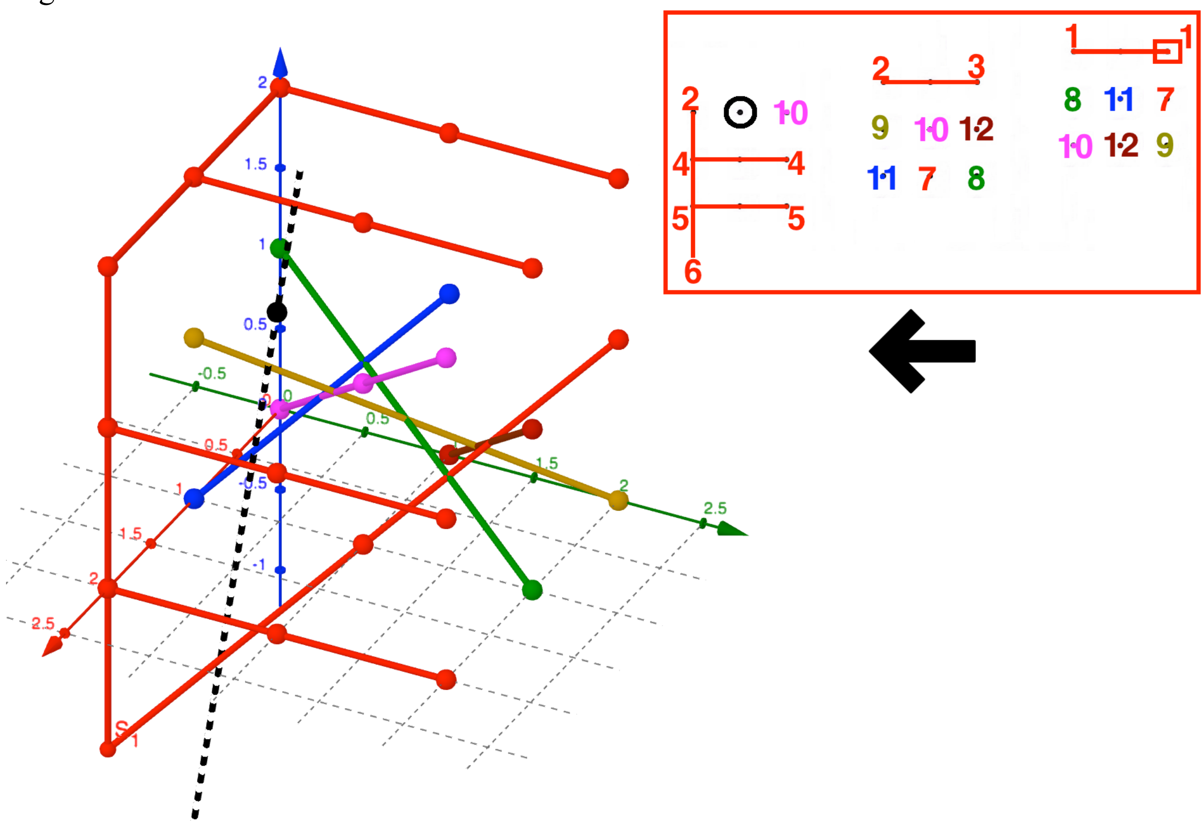

Figure 11. An inside the $(2 \times 2 \times 3)$ box tree with $t_u(3) - 1 = 12$ edges that covers all the points of G_3 except the black one. The black dotted line represents the direction (w-axis) to fit the remaining three collinear points of G_4 when we replicate three times the same pattern (picture realized with GeoGebra [11]).

Thus, we can generalize the result $\forall k \geq 4$,

$$t(k) \leq 3 \cdot t(k-1) + 1 \leq 39 \cdot 3^{k-4} + \sum_{i=1}^{k-5} 3^i + 1. \tag{3}$$

Hence,

$$t(k) \leq \frac{3^{k-4}-1}{2} + 13 \cdot 3^{k-3}. \tag{4}$$

Therefore, $\forall k \geq 4,\ h(k) - t(k) \geq 3^{k-4} \geq 1$. □

We are finally ready to remove the box constraint. Without any restriction to our *thinking outside the box* ability, we are free to apply in a clever way the idea introduced by Figure 11, in order to prove the existence of a covering tree for G_3 of size $t(3) = n^2 + n$ (here n assumes the odd value 3 - see [7], third section).

Theorem 2. $t(k) < h(k)$ iff $k \geq 3$.

Proof. Let $k = 1$; it is trivial to verify that $t(1) = h(1) = 1$.
If $k = 2$, then $t(2) = h(2) = 4$ (see [8]).
Thus, let $k = 3$. Figure 12 shows the existence of a covering tree of size

$$12 = t(3) < h(3) = 13. \tag{5}$$

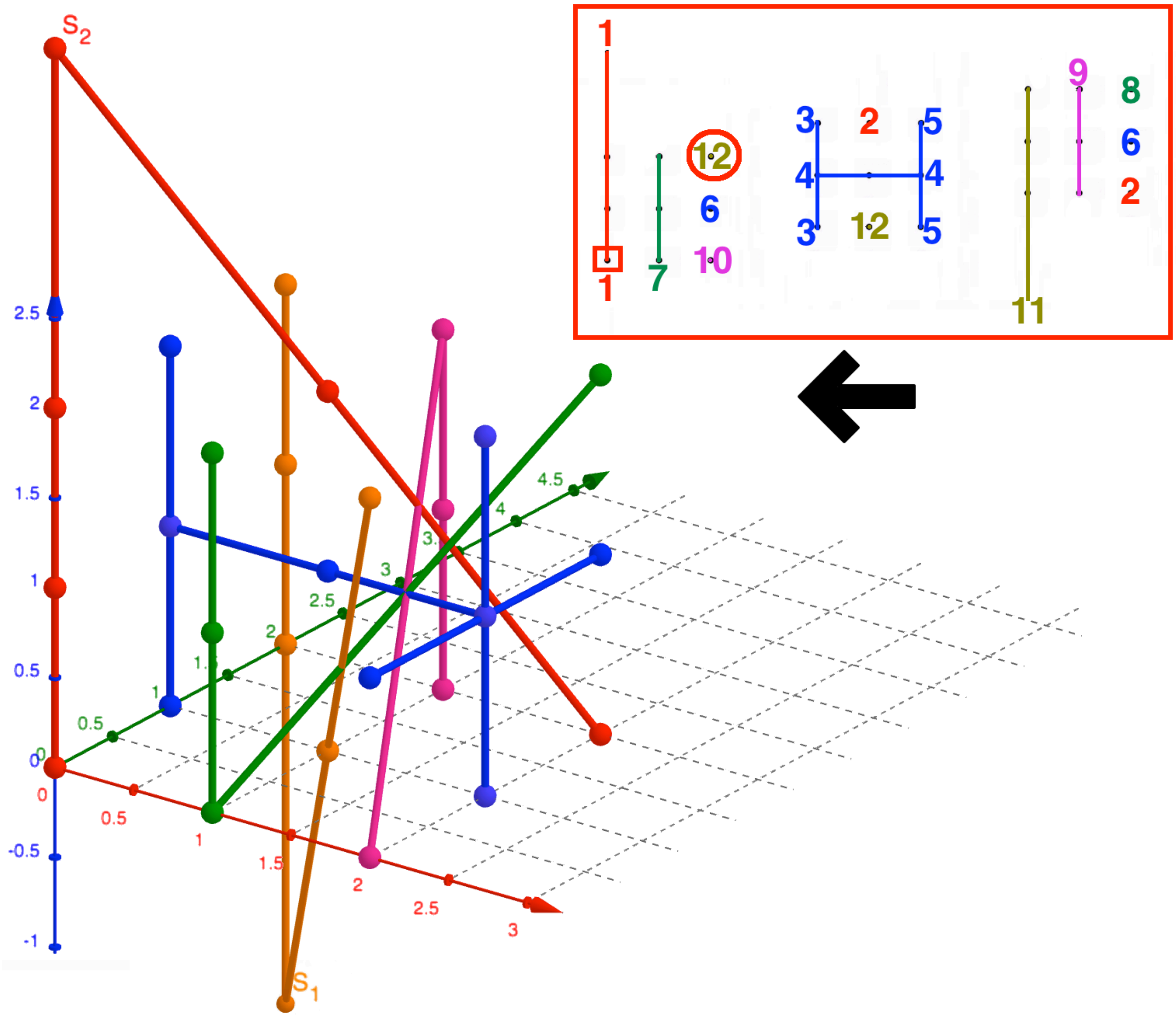

Figure 12. One covering tree with $t(3) = 12$ edges. $T(3)$ covers all the points of G_3 (picture realized with GeoGebra [11]).

If $k \geq 4$, then Lemma 1 states that $t(k) < h(k)$. In particular, equation (3) shows that

$$t(k) \leq (3 \cdot t(3) + 1) \cdot 3^{k-4} + \sum_{i=1}^{k-5} 3^i + 1. \quad (6)$$

Hence,

$$t(k) \leq \frac{25 \cdot 3^{k-3} - 1}{2}. \quad (7)$$

Since we already proved that $h(k) = \frac{3^k - 1}{2}$ is optimal,

$$h(k) - t(k) \geq \frac{3^k - 1}{2} - \frac{25 \cdot 3^{k-3} - 1}{2}. \quad (8)$$

Therefore, we conclude that, $\forall k \geq 3$, $h(k) - t(k) \geq 3^{k-3} \geq 1$. □

4 Conclusion

Given the k-dimensional grid G_k, the clockwise-algorithm let us easily draw different covering trails with $\frac{3^k - 1}{2}$ lines, and all of them remain inside the $(3 \times 3 \times \cdots \times 3)$ box. After the $(3^k - 1)$-th link, it is possible to switch from the previously applied $C(k-1)$ to another known solution of the 3^{k-1}-points problem, completing a new optimal trial that has a different endpoint (e.g., we can take the walk shown in Figure 7 and then apply $C(3)$ from Figure 9).

Let $X_k \equiv (1, 1, \ldots, 1)$ be the central node of G_k (see Definition 1 for the case $k = 3$). We conjecture that, $\forall k \in \mathbb{N} - \{0\}$, the 3^k-points problem is solvable (embracing also every outside the box optimal trail) starting from any node of $G_k - \{X_k\}$ with a covering trail of length $h(k) = \frac{3^k - 1}{2}$, while it is not if we include X_k as an endpoint of $C(k)$.

References

[1] Aggarwal, A., Coppersmith, D., Khanna, S., Motwani, R., Schieber, B. (1999). The angular-metric traveling salesman problem. *SIAM Journal on Computing* **29**, 697–711.

[2] Bereg, S., Bose, P., Dumitrescu, A., Hurtado, F., Valtr, P. (2009). Traversing a set of points with a minimum number of turns. *Discrete & Computational Geometry* **41(4)**, 513–532.

[3] Chein, J.M., Weisberg, R.W., Streeter, N.L. *et al.* (2010). Working memory and insight in the nine-dot problem. *Memory & Cognition* **38**, 883–892.

[4] Chitturi, B., Pai, J. (2018). Minimum-Link Rectilinear Covering Tour is NP-hard in R^4. *arXiv*, 1 Oct. 2018, `https://arxiv.org/abs/1810.00529`

[5] Collins, M.J. (2004). Covering a set of points with a minimum number of turns. *International Journal* of *Computational Geometry & Applications* **14(1-2)**, 105–114.

[6] Collins, M.J., Moret, M.E. (1998). Improved lower bounds for the link length of rectilinear spanning paths in grids. *Information Processing Letters* **68(6)**, 317–319.

[7] Dumitrescu, A., Tóth, C. (2014). Covering Grids by Trees. *26th Canadian Conference on Computational Geometry*.

[8] Keszegh, B. (2013). Covering Paths and Trees for Planar Grids. *arXiv*, 3 Nov. 2013, `https://arxiv.org/abs/1311.0452`

[9] Kihn, M. (1995). Outside the Box: The Inside Story. *FastCompany*.

[10] Kranakis, E., Krizanc, D., Meertens, L. (1994). Link length of rectilinear Hamiltonian tours in grids. *Ars Combinatoria* **38**, 177–192.

[11] Hohenwarter, M., Borcherds, M., Ancsin, G., Bencze, B., Blossier, M., Elias, J., Frank, K., Gal, L., Hofstaetter, A., Jordan, F., Karacsony, B., Konecny, Z., Kovacs, Z., Kuellinger, W., Lettner, E., Lizelfelner, S., Parisse, B., Solyom-Gecse, C., Tomaschko, M. (2018). GeoGebra - Dynamic Mathematics for Everyone - version 6.0.507.0-w, *International GeoGebra Institute*, 16 Oct. 2018, `https://www.geogebra.org/`

[12] Loyd, S. (1914). Cyclopedia of Puzzles. *The Lamb Publishing Company*, p. 301.

[13] Lung, C.T., Dominowski, R.L. (1985). Effects of strategy instructions and practice on nine-dot problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **11(4)**, 804–811.

[14] Ripà, M. (2020). Solving the n_1 ≤ n_2 ≤ n_3 Points Problem for n_3 < 6. *ResearchGate*, 25 Jun. 2020, `https://www.researchgate.net/publication/342331014_Solving_the_n_1_n_2_n_3_Points_Problem_for_n_3_6`

[15] Ripà, M., Remirez, P. (2013). The Nine Dots Puzzle extended to n X n X ... X n Points, *viXra*, 5 Sep. 2013, `https://vixra.org/pdf/1307.0021v4.pdf`

[16] Stein, C., Wagner, D.P. (2001). Approximation algorithms for the minimum bends traveling salesman problem. In: Aardal K., Gerards B. (eds) *Integer Programming and Combinatorial Optimization*. LNCS, vol. 2081, 406–421. Springer, Berlin.