Desktop Tower Defense Is NP-Hard

Vasin Suttichaya(✉)

Department of Computer Engineering, Faculty of Engineering,
Mahidol University, Salaya, Thailand
vasin.sut@mahidol.ac.th

Abstract. This paper proves the hardness of the Desktop Tower Defense game. Specifically, the problem of determining where to locate k turrets in the grid of size $n \times n$ in order to maximize the minimum distance from the starting point to the terminating point is shown to be NP-hard. The proof applied to the generalized version of the Desktop Tower Defense.

Keywords: Graph theory · Hamiltonian path · Complexity

1 Introduction

Tower Defense (TD) is a type strategy video game that concentrates on protecting some part of the territory from waves of enemies, also known as "*creeps*". Enemies always appear at the entrance and attempt to walk to the exit point. The player must plan defensive strategies for protecting their bases, usually achieved by placing turrets alongside the enemy's road. Winning TD can be a painful task since the player needs to concurrently optimize many factors, such as resource, location, and enemy's abilities.

Desktop Tower Defense (DTD) is a popular tower defense game. The major difference between classic TD and DTD is the player's ability to control the enemy's path from the starting point to the exit point. DTD allows the player to build turrets on any positions on the map. Turrets can be used as walls for blocking enemies, and force them to find the new shortest path to the exit point. The only limitation is the player cannot place turrets in the way such that they completely block the exit. Therefore, the best strategy for winning DTD is not only optimizing own resources, but also instantly extending the distance. The DTD's gameplay is illustrated in Fig. 1.

TD introduces many challenges in problem solving areas, such as resource allocation, and geometry problems. Therefore, it is interested by researchers in the artificial intelligence field. Avery et al. proposed a framework based on TD for testing artificial intelligence algorithms [3]. The dynamic difficulty adjustment of TD was proposed in [20]. The resource allocation algorithm for the turn-based game in [11] can also be applied to TD.

However, unlike other puzzle games such as Chess and Go, the hardness of TD and DTD is still unclear since they contain many sub-problems. This research attempts to formally prove one of the sub-problems in DTD, the problem of placing turrets on the map that maximizing the distance of opponent's path toward the exit point.

M. Numao et al. (Eds.): PRICAI 2016 Workshops, LNAI 10004, pp. 19–25, 2017.
DOI: 10.1007/978-3-319-60675-0_2

Fig. 1. Desktop tower defense gameplay

This paper is organized into 5 sections. Section 2, some mathematical notations are presented. The hardness of DTD is proved in Sect. 3. The analysis and discussion are elaborated in Sect. 4. The conclusion of this research is drawn in Sect. 5.

2 Preliminaries

This section starts by defining grid graph's terminology and the Hamiltonian path on the general grid graph. Then, the hardness of many games and puzzles are reviewed.

2.1 Grid Graph Terminology

Suppose G^{∞} is the infinity graph with vertex set contains all points of the Euclidean plane with integer coordinates. Any two vertices in G^{∞} are connected if and only if the Euclidean distance between them is 1. Let $v = \left(v_x, v_y\right)$ be a vertex in G^{∞} such that v_x and v_y are integer coordinates of v in G^{∞}.

For any positive integer m and n, let $R(m, n)$ be the rectangular grid graph, the grid graph whose vertex set is $V(R(m, n)) = \left\{v | 1 \leq v_x \leq m, 1 \leq v_y \leq n\right\}$.

The arbitrary grid graph G is a finite vertex-induced subgraph of G^{∞}. Clearly, each vertex in arbitrary grid graph has degree at most 4. In other words, the arbitrary grid graph G is the subgraph isomorphism of the rectangular grid graph $R(m, n)$.

Let $G = (V, E)$ be an undirected graph, and let $s, t \in V$ be distinct vertices of G. The Hamiltonian path problem, HamPath(G, s, t), has a solution if there exists a path from s to t that visits each node in G exactly once. In the decision version, HamPath(G, s, t), is used to determine whether there is a path from s to t that visits each node in G exactly once.

The problem of finding Hamiltonian path in the arbitrary grid graph is known to be NP-complete [10]. However, there exists linear-time algorithms for some special class of grid graphs [5, 14, 21, 23].

2.2 The Hardness of Games and Puzzles

Many classic board games were proven to be EXPTIME-complete, such as Chess [19], Go [17], Chinese checkers [12], and draughts [18]. Several metatheorems for proving the hardness of modern video games were established in [8, 22]. Some modern video games, such as Price of Persia and Doom, were proven to be PSPACE-complete. Many video games, such as Tetris and Super Mario Bros, were proven to be NP-hard as well [2, 4]. Kendal provided the survey of NP-complete puzzles in [13].

3 NP-Hardness of Desktop Tower Defense

This section starts by formally defining DTD in the term of mathematical modeling. Then, the hardness of DTD is proven.

3.1 Desktop Tower Defense Problem Definition

Let m, n, k be some positive integers. Suppose T is a rectangle grid of size $m \times n$. Without loss of generality, assume that each element in T is indexed by row-major order (i.e., the square grid's index starts from $T[1][1]$ at the upper left corner to $T[m][n]$ at the lower right corner). Each element in T is marked by 0 or 1, which indicates a path and a wall respectively. Let W be a set of positions in T such that the position (x, y) in T is marked as a wall. Namely,

$$W = \{(x, y)\,|T[\mathrm{x}][\mathrm{y}] = 1 \text{ where } x \leq m \text{ and } y \leq n\}.$$

Let $\mathbf{s} = (x_s, y_s)$ be a starting point and $\mathbf{t} = (x_t, y_t)$ be a terminating point in T, for some $1 \leq x_s, x_t \leq n$ and $1 \leq y_s, y_t \leq m$. The generalized DTD, $DtD(T, W, k, \mathbf{s}, \mathbf{t})$, is to determine where to locate k additional walls to the rectangle grid T so that they can maximize the shortest path from **s** to **t**. The only restriction is the position of all k walls must not completely block the terminating point **t**. Therefore, there always has at least one path from **s** to **t**.

The generalized DTD can be also stated in the decision form. Let d be a shortest distance from **s** to **t**. The decision version, denote as $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$, is to determine if there is a way to place k walls in T such that the shortest path is increased to d or more. The output is "*yes*" if there is a path of length at least d after placing k additional walls, and "*no*" otherwise.

3.2 The Hardness of Desktop Tower Defense

The NP-hardness of DTD can be shown by transforming an instance of the Hamiltonian path for an arbitrary grid graph problem to an instance of the generalized DTD. Formally, the arbitrary grid graph $G \subseteq R(m, n)$ is transformed to the rectangle grid of size $(2m - 1) \times (2n - 1)$ such that the solution of the generalized DTD yields the Hamiltonian path in the arbitrary grid graph G.

Theorem 1. The Generalized DTD is NP-hard.

Proof. Suppose that $G = (V, E) \subseteq R(m, n)$ be an arbitrary grid graph with $|E|$ edges and $|V|$ vertices. Given the instance of the Hamiltonian path problem *HamPath*(G, s, t), we construct the instance of DTD by first placing vertices and edges of G to the $(2m - 1) \times (2n - 1)$-rectangle grid in the way such that each vertex becomes a blank square and each edge becomes a blank square. Second, flag two squares that represent vertex s and vertex t as the starting point, $\mathbf{s}$, and terminating point, $\mathbf{t}$, respectively. The last step, fill the rest squares that do not flagged as a blank square with walls. For example, Fig. 2 illustrates the transformation of an arbitrary grid graph $G \subseteq R(4, 6)$ to (7×11)-rectangle grid T.

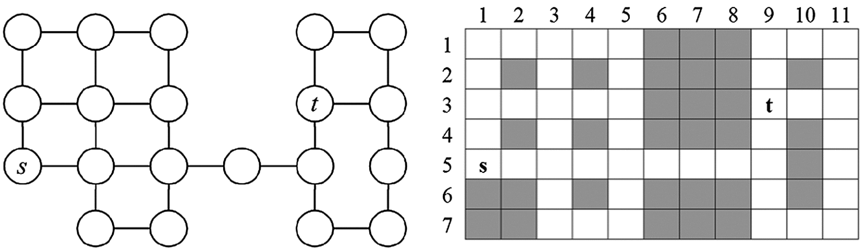

Fig. 2. Transform grid graph to DTD

To prove that the transform process is correct, it suffices to show that the original instance of *HamPath*(G, s, t) is a "*yes*" instance if and only if the transformed $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$ instance is also a "*yes*" instance. The proof is shown in the following lemma.

Lemma 1. *HamPath*(G, s, t) *has a solution if and only if* $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$, *where* $k = ||E| - (|V| - 1)|$ *and* $d = 2|V| - 1$, *has a solution.*

Proof. Suppose that *HamPath*(G, s, t) has a solution. There exists a path between vertex s and vertex t that visits each vertex in G exactly once. Note that, Hamiltonian path is also a longest path from s to t. This path has length exactly $|V| - 1$ since it needs to connect $|V|$ vertices without forming a cycle. This implies that $||E| - (|V| - 1)|$ edges are not included in the Hamiltonian path of the grid graph G. Each edge in G is represented by a blank square in T. This implies that if $||E| - (|V| - 1)|$ blank squares that are not included in Hamiltonian path are filled with walls. The shortest path in T must follow the Hamiltonian path of graph G. The distance in the transformed grid can be calculated by subtracting all blank squares with the number of walls. The minimum distance from $\mathbf{s}$ to $\mathbf{t}$ is

$$\begin{aligned} d &= |E| + |V| - k \\ &= |E| + |V| - (|E| - (|V| - 1)) \\ &= 2|V| - 1, \end{aligned}$$

since the shortest path must take all available cells. Therefore, the shortest path in $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$ is maximized by the longest path from s to t of G.

Suppose that $HamPath(G, s, t)$ has no solution. Then, the grid graph G does not have a Hamiltonian path between vertex s and vertex t. Therefore, there does not exist a path of length at least $|V| - 1$ that passes each node exactly once. This fact is also applied to the transformed grid T since all opponents in DTD always take the shortest path. They must not turn back to cells that have been passed. It follows that all paths from $\mathbf{s}$ to $\mathbf{t}$ in the transformed grid takes at most $2|V| - 1$ available cells. Thus, $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$ has no solution since it is impossible to get the minimum distance at least $2|V| - 1$ after placing $||E| - (|V| - 1)|$ walls. □

By Lemma 1, $HamPath(G, s, t) \leq_P DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$, and the result follows. □

4 Discussion

The proof in this research only considers the problem of maximizing the minimum distance. There are many problems that are embedded in the game.

In the real gameplay, there are many types of turrets. Each of them has its own firepower, ability, range, price, and cost of upgrading. The player should determine the number of turrets to buy so that the total price is less than or equal to the given gold. Moreover, the total firepower should be large enough for intercepting enemies. This problem can be classified as 0-1 Unbounded Multiple Constraint Knapsack Problem. Gens and Levner proved that this variant of Knapsack problem is NP-complete [9].

Enemies in DTD also have distinct abilities, such as the resistance to certain types of turrets, the weakness against some types of turrets, the ability to spawn itself after getting the damage, and the ability to fly over the map. The player must plan the defense strategy for the given combination of enemies. The hardness of planning can be shown to be NP-hard using the 3-SAT framework for proving Pushing Block puzzles [6, 7].

The problem of maximizing the shortest distance does not appear in the classic TD game. Enemies in the classic TD always walk on the predetermined road. The player cannot place any obstructions on this road. Therefore, the main problem in the classic TD is to find where to place turrets such that their ranges cover the road as much as possible. This problem can be seen as the special case of the Art Gallery problem. The Art Gallery problem and its variations are also shown to be NP-hard [1, 15, 16].

5 Conclusions

This research formally proves the hardness of maximizing the shortest path problem in the Desktop Tower Defense. The hardness of DTD follows from the NP-hardness of the Hamiltonian path problem. The proof shows that the instance of $HamPath(G, s, t)$ can be transformed to the instance of $DDtd(T, W, k, \mathbf{s}, \mathbf{t}, d)$ where the number of walls is $||E| - (|V| - 1)|$ and the minimum distance is $2|V| - 1$.

There are open problems related to DTD and TD that have not been proven yet. The first problem is the resource allocation problem. It is easy to see that this problem is similar to the Knapsack problem. The second problem is the area coverage problem. This problem resembling the Art Gallery problem. The major difference between the Art Gallery problem and the area coverage in TD game is sentinels in the Art Gallery problem must cover all internal regions in the polygon. In contrast, turrets in TD only need to cover some limited area around the polygon edge.

References

1. Aggarwal, A.: The art gallery theorem: ıts variations, applications and algorithmic aspects. Ph.D. thesis (1984)
2. Aloupis, G., Demaine, E.D., Guo, A., Viglietta, G.: Classic nintendo games are (computationally) hard. Theor. Comput. Sci. **586**, 135–160 (2015). http://dx.doi.org/10.1016/j.tcs.2015.02.037
3. Avery, P., Togelius, J., Alistar, E., van Leeuwen, R.P.: Computational intelligence and tower defence games. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2011, New Orleans, LA, USA, 5–8 June, 2011, pp. 1084–1091. IEEE (2011). http://dx.doi.org/10.1109/CEC.2011.5949738
4. Breukelaar, R., Demaine, E.D., Hohenberger, S., Hoogeboom, H.J., Kosters, W.A., Liben-Nowell, D.: Tetris is hard, even to approximate. Int. J. Comput. Geometry Appl. **14**(1–2), 41–68 (2004)
5. Chen, S.D., Shen, H., Topor, R.W.: An efficient algorithm for constructing hamiltonian paths in meshes. Parallel Comput. **28**(9), 1293–1305 (2002)
6. Demaine, E.D., Demaine, M.L., Hoffmann, M., O'Rourke, J.: Pushing blocks is hard. Comput. Geom. **26**(1), 21–36 (2003). doi:10.1016/S0925-7721(02)00170-0
7. Demaine, E.D., Demaine, M.L., O'Rourke, J.: Pushpush and push-1 are nphard in 2d. In: Proceedings of the 12th Canadian Conference on Computational Geometry, Fredericton, New Brunswick, Canada, 16–19 August 2000 (2000). http://www.cccg.ca/proceedings/2000/26.ps.gz
8. Forišek, M.: Computational complexity of two-dimensional platform games. In: Boldi, P., Gargano, L. (eds.) FUN 2010. LNCS, vol. 6099, pp. 214–227. Springer, Heidelberg (2010). doi:10.1007/978-3-642-13122-6_22
9. Gens, G., Levner, E.: Complexity of approximation algorithms for combinatorial problems: a survey. SIGACT News **12**(3), 52–65 (1980). http://doi.acm.org/10.1145/1008861.1008867
10. Itai, A., Papadimitriou, C.H., Szwarcfiter, J.L.: Hamilton paths in grid graphs. SIAM J. Comput. **11**(4), 676–686 (1982)
11. Johnson, R.W., Melich, M.E., Michalewicz, Z., Schmidt, M.: Coevolutionary optimization of fuzzy logic intelligence for strategic decision support. IEEE Trans. Evol. Comput. **9**(6), 682–694 (2005)

12. Kasai, T., Adachi, A., Iwata, S.: Classes of pebble games and complete problems. In: Austing, R.H., Conti, D.M., Engel, G.L. (eds.) Proceedings 1978 ACM Annual Conference, Washington, DC, USA, 4–6 December 1978, vol. II, pp. 914–918. ACM (1978). http://doi.acm.org/10.1145/800178.810161
13. Kendall, G., Parkes, A.J., Spoerer, K.: A survey of np-complete puzzles. ICGA J. **31**(1), 13–34 (2008)
14. Keshavarz-Kohjerdi, F., Bagheri, A.: Hamiltonian paths in some classes of grid graphs. J. Appl. Math. 2012 (2012)
15. Lee, D.T., Lin, A.K.: Computational complexity of art gallery problems. IEEE Trans. Inform. Theory **32**(2), 276–282 (1986). http://dx.doi.org/10.1109/TIT.1986.1057165
16. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press Inc., New York (1987)
17. Robson, J.M.: The complexity of go. In: IFIP Congress. pp. 413–417 (1983)
18. Robson, J.M.: N by N checkers is exptime complete. SIAM J. Comput. **13**(2), 252–267 (1984). http://dx.doi.org/10.1137/0213018
19. Shannon, C.E.: Programming a Computer for Playing Chess. In: Levy, D. (ed.) Computer chess compendium, pp. 2–13. Springer, New York (1988)
20. Sutoyo, R., Winata, D., Oliviani, K., Supriyadi, D.M.: Dynamic difficulty adjustment in tower defence. In: Procedia Computer Science, pp. 435–444 (2015)
21. Umans, C., Lenhart, W.: Hamiltonian cycles in solid grid graphs. In: FOCS, pp. 496–505. IEEE Computer Society (1997)
22. Viglietta, G.: Gaming is a hard job, but someone has to do it! Theory Comput. Syst. **54**(4), 595–621 (2014). http://dx.doi.org/10.1007/s00224-013-9497-5
23. Zamfirescu, C., Zamfirescu, T.: Hamiltonian properties of grid graphs. SIAM J. Discrete Math. **5**(4), 564–570 (1992)

Springer

http://www.springer.com/978-3-319-60674-3

Trends in Artificial Intelligence: PRICAI 2016 Workshops
PeHealth 2016, I3A 2016, AIED 2016, AI4T 2016, IWEC
2016, and RSAI 2016, Phuket, Thailand, August 22-23,
2016, Revised Selected Papers
Numao, M.; Theeramunkong, T.; Supnithi, T.; Ketcham,
M.; Hnoohom, N.; Pramkeaw, P. (Eds.)
2017, XII, 187 p. 88 illus., Softcover
ISBN: 978-3-319-60674-3