



On the Cooperative Graph Searching Problem

Chin-Fu Lin, Ondřej Navrátil, and Sheng-Lung Peng^(✉)

National Dong Hwa University, Hualien, Taiwan
{810521001,810321005,slpeng}@mail.ndhu.edu.tw

Abstract. In this paper, we introduce a new variation of graph searching problem, namely, cooperative graph searching problem. We define that a searcher is isolated if there is no other searchers on its close neighborhood. In this variant, we add an additional constrain that every searcher would not be isolated after each searching step. Therefore, we can make sure that every searcher can be cooperated by another searcher. We prove that the cooperative graph searching problem is NP-complete on general graphs and propose polynomial-time algorithms for the problem on grid graphs.

Keywords: Graph searching · Mixed searching
Cooperative graph searching

1 Introduction

The *graph searching problem* was first proposed by Breisch [1]. Parsons contributed some earlier works for the problem [2,3]. In the problem, a graph G represents a system of tunnels. Initially, all the edges of G are contaminated by a gas. An edge is *cleared* by some operations on G . A cleared edge is *recontaminated* if there is a path from an uncleared edge to the cleared edge without any searchers. The objective is to use as few searchers as possible to make all edges be cleared. The allowable operations are as follows:

1. Place a searcher on a vertex.
2. Remove a searcher from a vertex.
3. Move a searcher along an edge.

If an edge is only cleared by moving a searcher along the edge, then it is called the *edge searching problem*. On the other hand, if an edge is cleared by having two searchers on both its two ends, then it is called the *node searching problem*. If both of clearing rules are allowed, then it is called the *mixed searching problem*. The mixed searching problem was first proposed by Takahashi et al. [4]. It is obvious that the mixed searching problem is a natural generalization of edge and node searching problems. The mixed search number of G , denoted by $ms(G)$, is the minimum number k such that G is k -searchable by mixed search rules.

These variants of graph searching problems are not only interesting theoretically, but also have applications on problems like embedding tree queries [5,6], key graph searching [7,8], selectivity estimation [9], subgraph matching [10,11], min cut finding [12], and so on.

Since graph searching problems are so powerful, there are many variants of searching game for finding more appropriate strategies proposed for modeling real world problems. Blin proposed a searching problem named *exclusive graph searching problem* [13]. Latter, Markou studied a monotone version of exclusive graph searching [14]. Exclusive graph searching is a variant of mixed searching with the following extra constrains:

1. Initially, place all the searchers in the graph.
2. Every vertex can contain only one searcher.

Dyer introduced the fast searching problem based on edge searching in which we can traverse each edge only once [15]. Xue proposed algorithms for some special graphs [16]. Even though there are so many variants of searching game, most of them put emphasis on efficiency. However, in the real world, we should care not only performance but also searchers' safety. A searcher is *isolated* if there is no another searcher on its neighbor vertex or on the same vertex. To avoid a secondary distress/injure on rescue works and polices' raids, each searcher has better not to be isolated.

For guaranteeing that every searcher will not be isolated, we define a variant of searching game, called the *cooperative graph searching problem*. Initially, we have a graph $G = (V, E)$ in which every edge $e \in E$ is contaminated. Our target is to clean all the edges and make sure all searchers are under cooperative for each searching step. The clearing rules are based on mixed searching. However, for making a possible cooperation between searchers. we have the following possible operations.

1. Place a searcher on a vertex.
2. Place two searchers on the end-vertices of an edge.
3. Remove a searcher from a vertex.
4. Remove two searchers from the end-vertices of an edge.
5. Move a searcher along an edge.

For each step, only one operation above is allowed. We use $S_i = (E_i, C_i)$ to denote the status after operation i is applied, where E_i (respectively, C_i) denotes the set of uncleared (respectively, cleared) edges. A sequence of S_0, S_1, \dots, S_r that clears the graph G is called a search strategy of G and is denoted as \mathcal{S} . By the definition, $S_0 = (E, \emptyset)$ and $S_r = (\emptyset, E)$. Note that $E_i \cap C_i = \emptyset$ and $E_i \cup C_i = E$ for each step i . Let $|S_i|$ denote the number of searchers on graph after step i . Let $|\mathcal{S}| = \max_i |S_i|$ denote the number of searchers used to clear G . A search strategy is *cooperative* if for each step there is a searcher on some vertex, then there is another searcher at its neighbor vertex or stay at the same vertex. A cooperative search strategy is *optimal* if it uses the minimum number of searchers to clear G . The cooperative graph searching problem on G is to find an optimal cooperative search strategy to clear G .

We say a graph G is k -searchable if we can clear G by using at most k searchers. Similarly, G is k -cooperative searchable if G can be cleared by using at most k searchers using a k -cooperative search strategy. The cooperative search number of G , denoted by $\text{cos}(G)$, is the minimum number k such that G is k -cooperative searchable. Thus the decision version of the cooperative graph searching problem is called the k -cooperative graph searching problem that asks whether G is k -searchable or not?

2 Preliminary Results

Let $G = (V, E)$ be a simple, finite, and undirected graph. For a vertex subset $W \subseteq V$, let $G[W]$ be a subgraph of G that is induced by W . A *clique* is a complete subgraph and an *independent set* is a subgraph that has no edge. Let K_n (respectively, P_n and C_n) be a complete (respectively, path and cycle) graph with n vertices. The following lemma shows some easy results.

Lemma 1. *The following statements are true.*

1. $\text{cos}(K_n) = n - 1$ for $n \geq 3$.
2. $\text{cos}(P_n) = 2$ for $n \geq 2$.
3. $\text{cos}(C_n) = 4$ for $n \geq 5$.

Proof. For K_n , it is not hard to check that $n - 2$ searchers are not enough to clear the graph. Since after the $n - 2$ searchers are on the graph, there are two remaining vertices that cannot be guarded and therefore no way can clear the edge between them without recontamination. However, we can clear K_n by the following strategy.

1. Firstly, place 2 searchers on the end-vertices of an edge.
2. Then, place the remaining $n - 3$ searchers on any unguarded vertices one by one.
3. Finally, move any one searcher to the remaining unguarded vertex.

Since K_n is a complete graph, all the searchers on vertices are adjacent. Thus, it is a cooperative search strategy.

For P_n , let $P_n = (v_1, v_2, \dots, v_n)$. To clear P_n , we first place two searchers on v_1 and v_2 . Then, move the searcher at v_1 to v_2 and then move a searcher on v_2 to v_3 . The following steps are similar. That is, these two searchers cooperatively search the path until to v_n . Thus $\text{cos}(P_n) = 2$.

For C_n , The idea is similar to the search strategy of P_n . However, to avoid recontamination, we need two team from different directions to clear the graph. By the cooperative rules, the minimum number of each team is 2. Thus $\text{cos}(C_n) = 4$ for $n \geq 5$. \square

Lemma 2. *Let G' be an induced subgraph of G . Then $\text{cos}(G') \leq \text{cos}(G)$.*

Proof. Consider a cooperative search strategy \mathcal{S} of no recontamination for clearing G . Since \mathcal{S} has no recontamination, all the searchers on a vertex can be moved/removed only when the vertex is clear, *i.e.*, no any contaminated incident edge. In \mathcal{S} , we keep each step for cleaning edges in G' . We delete those steps that clean edges which are not in G' . Thus, G' can be cleared using the modified strategy. Clearly, $\text{cos}(G') \leq \text{cos}(G)$. \square

Theorem 1. *For any graph $G = (V, E)$, $ms(G) \leq \text{cos}(G) \leq 2ms(G)$ and the bound is tight.*

Proof. Since cooperative graph searching problem is a special case of mixed searching problem, the lower bound $ms(G) \leq \text{cos}(G)$ is obvious. Now consider an optimal mixed search strategy \mathcal{S} for G . In \mathcal{S} , for each step if it violates the cooperative search rules, then we add an extra searcher to a neighbor of the target vertex which will violate the rules. For example, if we want to clear the edge uv by moving a searcher from u to v , then there are the following two cases that the cooperative rules will be violated.

1. the new coming searcher at v is isolated.
2. the missing searcher of u causes an isolated searcher on a neighbor of u .

For both cases, we can just place an extra searcher to v such that the edge uv is cleared by both of its two ends having a searcher. Thus our upper bound is obtained.

Finally, by Lemma 1, for $n \geq 3$ $ms(K_n) = \text{cos}(K_n) = n - 1$ which meets the lower bound. Similarly, for $n \geq 4$ $\text{cos}(C_n) = 2ms(C_n) = 4$ which meets the upper bound. Thus the bounds are tight. \square

Corollary 1. *For any tree T , $\text{cos}(T) \leq 2ms(T) = O(\lg n)$.*

Lemma 3. *Let $G = (V, E)$ be a graph containing two cliques K_m and K_n such that $V = K_m \cup K_n$ and $m \geq n \geq |K_m \cap K_n|$. Then, $\text{cos}(G) = m - 1$.*

Proof. Since $K_m \subseteq G$, by Lemmas 1 and 2, $\text{cos}(G) \geq m - 1$. To show that $\text{cos}(G) = m - 1$, we propose the following strategy to clear G by using $m - 1$ searchers.

1. Let u be in $K_m \cap K_n$ and v be in $K_n \setminus K_m$.
2. Place $m - 1$ searchers on $K_m \setminus \{u\}$.
3. Move one searcher in $K_m \setminus K_n$ to u .
4. Remove all the searchers from $K_m \setminus K_n$.
5. Place searchers on $K_n \setminus (K_m \cup \{v\})$.
6. Move any searcher on G to v .

For the above strategy, it is easy to check that we use at most $m - 1$ searchers to clear G if $m \geq n$. It proves the lemma. \square

A graph $G = (V, E)$ is a split graph if V can be partitioned into two sets C and S such that the induced subgraph $G[C]$ is a clique and $G[S]$ is an independent set. For convenience, we use $G = (C \cup S, E)$ to denote a split graph.

Theorem 2. *Let $G = (C \cup S, E)$ be a split graph. Then $|C| - 1 \leq \text{cos}(G) \leq |C| + 1$.*

Proof. Since $G[C]$ is a clique of G , by Lemmas 1 and 2, we obtain that $|C| - 1$ is a lower bound of $\text{cos}(G)$. On the other hand, we can clear G by using $|C| + 1$ searchers as follows.

1. Place $|C|$ searchers on vertices of C .
2. Place one extra searcher on S one after one.

It is not hard to check that the above strategy can clear G using $|C| + 1$ searchers. Therefore, we have this theorem. \square

3 NP-Completeness Result

A search strategy can be recontaminated if there is an edge which is cleared at step i but becoming unclear at step j for $j > i$. However, for edge searching, LaPaugh showed the following theorem.

Theorem 3 ([17]). *If graph G is k -searchable, then there is a search strategy using at most k searchers without recontamination.*

By using a similar argument as the proof of Theorem 3, we can show the following theorem. We omit the proof for this version.

Theorem 4. *If graph G is k -cooperative searchable, then there is a cooperative search strategy using at most k searchers without recontamination.*

By Theorem 4, we may assume that the cooperative search strategy we considered does not be recontamination.

Lemma 4. *The k -cooperative graph searching problem is in NP.*

Proof. For any graph $G = (V, E)$ and a given integer k , we design a nondeterministic polynomial-time algorithm that checks whether G is k -cooperative searchable or not. The detailed algorithm is shown in Algorithm 1. It shows the lemma. \square

Theorem 5 ([18]). *The mixed searching problem is NP-Complete.*

Theorem 6. *The k -cooperative graph searching problem is NP-Complete.*

Proof. By Lemma 4, the k -cooperative graph searching problem is in NP. To complete the proof, the remaining work is to show it is NP-hard. Our reduction is from the mixed searching problem.

For any graph $G = (V, E)$, we construct an extended graph $G' = (V', E')$ by adding a universal vertex u . That is, $V' = V \cup \{u\}$ and $E' = E \cup \{uv \mid v \in V\}$. We claim that G can be mixed searched using at most $k - 1$ searchers if and

Algorithm: k -cooperative searchable

Data: a graph $G = (V, E)$ and an integer k

Result: G is k -cooperative searchable or not.

$F \leftarrow E$;

while F is not empty **do**

 nondeterministically, select an uncleared edge $e = uv \in F$

if both end-vertices of e do not have a searcher **then**

 | place two searchers on the end-vertices of e

else

 assume a searcher is at u ;

if all the edges incident to u except uv are clear **then**

 | move the searcher at u to v

else

 | place a searcher on v

end

end

$F \leftarrow F \setminus \{e\}$;

if the number of searchers used is greater than k **then**

 | return *False*

else

 Remove searchers if they are not needed under the constraints of cooperative search rules;

end

end

return *True*;

Algorithm 1. NP algorithm for checking k -cooperative searchable.

only if G' can be cooperatively searched by at most k searchers. The only if part is easy. In mixed searching, if v is the first vertex that is guarded by a searcher, then we simultaneously place another searcher on vertex u . Thus, they are cooperative since for each searcher on a vertex of V there is always a searcher on u . Then following the mixed searching strategy on G we can clear G' obeying the cooperative searching rules.

On the other hand, if we can cooperatively search G' by using at most k searchers, then G can be mixed searched with $k-1$ searchers. Consider a cooperative search strategy \mathcal{S} that does not contain recontaminated edges by Theorem 4. Since u does not exist in G , we remove every step that clears an edge incident to u . Let the resulting strategy be \mathcal{S}' . By definition \mathcal{S}' is a mixed search strategy that clears G . Since $|\mathcal{S}| = k$ and u is a universal vertex in G' , when there are k searchers on G' , u must be guarded by one. Otherwise, we need extra searcher to guard u for maintaining recontamination. Therefore $|\mathcal{S}'| = k-1$. This proves the theorem. \square

4 Polynomial-Time Algorithm for Grid Graphs

Grid graphs are a class of graphs that is the graph Cartesian product of path graphs. Two dimensional grid graphs are the most representative case of grid

graphs and have many applications. For example, we can simulate roads network of modern planned cities by grid graphs, or simulate electronic circuits by grid graphs with graph embedding skill. By definition, we can represent any two dimensional grid graphs by a production of two paths P_m and P_n . We denote it by $G_{m \times n}$ assuming that $m \geq n$.

Theorem 7. *For any grid graph $G_{m \times n}$, $n \geq 3$, $\text{cos}(G_{m \times n}) = n + 1$.*

Proof. We assume that in $G_{m \times n}$ there are m columns and n rows. We propose the following strategy to clear $G_{m \times n}$. It works column by column. We first clear the first column by placing n searchers on vertices of it. Suppose that the i -th column is cleared. We are going to clear the $(i + 1)$ -th column. For clearing edges between i -th column and $(i + 1)$ -th column, and obeying cooperative rules, we need an extra searcher, namely, $(n + 1)$ -th searcher to complete the work. The clearing progress is shown in Fig. 1. Thus $\text{cos}(G_{m \times n}) \leq n + 1$. By applying a separator theorem [19] on $G_{m \times n}$, $\text{cos}(G_{m \times n}) \geq n$.

Assume that we want to clear the graph by using only n searchers. To clear edge between two columns, our searchers have to guard vertices in these two columns. Since we can only move one searcher in an operation, the first searcher moving (or placing) to $(i + 1)$ -th column have to have a neighbor searcher at the same row. Otherwise, it is not cooperative. Note that at this moment, all the n searchers on column i cannot be moved (or removed); otherwise, a recontamination occurs. Thus n searchers are not enough to clear the graph $G_{m \times n}$. Hence, $n < \text{cos}(G_{m \times n}) \leq n + 1$, i.e., $\text{cos}(G_{m \times n}) = n + 1$. \square

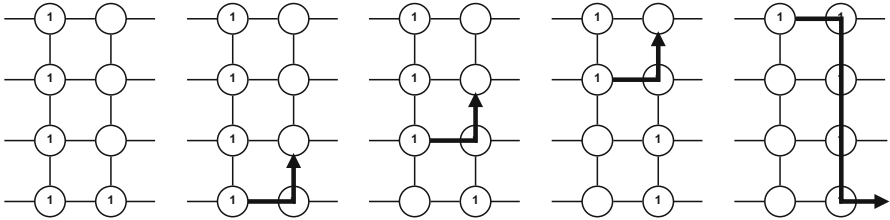


Fig. 1. A progress for clearing $G_{m \times n}$.

Grid graphs are an important class of interconnection networks. For a more general model, some vertices may be failure. That is, we can remove these failed vertices from G . To search such a grid, we use the same approach (column by column) mentioned above to clear it. Assume that all the vertices of the i -th column are good. Thus when we want to march to column $i + 1$, vertices on column $i + 1$ are either good or bad. It divides the column into a set of paths. In particular, some paths contain only one vertex. Thus for cooperative rules, we need a searcher stand at the vertex on column i for supporting the searcher marching to column $i + 1$. For those paths of length at least two, we use the same

technique to go to column $i + 1$. Assume that we have r P_1 's in column $i + 1$. By cooperative rules, we need r supporting vertices guarded on column i . The remaining problem is that do we have enough searchers to do the job? In the worst case, we have at least r missing vertices on column $i + 1$. Thus, n searchers are sufficient to guard vertices on column $i + 1$ and supporting vertices on column i . However, we still need one extra searcher to help searchers on column i to column $i + 1$. Therefore, we need $n + 1$ searchers for clearing this kind of grid graphs. For the other special cases, the arguments are similar. We omit the detail in this version. Finally, we have the following theorem.

Theorem 8. *Grid graphs $G_{m \times n}$ without some vertices are $(n + 1)$ -cooperative searchable.*

5 Conclusion

In this paper, we propose the cooperative graph searching problem, a new variant of graph searching problem by including cooperative rules. The rules make sure that no searcher will be alone or isolated during a searching process. We believe it is more suitable to model some real world problems. We propose some properties on graphs for this problem. In particular, we show that this problem is NP-complete on general graphs and it can be solved on grid graphs. In [16], the author solved the exclusive graph searching problem in polynomial time. We believe that by a similar idea our algorithm can be generalized for generalized grid graphs, *i.e.*, k dimensional grid graphs for any k . In the future, we want to study features of cooperative graph searching problem on other graph classes, *i.e.*, trees. We believe that this problem on trees can be solved in polynomial time.

Acknowledgement. This work was partially supported by the Ministry of Science and Technology of Taiwan, under Contract No. MOST 105-2221-E-259-018.

References

1. Breisch, R.: An intuitive approach to speleotopology. *Southwest. Cavers* **6**(5), 72–79 (1967)
2. Parsons, T.: Pursuit-evasion in a graph. In: Alavi, Y., Lick, D.R. (eds.) *Theory and Applications of Graphs*, pp. 426–441. Springer, Heidelberg (1976). <https://doi.org/10.1007/BFb0070400>
3. Parsons, T.: The search number of a connected graph. In: *Proceedings of the 9th Southeastern Conference on Combinatorics, Graph Theory and Computing*, pp. 549–554 (1978)
4. Takahashi, A., Ueno, S., Kajitani, Y.: Mixed searching and proper-path-width. *Theor. Comput. Sci.* **137**, 253–268 (1995)
5. Cole, R., Hariharan, R., Indyk, P.: Tree pattern matching and subset matching in deterministic $O(n \log^3 n)$ -time. In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithm*, pp. 245–254 (1999)

6. Hoffman, C., O'Donnell, J.: Pattern matching in trees. *J. ACM* **29**, 68–95 (1982)
7. Cook, D., Holder, L.: Substructure discovery using minimum description length and background knowledge. *J. Artif. Intell. Res.* **1**, 231–255 (1993)
8. Djoko, S., Cook, D., Holder, L.: An empirical study of domain knowledge and its benefits to substructure discovery. *IEEE Trans. Knowl. Data Eng.* **9**, 575–586 (1997)
9. Chen, Z., Korn, F., Koudas, N., Muthukrishnan, S.: Selectivity estimation for Boolean queries. In: *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 216–225 (2000)
10. Almohamad, H., Duffuaa, S.: A linear programming approach for the weighted graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 522–525 (1993)
11. Christmas, W., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.* **17**, 749–764 (1995)
12. Chung, M., Makedon, F., Sudborough, I., Turner, J.: Polynomial time algorithm for the min cut problem on degree restricted trees. *SAIM J. Comput.* **14**, 158–177 (1985)
13. Blin, L., Burman, J., Nisse, N.: Exclusive graph searching. In: Bodlaender, H.L., Italiano, G.F. (eds.) *ESA 2013*. LNCS, vol. 8125, pp. 181–192. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40450-4_16
14. Markou, E., Nisse, N., Pérennes, S.: Exclusive graph searching vs pathwidth. *Inf. Comput.* **252**, 243–260 (2017)
15. Dyer, D., Yang, B., Yaşar, Ö.: On the fast searching problem. In: Fleischer, R., Xu, J. (eds.) *AAIM 2008*. LNCS, vol. 5034, pp. 143–154. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68880-8_15
16. Xue, Y., Yang, B.: The fast search number of a Cartesian product of graphs. *Discrete Appl. Math.* **224**, 106–119 (2017)
17. LaPaugh, A.: Recontamination does not help to search a graph. *J. ACM* **40**, 224–245 (1993)
18. Yang, B., Cao, Y.: Monotonicity in digraph search problems. *Theor. Comput. Sci.* **407**, 523–544 (2008)
19. Escalante, F.: Schnittverbände in graphen. *Abh. Math. Semin. Univ. Hambg.* **38**, 199–220 (1972)