Operations Research Letters 43 (2015) 586–591

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Using shortcut edges to maximize the number of triangles in graphs

Sina Dehghani [b], Mohammad Amin Fazli [a,*], Jafar Habibi [a], Sadra Yazdanbod [c]

[a] Computer Engineering Department, Sharif University of Technology, Iran

[b] School of Computer Science, University of Maryland at College Park, United States

[c] School of Computer Science, Georgia Institute of Technology, United States

ARTICLE INFO

Article history:
Received 30 October 2014
Received in revised form
1 September 2015
Accepted 1 September 2015
Available online 16 September 2015

Keywords:
Triangles
Graphs
NP-complete
Approximation algorithm

ABSTRACT

In this paper, we consider the following problem: given an undirected graph $G = (V, E)$ and an integer k, find $I \subseteq V^2$ with $|I| \leq k$ in such a way that $G' = (V, E \cup I)$ has the maximum number of triangles (a cycle of length 3). We first prove that this problem is NP-hard and then give an approximation algorithm for it.

1. Introduction

The problem of augmenting networks in order to optimize their properties has been extensively tackled with a number of different approaches in recent years. The main goal of such augmentation is to improve network efficiency or constructing models with desired properties.

In general, to improve the network efficiency one would have to either change the transmission protocols [17,26,28] or change the underlying structure [4,8,32,9,20]. To support the latter approach, which is the main focus of this paper, an active line of research studies the impact of different structural properties on the performance of different network dynamics [14,29,1,31]. Therefore, to improve different network dynamics, we can optimize their associated structural properties.

The second important application of such network optimization problems, is to calibrate structural network models. These models are simply artificial graphs generated with real network properties and are used as a base for simulating different network dynamics. The main goal of these models is to study network behaviors under different conditions. Although numerous structural network models have been proposed over the years, none of them is complete because each focuses on only a subset of these properties and thus misses the others [30,3,2,18]. When a model N does not satisfy a property P, we can calibrate N by optimizing P with minor modifications to the structure of N.

While heuristics have been applied extensively for a wide range of network properties such as diameter and average path length [23], robustness [16,4,32,21] and synchronizability [8,21], approximation algorithms with guaranteed approximation factors have not received much attention. To the best of our knowledge, the only structural properties for which approximation algorithms and non-approximability results are proposed, are diameter, average path length [20,9,6,7] and Eulerian extension [10,15].

A high density of triangles (a cycle of length 3) is a beneficial structural property of graphs. The main behavior of graphs with this property is their fast collective dynamics [24,27]. Examples of such dynamics can be seen in a wide variety of fields such as relaxation oscillations in gene regulatory networks [19,11], synchronization in biological circuits [14,13], opinion formation in social networks [25] and consensus dynamics of agents in multi agent systems [22]. Thus, optimizing the number of triangles in networks with minor changes in their structure is an important problem.

In this paper, we concentrate on the problem of changing the structure of networks in a way that maximizes the number of their triangles. The change in the structure is done through drawing shortcut edges. We consider the limited budget case where we are only allowed to purchase at most k such shortcut edges.

Definition 1 (Triangle-Max *Problem*)**.** Given an undirected graph $G = (V, E)$ and an integer $k < \binom{|V|}{2} - |E|$. Find a set $I \subseteq V^2$ of at most k shortcut edges ($|I| \leq k$) such that $T(G')$ is maximized,

* Corresponding author.
E-mail address: mohammadamin.fazli@gmail.com (M.A. Fazli).

http://dx.doi.org/10.1016/j.orl.2015.09.003
0167-6377/

where $G' = (V, E \cup I)$ and $T(G')$ defines the number of triangles in G'.

In this paper, we first show that Triangle-Max is NP-hard. Then for instances of order n, we give a constant factor approximation algorithm for $k \geq n$ and an $O(n^{\frac{1}{4}})$-factor approximation algorithm for $k < n$.

2. Hardness

In this section, our main goal is to prove the NP-hardness of the Triangle-Max problem. First we define a modified class of the problem in which we only want to maximize the number of triangles with exactly i newly added edges.

Definition 2 (Triangle-Max$^{(i)}$ *(for* $1 \leq i \leq 3$*)*). Let $G = (V, E)$ be an undirected graph and $k < \binom{|V|}{2} - |E|$ be an integer. Find a set $I \subseteq V^2$ of at most k shortcut edges such that $T_i(G', I)$ is maximized where $G' = (V, E \cup I)$ and $T_i(G', I)$ defines the number of i-triangles in G' i.e. the triangles having exactly i edges in I.

Observation 1 shows that the Triangle-Max$^{(1)}$ problem can be solved in polynomial time. A simple greedy algorithm will work for this problem. For each shortcut edge $e = (u, v) \notin E$ with $u, v \in V$, define $F(e)$ to be the set of 1-triangles generated by drawing e. One can see that for each $e \neq e'$, $F(e) \cap F(e') = \emptyset$, therefore selecting k of these shortcut edges with maximum cardinality of F will obtain the optimal solution.

Observation 1. Triangle-Max$^{(1)}$ *is solvable in polynomial time.*

Although Triangle-Max$^{(1)}$ is in P, the other two problems in this class i.e. Triangle-Max$^{(2)}$ and Triangle-Max$^{(3)}$ are both NP-hard. Theorems 1 and 2 prove the hardness of these problems.

Theorem 1. Triangle-Max$^{(2)}$ *is NP-hard.*

Proof. We shall reduce the densest k-subgraph problem (DkS) to Triangle-Max$^{(2)}$ problem. The DkS problem is defined as follows: Given a graph G with n vertices and an integer $k \leq n$, the problem is to find a subgraph of G induced by k of its vertices with maximum number of edges. Let $G = (V, E)$ and k specify an instance of DkS. Assume that $V = \{v_1, v_2, \ldots, v_n\}$ and let x be the number of edges in the densest subgraph of G of size k.

Algorithm 1 Reducing DkS to Triangle-Max$^{(2)}$

input: G and k
output: One of the densest subgraphs of G with k vertices

1: Define $V' = \{v'_1, v'_2, \ldots, v'_n\}$ and $U = \{u_1, u_2, \ldots, u_{n^3}\}$. Let $V(G') = V' \cup U \cup \{v\}$.
2: For each $e = v_i v_j \in E(G)$ draw an edge between v'_i and v'_j such that $G'[V']$ and G are isomorphic ($G'[V']$ is the subgraph of G' which is induced by V').
3: Draw an edge between all pairs of vertices in U such that $G[U]$ becomes a clique with n^3 vertices.
4: Insert an edge between every two vertices $v'_i \in V'$ and $u_j \in U$.
5: Set $k' = n^3 + k$.
6: Solve Triangle-Max$^{(2)}$ on the input (G', k'). Let Q be the set of v's neighboring vertices in the returned solution.
7: Define T to be a set of k randomly selected vertices from $Q \setminus U$.
8: **return** vertices in G corresponding to those in T.

Algorithm 1 describes a polynomial-time reduction of the DkS problem to the Triangle-Max$^{(2)}$ problem. In the steps of 1 through 4 of this algorithm, an instance (G', k') of the Triangle-Max$^{(2)}$ problem will be built. G' would be a combination of a clique with

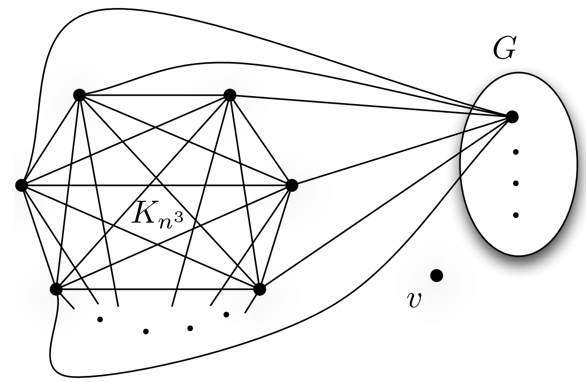

Fig. 1. G' graph.

n^3 vertices, an isomorphic graph to G whose vertices are connected to all vertices of the clique and an isolated vertex v (see Fig. 1).

We claim that any solution of Triangle-Max$^{(2)}$ to the instance (G', k') gives a solution for the DkS problem to the instance (G, k) by the steps 6 through 8 of Algorithm 1.

To prove this, first we need to show that all k' edges in the optimum solution (which we call OPT from now) of Triangle-Max$^{(2)}$ are adjacent to v. Let S be the set of edges in OPT which are not adjacent to v. Edges in S connect vertices in V', so $|S| \leq \binom{n}{2}$. Thus the number of edges adjacent to v is at least $n^3 + k - \binom{n}{2} \geq n^3 - n^2$. There is an optimal solution where all these edges are adjacent to vertices of U. We choose this optimal solution because in this case the maximum number of 2-triangles can be generated. Hence adding an edge to v would increase the number of 2-triangles by at least $n^3 - n^2$.

By removing S's edges, the number of 2-triangles would be decreased by at most $\binom{|S|}{2}$, because each pair of these edges can make at most one 2-triangle. Therefore removing edges in S and adding $|S|$ adjacent edges to v instead, the number of 2-triangles would be increased by at least

$$|S|(n^3 - n^2) - \binom{|S|}{2} \geq |S|\left(n^3 - n^2 - \frac{|S| - 1}{2}\right) \geq n^3 - \frac{3}{2}n^2,$$

which is greater than 0 for $n > 1$. Thus, $S = \emptyset$, i.e. all edges in OPT are adjacent to v.

Now, we prove that the set T returned by Algorithm 1 is one of the densest subgraphs for graph $G'[V']$ (and their corresponding vertices in $V(G)$ for G). Each edge in $G'[Q]$ is included in only one 2-triangle. So the number of 2-triangles created by the edges in OPT is equal to the number of edges in $G'[Q]$. First, notice that there exists a solution for Triangle-Max$^{(2)}$ to the instance (G', k') which creates $y = \binom{n^3}{2} + x + k \cdot n^3$ 2-triangles. It is enough to connect v to the vertices of $U \cup D$ where D is the set of vertices in the densest subgraph of $G'[V']$. We will show that OPT cannot generate more than y 2-triangles and if the equation holds, T must be a densest subgraph of $G'[V']$.

$T \subseteq V'$ and $|T| = k$, therefore the number of edges in $G'[T]$ is less than or equal to x. Also the number of edges in $G'[Q \setminus T]$ is less than or equal to $\binom{n^3}{2}$, because $Q \setminus T$ has exactly $k' - k = n^3$ vertices. Moreover the number of edges between these two subgraphs is less than or equal to $k \cdot n^3$. Thus the number of edges in $G'[Q]$ is less than or equal to y and the equality can only happen when T is a densest subgraph of $G'[V']$, $G'[Q \setminus T]$ is a clique and all vertices in T are connected to all vertices in $Q \setminus T$. □

Our next step is to prove the NP-hardness of the Triangle-Max$^{(3)}$ problem.

Theorem 2. Triangle-Max$^{(3)}$ *is NP-hard.*

Proof. We shall reduce the Maximum Independent Set problem to the Triangle-Max$^{(3)}$ problem. In the Maximum Independent Set problem, we are given an undirected graph $G = (V, E)$ and a number $k > 0$ and we want to find a subset $C \subseteq V(G)$ with $|C| = k$ for which $G[C]$ is an empty graph. Let $k' = \binom{k}{2}$. We show that the maximum independent set of G is equal to k if and only if the optimum solution to the instance (G, k') of the Triangle-Max$^{(3)}$ problem is equal to $\binom{k}{3}$.

Assume G has an independent set C of size k. Thus, there are k' edges available to build $\binom{k}{3}$ 3-triangles, i.e. adding edges in $\{(u, v)|u, v \in C\}$ to G builds $\binom{k}{3}$ 3-triangles. One can see that achieving more than $\binom{k}{3}$ 3-triangles is impossible.

To prove the only if part, assume that adding edges in I generates $\binom{k}{3}$ 3-triangles and define $G' = (V, E \cup I)$. We claim that there is no edge present in more than $\frac{3\binom{k}{3}}{\binom{k}{2}} = (k-2)$ 3-triangles. Assume to the contrary, there exists an edge $(u, v) \in I$ included in at least $(k-1)$ 3-triangles. Thus $deg_{G',I}(u) \geq k$ ($deg_{G',I}(u)$ is the degree of vertex u in graph G' through edges in I) because u and v must be connected to at least $(k-1)$ vertices by the edges in I to generate these 3-triangles. Therefore by deleting these edges, the remaining graph has $\binom{k}{2} - k = \binom{k-1}{2} - 1$ shortcut edges which can build less than $\binom{k-1}{3}$ 3-triangles. u can build at most $\binom{k}{2}$ triangles with its adjacent short-cut edges. Hence, the number of 3-triangles in G' is less than $\binom{k-1}{3} + \binom{k}{2} = \binom{k}{3}$ which is a contradiction.

Since the total number of shortcut edges in G' is $\binom{k}{2}$ and none of them is present in more than $k-2$ triangles, every shortcut edge in G' is included in exactly $k-2$ triangles. Also, as shown, for every vertex $x \in G'$, $deg_{G',I}(x) \leq k-1$. Thus $deg_{G',I}(x) = k-1$, so edges in I make a k-vertex clique which is equivalent to an independent set of size k in G. □

Theorem 3 is the main result of this section. In this theorem, we will show that the Triangle-Max problem is not solvable in polynomial time, unless $P = NP$.

Theorem 3. Triangle-Max *is NP-hard.*

Proof. We shall reduce the Triangle-Max$^{(3)}$ problem to Triangle-Max problem. Let $G = (V, E)$ and $0 < k < \binom{|V|}{2} - |E|$ provide an instance of Triangle-Max$^{(3)}$. We construct $G' = (V_1 \cup V_2, E')$ as follows.

For every vertex $u \in V$, we add a vertex u' in V_1 and for every pair of $u, v \in V$ with $(u, v) \notin E$, we add $|V|^3$ vertices to V_2 and connect them to both u' and v'. Therefore $|V_1 \cup V_2| = \left(\binom{|V|}{2} - |E|\right)|V|^3 + |V|$.

Now, consider the instance (G', k) of the Triangle-Max problem. Suppose that C is the set of edges of size k for which $G'' = (V_1 \cup V_2, E' \cup C)$ generates the maximum number of triangles. We prove that for every edge $e = (u', v') \in C$, $u', v' \in V_1$ and $(u, v) \notin E$. Assume to the contrary that there exists an edge $(w', z') \in C$ which does not satisfy these conditions. Since $k < \binom{|V|}{2} - |E|$, there exists another edge $(x', y') \notin C$, where $x', y' \in V_1$ and $(x, y) \notin E$. If one of the vertices w' or z' is in V_2, its degree in G'' would be at most $2 + k \leq |V|^2$. Suppose that this edge is present in b triangles. We have $b \leq \min\{deg_{G''}(w'), deg_{G''}(z')\} \leq |V|^2$. Thus replacing (w', z') by (x', y'), will lead to more triangles ($|V|^3$ triangles). Moreover, if $(w', z') \in E$, this edge would be present in at most $|V| + k \leq |V|^2$ triangles, which is less than the $|V|^3$ triangles added by (x', y').

For each edge $(u', v') \in C$ add its corresponding edge (u, v) to G. We will prove that this is an optimal solution for the (G, k) instance of the Triangle-Max$^{(3)}$ problem. First consider that by the above reasoning this solution is feasible. Since $G'[V_1]$ is initially an empty graph and all shortcut edges are added inside V_1, we will have no 2-triangle in G'. The number of 1-triangles in G' is exactly $k \cdot |V|^3$ and so is independent from the locations of C's edges in $G'[V_1]$. Therefore edges in C generate the maximum number of 3-triangles which can only be built in $G'[V_1]$. Thus since the solution is feasible, C's corresponding edges in G also generate the maximum number of 3-triangles in G. □

3. Approximation algorithm

In this section, we provide an approximation algorithm for the Triangle-Max problem. To reach this aim, we also take the Triangle-Max$^{(i)}$ problems into consideration.

Assume that a graph G and a number k is given. Let OPT denote the number of triangles in the optimal solution of the Triangle-Max problem. Similarly define $OPT^{(i)}$ to be the number of i-triangles in the optimal solution of the Triangle-Max$^{(i)}$ problem. The general idea is to find sets $I_i \subseteq V^2$ and numbers $f_i > 0$ for $1 \leq i \leq 3$, such that $\forall_{1\leq i\leq 3}|I_i| \leq k$ and the number of triangles in $G_i = (V, E \cup I_i)$ is at least $\frac{1}{f_i} \cdot OPT^{(i)}$ and then compute the final algorithm by combining these intermediate solutions.

From Observation 1, we know that Triangle-Max$^{(1)}$ is solvable in polynomial time. We solve this problem and set I_1 equal to the optimum solution of this problem. We will have

$$T(G_1) \geq OPT^{(1)},$$

where $G_1 = (V, E \cup I_1)$.

Now we focus on finding the set I_3. Theorem 4 shows that choosing an arbitrary subset of vertices and adding (or redrawing) k of its inside edges will suffice.

Theorem 4. *Given an undirected graph $G = (V, E)$ and an integer k, there is a polynomial time algorithm which finds a set of shortcut edges I_3, such that $|I_3| \leq k$ and $T(G_3) \geq \frac{1}{4}OPT^{(3)}$, where $G_3 = (V, E \cup I_3)$.*

Proof. Assume that $\binom{k'}{2} \leq k < \binom{k'+1}{2}$. Let $S \subseteq V$ be an arbitrary subset of V of size k'. For every pair of vertices $u, v \in S$ add (u, v) to I_3, hence $|I_3| = \binom{k'}{2}$ and there are at least $\binom{k'}{3}$ triangles in G_3. One can easily show that $OPT^{(3)} < \binom{k'+1}{3}$. If $k' < 3$, then $k < 3$ and $OPT^{(3)} = 0$, therefore we assume $k' \geq 3$, i.e., $\binom{k'+1}{3}/\binom{k'}{3} \leq 4$. Thus we can conclude

$$T(G_3) \geq \binom{k'}{3} \geq \frac{1}{4}\binom{k'+1}{3} \geq \frac{1}{4}OPT^{(3)}. \quad \square$$

Finding the set I_2 is the most difficult part. Define $\mathcal{E}_G(S)$ as the set of G's edges induced by $S \subseteq V(G)$ and

$$dns_G(S) = \frac{|\mathcal{E}_G(S)|}{|S|}$$

as the density of edges in set S. Moreover assume that $N_{G',I}(x)$ is the set of x's neighbors in G' through edges in I. Lemma 1 gives another formulation for $T_2(G', I)$ (the number of 2-triangles) and an upper bound for $T(G')$, where $G' = (V, E \cup I)$. Consider that in the second part of this lemma $I \cap E$ may be non-empty.

Lemma 1. *Let $G = (V, E)$ be an undirected graph and I be a set of shortcut edges. Let $G' = (V, E \cup I)$. We have the following statements:*

1. *If $I \cap E = \emptyset$, then $T_2(G', I) = \sum_{(u,v)\in I} dns_G(N_{G',I}(u)) + dns_G(N_{G',I}(v))$.*
2. *$T(G') \geq \frac{1}{3}\sum_{(u,v)\in I} dns_G(N_{G',I}(u)) + dns_G(N_{G',I}(v))$.*

Proof. First assume that $I \cap E = \emptyset$. Each 2-triangle of I in G can be represented using the joint vertex of its two edges of I.

Therefore the number of triangles corresponding to vertex x is equal to $|\mathcal{E}_G(N_{G',I}(x))|$. Thus,

$$\begin{aligned} T_2(G,I) &= \sum_{x\in V} |\mathcal{E}_G(N_{G',I}(x))| = \sum_{x\in V}\sum_{y\in N_{G',I}(x)} \frac{|\mathcal{E}_G(N_{G',I}(x))|}{|N_{G',I}(x)|} \\ &= \sum_{(u,v)\in I} \frac{|\mathcal{E}_G(N_{G',I}(u))|}{|N_{G',I}(u)|} + \frac{|\mathcal{E}_G(N_{G',I}(v))|}{|N_{G',I}(v)|} \\ &= \sum_{(u,v)\in I} dns_G(N_{G',I}(u)) + dns_G(N_{G',I}(v)). \end{aligned}$$

For the second part, consider a 2-triangle T created by three edges (a,b), (b,c) and (a,c). In the sum $\sum_{(u,v)\in I} dns_G(N_{G',I}(u)) + dns_G(N_{G',I}(v))$, T is counted at most three times and this happens when $(a,b), (b,c), (a,c) \in I \cap E$. $\square$

Algorithm 2 Finding I_2

input: G, k, S

1: $I_2' \leftarrow \emptyset$
2: **for all** $u \in V(G)$ **do**
3: $mark[u] \leftarrow false$
4: **end for**
5: **while** there exists a node $u \in V(G) \setminus S$ with $mark[u]$ =false **and** $|I_2'| + |S| \le k$ **do**
6: $mark[u] = true$
7: **for all** $v \in S$ **do**
8: Add edge (u,v) to I_2'
9: **end for**
10: **end while**
11: **if** $|I_2'| \ne |S||V \setminus S|$ **then**
12: **return** I_2'
13: **end if**
14: $I_2'' \leftarrow \emptyset$
15: $k_1 \leftarrow k - |I_2'|$
16: **for all** $u \in V(G)$ **do**
17: $mark[u] \leftarrow false$
18: **end for**
19: $v_{max} \leftarrow$ a vertex $x \in S$ with maximum degree $\Delta = \max deg_S(x)$
20: $freeNodes \leftarrow |S|$ ▷ $freeNodes$ denotes the number of unmarked vertices in S
21: **while** there is a node $u \ne v_{max} \in S$ with $mark[u]$=$false$ **and** $|I_2''| + freeNodes - 1 \le k_1$ **do**
22: $mark[u] = true$
23: $freeNodes \leftarrow freeNodes - 1$
24: **for all** $v \in S$ **do**
25: **if** $mark[v] = false$ **then**
26: Add edge (u,v) to I_2''
27: **end if**
28: **end for**
29: **end while**
30: **if** $|I_2''| \ne \binom{|S|}{2}$ **then**
31: **return** $I_2' \cup I_2''$
32: **end if**
33: $I_2''' \leftarrow \emptyset$
34: $k_2 \leftarrow k - |I_2'| - |I_2''|$
35: **while** $|I_2'''| < k_2$ **and** there exist two vertices $u, v \in V(G) \setminus S$ such that $(u,v) \notin E(G) \cup I_2'''$ **do**
36: Add edge (u,v) to I_2'''
37: **end while**
38: **return** $I_2' \cup I_2'' \cup I_2'''$

Define $dns^k_{\max}(G)$ to be the density of the densest subgraph of G with at most k vertices. The following theorem is critical for finding the set I_2.

Theorem 5. *Given an undirected graph $G = (V,E)$, an integer k, and a set of vertices $S \subseteq V$ where $|S| \le k$, there is a polynomial time algorithm which finds a set of shortcut edges I_2, such that $|I_2| \le k$ and $T(G_2) \ge \lfloor \frac{1}{24} \frac{dns_G(S)}{dns^k_{\max}(G)} OPT^{(2)} \rfloor$, where $G_2 = (V, E\cup I_2)$.*

Proof. We show that Algorithm 2 satisfies the conditions of this theorem. So we set I_2 to be the returned edge set of this algorithm. By using the first part of Lemma 1, we have

$$2k \cdot dns^k_{\max}(G) \ge OPT^{(2)}. \quad (1)$$

Algorithm 2 has three phases in which 3 sets I_2', I_2'' and I_2''' of shortcut edges will be computed and added to the result set. When the cardinality of the result set becomes k, the algorithm terminates. In the first phase, shortcut edges are chosen from edges between S and $V \setminus S$. In the second and third phases, these edges are chosen from the inside edges of S and $(V \setminus S)$, respectively. In the following, we count the number of 2-triangles created by this process and compute a lower-bound for each phase.

Consider the lines 1–13 of this algorithm, and assume that $G_2' = (V, E \cup I_2')$. In each iteration of the while loop in lines 5–10, Algorithm 2 chooses a vertex $u \in V - S$ and for every vertex $v \in S$ adds the edge (u,v) to I_2', hence $|S|$ edges will be added to I_2'. These edges result in $|\mathcal{E}_G(S)|$ triangles in G_2'. This process will continue until $|S| + |I_2'| \ge k$.

If $S = V$, no edge will be chosen to be added to I_2' and thereafter $|I_2'| = |S|\,|V \setminus S| = 0$ and I_2' will not be returned by the algorithm. So assume that $S \ne V$. By the above discussion, at the end of the while loop, the number of triangles in G_2' will be at least

$$T(G_2') \ge \frac{|I_2'|}{|S|} |\mathcal{E}_G(S)| = |I_2'| dns_G(S).$$

If $|I_2'| \ne |S|\,|V - S|$ then I_2' will be returned by the algorithm, thus $I_2 = I_2'$ and $G_2 = G_2'$. At the end of the while loop, we have $|I_2'| + |S| \ge k$. We know $|I_2'| \ge S$, thus $|I_2'| \ge \frac{1}{2}k$. Applying Eq. (1), we conclude

$$T(G_2) = T(G_2') \ge \frac{k}{2} dns_G(S) \ge \frac{1}{4}\frac{dns_G(S)}{dns^k_{\max}(G)} OPT^{(2)},$$

as desired. Now we assume $|I_2'| = |S|\,|V \setminus S|$. Hence every vertex in $V \setminus S$ is connected to all vertices in S through edges in I_2'.

Now consider lines 14–32 of Algorithm 2 and define $G_2'' = (V, E \cup I_2' \cup I_2'')$. In each iteration of the while loop in lines 21–29, Algorithm 2 picks a vertex $u \ne v_{\max} \in S$ and for every other vertex $v \in S$ adds a shortcut edge (u,v) to I_2''. Assume that the chosen vertex is u and the H is the induced subgraph of G_2'' on S. So $N_{H,I_2''}(u) = S \setminus \{u\}$. We show that $dns_G(S \setminus \{u\}) \ge \lfloor \frac{1}{2} dns_G(S) \rfloor$ for every chosen vertex u by the lines 14–32 of Algorithm 2. This proves that for each edge $(u,v) \in I_2''$, $dns_G(N_{H,I_2''}(u)) + dns_G(N_{H,I_2''}(v)) \ge \lfloor \frac{1}{2} dns_G(S) \rfloor$, as at least one of the two end points was chosen in the algorithm as vertex u. Thus by Lemma 1 part 2, we have,

$$\begin{aligned} T(H) &\ge \frac{1}{3} \sum_{(u,v)\in I_2''} dns_G(N_{H,I_2''}(u)) + dns_G(N_{H,I_2''}(v)) \\ &\ge \left\lfloor \frac{1}{6} |I_2''| dns_G(S) \right\rfloor. \end{aligned}$$

To prove the claim, consider first the case when $v_{\max}$ is the only vertex with degree Δ. Since $u \ne v_{\max}$, we have $deg_G(u) < \delta$ and thus $dns_G(S \setminus \{u\}) \ge \frac{1}{2} dns_G(S)$, which proves the claim.

If $deg_G(u) = \Delta$, we know that $v_{\max} \in S \setminus \{u\}$ and its degree is Δ. Thus, if the number of edges in $G[S \setminus \{u, v_{\max}\}]$ is K, $dns_G(S \setminus \{u\}) \ge \frac{\Delta - 1 + K}{|S|-1}$ which is greater than or equal to $\lfloor \frac{2\Delta + K}{2|S|} \rfloor \ge \lfloor \frac{1}{2} dns_G(S) \rfloor$ for $|S| \ge 2$.

If $|I_2''| \neq \binom{|S|}{2}$ then $I_2 = I_2' \cup I_2''$ will be returned by the algorithm. $|I_2'| + |I_2''| \geq \frac{1}{2}k$ because $|I_2''| + |S| > k_2 = k - |I_2'|$. Thus, G_2'' has at least $|I_2'|dns_G(S)$ triangles with exactly two vertices in S and at least $\frac{1}{6}|I_2''|dns_G(S)$ triangles with exactly three vertices in S. Therefore, we have

$$\begin{aligned} T(G_2) = T(G_2'') &\geq \left\lfloor \left(|I_2'| + \frac{1}{6}|I_2''|\right) dns_G(S) \right\rfloor \\ &\geq \left\lfloor \frac{1}{6}(|I_2'| + |I_2''|)dns_G(S) \right\rfloor \\ &\geq \left\lfloor \frac{1}{12}k \cdot dns_G(S) \right\rfloor \\ &\geq \left\lfloor \frac{1}{24}\frac{dns_G(S)}{dns_{\max}(G)} OPT^{(2)} \right\rfloor . \end{aligned}$$

Thus we assume $|I_2''| = \binom{|S|}{2}$, therefore every two vertices in S are connected through edges in I_2''. Now consider lines 33–38 of the algorithm. When Algorithm 2 reaches these lines, $I_2 = I_2' \cup I_2'' \cup I_2'''$ will be returned, therefore $G_2 = (V, E \cup I_2' \cup I_2'' \cup I_2''')$. Assume that (u, v) with $u, v \in V \setminus S$ is an arbitrary edge in I_2'''. Since for every vertex $x \in S$, $(u, x), (v, x) \in E \cup I_2' \cup I_2''$, there are $|S|$ triangles with (u, v) and two edges between $V \setminus S$ and S. Thus we can conclude that $a = |I_2'''|\,|S|$ of such triangles exist in G_2. Since $dns_G(S) \leq \frac{|S|-1}{2}$, we have $a \geq 2|I_2'''|dns_G(S)$. Seeing that $k = |I_2'| + |I_2''| + |I_2'''|$, we have

$$\begin{aligned} T(G_2) &\geq \left\lfloor \frac{1}{6}(|I_2'| + |I_2''| + |I_2'''|)dns_G(S) \right\rfloor \\ &\geq \left\lfloor \frac{1}{6}k \cdot dns_G(S) \right\rfloor \\ &\geq \left\lfloor \frac{1}{12}\frac{dns_G(S)}{dns_{\max}(G)} OPT^{(2)} \right\rfloor . \quad \square \end{aligned}$$

Theorem 6. *Given an undirected graph $G = (V, E)$ of order n and an integer k. There is a polynomial time approximation algorithm which finds a set of shortcut edges I, such that $|I| \leq k$ and*

- *for $k \geq n$, $T(G') \geq \frac{1}{30}OPT$*
- *for $k < n$, $T(G') \geq c \cdot n^{-\frac{1}{4}}OPT$ ($c \in O(1)$)*

where $G' = (V, E \cup I)$ and OPT is the optimal solution of the Triangle-Max *problem.*

Proof. Let t_i be the number of i-triangles in the optimal solution, thus $t_0 + t_1 + t_2 + t_3 = OPT$. From Observation 1, Theorems 5 and 4, we can find sets $I_i \subseteq V^2$ for $1 \leq i \leq 3$, such that $\forall_{1 \leq i \leq 3} |I_i| \leq k$ and

$$\begin{aligned} &T(G_1) \geq OPT^{(1)} \geq t_1, \\ &24f \cdot T(G_2) \geq OPT^{(2)} \geq t_2, \\ &4T(G_3) \geq OPT^{(3)} \geq t_3, \end{aligned}$$

where $G_i = (V, E \cup I_i)$ and $f = \frac{dns^k_{\max}(G)}{dns_G(S)}$. Therefore,

$$T(G) + T(G_1) + 24f \cdot T(G_2) + 4T(G_3) \geq t_0 + t_1 + t_2 + t_3 = OPT.$$

Let $m = \max\{T(G), T(G_1), T(G_2), T(G_3)\}$. Thus $(6 + 24f)m \geq OPT$ and by choosing the set with maximum number of triangles, we have a $(6 + 24f)$-factor approximation for the Triangle-Max problem.

To have an upper bound for f, we must compute the densest subgraph of G with maximum k vertices. Existing algorithms provide the optimal value of $dns^k_{\max}(G)$ for $k \geq n$ [12] (i.e. there is no limitation on k) and an $O(n^{\frac{1}{4}})$-factor solution for $k < n$ [5] which finalizes the proof. $\square$

4. Conclusion

An important follow-up work is to consider optimizing other essential graph properties by limited number of structural modifications. Solving these problems may have applications in scientific and technological contexts.

References

[1] D. Acemoglu, A. Ozdaglar, A. Tahbaz-Salehi, Cascades in networks and aggregate volatility, Technical Report, National Bureau of Economic Research, 2010.
[2] W. Aiello, F. Chung, L. Lu, A random graph model for massive graphs, in: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, ACM, 2000, pp. 171–180.
[3] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.
[4] A. Beygelzimer, G. Grinstein, R. Linsker, I. Rish, Improving network robustness by edge modification, Physica A 357 (2005) 593–612.
[5] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, A. Vijayaraghavan, Detecting high log-densities: an $o(n^{1/4})$ approximation for densest k-subgraph, in: Proceedings of the 42nd ACM Symposium on Theory of Computing, 2010, pp. 201–210.
[6] D. Bilò, L. Gualà, G. Proietti, Improved approximability and non-approximability results for graph diameter decreasing problems, Theoret. Comput. Sci. 417 (2012) 12–22.
[7] V. Chaoji, S. Ranu, R. Rastogi, R. Bhatt, Recommendations to boost content spread in social networks, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 529–538.
[8] M. Dadashi, I. Barjasteh, M. Jalili, Rewiring dynamical networks with prescribed degree distribution for enhancing synchronizability, Chaos 20 (2010) 043119.
[9] E. Demaine, M. Zadimoghaddam, Minimizing the diameter of a network using shortcut edges, in: Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, 2010, pp. 420–431.
[10] F. Dorn, H. Moser, R. Niedermeier, M. Weller, Efficient algorithms for eulerian extension and rural postman, SIAM J. Discrete Math. 27 (2013) 75–94.
[11] T. Gardner, D. di Bernardo, D. Lorenz, J. Collins, Inferring genetic networks and identifying compound mode of action via expression profiling, Sci. Signal 301 (2003) 102.
[12] A. Goldberg, Finding a maximum density subgraph, Technical Report, Computer Science Division, University of California, 1984.
[13] C. Grabow, S. Grosskinsky, M. Timme, Speed of complex network synchronization, Eur. Phys. J. B 84 (2011) 613–626.
[14] C. Grabow, S. Hill, S. Grosskinsky, M. Timme, Do small worlds synchronize fastest? Europhys. Lett. 90 (2010) 48002.
[15] W. Höhn, T. Jacobs, N. Megow, On Eulerian extensions and their application to no-wait flowshop scheduling, J. Sched. 15 (2012) 295–309.
[16] Z. Jiang, M. Liang, D. Guo, Enhancing network performance by edge addition, Internat. J. Modern Phys. C 22 (2011) 1211–1226.
[17] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Commun. Mag. 51 (2013) 114–119.
[18] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, Z. Ghahramani, Kronecker graphs: An approach to modeling networks, J. Mach. Learn. Res. 11 (2010) 985–1042.
[19] D. McMillen, N. Kopell, J. Hasty, J. Collins, Synchronizing genetic relaxation oscillators by intercell signaling, Proc. Natl. Acad. Sci. USA 99 (2002) 679–684.
[20] A. Meyerson, B. Tagiku, Minimizing average shortest path distances via shortcut edge addition, in: Proceedings of the 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2009, pp. 272–285.
[21] I. Mishkovski, M. Righero, M. Biey, L. Kocarev, Enhancing robustness and synchronizability of networks homogenizing their degree distribution, Physica A 390 (2011) 4610–4620.
[22] R. Olfati-Saber, Ultrafast consensus in small-world networks, in: Proceedings of the American Control Conference, 2005, pp. 2371–2378.
[23] M. Papagelis, F. Bonchi, A. Gionis, Suggesting ghost edges for a smaller world, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 2011, pp. 2305–2308.
[24] A. Pikovsky, M. Rosenblum, J. Kurths, Synchronization: A Universal Concept in Nonlinear Sciences, Cambridge University Press, 2003.
[25] A. Pluchino, V. Latora, A. Rapisarda, Changing opinions in a changing world: A new perspective in sociophysics, Internat. J. Modern Phys. C 16 (2005) 515–531.
[26] L. Ravindranath, C. Newport, H. Balakrishnan, S. Madden, Improving wireless network performance using sensor hints, in: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, 2011, pp. 21–21.
[27] S. Strogatz, Exploring complex networks, Nature 410 (2001) 268–276.
[28] S. Vissicchio, L. Vanbever, C. Pelsser, L. Cittadini, P. Francois, O. Bonaventure, Improving network agility with seamless bgp reconfigurations, IEEE/ACM Trans. Netw. (2012) 990–1002.

[29] D. Wang, Y. Li, X. Jin, Effect of complex network structure on synchronizability, in: 10th World Congress on Intelligent Control and Automation, WCICA, IEEE, 2012, pp. 3481–3484.
[30] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440–442.
[31] Y. Yu, T.Y. Berger-Wolf, J. Saia, et al., Finding spread blockers in dynamic networks, in: Advances in Social Network Mining and Analysis, Springer, 2010, pp. 55–76.
[32] J. Zhao, K. Xu, Enhancing the robustness of scale-free networks, J. Phys. A 42 (2009) 195003.