J Comb Optim
DOI 10.1007/s10878-014-9822-1

Optimal space coverage with white convex polygons

**Shayan Ehsani · MohammadAmin Fazli ·
Mohammad Ghodsi · MohammadAli Safari**

Abstract Assume that we are given a set of points some of which are black and the rest are white. The goal is to find a set of convex polygons with maximum total area that cover all white points and exclude all black points. We study the problem on three different settings (based on overlapping between different convex polygons): (1) In case convex polygons are permitted to have common area, we present a polynomial algorithm. (2) In case convex polygons are not allowed to have common area but are allowed to have common vertices, we prove the NP-hardness of the problem and propose an algorithm whose output is at least $\left(\frac{OPT}{log(2n/OPT)+2log(n)}\right)^{1/4}$. (3) Finally, in case convex polygons are not allowed to have common area or common vertices, also we prove the NP-hardness of the problem and propose an algorithm whose output is at least $\frac{3\sqrt{3}}{4.\pi}\left(\frac{OPT}{log(2n/OPT)+2log(n)}\right)^{1/4}$.

Keywords White convex polygon · Convex covering · NP-hardness · Algorithm

S. Ehsani
Department of Management Science and Engineering, Stanford University, Stanford, CA, USA
e-mail: shayane@stanford.edu

M. Fazli (✉)· M. Ghodsi · M. Safari
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
e-mail: fazli@ce.sharif.edu

M. Safari
e-mail: msafari@sharif.edu

M. Ghodsi
Institute of Research in Fundamental Sciences (IPM), Tehran, Iran
e-mail: ghodsi@sharif.edu

Published online: 09 December 2014

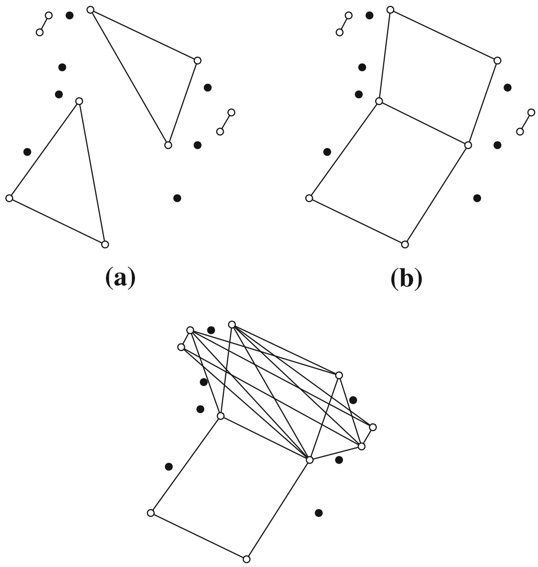

Fig. 1 Three types of different problems

1 Introduction

Assume that we are given a set of n points, some are black and the others are white. A convex polygon is defined as a white convex polygon (WCP) if and only if its vertices are white points and it is free from black points (it only cover white points). The problem of finding the maximum area WCP has been studied by Fischer (1993) where he proposed a dynamic $O(n^4 log(n))$ time algorithm for this problem. In this paper, we take one step forward and consider the problem of covering all white points with a set of WCPs whose union has the maximum area.

Based on whether to allow WCPs to share a white point as a vertex or to allow them to overlap, three different problems can be defined. Differences between these problem is depicted in Fig. 1. Please consider that a single point or two points connected via a segment as a convex hull of area 0. Under such a assumption, it is always possible to cover all white points with convex hulls.

Definition 1 Three different problems of finding WCPs are defined as follows:

- *Totally disjoint convex covering (TDCC)* In this problem WCPs are not allowed to have common vertices and common area (Fig. 1a)
- *Non-overlapping convex covering (NOCC)* In this problem WCPs are not allowed to have common area but are allowed to have common vertices (Fig. 1b).
- *Convex covering with no restriction (CCR)* In this problems WCPs are allowed to have common vertices and common area (Fig. 1c)

As an application, consider a classical problem in communication and wireless networks that is called “Finding White Space Regions” (See Ehsani et al. 2011). We became familiar with this problem in the context of “Cognitive Radio”. This concept was conceived based on the FCC Task Force report in 2004 on the under-utilization of the wireless radio environment. According to the report primary users did not use their licensed spectrum from 15 to 85 % time. Therefore, these silence ranges in time and frequency can be exploited by unlicensed users. The only point these users should keep in mind is not to cause any kind of interference to primary users. Now consider a primary network with fixed base stations that communicate occasionally in a licensed band along with spatially random distributed spectrum sensing base stations of the cognitive radio networks distributed in a large region. The goal is to find the largest area in which no primary user transmission is detected. We call this area the white space region or WSR for short. Then the cognitive radio users residing in the WSR exploit the opportunity to transmit to their neighboring cognitive radio users. It is vital for the cognitive radio that the total area which is covered by WSRs be as much as possible because it would allow better transmission and better communication.

Among the three problems defined in Definition 1, CCR has the most straight forward solution. In the following theorem, we prove that the CCR problem can be solved in polynomial time via a trivial polynomial algorithm.

Theorem 1 *CCR can be solved in polynomial time.*

Proof It is enough to consider all the triangles with white vertices and no black point residing in them and output their union. This union is the maximum area convex covering. Since the number of possible triangles is polynomial $\left(\leq\binom{n}{3}\right)$ the total time will be polynomial. □

For the other two problems, a trivial approach is to greedily select the maximum area WCP in each iteration (by Fischer 1993) and remove it. In this paper, we experimentally show that the performance of this greedy algorithm is high i.e. the ratio of optimal solution to the output of this algorithm is low (less than 2). We also construct instances for which this ratio can be more than any large number that shows the low performance of this algorithm in theory.

We prove the NP-hardness of NOCC and TDCC problems and finally, we propose algorithms to find WCPs with total area of at least $\left(\frac{OPT}{2log(2n/OPT)+2log(n)}\right)^{1/4}$ and $\frac{3\sqrt{3}}{4.\pi}\left(\frac{OPT}{log(2n/OPT)+2log(n)}\right)^{1/4}$ in NOCC and TDCC respectively (OPT is the area of the optimal answer).

2 Non-overlapping convex covering (NOCC)

2.1 Studying the greedy method

In this subsection, we propose a greedy algorithm that leverages the result of Fischer (1993) (finding the maximum area WCP). The algorithm chooses the WCP with the maximum area in each iteration. The area of this WCP must not overlap with convex polygons which will be chosen in successor iterations.

Assume that the convex polygon chosen in current iteration is C. We should eliminate the white points inside C and add some dummy black points to prevent next convex polygons to overlap C. Suppose that two white points A and B are outside or on the boundary of C. Consider the segment connecting A and B. If this segment is going through the interior of C then it can not be an edge of next iterations' convex polygons. We add a dummy black point on the $\overline{AB}$ segment inside C. At the end of this iteration to prevent choosing C again, we add another dummy black point inside it. This scenario adds only $O(n^2)$ points, so the number of the points remains polynomial. Algorithm 1 describe the process in detail.

This greedy algorithm is not optimal since we prove the NP-hardness of the problem (See Sect. 2.2). This algorithm performs very bad in worst case by a factor of $\Omega(n)$ i.e. there are instances in which the output of GreedyNOCC is at most $O\left(\frac{OPT}{n}\right)$ but by experimental evaluations, we show the good performance of this algorithm in practice.

Algorithm 1 GreedyNOCC

Input: P: A set of white and black points.
Output: A set of non-overlapping WCPs.

C = maximum convex polygon of white points of P by the Fischer algorithm in Fischer (1993)
while The area of C is greater than 0 **do**
  Remove all the white points inside C
  for each pair (A, B) of white points which coincide on the outside or boundary of C **do**
    if segment $\overline{AB}$ has intersection with C **then**
      Add a dummy black point on $\overline{AB}$ which resides inside C.
    end if
  end for
  Add a dummy black point inside C.
  C = maximum convex polygon of white points of P.
end while
return all selected convex polygons

Theorem 2 *There is an instance L of n points such that*

$$\frac{OPT(L)}{GreedyNOCC(L)} \in \Omega(n),$$

where $GreedyNOCC(L)$ is the total area of output WCPs from our greedy algorithm on the instance L and $OPT(L)$ is the area of optimal answer.

Proof Consider a n-regular polygon C whose area is S and whose vertices are white. Suppose that the vertices of C are $a_1, a_2, \ldots a_n$. For each edge $\overline{a_i a_{i+1}}$ of C consider a segment $\overline{d_i e_i}$ which is

- parallel with $\overline{a_i a_{i+1}}$ and very close to it.
- inside C.
- shorter than $\overline{a_i a_{i+1}}$.

Establish a triangle $b_i d_i e_i$ with area of $S - \epsilon$ (See Fig. 2) . b_i must be located outside C for which none of the segments $\overline{b_i a_j}$ for $1 \le j \le n$ intersects $\overline{d_i e_i}$. Call the triangle formed by vertices a_i, b_i and b_{i-1}, Δ_i. We must insert some black points in the

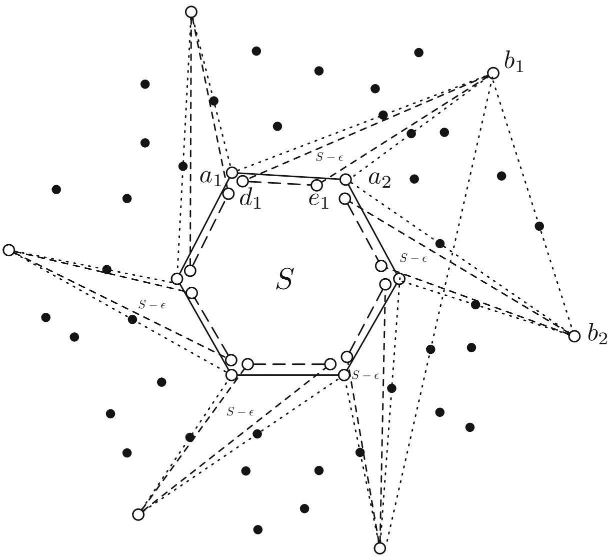

Fig. 2 Worst case example of GreedyNOCC

plane to prevent formation of the convex polygons which include vertices of different triangles. For each triple (A, B, Δ) such that $A, B \in \{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ and $\Delta \in \{\Delta_1, \ldots, \Delta_n\}$, if the segment $\overline{AB}$ intersects Δ we put a black point on $\overline{AB}$ and inside Δ. We put a black point on each of the segments $\overline{a_i b_i}$ and $\overline{a_{i+1} b_i}$ for each i.

The GreedyNOCC algorithm picks C as the first convex polygon and then it can not find any other convex polygon with area more than 0. Therefore the area of the output of GreedyNOCC is S. However the optimal answer picks all the n triangles, so $OPT = n.(S - \epsilon)$. Therefore

$$\frac{OPT(L)}{GreedyNOCC(L)} = \frac{n.(S - \epsilon)}{S} \in \Omega(n).$$

□

We empirically experiment GreedyNOCC algorithm on seven different random distributions of the points $RAN_1, RAN_2, \ldots, RAN_7$. The area of the optimal covering is obtained via a simple backtrack algorithm. The results are shown in Table 1. The first and second entries of row i contain the number of white and black points of RAN_i and the third entry shows the quotient of the output of GreedyNOCC divided by the optimal solution of the NOCC problem for the input RAN_i. All these ratios are more than 0.5 which shows the good performance of this greedy algorithm dealing with random inputs.

2.2 NP-hardness

In this subsection, we prove the NP-hardness of NOCC.

Table 1 The output of GreedyNOCC for 7 random test data

Test data	#White points	#Black points	GreedyNOCC/optimal
RAN_1	10	3	0.54
RAN_2	12	12	0.81
RAN_3	16	10	0.78
RAN_4	14	20	0.65
RAN_5	15	16	0.51
RAN_6	20	40	0.76
RAN_7	20	20	0.81

Definition 2 Suppose that we are given a family of sets $F = \{S_1, S_2, \ldots S_n\}$. An **Intersection Graph** of these sets is a graph which has a vertex v_i for each S_i. There is an edge between v_i and v_j if and only if $S_i \bigcap S_j \neq \emptyset$. If S_is are line segments, the intersection graph of these sets is called a **Segment Intersection Graph**.

The maximum independent set problem in the triangle-free planar graphs is NP-hard (Madhavan 1984). We are reducing this NP-hard problem to NOCC. De Castro et al. (2002) proved that every triangle-free planar graph has an equivalent intersection graph. Their proof is constructive and their proposed construction process can be run in polynomial time.

Lemma 1 (See De Castro et al. 2002) *Every triangle-free planar graph is the intersection graph of a set of segments in three directions.*

Finally in Theorem 3, we prove the NP-hardness of NOCC.

Theorem 3 *NOCC is NP-hard.*

Proof Suppose that we are given a triangle-free planar graph G whose maximum independent set is I. By Lemma 1, we have an equivalent segment intersection graph, H. Suppose that the segments contributed in building H are $\{s_1, \ldots, s_n\}$.

We can w.l.o.g. assume no two segments have intersection at their endpoints, otherwise we can make these segments a little longer.

Our approach forms a list of triangles whose intersection graph is as same as G. We claim that there is an $\epsilon > 0$ for which we can change each s_i to a triangle t_i with area ϵ such that for each $j \neq i$, s_i intersects s_j if and only if t_i intersects t_j. This approach is depicted in Fig. 3.

To prove this claim, it is enough to consider all pair of points (A, B) where A and B belong to different non-intersecting segments from $\{s_1, \ldots, s_n\}$. Call the set consisting all these pairs, P. Suppose that $d_{min} = min_{(A,B)\in P}\{d_{A,B}\}$ where $d_{A,B}$ is equal to the distance between A and B. Also suppose that $l_{min} = min\{|s_i|\}$ where $|s_i|$ is equal to the length of segment s_i. Set $\epsilon_0 = \frac{l_{min}.d_{min}}{4}$. Our claim is true for ϵ_0, since it is enough to put a point with distance $\frac{2\epsilon_0}{|s_i|}$ from s_i and form a triangle with this newly added point and endpoints of s_i.

The intersection graph of these triangles is G. Assume for reductio ad absurdum reasoning, we have a polynomial time algorithm ALG_{NOCC} for solving the NOCC

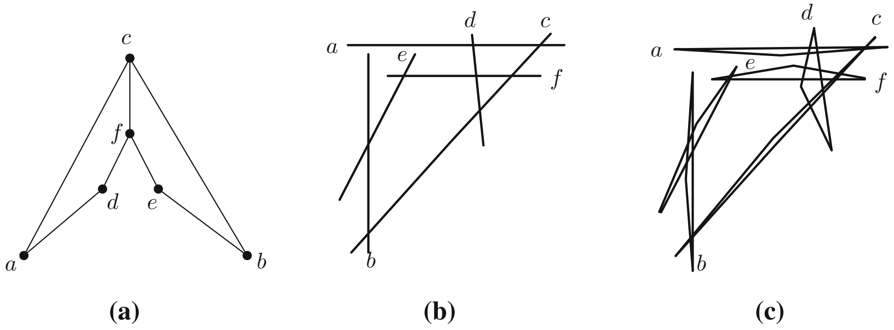

Fig. 3 Forming an intersection graph of equal area triangles from an initial graph: **a** the initial graph G **b** the segment intersection graph H constructed from G **c** the triangle intersection graph constructed from H

problem. Like the idea of Theorem 2, we can insert polynomial number of black points to prevent ALG_{NOCC} from choosing other convex polygons than our constructed triangles (for each triple (A, B, Δ) in which A and B are two vertices of different triangles and Δ is an area which is not included in any of the triangles, put a black point on the segment $\overline{AB}$ and inside Δ). Now run ALG_{NOCC}. Since we have assumed that no two segments intersect at their endpoints, no two triangles would have common vertices. Since we have inserted enough black points no convex polygon other than t_is would be selected. Therefore since the area of all triangles are equal to ϵ_0 and ALG_{NOCC} will choose maximum number of non-overlapping triangles, the output of ALG_{NOCC} will be $|I|.\epsilon_0$. So we can compute the size of the maximum independent set of G in polynomial time which is a contradiction. □

2.3 Approximation algorithm

As we proved, the NOCC problem is NP-hard and no polynomial time algorithm exists for it unless $P = NP$. We propose an algorithm with an approximation for the optimal answer as output.

Similarly, we use the idea of intersection graphs. First notice that w.l.o.g. we can assume the answer of the NOCC problem is a set of triangles, because every polygon can be triangulated. The triangles which contribute in the triangulation of a polygon have common vertices and common edges. In our intersection graph, we connect two triangles via an edge if and only if they have common area (in this setting two triangles with common edge or common vertices but no common area are considered independent) There are at most $\binom{n}{3}$ of these triangles not containing any black point. As described above, construct the intersection graph of these triangles and call it G . For each vertex of G define a weight equal to the area of its associated triangle. In this model, the optimal answer of the NOCC problem will be the maximum weighted independent set of G.

Agarwal and Mustafa (2006) proposed an approximation algorithm for finding an independent set of size at least $\left(\frac{\alpha}{2log(2n/\alpha)}\right)^{1/3}$ in the intersection graph of a set of convex polygons (α is the size of maximum independent set). In this paper, we use their method to develop an approximation algorithm for the NOCC problem. There is a big difference between our problem and theirs; In our setting polygons (triangles) are weighted, but in theirs polygons are not weighted.

Lemma 2 *w.l.o.g. we can assume all the triangles have an area more than one.*

Proof Since we can use a homothety from a random center point and scale every thing up until the area of all the triangles is more than one. It doesn't violate anything but helps us in developing our algorithm. □

Lemma 3 *Given a set I of weighted objects with weights greaters than 1 (i.e. for each $x \in I$ we have $w(x) \geq 1$) and a partially ordering $\preceq$ over the members of I. Define $S = \sum_{x \in I} w(x)$. Then a chain or an antichain of the I's members with total weight greater than $\sqrt{S}$ exists.*

Proof Suppose that the maximum cardinality antichain has less than $\sqrt{S}$ members. From the Dilworth's theorem (Dilworth 1950) the maximum number of chains which partition I is less than $\sqrt{S}$. This means one of these anti chains has total weight greater than $\frac{S}{\sqrt{S}} = \sqrt{S}$. □

Give a vertical line l, some of the triangles intersect with l. Others reside wholly in the left or in the right of that line. Suppose that the number of these triangles is $left(l)$ and $right(l)$ respectively. First we must find a line L for which $|left(L) - right(L)|$ is minimized. Name the set of triangles which intersect L, Δ_L. Also suppose $OPT(\Delta_L)$ is the total area of the triangles of maximum weighted independent set of Δ_L. For a $s_i \in \Delta_L$ define $r(s_i)$ (resp. $l(s_i)$) to be the smallest (largest) x-coordinate of all the points $p \in s_i$. Also define $c(s_i)$ to be the maximum y-coordinate of the intersection of s_i with the L. The following lemma is extracted with minor changes from Agarwal and Mustafa (2006).

Lemma 4 *There exists a sequence $I =< s_{i_1}, s_{i_2}, ..., s_{i_m} >$ of the members of Δ_L where $\sum_{j=1}^{m} w(s_{i_j}) \geq \left|\sum_{x \in \Delta_L} w(x)\right|^{1/4}$ such that $w(x)$ is the area of the triangle x and I has one of the following structures:*

(C_1) $r(s_{i_j}) < r(s_{i_{j+1}})$ and $c(s_{i_j}) < c(s_{i_{j+1}})$.
(C_2) $r(s_{i_j}) < r(s_{i_{j+1}})$ and $c(s_{i_j}) > c(s_{i_{j+1}})$.
(C_3) $r(s_{i_j}) > r(s_{i_{j+1}})$.

Proof We apply Lemma 3 two times one for the partially order defined by C_1 and one for C_2. The first proves the existence of a sequence of length at least $\left|\sum_{x \in \Delta_L} w(x)\right|^{1/2}$. The second proves the existence of a sequence of length at least $\left|\sum_{x \in \Delta_L} w(x)\right|^{1/4}$ among the members of the first sequence. □

For each C_i, $1 \leq i \leq 3$, we should compute the sequence with maximum total area which satisfies C_i. This can be done by a dynamic programming technique. For example we construct a dynamic algorithm for computing the answer for structure C_3. Suppose that members of Δ_L are $s_1, \ldots, s_p$. First sort these members with respect to $c(s_i)$. Then we have $c(s_1) \leq c(s_2) \leq \ldots \leq c(s_p)$. Consider i, j such that $s_i \cap s_j = \emptyset$

Algorithm 2 ApproxNOCC

Input: Δ: a set of triangles
Output: An approximation for NOCC

1: Find the line L which divides Δ into two almost equal cardinality sets.
2: Δ_L = all the triangles intersect L.
3: Δ_{left} = all the triangles wholly reside in the left of L.
4: Δ_{right} = all the triangles wholly reside in the right of L.
5: Compute $\zeta(OPT(\Delta_L))$ via the dynamic programming algorithm.
6: **return** $max\{ApproxNOCC(\Delta_{left}) + ApproxNOCC(\Delta_{right}), \zeta(OPT(\Delta_L)\}$.

and $i \leq j$, define $\phi(i, j)$ to be the size of the maximum total area sequence from the set $\{s_i, s_{i+1}, \ldots, s_j\}$ which satisfies C_3. Then $\phi(i, j)$ is

$$max_{\substack{i \leq k \leq j \\ s_k \cap s_i = \emptyset \\ s_k \cap s_j = \emptyset}} \phi(i, k-1) + \phi(k+1, j) + w(s_k).$$

We do the same computations for the structures C_1 and C_2. Call the answers $\varphi(i, j)$ and $\psi(i, j)$. The computation of the return value ($\zeta(OPT(\Delta_L))$) will be completed by setting

$$\zeta(OPT(\Delta_L)) = max\,\{\phi(1, p), \varphi(1, p), \psi(1, p)\} \geq \left| \sum_{x \in \Delta_L} w(x) \right|^{1/4}$$
$$\geq OPT(\Delta_L)^{1/4}.$$

Now, we can continue with the recursive algorithm proposed by Agarwal and Mustafa (2006). This algorithm computes $\zeta(OPT(\Delta_L))$, then recursively finds the maximum weighted independent set of triangles which wholly reside on the left or right side of L returning the maximum of these two answers. Algorithm 2 explains the details.

Suppose that the output of the ApproxNOCC algorithm is $\mu(t, OPT)$ (t is the number of triangles and OPT is the optimal answer of the NOCC problem). From the structure of our algorithm, we know

$$\mu(t, OPT) \geq max\{\mu(t_L, OPT_L) + \mu(t_R, OPT_R), \zeta(OPT(\Delta_L))\},$$

where $t_L = |\Delta_{left}|$, $t_R = |\Delta_{right}|$ and OPT_L and OPT_R are the optimal answer for the triangles in Δ_{left} and Δ_{right}. This recursive relation has been solved by Agarwal and Mustafa (2006) and a lower bound has been computed for $\mu(t, OPT)$ that is

$$\mu(t, OPT) \geq \zeta\left(\frac{OPT}{2log(2t/OPT)}\right).$$

So we have

$$\mu(t, OPT) \geq \left|\frac{OPT}{2log(2t/OPT)}\right|^{1/4} \geq \left|\frac{OPT}{2log(2n/OPT) + 2log(n)}\right|^{1/4}.$$

The last inequality is resulted from $t \leq \binom{n}{3}$.

Theorem 4 *For a given set of n black and white points, if the optimal answer of the NOCC problem is OPT, there is an approximation algorithm whose output is at least* $\left|\frac{OPT}{2log(2n/OPT)+2log(n)}\right|^{1/4}$.

3 Totally disjoint convex covering (TDCC)

3.1 Greedy method

Similar to the GreedyNOCC Algorithm 2.1, we can propose a greedy method for solving the TDCC problem. The details of the algorithm GreedyTDCC is shown in Algorithm 3. The minor difference between these two algorithms is Line 4 in which we change the color of the selected polygon in order to prevent future convex polygons to have any intersection with its boundaries.

Algorithm 3 GreedyTDCC

Input: P: A set of white and black points.
Output: A set of non-overlapping WCPs.

```
1:  C = maximum convex polygon of white points of P by the Fischer algorithm in Fischer (1993)
2:  while The area of C is greater than 0 do
3:     Remove all the white points inside C
4:     Change the color of C's vertices from white to black
5:     for each pair (A, B) of points which coincide on the outside or boundary of C do
6:        if segment AB has intersection with C then
7:           Add a dummy black point on AB which resides inside C.
8:        end if
9:     end for
10:    Add a dummy black point inside C.
11:    C = maximum convex polygon of white points of P.
12: end while
13: return all selected convex polygons
```

Like GreedyNOCC, GreedyTDCC works well in practice but there are instances for which the ratio of this algorithm's output to the optimal answer is lower than any small number. Theorem 5 explains this fact.

Theorem 5 *There exists an instance L of n points such that*

$$\frac{OPT(L)}{GreedyTDCC(L)} \in \Omega(n),$$

where $GreedyTDCC(L)$ is the total area of output WCPs from our greedy algorithm on the instance L and $OPT(L)$ is the area of optimal answer.

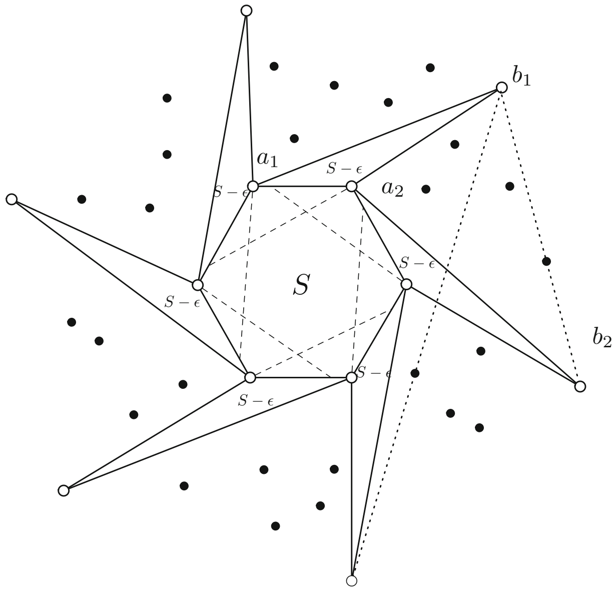

Fig. 4 Worst case example of GreedyTDCC

Proof The proof is very similar to the proof of Theorem 2. Figure 4 illustrates an instance of TDCC. Since b_i can not see any of a_js other than a_i and a_{i+1}, the only probable convex polygons are triangles with area of $S - \epsilon$ and n-regular polygon with area of S. The GreedyTDCC algorithm chooses C as the first convex polygon and stops since no more convex polygon with area more than 0 exists. Therefore the area of the output of GreedyTDCC is S, however the optimal answer is $\frac{n}{2}$ disjoint triangles chosen from n perimeter triangles i.e. $OPT = \frac{n}{2}(S - \epsilon)$. Hence:

$$\frac{OPT(L)}{GreedyTDCC(L)} = \frac{\frac{n}{2} \cdot (S - \epsilon)}{S} \in \Omega(n).$$

□

Table 2 illustrates the results of an empirical experiment of GreedyTDCC on 7 different random test cases $RAN_1 \ldots RAN_7$. The first and second entries of the ith row are the number of white and black points of RAN_i. The third column contains the ratio of GreedyTDCC's output's area to the area of the optimal answer for each test case.

3.2 NP hardness

Like the NOCC problem, the TDCC problem is NP-hard. The proof is similar to the proof of Theorem 3.

Theorem 6 *The TDCC problem is NP-hard.*

Table 2 The output of GreedyTDCC for 7 random test data

Test data	#White points	#Black points	GreedyTDCC/optimal
RAN_1	10	3	0.60
RAN_2	12	12	0.96
RAN_3	16	10	0.97
RAN_4	14	20	0.70
RAN_5	15	16	0.52
RAN_6	20	40	0.73
RAN_7	20	20	0.52

3.3 Approximation algorithm

Similar to Algorithm 2, we take all $t \leq \binom{n}{3}$ triangles containing no black point and make the intersection graph out of these triangles. However, unlike the ApproxNOCC problem, for this problem we assume two triangles with a common vertex are connected. Again we find a set of totally-disjoint triangles with maximum total sum and show that this action does not seriously affect the output of ApproxNOCC. Theorem 7 states this idea formally:

Theorem 7 *For a given set of n black and white points, if the optimal answer of the TDCC problem is OPT, there is an approximation algorithm whose output's area is at least* $\frac{3\sqrt{3}}{4\pi}\left|\frac{OPT}{2log(2n/OPT)+2log(n)}\right|^{1/4}$.

Proof Let B be a compact convex body in the plane and B_k be a largest area k-gon inscribed in B. From Sas (1941) it is known that $area(B_k) \geq area(B) \cdot \frac{k}{2\pi} sin\frac{2\pi}{k}$, where equality holds if and only if B is an ellipse. Assuming $k = 3$, we can deduce that every convex polygon B has an inscribed triangle with the area greater than or equal to $\frac{3\sqrt{3}}{4\pi} area(B)$.

Consider the optimal convex polygons for the TDCC problem. If we replace each of the convex polygons in this solution with its maximum inscribed triangle, we have another convex covering with area more than $\frac{3\sqrt{3}}{4\pi}.OPT$ only consists of triangles as depicted in Figure 5. With this perspective, like the ApproxNOCC algorithm, we can find the maximum weighted independent set of the intersection graph formed by the black point free triangles and be determined that its answer will have area more than $\frac{3\sqrt{3}}{4\pi}\left|\frac{OPT}{2log(2n/OPT)+2log(n)}\right|^{1/4}$. □

4 Conclusion and further works

A good motivation to continue our work is to find better approximation algorithms or inapproximability results for the independent set of intersection graphs of convex shapes with maximum area, specially for the triangles. These results will help us to propose better approximation algorithms for the problems of this paper. Moreover,

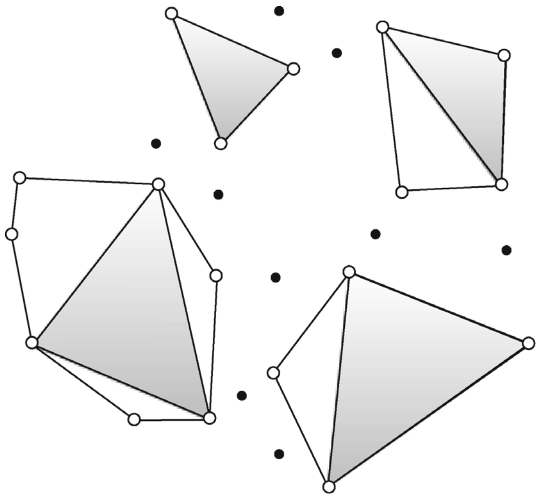

Fig. 5 Optmial convex covering for the TDCC problem and replacing each convex polygon with its maximum inscribed triangle

another types of problems can be proposed in this context. For example we can consider the weighted version of this problem in which black points have weights. We want to find a convex covering which maximizes the sum of the areas and the total weight of the black points covered by one convex is less than a given number W.

References

Agarwal PK, Mustafa NH (2006) Independent set of intersection graphs of convex objects in 2d. Comp Geom 34(2):83–95

De Castro N, Cobos FJ, Dana JC, Marquez A, Noy M (2002) Triangle-free planar graphs as segment intersection graphs. J Graph Algorithms Appl 6(1):7–26

Dilworth RP (1950) A decomposition theorem for partially ordered sets. Ann Math 51(1):161–166

Ehsani S, Fazli MA, Ghodsi M, Safari MA, Saghafian M, Tavakkoli M (2011) White space regions. In: Cerna I, Gyimothy T (eds) SOFSEM 2011: Theory and practice of computer science. Springer, Berlin, pp 226–237

Fischer P (1993) Finding maximum convex polygons. In: Esik Z (ed) Fundamentals of computation theory. Springer, Berlin, pp 234–243

Madhavan C (1984) Approximation algorithm for maximum independent set in planar traingle-free graphs. In: Joseph M, Shyamasundar R (eds) Foundations of software technology and theoretical computer science. Springer, Berlin, pp 381–392

Sas E (1941) On a certain extremum-property of the ellipse. Mat Fiz Lapok 48:533–542