A Linear Edge Kernel for Two-Layer Crossing Minimization

Yasuaki Kobayashi[1,a) Hirokazu Maruta[1,b) Yusuke Nakae[2,c) Hisao Tamaki[1,d)

Abstract: We consider a simple generalization of two-layer crossing minimization problem (TLCM) called leaf-edge-weighted TLCM (LEW-TLCM), where we allow positive weights on edges incident to leaves, and show that this problem admits a kernel with $O(k)$ edges provided that the given graph is connected. As a straightforward consequence, LEW-TLCM (and hence TLCM) has a fixed parameter algorithm that runs in $2^{O(k \log k)} + n^{O(1)}$ time which improves on the previously best known algorithm with running time $2^{O(k^3)}n$.

1. Introduction

A *two-layer drawing* of a bipartite graph G with bipartition (X, Y) of vertices places vertices in X on one line and those in Y on another line parallel to the first and draws edges as straight line segments between these two lines. We call these parallel lines *layers* of the drawing. A *crossing* in a two-layer drawing is a pair of edges that intersect each other at a point not representing a vertex. Note that the set of crossings in a two-layer drawing of G is completely determined by the order of the vertices in X on one layer and the order of the vertices in Y on the other layer. The problem to find a two-layer drawing whose crossing number is the minimum is called *two-layer crossing minimization*, TLCM for short. This problem is known to be NP-hard [12] (although it is polynomial time solvable for trees [15] and permutation graphs [16]). We consider this problem and its generalization on a parameterized perspective described as follows. An edge is a *leaf edge* if it is incident to a leaf (a vertex of degree one); a *internal edge* otherwise.

Two-Layer Crossing Minimization (TLCM)
Instance: bipartite graph $G = (V(G), E(G))$
Parameter: k
Task: Find a two-layer drawing of G with at most k crossings?

Leaf-Edge-Weighted TLCM (LEW-TLCM)
Instance: bipartite graph $G = (V(G), E(G))$, function $w : E(G) \to \mathbb{N}$ with $w(e) = 1$ for every internal edge $e \in E(G)$
Parameter: k
Task: Find a two-layer drawing of G with crossings of total weight at most k, where a crossing (e, e') has weight $w(e)w(e')$?

1 Meiji University, Kawasaki, Kanagawa 214–8571
2 Business Information Technical Systems, Taito-ku, Tokyo, 111–0034
a) yasu0207@cs.meiji.ac.jp
b) maruta@cs.meiji.ac.jp
c) dim_double@yahoo.co.jp
d) tamaki@cs.meiji.ac.jp

Clearly, LEW-TLCM is a generalization of TLCM. In this paper, we give kernelizations for these problems.

A *kernelization* [4] for a parameterized problem is an algorithm that, given an instance I and a parameter k, computes an instance I' and a parameter k' of the same problem in time polynomial in k and the size of I such that
(1) (I, k) is feasible if and only if (I', k') is feasible,
(2) the size of I' is bounded by a computable function f in k, and
(3) k' is bounded by a function in k.
When the function f is polynomial, we call the algorithm a *polynomial kernelization* and its output a *polynomial kernel*.

TLCM is a special case of a problem called h-layer crossing minimization which decides if a given graph has an h-layer drawing with at most k crossings. This problem is fixed parameter tractable [6] when parameterized by $k + h$. The running time of the algorithm of [6] is $2^{O((h+k)^3)}n$. Besides having a large exponent in the running time, this algorithm is rather complicated, involving, in particular, the fixed parameter algorithm for pathwidth due to Bodlaender and Kloks [1], [2] and is not easy to implement. It is natural to ask if we can obtain a simpler and faster fixed parameter algorithm for the special case of $h = 2$, namely TLCM. To the best of the present authors' knowledge, neither a faster algorithm for TLCM than the one given in [6] nor its polynomial kernelization is previously known.

In contrast to TLCM, several fixed parameter algorithms are known for *two-layer planarization* (TLP), where the objective is to find a subset of edges of size at most k of the given graph whose removal enables a two-layer drawing without any crossings. This problem is fixed parameter tractable [5] and can be solved in time $O(k \cdot 3.562^k + n)$ where n is the number of vertices of the given graph [17]. Moreover, a kernel with $O(k)$ edges for TLP is known [5].

Another related problem is *one-sided crossing minimization*

(OSCM), which asks for a two-layer drawing of the given bipartite graph with minimum number of crossings, but with the vertex order in one layer fixed as part of the input. OSCM is also NP-hard [10]. The fixed parameter tractability of OSCM seems to be better-studied [7], [8], [11], [14] than TLCM. More specifically, [8] gives the first fixed parameter algorithm, [7] gives a faster fixed parameter algorithm and a kernel with $O(k^2)$ edges, and [11] and [14] give subexponential fixed parameter algorithms with running time $2^{O(\sqrt{k}\log k)} + n^{O(1)}$ and $O(3^{\sqrt{2k}} + n)$, respectively.

Our results are as follows.

Theorem 1. *TLCM admits a kernel with $O(k^2)$ edges provided that the given graph is connected.*

Theorem 2. *LEW-TLCM admits a kernel with $O(k)$ edges provided that the given graph is connected.*

The second theorem implies a fixed parameter algorithm with running time $2^{O(k\log k)} + n^{O(1)}$ for both TLCM and LEW-TLCM via the standard approach: given an instance of TLCM or LEW-TLCM, we construct a kernel with $O(k)$ edges in LEW-TLCM for each connected component and then do an exhaustive search to find a solution for each of these kernels. Note that all but k connected components are crossing-free, assuming that the given instance is feasible, which can be detected by a simple linear time algorithm [9].

We remark that, although some of the lemmas needed for the kernelization are non-trivial, the kernelization algorithm itself is quite simple and easy to implement.

The rest of this paper is organized as follows. In section 2, we give preliminaries for TLCM. In section 3, we describe a kernelization for TLCM whose output has $O(k^2)$ edges, proving some lemmas necessary for this kernelization. In section 4, we show that the same method works for LEW-TLCM and gives a kernelization whose output has $O(k)$ edges. Finally, Section 5 contains the conclusion.

2. Preliminaries

Let G be a bipartite graph with a prescribed bipartition of the vertex set. We denote by $V(G)$ the set of vertices of G, by $(X(G), Y(G))$ the bipartition of $V(G)$, and by $E(G) \subseteq X(G) \times Y(G)$ the set of edges of G. We also view G as a triple $(X(G), Y(G), E(G))$. For each vertex $v \in V(G)$, we denote by $d(v)$ the degree of v. A *leaf* is a vertex v with $d(v) = 1$. We call an edge a *leaf edge* if the edge is incident to a leaf; otherwise a *internal edge*. For an edge e in G, the graph $G - e$ is a subgraph obtained from G by deleting e. A *cut vertex* (a *bridge*) is a vertex (an edge) whose removal increases the number of components. A *block* is a maximal connected subgraph without a cut vertex. We say that a block is *trivial* if it has at most two vertices. Otherwise, the block is *non-trivial*. For each subset U of $V(G)$, we denote by $G[U]$ the subgraph of G induced by U. For each subgraph G' of G, we denote $X(G) \cap V(G')$ by $X(G')$ and $Y(G) \cap V(G')$ by $Y(G')$.

A two-layer drawing $\mathcal{D}$ of G is defined as a triple $(G, <_X, <_Y)$ where $<_X$ and $<_Y$ are total orders on $X(G)$ and $Y(G)$, respectively. The number of crossings in $\mathcal{D}$ is the number of pairs of edges that intersect each other:

$$bcr(\mathcal{D}) = \sum_{\{u,v\}\in E(G)} |\{(x, y) \in E(G) : x <_X u, v <_Y y\}|.$$

The *bipartite crossing number* of G, denoted by $bcr(G)$, is the minimum number of crossings over all two-layer drawings of G. For each subgraph G' of G, we denote by $\mathcal{D} \mid G'$ the two-layer drawing of G' in which the vertices of $V(G')$ are placed in the same order as in $\mathcal{D}$. For distinct vertices $u, v \in X(G)$, we say that a vertex u is *to the left* (*right*) of v in $\mathcal{D}$ if $u <_X v$ ($v <_X u$). A vertex $u \in X(G)$ is the *leftmost* (*rightmost*) in $\mathcal{D}$ if $u <_X v$ ($v <_X u$) for all $v \in X(G) \setminus \{u\}$. We may omit the reference to $\mathcal{D}$ when it is clear from the context. We use similar terminology for vertices in $Y(G)$.

3. A kernel with $O(k^2)$ edges for TLCM

In this section, we give a kernelization for TLCM whose output has $O(k^2)$ edges. The same approach is extended to a kernelization for LEW-TLCM in the next section. This kernelization is based on some lower bound on the bipartite crossing number and some reduction rule on bridges. First, we show a technical lemma.

Lemma 1. *Let $\mathcal{D}$ be a two-layer drawing of a bipartite graph G and let P be a path in $\mathcal{D}$ from a leftmost vertex u, either in $X(G)$ or $Y(G)$, to a rightmost vertex v, either in $X(G)$ or $Y(G)$. Then, each edge not incident to $V(P)$ has a crossing with some edge in P.*

Proof. Consider an arbitrary geometric representation of $\mathcal{D}$ and connect u and v with a curve Q not intersecting any edge in the drawing. Then, the closed curve consisting of Q and the polygonal curve representing P divides the plane into several regions. One of them contains $X(G) \setminus V(P)$ in its interior and another region, distinct from the first, contains $Y(G) \setminus V(P)$ in its interior. Therefore, each edge not incident to $V(P)$ must intersect with the closed curve and hence cross some edge of P, since it does not intersect with Q. See. Fig. 1 for an example. □

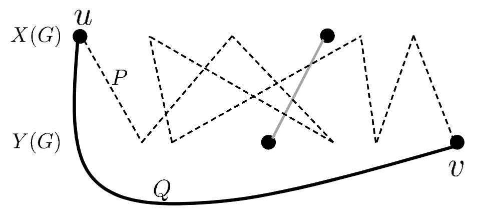

Fig. 1 The dotted polygonal curve indicates a path P and the solid black line indicates a curve Q. Each edge not incident to $V(P)$ has a crossing with P.

Lemma 2. *Let G be a biconnected bipartite graph. Then $bcr(G) \geq \frac{|E(G)|-1}{3}$.*

The proof of this lemma is omitted in this version and can be found in [13]. The bound in this lemma is tight. See Fig. 2 for an example, which can be generalized to a biconnected graph with $3k + 1$ edges and k crossings for arbitrary $k \geq 1$.

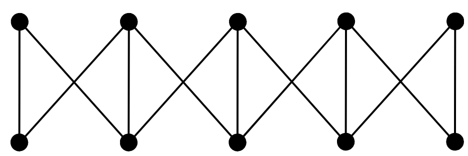

Fig. 2 A biconnected graph G with $bcr(G) = \frac{|E(G)|-1}{3}$.

Fix a bipartite graph G and a positive integer k. We assume that G is connected since our goal is to establish Theorem 1. In the rest of this section, we show that G can be reduced to a graph G' such that $bcr(G) = bcr(G')$ and $|E(G')| \le f(k)$, where $f(k) = O(k^2)$, if $bcr(G) \le k$. This implies a kernelization with $O(k^2)$ edges: we output a trivial infeasible instance if $|E(G')| > f(k)$.

We say a bridge e of G is *order-inducing* if each of the two connected components of $G - e$ has more than k edges; otherwise e is *non-order-inducing*. Let us note that every leaf edge is non-order-inducing. The following lemma justifies the name of order-inducing bridges.

Lemma 3. *Let $e = (u, v)$ be an order-inducing bridge of G and let G_1 and G_2 be the connected components of $G - e$ with $u \in X(G_1)$ and $v \in Y(G_2)$. If $bcr(G) \le k$ then there is a two-layer drawing $\mathcal{D} = (G, <_X, <_Y)$ with $bcr(\mathcal{D}) \le k$ in which the vertices of G_1 are placed entirely to the left of the vertices of G_2. That is, $a <_X b$ for $a \in X(G_1), b \in X(G_2)$ and $a <_Y b$ for $a \in Y(G_1), b \in Y(G_2)$.*

Proof. Let $\mathcal{D}'$ be an arbitrary two-layer drawing with $bcr(\mathcal{D}') \le k$ in which G_1 contains the leftmost vertex y_l of $Y(G)$.

First, we claim that the rightmost vertex x_r of $X(G)$ in $\mathcal{D}'$ is contained in $X(G_2)$. Assume otherwise, that G_1 contains both y_l and x_r. Then, by Lemma 1, the path from y_l to x_r in G_1 have a crossing with each edge in $E(G_2)$, contradicting the assumption that $bcr(\mathcal{D}') \le k$.

We construct a two-layer drawing $\mathcal{D} = (G, <_X, <_Y)$ as follows:
(1) $a <_X b$ for $a \in X(G_1), b \in X(G_2)$,
(2) $a <_Y b$ for $a \in Y(G_1), b \in Y(G_2)$,
(3) $\mathcal{D}' \mid G_1 = \mathcal{D} \mid G_1$, and
(4) $\mathcal{D}' \mid G_2 = \mathcal{D} \mid G_2$.

Clearly, $<_X$ and $<_Y$ are total orders on X and Y respectively. In the following we show that $bcr(\mathcal{D}') \ge bcr(\mathcal{D})$. Since each edge of G_1 has no crossings with any edge of G_2 in $\mathcal{D}$ and the crossings within G_1 and within G_2 are preserved, to prove the inequality, it suffices to show that each edge $f \in E(G_1)$ that crosses e in $\mathcal{D}$ has at least one crossing with $E(G_2) \cup \{e\}$ in $\mathcal{D}'$, together with the symmetric property for edges in G_2.

Let $f = (x, y)$ be an edge of G_1 that crosses e in $\mathcal{D}$. Let P be a path consisting of e and a path from v to x_r in G_2. Since $e = (u, v)$ and $f = (x, y)$ cross each other and $y \in Y(G_1)$ is to the left of $v \in Y(G_2)$ in $\mathcal{D}$, x is to the right of u in $\mathcal{D}$. This order is the same in $\mathcal{D}'$ as $u, x \in X(G_1)$. Moreover, $x \ne x_r$ since $x \in X(G_1)$ and $x_r \in X(G_2)$. Therefore, $f = (x, y)$ crosses some edge of P, by an argument similar to the proof of Lemma 1. We are done since $E(P) \subseteq E(G_2) \cup \{e\}$. See Fig. 3 for an example. □

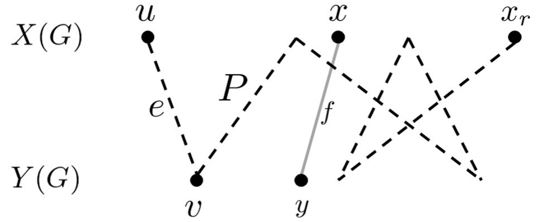

Fig. 3 Path P and edge $f = (x, y)$ in the drawing $\mathcal{D}'$. P, which include e, is shown in dotted lines. Vertex x is to the right of u as it is in $\mathcal{D}$. Edge f crosses P no matter whether y is to the left or to the right of v.

We say that an order-inducing bridge e of G is *contractable* if each end of e is incident to an order-inducing bridge distinct from e and is not incident to any internal edge other than e and this order-inducing bridge.

Lemma 4. *Suppose G has a contractable bridge e and let H be the result of contracting e in G. Then, $bcr(G) \le k$ if and only if $bcr(H) \le k$.*

Proof. Let $e = (v_1, v_2)$ and let e_i, $i = 1, 2$, be the order-inducing edge that is incident to v_i and is distinct from e. For $i = 1, 2$, let G_i be the connected component of $G - e_i$ that does not contain e.

Suppose first that $bcr(G) \le k$ and let $\mathcal{D}$ be a drawing of G with $bcr(\mathcal{D}) = bcr(G)$. Since e, e_1, and e_2 are order-inducing, by Lemma 3, we may assume that, in $\mathcal{D}$, the vertices of $V(G_1)$ lie entirely to the left of v_1 and v_2 while the vertices of $V(G_2)$ lie entirely to the right of v_1 and v_2. Therefore, the edges of G_i have no crossings with edges not in $E(G_i) \cup \{e_i\}$, for $i = 1, 2$. Let $\mathcal{D}'$ be the drawing of H naturally derived from $\mathcal{D}$ as follows. Starting from the drawing $\mathcal{D}$, take its subdrawing $\mathcal{D} \mid G'_1$, where $G'_1 = G[V(G_1) \cup \{v_1\}]$, and flip it upside down, that is, place the vertices in $X(G)$ on the layer for $Y(G)$ and vice versa and keep the order of vertices within G'_1. Then, contract v_1 and v_2, now in the same layer, into one vertex. Finally, place all the leaves adjacent to this contracted vertex between the drawings of G_1 and G_2. Clearly, we have $bcr(\mathcal{D}') = bcr(\mathcal{D})$. See Fig. 4.

To show the reverse direction, suppose $bcr(H) \le k$ and let $\mathcal{D}'$ be a drawing of H with $bcr(\mathcal{D}') = bcr(H)$. Let v be the vertex of H into which e is contracted and e'_i the edge e_i with v_i replaced by v, for $i = 1, 2$. Since e_1 and e_2 are order-inducing in G, e'_1 and e'_2 are order-inducing in H. Therefore, we may assume that, in $\mathcal{D}'$, the vertices of $V(G_1)$ lie entirely to the left of v and $V(G_2)$ while the vertices of $V(G_2)$ lie entirely to the right of v and $V(G_1)$. By a conversion that is an inverse of the above conversion from $\mathcal{D}$ to $\mathcal{D}'$, we obtain a drawing $\mathcal{D}$ of G such that $bcr(\mathcal{D}) = bcr(\mathcal{D}')$. □

Repeating the contraction of contractable bridges until there is no contractable bridges, we obtain a kernel of the given instance. The following lemma bounds the size of the kernel.

Lemma 5. *Suppose $bcr(G) \le k$ and G does not have any contractable bridge. Then, the number of internal edges of G is at most $10k + 3$.*

The proof of this lemma is omitted in this version and can be found in [13].

To obtain a kernel with $O(k^2)$ edges, we need an upper bound on the number of leaf edges of the reduced graph G. Note that we can assume each vertex is incident to at most $k + 1$ leaf edges. This follows from the fact that there is a two-layer drawing $\mathcal{D}$ of G with $bcr(G) = bcr(\mathcal{D})$ such that the leaves of G with a common neighbor appear consecutively in $\mathcal{D}$. This means that, if a vertex has more than k leaf neighbors, then all but $k + 1$ of them can be discarded without changing the feasibility of the instance. Therefore, Lemma 4 implies that the kernel obtained from a feasible instance has at most $10k + 3 + (10k + 4)(k + 1) = 10k^2 + 24k + 7$ edges.

4. A kernel with $O(k)$ edges for LEW-TLCM

It is clear that the kernel for TLCM described in the previous

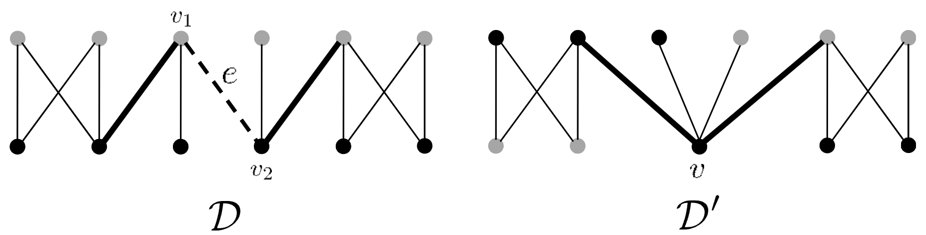

Fig. 4 An example of two-layer drawings $\mathcal{D}$ and $\mathcal{D}'$. The dotted line indicates a contractable bridge and thick lines indicate order-inducing bridges.

section can be represented by an instance of LEW-TLCM with $O(k)$ edges. Although this observation is sufficient for algorithmic purposes, we would like to say that LEW-TLCM has a kernel with $O(k)$ edges for an aesthetic reason. To this end, we confirm that the lemmas in the previous section are applicable to instances of LEW-TLCM.

For each leaf-edge-weighted graph G with weight function w, let unfold(G, w) denote the unweighted graph equivalent to G with weight w: each vertex v that is incident to a leaf edge e in G is incident to $w(e)$ leaf edges in unfold(G, w).

To adapt the lemmas of the previous section to leaf-edge-weighted instances, we read "the number of edges" as "the sum of weights of edges" in the definitions and lemmas. Then the statements of the lemmas for a weighted instance (G, w, k) are equivalent to the statements for the unweighted instance (unfold$(G, w), k)$ and hence do hold. Under this interpretation, the kernelization in the previous section works for an LEW-TLCM instance and produces a kernel with at most $10k + 3 + (10k + 4) = 20k + 7$ edges.

5. Concluding remarks

We have given an $O(k)$ edge kernel for connected instances of a simple generalization of TLCM. Its consequences are not limited to the fixed parameter algorithm mentioned in the introduction, which applies a brute-force search to the kernel. There are other methods for exactly solving TLCM such as integer programming [18] and semidefinite programming [3]. Our kernelization is expected to broaden the class of instances practically solvable by such methods. We are planning an extensive experiments along this line.

References

[1] H. L. Bodlaender: A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6), pp.1305–1317, 1996.
[2] H. L. Bodlaender and T. Kloks: Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2), pp.358–402, 1996.
[3] M. Chimani, P. Hungerländer, M. Jünger and P. Mutzel: An SDP approach to multi-level crossing minimization. *Journal of Experimental Algorithmics*, 3(3), pp.1–16, 2012.
[4] R. G. Downey and M. R. Fellows: *Parameterized complexity*. Springer, 1999.
[5] V. Dujmović, M. R. Fellows, M. Hallet, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. Rosamond, S. Whitesides and D. R. Wood: A fixed parameter approach to two-layer planarization. *Algorithmica*, 45(2), pp.159–182, 2006.
[6] V. Dujmović, M. R. Fellows, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. Rosamond, M. Suderman, S. Whitesides and D. R. Wood: On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2), pp.267–292, 2008.
[7] V. Dujmović, H. Fernau and M. Kaufmann: Fixed parameter algorithms for one-sided crossing minimization revisited. *Journal of Discrete Algorithms*, 6(2), pp.313–323, 2008.
[8] V. Dujmović and S. Whitesides: An efficient fixed parameter tractable algorithm for 1-sided crossing minimization. *Algorithmica*, 40(1), pp.15–31, 2004.
[9] P. Eades, B. D. McKay and N. C. Wormald: On an edge crossing problem. In *Proceedings of 9th Australian Computer Science Conference*, pp.327–334, 1986.
[10] P. Eades and N. C. Wormald: Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4), pp.379–403, 1994.
[11] H. Fernau, F. V. Fomin, D. Lokshtanov, M. Mnich, G. Philip and S. Saurabh: Ranking and drawing in subexponential time. In *Proceedings of the 21st International Workshop on Combinatorial Algorithms*, IWOCA2010, pp.337-348, 2011.
[12] M. R. Garey and D. S. Johnson: Crossing number is NP-complete. *SIAM. J. on Algebraic and Discrete Methods*, 4(3), pp.312–316, 1982.
[13] Y. Kobayashi, H. Maruta, Y. Nakae and H. Tamaki: A linear edge kernel for two-layer crossing minimization. To appear in *Proceedings of the 19th Annual International Computing and Combinatorics Conference*, COCOON2013, 2013.
[14] Y. Kobayashi and H. Tamaki: A fast and simple subexponential fixed parameter algorithm for one-sided crossing minimization. In *Proceedings of the 20th Annual European Symposium on Algorithms*, ESA2012, pp.683–694, 2012.
[15] F. Shahrokhi, O. Sýkora, L. A. Székely and I. Vrt'o: On bipartite Drawings and the linear arrangement problem. *SIAM Journal on Computing*, 30(6), pp.1773–1789, 2001
[16] J. Spinrad, A. Brandstädt and L. Stewart: Bipartite permutation graphs. *Discrete Applied Mathematics*, 18(3), pp.279–292, 1987
[17] M. Suderman: Layered graph drawing. PhD thesis, School of Computer Science, McGill University Montréal, 2005.
[18] L. Zheng and C. Buchheim: A new exact algorithm for the two-sided crossing minimization problem. In Proceedings of the 1st International Conference on Combinatorial Optimization and Applications, COCOA2007, pp.301–310, 2007.

