

Local Search: Is Brute-Force Avoidable?

Michael R. Fellows,
Frances A. Rosamond,
University of Newcastle,
Newcastle, Australia.

{michael.fellows, frances.rosamond}
@newcastle.edu.au

Fedor V. Fomin, Daniel Lokshtanov,
Saket Saurabh, Yngve Villanger
University of Bergen,
5020 Bergen, Norway.

{fomin, daniello, saket.saurabh, yngvev}
@ii.uib.no

Abstract

Many local search algorithms are based on searching in the k -exchange neighborhood. This is the set of solutions that can be obtained from the current solution by exchanging at most k elements. As a rule of thumb, the larger k is, the better are the chances of finding an improved solution. However, for inputs of size n , a naïve brute-force search of the k -exchange neighborhood requires $n^{O(k)}$ time, which is not practical even for very small values of k . We show that for several classes of sparse graphs, like planar graphs, graphs of bounded vertex degree and graphs excluding some fixed graph as a minor, an improved solution in the k -exchange neighborhood for many problems can be found much more efficiently. Our algorithms run in time $O(\tau(k) \cdot n^c)$, where τ is a function depending on k only and c is a constant independent of k . We demonstrate the applicability of this approach on different problems like r -CENTER, VERTEX COVER, ODD CYCLE TRANSVERSAL, MAX-CUT, and MIN-BISECTION. In particular, on planar graphs, all our algorithms searching for a k -local improvement run in time $O(2^{O(k)} \cdot n^2)$, which is polynomial for $k = O(\log n)$. We also complement the algorithms with complexity results indicating that—brute force search is unavoidable—in more general classes of sparse graphs.

1 Introduction

The idea of local search is to improve a solution by searching for a better solution in a neighborhood which is defined in a problem specific way. For example, for the classical TRAVELING SALESPERSON problem, the neighborhood of a tour can be defined as the set of all tours that differ from it in at most k edges, this is called the k -exchange neighborhood [Lin and Kernighan, 1973; Papadimitriou and Steiglitz, 1977]. Another classical example is the MIN-BISECTION problem [Johnson *et al.*, 1988], where for a given graph with $2n$ vertices and weights on the edges, the task is to partition the vertices into two sets of n vertices such that the sum of the weights of the edges between the sets is minimized. A natural k -exchange neighborhood for this problem would be

the set of partitions that can be obtained from each other by swapping at most k pairs of vertices.

Most of the literature on local search is primarily devoted to experimental studies of different heuristics. The theoretical study of local search has been developing mainly in three directions. The first direction is the study of performance guarantees of local search, i.e. the quality of the solution [Papadimitriou and Steiglitz, 1982; Khanna *et al.*, 1998]. The second direction of the theoretical work is on the asymptotic convergence of local search in probabilistic settings, such as simulated annealing [Aarts and Lenstra, 1997]. The third direction, which is the most relevant to our work, concerns the time required to reach a local optimum. An illustrative example here is the simplex method, which can be seen as a local search algorithm with feasible solutions corresponding to “vertices” of a polytope. The neighbors of a solution are the solutions that can be reached from it by a single “pivot”. There can be different rules of choosing a pivot when several neighbors improve a solution. However, for each of the known rules there are examples requiring an exponential number of iterations for reaching the local optimum. Motivated by the fact that many local search algorithms are based on neighborhood structures for which locally optimal solutions are not known to be computable in polynomial time, [Johnson *et al.*, 1988] defined a complexity class PLS which can be seen as an analogue of the class NP for local search problems. Many natural local search problems appear to be PLS-complete. We refer to books [Aarts and Lenstra, 1997; Michiels *et al.*, 2007] for more information on different aspects of local search.

In this paper we take a different twist in the study of local search and endeavor to answer the following natural question. Is there a faster way of searching the k -exchange neighborhood than brute-force? This question is important because the typical running time of a brute-force algorithm is $n^{O(k)}$, where n is the input length, which becomes a real obstacle in using k -exchange neighborhoods in practice even for sufficiently small values of k . For many years most algorithms searching for improved solution in the k -exchange neighborhood had an $n^{O(k)}$ running time, and thus creating an impression that this cannot be done significantly faster than the brute-force search. But is there mathematical evidence for this common belief? Or maybe for some problems it is possible to search k -exchange neighborhoods in time $O(\tau(k)n^c)$,

where c is a small constant, which can make local search much more powerful?

An appropriate tool to answer all these questions is parameterized complexity. In the parameterized framework, for decision problems with input size n , and a parameter k , the goal is to design an algorithm with runtime $\tau(k) \cdot n^{O(1)}$, where τ is a function of k alone. Problems having such an algorithm are said to be fixed parameter tractable (FPT). There is also a theory of hardness that allows to identify parameterized problems that are not amenable to such algorithms. The hardness hierarchy is represented by $W[i]$ for $i \geq 1$. For an introduction see [Downey and Fellows, 1999].

Relevant results. The parameterized complexity of local search remains unexplored with exceptions that are few and far between. As it was explicitly mentioned by Marx in a recent survey on local search problems [Marx, 2008a]: “*So far, there are only a handful of parameterized complexity results in the literature, but they show that this is a fruitful research direction. The fixed-parameter tractability results are somewhat unexpected and this suggests that there are many other such results waiting to be discovered.*”

The first breakthrough in the area was done by [Marx, 2008b] who proved that finding a local improvement in k -exchange neighborhood for TSP is $W[1]$ -hard. Finding k -local optimum is not $W[1]$ -hard for every NP-hard problem. An example of this can be found in the work of [Khuller *et al.*, 2003] who investigated the NP-hard problem of finding a feedback edge set that is incident to the minimum number of vertices. One of the results obtained in [Khuller *et al.*, 2003] is that checking whether it is possible to improve a solution by replacing at most k edges can be done in time $O(n^2 + n\tau(k))$, i.e., it is FPT parameterized by k . Very recently, [Krokhnin and Marx, 2008] investigated the local search problem for finding a minimum weight assignment for a Boolean constraint satisfaction instance.

Our results. In this paper, we initiate the systematic study of parameterized complexity of local search for different graph problems. We investigate local search for problems on graphs of bounded local treewidth, a wide class of graphs containing planar graphs, graphs embeddable on a surface of bounded genus and graphs of bounded vertex degree. We show that many local search problems become FPT when the input graph is of bounded local treewidth. In particular, we show that finding k -local improvement on graphs of bounded local treewidth is FPT for many natural problems including VERTEX COVER, ODD CYCLE TRANSVERSAL, DOMINATING SET, and r -CENTER. We also show that finding k -local improvement for MAX-CUT, and MIN-BISECTION is FPT for apex-minor-free graphs. All these results are based on the idea of reducing the search in k -exchange neighborhood to searching for an improvement in a ball of small diameter around some vertex in the metric of the input graph. For example, on planar graphs, this approach leads to algorithms with running time $O(2^{O(k)} \cdot n^2)$ for many of the problems mentioned above. We also extend these results to more general classes of graphs, namely, graphs excluding a fixed graph as a minor. Finally, we show that existence of FPT algorithms is highly unlikely for larger classes of sparse graphs.

We prove that most of the problems remain $W[1]$ -hard on 3-degenerated graphs.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph where V (or $V(G)$) is the set of vertices and E (or $E(G)$) is the set of edges. We denote the number of vertices by n and the number of edges by m . For a subset $V' \subseteq V$, by $G[V']$ we mean the subgraph of G induced by V' . By $N(u)$ we denote (open) neighborhood of u that is set of all vertices adjacent to u and by $N[u] = N(u) \cup \{u\}$. Similarly, for a subset $D \subseteq V$, we define $N[D] = \cup_{v \in D} N[v]$. The *distance* $d_G(u, v)$ between two vertices u and v of G is the length of the shortest path in G from u to v . The *diameter* of a graph G , denoted by $diam(G)$, is defined to be the maximum length of a shortest path between any pair of vertices of $V(G)$. By an abuse of notation, we define diameter of a graph as the maximum of the diameters of its connected components. For $r \geq 0$, the r -neighborhood of a vertex $v \in V$ is defined as $N_G^r[v] = \{u \mid d_G(v, u) \leq r\}$. We also let $B(r, v) = N_G^r[v]$ and call it a ball of radius r around v . Similarly $B(r, A) = \cup_{v \in A} N_G^r[v]$ for $A \subseteq V(G)$. Given a weight function $w : V \rightarrow \mathbb{R}^+ \cup \{0\}$ and $A \subseteq V(G)$, $w(A) = \sum_{u \in A} w(u)$.

Given an edge $e = (u, v)$ of a graph G , the graph G/e is obtained by contracting the edge (u, v) ; that is, we get G/e by identifying the vertices u and v and removing all the loops and duplicate edges. A *minor* of a graph G is a graph H that can be obtained from a subgraph of G by contracting edges. A graph class \mathcal{C} is *minor closed* if any minor of any graph in \mathcal{C} is also an element of \mathcal{C} . A minor closed graph class \mathcal{C} is *H -minor-free* or simply *H -free* if $H \notin \mathcal{C}$. A graph H is called an apex graph if the removal of one vertex makes it a planar graph.

A *tree decomposition* of a (undirected) graph G is a pair (X, T) where T is a tree whose vertices we will call *nodes* and $X = (\{X_i \mid i \in V(T)\})$ is a collection of subsets of $V(G)$ such that (a) $\bigcup_{i \in V(T)} X_i = V(G)$, (b) for each edge $(v, w) \in E(G)$, there is an $i \in V(T)$ such that $v, w \in X_i$, and (c) for each $v \in V(G)$ the set of nodes $\{i \mid v \in X_i\}$ forms a subtree of T . The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ equals $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G . We use notation $tw(G)$ to denote the treewidth of a graph G . The definition of treewidth can be generalized to take into account the local properties of G and is called *local treewidth* [Eppstein, 2000].

Definition 1 (Local tree-width). *The local tree-width of a graph G is a function $ltw^G : \mathbb{N} \rightarrow \mathbb{N}$ which associates to every integer $r \in \mathbb{N}$ the maximum tree-width of an r -neighborhood of vertices of G , i.e. $ltw^G(r) = \max_{v \in V(G)} \{tw(G[N_G^r(v)])\}$.*

For a function $h : \mathbb{N} \rightarrow \mathbb{N}$ we define the graph class \mathcal{G}_h such that for each graph $G \in \mathcal{G}$, and for each integer $r \in \mathbb{N}$, we have $ltw^G(r) \leq h(r)$. We say that a class of graphs \mathcal{G} is of *bounded local treewidth* if $\mathcal{G} \subseteq \mathcal{G}_h$ for some $h : \mathbb{N} \rightarrow \mathbb{N}$. Well known graph classes of bounded local treewidth are planar graphs, graphs of bounded genus, and graphs of bounded

maximum vertex degree. By the result of [Robertson and Seymour, 1986] (see also [Bodlaender, 1998]), $h(r)$ can be chosen as $3r$ for planar graphs. Similarly [Eppstein, 2000] showed that $h(r)$ can be chosen as $c_g g(\Sigma)r$ for graphs embeddable in a surface Σ , where $g(\Sigma)$ is the genus of the surface Σ and c_g is a constant depending only on $g(\Sigma)$. [Demaine and Hajiaghayi, 2004] extended this result and showed that for minor closed families of graphs which do not contain some fixed apex graph as a minor, $h(r) = \mathcal{O}(r)$.

3 Framework of Study

For many optimization problems defined on graphs, the solution is a subset of vertices or edges of the graph. This is the case for the problems VERTEX COVER, INDEPENDENT SET, and DOMINATING SET to name a few. A problem P is a *vertex subset* problem (or an *edge subset* problem) if a feasible solution to P is $V' \subseteq V$ ($E' \subseteq E$). We use \mathcal{S} to denote the set of feasible solutions to a problem P .

For vertex subset (or edge subset) problems, a natural neighborhood function is obtained by exchanging k elements of the current solution. The neighborhood function in which we are interested is called *k -exchange neighborhoods* (k -ExN). We elaborate this further for minimization vertex-subset problems. Let $w : V \rightarrow \mathbb{R}^+$ be a weight function. Then the cost function $c : \mathcal{S} \rightarrow \mathbb{R}^+$ is defined as $\sum_{v \in s} w(v)$ for all $s \in \mathcal{S}$. For a pair of elements $s_1, s_2 \in \mathcal{S}$, let $H(s_1, s_2)$ denote the Hamming distance that is the size of the set $|s_1 \setminus s_2|$. We say that s' is neighbor of s with respect to k -ExN if $H(s, s') \leq k$. Let $\mathcal{N}_k^{en}(s)$ denote the set of neighbors of s with respect to k -ExN. Then the generic problem of local search for a vertex subset graph optimization problem P with respect to k -ExN is defined as follows.

k -LOCAL SEARCH P (k -LS- P)

Input: A graph $G = (V, E)$, a weight function $w : V \rightarrow \mathbb{R}^+$, a solution S and an integer $k \geq 0$.

Parameter: A positive integer k .

Question: Does there exist a solution $T \in \mathcal{N}_k^{en}(S)$ such that $c(T) < c(S)$ if P is minimization problem (and $c(S) > c(T)$ if P is maximization problem)?

In general, an input to parameterized local search problem for a graph optimization problem P consists of five parts: $(G = (V, E), w, \mathcal{M}, S, k)$, where S is an initial solution, $w : V \rightarrow \mathbb{R}^+$, \mathcal{M} is a distance function defined on the solution space \mathcal{S} and $k \in \mathbb{N}$ is a positive integer. If w is a unit weight function then we ignore w and only have four parts $(G = (V, E), \mathcal{M}, S, k)$. Unless otherwise mentioned, we only consider the Hamming distance H in this paper, and hence this term is deleted from the input.

4 FPT Algorithms for k -LS in Graphs of Bounded Local Treewidth

In this section we show that, many local search problems become fixed parameter tractable in graphs of bounded local treewidth.

4.1 Domination Problems

In this section we give an algorithm for the local search variant of a generalization of DOMINATING SET, called the

WEIGHTED r -CENTER problem. A subset $S \subseteq V$ is called an r -center if $V \subseteq B(r, S)$. In k -LS- r -CENTER, we are given an undirected graph $G = (V, E)$, with weight function $w : V \rightarrow \mathbb{R}^+$, a r -center $S \subseteq V$ and integers k, r . The problem asks whether there exists a $S' \in \mathcal{N}_k^{en}(S)$ such that $c(S') < c(S)$. Here k and r are the parameters.

When $r = 1$ this is the k -LS-DOMINATING SET problem. Our result is based on a combinatorial characterization of changed solutions. We prove that if there is an improved solution that is close to the current solution in the solution space, then there is another improved solution close to the current one in the solution space with the additional property that all changes are concentrated locally in the input graph.

Lemma 1. *Let S_1 and S_2 be r -centers of a weighted graph $G = (V, E)$ with weight function $w : V \rightarrow \mathbb{R}^+$, such that the cardinality of the symmetric difference $S_1 \Delta S_2$ is at most p , and $c(S_1) > c(S_2)$. Then there are sets $F_1, F_2 \subseteq V$ such that (a) the set $S = (S_1 \setminus F_1) \cup F_2$ is an r -center of G and $c(S) < c(S_1)$; (b) $|F_1 \cup F_2| \leq p$; and (c) there is a vertex $z \in V$ such that $F_1 \cup F_2$ is in $B(z, 2pr)$.*

Proof. Let $X = S_1 \setminus S_2$ and $Y = S_2 \setminus S_1$. Furthermore, (a) $X \cap Y = \emptyset$ and (b) $S_1 \setminus X = S_2 \setminus Y = S_1 \cap S_2$. This implies that $c(S_1) = w(S_1 \setminus X) + w(X) > c(S_2) = w(S_2 \setminus Y) + w(Y)$, and hence $w(X) > w(Y)$. Consider the auxiliary graph $G^* = (X \cup Y, E')$, where we give an edge between two vertices $u, v \in (X \cup Y)$ if $d(u, v) \leq 2r$, that is, the shortest path distance between u and v is at most $2r$ in G . Let C_1, \dots, C_p be the connected components of G^* . For every connected component C_i we assign a tuple $\mu(C_i) = (x, y)$ where $x = w(X \cap C_i)$ and $y = w(Y \cap C_i)$. Since $w(X) > w(Y)$, there exists a connected component C_j such that $x > y$ in $\mu(C_j)$. We claim that we can take $S_1 \setminus (X \cap C_j) \cup (Y \cap C_j)$ as the desired S , $F_1 = X \cap C_j$ and $F_2 = Y \cap C_j$.

First, $c(S) = w(S_1) - w(X \cap C_j) + w(Y \cap C_j) < c(S_1)$, since $x > y$. Now we show that S is a r -center of G . We know that S_1 is a r -center of G and hence only vertices which could be at distance more than r from the vertices of S are those which were distance at most r from the vertices in $(X \cap C_j)$. Let u be a vertex which is at distance more than r from any vertex in S . Then there exists a vertex, say $u_1 \in (X \cap C_j)$, such that $d(u, u_1) \leq r$. Furthermore, S_2 is an r -center not containing any of the vertices of $(X \cap C_j)$. Hence, there exists $v \in Y$ such that $d(u, v) \leq r$. But this implies that $d(u_1, v) \leq 2r$ and hence $v \in (Y \cap C_j)$. This proves that S is an r -center of G . We can select any vertex in $X \cap C_j$ for z . The size of C_j is at most p and any vertex adjacent in this component are at distance at most $2r$ in G . Hence, the ball around z of radius $2pr$ contains all the vertices of $F_1 \cup F_2$; that is, $(F_1 \cup F_2) \subseteq B(z, 2pr)$. \square

We define a generalized form of k -LS- r -CENTER which we call k -LS-GENERALIZED- r -CENTER where apart from S , we are also given a subset $T \subseteq S$, and we need to find a solution $S' \in \mathcal{N}_k^{en}(S)$ such that $c(S') < c(S)$ and $T \subseteq S'$. So an instance of k -LS-GENERALIZED- r -CENTER looks like (G, w, S, T, k) .

Lemma 2. *Let \mathcal{G} be the class of graphs which is closed under taking induced subgraphs and for which we can solve k -LS-GENERALIZED- r -CENTER in time $f(\ell) \cdot |G|^{O(1)}$ whenever $G \in \mathcal{G}$ and of diameter at most ℓ . Then k -LS- r -CENTER is FPT for \mathcal{G} .*

Proof. Let (G, w, S, k) be an instance of k -LS- r -CENTER where $G \in \mathcal{G}$. Lemma 1 implies that if the given instance is an YES instance then there exists a solution $S' \in \mathcal{N}_k^{en}(S)$ with $c(S') < c(S)$ and $z \in S \setminus S'$ such that $S \triangle S' \subseteq B(z, 4rk)$.

We go through every vertex v in S as the desired z and do as follows. We do BFS starting at v . Let the layers created by doing BFS on v be $L_0^v, L_1^v, \dots, L_t^v$. We have two cases: either (a) $t \leq 4rk + r$ or (b) $t > 4rk + r$. In case (a) we set $T_v = \emptyset$ and get (G, w, S, T_v, k) as an instance for k -LS-GENERALIZED- r -CENTER. In the other case, we first take $4rk + r$ layers; that is,

$$B(v, 4rk + r) = \cup_{j=0}^{4rk+r} L_j^v.$$

We know that all the changed vertices, those that go out and those that will come in (that is, vertices in the set $S \triangle S'$) are in $B(v, 4rk)$. Let $T_v := (S \cap (\cup_{j=4rk+1}^{4rk+r} L_j^v))$. Furthermore, for $v \in S$, let $S_v = S \cap B(v, 4rk + r)$. For every $v \in S$, we get the following instance $(B(v, 4rk + r), w, S_v, T_v, k)$ for k -LS-GENERALIZED- r -CENTER. An instance (G, w, S, k) is an YES instance for k -LS- r -CENTER if and only if there exists a $v \in S$ for which $(B(v, 4rk + r), w, S_v, T_v, k)$ is an YES instance for k -LS-GENERALIZED- r -CENTER. Since k -LS-GENERALIZED- r -CENTER is solvable in time $f(4rk + r) \cdot |B(v, 4rk + r)|^{O(1)}$ for $G[B(v, 4rk + r)]$ for $v \in S$, we have that k -LS- r -CENTER for G is solvable in time $f(4rk + r) \cdot |G|^{O(1)}$. This concludes the proof of the lemma. \square

The proof of the following lemma is similar to the proof for k -LS- r -CENTER from [Demaine *et al.*, 2005a] and we omit it here.

Lemma 3. *Let \mathcal{G} be the class of graphs of treewidth at most t , then k -LS-GENERALIZED- r -CENTER can be solved in time $\mathcal{O}((2r + 1)^t \cdot |G|^{O(1)})$ for graphs $G \in \mathcal{G}$.*

Theorem 1. *Let $h : \mathbb{N} \rightarrow \mathbb{N}$ be a given function. Then k -LS- r -CENTER can be solved in time $\mathcal{O}((2r + 1)^{h(2k)} \cdot |G|^{O(1)})$ for graphs $G \in \mathcal{G}_h$.*

Proof. Let (G, w, S, k) be an instance of k -LS- r -CENTER. Since $G \in \mathcal{G}_h$, the treewidth $\text{tw}(G[B(v, 4rk + r)]) \leq h(4rk + r)$ for all $v \in S$. Using Lemma 3 we solve k -LS-GENERALIZED- r -CENTER for instances $(B(v, 4rk + r), S_v, T_v, k)$ for every $v \in S$ in time $\mathcal{O}((2r + 1)^{h(4rk+r)} \cdot |B(v, 4rk + r)|^{O(1)})$. This in combination with Lemma 2 completes the proof. \square

Let us remark that by Theorem 1, k -LS- r -CENTER is FPT for planar graphs, graphs of bounded genus and graphs of bounded maximum degree.

In the VERTEX COVER problem one seeks for a vertex subset of minimum weight such that every edge has at least one endpoint in this set.

Our algorithm for k -LS-VERTEX COVER is based on the FPT algorithm for k -LS- r -CENTER. In fact, it is possible to give a local search preserving parameterized reduction from k -LS-VERTEX COVER to k -LS-DOMINATING SET that preserves the property of local treewidth. We omit the details here.

Theorem 2. *Let $h : \mathbb{N} \rightarrow \mathbb{N}$ be a given function such that $h(i) \geq 2$ for every i . Then k -LS-VERTEX COVER can be solved in time $\mathcal{O}(3^{h(2k)} \cdot |G|^{O(1)})$ for graphs $G \in \mathcal{G}_h$.*

4.2 Odd Cycle Transversal

In the ODD CYCLE TRANSVERSAL problem one seeks for a vertex subset S of minimum weight in a graph, such that every cycle of odd length in the graph contains at least one vertex from S . In other words, after removal of S , the remaining vertices induce a bipartite graph. This problem is interesting to look at because of the following. All the problems we considered so far have a certain “locality” involved. At first glance ODD CYCLE TRANSVERSAL look like a “non-local” problem. For ease of presentation we only deal with the unweighted case, but weighted cases can be handled in a similar manner. Our algorithm is based on a combinatorial observation, which allows us to construct local search preserving parameterized reduction to k -LS-INDEPENDENT SET. (The MAXIMUM INDEPENDENT SET problem is to find a maximum set of pairwise non-adjacent vertices in a graph.)

Given a graph $G = (V, E)$, we define a new graph $\tilde{G} = (V', E')$ as follows. Let $V_i = \{u_i \mid u \in V\}$, $i \in \{1, 2\}$. Take $V' = V_1 \cup V_2$ and $E' = \{u_1 u_2 \mid u \in V\} \cup \{u_i v_i \mid uv \in E, i \in \{1, 2\}\}$. Given a set $T \subseteq V$ such that $G[T]$ is a bipartite graph with bipartition T_1 and T_2 then by \tilde{T} denote the set $\tilde{T}_1 = \{u_1 \mid u \in T_1\} \cup \tilde{T}_2 = \{u_2 \mid u \in T_2\}$ in V' . The graph \tilde{G} has some interesting properties which we make use of when designing the local search algorithm for ODD CYCLE TRANSVERSAL.

The proofs of the following two lemmas is omitted due to space restrictions. The proof of Lemma 4 is based on a result from [Saurabh, Thesis 2008].

Lemma 4. *(G, S, k) is an YES instance of k -LS-INDUCED BIPARTITE SUBGRAPH if and only if $(\tilde{G}, \tilde{S}, k)$ is an YES instance of k -LS-INDEPENDENT SET.*

Lemma 5. *Let $h : \mathbb{N} \rightarrow \mathbb{N}$ and $g = 2h + 1$ be two given functions. If $G \in \mathcal{G}_h$, then $\tilde{G} \in \mathcal{G}_g$.*

By Theorem 2, we have that k -LS-VERTEX COVER is FPT on graphs of bounded local treewidth. Since a set S is a vertex cover of a graph $G = (V, E)$ if and only if $V \setminus S$ is an independent set, we get that the same result also holds for k -LS-INDEPENDENT SET. Combining Theorem 2 with Lemmas 4 and 5, we arrive at the following.

Theorem 3. *Let $h : \mathbb{N} \rightarrow \mathbb{N}$ be a given function. Then k -LS-ODD CYCLE TRANSVERSAL and k -LS-INDUCED BIPARTITE SUBGRAPH can be solved in time $\mathcal{O}(3^{g(2k)} \cdot |G|^{O(1)})$ for graphs $G \in \mathcal{G}_h$ where $g = 2h + 1$.*

Planar graphs, and more generally, graphs of bounded genus and apex-minor-free graphs, are in \mathcal{G}_h for some linear function h , and Theorems 1, 2, and 3 yield the following.

Corollary 1. *On apex-minor-free graphs k -LS- r -CENTER is solvable in time $r^{O(k)} \cdot |G|^{O(1)}$ and k -LS-VERTEX COVER, and k -LS-ODD CYCLE TRANSVERSAL are solvable in time $2^{O(k)} \cdot |G|^{O(1)}$.*

4.3 Graph Partitioning Problems

In this section we look at local search algorithms for graph partitioning problems such as MAX-CUT and MIN- (MAX-) BISECTION. Let $G = (V, E)$ be a given graph and $w : E \rightarrow \mathbb{R}^+$ be a weight function. Then MAX-CUT asks for a partition of V into V_1 and V_2 such that the total weight of edges (u, v) with $u \in V_1$ and $v \in V_2$ is maximized whereas in MAX- (MIN-) BISECTION the objective is to find a partition of V into V_1 and V_2 such that (a) $|V_1| = |V_2| = |V|/2$ and (b) the total weight of edges (u, v) with $u \in V_1$ and $v \in V_2$ is maximized (minimized).

We give an algorithm for a local search variant of MIN-BISECTION. Others follow along similar lines. Given a partition (V_1, V_2) let $\mathcal{E}(V_1, V_2)$ be the edges with one endpoint in V_1 and other in V_2 , and let $c((V_1, V_2)) = \sum_{e \in \mathcal{E}(V_1, V_2)} w(e)$. We now define the notion of $\mathcal{N}_k^{en}(V_1, V_2)$ for this problem. A partition $(V'_1, V'_2) \in \mathcal{N}_k(V_1, V_2)$ if there exist subsets $X \subseteq V_1$ and $Y \subseteq V_2$ such that (a) $|X| = |Y| = r$, $r \leq k$ and (b) $V'_1 = (V_1 \setminus X) \cup Y$ and $V'_2 = (V_2 \setminus Y) \cup X$.

Theorem 4. *Let $h : \mathbb{N} \rightarrow \mathbb{N}$ be a given function. Then k -LS-MINIMUM-BISECTION, k -LS-MAXIMUM-BISECTION and k -LS-MAXIMUM-CUT can be solved in time $O(2^{h(ck)} \cdot |G|^{O(1)})$ for graphs $G \in \mathcal{G}_h$ such that for every minor H of G , $H \in \mathcal{G}_h$. In particular k -LS-MINIMUM-BISECTION is FPT for planar graphs and graphs of bounded genus.*

Proof. We give the proof for k -LS-MINIMUM-BISECTION, and the proofs for other problems follow the same arguments. Let $(G, w, (V_1, V_2), k)$ be the input to k -LS-MINIMUM-BISECTION. In the first part of the proof we show that we can solve k -LS-MINIMUM-BISECTION by solving an equivalent problem on graphs of bounded diameter (or bounded treewidth).

Reducing to graphs of bounded treewidth: We start with a BFS starting at a vertex $v \in V$. Let the layers created by doing BFS on v be $L_0^v, L_1^v, \dots, L_t^v$. If $t \leq 6k + 10$, we move to the second phase of the algorithm. From now onwards we assume that $t > 6k + 3$. We create thick layers from the above layers: $W_i^v = \bigcup_{j=3i}^{3i+2} L_j^v$, where $i \in \{0, \dots, s = \lfloor \frac{t+1}{3} \rfloor\}$. The last thick layer may contain less than 3 layers. Now we make following partition of the vertex set T_q , $q \in \{0, \dots, 2k + 1\}$. We define $T_q = \bigcup W_{q+i(2k+2)}$, $i \in \{0, \dots, \lfloor \frac{s+1}{2k+2} \rfloor\}$. It is clear from the definition of T_q that it partitions the vertex set. If the input is an YES instance then there exists a partition $(V'_1, V'_2) \in \mathcal{N}_k(V_1, V_2)$ such that the total number of vertices which will flip its side (or vertices which will participate in change) is upper bounded by $2k$. Using the pigeon hole principle, we conclude that there exist T_a such that it does not contain any of these changed vertices and does not contain W_s . We can find the desired T_a by trying all T_q 's.

Now for every $W_i^v = \bigcup_{j=3i}^{3i+2} L_j^v$ contained in T_a , we remove the vertices of L_{3i+1}^v (that is, the central layer). Let this set of vertices be called V' and the resultant graph be G' with connected components C_1, \dots, C_r . We show that each connected component C_i of G' has bounded treewidth. More precisely, every connected component C_i of G' is a subset of at most $6k + 10$ layers of the BFS. If we start with G , delete all BFS layers outside of these layers and contract all BFS layers inside of these layers into v we obtain a minor H of G . H has diameter at most $6k + 11$, and also H contains C_i as an induced subgraph. Since every minor of G has bounded local treewidth, C_i has bounded treewidth, that is $\text{tw}(C_i) \leq h(6k + 10)$ for every i . The reason for removing central layers from each W_i^v is that this retains all the edges which could participate in improved cuts, as well as reduces the treewidth of the graph. In certain sense, the first and third layers of vertices in W_i^v shields the vertices of the middle layer by not participating in change. Notice that since every connected component of G' has bounded treewidth, G' itself has also bounded treewidth.

Finding appropriate solutions using Dynamic Programming: Let $\tilde{V}_1 = V_1 \setminus V'$ and $\tilde{V}_2 = V_2 \setminus V'$. Furthermore $T' = T_a \setminus V'$. Then there exists $(V'_1, V'_2) \in \mathcal{N}_k^{en}(V_1, V_2)$ such that $c(V_1, V_2) > c(V'_1, V'_2)$ if and only if there exists $(\tilde{V}'_1, \tilde{V}'_2) \in \mathcal{N}_k^{en}(\tilde{V}_1, \tilde{V}_2)$ such that $c(\tilde{V}_1, \tilde{V}_2) > c(\tilde{V}'_1, \tilde{V}'_2)$ and $T' \cap (\tilde{V}'_1 \Delta \tilde{V}_1) = \emptyset$ (that is, changed vertices should not include any vertex from T'). Searching for $(\tilde{V}'_1, \tilde{V}'_2)$ reduces to a problem in G' , a graph of bounded treewidth. We can solve this problem using standard dynamic programming over graphs of bounded treewidth in time $2^{h(6k+11)} \cdot n^{O(1)}$. We can also keep appropriate information during dynamic programming to find the desired cut explicitly. This concludes the proof of the theorem. \square

5 Limits for FPT Algorithms

The results for k -LS variant for various problems considered in the previous sections can be extended for graphs excluding a fixed graph H as a minor in a non-trivial manner. This extension is based on the structural theorem of [Robertson and Seymour, 2003], which provides a description of an H -minor free graph in terms of clique-sum decomposition over almost embeddable graphs. By making use of this decomposition it is possible to replace dynamic programming algorithms from previous section over graphs of bounded treewidth by dynamic programming over “clique-sum decomposition of H -minor graphs”. For all our algorithms for k -LS problems for H -minor free graphs, we will eventually reach a stage where we need to solve an “appropriate” problem in graphs of bounded diameter (previously at this step we applied a dynamic programming algorithm over graphs of bounded treewidth). We first obtain a clique-sum decomposition for the input graph G and then do two layer dynamic programming over clique-sum decomposition as done in [Demaine *et al.*, 2005b]. Here we use a non trivial modification of the problem to cope with its local search variant. This proof is long and technical, we postpone it for the full version of the paper.

Theorem 5. k -LS-VERTEX COVER, k -LS- r -CENTER, and k -LS-ODD CYCLE TRANSVERSAL are FPT on the class of H -minor free graphs.

Finally, we observe that, in some sense, H -minor free graphs are the most general classes of graphs for which positive results for these local search problems can be expected. We show that for more general classes of sparse graphs many local search problems become $W[1]$ -hard. Thus the existence of k -exchange FPT algorithms for these problems would contradict the widely believed assumption from parameterized complexity, namely $FPT \neq W[1]$.

A graph G is d -degenerated if every induced subgraph of G has a vertex of degree at most d .

Theorem 6. k -LS-ODD CYCLE TRANSVERSAL is $W[1]$ -hard even when restricted to 2-degenerate graphs. Furthermore, k -LS-INDEPENDENT SET and k -LS-DOMINATING SET are $W[1]$ -hard when restricted to 3-degenerate graphs.

6 Conclusions and Future Directions

In this paper we studied parameterized complexity of local search for different graph problems. We have shown that, one can search k -exchange neighborhood of a solution significantly better than the brute force for many natural problems for general classes of sparse graphs. We also provided hardness proofs that indicate that most of our results can not be further generalized. There are several open questions that need to be resolved. For instance, all our algorithmic results either depend on an observation that allows us to consider small diameter graphs, or are reducible to problems where such an approach is feasible. An important problem for which these techniques do not seem to be applicable is the TRAVELLING SALESMAN PROBLEM. Thus, the main problem we leave open in this paper is to resolve the complexity of the k -LS-TRAVELLING SALESMAN PROBLEM in planar graphs. Finally, one wonders whether running times of the form $O(2^{O(k)}n)$ are the best one can hope for local search problems in planar graphs, or whether algorithms running in time $O(2^{o(k)}n^c)$ could be feasible.

References

[Aarts and Lenstra, 1997] Emile H. L. Aarts and Jan Karel Lenstra. *Local Search in Combinatorial Optimization*. Princeton University Press, 1997.

[Bodlaender, 1998] Hans L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.

[Demaine and Hajiaghayi, 2004] Erik D. Demaine and Mohammad Taghi Hajiaghayi. Equivalence of local treewidth and linear local treewidth and its algorithmic applications. In *SODA*, pages 840–849, 2004.

[Demaine et al., 2005a] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r) -center in planar graphs and map graphs. *ACM Transactions on Algorithms*, 1(1):33–47, 2005.

[Demaine et al., 2005b] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and -minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[Downey and Fellows, 1999] Rod G. Downey and Michael R. Fellows. *Parameterized complexity*. Springer-Verlag, New York, 1999.

[Eppstein, 2000] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.

[Johnson et al., 1988] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988.

[Khanna et al., 1998] Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998.

[Khuller et al., 2003] Samir Khuller, Randeep Bhatia, and Robert Pless. On local search and placement of meters in networks. *SIAM J. Comput.*, 32(2):470–487, 2003.

[Krokhin and Marx, 2008] Andrei Krokhin and Dániel Marx. On the hardness of losing weight. In *ICALP*, volume 5125, pages 662–673, 2008.

[Lin and Kernighan, 1973] S. Lin and B. W. Kernighan. An effective heuristic algorithm for traveling-salesman problem. *Operations Research*, 21:498–516, 1973.

[Marx, 2008a] Dániel Marx. Local search. *Parameterized Complexity Newsletter*, 3:7–8, 2008.

[Marx, 2008b] Dániel Marx. Searching the k -change neighborhood for tsp is $w[1]$ -hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.

[Michiels et al., 2007] Wil Michiels, Emile H. L. Aarts, and Jan Korst. *Theoretical Aspects of Local Search*. Springer-Verlag, 2007.

[Papadimitriou and Steiglitz, 1977] Christos H. Papadimitriou and Kenneth Steiglitz. On the complexity of local search for the traveling salesman problem. *SIAM J. Comput.*, 6(1):76–83, 1977.

[Papadimitriou and Steiglitz, 1982] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, 1982.

[Robertson and Seymour, 1986] Neil Robertson and Paul D. Seymour. Graph minors. v. excluding a planar graph. *J. Comb. Theory, Ser. B*, 41(1):92–114, 1986.

[Robertson and Seymour, 2003] Neil Robertson and Paul D. Seymour. Graph minors. xvi. excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43–76, 2003.

[Saurabh, Thesis 2008] Saket Saurabh. *Exact Algorithms for Optimization and Parameterized versions of some Graph Theoretic Problems*. Homi Bhabha National Institute, Thesis, 2008.