Don't Be Strict in Local Search!

Serge Gaspers
The University of New South Wales and
Vienna University of Technology
gaspers@kr.tuwien.ac.at

Eun Jung Kim
LAMSADE-CNRS, Université de Paris-Dauphine
eunjungkim78@gmail.com

Sebastian Ordyniak
Vienna University of Technology
ordyniak@kr.tuwien.ac.at

Saket Saurabh
The Institute of Mathematical Sciences
saket@imsc.res.in

Stefan Szeider
Vienna University of Technology
stefan@szeider.net

Abstract

Local Search is one of the fundamental approaches to combinatorial optimization and it is used throughout AI. Several local search algorithms are based on searching 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.

Fellows et al. (IJCAI 2009) studied whether this brute-force search is avoidable and gave positive and negative answers for several combinatorial problems. They used the notion of local search in a strict sense. That is, an improved solution needs to be found in the k-exchange neighborhood even if a global optimum can be found efficiently.

In this paper we consider a natural relaxation of local search, called *permissive* local search (Marx and Schlotter, IWPEC 2009) and investigate whether it enhances the domain of tractable inputs. We exemplify this approach on a fundamental combinatorial problem, VERTEX COVER. More precisely, we show that for a class of inputs, finding an optimum is hard, strict local search is hard, but permissive local search is tractable.

We carry out this investigation in the framework of parameterized complexity.

Introduction

Local search is one of the most common approaches applied in practice to solve hard optimization problems. It is used as a subroutine in several kinds of heuristics, such as evolutionary algorithms and hybrid heuristics that combine local search and genetic algorithms. The history of employing local search in combinatorial optimization and operations research dates back to the 1950s with the first edge-exchange algorithms for the traveling salesperson (Bock 1958; Croes 1958).

In general, such algorithms start from a feasible solution and iteratively try to improve the current solution. Local search algorithms, also known as neighborhood search algorithms, form a large class of improvement algorithms. To perform local search, a problem specific neighborhood distance function is defined on the solution space and a better solution is searched in the neighborhood of the current solution. In particular, many local search algorithms are based on searching the *k-exchange neighborhood*. This is the set of solutions that can be obtained from the current solution by exchanging at most k elements.

Most of the literature on local search is primarily devoted to experimental studies of different heuristics. The theoretical study of local search has developed mainly in four directions.

The first direction is the study of performance guarantees of local search, i.e., the quality of the solution (Alimonti 1995; 1997; Gupta and Tardos 2000; Khanna et al. 1998; Papadimitriou and Steiglitz 1977). The second direction of the theoretical work is on the asymptotic convergence of local search in probabilistic settings, such as simulated annealing (Aarts, Korst, and van Laarhoven 1997). The third direction concerns the time required to reach a local optimum. The fourth direction is concerned with so-called kernelization techniques (Guo and Niedermeier 2007) for local search, and aims at providing the basis for putting our theoretical results to work in practice.

In a recent paper by Fellows et al. (2009) another twist in the study of local search has been taken with the goal of answering 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. Such a running time becomes a real obstacle in using k-exchange neighborhoods in practice even for very small values of k. For many years most algorithms searching an improved solution in the k-exchange neighborhood had an $n^{O(k)}$ running time, creating the impression that this cannot be done significantly faster than brute-force search.

But is there mathematical evidence for this common belief? Or is it possible for some problems to search k-exchange neighborhoods in time $O(f(k)n^c)$, 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 complexity framework, for decision problems with input size n, and a parameter k, the goal is to design an algorithm with running time $f(k)n^{\mathcal{O}(1)}$, where f 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 us to identify parameterized problems that are not amenable to such algorithms. The hardness hierarchy is represented by $W[i]$ for $i \geq 1$. The theory of parameterized complexity was developed by Downey and Fellows (1999). For recent developments, see the book by Flum and Grohe (2006).

In this paper we consider two variants of the local search problem for the well-known VERTEX COVER problem, that is, the *strict* and the *permissive* variant of local search (Marx and Schlotter 2011; Krokhin and Marx). In the strict variant the task is to either determine that there is no better solution in the k-exchange neighborhood, or to find a better solution in the k-exchange neighborhood. In the permissive variant, however, the task is to either determine that there is no better solution in the k-exchange neighborhood, or to find a better solution, which may or may not belong to the k-exchange neighborhood. Thus, permissive local search does not require the improved solution to belong to the local neighborhood, but still requires that at least the local neighborhood has been searched before abandoning the search. It can therefore be seen as a natural relaxation of strict local search with the potential to make local search applicable to a wider range of problems or instances. Indeed, we will present a class of instances for VERTEX COVER where strict local search is W[1]-hard, but permissive loal search is FPT.

In heuristic local search, there is an abundance of techniques, such as random restarts and large neighborhood search, to escape local minima and boost the performance of algorithms (Hoos and Stützle 2004). Permissive local search is a specific way to escape the strictness of local search, but allows a rigorous analysis and performance guarantees.

Relevant results. Recently, the parameterized complexity of local search has gained more and more attention. Starting with the first breakthrough in this area by Marx (2008) who investigated the parameterized complexity of TSP, several positive and negative results have been obtained in many areas of AI. For instance, the local search problem has already been investigated for a variant of the feedback edge set problem (Khuller, Bhatia, and Pless 2003), for the problem of finding a minimum weight assignment for a Boolean constraint satisfaction instance (Krokhin and Marx), for the stable marriage problem with ties (Marx and Schlotter 2011), for combinatorial problems on graphs (Fellows et al. 2009), for feedback arc set problem on tournaments (Fomin et al. 2010), for the satisfiability problem (Szeider 2011), and for Bayesian network structure learning (Ordyniak and Szeider 2010).

Our results. We investigate local search for the fundamental VERTEX COVER problem. This well-known combinatorial optimization problem has many applications (Abu-Khzam et al. 2004; Gomes et al. 2006) and is closely related to two other classic problems, INDEPENDENT SET and CLIQUE. All our results for VERTEX COVER also hold for the INDEPENDENT SET problem, and for the CLIQUE problem on the complement graph classes.

- We give the first compelling evidence that it is possible to enhance the tractability of local search problems if permissive local search is considered instead of strict local search. Indeed, the permissive variant allows us to solve the local search problem for VERTEX COVER for a significantly larger class of sparse graphs than strict local search.

- We show that the strict local search VERTEX COVER problem remains $W[1]$-hard for special sparse instances, improving a result from Fellows et al. (2009). On the way to this result we introduce a size-restricted version of a Hall set problem which be believe to be interesting in its own right.

- We answer a question of Krokhin and Marx in the affirmative, who asked whether there was a problem where finding the optimum is hard, strict local search is hard, but permissive local search is FPT.

Preliminaries

The *distance* between two sets S_1 and S_2 is $\mathsf{dist}(S_1, S_2) = |S_1 \cup S_2| - |S_1 \cap S_2|$. We say that S_1 is in the k-*exchange neighborhood* of S_2 if $\mathsf{dist}(S_1, S_2) \leq k$. If we consider a universe V with $S_1, S_2 \subseteq V$, the characteristic functions of S_1 and S_2 with respect to V are at Hamming distance at most k if $\mathsf{dist}(S_1, S_2) \leq k$.

All graphs considered in this paper are finite, undirected, and simple. Let $G = (V, E)$ be a graph, $S \subseteq V$ be a vertex set, and $u, v \in V$ be vertices. The *distance* $\mathsf{dist}(u, v)$ between u and v is the minimum number of edges on a path from u to v in G. The *(open) neighborhood* of v is $N(v) = \{ u \in V \mid uv \in E \}$, i.e., the vertices at distance one from v, and its *closed neighborhood* is $N[v] = N(v) \cup \{v\}$. We also define $N(S) = \bigcup_{u \in S} N(u) \setminus S$ and $N[S] = N(S) \cup S$. More generally, $N^d(S)$ and $N^d[S]$ denote the set of vertices at distance d and at distance at most d from a vertex in S, respectively. We write $N^d(v)$ and $N^d[v]$ for $N^d(\{v\})$ and $N^d[\{v\}]$, respectively. The *degree* of v is $d(v) = |N(v)|$. These notations may be subscripted by G, especially if the graph is not clear from the context.

The graph $G \setminus S$ is obtained from G by removing all vertices in S and all edges incident to vertices in S. The subgraph of G induced by S is $G \setminus (V \setminus S)$ and it is denoted $G[S]$. The set S is a *vertex cover* of G if $G \setminus S$ has no edge. The set S is an *independent set* of G if $G[S]$ has no edge. The graph G is *bipartite* if its vertex set can be partitioned into two independent sets A and B. In this case, we also denote the graph by a triple $G = (A, B, E)$.

The instances considered in this paper are d-degenerate graphs.

The *degeneracy* of G is the minimum d such that every subgraph of G has a vertex of degree at most d. Degeneracy is a fundamental sparsity measure of graphs. A graph G' is obtained from G by *subdividing* an edge $xy \in E$ if G' is obtained by removing the edge xy, and adding a new vertex z_{xy} and edges xz_{xy} and $z_{xy}y$. A graph G' is obtained from G by *subdividing* an edge $xy \in E$ *twice* if G' is obtained by removing the edge xy, and adding new vertices z_{xy} and z'_{xy} and edges xz_{xy}, $z_{xy}z'_{xy}$ and $z'_{xy}y$. The graph G is *2-subdivided* if G can be obtained from a graph G' by subdividing each edge of G' twice.

Hardness proofs

In this section we show that strict local search for VERTEX COVER is W[1]-hard on 2-subdivided graphs.

LS-VERTEX COVER	
Input:	A graph $G = (V, E)$, a vertex cover $S \subseteq V$ of G, and an integer k.
Parameter:	The integer k.
Question:	Is there a vertex cover $S' \subseteq V$ in the k-exchange neighborhood of S with $|S'| < |S|$?

Our proof will strengthen the following result of Fellows et al. (2009).

Theorem 1 (Fellows et al., 2009). LS-VERTEX COVER *is* W[1]-*hard and remains* W[1]-*hard when restricted to 3-degenerate graphs.*

As 2-subdivided graphs are 2-degenerate, our result implies that LS-VERTEX COVER is W[1]-hard when restricted to 2-degenerate graphs as well.

We first show that the following intermediate problem is W[1]-hard for 2-subdivided graphs.

HALL SET	
Input:	A bipartite graph $G = (A, B, E)$ and an integer k.
Parameter:	The integer k
Question:	Is there a set $S \subseteq A$ of size at most k such that $|N(S)| < |S|$?

As HALL SET is a very natural problem related to matching theory, and to give an intuition for the W[1]-hardness proof for HALL SET restricted to 2-subdivided graphs, we first show that HALL SET is W[1]-hard on general graphs.

Lemma 2. HALL SET *is* W[1]-*hard.*

Proof. We prove the lemma by a parameterized reduction from CLIQUE, which is W[1]-hard (Downey and Fellows 1999).

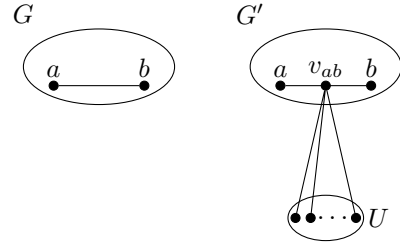| CLIQUE	
Input:	A graph G and an integer k.
Parameter:	The integer k.
Question:	Does G have a clique of size k?

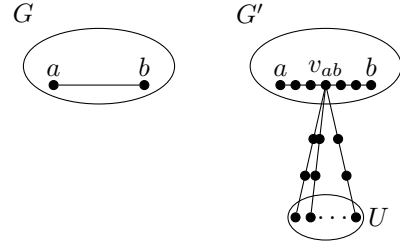Figure 1: Reduction from Lemma 2 illustrated for one edge of G.

Figure 2: Reduction from Lemma 3 illustrated for one edge of G.

Let (G, k) be an instance for CLIQUE. We construct an instance (G', k') for HALL SET as follows. Set $k' := \binom{k}{2}$. Subdivide each edge e of G by a new vertex v_e, then add a set of t new vertices $U = \{u_1, \ldots, u_t\}$, with $t := k' - k - 1$, and add an edge $v_e u$ for each $e \in E$ and $u \in U$. Set $A := \{v_e \mid e \in E\}$ and $B := V \cup U$.

Suppose G has a clique C of size k. Consider the set $S := \{v_e \in A \mid e \subseteq C\}$, i.e., the set of vertices introduced in G' to subdivide the edges of C. Then, $|S| = \binom{k}{2} = k'$. Moreover, $|N(S)| = |C \cup U| = k + t = k' - 1$. Thus, S is a Hall set of size k'.

On the other hand, suppose $S \subseteq A$ is a Hall Set of size at most k'. As $S \neq \emptyset$, we have that $U \subseteq N(S)$. Since each vertex from S has two neighbors in V, we have $|S| \leq \binom{|V \cap N(S)|}{2}$. From $|S| > t + |V \cap N(S)|$ it follows that $|S| - \binom{k}{2} + k + 1 > |V \cap N(S)|$, which can only be achieved if $|S| = \binom{k}{2}$ and $|V \cap N(S)| = k$. But then, $V \cap N(S)$ is a clique of size k in G. $\square$

We now generalize the above proof and reduce CLIQUE to HALL SET restricted to 2-subdivided graphs.

Lemma 3. HALL SET *is* W[1]-*hard even if restricted to 2-subdivided graphs.*

Proof. Let (G, k) be an instance for CLIQUE. We construct an instance (G', k') for HALL SET as follows. Set $t = \binom{k}{2} - k - 1$ and $k' = (3 + t) \cdot \binom{k}{2}$. Subdivide each edge e of G by a new vertex v_e. Then add a set of t new vertices $U = \{u_1, \ldots, u_t\}$, and add an edge $v_e u$ for each $e \in E$ and $u \in U$. This graph is bipartite with bipartition (A, B) where $A := \{v_e \mid e \in E\}$ and $B := V \cup U$. Now, make a 2-subdivision of each edge. Choose $A' \supseteq A$ and $B' \supseteq B$ so

that (A', B') is a bipartition of the vertex set of the resulting graph $G' = (V', E')$.

Suppose G has a clique C of size k. Consider the set $S := \{ v_e \in A \mid e \subseteq C \}$, i.e., the set of vertices introduced to subdivide the edges of C. Set $S' := S \cup N_{G'}^2(S)$. Then, $S' \subseteq A'$ and $|S'| = (3+t)\binom{k}{2} = k'$. Moreover, $|N(S')| = |C \cup N_{G'}(S) \cup U| = k + (2+t)\binom{k}{2} + t = k' - 1$. Thus, S' is a Hall set of size k'.

On the other hand, suppose $S \subseteq A$ is a Hall Set of G' of size at most k'. Set $S' := S$ and exhaustively apply the following rule.

Minimize If there is a vertex $v \in A$ such that $\emptyset \neq S' \cap (\{v\} \cup N^2(v)) \neq \{v\} \cup N^2(v)$, then remove $\{v\} \cup N^2(v)$ from S'.

To see that the resulting S' is a Hall Set in G', consider a set S_2 that is obtained from a Hall Set S_1 by one application of the **Minimize** rule. Suppose $v \in A$ such that $\emptyset \neq S_1 \cap (\{v\} \cup N^2(v)) \neq \{v\} \cup N^2(v)$ but $S_2 \cap (\{v\} \cup N^2(v)) = \emptyset$. If $v \notin S_1$, then removing a vertex $u \in N^2(v)$ from S_1 decreases $|S_1|$ by one and $|N(S_1)|$ by at least one; namely the vertex in $N(u) \cap N(v)$ disappears from $N(S_1)$ when removing v from S_1. After removing all vertices in $N^2(v)$ from S_1, we obtain a set S_2 such that $|N(S_1)| - |N(S_2)| \geq |S_1| - |S_2|$. As $|N(S_1)| \leq |S_1| - 1$, we obtain that $|N(S_2)| \leq |S_2| - 1$ and therefore, S_2 is a Hall Set of size at most k'. On the other hand, if $v \in S_1$, then there is a vertex $u \in N^2(v) \setminus S_1$. Then, removing v from S_1 decreases $|S_1|$ by one and $|N(S_1)|$ by at least one; namely, the vertex in $N(u) \cap N(v)$ disappears from $N(S_1)$ when removing v from S_1. Thus, $S_1 \setminus \{v\}$ is a Hall Set of size at most k' and we can appeal to the previous case with $S_1 := S_1 \setminus \{v\}$.

We have now obtained a Hall set S' of size at most k' such that for every $v \in A$, either $S' \cap (\{v\} \cup N^2(v)) = \emptyset$ or $\{v\} \cup N^2(v) \subseteq S'$

As $S' \neq \emptyset$, we have that $U \subseteq N(S')$. Expressing the size of S' in terms of vertices from A we obtain that

$$|S'| = (3+t)|S' \cap A|.$$

Similarly, we express $|N(S')|$, which contains all vertices from $N(S' \cap A)$, all vertices in U and some vertices from $V \cap N^3(S')$.

$$|N(S')| = (2+t)|S' \cap A| + t + |V \cap N^3(S')|.$$

Now,

$$|S'| > |N(S')|$$

$$(3+t)|S' \cap A| > (2+t)|S' \cap A| + t + |V \cap N^3(S')|$$

$$|S' \cap A| - \binom{k}{2} > |V \cap N^3(S')| - k - 1.$$

Since there are two vertices in $V \cap N^3(v)$ for each vertex $v \in S' \cap A$, we have $|S' \cap A| \leq \binom{|V \cap N^3(S')|}{2}$. Moreover, $|S' \cap A| \leq \frac{k'}{3+t} = \binom{k}{2}$. Therefore, the previous inequality can only be satisfied if $|S' \cap A| = \binom{k}{2}$ and $|V \cap N^3(S')| = k$. Then, $|V \cap N^3(S')|$ is a clique of size k in G. $\qquad \square$

Finally, we rely on the previous lemma to establish W[1]-hardness of LS-VERTEX COVER for 2-subdivided graphs. The reduction will make clear that the HALL SET problem captures the essence of the LS-VERTEX COVER problem.

Theorem 4. LS-VERTEX COVER *is* W[1]-*hard when restricted to 2-subdivided graphs.*

Proof. The proof uses a reduction from HALL SET restricted to 2-subdivided graphs. Let (G, k) be an instance for HALL SET where $G = (A, B, E)$ is a 2-subdivided graph. The set A is a vertex cover for G. Consider (G, A, k'), with $k' := 2k - 1$ as an instance for LS-VERTEX COVER.

Let $S \subseteq A$ be a Hall Set of size at most k for G, i.e., $|N(S)| < |S|$. Then, $(A \setminus S) \cup N(S)$ is a vertex cover for G of size at most $|A| - 1$. Moreover, this vertex cover is in the k'-exchange neighborhood of A.

On the other hand, let C be a vertex cover in the k'-exchange neighborhood of A such that $|C| < |A|$. Set $C' := C$. If $|A \setminus C| > k$, then add $|A \setminus C| - k$ vertices from $A \setminus C$ to C'. The resulting set C' is also in the k'-exchange neighborhood of A and $|C'| < |A|$. Set $S := A \setminus C'$. Then, $|S| \leq k$. As C' is a vertex cover, $N(S) \subseteq C'$. But since C' is smaller than A, we have that $|C' \cap B| \leq |S| - 1$ Therefore, $|N(S)| \leq |C' \cap B| \leq |S| - 1$, which shows that S is a Hall Set of size at most k for G. $\qquad \square$

FPT Algorithm

In this section we will show that the permissive version of LS-VERTEX COVER is fixed-parameter tractable for a generalization of 2-subdivided graphs.

PLS-VERTEX COVER	
Input:	A graph G, a vertex cover S, and a positive integer k.
Parameter:	The integer k.
Task:	Determine that G has no vertex cover S' with $\mathsf{dist}(S, S') \leq k$ and $|S'| < |S|$ or find a vertex cover S'' with $|S''| < |S|$.

The initial algorithm will be randomized, and we will exploit the following pseudo-random object and theorem to derandomize it.

Definition 1 (Naor, Schulman, and Srinivasan, 1995)**.** *An (n, t)-universal set $\mathcal{F}$ is a set of functions from $\{1, \ldots, n\}$ to $\{0, 1\}$, such that for every subset $S \subseteq \{1, \ldots, n\}$ with $|S| = t$, the set $\mathcal{F}|_S = \{ f|_S \mid f \in \mathcal{F} \}$ is equal to the set 2^S of all the functions from S to $\{0, 1\}$.*

Theorem 5 (Naor, Schulman, and Srinivasan, 1995)**.** *There is a deterministic algorithm with running time $O(2^t t^{O(\log t)} n \log n)$ that constructs an (n, t)-universal set $\mathcal{F}$ such that $|\mathcal{F}| = 2^t t^{O(\log t)} \log n$.*

Our FPT algorithm will take as input a β-separable graph.

Definition 2. *For a fixed non-negative integer β, a graph $G = (V, E)$ is β-separable if there exists a bipartition of V into V_1 and V_2 such that*

- *for each $v \in V_1$, $|N(v) \cap V_1| \leq \beta$, and*
- *for each $w \in V_2$, $|N(w)| \leq \beta$.*

A bipartition of V satisfying these properties is a partition certifying β-separability. By $\mathcal{G}(\beta)$ we denote the set of all β-separable graphs.

Remark 1. Observe that a graph of degree at most d is d-separable. Similarly every 2-subdivided graph is 2-separable.

The following lemma characterizes solutions for PLS-VERTEX COVER that belong to the k-exchange neighborhood of S.

Lemma 6. *Let $G = (V, E)$ be a graph, S be a vertex cover of G and k be a positive integer. Then there exists a vertex cover S' such that $|S'| < |S|$ and $\mathsf{dist}(S, S') \leq k$ if and only if there exists a set $S^* \subseteq S$ such that*

1. *S^* is an independent set,*
2. *$|N(S^*) \setminus S| < |S^*|$, and*
3. *$|N(S^*) \setminus S| + |S^*| \leq k$.*

Proof. We first show the forward direction of the proof. Let $S^* = S \setminus S'$. Since $I = V \setminus S'$ is an independent set and $S^* \subseteq I$ we have that S^* is an independent set. Furthermore, since S^* is in I we have that $N(S^*) \subseteq S'$ and in particular $N(S^*) \setminus S$ is the set of vertices that are present in S' but not in S. Since $|S'| < |S|$ we have that $|N(S^*) \setminus S| < |S^*|$ and by the fact that $\mathsf{dist}(S, S') \leq k$ we have that $|N(S^*) \setminus S| + |S^*| \leq k$. For the reverse direction it is easy to see that $(S \setminus S^*) \cup (N(S^*) \setminus S)$ is the desired S'. This completes the proof. $\qquad\square$

To obtain the FPT algorithm for PLS-VERTEX COVER on $\mathcal{G}(\beta)$ we will use Lemma 6. More precisely, our strategy is to obtain an FPT algorithm for finding a subset $Q \subseteq S$ such that Q is an independent set and $S^* \subseteq Q$. Here, S^* is as described in Lemma 6. Thus, our main technical lemma is the following.

Lemma 7. *Let β be a fixed non-negative integer. Let G be a β-separable graph, S be a vertex cover of G and k be a positive integer. There is a $O(2^q q^{O(\log q)} n \log n)$ time algorithm finding a family $\mathcal{Q}$ of subsets of S such that (a) $|\mathcal{Q}| \leq 2^q q^{O(\log q)} \log n$, (b) each $Q \in \mathcal{Q}$ is an independent set, and (c) if there exists a S^* as described in Lemma 6, then there exists a $Q \in \mathcal{Q}$ such that $S^* \subseteq Q$. Here, $q = k + \beta k$.*

We postpone the proof of Lemma 7 and first give the main result that uses Lemma 7 crucially.

Theorem 8. *Let β be a fixed non-negative integer. PLS-VERTEX COVER is FPT on $\mathcal{G}(\beta)$ with an algorithm running in time $2^q q^{O(\log q)} n^{O(1)}$, where $q = k + \beta k$.*

Proof. Let G be the input graph from $\mathcal{G}(\beta)$, S be a vertex cover of G, and k be a positive integer. Fix $q = k + \beta k$ and $I = V \setminus S$. We first apply Lemma 7 and obtain a family $\mathcal{Q}$ of subsets of S such that (a) $|\mathcal{Q}| \leq 2^q q^{O(\log q)} \log n$ and (b) each $Q \in \mathcal{Q}$ is an independent set. The family Q has the additional property that if there exists a set S^* as described in Lemma 6, then there exists a $Q \in \mathcal{Q}$ such that $S^* \subseteq Q$.

For every $Q \in \mathcal{Q}$, the algorithm proceeds as follows. Consider the bipartite graph $G[Q \cup I]$. Now in polynomial time check whether there exists a subset $W \subseteq Q$ such that

$|N(W)| < |W|$ in $G[Q \cup I]$. This is done by checking Halls' condition that says that there exists a matching saturating Q if and only if for all $A \subseteq Q$, $|N(A)| \geq |A|$. A polynomial time algorithm that finds a maximum matching in a bipartite graph can be used to find a violating set A if there exists one. See Kozen (1991) for more details. Returning to our algorithm, if we find such set W then we return $S' = (S \setminus W) \cup N(W)$. Clearly, S' is a vertex cover and $|S'| < |S|$. Now we argue that if for every $Q \in \mathcal{Q}$ we do not obtain the desired W, then there is no vertex cover S' such that $|S'| < |S|$ and $\mathsf{dist}(S, S') \leq k$. However, this is guaranteed by the fact that if there would exist such a set S', then by Lemma 6 there exist a desired S^*. Thus, when we consider the set $Q \in \mathcal{Q}$ such that $S^* \subseteq Q$ then we would have found a $W \subseteq Q$ such that $|N(W)| < |W|$ in $G[Q \cup I]$. This proves the correctness of the algorithm. The running time of the algorithm is governed by the size of the family $\mathcal{Q}$. This completes the proof. $\qquad\square$

To complete the proof of Theorem 8, the only remaining component is a proof of Lemma 7 which we give below.

Proof of Lemma 7. Let G be a β-separable graph, S be a vertex cover of G, and k be a positive integer. By the proof of Lemma 6 we know that if there exists a vertex cover S' such that $|S'| < |S|$ and $\mathsf{dist}(S, S') \leq k$ then there exists a set $S^* \subseteq S$ such that

1. *S^* is an independent set,*
2. *$|N(S^*) \setminus S| < |S^*|$, and*
3. *$|N(S^*) \setminus S| + |S^*| \leq k$.*

We first give a randomized procedure that produces a family $\mathcal{Q}$ satisfying the properties of the lemma with high probability. In a second stage, we will derandomize it using universal sets. For our argument we fix one such S^* and let V_1 and V_2 be a partition certifying β-separability of G. Let $S_1 = S^* \cap V_1$ and $S_2 = S^* \cap V_2$. Since G is a β-separable graph, we have that $|N[S_1] \cap (V_1 \cap S)| + |N[S_2] \cap S| \leq \beta|S_1| + |S_1| + \beta|S_2| + |S_2| \leq k + \beta k$. We also know that $|S^*| \leq k$. Let $q = k + \beta k$ and $A = (N[S_1] \cap (V_1 \cap S)) \cup (N[S_2] \cap S)$. Now, uniformly at random color the vertices of S with $\{0, 1\}$, that is, color each vertex of S with 0 with probability $\frac{1}{2}$ and with 1 otherwise. Call this coloring f. The probability that for all $x \in S^*$, $f(x) = 0$ and for all $y \in (A \setminus S^*)$, $f(y) = 1$, is

$$\frac{1}{2^{|A|}} \geq \frac{1}{2^q}.$$

Given the random coloring f we obtain a set $Q(f) \subseteq S$ with the following properties

- $Q(f)$ is an independent set; and
- with probability at least 2^{-q}, $S^* \subseteq Q(f)$.

We obtain the set $Q(f)$ as follows.

Let $C_0 = \{ v \mid v \in S, \ f(v) = 0 \}$, that is, C_0 contains all the vertices of S that have been assigned 0 by f. Let $C_0^1 \subseteq C_0 \cap V_2$ be the set of vertices that have degree at least 1 in $G[C_0]$. Let $C_0' := C_0 \setminus C_0^1$. Let E_0' be the set of edged in the induced graph $G[C_0']$ and $V(E_0')$ be the

set of end-points of the edges in E_0'. Define $Q(f) := C_0' \setminus V(E_0')$.

By the procedure it is clear that $Q(f)$ is an independent set. However, note that it is possible that $Q(f) = \emptyset$. Now we show that with probability at least e^{-q}, $S^* \subseteq Q(f)$. Let $C_i = \{ v \mid v \in S, \ f(v) = i \}$, $i \in \{0, 1\}$. By the probability computation above we know that with probability at least e^{-q}, $S^* \subseteq C_0$ and $A \setminus S^* \subseteq C_1$. Now we will show that the procedure that prunes C_0 and obtains $Q(f)$ does not remove any vertices of S^*. All the vertices in the set $N(S_2) \cap S$ are contained in C_1 and thus there are no edges incident to any vertex in S_2 in $G[C_0]$. Therefore the only other possibility is that we could remove vertices of $S_1 \cap C_0$. However, to do so there must be an edge between a vertex in S_1 and a vertex in $V_1 \cap S$, but we know that all such neighbors of vertices of S_1 are in C_1. This shows that with probability at least 2^{-q}, $S^* \subseteq Q(f)$.

We can boost the success probability of the above random procedure to a constant, by independently repeating the procedure 2^q times. Let the random functions obtained while repeating the above procedure be f_j, $j \in \{1, \ldots, 2^q\}$ and let $Q(f_j)$ denote the corresponding set obtained after applying the above pruning procedure. The probability that one of the $Q(f_j)$ contains S^* is at least

$$1 - \left(1 - \frac{1}{2^q}\right)^{2^q} \geq 1 - \frac{1}{e} \geq \frac{1}{2}.$$

Thus we obtain a collection $\mathcal{Q}$ of subsets of S with the following properties.

- $|\mathcal{Q}| \leq 2^q$, where $\mathcal{Q} = \{ Q(f_j) \mid j \in \{1, \ldots, 2^q\} \}$,
- every set $Q \in \mathcal{Q}$ is an independent set, and
- with probability at least $\frac{1}{2}$, there exists a set $Q \in \mathcal{Q}$ such that $S^* \subseteq Q$.

Finally, to derandomize the above procedure we will use Theorem 5. We first compute a $(|S|, q)$-universal set $\mathcal{F}$ with the algorithm described in Theorem 5 in time $O(2^q q^{O(\log q)} |S| \log |S|)$ of size $2^q q^{O(\log q)} \log |S|$. Now every function $f \in \mathcal{F}$ can be thought of as a function from S to $\{0, 1\}$. Given this f we obtain $Q(f)$ as described above. Let $\mathcal{Q} = \{ Q(f) \mid f \in \mathcal{F} \}$. Clearly, $|\mathcal{Q}| \leq 2^q q^{O(\log q)} \log n$. Now if there exists a set S^* of the desired type then the $Q(f)$ corresponding to the function $f \in \mathcal{F}$, that assigns 0 to every vertex in S^* and 1 to every vertex in $A \setminus S^*$, has the property that $S^* \subseteq Q(f)$ and $Q(f)$ is an independent set. This completes the proof. □

It is easily seen that finding minimum vertex cover of a 2-subdivided graph is NP-hard. Indeed, it follows from the NP-hardness of the VERTEX COVER problem on general graphs since: if G' is a 2-subdivision of a graph G with m edges, then G has a vertex cover of size at most k if and only if G' has a vertex cover of size at most $k + m$.

Thus, Theorems 4 and 8 together resolve a question raised by (Krokhin and Marx), who asked for a problem where finding the optimum is hard, strict local search is hard, but permissive local search is FPT.

Conclusion

In this paper we have shown that from the parameterized complexity point of view, permissive Local Search is indeed more powerful than the strict Local Search and thus may be more desirable. We have demonstrated this on one example, namely VERTEX COVER, but it would be interesting to find a broader set of problems where the complexity status of the strict and permissive versions of local search differ. We believe that the results in this paper have opened up a complete new direction of research in the domain of parameterized local search, which is still in nascent stage. It would be interesting to undertake a similar study for FEEDBACK VERTEX SET, even on planar graphs.

References

Aarts, E. H. L.; Korst, J.; and van Laarhoven, P. J. M. 1997. Simulated annealing. *Local Search in Combintaorial Optimization, Wiley* 91–120.

Abu-Khzam, F. N.; Collins, R. L.; Fellows, M. R.; Langston, M. A.; Suters, W. H.; and Symons, C. T. 2004. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proceedings of the 6th Workshop on Algorithm Engineering and Experiments (ALENEX 2004)*, 62–69. SIAM.

Alimonti, P. 1995. Non-oblivious local search for graph and hypergraph coloring problems. In *WG*, volume 1017 of *Lecture Notes in Computer Science*, 167–180. Springer Verlag.

Alimonti, P. 1997. Non-oblivious local search for MAX 2-CCSP with application to max dicut. In *WG*, volume 1335 of *Lecture Notes in Computer Science*, 2–14. Springer Verlag.

Bock, F. 1958. An algorithm for solving 'traveling-salesman' and related netw ork optimization problems. *Research report, Armour Research Foundation*.

Croes, G. 1958. A method for solving traveling-salesman problems. *Operations Research* 6:791–812.

Downey, R. G., and Fellows, M. R. 1999. *Parameterized Complexity*. Monographs in Computer Science. New York: Springer Verlag.

Fellows, M. R.; Rosamond, F. A.; Fomin, F. V.; Lokshtanov, D.; Saurabh, S.; and Villanger, Y. 2009. Local search: Is brute-force avoidable? In Boutilier, C., ed., *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, 486–491.

Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*, volume XIV of *Texts in Theoretical Computer Science. An EATCS Series*. Berlin: Springer Verlag.

Fomin, F. V.; Lokshtanov, D.; Raman, V.; and Saurabh, S. 2010. Fast local search algorithm for weighted feedback arc set in tournaments. In Fox, M., and Poole, D., eds., *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.* AAAI Press.

Gomes, F. C.; de Meneses, C. N.; Pardalos, P. M.; and Viana, G. V. R. 2006. Experimental analysis of approximation algorithms for the vertex cover and set covering problems. *Computers & OR* 33(12):3520–3534.

Guo, J., and Niedermeier, R. 2007. Invitation to data reduction and problem kernelization. *ACM SIGACT News* 38(2):31–45.

Gupta, A., and Tardos, É. 2000. A constant factor approximation algorithm for a class of classification problems. In *STOC*, 652–658.

Hoos, H. H., and Stützle, T. 2004. *Stochastic Local Search: Foundations and Applications*. Elsevier/Morgan Kaufmann.

Khanna, S.; Motwani, R.; Sudan, M.; and Vazirani, U. V. 1998. On syntactic versus computational views of approximability. *SIAM J. Comput.* 28(1):164–191.

Khuller, S.; Bhatia, R.; and Pless, R. 2003. On local search and placement of meters in networks. *SIAM J. Comput.* 32(2):470–487.

Kozen, D. C. 1991. *The Design and Analysis of Algorithms (Monographs in Computer Science)*. Springer-Verlag.

Krokhin, A., and Marx, D. On the hardness of losing weight. To appear in *ACM Transactions on Algorithms*.

Marx, D., and Schlotter, I. 2011. Stable assignment with couples: Parameterized complexity and local search. *Discrete Optimization* 8(1):25–40.

Marx, D. 2008. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.* 36(1):31–36.

Naor, M.; Schulman, L. J.; and Srinivasan, A. 1995. Splitters and near-optimal derandomization. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS 1995)*. IEEE Computer Society. 182–191.

Ordyniak, S., and Szeider, S. 2010. Algorithms and complexity results for exact bayesian structure learning. In Grünwald, P., and Spirtes, P., eds., *Proceedings of UAI 2010, The 26th Conference on Uncertainty in Artificial Intelligence, Catalina Island, California, USA, July 8-11, 2010.* AUAI Press, Corvallis, Oregon.

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

Szeider, S. 2011. The parameterized complexity of k-flip local search for sat and max sat. *Discrete Optimization* 8(1):139–145.