Finding 3-Swap-Optimal Independent Sets and Dominating Sets Is Hard

Christian Komusiewicz
Philipps-Universität Marburg, Fachbereich Mathematik und Informatik, Marburg, Germany

Nils Morawietz
Philipps-Universität Marburg, Fachbereich Mathematik und Informatik, Marburg, Germany

Abstract

For PLS-complete local search problems, there is presumably no polynomial-time algorithm which finds a locally optimal solution, even though determining whether a solution is locally optimal and replacing it by a better one if this is not the case can be done in polynomial time.

We study local search for Weighted Independent Set and Weighted Dominating Set with the 3-swap neighborhood. The 3-swap neighborhood of a vertex set S in G is the set of vertex sets which can be obtained from S by exchanging at most three vertices. We prove the following dichotomy: On the negative side, the problem of finding a 3-swap-optimal independent set or dominating set is PLS-complete. On the positive side, locally optimal independent sets or dominating sets can be found in polynomial time when allowing all 3-swaps except a) the swaps that remove two vertices from the current solution and add one vertex to the solution or b) the swaps that remove one vertex from the current solution and add two vertices to the solution.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Discrete optimization; Theory of computation → Graph algorithms analysis

Keywords and phrases Local Search, Graph problems, 3-swap neighborhood, PLS-completeness

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.66

Funding *Nils Morawietz*: Supported by the Deutsche Forschungsgemeinschaft (DFG), project OPERAH, KO 3669/5-1.

1 Introduction

Local search is one of the most successful paradigms for developing algorithms for NP-hard optimization problems. In the most fundamental version of local search, the hill-climbing algorithm, one starts by computing some feasible solution to the problem at hand. This solution is then replaced by a better solution in its local neighborhood, as long as such a better solution exists. The final output is a locally optimal solution. Even though such a locally optimal solution might be arbitrarily bad in comparison to a globally optimal solution, local search approaches turned out to find good solutions in practice [1, 3, 6]. A crucial aspect in this approach is the definition of the local neighborhood of solutions. Intuitively, there is the following trade-off: for more restricted neighborhoods, computing a local optimum should be easier but the quality of the local optimum may be worse than for less restricted neighborhoods. Note that the choice of the neighborhood may affect the computational difficulty of each improvement step as well as the number of necessary improvement steps. Naturally, in applications of hill-climbing, the neighborhoods are chosen in such a way that each improvement step can be performed efficiently.

To study the difficulty of computing locally optimal solutions in such a setting, Johnson et al. [5] introduced the complexity class PLS. This class contains all local search problems for which one can compute some starting solution and search the local neighborhood of a feasible solution S in polynomial time. Thus, the local search problems in PLS are exactly those that have a hill-climbing algorithm where each step only takes polynomial time.

For many unweighted problems, membership in PLS directly implies a polynomial-time algorithm for computing locally optimal solutions, since the maximum objective value is bounded by a polynomial of the input size. The situation is different for weighted problems, where we may need a superpolynomial number of improvement steps to reach a local optimum. To give evidence that for some local search problems in PLS it may be hard to compute locally optimal solutions, Johnson et al. [5] introduced PLS-reductions and showed that there are PLS-complete problems. These problems are as hard as any problem in PLS and none of them is known to be solvable in polynomial time.

We study two famous graph problems, Weighted Independent Set and Weighted Dominating Set through the lens of PLS-completeness. The solutions in both problems are vertex sets and we consider the most fundamental neighborhoods for such solutions: k-swap-neighborhoods, where we may add or remove up to k vertices from a solution.

Previous Work. PLS-completeness has been shown for a number of problems [4, 5, 11, 13, 14]. The initial PLS-complete problem is called Flip, where the input is a Boolean circuit [5]. Another prominent PLS-complete problem is Max Cut with the flip neighborhood, where the neighbors of a partition (A, B) are the partitions that can be obtained by moving one vertex from A to B or vice versa [13]. Max Cut with the flip neighborhood is PLS-complete on graphs with maximum degree 5 [4] and polynomial-time solvable on 3-regular graphs [12].

Johnson et al. [5] already considered the Weighted Independent Set problem and argued that one can show PLS-completeness for a neighborhood that is inspired by the Kernighan-Lin algorithm for Max Cut [7]. Moreover, Johnson et al. [5] specifically called for the study of simpler neighborhoods for the Weighted Independent Set problem. Schäffer and Yannakakis [13] argued that Weighted Independent Set is PLS-complete for a 2-step neighborhood which consists of an addition of a vertex v to the independent set S and a removal of all its neighbors from S in the first step and a (maximal) series of improving vertex additions in the second step. Note that the first step is not necessarily improving. Further studies have shown PLS-completeness for Weighted Independent Dominating Set with a k-swap neighborhood with constant but unspecified k [8] and for Weighted Max-Π-Subgraph with hereditary properties Π and the above-described 2-step-neighborhood [14]. We are not aware of any results for the k-swap neighborhoods with small constant k considered in this work.

From a more applied point of view, local search has been shown to give very good results for Independent Set [1, 2, 3, 6, 9] and weighted Weighted Independent Set [10]. In particular, for Independent Set, Andrade et al. [1] presented fast algorithms for finding improving 5-swaps and showed that the subroutine of finding 5-swap optimal solutions gives very good results when used in an iterative local search framework. Later, it was shown that k-swap-optimal solutions for Independent Set can be efficiently computed for k up to 25 and that, on a set of large sparse real-world networks globally optimal solutions can be found via a simple hill-climbing algorithm for $k \geq 9$ [6].

Our Results. We provide a complexity analysis for Weighted Independent Set with the k-swap neighborhood, denoted Weighted Independent Set/k-swap. Our main result is the PLS-completeness of Weighted Independent Set/3-swap on graphs of constant[1] maximum degree. We first show in Section 3 that Weighted Independent Set/7-swap is PLS-complete on graphs of maximum degree at most 6. Then, in Section 4, we extend the constructed instance of Weighted Independent Set/7-swap to obtain PLS-completeness for Weighted Independent Set/3-swap. Next, we show that by a small modification of the construction, the PLS-completeness for 3-swaps can be transferred to Weighted Dominating Set. Finally, in Section 5, we show that if we allow all 3-swaps except either a) the swaps that remove two vertices and add one or b) the swaps that remove one vertex and add two, we can find locally optimal solutions for Weighted Independent Set and Weighted Dominating Set in polynomial time. We extend this result to a slightly more general neighborhood and a certain type of subset optimization problems. Proofs of statements marked with a "(*)" are deferred to the full version.

2 Preliminaries

For integers i and j with $i \leq j$, we define $[i, j] := \{k \in \mathbb{N} \mid i \leq k \leq j\}$. For two sets A and B, we denote with $A \oplus B := (A \setminus B) \cup (B \setminus A)$ the *symmetric difference* of A and B.

An (undirected) graph $G = (V, E)$ consists of a set of vertices V and a set of edges $E \subseteq \{\{u, v\} \mid u \in V, v \in V, u \neq v\}$. For vertex sets $S \subseteq V$ and $T \subseteq V$ we denote with $E_G(S, T) := \{\{s, t\} \in E \mid s \in S, t \in T\}$ the edges between S and T. Moreover, we define $G[S] := (S, E_G(S, S))$ as the *subgraph of G induced by S*. For a vertex $v \in V$, we denote with $N_G(v) := \{w \in V \mid \{v, w\} \in E\}$ the *open neighborhood* of v in G and with $N_G[v] := \{v\} \cup N_G(v)$ the *closed neighborhood* of v in G. Analogously, for a vertex set $S \subseteq V$, we define $N_G[S] := \bigcup_{v \in S} N_G[v]$ and $N_G(S) := \bigcup_{v \in S} N_G(v) \setminus S$. If G is clear from the context, we may omit the subscript.

A vertex set $S \subseteq V$ is an *independent set* in G if there is no edge between any pair of vertices of S in G and a *clique* in G if there is an edge between each pair of vertices of S in G. A vertex set $S \subseteq V$ is a *dominating set* in G if for each vertex v of G, at least one vertex of $N[v]$ is contained in S.

The two main problems considered here are now defined as follows.

Weighted Independent Set
Input: A graph $G = (V, E)$ and a vertex-weight function $\omega : V \to \mathbb{N}$.
Output: An independent set in G of maximum total weight.

Weighted Dominating Set
Input: A graph $G = (V, E)$ and a vertex-weight function $\omega : V \to \mathbb{N}$.
Output: A dominating set in G of minimum total weight.

An *optimization problem* L consists of a set $\mathcal{D}_L$ of *instances*, for each instance $I \in \mathcal{D}_L$, a set of *feasible solutions* $\mathcal{S}_L(I)$ for I, an *objective function* val_L which assigns a non-negative rational number to each pair (I, s), and is specified to be either a *minimization* or a *maximization* problem. An optimization problem L is an NP-*optimization problem* if the encoding length of each solution $s \in \mathcal{S}_L(I)$ of I is polynomially bounded by $|I|$, one can determine in polynomial time for each pair (I, s) whether $s \in \mathcal{S}_L(I)$, and the objective function can be evaluated in polynomial time.

[1] Our proof gives a degree bound of 3140.

Let I be an instance and of an optimization problem L and let s and s' be feasible solutions for I. We say that *s is better than s'* if a) L is a maximization problem and $\mathrm{val}_L(I,s) > \mathrm{val}_L(I,s')$ or b) L is a minimization problem and $\mathrm{val}_L(I,s) < \mathrm{val}_L(I,s')$.

▶ **Definition 2.1.** *An* NP*-optimization problem L is a* subset-weight optimization problem *if it consists of*

- *a polynomial-time computable function U that maps each instance I of L to a universe $U(I)$ and each feasible solution of I is a subset of $U(I)$,*
- *a polynomial-time computable function f which checks for an instance I of L and a set $S \subseteq U(I)$ if S is a feasible solution for I,*
- *a polynomial-time computable function g which computes for an instance I of L some feasible solution for I, and*
- *a polynomial-time computable weight function ω which assigns a non-negative rational weight to each pair (I,u), where I is an instance of L and u is an element of $U(I)$*

and one wants to find a feasible solution S for I of either minimal or maximal weight, where the weight of S is defined as $\omega(I,S) := \sum_{u\in S} \omega(I,u)$.

If one wants to find a feasible solution of maximal weight, L is a *subset-weight maximization problem*; otherwise, L is a *subset-weight minimization problem*.

Weighted Independent Set is a subset-weight maximization problem: the feasible solutions are the independent sets of G, these are all subsets of the vertex set V (the universe), one can check in polynomial time if a vertex set S is an independent set, and the total weight of S is defined as the sum of the weights of the vertices of S. Similarly, Weighted Dominating Set is a subset-weight minimization problem.

Let L be a subset-weight optimization problem, let I be an instance of L, and let $S \subseteq U(I)$ be a feasible solution for I. A *k-swap*, for $k \in \mathbb{N}$, is a subset $W \subseteq U(I)$ of size at most k. We say that W is *valid for S in I* if $S \oplus W$ is also a feasible solution for I. We say that two feasible solutions S and S' for I are *k-neighbors in I* if $W := S \oplus S'$ has size at most k. Additionally, we say that S' is an *improving k-neighbor of S in I* and that W is an *improving k-swap* if the total weight of S' is better than the total weight of S. If there is no improving k-neighbor of S in I, S is *k-optimal in I*. Let S be a subset of $U(I)$ and let k, k_{in}, and k_{out} be natural numbers such that $k_{\mathrm{in}} + k_{\mathrm{out}} \leq k$. A k-swap W is a *$(k_{\mathrm{in}}, k_{\mathrm{out}})$-swap for S in G*, if $|W \setminus S| \leq k_{\mathrm{in}}$ and if $|W \cap S| \leq k_{\mathrm{out}}$. Similar to k-swaps, we also define the notions of *valid $(k_{\mathrm{in}}, k_{\mathrm{out}})$-swaps*, *improving $(k_{\mathrm{in}}, k_{\mathrm{out}})$-swaps*, *$(k_{\mathrm{in}}, k_{\mathrm{out}})$-neighbors*, *improving $(k_{\mathrm{in}}, k_{\mathrm{out}})$-neighbors*, and *$(k_{\mathrm{in}}, k_{\mathrm{out}})$-optimal*.

A *partition of a graph $G = (V,E)$* is a pair (A,B), where $A \cup B = V$ and $A \cap B = \emptyset$. The *cut* of a partition (A,B) is the edge set $E_G(A,B)$, that is, the set of edges having one endpoint in A and one endpoint in B. Let $\omega : E \to \mathbb{N}$ be an edge-weight function. A *flip of a vertex $v \in V$ in a partition* (A,B) is the partition (A',B'), where $A' := A \oplus \{v\}$ and $B' := B \oplus \{v\}$. Moreover, we say that (A',B') *is improving over* (A,B) if the total weight of cut $E_G(A',B')$ is larger than the total weight of the cut $E_G(A,B)$, that is, if $\omega(E_G(A',B')) > \omega(E_G(A,B))$. Furthermore, we say that a partition (A,B) is *flip-optimal* if there is no vertex $v \in V$ such that the flip (A',B') of v in (A,B) is improving over (A,B).

In the corresponding minimization problem one wants to find a cut of maximal weight.

Max Cut
Input: A graph $G = (V,E)$ and an edge-weight function $\omega : E \to \mathbb{N}$.
Output: A partition (A,B) of G such that $E_G(A,B)$ has maximum total weight.

A *local search problem* $(L, \mathcal{N})$ consists of

- an optimization problem L and
- a *neighborhood structure* $\mathcal{N}$ *for* L that maps for each instance I of L, each valid solution S of I to a set $\mathcal{N}(I,S) \subseteq \mathcal{S}_L(I)$ of valid solutions for I, the *neighbors of* S *with respect to* $\mathcal{N}$.

The goal of $(L, \mathcal{N})$ is to find for a given instance I of L a *locally optimal solution* S *with respect to* $\mathcal{N}$, that is, a feasible solution for I such that no solution in $\mathcal{N}(I,S)$ is better than S. We may write a local search problem $(L, \mathcal{N})$ as $L/\mathcal{N}$. An example for a local search problem is Weighted Independent Set/k-swap, where the neighbors of an independent set S are the valid k-swap neighbors of S.

A local search problem $(L, \mathcal{N})$ *is in the complexity class* PLS, if

- there is an algorithm which computes in polynomial time some feasible solution S for a given instance I of L and
- there is an algorithm which in polynomial time determines whether a given solution S is locally optimal with respect to $\mathcal{N}$ for an instance I of L and, if this is not the case, outputs a better neighbor for S.

Note that for each constant k, Weighted Independent Set/k-swap is contained in PLS.

Let $(L_1, \mathcal{N}_1)$ and $(L_2, \mathcal{N}_2)$ be local search problems. We say that $(L_1, \mathcal{N}_1)$ *is* PLS*-reducible to* $(L_2, \mathcal{N}_2)$ if for each instance I_1 of L_1, one can compute in polynomial time an instance I_2 of L_2 and a *solution mapper* f, that is, a polynomial-time computable function f that maps each solution S_2 of I_2 to a solution $f(S_2)$ of I_1 such that if S_2 is locally optimal for I_2 with respect to $\mathcal{N}_2$, then $f(S_2)$ is locally optimal for I_1 with respect to $\mathcal{N}_1$. A local search problem $(L, \mathcal{N})$ is PLS*-hard* if for each local search problem $(L', \mathcal{N}')$ of PLS, there is a PLS-reduction from $(L', \mathcal{N}')$ to $(L, \mathcal{N})$. Due to the transitivity of PLS-reductions, this can be done by proving a PLS-reduction from any PLS-hard local search problem $(L', \mathcal{N}')$. Moreover, $(L, \mathcal{N})$ is PLS*-complete* if $(L, \mathcal{N})$ is contained in PLS and PLS-hard. An example for a PLS-complete local search problem is Max Cut/flip, where the neighbors of a partition (A, B) are the improving partitions of (A, B) one can obtain by flipping some vertex of G [13]. This is the case even on graphs of degree at most 5 [4].

3 Hardness of Finding 7-optimal Independent Sets

To obtain the PLS-completeness of Weighted Independent Set/3-swap on graphs of constant maximum degree, we first show PLS-completeness for Weighted Independent Set/7-swap and use the obtained graph as starting point in a subsequent reduction to Weighted Independent Set/3-swap.

▶ **Theorem 3.1.** *Weighted Independent Set/7-swap is* PLS*-complete on graphs of maximum degree 6.*

As mentioned above, Weighted Independent Set/k-swap is contained in PLS for each constant value of k. Hence, we only have to show that Weighted Independent Set/7-swap is PLS-hard.

Construction. We present a PLS-reduction from Max Cut/flip to Weighted Independent Set/7-swap. Let $I = (G = (V, E), \omega)$ be an instance of Max Cut/flip where G has a maximum degree of five. For these instances Max Cut/flip is known to be PLS-complete [4]. We describe how to obtain in polynomial time an instance $I' = (G' = (V', E'), \omega')$ of Weighted Independent Set/7-swap and a polynomial-time computable solution-mapper f for I and I' such that if an independent set S is 7-optimal in I', then $f(S)$ is flip-optimal in I.

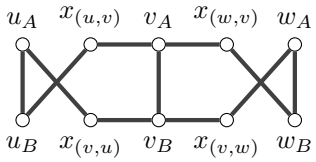

Figure 1 An example for the vertices and edges added to G' for a vertex $v \in V$ with two neighbors u and w in G in the reduction from Max Cut/flip to Weighted Independent Set/7-swap.

We start with an empty graph G' and add, for each vertex $v \in V$, two new adjacent vertices v_A and v_B to G'. Next, for each edge $\{u, v\} \in E$, we add two new vertices $x_{(u,v)}$ and $x_{(v,u)}$ to G' and make $x_{(u,v)}$ adjacent to u_B and v_A and $x_{(v,u)}$ adjacent to u_A and v_B. This completes the construction of G'. Figure 1 shows an example for the vertices and edges added to G' for a vertex $v \in V$ with two neighbors u and w in G. Note that G' has a maximum degree of six. In the following, let $V_A := \{v_A \mid v \in V\}$ and $V_B := \{v_B \mid v \in V\}$.

Next, we define the weight function $\omega' : V' \to \mathbb{N}$. Let $Z := \sum_{e \in E} \omega(e)$ denote the total weight of all edges. We set for each vertex $v \in V$, $\omega'(v_A) := \omega'(v_B) := 16 \cdot Z$ and for each edge $\{u, v\} \in E$, we set $\omega'(x_{(u,v)}) := \omega'(x_{(v,u)}) := 8 \cdot \omega(\{u, v\})$. In principle, the factors of 8 can be omitted but they will come in handy later when we present the PLS-reduction to Weighted Independent Set/3-swap.

This completes the construction of I'. It remains to define the solution-mapper f. For an independent set S, we define $f(S)$ to be the partition (A, B) of G where $A := \{v \in V \mid v_A \in S\}$ and $B := V \setminus A$. Recall that for vertex $v \in V$, the vertices v_A and v_B are adjacent in G'.

Correctness. To show the correctness of the reduction, we first analyze the structure of 7-optimal independent sets in G'. To this end, we define a notion of *nice* independent sets in G' and show that all 7-optimal independent sets in G' are nice. Recall that if some vertex v of V is contained in A for $f(S) = (A, B)$, then v_B is not contained in S.

▶ **Definition 3.2.** *Let S be an independent set in G' and let $A := \{v \in V \mid v_A \in S\}$ and let $B := \{v \in V \mid v_B \in S\}$. We call S* nice *if (A, B) is a partition of G and for each edge $\{u, v\} \in E$ with $(u, v) \in A \times B$, $x_{(u,v)} \in S$.*

▶ **Lemma 3.3 (*).** *If an independent set S in G is not nice, then S is not 7-optimal in G'.*

Hence, we only have to consider nice independent sets in G' when considering 7-optimal independent sets in G'. Now, to prove Theorem 3.1 it remains to show the following.

▶ **Lemma 3.4.** *Let S be a nice independent set in G'. If $f(S)$ is not flip-optimal in G, then S is not 7-optimal in G'.*

Proof. Let $(A, B) := f(S)$. By definition of f and the fact that S is nice, $A = \{v \in V \mid v_A \in S\}$ and $B = \{v \in V \mid v_B \in S\}$. Suppose that (A, B) is not flip-optimal in G. Then, there is some vertex $v \in V$ where the total weight of the edges that are incident with v and that are in the cut $E_G(A, B)$ is less than the total weight of the edges that are incident with v and that are not in the cut $E_G(A, B)$. That is, either a) $v \in A$ and $\omega(E_G(\{v\}, A)) > \omega(E_G(\{v\}, B))$ or b) $v \in B$ and $\omega(E_G(\{v\}, B)) > \omega(E_G(\{v\}, A))$.

Without loss of generality we may assume that $v \in A$ and $\omega(E_G(\{v\}, A)) > \omega(E_G(\{v\}, B))$. Let $X_A := N_G(v) \cap A$ denote the neighbors of v in A and let $X_B := N_G(v) \cap B$ denote the neighbors of v in B. Since S is nice, we know that $x_{(v,u)} \in S$ for each $u \in X_B$. Moreover,

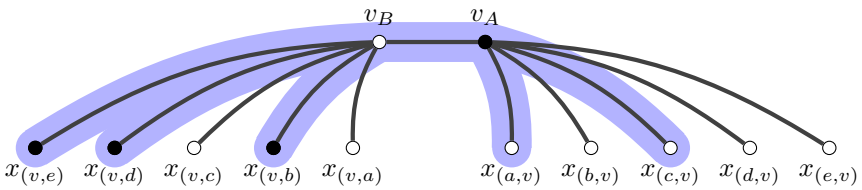

Figure 2 An example of an improving 7-swap W for a nice independent set S in G' that simulates the flip of a vertex $v \in V$ from A to B in G, where $N_G(v) = \{a, b, c, d, e\}$ and $N_G(v) \cap B = \{b, d, e\}$. The black vertices are the vertices of S and all vertices of W are highlighted by the blue shape.

for each $w \in X_A$, $x_{(v,w)} \notin S$ since $w_A \in S$ and $x_{(w,v)} \notin S$ since $v_A \in S$. We show that the swap $W := \{v_A, v_B\} \cup \{x_{(v,u)} \mid u \in X_B\} \cup \{x_{(w,v)} \mid w \in X_A\}$ is a valid improving 7-swap for S in G'. An example of the swap W is illustrated in Figure 2. First of all, note that W has size at most 7 since G has a maximum degree of 5 and thus $|X_A \cup X_B| \leq 5$. Moreover, by the fact that $\omega'(x_{(y,z)}) := \omega'(x_{(z,y)}) := 8 \cdot \omega(\{y, z\})$ for each edge $\{y, z\} \in E$,

$$\begin{aligned}\omega'(\{x_{(w,v)} \mid w \in X_A\}) &= 8 \cdot \omega(\{\{v, w\} \mid w \in X_A\}) \\ &> 8 \cdot \omega(\{\{v, u\} \mid u \in X_B\}) = \omega'(\{x_{(v,u)} \mid u \in X_B\}).\end{aligned}$$

Hence, W is improving, since $\omega'(v_A) = \omega'(v_B)$. It remains to show that W is valid. Since W removes all adjacent vertices of v_B from S, $S \oplus W$ does not contain any neighbor of v_B. Moreover, since $v_A \in W$ and for each $w \in X_A$, $w_A \in S \oplus W$ and thus $w_B \notin S \oplus W$, no vertex $x_{(w,v)}$ is adjacent to any vertex in $S \oplus W$. As a consequence, W is valid and thus S is not 7-optimal. ◀

4 Hardness of Finding 3-optimal Independent Sets

The main result of this section is the following.

▶ **Theorem 4.1.** *Weighted Independent Set/3-swap is* PLS*-complete on graphs of constant maximum degree.*

Construction. To show this, we extend the graph G' by additional gadgets to simulate 7-swaps by a sequence of 3-swaps. We describe how to obtain in polynomial time an instance $I'' = (G'' = (V'', E''), \omega'')$ of Weighted Independent Set/3-swap and a polynomial-time computable solution-mapper f for I and I'' such that if an independent set S is 3-optimal in I'', then $f(S)$ is flip-optimal in I. As described above, we extend the graph G' and the weight function ω' of the instance $I' = (G' = (V', E'), \omega')$ of Weighted Independent Set/7-swap described above. As above, let $V_A := \{v_A \mid v \in V\}$ and let $V_B := \{v_B \mid v \in V\}$. Moreover, we set for each $v \in V$, $X_v := \{x_{(v,w)}, x_{(w,v)} \mid w \in N(v)\}$, that is, X_v is the set of vertices in G' that were introduced for the incident edges of v in G. We write $N_G(v)$ and $N_G[v]$ when considering the neighborhood of a vertex v of V in G and $N(v)$ or $N[v]$ when considering the neighborhood of a vertex v of V'' in G''.

Initially, we add edges such that for each vertex $v \in V$, $u \in N_G(v)$, and $w \in N_G(V)$, the vertices $x_{(u,v)}$ and $x_{(v,w)}$ are adjacent in G''. Note that this includes edges between $x_{(u,v)}$ and $x_{(v,u)}$ in G'' for each edge $\{u, v\} \in E$. The idea is that in an independent set S in G'', for each vertex $v \in V$, there is no vertex $x_{(u,v)} \in S$ if S already contains a vertex $x_{(v,w)}$. Hence, for each vertex v of G, at most one of v_A and v_B has neighbors in $X_v \cap S$.

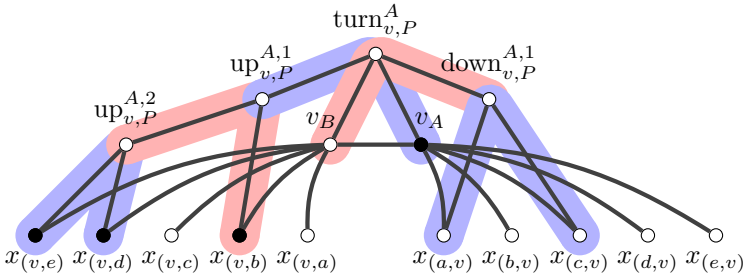

Figure 3 A sequence $(W_1, W_2, W_3, W_4, W_5)$ of improving 3-swaps (from left to right highlighted alternating by either a blue or a red shape) for a nice independent set S in G'' that simulates the flip of a vertex $v \in V$ from A to B in G, where $N_G(v) = \{a, b, c, d, e\}$ and $P = N_G(v) \cap B = \{b, d, e\}$ with $P(1) = b$, $P(2) = d$, and $P(3) = e$. The black vertices are the vertices of S. Note that not all edges of this subgraph are shown but only the important ones for the sequence of improving 3-swaps.

Next, we add gadgets to allow us to simulate 7-swaps in G' by a sequence of 3-swaps in G''. To this end, we first compute for each vertex $v \in V$ the collection $\mathcal{P}_v$ of subsets $P \subseteq N_G(v)$ fulfilling $\omega(E_G(\{v\}, P)) < \omega(E_G(\{v\}, N_G(v) \setminus P))$. Intuitively, if in a partition (A, B) of G, flipping a vertex v from A to B is improving, then the set $N_G(v) \cap B$ is contained in $\mathcal{P}_v$ and vice versa. Note that $\emptyset \in \mathcal{P}_v$ and $N_G(v) \notin \mathcal{P}_v$ for each $v \in V$. Next, we add, for each $P \in \mathcal{P}_v$ with $P \neq \emptyset$, a set of $|N_G(v)| - 1$ new vertices to G''. Let $Q := N_G(v) \setminus P$. Now, fix an ordering on both P and Q and let $P(i)$ denote the ith element of P and let $Q(j)$ denote the jth element of Q where $i \in [1, |P|]$ and $j \in [1, |Q|]$.

These vertices are of three types: up, turn, and down. To simulate a flip of v from A to B, we have to remove the neighbors of v_B from S, add v_B to S and add the vertices representing edges of $(A \cap N_G(v)) \times \{v\}$ to S. Intuitively, the up-vertices allow us – with a sequence of improving 3-swaps – to remove all neighbors of v_B from S except v_A and the up-vertex of highest level. Afterwards, the turn-vertex allows us – with two improving 3-swaps – to remove v_A from S and add v_B and the down-vertex of highest level to S. Finally, the down-vertices allow us – with a sequence of improving 3-swaps – to add the vertices representing edges of $(A \cap N_G(v)) \times \{v\}$ to S such that at the end, none of these auxiliary vertices remains in S. In total, this allows us to simulate improving 7-swaps in I' and thus improving flips in I by a sequence of improving 3-swaps in I''. An example for a sequence of improving 3-swaps in G'' to simulate a flip in G is illustrated in Figure 3. This figure shows only the edges that are important for the sequence of improving 3-swaps and not all edges of this subgraph.

First, we add for each $i \in [1, |P| - 1]$, a new vertex $\text{up}^{A,i}_{v,P}$ to G such that the neighborhood of $\text{up}^{A,i}_{v,P}$ is exactly X_v minus the vertices $\{x_{(v,P(j))} \mid j < i\}$. Hence, $\text{up}^{A,1}_{v,P}$ is adjacent to all vertices of X_v. Furthermore, we add an edge between $\text{up}^{A,i}_{v,P}$ and each vertex of $\{w_A \mid w \in P\} \cup \{w_B \mid w \in Q\}$ and an edge between $\text{up}^{A,i}_{v,P}$ and v_B. Moreover, we set

$$\omega''(\text{up}^{A,i}_{v,P}) := 4 - i + 8 \cdot \sum_{j=i}^{|P|} \omega(\{v, P(j)\}) = 4 - i + \sum_{j=i}^{|P|} \omega''(x_{(v,P(j))}).$$

Intuitively, one can obtain an improving 3-neighbor for S in G as follows:

1. if $w_A \notin S$ for any neighbor $w \in N_G(v)$ in P and $u_B \notin S$ for any neighbor $u \in N_G(v)$ in P, then we remove $x_{(v,P(|P|-1))}$ and $x_{(v,P(|P|))}$ and add $\text{up}^{A,|P|-1}_{v,P}$, and
2. if $\text{up}^{A,j}_{v,P} \in S$, where $j \in [2, |P| - 1]$, then we remove $\text{up}^{A,j}_{v,P}$ and $x_{(v,P(j))}$ and add $\text{up}^{A,j-1}_{v,P}$.

Hence, during the simulation of an improving flip of a vertex v in a partition (A, B) from A to B in G, we can replace all vertices of X_v by the vertex $\text{up}_{v,P}^{A,1}$ with a sequence of improving 3-swaps, where $P = N_G(v) \cap B$.

Second, we add a vertex $\text{turn}_{v,P}^{A}$ to G'' which is adjacent to all vertices of X_v, $\{v_A, v_B\}$, and to the vertices of $\{w_A \mid w \in P\} \cup \{w_B \mid w \in Q\}$. Recall that $\omega''(v_A) = \omega''(v_B) = 16 \cdot Z$ where $Z = \sum_{e \in E} \omega(e)$. We set

$$\omega''(\text{turn}_{v,P}^{A}) := 4 + \omega''(v_A) + 8 \cdot \sum_{w \in P} \omega(\{v, w\}) = 4 + 16 \cdot Z + \sum_{w \in P} \omega''(x_{(v,w)}).$$

Intuitively, one can obtain an improving 3-neighbor for S in G if $\text{up}_{v,P}^{A,1}$ is contained in S, by removing both $\text{up}_{v,P}^{A,1}$ and v_A from S and adding $\text{turn}_{v,P}^{A}$ to S.

Third, we add, similar to $\text{up}_{v,P}^{A,i}$, for each $i \in [1, |Q| - 1]$ a new vertex $\text{down}_{v,P}^{A,i}$ to G such that the neighborhood of $\text{down}_{v,P}^{A,i}$ is exactly $X_v \setminus \{x_{(Q(j),v)} \mid j < i\}$. Hence, $\text{down}_{v,P}^{A,1}$ is adjacent to all vertices of X_v. Furthermore, we add an edge between $\text{down}_{v,P}^{A,i}$ and each vertex of $\{w_A \mid w \in P\} \cup \{w_B \mid w \in Q\}$ and we also an edge between $\text{down}_{v,P}^{A,i}$ and v_A. Moreover, we set

$$\omega''(\text{down}_{v,P}^{A,i}) := i - 4 + 8 \cdot \sum_{j=i}^{|Q|} \omega(\{v, Q(j)\}) = i - 4 + \sum_{j=i}^{|Q|} \omega''(x_{(v,Q(j))}).$$

Note that $\omega''(\text{down}_{v,P}^{A,i}) > 0$ since $Q \neq \emptyset$ and the images of ω are positive numbers. Intuitively, one can obtain an improving 3-neighbor for S in G as follows:

1. if $\text{turn}_{v,P}^{A} \in S$, then we remove $\text{turn}_{v,P}^{A}$ and add $\text{down}_{v,P}^{A,1}$ and v_B,
2. if $\text{down}_{v,P}^{A,j} \in S$ for some $j < |Q| - 1$, then we remove $\text{down}_{v,P}^{A,j}$ and add $\text{down}_{v,P}^{A,j+1}$ and $x_{(Q(j),v)}$, and
3. if $\text{down}_{v,P}^{A,|Q|-1} \in S$, then we remove $\text{down}_{v,P}^{A,|Q|-1}$ and add $x_{(Q(|Q|-1),v)}$ and $x_{(Q(|Q|),v)}$.

With the current graph, it is possible to simulate a flip of a vertex v from A to B. To also simulate a flip of vertex v from B to A, we add symmetric vertices to G'', that is, for each vertex $\text{up}_{v,P}^{A,j}$, we add a vertex $\text{up}_{v,P}^{B,j}$, for each vertex $\text{turn}_{v,P}^{A}$, we add a vertex $\text{turn}_{v,P}^{B}$, and for each vertex $\text{down}_{v,P}^{A,j}$, we add a vertex $\text{down}_{v,P}^{B,j}$.

The formal definition of these symmetric vertices is deferred to the full version.

Note that we did not add any vertices for $P = \emptyset \in \mathcal{P}_v$ to the graph G''. In the correctness proof, we show that a single improving 3-swap for S is sufficient to simulate the flip of a vertex v of G if no edge incident with v is currently in the cut.

For each vertex $v \in V$, let V_v denote the set of the additional vertices associated with v:

$$V_v := \bigcup_{P \in \mathcal{P}_v, P \neq \emptyset, C \in \{A,B\}} (\{\text{up}_{v,P}^{C,i} \mid i < |P|\} \cup \{\text{turn}_{v,P}^{C}\} \cup \{\text{down}_{v,P}^{C,i} \mid i < |N_G(v) \setminus P|\}).$$

To complete the construction, for each $v \in V$, we make the set $\bigcup_{w \in N_G[v]} V_w$ a clique in G''.

▶ **Lemma 4.2 (*).** *The graph G'' has maximum degree at most 3140.*

It remains to define the solution-mapper f. Analogously to the solution-mapper of the presented PLS-reduction for Weighted Independent Set/7-swap, for an independent set S, we define $f(S)$ to be the partition (A, B) of G where $A := \{v \in V \mid v_A \in S\}$ and $B := V \setminus A$.

Correctness. To show the correctness of the PLS-reduction, we first analyze the structure of 3-optimal independent sets in G''. To this end, we show in the auxiliary Lemmas 4.3 – 4.5 that we can assume that each 3-optimal independent set in G'' contains for each $v \in V$ either the vertex v_A or the vertex v_B. Recall that for vertex $v \in V$, the vertices v_A and v_B are adjacent in G''. As a consequence, this then implies that for $f(S) = (A, B)$, B is exactly the set $\{v \in V \mid v_B \in S\}$. Finally, in Lemma 4.6, we then show that such an independent set S is not 3-optimal in I'' if $f(S)$ is not flip-optimal in G.

▶ **Lemma 4.3.** *Let S be an independent set in G''. If S contains a vertex $\mathrm{up}^{C,i}_{v,P}$ for $C \in \{A, B\}$, then S is not 3-optimal.*

Proof. First, we show the statement for $i = 1$. By construction, the closed neighborhood $N[\mathrm{turn}^{C}_{v,P}]$ is exactly $N[\mathrm{up}^{C,1}_{v,P}] \cup \{v_C\}$ and $\omega''(\mathrm{turn}^{C}_{v,P}) = 1 + \omega''(\mathrm{up}^{C,1}_{v,P}) + \omega''(v_C)$. Hence, S is not 3-optimal in G'' since $S' := (S \cup \{\mathrm{turn}^{C}_{v,P}\}) \setminus \{\mathrm{up}^{C,1}_{v,P}, v_C\}$ is an improving 3-neighbor of S in G''. This holds even if v_C is not in S.

Second, we show the statement for $i > 1$. Let $r := x_{(v,P(i-1))}$ if $C = A$ and $r := x_{(P(i-1),v)}$ if $C = B$. Note that by construction, the closed neighborhood $N[\mathrm{up}^{C,i-1}_{v,P}]$ is exactly $N[\mathrm{up}^{C,i}_{v,P}] \cup \{r\}$ and $\omega''(\mathrm{up}^{C,i-1}_{v,P}) = 1 + \omega''(\mathrm{up}^{C,i}_{v,P}) + \omega''(r)$. Hence, if an independent set S contains $\mathrm{up}^{C,i}_{v,P}$ with $i > 1$, then S is not 3-optimal in G'' since $(S \cup \{\mathrm{up}^{C,i-1}_{v,P}\}) \setminus \{\mathrm{up}^{C,i}_{v,P}, r\}$ is an improving 3-neighbor of S in G''. This holds even if r is not in S. ◀

▶ **Lemma 4.4 (*).** *Let S be an independent set in G''. If there is a vertex $v \in V$ such that S avoids v_A, v_B, and $T_v := \{\mathrm{turn}^{A}_{v,P}, \mathrm{turn}^{B}_{v,P} \mid P \in \mathcal{P}_v, P \neq \emptyset\}$, then S is not 3-optimal.*

Hence, we can assume that each 3-optimal independent set S in G'' contains for each vertex $v \in V$ either v_A, v_B, or exactly one vertex of $T_v := \{\mathrm{turn}^{A}_{v,P}, \mathrm{turn}^{B}_{v,P} \mid P \in \mathcal{P}_v, P \neq \emptyset\}$, and no vertex of $\{\mathrm{up}^{A,i}_{v,P}, \mathrm{up}^{B,i}_{v,P} \mid P \in \mathcal{P}_v, P \neq \emptyset, i < |P|\}$. In the following, we call an independent set S of G'' *nice* if $S \subseteq V' = V_A \cup V_B \cup \bigcup_{v \in V} X_v$ and if S is a nice independent set for the instance I' of Weighted Independent Set/7-swap. That is, if for $A := \{v \in V \mid v_A \in S\}$ and $B := \{v \in V \mid v_B \in S\}$, (A, B) is a partition of G and for each edge $\{u, v\} \in E$ with $(u, v) \in A \times B$, the vertex $x_{(u,v)}$ is contained in S. Next, we show similar to Lemma 3.4, that an independent set S in G'' is not 3-optimal if S is not nice.

▶ **Lemma 4.5.** *Let S be an independent set in G''. If S is not nice, then S is not 3-optimal.*

Proof. Let S be an independent set in G'' which is not nice. Due to Lemma 4.3, we know that S is not 3-optimal if some vertex $\mathrm{up}^{C,i}_{v,P}$ for $C \in A, B$ is contained in S. Hence, in the following, we assume that none of these vertices is contained in S. Moreover, due to Lemma 4.4, we also know that S is not 3-optimal if there is some vertex $v \in V$ such that S does not contain v_A, v_B, and no vertex of $\{\mathrm{turn}^{A}_{v,P}, \mathrm{turn}^{B}_{v,P} \mid P \in \mathcal{P}_v, P \neq \emptyset\}$. Hence, in the following we further assume that S contains one of these vertices for some $v \in V$.

In a first step, we show that if for some $v \in V$, S contains some vertex of $\{\mathrm{turn}^{A}_{v,P}, \mathrm{turn}^{B}_{v,P} \mid P \in \mathcal{P}_v, P \neq \emptyset\}$, then S is not 3-optimal. Assume without loss of generality that $\mathrm{turn}^{A}_{v,P} \in S$ and let $Q := N_G(v) \setminus P$. Recall that $\bigcup_{w \in N_G[v]} V_w$ is a clique in G'' which implies that for each $w \in N_G(v)$, S contains no vertex of $\{\mathrm{turn}^{A}_{w,P'}, \mathrm{turn}^{B}_{w,P'} \mid P' \in \mathcal{P}_w, P' \neq \emptyset\} \subseteq V_w$. Hence, due to Lemma 4.4, to for each $w \in N_G(v)$, S contains either w_A or w_B. Moreover, $\mathrm{turn}^{A}_{v,P}$ is adjacent to all vertices of X_v, all vertices of $\{w_A \mid w \in P\}$, and all vertices of $\{w_B \mid w \in Q\}$. As a consequence, for each $w \in P$, $w_B \in S$, and $w_A \notin S$, and for each $u \in Q$, $u_A \in S$ and $u_B \notin S$. Furthermore, S contains no other neighbor of v_A or v_B. Recall that $P \neq N_G(v)$ which implies $Q = \emptyset$. In the following, it suffices to distinguish between $|Q| = 1$ and $|Q| > 1$.

First, suppose that $|Q| = 1$ and let w denote the unique vertex of Q. We show that $S' := (S \cup \{v_B, x_{(w,v)}\}) \setminus \{\text{turn}^A_{v,P}\}$ is an improving 3-neighbor of S in G''. By construction, v_B and $x_{(w,v)}$ are non-adjacent in G'' and

$$\omega''(v_B) + \omega''(x_{(w,v)}) = 16 \cdot Z + 8 \cdot \omega(\{v,w\}) = 16 \cdot Z + 8 \cdot \sum_{u \in Q} \omega(\{v,u\})$$
$$> 16 \cdot Z + 4 + 8 \cdot \sum_{u \in P} \omega(\{v,u\}) = \omega''(\text{turn}^A_{v,P})$$

since the images of ω are positive numbers and $P \in \mathcal{P}_v$ which implies $\sum_{u \in Q} \omega(\{v,u\}) > \sum_{u \in P} \omega(\{v,u\})$. Hence, it remains to show that S is an independent set. Since by definition, $N[v_B] \subseteq N[\text{turn}^A_{v,P}]$, we only have to show that there is no other neighbor of $x_{(w,v)}$ in S besides $\text{turn}^A_{v,P}$. By construction, the neighborhood of $x_{(w,v)}$ in G'' is a subset of $\{v_A, w_B\} \cup V_v \cup V_w \cup X_v \cup X_w$. Now, $\text{turn}^A_{v,P}$ is adjacent to all vertices of this set except for some vertices of X_w. Hence, we only have to consider the neighbors of $x_{(w,v)}$ in X_w. By construction, these are the vertices $x_{(u,w)}$ where $u \in N_G(w)$. Since w_A is contained in S and each of these vertices $x_{(u,w)}$ is adjacent to w_A in G'', $x_{(w,v)}$ has no neighbor in S besides $\text{turn}^A_{v,P}$. As a consequence, S' is an improving 3-neighbor of S.

Second, suppose that $|Q| > 1$. Then, the vertex $\text{down}^{A,1}_{v,P}$ exists. We show that $S' := (S \cup \{v_B, \text{down}^{A,1}_{v,P}\}) \setminus \{\text{turn}^A_{v,P}\}$ is an improving 3-neighbor of S in G''. By construction, v_B and $\text{down}^{A,1}_{v,P}$ are non-adjacent in G'' and

$$\omega''(v_B) + \omega''(\text{down}^{A,1}_{v,P}) = 16 \cdot Z - 3 + 8 \cdot \sum_{w \in Q} \omega(\{v,w\})$$
$$> 16 \cdot Z + 4 + 8 \cdot \sum_{u \in P} \omega(\{v,u\}) = \omega''(\text{turn}^A_{v,P})$$

since the images of ω are positive numbers and $P \in \mathcal{P}_v$ which implies $\sum_{u \in Q} \omega(\{v,u\}) > \sum_{u \in P} \omega(\{v,u\})$. Hence, it remains to show that S is an independent set. By definition, $N[v_B] \subseteq N[\text{turn}^A_{v,P}]$ and $N[\text{down}^{A,1}_{v,P}] \subseteq N[\text{turn}^A_{v,P}]$, which implies that $\text{turn}^A_{v,P}$ is the unique neighbor of both v_B and $\text{turn}^A_{v,P}$ in S. As a consequence, S' is an independent set and thus an improving 3-neighbor of S. Hence, in the following, we can also assume that for each $v \in V$, S contains either v_A or v_B.

Next, we show that if for some $v \in V$, there is some vertex $r \in V_v$ contained in S, then S is not 3-optimal. Note that we already showed that this is the case if r is $\text{up}^{A,i}_{v,P}$, $\text{up}^{B,i}_{v,P}$, $\text{turn}^A_{v,P}$, or $\text{turn}^B_{v,P}$. Hence, it remains to show the claim for r being $\text{down}^{A,i}_{v,P}$ or $\text{down}^{B,i}_{v,P}$. Assume without loss of generality that there is a nonempty set $P \in \mathcal{P}_v$ and some $i < |Q|$ with $Q := N_G(v) \setminus P$ such that $\text{down}^{A,i}_{v,P}$ is contained in S. The proof of the fact that in this case S is not 3-optimal is deferred to the full version.

Summarizing, we can assume in the following that $S \subseteq V' = V_A \cup V_B \cup \bigcup_{v \in V} X_v$ and that (A, B) is a partition of G, where $A := \{v \in V \mid v_A \in S\}$ and $B := \{v \in V \mid v_B \in S\}$. Note that for each edge $\{v,w\} \in E$ with $(v,w) \notin A \times B$, the vertex $x_{(v,w)}$ is not contained in S since $v_B \in S$ or $w_A \in S$, and both these vertices are adjacent to $x_{(v,w)}$. Hence, it remains to show that if there is some edge $\{v,w\} \in E$ with $(v,w) \in A \times B$ such that $x_{(v,w)}$ is not contained in S, then S is not 3-optimal in G''. To this end, we show that $S \cup \{x_{(v,w)}\}$ is an independent set in G''. By construction, $x_{(v,w)}$ is adjacent to v_B, w_A, some vertices of $V_v \cup V_w$, and the vertices $\{x_{(u,v)} \mid u \in N_G(v)\} \cup \{x_{(w,u)} \mid u \in N_G(w)\}$. Recall that S contains no vertex of $V_v \cup V_w$. Moreover, since $(v,w) \in A \times B$, v_A and w_B are contained

in S which implies that v_B and w_A are not contained in S. Furthermore, since all vertices of $\{x_{(u,v)} \mid u \in N_G(v)\}$ are adjacent to v_A, all vertices of $\{x_{(w,u)} \mid u \in N_G(w)\}$ are adjacent to w_B, and v_A and w_B are contained in S, $S \cup \{x_{(v,w)}\}$ is an independent set in G''.

We conclude that if S is not nice in G'', then S is not 3-optimal. ◀

Now the following implies the correctness of the PLS-reduction.

▶ **Lemma 4.6 (*).** *Let S be a nice independent set in G'' and let $A := \{v \in V \mid v_A \in S\}$ and $B := \{v \in V \mid v_B \in S\}$. If (A, B) is not flip-optimal in G, then S is not 3-optimal.*

Next, we obtain similar results for Weighted Dominating Set.

▶ **Theorem 4.7 (*).** *Let $k \geq 3$. There is a* PLS*-reduction from Weighted Independent Set/k-swap to Weighted Dominating Set/k-swap where the maximum degree of the output graph is at most two times the maximum degree of the input graph.*

Hence, we obtain the following due to Theorem 4.7, Theorem 3.1, and Theorem 4.1.

▶ **Corollary 4.8.** *Weighted Dominating Set/7-swap is* PLS*-complete on graphs of maximum degree at most* 12 *and Weighted Dominating Set/*3*-swap is* PLS*-complete on graphs of constant maximum degree.*

5 Finding Locally Optimal Solutions for Restricted 3-Swaps

We now show that we can find locally optimal solutions in polynomial time for Weighted Independent Set and Weighted Dominating Set if we restrict the allowed 3-swaps as follows: we either only allow swaps that add at most one vertex to the current solution or we allow only swaps that remove at most one vertex from the solution. These are exactly the $(1, 2)$-swaps and $(2, 1)$-swaps, respectively, since a 3-swap that removes three vertices from the current solution or that adds three vertices to the solution is either not improving, or the solution is not 1-optimal.

Recall that a $(k^{\text{in}}, k^{\text{out}})$-swap W for a set S is a k-swap with $k^{\text{in}} + k^{\text{out}} \leq k$ such that $|W \setminus S| \leq k^{\text{in}}$ and $|W \cap S| \leq k^{\text{out}}$.

First, we show that we can compute in $\mathcal{O}(n \cdot \log(n) + m)$ time a $(1, 2)$-optimal independent set S. More precisely, we show that S is even $(1, k)$-optimal for every $k \in \mathbb{N}$.

▶ **Theorem 5.1 (*).** *One can compute in $\mathcal{O}(n \cdot \log(n) + m)$ time an independent set which is $(1, k)$-optimal for every $k \in \mathbb{N}$.*

Second, we show that we can also compute in $\mathcal{O}(n \cdot \log(n) + m)$ time a $(2, 1)$-optimal dominating set S. More precisely, we show that S is even $(k, 1)$-optimal for every $k \in \mathbb{N}$.

▶ **Theorem 5.2 (*).** *One can compute in $\mathcal{O}(n \cdot \log(n) + m)$ time a dominating set which is $(k, 1)$-optimal for every $k \in \mathbb{N}$.*

▶ **Theorem 5.3.** *Let L be a subset-weight maximization problem, let I be an instance of L, and let $k \in \mathcal{O}(1)$. One can compute in polynomial time a $(k, 1)$-optimal solution S for I.*

Proof. Recall that since L is a subset-weight maximization problem, L consists of functions U, f, g, and ω, where for each instance I of L, $U(I)$ is the universe of I, $f(I, S)$ checks if S is a solution for I, $g(I)$ computes some feasible solution for I, and $\omega(I, u)$ assigns a weight to each $u \in U(I)$. Moreover, the functions U, f, g, and ω are polynomial-time computable.

Let I be an instance of L and let $k \geq 0$. Note that we can compute some feasible solution $S_0 := g(I)$ in polynomial time. Since $U(I)$ can be computed in polynomial time, $U(I)$ has polynomial size. Let $n := |U(I)|$. Since f and ω can be computed in polynomial time, we can check for a given solution S in $n^{\mathcal{O}(k)}$ time if there is an improving $(k+1)$-swap W such that W is a $(k,1)$-swap for S by considering all subsets of size at most $k+1$ of $U(I)$. Note that this is polynomial time since k is a constant. Hence, we can determine whether a solution S is $(k,1)$-optimal and, if this is not the case, replace S by a $(k,1)$-neighbor, both in polynomial time. Let S_x be a $(k,1)$-optimal feasible solution in I which can be obtained by a sequence $(S_0, S_1, \ldots, S_x)$ of consecutive improving $(k,1)$-neighbors in I starting from $g(I) = S_0$. We show that $x \in \mathcal{O}(n^3)$, which implies that a $(k,1)$-optimal feasible solution for I can be computed in polynomial time. Let $\ell \in [1, x]$. To this end, we first show that S_ℓ is never smaller than $S_{\ell-1}$. Assume towards a contradiction that $|S_\ell| < |S_{\ell-1}|$. Then there is some $u \in S_{\ell-1}$ such that $S_\ell = S_{\ell-1} \setminus \{u\}$ since S_ℓ is a $(k,1)$-neighbor of $S_{\ell-1}$ in I. Since the weight of S_ℓ is defined as the sum of weights of the elements of S_ℓ and each element $u \in U(I)$ has a positive weight $\omega(I,u)$, the total weight of S_ℓ is not larger than the total weight of $S_{\ell-1}$, a contradiction. As a consequence, there are at most $y \leq |U(I)|$ distinct indices $\ell_1, \ldots, \ell_y$ such that S_{ℓ_i} is larger than $S_{\ell_i - 1}$. To prove that $x \in \mathcal{O}(n^3)$, we now show that for each $\ell \in [1, x - n^2]$, there is some $z \in [\ell, \ell + n^2]$ where S_z is larger than $S_{\ell-1}$. In other words, after at most n^2 improving swaps that do not increase the size of the solution, the next improving swap increases the size of the solution.

Let $\ell \in [1, x - U(I)^2]$, let $\sigma = (u_1, \ldots, u_n)$ be a fixed increasing order of the elements of $U(I)$ according to their weight, and let for each $j \in [\ell, \ell + n^2]$, $q_j := \sum_{u_i \in S_j} i$ denote the sum of the indices of S_j in σ. Note that if S_j and S_{j-1} have same size, $S_j = (S_{j-1} \cup \{u_2\}) \setminus \{u_1\}$ for two elements $u_1 \in U(I)$ and $u_2 \in U(I)$ where the weight of u_2 is larger than the weight of u_1. Hence, q_j is larger than q_{j-1} if S_j and S_{j-1} have same size. Since the value q_j can never exceed n^2, there is some $z \in [\ell, \ell + n^2]$ where the size of S_z is larger than the size of $S_{\ell-1}$. We conclude that $x \in \mathcal{O}(n^3)$ which implies that we can compute in polynomial time a feasible solution S of I such that S is $(k,1)$-optimal in I. ◀

▶ **Corollary 5.4 (*).** *Let L be a subset-weight minimization problem, let I be an instance of L and let $k \in \mathcal{O}(1)$. One can compute in polynomial time a $(1,k)$-optimal solution S for I.*

By the fact that for a maximization problem there is no improving swap that only removes elements from the solution and for a minimization problem there is no improving swap that only adds elements to the solution, Theorem 5.3 and Corollary 5.4 imply that one can find for each subset weight optimization problem L, a 2-optimal solution in polynomial time.

Since Weighted Independent Set is a subset-weight maximization problem and since Weighted Dominating Set is a subset-weight minimization problem, we conclude the following.

▶ **Corollary 5.5.** *Let k be a constant. One can compute in polynomial time a $(k,1)$-optimal independent set and one can compute in polynomial time a $(1,k)$-optimal dominating set.*

Consequently, if we allow only $(1,2)$-swaps or only $(2,1)$-swaps, we can find locally optimal solutions for Weighted Independent Set and Weighted Dominating Set in polynomial time. In contrast, if we allow $(1,2)$-swaps and $(2,1)$-swaps, then we allow all 3-swaps and both problems become PLS-complete even on graphs of constant maximum degree.

6 Conclusion

From a theoretical point of view, the most important open topic is to determine the precise degree bounds that separate the polynomial-time solvable and PLS-complete cases for Weighted Independent Set and Weighted Dominating Set for k-swaps with small constant k-values. From a practical point of view, our findings motivate, for both problems, the use of gap-variants of local search where we are searching for solutions that improve the objective value by at least some threshold d, the hope being that this decreases the number of necessary iterations while preserving solution quality.

References

1 Diogo Vieira Andrade, Mauricio G. C. Resende, and Renato Fonseca F. Werneck. Fast local search for the maximum independent set problem. *Journal of Heuristics*, 18(4):525–547, 2012.

2 Shaowei Cai, Kaile Su, Chuan Luo, and Abdul Sattar. NuMVC: An efficient local search algorithm for minimum vertex cover. *Journal of Artificial Intelligence Research*, 46:687–716, 2013.

3 Jakob Dahlum, Sebastian Lamm, Peter Sanders, Christian Schulz, Darren Strash, and Renato F. Werneck. Accelerating local search for the maximum independent set problem. In *Proceedings of the 15th International Symposium on Experimental Algorithms (SEA '16)*, volume 9685 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2016.

4 Robert Elsässer and Tobias Tscheuschner. Settling the complexity of local max-cut (almost) completely. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP '11)*, volume 6755 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2011. doi:10.1007/978-3-642-22006-7_15.

5 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

6 Maximilian Katzmann and Christian Komusiewicz. Systematic exploration of larger local search neighborhoods for the minimum vertex cover problem. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, (AAAI '17)*, pages 846–852. AAAI Press, 2017.

7 Brian W. Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, 1970.

8 Hartmut Klauck. On the hardness of global and local approximation. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT '96)*, volume 1097 of *Lecture Notes in Computer Science*, pages 88–99. Springer, 1996.

9 Sebastian Lamm, Peter Sanders, Christian Schulz, Darren Strash, and Renato F. Werneck. Finding near-optimal independent sets at scale. *Journal of Heuristics*, 23(4):207–229, 2017.

10 Ruizhi Li, Shuli Hu, Shaowei Cai, Jian Gao, Yiyuan Wang, and Minghao Yin. NuMWVC: A novel local search for minimum weighted vertex cover problem. *Journal of the Operational Research Society*, 71(9):1498–1509, 2020.

11 Wil Michiels, Emile H. L. Aarts, and Jan H. M. Korst. *Theoretical aspects of local search*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2007.

12 Svatopluk Poljak. Integer linear programs and local search for max-cut. *SIAM Journal on Computing*, 24(4):822–839, 1995.

13 Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM Journal on Computing*, 20(1):56–87, 1991.

14 Shinichi Shimozono. Finding optimal subgraphs by local search. *Theoretical Computer Science*, 172(1-2):265–271, 1997.