Can Local Optimality Be Used for Efficient Data Reduction?

Christian Komusiewicz[0000−0003−0829−7032], Nils Morawietz*

Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, Marburg, Germany
`{komusiewicz,morawietz}@informatik.uni-marburg.de`

Abstract. An independent set S in a graph G is k-swap optimal if there is no independent set S' such that $|S'| > |S|$ and $|(S' \setminus S) \cup (S \setminus S')| \leq k$. Motivated by applications in data reduction, we study whether we can determine efficiently if a given vertex v is contained in some k-swap optimal independent set or in all k-swap optimal independent sets. We show that these problems are NP-hard for constant values of k even on graphs with constant maximum degree. Moreover, we show that the problems are Σ_2^{P}-hard when k is not constant, even on graphs of constant maximum degree. We obtain similar hardness results for determining whether an edge is contained in a k-swap optimal max cut. Finally, we consider a certain type of edge-swap neighborhood for the Longest Path problem. We show that for a given edge we can decide in $f(\Delta + k) \cdot n^{\mathcal{O}(1)}$ time whether it is in some k-optimal path.

Keywords: local search, independent set, max-cut, longest path, NP-hardness

1 Introduction

Local search and data reduction are two widely successful strategies for coping with hard computational problems. Local search, which applies most naturally to optimization problems, aims at computing good heuristic solutions by using the following generic approach: Define a *local neighborhood relation* on the set of feasible solutions. Then, compute some feasible solution S. Now check whether there is a better solution S' in the local neighborhood of S. If this is the case, then replace S by S'. Otherwise, output the locally optimal solution S and stop.

Local search algorithms have been explored thoroughly from a practical and theoretical point of view [2,8,9,10,14]. The theoretical framework most closely related to our investigations is *parameterized local search*. Here, the local search neighborhood comes equipped with an operational parameter k that bounds the radius of the local search neighborhood. The size of the neighborhood is assumed to be polynomial in the input size for every fixed value of k. For example, in Independent Set one is given a graph G and asks for a largest vertex

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

To appear in Proceedings of the 12th International Conference on Algorithms and Complexity (CIAC '21), *Larnaca, Cyprus, May 2021. © Springer.*

set S such that no two vertices in S are adjacent. The feasible solutions are the independent sets of G. A natural neighborhood with a radius k is the k-swap neighborhood: Two vertex sets S and S' are in their respective k-swap neighborhoods if $|S \oplus S'| \leq k$, where $\oplus$ denotes the symmetric difference of two sets. In LS-Independent Set, one is then given a graph G, an independent set S, and an integer k and asks whether the k-swap neighborhood of S contains a larger independent set S'. LS-Independent Set can be solved in $\Delta^{\mathcal{O}(k)} \cdot n$ time [6,10]; the currently best fixed-parameter algorithm for LS-Independent Set is efficient in practice, solving the parameterized local search problem for $k \approx 20$ even on large real-world graphs [10]. Summarizing, for moderate values of k, a k-swap optimal independent set can be computed faster than an optimal one.

In data reduction, the idea is to preprocess any instance of a hard problem by identifying those parts of the instance that are easy to solve. Usually, data reduction algorithms are stated as a collection of data reduction rules which can be applied if a certain precondition is fulfilled and reduce the instance size whenever they apply. Two classic trivial reduction rules for Independent Set are as follows: First, remove any vertex v that has no neighbors in G and add it to the independent set that is computed for the remaining instance. Second, remove any vertex v that has two degree-one neighbors.

In this work, we aim to explore the usefulness of local search or, more precisely, of local optimality, in the context of data reduction for hard problems. The correctness of the first reduction rule above is rooted in the observation that v is contained in every maximum independent set. Similarly, the correctness of the second rule is rooted in the fact that v is contained in no maximum independent set. Most of the known data reduction rules employ this principle and the crux of proving the correctness of a data reduction rule lies in proving that v is contained in all or no optimal solution. In general, given a vertex v and computing whether v is contained in a largest independent set is just as hard as computing an optimal solution in the first place. This is why data reduction rules use specific preconditions that allow proving optimality of including or excluding v without computing an optimal solution.

One may avoid such specific preconditions by relying on *locally* optimal solutions instead of optimal solutions: If we could compute efficiently that a vertex v is *not* contained in a locally optimal solution for *some* local search neighborhood, then we could remove v from the graph. Conversely, if we could compute efficiently that some vertex v is contained in *every* locally optimal solution, then we could remove v and its neighborhood from the graph as described above. The hope is that, since computing locally optimal solutions is easier for suitable local search neighborhoods, this approach could help in circumventing the previous dilemma: to say something about the optimal solution, one more or less needs to compute it. Going back to the Independent Set problem, we would like to determine whether a given vertex is in some k-swap optimal independent set.

Some Locally Optimal Independent Set ($\exists$-LO-IS)
Input: An undirected graph $G = (V, E)$, a vertex $v \in V$, and $k \in \mathbb{N}$.
Question: Is v contained in some k-swap optimal independent set in G?

If the answer is no, then v can be removed from G without destroying any optimal solution. We may also ask whether v is in every locally optimal independent set.

> Every Locally Optimal Independent Set ($\forall$-LO-IS)
> **Input:** An undirected graph $G = (V, E)$, a vertex $v \in V$, and $k \in \mathbb{N}$.
> **Question:** Is v contained in every k-swap optimal independent set in G?

If the answer is yes, then v belongs to every optimal independent set and we can apply a data reduction rule that removes v and its neighbors and adds v to the independent set that is computed for the remaining graph. Motivated by the usefulness of efficient algorithms for these two problems in data reduction for Independent Set, we study their complexity. We consider related problems for two further classic NP-hard problems: Max Cut and Longest Path.

Our Results. For Independent Set our results are decidedly negative. $\exists$-LO-IS and $\forall$-LO-IS are NP-complete and coNP-complete, respectively, even if $k = 3$ and $\Delta = 4$ and also if $k = 5$ and $\Delta = 3$. These results are tight [1] in the following sense: when $k = 1$ or when $\Delta = 2$, then both problems can be solved in polynomial time. Moreover, when k is not constant, we show that the problems are even Σ_2^{P}-complete and Π_2^{P}-complete, respectively. Thus, both problems are substantially harder than LS-Independent Set. For Max Cut the situation is similar: Deciding whether some edge is contained in a k-swap optimal cut or in every k-swap optimal cut is NP-complete and coNP-complete even if $k = 1$ and $\Delta = 5$. Moreover, if k is not constant then the problems are, again, Σ_2^{P}-complete and Π_2^{P}-complete, respectively.

Finally, we consider Longest Path with a certain edge-swap neighborhood. We show that for this neighborhood we can determine in $f(\Delta + k) \cdot n^{\mathcal{O}(1)}$ time whether some edge is contained in a k-optimal path. If the answer is no, then this edge can be safely removed from the input graph. Since Longest Path is NP-hard on cubic graphs, our results indicate that there are scenarios in which testing for the containment of edges in locally optimal solutions is a viable approach to obtain data reduction rules for NP-hard problems.

Related Work. A related problem is to determine if there is a maximal (and, thus, 1-swap optimal) independent set S containing only vertices of a specific subset of vertices $U \subseteq V$ [3,12]. In other words, this problem asks for a 1-swap optimal independent set containing no vertex of $V \setminus U$, a generalization of the complement problem of $\exists$-LO-IS for $k = 1$. This problem is NP-hard even on graphs where Independent Set can be solved in polynomial time, like bipartite graphs [3]. In the scope of 1-swaps, the Independent Set Reconfiguration problem was analyzed extensively [1,4,5,13]. Here, we are given two independent sets X and Y of a graph G and an integer k and we want to determine if there is a sequence $S_1, \ldots, S_r$ of independent sets such that $S_1 = X$, $S_r = Y$, $|S_j| \geq k$ for all $j \in [1, r]$, and $|S_j \oplus S_{j+1}| = 1$ for all $j \in [1, r-1]$. Hence, one wants to add or

[1] For even k, an independent set is k-swap optimal if and only if it is $(k-1)$-swap optimal [10]. Thus, only odd values of k are interesting.

remove a single vertex at a time and transform X into Y without decreasing the size of the current independent set below k. This problem is PSPACE-complete even on bipartite graphs [13].

The proofs of statements marked with a (*) are deferred to a full version.

2 Preliminaries

Sets and Graphs. For a set A, we denote with $\binom{A}{2} := \{\{a,b\} \mid a \in A, b \in A\}$ the collection of all size-two subsets of A. 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 $E \subseteq \binom{V}{2}$. 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 $E_G(S) := E_G(S,S)$. 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 $N_G[v] := N_G(v) \cup \{v\}$ 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 $N_G(S) := N_G[S] \setminus S$. If G is clear from the context, we may omit the subscript. Moreover, we denote with $\Delta(G) := \max\{|N_G(v)| \mid v \in V\}$ the *maximum degree* of G.

A sequence of distinct vertices $P = (v_0, \ldots, v_k)$ is a *path* or (v_0, v_k)*-path* of length k in G if $\{v_{i-1}, v_i\} \in E(G)$ for all $i \in [1,k]$. We denote with $V(P)$ the vertices of P and with $E(P)$ the edges of P.

Satisfiability. For *variable set* Z, we define the set of *literals* $\mathcal{L}(Z) := Z \cup \{\neg z \mid z \in Z\}$. A literal set $\tilde{Z} \subseteq \mathcal{L}(Z)$ is an *assignment* of Z if $|\{z, \neg z\} \cap \tilde{Z}| = 1$ for all $z \in Z$. For a subset $X \subseteq Z$ of the variables we denote with $\tau_Z(X) := X \cup \{\neg z \mid z \in Z \setminus X\}$, the assignment of Z where all variables of X occur positively and all variables of $Z \setminus X$ occur negatively. A *clause* $\phi \subseteq \mathcal{L}(Z)$ is *satisfied* by an assignment τ of Z if $\phi \cap \tau \neq \emptyset$, and we write $\tau \models \phi$. Analogously, a set $\Phi \subseteq \mathbb{P}(\mathcal{L}(Z))$ of clauses is satisfied by τ if $\tau \models \phi$ for all $\phi \in \Phi$, and we write $\tau \models \Phi$.

3 Independent Set

In this section, we analyze the complexity of ∃-LO-IS and ∀-LO-IS. First, let us set the following notation. Let $G = (V, E)$ be a graph and let k be an integer. We call a subset $W \subseteq V$ a k-swap in G if $|W| \leq k$. A set $S \subseteq V$ is an *independent set* if $\{v,w\} \notin E$ for all $v, w \in S$. Further, an independent set S is k*-swap optimal* in G if there is no k-swap W in G such that $S \oplus W$ is an independent set and $|S| < |S \oplus W|$. We will make use of the following observation on improving k-swaps.

Observation 1 ([10, Lemma 2]) *Let S be an independent set for a graph $G = (V, E)$ and let k be an integer. Then, S is k-swap optimal if and only if there is no swap $C \subseteq V$ of size at most k such that $G[C]$ is connected and $|S \oplus C| = |S| + 1$.*

Observation 2 *An instance (G, v, k) of* $\forall$-LO-IS *is a yes-instance if and only if (G, w, k) is a no-instance of* $\exists$-LO-IS *for every $w \in N(v)$.*

First, we observe that we can solve $\exists$-LO-IS in polynomial time for the following almost trivial cases. Note that for $k = 1$, we ask whether a vertex is contained in some maximal independent set.

Proposition 1 (*). $\exists$-LO-IS *and* $\forall$-LO-IS *can be solved in polynomial time if $k = 1$ or $\Delta \leq 2$.*

We now show that we cannot improve upon Proposition 1, neither in terms of k nor in terms of Δ.

Theorem 3 (*). $\exists$-LO-IS *is* NP*-complete and* $\forall$-LO-IS *is* coNP*-complete even if $k = 3$ and $\Delta = 4$.*

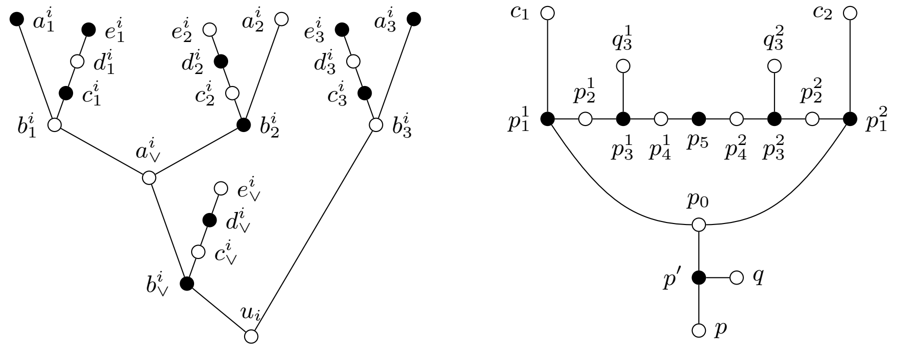
Theorem 4. $\exists$-LO-IS *is* NP*-complete and* $\forall$-LO-IS *is* coNP*-complete even if $k = 5$ and $\Delta = 3$.*

Proof. First, we show the statement for $\exists$-LO-IS via reducing from SAT. Given an instance $I = (Z, \Phi)$ of SAT, we construct in polynomial time an equivalent instance $I' = (G = (V, E), v_\exists, k)$ of $\exists$-LO-IS where $k = 5$ and where G has maximum degree three. We may assume that every clause has size three, every variable of Z occurs twice positively and twice negatively in Φ, and every variable occurs at most once per clause since SAT remains NP-hard in this case [15].

Let ψ denote the number of clauses and let $\Phi = \{\phi_1, \ldots, \phi_\psi\}$. We start with an empty graph G and add for every variable $z \in Z$ a cycle $(z^1, \neg z^1, z^2, \neg z^2, z^1)$. We add for every clause $\phi_i = \{\ell_1, \ell_2, \ell_3\} \in \Phi$ the subgraph $G_{\phi_i} = (V_{\phi_i}, E_{\phi_i})$ shown in Fig. 1a. The graph G_{ϕ_i} contains the vertex u_i, for each $j \in \{1, 2, 3, \vee\}$, the path $(a^i_j, b^i_j, c^i_j, d^i_j, e^i_j)$, and the edges $\{b^i_1, a^i_\vee\}$, $\{b^i_2, a^i_\vee\}$,$\{b^i_\vee, u_i\}$, and $\{b^i_3, u_i\}$. We connect a gadget G_ϕ with the cycles of the variables of ϕ as follows: for every $j \in [1, 3]$ we add the edge $\{\ell^1_j, a^i_j\}$ if ℓ^1_j is not connected to any clause gadget already. Otherwise, we add the edge $\{\ell^2_j, a^i_j\}$. Since every variable occurs twice positively and twice negatively in Φ, the vertices ℓ^1_j and ℓ^2_j are each connected to exactly one subgraph G_{ϕ_i} at the end of the construction. The idea behind G_{ϕ_i} is that every 5-swap optimal independent set S for G containing the vertices a^i_1, a^i_2, and a^i_3, also contains the vertex u_i. Hence, if no vertex representing any literal of ϕ_i is contained in S (that is, if ϕ_i is not satisfied), then u_i is contained in S which will imply that $v_\exists$ is not contained in S by the remaining gadgets.

Next, we add a binary tree with leaf vertices $\{u_i \mid \phi_i \in \Phi\}$, the set of inner vertices T, and root r to G. Afterwards, we remove all edges of this tree and connect every parent vertex p with his two child vertices c_1 and c_2 by adding the subgraph $G_p = (V_p, E_p)$ shown in Fig. 1b. This subgraph contains the vertices of the binary tree p, c_1 and c_2, the vertices p', q, q^1_3, q^2_3, and a cycle $(p_0, p^1_1, p^1_2, p^1_3, p^1_4, p_5, p^2_4, p^2_3, p^2_2, p^2_1, p_0)$. Further, the set E_p contains the edges $\{p^1_3, q^1_3\}$, $\{p^2_3, q^2_3\}$, $\{p^1_1, c_1\}$, $\{p^2_1, c_2\}$, $\{p, p'\}$, $\{p', q\}$, and $\{p', p_0\}$.

Finally, we add a path (v_0, v_1, v_2, v_3, r) to G and set $v_\exists := v_1$. This completes the construction of I'. Note that the constructed graph has a maximum degree of



(a) The subgraph G_{ϕ_i} for a clause $\phi_i = \{\ell_1, \ell_2, \ell_3\} \in \Phi$. Black vertices belong to $V_{\phi_i}(\tau)$ with $\tau \cap \phi_i = \{\ell_2\}$.

(b) The subgraph G_p for some parent vertex $p \in T$ with the child vertices c_1 and c_2. Black vertices are contained in every 5-swap optimal independent set containing $v_\exists$.

Fig. 1: The gadgets of the reduction of Theorem 4.

three. We show that I is a yes-instance of SAT if and only if I' is a yes-instance of $\exists$-LO-IS.

($\Rightarrow$) Suppose that I is a yes-instance of SAT. Thus, there is an assignment τ for Z such that $\tau \models \Phi$. Let $\phi_i = \{\ell_1, \ell_2, \ell_3\}$ be a clause of Φ and let τ be an assignment of Z. We set

$$V_{\phi_i}(\tau) := \{b_j^i, d_j^i \mid \ell_j \in \tau\} \cup \{a_j^i, c_j^i, e_j^i \mid \ell_j \notin \tau\} \cup \begin{cases} \{b_\vee^i, d_\vee^i\} & \tau \models \{\ell_1, \ell_2\} \\ \{a_\vee^i, c_\vee^i, e_\vee^i\} & \text{otherwise} \end{cases}.$$

Note that $V_{\phi_i}(\tau)$ is an independent set. We set $S := \{\ell^1, \ell^2 \mid \ell \in \tau\} \bigcup_{\phi_i \in \Phi} V_{\phi_i}(\tau) \cup \{p', p_1^1, p_3^1, p_5, p_3^2, p_1^2 \mid p \in T\} \cup \{v_1, v_3\}$. That is, S contains the vertices representing the literals of τ, the vertices of the clause gadgets according to τ, the black vertices of V_p shown in Fig. 1b for every $p \in T$, and the vertices v_3 and $v_1 = v_\exists$. By construction, S is an independent set. It remains to show that S is 5-swap optimal. To this end, we observe the following.

Claim 1 (*). For each vertex $w \in V \setminus S$ with $|N(w) \cap S| \leq 1$, we have $|N(w)| = 1$.

Since a 5-swap W for S with $|S \oplus W| > |S|$ has to contain two distinct vertices $w_1, w_2 \in V \setminus S$ with $|N(w_1) \cap S| = |N(w_2) \cap S| = 1$ we obtain by Claim 1 and the fact that degree-one vertices in G have pairwise distance at least six, that there is no such W. Consequently, S is a 5-swap optimal independent set in G with $v_\exists = v_1 \in S$.

($\Leftarrow$) Suppose that I' is a yes-instance of $\exists$-LO-IS. Thus, there is a 5-swap optimal independent set S in G with $v_\exists = v_1 \in S$. We first observe the following.

Claim 2 (*). Let $p \in T$ with the child vertices c_1 and c_2. If $p \notin S$ and $p' \in S$, then $|N(c_1) \cap S| \geq 2$ and $|N(c_2) \cap S| \geq 2$.

Note that this also implies, that $c_1 \notin S$ and $c_2 \notin S$.

Recall that r is the root of the binary tree and that r' is the unique neighbor of r in G_r. Now, observe that $\{v_1, v_3, r'\} = \{v_\exists, v_3, r'\} \subseteq S$ and $\{v_0, v_2, r\} \subseteq V \setminus S$ as, otherwise, S would not be a 5-swap optimal independent set in G. Thus, by Claim 2 one can show inductively, that for all vertices c of the binary tree it holds that $|N(c) \cap S| \geq 2$. Consequently, this also holds for the leaves of the tree, namely the vertices $\{u_i \mid i \in [1, \psi]\}$. Hence, for every $\phi_i \in \Phi$ it holds that $u_i \notin S$ and that $b^i_\vee \in S$ or $b^i_3 \in S$.

We set $\tau = \{\ell \in \mathcal{L}(Z) \mid \{\ell^1, \ell^2\} \cap S \neq \emptyset\}$ and show that $\tau \models \Phi$. Note that τ contains at most one of z and $\neg z$ since S is an independent set. Let $\phi_i = \{\ell_1, \ell_2, \ell_3\}$ be a clause of Φ, we show that $\tau \models \phi_i$. First, we show that if there is some $j \in \{1, 2, 3, \vee\}$ with $N(a^i_j) \cap S = \{b^i_j\}$, then S is not 5-swap optimal. Since S is an independent set, it holds that $c^i_j \notin S$. If $d^i_j \notin S$, then $S \oplus \{a^i_j, b^i_j, c^i_j\}$ is an independent set and, thus, S is not 5-swap optimal. Otherwise, $d^i_j \in S$ and, thus, $e^i_j \notin S$. Hence, $S \oplus \{a^i_j, b^i_j, c^i_j, d^i_j, e^i_j\}$ is an independent set and, thus, S is not 5-swap optimal.

We may thus assume that if $a^i_j \notin S$, then $N(a^i_j)$ is a subset of S containing ℓ^1 or ℓ^2. We now use this fact to argue that at least one literal vertex adjacent to any vertex a^i_j is contained in S and, thus, ϕ_i is satisfied by τ. Recall that $b^i_\vee \in S$ or $b^i_3 \in S$. Consequently, if $b^i_\vee \in S$, then $|N(a^i_\vee) \cap S| \geq 2$ and, therefore, $\{b^i_1, b^i_2\} \cap S \neq \emptyset$. Hence, there is some $j \in \{1, 2, 3\}$ such that $b^i_j \in S$ and, thus, $N(a^i_j) \subseteq S$. As a consequence, $\{\ell^1_j, \ell^2_j\} \cap S \neq \emptyset$ and, therefore, $\ell_j \in \tau$. Hence, ϕ_i is satisfied by τ. Consequently, I is a yes-instance of SAT.

Due to Observation 2 and the fact that $N(v_0) = \{v_\exists\}$ it follows that (G, v_0, k) is a yes-instance of $\forall$-LO-IS if and only if I' is a no-instance of $\exists$-LO-IS. Consequently, $\forall$-LO-IS is coNP-hard even if $k = 5$ and where the input graph has a maximum degree of three. □

Corollary 1. *For every fixed odd $k \geq 5$, $\forall$-LO-IS is coNP-complete and $\exists$-LO-IS is NP-complete even if $\Delta = 3$.*

Proof. Let $k \geq 7$ and let $I = (G = (V, E), v_\exists, 5)$ be an instance of $\exists$-LO-IS constructed as in the proof of Theorem 4. By adding for every degree-one vertex w in G a path P_w with $k - 5$ vertices to G and connecting w with one endpoint of P_w, we obtain an equivalent instance $I' = (G', v_\exists, k)$ of $\exists$-LO-IS. □

Next, we analyze the case where the swap distance k is unbounded.

Theorem 5 (*). *$\forall$-LO-IS is Π^P_2-complete ($\exists$-LO-IS is Σ^P_2-complete) if $\Delta = 3$.*

4 Max Cut

We now analyze the complexity of deciding whether an edge is a cut edge of some locally optimal cut for the Max Cut problem. We formally define cuts and their local neighborhoods as follows. Let $S, T \subseteq V$. The pair (S, T) is a *cut* in G if $S \cup T = V$ and $S \cap T = \emptyset$. A cut (S, T) is a *k-swap optimal cut* in G if

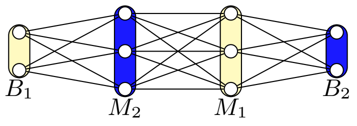


Fig. 2: A $(2,3)$-enforcer $F = (B_1 \cup B_2 \cup M_1 \cup M_2, F_E)$.

there is no k-swap W in G such that $|E(S', T')| > |E(S, T)|$ where $S' = S \oplus W$ and $T' = T \oplus W = V \setminus S'$. Let $A, B \subseteq V$. We say that A and B are *in the same part of the cut* if $A \cup B \subseteq S$ or $A \cup B \subseteq T$. Moreover, A and B are *in opposite parts of the cut* if $A \subseteq S$ and $B \subseteq T$ or vice versa.

Every Locally Optimal Max-Cut ($\forall$-LO-MC)
Input: An undirected graph $G = (V, E)$, an edge $e \in E$, and $k \in \mathbb{N}$.
Question: Is e contained in every k-swap optimal cut in G?

Analogously, we ask in Some Locally Optimal Max-Cut ($\exists$-LO-MC) if e is contained in some k-swap optimal cut in G.

For every fixed value of k, we can check in polynomial time if a given cut (S, T) is k-swap optimal in G. Consequently, we obtain the following.

Lemma 1. *For every fixed value of k, $\forall$-LO-MC is contained in coNP and $\exists$-LO-MC is contained in NP.*

We now show that both problems are hard, even for constant k and Δ. In the reduction, we use a graph called $(2,3)$-*enforcer* which is shown in Fig. 2. For a $(2,3)$-enforcer F, we denote $F(i) := B_i \cup M_i$ for $i \in \{1, 2\}$.

Proposition 2 (*). *Let G be a graph with $\Delta(G) = 5$. If G contains a $(2,3)$-enforcer $F = (B_1 \cup B_2 \cup M_1 \cup M_2, E')$ as an induced subgraph, then for every 1-swap optimal cut (S, T) in G it holds that $F(1)$ and $F(2)$ are in opposite parts of the cut.*

Theorem 6. *$\exists$-LO-MC is NP-complete and $\forall$-LO-MC is coNP-complete even if $k = 1$ and $\Delta = 5$.*

Proof. We reduce SAT to $\exists$-LO-MC. Given an instance $I = (Z, \Phi)$ of SAT, we construct in polynomial time an equivalent instance $I' = (G = (V, E), e_\exists, k)$ of $\exists$-LO-MC where $k = 1$ and where G has a maximum degree of five. We can assume without loss of generality that every clause has size three, every variable of Z occurs twice positively and twice negatively in Φ, and every variable occurs at most once per clause since SAT remains NP-hard in this case [15]. Further, we let ψ denote the number of clauses and we assume that $\psi = 2^r$ for some even r.

Let $\Phi = \{\phi_1, \ldots, \phi_\psi\}$. We start with a balanced binary tree with $r + 1$ levels. We denote with $L_p := \{u_q^p \mid q \in [1, 2^p]\}$ the vertices of the pth level of the tree for all $p \in [0, r]$. The leaf u_q^r represents the clause ϕ_q for all $q \in [1, \psi]$. Further, we add a $(2,3)$-enforcer $F_\exists$ with $B_1 = \{w_\exists^1, w_\exists^2\}$ and $B_2 = \{w_\forall^1, w_\forall^2\}$ and, for each variable $v \in Z$, a $(2,3)$-enforcer F_v with $B_1 = \{v, v'\}$ and $B_2 = \{\neg v, \neg v'\}$.

The idea is that $F_v(1)$ corresponds to the true-assignment of v and that $F_v(2)$ corresponds to the false-assignment of v since $v \in F_v(1)$ and $\neg v \in F_v(2)$. By Proposition 2, $F_v(1)$ and $F_v(2)$ are in opposite parts of every 1-swap optimal cut for G. Thus, for every 1-swap optimal cut (S,T) for G, both $S \cap \mathcal{L}(Z)$ and $T \cap \mathcal{L}(Z)$ are assignments for the variables of Z.

For each clause $\phi_q \in \Phi$ and each literal $\ell \in \phi_q$ we add the edge $\{\ell, u_q^r\}$. Furthermore, we connect the vertices of $L_1 \cup \{w_\exists^2\}$ to a cycle of length three and for $p \in [2, r-1]$ we connect the vertices of L_p to a cycle of length 2^p. Finally, we add the edges between u_1^0 and each of the vertices $w_\forall^1, w_\forall^2$, and $w_\exists^1$ and set $e_\exists = \{w_\exists^1, u_1^0\}$. This completes the construction of I'. We now show that I is a yes-instance of SAT if and only if I' is a yes-instance of $\exists$-LO-MC.

($\Rightarrow$) Suppose that I is a yes-instance of SAT. Thus, there is $\tilde{Z} \subseteq Z$ such that $\tau_Z(\tilde{Z})$ satisfies Φ. We set

$$S := F_\exists(1) \cup \bigcup_{\text{odd } p \in [1,r-1]} L_p \cup \bigcup_{v \in \tilde{Z}} F_v(1) \cup \bigcup_{v \in Z \setminus \tilde{Z}} F_v(2)$$

and $T = V \setminus S$ and show that (S,T) is a 1-swap optimal cut in G and contains $e_\exists$. Note that for every $\ell \in \mathcal{L}(Z)$ it holds that $\ell \in S$ if and only if $\ell \in \tau_Z(\tilde{Z})$.

By construction, G has a maximum degree of five. Thus, for every vertex v in any enforcer it follows directly that at least half of the neighbors of v are in the opposite part of the cut of v. Further, since for every $v \in L_p, p \in [1, r-1]$, it holds that $|N_G(v) \cap (L_{p-1} \cup L_{p+1})| = 3$ and, thus, by construction of (S,T) that at least half of the neighbors of v are in the opposite part of the cut of v. The same also holds for the root u_1^0 since $u_1^0 \in T$ and $N_G(u_1^0) \cap S = L_1 \cup \{w_\exists^1\}$. Since $L_r \subseteq T$, it remains to show that for every $q \in [1, \psi]$, $|N_G(u_q^r) \cap S| \geq 2$. By definition of (S,T) it follows that $|N_G(u_q^r) \cap L_{r-1} \cap S| = 1$. The fact that $\tau_Z(\tilde{Z}) \models \Phi$ implies that $\phi_q \cap S \neq \emptyset$. Consequently, for every $v \in V$, at least half of the neighbors of v are in the opposite part of the cut of v. Therefore, (S,T) is a 1-swap optimal cut in G and contains $e_\exists$.

($\Leftarrow$) Let (S,T) be a 1-swap optimal cut in G which contains $e_\exists$. By Proposition 2, we may assume without loss of generality that $F_\exists(1) \subseteq S$ and $F_\exists(2) \subseteq T$. Further, for every $v \in Z$, either $F_v(1) \subseteq S$ or $F_v(2) \subseteq S$. We set $\tilde{Z} := \{v \in Z \mid F_v(1) \subseteq S\}$ and show that $\tau_Z(\tilde{Z}) \models \Phi$. Note that for every literal $\ell \in \mathcal{L}(Z)$ it holds that $\ell \in \tau_Z(\tilde{Z})$ if and only if $\ell \in S$.

Claim 3 (*). For every $p \in [0, r]$, it holds that $L_p \subseteq S$ if p is odd and $L_p \subseteq T$ if p is even.

As a consequence, $u_q^r \in T$ for all $q \in [1, \psi]$. Since (S,T) is 1-swap optimal in G, it holds that u_q^r has at least two neighbors in S: the single neighbor in L_{q-1} and at least one neighbor in $N_G(u_q^r) \setminus L_{q-1} = \phi_q$ and, thus, $S \cap \phi_q \neq \emptyset$. Consequently, $\tau_Z(\tilde{Z}) \models \Phi$.

Thus, $\exists$-LO-MC is NP-complete if $k = 1$ and $\Delta = 5$. Due to Proposition 2 and the fact that the $(2,3)$-enforcer $F_\exists$ contains both $w_\exists^1$ and $w_\forall^1$, we know

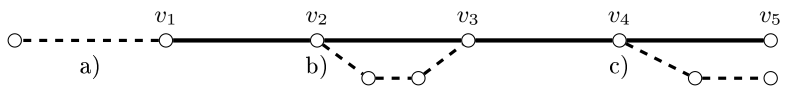


Fig. 3: The three possible kinds of minimal improving $(1, k)$-swaps: a) appending an edge to one of the endpoints, b) replacing an edge $\{u, v\}$ with a (u, v)-path of length at most k, and c) removing one endpoint and appending a path of length two to its neighbor in the path.

that for every 1-swap optimal cut (S, T) in G it holds that $w_\exists^1 \in S$ if and only if $w_\forall^1 \in T$. Hence, $I' = (G, e_\exists, 1)$ is a no-instance of $\exists$-LO-MC if and only if $I'' = (G, e_\forall, 1)$ is a yes-instance of $\forall$-LO-MC, where $e_\forall := \{w_\forall^1, u_1^0\}$. Consequently, $\forall$-LO-MC is coNP-complete if $k = 1$ and $\Delta = 5$. $\square$

Theorem 7 (*). *$\forall$-LO-MC is Π_2^P-complete and $\exists$-LO-MC is Σ_2^P-complete even if $\Delta = 3$.*

5 Longest Path

Finally, we consider Longest Path which is NP-hard even on cubic graphs [7]. Again, we want to find out whether some small part of the graph is contained in a locally optimal solution. We consider the following neighborhood.

Definition 1. *Let k be an integer. Two paths P_1 and P_2 are $(1, k)$-swap neighbors, if 1) the relative ordering of the common vertices of P_1 and P_2 is equal in both paths and 2) $|E(P_1) \setminus E(P_2)| \leq k$ and $|E(P_2) \setminus E(P_1)| \leq 1$ or $|E(P_1) \setminus E(P_2)| \leq 1$ and $|E(P_2) \setminus E(P_1)| \leq k$.*

All three kinds of minimal improving $(1, k)$-swaps are shown in Fig. 3.

Some Locally Optimal Path ($\exists$-LO-Path)
Input: An undirected graph $G = (V, E)$, an edge $e^* \in E$, and $k \in \mathbb{N}$.
Question: Is e^* contained in some $(1, k)$-swap optimal path P^* in G?

First, we observe that, unfortunately, the problem is hard already for fixed k.

Theorem 8 (*). *$\exists$-LO-Path is contained in* NP *for every fixed value of k and* NP*-hard for every $k \geq 2$.*

We now show that on bounded-degree graphs, we can obtain an efficient algorithm for small k. Our algorithm is based on a sufficient and necessary condition for the existence of a $(1, k)$-swap optimal path in G containing a specific edge. To formulate the condition, we define two collections of vertex sets. We denote for every integer k and every pair of distinct vertices $v, w \in V$ with $\mathcal{V}_k(v, w) := \{V(P) \setminus \{v, w\} \mid P \text{ is an } (v, w)\text{-path with } 2 < |V(P)| \leq k + 1\}$ the collection of sets of inner vertices of (v, w)-path with length at most k. Moreover, for every vertex $v \in V$ we denote by $\mathcal{V}_2(v) := \{\{x, y\} \mid x, y \in V, (v, x, y) \text{ is a path in } G\}$ the collection of subsets of $\tilde{V} \subseteq V \setminus \{v\}$, such that G contains a path P of length two starting in v with $V(P) = \tilde{V} \cup \{v\}$.

Lemma 2. *Let $e^* = \{v^*, w^*\}$ be an edge which is not isolated and let $k \geq 2$. There is a $(1,k)$-swap optimal path P^* containing e^* if and only if there are (not necessarily distinct) vertices s, s', t' and t such that there is an (s,t)-path P in G containing the edges $e^*, \{s,s'\}, \{t',t\}$, the vertices $N(s) \cup N(t)$, and where $V(P)$ is a hitting set for $\mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_2(t') \cup \mathcal{V}_k(t',t)$.*

Proof. ($\Rightarrow$) Let P^* be a $(1,k)$-swap optimal path containing e^*. Then, $V(P^*)$ contains at least two vertices and, thus, there are vertices $s, s', t', t \in V$ such that P^* contains the edges $e^*, \{s,s'\}$, and $\{t',t\}$. Moreover, $V(P^*)$ contains $N(s) \cup N(t)$ as, otherwise, P^* is not $(1,k)$-swap optimal. Assume towards a contradiction, that $V(P^*)$ is not a hitting set for $\mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_r(t',t) \cup \mathcal{V}_2(t')$.

Case 1: $V(P^*)$ is not a hitting set for $\mathcal{V}_k(x,y)$ for some $\{x,y\} \in \{\{s,s'\},\{v^*,w^*\},\{t',t\}\}$. Then, there is some $\tilde{V} \in \mathcal{V}_k(x,y)$ such that there is some (x,y)-path P' in G of length at most k and $V(P') = \tilde{V} \cup \{x,y\}$, such that $V(P') \cap V(P^*) = \{x,y\}$. Replacing the edge $\{x,y\}$ by the (x,y)-path P' is an improving $(1,k)$-swap, since $\tilde{V} \neq \emptyset$. This contradicts the fact that P^* is $(1,k)$-swap optimal in G.

Case 2: $V(P^*)$ is not a hitting set for $\mathcal{V}_2(x)$ for some $x \in \{s',t'\}$. Then, there is some $\tilde{V} \in \mathcal{V}_2(x)$ such that there is some path P' of length two in G starting in x with $V(P') = \tilde{V} \cup \{x\}$ and $V(P') \cap V(P^*) = \{x\}$. Hence, replacing the edge $\{x,x'\}$ by the path P' is an improving $(1,k)$-swap, since P' contains two edges. This contradicts the fact that P^* is $(1,k)$-swap optimal in G.

($\Leftarrow$) Suppose that there are vertices $s, s', t', t \in V$ such that there is an (s,t)-path P in G containing the edges $e^*, \{s,s'\}, \{t',t\}$, the vertices $N(s) \cup N(t)$, and where $V(P)$ is a hitting set for $\mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_k(t',t) \cup \mathcal{V}_2(t')$.

Since P contains 1) at least one inner vertex of every (s,s')-path in G of length at least two and at most k, and 2) at least one inner vertex besides s' of every path of length two starting in s', there is no improving $(1,k)$-swap that removes $\{s,s'\}$ from P. This also holds for the edge $\{t',t\}$. Since $V(P)$ is a hitting set for $\mathcal{V}_k(v^*,w^*)$, P contains at least one inner vertex of every (v^*,w^*)-path in G of length at least two and at most k.

In the following, we will show, that every path P^* which can be reached by an arbitrary number of improving $(1,k)$-swaps, also fulfills all properties of P. Note that this implies that there is a $(1,k)$-swap optimal path P^* in G containing e^*.

Since e^* is not an isolated edge and P contains the vertices $N(s) \cup N(t)$, and the edges $e^*, \{s,s'\}, \{t',t\}$, we obtain that P contains at least three vertices. Hence, not all edges of P can be removed by a single improving $(1,k)$-swap. Note that an improving $(1,k)$-swap can only remove a vertex from the path if this vertex is one of the endpoints.

By the above, none of the edges $e^*, \{s,s'\}$, or $\{t',t\}$ can be removed by an improving $(1,k)$-swap. Hence, for every improving $(1,k)$-swap neighbor P' of P it holds that $V(P) \subseteq V(P')$, P' contains the edges $e^*, \{s,s'\}$, and $\{t',t\}$. Moreover, since $N(s) \cup N(t) \subseteq V(P)$, we obtain that P' is also an (s,t)-path. Consequently, one can show via induction, that for every path P^* which can be reached by an arbitrary number of improving $(1,k)$-swap starting from P, that P^* is an (s,t)-path containing the edges $e^*, \{s,s'\}$, and $\{t',t\}$, the vertices $N(s) \cup N(t)$, and $V(P^*)$ is a hitting set for $\mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_2(t') \cup \mathcal{V}_k(t',t)$. $\square$

Theorem 9. $\exists$-LO-Path *can be solved in time* $\mathcal{O}(f(\Delta + k) \cdot n^4)$ *for some computable function* f.

Proof. Let $I = (G = (V,E), e^* = \{v^*, w^*\}, k)$ be an instance of Some Locally Optimal Path. First, if e^* is an isolated edge, then, there is exactly one path $P^* = (v^*, w^*)$ in G that contains the edge e^*. This path is $(1,k)$-swap optimal in G if and only if G does not contain a path with two edges. Hence, to determine if I is a yes-instance of Some Locally Optimal Path, we only have to check if G contains a path with two edges, which can be done in $\mathcal{O}(n^2)$ time. Second, if e^* is not an isolated edge, then, due to Lemma 2, it is sufficient to check if there are vertices $s, s', t', t \in V$ such that there is an (s,t)-path P in G containing the edges $e^*, \{s,s'\}$, and $\{t',t\}$, the vertices $N(s) \cup N(t)$, and where $V(P)$ is a hitting set for $\mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_k(t',t) \cup \mathcal{V}_2(t')$. In the following, we describe how we can check in $\mathcal{O}(f(\Delta + r) \cdot n^4)$ time, if such a path exists.

For every combination of vertices $s \in V, s' \in N(s), t \in V$, and $t' \in N(t)$, we compute the collections $\mathcal{V}_k(s,s'), \mathcal{V}_2(s'), \mathcal{V}_k(v^*,w^*), \mathcal{V}_k(t',t)$, and $\mathcal{V}_2(t')$. Let $\mathcal{V} := \mathcal{V}_k(s,s') \cup \mathcal{V}_2(s') \cup \mathcal{V}_k(v^*,w^*) \cup \mathcal{V}_k(t',t) \cup \mathcal{V}_2(t')$. Each of these collections contains at most Δ^{k-1} sets of size at most $k-1$ and each can be computed in $\mathcal{O}(\Delta^k)$ time. Moreover, for each set $V' \in \mathcal{V}_k(x,y)$ it holds that $V' \subseteq N^{k/2}[x] \cup N^{k/2}[y]$, where $N^{k/2}[u]$ denotes the set of vertices having distance at most $k/2$ to u. Hence, $V^* := \bigcup_{V' \in \mathcal{V}} V'$ is a subset of $\bigcup_{x \in \{s,s',v^*,w^*,t',t\}} N^{k/2}[x] \cup N^2[s'] \cup N^2[t']$. Consequently, there is some $\lambda \in \mathcal{O}(\Delta^{\max(k/2,2)})$ such that $|V^*| \leq \lambda$.

Next, we check for each $V' \subseteq V^*$ if it is a hitting set for $\mathcal{V}$. If this is the case, then we check if there is an (s,t)-path P in G containing the edges $e^*, \{s,s'\}$, and $\{t',t\}$ such that $V' \cup N(s) \cup N(t) \subseteq V(P)$. This can be done by checking if there is an ordering $\pi = (x_1, \dots, x_\ell)$ of the vertices of $\tilde{V} := V' \cup N(s) \cup N(t) \cup \{s,t,v^*,w^*\}$ such that there are pairwise vertex-disjoint (x_i, x_{i+1})-paths for all $i \in [1, \ell-1]$ where $x_1 = s, x_2 = s', x_{\ell-1} = t', x_\ell = t$, and v^* and w^* are consecutive in π. This can be done in $g(|\tilde{V}|) \cdot n^2$ time [11] for some computable function g. Since we check all combinations of s, s', t', and t as well as every possible hitting set for $\mathcal{V}$, the algorithm is correct and has overall running time $\mathcal{O}(n^2 \cdot \Delta^2 \cdot \lambda \cdot 2^\lambda \cdot (\lambda + 4 + 2\Delta)! \cdot g(\lambda + 4 + 2\Delta) \cdot n^2) \subseteq \mathcal{O}(f(\Delta + k) \cdot n^4)$. $\square$

6 Conclusion

We proposed a generic approach to the design of data reduction rules via local optimality and examined its viability for well-known NP-hard problems. It seems interesting to find further positive applications of this approach along the lines of the Longest Path problem. One might, for example, consider extensions of the $(1,k)$-swap neighborhood for paths. Finally, regardless of the connection to data reduction, it seems interesting in its own right to study which properties of locally optimal solutions can be computed efficiently.

References

1. Rémy Belmonte, Tesshu Hanaka, Michael Lampis, Hirotaka Ono, and Yota Otachi. Independent set reconfiguration parameterized by modular-width. In *Proceedings of the 45th International Workshop on Graph-Theoretic Concepts in Computer Science (WG '19), Vall de Núria, Spain, June 19-21, 2019, Revised Papers*, volume 11789 of *Lecture Notes in Computer Science*, pages 285–297. Springer, 2019.
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. Katrin Casel, Henning Fernau, Mehdi Khosravian Ghadikolaei, Jérôme Monnot, and Florian Sikora. Extension of vertex cover and independent set in some classes of graphs. In *Proceedings of the 11th International Conference on Algorithms and Complexity (CIAC '19)*, volume 11485 of *Lecture Notes in Computer Science*, pages 124–136. Springer, 2019.
4. Keren Censor-Hillel and Mikaël Rabie. Distributed reconfiguration of maximal independent sets. *J. Comput. Syst. Sci.*, 112:85–96, 2020.
5. Mark de Berg, Bart M. P. Jansen, and Debankur Mukherjee. Independent-set reconfiguration thresholds of hereditary graph classes. *Discret. Appl. Math.*, 250:165–182, 2018.
6. Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances A. Rosamond, Saket Saurabh, and Yngve Villanger. Local search: Is brute-force avoidable? *J. Comput. Syst. Sci.*, 78(3):707–719, 2012.
7. M. R. Garey, David S. Johnson, and Robert Endre Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976.
8. Jiong Guo, Sepp Hartung, Rolf Niedermeier, and Ondrej Suchý. The parameterized complexity of local search for TSP, more refined. *Algorithmica*, 67(1):89–110, 2013.
9. 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.
10. 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.
11. Ken-ichi Kawarabayashi, Yusuke Kobayashi, and Bruce A. Reed. The disjoint paths problem in quadratic time. *J. Comb. Theory, Ser. B*, 102(2):424–435, 2012.
12. Mehdi Khosravian Ghadikolaei, Nikolaos Melissinos, Jérôme Monnot, and Aris Pagourtzis. Extension and its price for the connected vertex cover problem. In *Proceedings of the 30th International Workshop on Combinatorial Algorithms (IWOCA '19)*, volume 11638 of *Lecture Notes in Computer Science*, pages 315–326. Springer, 2019.
13. Daniel Lokshtanov and Amer E. Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Trans. Algorithms*, 15(1):7:1–7:19, 2019.
14. Dániel Marx. Searching the k-change neighborhood for TSP is W[1]-hard. *Oper. Res. Lett.*, 36(1):31–36, 2008.
15. Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984.