Reconfiguring Independent Sets in Claw-Free Graphs

Paul Bonsma[1], Marcin Kamiński[2], and Marcin Wrochna[2,⋆]

[1] University of Twente, Faculty of EEMCS, PO Box 217, 7500 AE Enschede, The Netherlands
p.s.bonsma@ewi.utwente.nl
[2] Uniwersytet Warszawski, Institute of Computer Science, Warsaw, Poland
mjk@mimuw.edu.pl, mw290715@students.mimuw.edu.pl

Abstract. We present a polynomial-time algorithm that, given two independent sets in a claw-free graph G, decides whether one can be transformed into the other by a sequence of elementary steps. Each elementary step is to remove a vertex v from the current independent set S and to add a new vertex w (not in S) such that the result is again an independent set. We also consider the more restricted model where v and w have to be adjacent.

1 Introduction

Reconfiguration Problems. To obtain a reconfiguration version of an algorithmic problem, one defines a *reconfiguration rule* – a (symmetric) adjacency relation between solutions of the problem, describing small transformations one is allowed to make. The main focus is on studying whether one given solution can be transformed into another by a sequence of such small steps. We call this a *reachability problem.* For example, in a well-studied reconfiguration version of vertex coloring [1,2,3,4,5,6], we are given two k-colorings of the vertices of a graph and we should decide whether one can be transformed into the other by recoloring one vertex at a time so that all intermediate solutions are also proper k-colorings.

A useful way to look at reconfiguration problems is through the concept of the *solution graph.* Given a problem instance, the vertices of the solution graph are all solutions to the instance, and the reconfiguration rule defines its edges. Clearly, one solution can be transformed into another if they belong to the same connected component of the solution graph. Other well-studied questions in the context of reconfiguration are as follows: can one efficiently decide (for every instance) whether the solution graph is connected? Can one efficiently find shortest paths between two solutions? Common non-algorithmic results are giving upper and lower bounds on the possible diameter of components of the

⋆ The first author was supported by the European Community's Seventh Framework Programme (FP7/2007-2013), grant agreement n° 317662. The second and third author were supported by the Foundation for Polish Science (HOMING PLUS/2011-4/8) and the National Science Center (SONATA 2012/07/D/ST6/02432).

R Ravi and I.L. Gørtz (Eds.): SWAT 2014, LNCS 8503, pp. 86–97, 2014.

solution graph, in terms of the instance size, or studying how much the solution space needs to be increased in order to guarantee connectivity.

Reconfiguration is a natural setting for real-life problems in which solutions evolve over time and an interesting theoretical framework that has been gradually attracting more attention. The theoretical interest is based on the fact that reconfiguration problems provide a new perspective and offer a deeper understanding of the solution space as well as a potential to develop heuristics to navigate that space.

The reconfiguration paradigm has recently been applied to a number of algorithmic problems: vertex coloring [1,2,3,4,5], list-edge coloring [7], clique, set cover, integer programming, matching, spanning tree, matroid bases [8], block puzzles [9], satisfiability [10], independent set [9,8,11], shortest paths [12,13,14], and dominating set [15]; recently also in the setting of parameterized complexity [16]. A recent survey [17] gives a good introduction to this area of research.

Reconfiguration of Independent Sets. The topic of this paper is reconfiguration of independent sets. An *independent set* in a graph is a set of pairwise nonadjacent vertices. We will view the elements of an independent set as tokens placed on vertices. Three different reconfiguration rules have been studied in the literature: token sliding (TS), token jumping (TJ), and token addition/removal (TAR). The reconfiguration rule in the TS model allows to slide a token along an edge. The reconfiguration rule in the TJ model allows to remove a token from a vertex and place it on another unoccupied vertex. In the TAR model, the reconfiguration rule allows to either add or remove a token as long as at least k tokens remain on the graph at any point, for a given integer k. In all three cases, the reconfiguration rule may of course only be applied if it maintains an independent set. A sequence of moves following these rules is called a *TS-sequence, TJ-sequence*, or *k-TAR-sequence*, respectively. Note that the TS model is more restricted than the TJ model, in the sense that any TS-sequence is also a TJ-sequence. Kamiński et al. [11] showed that the TAR model generalizes the TJ model, in the sense that there exists a TJ-sequence between two solutions I and J with $|I| = |J|$ if and only if there exists a k-TAR-sequence between them, with $k = |I| - 1$. TS seems to have been introduced by Hearn and Demaine [9], TAR was introduced by Ito et al. [8] and TJ by Kamiński et al. [11].

In all three models, the corresponding reachability problems are PSPACE-complete in general graphs [8] and even in perfect graphs [11] or in planar graphs of maximum degree 3 [9] (see also [3]). We remark that in [9], only the TS-model was explicitly considered, but since only maximum independent sets are used, this implies the result for the TJ model (see Proposition 2 below) and for the TAR model (using the aforementioned result from [11]).

Claw-Free Graphs. A *claw* is the tree with four vertices and three leaves. A graph is *claw-free* if it does not contain a claw as an induced subgraph. A claw is not a line graph of any graph and thus the class of claw-free graphs generalizes the class of line graphs. The structure of claw-free graphs is not simple but has been recently described by Chudnovsky and Seymour in the form of a decomposition theorem [18].

There is a natural one-to-one correspondence between matchings in a graph and independent sets in its line graph. In particular, a maximum matching in a graph corresponds to a maximum independent set in its line graph. Hence, Edmonds' maximum matching algorithm [19] gives a polynomial-time algorithm for finding maximum independent sets in line graphs. This results has been extended to claw-free graphs independently by Minty [20] and Sbihi [21]. Both algorithms work for the unweighted case, while the algorithm of Minty, with a correction proposed by Nakamura and Tamura in [22], applies to weighted graphs (see also [23, Section 69]). Recently Nobili and Sassano [24] improved this to give an $\mathcal{O}(n^4 \log n)$ algorithm, while Faenza et al. [25] proved a decomposition theorem that allows to solve the problem in $\mathcal{O}(n^3)$ time.

Our Results. In this paper, we study the reachability problem for independent set reconfiguration, using the TS and TJ model. Our main result is that these problems can be solved in polynomial time for the case of claw-free graphs. Along the way, we prove some results that are interesting in their own right. For instance, we show that for connected claw-free graphs, the existence of a TJ-sequence implies the existence of a TS-sequence between the same pair of solutions. This implies that for connected claw-free and even-hole-free graphs, the solution graph is always connected, answering an open question posed in [11].

Since claw-free graphs generalize line graphs, our results generalize the result by Ito et al. [8] on matching reconfiguration. Since a vertex set I of a graph G is an independent set if and only if $V(G) \backslash I$ is a vertex cover, our results also apply to the recently studied vertex cover reconfiguration problem [16]. The new techniques we introduce can be seen as an extension of the techniques introduced for finding maximum independent sets in claw-free graphs, and we expect them to be useful for addressing similar reconfiguration questions, such as efficiently deciding whether the solution graph is connected.

Because of space constraints, some proof details are omitted. Statements for which more proof details can be found in the full version of this paper [26] are marked with a star.

2 Preliminaries

For graph theoretic terminology not defined here, we refer to [27]. For a graph G and vertex set $S \subseteq V(G)$, we denote the subgraph induced by S by $G[S]$, and denote $G - S = G[V \backslash S]$. The set of neighbors of a vertex $v \in V(G)$ is denoted by $N(v)$, and the closed neighborhood of v is $N[v] = N(v) \cup \{v\}$. A *walk* from v_0 to v_k of length k is a sequence of vertices $v_0, v_1, \ldots, v_k$ such that $v_i v_{i+1} \in E(G)$ for all $i \in \{0, \ldots, k-1\}$. It is a *path* if all of its vertices are distinct, and a *cycle* if $k \geq 3$, $v_0 = v_k$ and $v_0, \ldots, v_{k-1}$ is a path. We use $V(C)$ to denote the vertex set of a path or cycle, viewed as a subgraph of G. A path or graph is called *trivial* if it contains only one vertex. Edges of a directed graph or *digraph* D are called *arcs*, and are denoted by the ordered tuple (u, v). A *directed path* in D is a sequence of distinct vertices $v_0, \ldots, v_k$ such that for all $i \in \{0, \ldots, k-1\}$, (v_i, v_{i+1}) is an arc of D.

We denote the distance of two vertices $u, v \in V(G)$ by $\mathrm{d}_G(u, v)$. By $\mathrm{diam}(G)$ we denote the *diameter* of a connected graph G, defined as $\max_{u,v \in V(G)} \mathrm{d}_G(u, v)$. For a vertex set S of a graph G and integer $i \in \mathbb{N}$, we denote $N_i(S) = \{v \in V(G) \backslash S : |N(v) \cap S| = i\}$.

For a graph G, by $\mathrm{TS}_k(G)$ we denote the graph that has as its vertex the set of all independent sets of G of size k, where two independent sets I and J are adjacent if there is an edge $uv \in E(G)$ with $I \backslash J = \{u\}$ and $J \backslash I = \{v\}$. We say that J can be obtained from I by *sliding a token from* u *to* v, or by the *move* $u \rightarrow v$ for short. A walk in $\mathrm{TS}_k(G)$ from I to J is called a *TS-sequence from* I *to* J. We write $I \leftrightarrow_{\mathrm{TS}} J$ to indicate that there is a TS-sequence from I to J.

Analogously, by $\mathrm{TJ}_k(G)$ we denote the graph that has as its vertex set the set of all independent sets of G of size k, where two independent sets I and J are adjacent if there is a vertex pair $u, v \in V(G)$ with $I \backslash J = \{u\}$ and $J \backslash I = \{v\}$. We say that J can be obtained from I by *jumping a token from* u *to* v. A walk in $\mathrm{TS}_k(G)$ from I to J is called a *TJ-sequence from* I *to* J. We write $I \leftrightarrow_{\mathrm{TJ}} J$ to indicate that there exists a TJ-sequence from I to J. Note that $\mathrm{TS}_k(G)$ is a spanning subgraph of $\mathrm{TJ}_k(G)$.

The *reachability problem* for token sliding (resp. token jumping) has as input a graph G and two independent sets I and J of G with $|I| = |J|$, and asks whether $I \leftrightarrow_{\mathrm{TS}} J$ (resp. $I \leftrightarrow_{\mathrm{TJ}} J$). These problems are called *TS-Reachability* and *TJ-Reachability*, respectively.

If H is a claw with vertex set $\{u, v, w, x\}$ such that $N(u) = \{v, w, x\}$, then H is called a *u-claw with leaves* v, w, x. Sets $I \backslash \{v\}$ and $I \cup \{v\}$ are denoted by $I - v$ and $I + v$ respectively. The symmetric difference of two sets I and J is denoted by $I \Delta J = (I \backslash J) \cup (J \backslash I)$. The following observation is used implicitly in many proofs:

Proposition 1. *Let I and J be independent sets in a claw-free graph G. Then every component of $G[I \Delta J]$ is a path or an even length cycle.*

By $\alpha(G)$ we denote the size of the largest independent set of G. An independent set I is called *maximum* if $|I| = \alpha(G)$. A vertex set $S \subseteq V(G)$ is a *dominating set* if $N[v] \cap S \neq \emptyset$ for all $v \in V(G)$. Observe that a maximum independent set is a dominating set, thus the only possible token jumps from it are between adjacent vertices, and hence all are token slides:

Proposition 2. *Let G be any graph and $k = \alpha(G)$. Then, $TS_k(G) = TJ_k(G)$. In particular, for any two maximum independent sets I and J in G, $I \leftrightarrow_{\mathrm{TS}} J$ if and only if $I \leftrightarrow_{\mathrm{TJ}} J$.*

3 The Equivalence of Sliding and Jumping

In our main result (Theorem 17), we will consider equal size independent sets I and J of a claw-free graph G, and show that in polynomial time, it can be verified whether $I \leftrightarrow_{\mathrm{TS}} J$ and whether $I \leftrightarrow_{\mathrm{TJ}} J$. In this section, we show that if G is connected and $G[I \Delta J]$ contains no cycles, then $I \leftrightarrow_{\mathrm{TS}} J$. From this, we will subsequently conclude that for connected claw-free graphs $I \leftrightarrow_{\mathrm{TS}} J$ holds if and only if $I \leftrightarrow_{\mathrm{TJ}} J$ holds, even in the case of nonmaximum independent sets.

Lemma 3 (*). *Let I and J be independent sets in a connected claw-free graph G with $|I| = |J|$. If $G[I \Delta J]$ contains no cycles, then $I \leftrightarrow_{\mathrm{TS}} J$.*

Proof sketch: We show that I or J can be modified using token slides so that the two resulting independent sets are closer to each other in the sense that either $|I \setminus J|$ is smaller, or it is unchanged and the minimum distance between vertices u, v with $u \in I \setminus J$ and $v \in J \setminus I$ is smaller. The claim then follows by induction.

Suppose first that $G[I \Delta J]$ contains at least one nontrivial component C. Since it is not a cycle by assumption, it must be a path. Choose an end vertex u of this path, and let v be its unique neighbor on the path. If $u \in J$ then $N(u) \cap I = \{v\}$, so we can obtain a new independent set $I' = I + u - v$ from I using a single token slide. The new set I' is closer to J in the sense that $|I' \backslash J| < |I \backslash J|$, so we may use induction to conclude that $I' \leftrightarrow_{\mathrm{TS}} J$, and thus $I \leftrightarrow_{\mathrm{TS}} J$. On the other hand, if $u \in I$ then we can obtain a new independent set $J' = J - v + u$ from J, and conclude the proof similarly by applying the induction assumption to J' and I.

In the remaining case, we may assume that $G[I \Delta J]$ consists only of isolated vertices. Choose $u \in I \backslash J$ and $v \in J \backslash I$, such that the distance $d := \mathrm{d}_G(u, v)$ between these vertices is minimized. Starting with I, we intend to slide the token on u to v, to obtain an independent set $I' = I - u + v$ that is closer to J. To this end, we choose a shortest path $P = v_0, \ldots, v_d$ in G from $v_0 = u$ to $v_d = v$. If the token can be moved along this path while maintaining an independent set throughout, then $I \leftrightarrow_{\mathrm{TS}} I'$, and the proof follows by induction as before.

So now suppose that this cannot be done, that is, at least one of the vertices on P is equal to or adjacent to a vertex in $I - u$. In that case, we choose i maximum such that $N(v_i) \cap I \neq \emptyset$. Using some simple observations (including the fact that G is claw-free), one can now show that $N(v_i) \cap I$ consists of a single vertex x. By choice of v_i, starting with I, the token on x can be moved along the path $x, v_i, v_{i+1}, \ldots, v_d$ while maintaining an independent set throughout. This yields an independent set $I'' = I - x + v$, with $I \leftrightarrow_{\mathrm{TS}} I''$. It can also easily be shown that $\mathrm{d}_G(u, x) < \mathrm{d}_G(u, v)$ and $\mathrm{d}_G(x, v) < \mathrm{d}_G(u, v)$. So considering the choice of u and v, it follows that $x \in I \cap J$, and thus $|I'' \backslash J| = |I \backslash J|$. Since now the pair $u \in I'' \backslash J$ and $x \in J \backslash I''$ has a smaller distance $\mathrm{d}_G(u, x) < \mathrm{d}_G(u, v) = d$, we may assume by induction that $I'' \leftrightarrow_{\mathrm{TS}} J$, and thus $I \leftrightarrow_{\mathrm{TS}} J$. □

Corollary 4. *Let I and J be independent sets in a connected claw-free graph G. Then $I \leftrightarrow_{\mathrm{TS}} J$ if and only if $I \leftrightarrow_{\mathrm{TJ}} J$.*

Proof: Let J be obtained from I by jumping a token from u to v. Then $G[I \Delta J]$ contains only two vertices and therefore no cycles. So by Lemma 3, any token jump can be replaced by a sequence of token slides. □

We now consider implications of the above corollary for graphs that are claw- and even-hole-free. A graph is *even-hole-free* if it contains no even cycle as an induced subgraph. Kamiński et al. [11] proved the following statement.

Theorem 5 ([11]). *Let I and J be two independent sets of a graph G with $|I| = |J|$. If $G[I \Delta J]$ contains no even cycles, then there exists a TJ-sequence from I to J of length $|I \backslash J|$, which can be constructed in linear time.*

In particular, if G is even-hole-free, then $\mathrm{TJ}_k(G)$ is connected (for every k). However, $\mathrm{TS}_k(G)$ is not necessarily connected (consider a claw with two tokens). This motivated the question asked in [11] whether for connected, claw-free and even-hole-free graph G, $\mathrm{TS}_k(G)$ is connected. Combining Corollary 4 with Theorem 5 shows that the answer to this question is affirmative.

Corollary 6. *Let G be a connected claw-free and even-hole-free graph. Then $TS_k(G)$ is connected.*

4 Nonmaximum Independent Sets

We now continue studying connected claw-free graphs. By Lemma 3 it remains to consider the case that $G[I \Delta J]$ contains (even length) cycles. In this section, we show that if I and J are not maximum independent sets of G, such cycles can always be resolved. This requires various techniques developed for finding maximum independent sets in claw-free graphs, and the following definitions.

A vertex $v \in V(G)$ is *free* (with respect to an independent set I of G) if $v \notin I$ and $|N(v) \cap I| \leq 1$. Let $W = v_0, \ldots, v_k$ be a walk in G, and let $I \subseteq V(G)$. Then W is called *I-alternating* if $|\{v_i, v_{i+1}\} \cap I| = 1$ for $i = 0, \ldots, k-1$. In the case that W is a path, W is called *chordless* if $G[\{v_0, \ldots, v_k\}]$ is a path. In the case that W is a cycle (so $v_0 = v_k$), W is called *chordless* if $G[\{v_0, \ldots, v_{k-1}\}]$ is a cycle. A cycle $W = v_0, \ldots, v_k$ is called *I-bad* if it is I-alternating and chordless. A path $W = v_0, \ldots, v_k$ with $k \geq 2$ is called *I-augmenting* if it is I-alternating and chordless, and v_0 and v_k are both free vertices. This definition of I-augmenting paths differs from the usual definition, as it is used in the setting of finding *maximum independent sets*, since the chordless condition is stronger than needed in such a setting. However, we observe that in a claw-free graph G, the two definitions are equivalent, so we may apply well-known statements about I-augmenting paths proved elsewhere. In particular, we use the following two results originally proved by Minty [20] and Sbihi [21] (see also [23, Section 69.2]).

Theorem 7 ([23]). *Let I be an independent set in a claw-free graph G. In polynomial time, it can be decided whether an I-augmenting path between two given free vertices x and y exists, and if so, one can be computed.*

Proposition 8 ([23]). *Let I be a nonmaximum independent set in a claw-free graph G. Then I is not a dominating set, or there exists an I-augmenting path.*

We use Proposition 8 to handle the case of nonmaximum independent sets. The next statement is formulated for token jumping, and (by Corollary 4) implies the same result for token sliding, in the case of connected graphs.

Lemma 9 (*). *Let I be a nonmaximum independent set in a claw-free graph G. Then for any independent set J with $|J| = |I|$, $I \leftrightarrow_{\mathrm{TJ}} J$ holds.*

Proof sketch: By Theorem 5, it suffices to consider the case where $G[I \Delta J]$ contains at least one cycle C. Let $C = u_1, v_1, u_2, v_2, \ldots, v_k, u_1$, so that $u_i \in I$ and $v_i \in J$ for all i.

Suppose first that I is not a dominating set. Then we can choose a vertex w with $N[w] \cap I = \emptyset$. With a single token jump, we can obtain the independent set $I' = I + w - u_1$ from I. Next, apply the moves $u_k \to v_k$, $u_{k-1} \to v_{k-1}, \dots,$ $u_2 \to v_2$, in this order. (This is possible since C is chordless.) Finally, jump the token from w to v_1. It can be verified that this yields a token jumping sequence from I to $I' = I \Delta V(C)$. This way, all cycles can be resolved one by one, until no more cycles remain and Theorem 5 can be applied to prove the statement.

On the other hand, if I is a dominating set, then Proposition 8 shows that there exists an I-augmenting path $P = v_0, u_1, v_1, \dots, u_d, v_d$, with $u_i \in I$ for all i. Since v_d is a free vertex, we can first apply the moves $u_d \to v_d$, $u_{d-1} \to$ $v_{d-1}, \dots u_1 \to v_1$, in this order (which can be done since P is chordless), to obtain an independent set I' from I, with $I \leftrightarrow_{\mathrm{TS}} I'$. Then v_0 is not dominated by I', so the previous argument shows that $I' \leftrightarrow_{\mathrm{TJ}} J$, which implies $I \leftrightarrow_{\mathrm{TJ}} J$. □

5 Resolving Cycles

It now remains to study the case where $G[I \Delta J]$ contains (even) cycles and both I and J are maximum independent sets. In this case, there may not be a TS-sequence from I to J (even though we assume that G is connected and claw-free) – consider for instance the case where G itself is an even cycle. In this section, we characterize the case where $I \leftrightarrow_{\mathrm{TS}} J$ holds, by showing that this is equivalent with every cycle being resolvable in a certain sense (Theorem 11 below). Subsequently, we show that resolvable cycles fall into two cases: internally or externally resolvable cycles, which are characterized next. We first define the notion of resolving a cycle.

Cycles in $G[I \Delta J]$ are clearly both I-bad and J-bad. The *I-bipartition* of an I-bad cycle is the ordered tuple $[V(C) \cap I, V(C) \backslash I]$. We say that an I-bad cycle C with I-bipartition $[A, B]$ is *resolvable* (with respect to I) if there exists an independent set I' such that $I \leftrightarrow_{\mathrm{TS}} I'$ and $G[I' \cup B]$ contains no cycles. A corresponding TS-sequence from I to I' is called a *resolving sequence* and is said to *resolve* C. By combining such a resolving sequence with a sequence of moves similar to the previous proof, and then reversing the moves in the sequence from I' to I, except for moves of tokens on the cycle, one can show that every resolvable cycle can be 'turned':

Lemma 10 (*). *Let I be an independent set in a claw-free graph G and let C be an I-bad cycle. If C is resolvable with respect to I, then $I \leftrightarrow_{\mathrm{TS}} I \Delta V(C)$.*

We can now prove the following useful characterization: $I \leftrightarrow_{\mathrm{TS}} J$ if and only if every cycle in $G[I \Delta J]$ is resolvable. By symmetry, it does not matter whether one considers resolvability with respect to I or to J.

Theorem 11. *Let I and J be independent sets in a claw-free connected graph G. Then $I \leftrightarrow_{\mathrm{TS}} J$ if and only if every cycle in $G[I \Delta J]$ is resolvable with respect to I.*

Proof: Consider an I-bad cycle C in $G[I\Delta J]$ with I-bipartition $[A, B]$, and a TS-sequence from I to J. Since $N_2(B)$ eventually contains no tokens, this sequence must contain a move $u \to v$ with $u \in N_2(B)$ and $v \notin N_2(B)$. The first such move can be shown to resolve the cycle.

The other direction is proved by induction on the number k of cycles in $G[I\Delta J]$. If $k = 0$, then by Lemma 3, $I \leftrightarrow_{\text{TS}} J$. If $k \geq 1$, then consider an I-bad cycle C in $G[I\Delta J]$. Let $I' = I\Delta V(C)$. By Lemma 10, $I \leftrightarrow_{\text{TS}} I'$. The graph $G[I'\Delta J]$ has one cycle fewer than $G[I\Delta J]$. Every cycle in $G[I'\Delta J]$ remains resolvable with respect to I' (one can first consider a TS-sequence from I' to I, and subsequently a TS-sequence from I that resolves the cycle). So by induction, $I' \leftrightarrow_{\text{TS}} J$, and therefore, $I \leftrightarrow_{\text{TS}} J$. □

Finally, we show that if an I-bad cycle C can be resolved, it can be resolved in at least one of two very specific ways. Let $[A, B]$ be the I-bipartition of C. A move $u \to v$ is called *internal* if $\{u, v\} \subseteq N_2(B)$ and *external* if $\{u, v\} \subseteq N_0(B)$. A resolving sequence $I_0, \ldots, I_m$ for C is called *internal* (or *external*) if every move except the last is an internal (respectively, external) move. (Obviously, to resolve the cycle, the last move can neither be internal nor external, and can in fact be shown to always be a move from $N_2(B)$ to $N_1(B)$.) The I-bad cycle C is called *internally resolvable* resp. *externally resolvable* if such sequences exist.

Lemma 12 (*). *Let I be an independent set in a claw-free graph G and let C be an I-bad cycle. Then any shortest TS-sequence that resolves C is an internal or external resolving sequence.*

Proof sketch: Let $[A, B]$ be the I-bipartition of C. Since G is claw-free, it follows that there are no edges between vertices in $N_2(B)$ and $N_0(B)$. This can be used to show that informally, any resolving sequence for C remains a resolving sequence after either omitting all noninternal moves or omitting all nonexternal moves, while keeping the last move, which subsequently resolves the cycle. □

Theorem 11 and Lemma 12 show that to decide whether $I \leftrightarrow_{\text{TS}} J$, it suffices to check whether every cycle in $G[I\Delta J]$ is externally or internally resolvable. Next we give characterizations that allow polynomial-time algorithms for deciding whether an I-bad cycle is internally or externally resolvable. For the external case, we use the assumption that I is a maximum independent set to show that in a *shortest* external resolving sequence $I_0, \ldots, I_m$, every token moves at most once (that is, for every move $u \to v$, both $u \in I_0$ and $v \in I_m$ hold), so these moves outline an augmenting path in a certain auxiliary graph.

Theorem 13 (*). *Let I be a maximum independent set in a claw-free graph G and let C be an I-bad cycle with I-bipartition $[A, B]$. Then C is externally resolvable if and only if there exists an $(I \backslash A)$-augmenting path in $G - A - B$ between a pair of vertices $x \in N_0(B)$ and $y \in N_1(B)$.*

For a given I-bad cycle C with I-bipartition $[A, B]$, there is a quadratic number of vertex pairs $x \in N_0(B)$ and $y \in N_1(B)$ that need to be considered, and

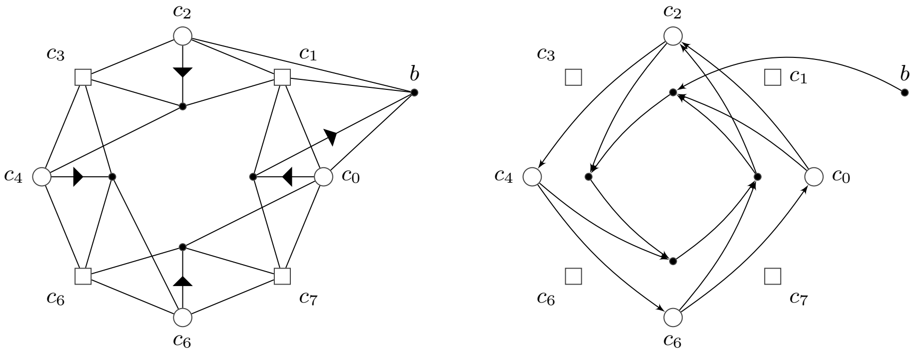

Fig. 1. An example of a claw-free graph G with an internally resolvable cycle, along with the corresponding auxiliary digraph $D(G, C)$.

for every such a pair, testing whether there is an $(I \setminus A)$-augmenting path between these in $G - A - B$ can be done in polynomial time (Theorem 7). So from Theorem 13 we conclude:

Corollary 14. *Let I be a maximum independent set in a claw-free graph G, and let C be an I-bad cycle. In polynomial time, it can be decided whether C is externally resolvable.*

Next, we characterize internally resolvable cycles. Shortest internal resolving sequences cannot be as easy to describe as external ones, since a token can move several times (see Figure 1). Nevertheless, these sequences can be shown to have a very specific structure, which can be characterized using paths in the following auxiliary digraphs.

To define these digraphs, consider an I-bad cycle $C = c_0, c_1, \ldots, c_{2n-1}, c_0$ in G, with $c_i \in I$ for even i. Let $[A, B]$ be the I-bipartition of C. For every $i \in \{0, \ldots, n-1\}$, define the corresponding *layer* as follows: $L_i = \{v \in V(G) \mid N(v) \cap B = N(c_{2i}) \cap B\}$. So when starting with I and using only internal moves, it can be seen that the token that starts on c_{2i} will stay in the layer L_i.

For such an I-bad cycle C of length at least 8, define $D(G, C)$ to be a digraph with vertex set $V(G)$, with the following arc set. For every $i \in \{0, \ldots, n-1\}$ and all pairs $u \in L_i, v \in L_{(i+1) \bmod n}$ with $uv \notin E(G)$, add an arc (u, v). For every $i \in \{0, \ldots, n-1\}$ and $b \in N_1(B)$ with $N(b) \cap B = \{c_{(2i-1) \bmod 2n}\}$, and every $v \in L_i$ with $bv \notin E(G)$, add an arc (b, v). We denote the reversed cycle by $C^{rev} = c_0, c_{2n-1}, \ldots, c_1, c_0$. This defines a similar digraph $D(G, C^{rev})$ (where arcs between layers are reversed, and arcs from $N_1(B)$ go to different layers). These graphs can be used to characterize whether C is internally resolvable.

Theorem 15 (*). *Let I be an independent set in a claw-free graph G. Let $C = c_0, c_1, \ldots, c_{2n-1}, c_0$ be an I-bad cycle ($c_0 \in I$) with I-bipartition $[A, B]$, of length at least 8. Then C is internally resolvable if and only if $D(G, C)$ or $D(G, C^{rev})$ contains a directed path from a vertex $b \in N_1(B)$ with $N(b) \cap I \subseteq A$ to a vertex in A.*

Corollary 16. *Let I be an independent set in a claw-free graph G on n vertices and let C be an I-bad cycle. It can be decided in polynomial time whether C is internally resolvable.*

Proof: If C has length at least 8, then Theorem 15 shows that it suffices to make a polynomial number of depth-first-searches in $D(G, C)$ and $D(G, C^{rev})$. Otherwise, let $[A, B]$ be the I-bipartition of C. $|A| \leq 3$, so there are only $\mathcal{O}(n^3)$ independent sets I' with $|I'| = |I|$ and $I \backslash A \subseteq I'$. So in polynomial time we can generate the subgraph of $\mathrm{TS}_k(G)$ induced by these sets, and search whether it contains a path from I to an independent set I^* with $I \backslash A \subseteq I^*$ where $G[B \cup I^*]$ contains no cycle. C is internally resolvable if and only if such a path exists. □

6 Summary of the Algorithm

We now summarize how the previous lemmas yield a polynomial time algorithm for TS-Reachability and TJ-Reachability in claw-free graphs.

Theorem 17. *Let I and J be independent sets in a claw-free graph G. We can decide in polynomial time whether $I \leftrightarrow_{\mathrm{TS}} J$ and whether $I \leftrightarrow_{\mathrm{TJ}} J$.*

Proof: Assume $|I| = |J|$; otherwise, we immediately return NO. We first consider the case when G is connected. By Corollary 4, since G is connected, $I \leftrightarrow_{\mathrm{TS}} J$ if and only if $I \leftrightarrow_{\mathrm{TJ}} J$, thus we only need to consider the sliding model.

We test whether I and J are maximum independent sets of G, which can be done in polynomial time (by combining Proposition 8 and Theorem 7; see also [20,21,23]). If not, then by Lemma 9, $I \leftrightarrow_{\mathrm{TJ}} J$ holds, and thus $I \leftrightarrow_{\mathrm{TS}} J$, so we may return YES.

Now consider the case that both I and J are maximum independent sets. Theorem 11 shows that $I \leftrightarrow_{\mathrm{TS}} J$ if and only if every cycle in $G[I \Delta J]$ is resolvable with respect to I. By Lemma 12, it suffices to check for internal and external resolvability of such cycles. This can be done in polynomial time by Corollary 14 (since I is a maximum independent set of G) and Corollary 16. We return YES if and only if every cycle in C was found to be internally or externally resolvable.

Now let us consider the case when G is disconnected. Clearly tokens cannot slide between different connected components, so for deciding whether $I \leftrightarrow_{\mathrm{TS}} J$, we can apply the argument above to every component, and return YES if and only if the answer is YES for every component. If I is a not a maximum independent set then Lemma 9 shows that $I \leftrightarrow_{\mathrm{TJ}} J$ always holds. If I is maximum, then Proposition 2 shows that $I \leftrightarrow_{\mathrm{TJ}} J$ holds if and only if $I \leftrightarrow_{\mathrm{TS}} J$. □

7 Discussion

The results presented here have two further implications. Firstly, combined with techniques from [28], it follows that $I \leftrightarrow_{\mathrm{TJ}} J$ can be decided for any graph G

that can be obtained from a collection of claw-free graphs using *disjoint union* and *complete join* operations. See [28] for more details.

Secondly, a closer look at constructed reconfiguration sequences shows that when G is claw-free, components of both $\mathrm{TS}_k(G)$ and $\mathrm{TJ}_k(G)$ have diameter bounded polynomially in $|V(G)|$. This is not surprising, since the same behavior has been observed many times. To our knowledge, the only known examples of polynomial time solvable reconfiguration problems that nevertheless require exponentially long reconfiguration sequences are on artificial instance classes, which are constructed particularly for this purpose (see e.g. [3,14]).

References

1. Bonamy, M., Bousquet, N.: Recoloring bounded treewidth graphs. Electronic Notes in Discrete Mathematics 44, 257–262 (2013)
2. Bonamy, M., Johnson, M., Lignos, I., Patel, V., Paulusma, D.: Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. Journal of Combinatorial Optimization 27(1), 132–143 (2014)
3. Bonsma, P., Cereceda, L.: Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. Theor. Comput. Sci. 410(50), 5215–5226 (2009)
4. Cereceda, L., van den Heuvel, J., Johnson, M.: Connectedness of the graph of vertex-colourings. Discrete Math. 308(5-6), 913–919 (2008)
5. Cereceda, L., van den Heuvel, J., Johnson, M.: Mixing 3-colourings in bipartite graphs. European J. of Combinatorics 30(7), 1593–1606 (2009)
6. Ito, T., Kawamura, K., Ono, H., Zhou, X.: Reconfiguration of list L(2, 1)-labelings in a graph. In: Chao, K.-M., Hsu, T.-S., Lee, D.-T. (eds.) ISAAC 2012. LNCS, vol. 7676, pp. 34–43. Springer, Heidelberg (2012)
7. Ito, T., Kamiński, M., Demaine, E.D.: Reconfiguration of list edge-colorings in a graph. In: Dehne, F., Gavrilova, M., Sack, J.-R., Tóth, C.D. (eds.) WADS 2009. LNCS, vol. 5664, pp. 375–386. Springer, Heidelberg (2009)
8. Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theoret. Comput. Sci. 412(12-14), 1054–1065 (2011)
9. Hearn, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. Theor. Comput. Sci. 343(1-2), 72–96 (2005)
10. Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of Boolean satisfiability: Computational and structural dichotomies. SIAM J. Comput. 38(6), 2330–2355 (2009)
11. Kamiński, M., Medvedev, P., Milanič, M.: Complexity of independent set reconfigurability problems. Theor. Comput. Sci. 439, 9–15 (2012)
12. Bonsma, P.: Rerouting shortest paths in planar graphs. In: D'Souza, D., Kavitha, T., Radhakrishnan, J. (eds.) FSTTCS. LIPIcs, vol. 18, pp. 337–349. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2012)
13. Bonsma, P.: The complexity of rerouting shortest paths. Theor. Comput. Sci. 510, 1–12 (2013)
14. Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths. Theor. Comput. Sci. 412(39), 5205–5210 (2011)

15. Suzuki, A., Mouawad, A.E., Nishimura, N.: Reconfiguration of dominating sets. CoRR abs/1401.5714 (2014)
16. Mouawad, A.E., Nishimura, N., Raman, V., Simjour, N., Suzuki, A.: On the parameterized complexity of reconfiguration problems. In: Gutin, G., Szeider, S. (eds.) IPEC 2013. LNCS, vol. 8246, pp. 281–294. Springer, Heidelberg (2013)
17. van den Heuvel, J.: The complexity of change. Surveys in Combinatorics, 127–160 (2013)
18. Chudnovsky, M., Seymour, P.D.: The structure of claw-free graphs. In: Webb, B.S. (ed.) Surveys in Combinatorics. London Mathematical Society Lecture Note Series, vol. 327, pp. 153–171. Cambridge University Press (2005)
19. Edmonds, J.: Paths, trees, and flowers. Canad. J. Math. 17, 449–467 (1965)
20. Minty, G.J.: On maximal independent sets of vertices in claw-free graphs. J. Comb. Theory, Ser. B 28(3), 284–304 (1980)
21. Sbihi, N.: Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. Discrete Mathematics 29(1), 53–76 (1980)
22. Nakamura, D., Tamura, A.: A revision of Minty's algorithm for finding a maximum weight stable set of a claw-free graph. Journal of the Operations Research Society of Japan 44(2), 194–204 (2001)
23. Schrijver, A.: Combinatorial optimization: Polyhedra and efficiency, vol. 24. Springer (2003)
24. Nobili, P., Sassano, A.: A reduction algorithm for the weighted stable set problem in claw-free graphs. Discrete Applied Mathematics (2013)
25. Faenza, Y., Oriolo, G., Stauffer, G.: An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$-algorithm for the weighted stable set problem. In: SODA, pp. 630–646. SIAM (2011)
26. Bonsma, P., Kamiński, M., Wrochna, M.: Reconfiguring independent sets in claw-free graphs. CoRR abs/1403.0359 (2014)
27. Diestel, R.: Graph Theory. Electronic Edition. Springer-Verlag (2005)
28. Bonsma, P.: Independent set reconfiguration in cographs. CoRR abs/1402.1587 (2014); Extended abstract accepted for WG 2014