博士学位論文要約（平成31年３月）

グラフ彩色，準同型，制約充足の遷移問題に関する研究

畑中 達彦
指導教員: 周 暁，学位論文指導教員: 伊藤 健洋

Reconfiguration Problems for Graph Coloring, Homomorphism, and Constraint Satisfaction

Tatsuhiko HATANAKA
Supervisor: Xiao ZHOU, Research Advisor: Takehiro ITO

The framework of (combinatorial) reconfiguration models several “dynamic” situations, where we wish to find a step-by-step transformation between two feasible solutions of a combinatorial search problem such that all intermediate solutions are also feasible and each step respects a fixed reconfiguration rule. In this thesis, we mainly study the reconfiguration problem for the well-known constraint satisfaction problem (CSP), which is a generalization of several combinatorial search problems including graph coloring, Boolean satisfiability, graph homomorphism, and so on. In the reconfiguration problem for CSP, we are given an instance of CSP together with its two satisfying assignments, and asked to determine whether one assignment can be transformed into the other by changing a single variable assignment at a time, while always remaining satisfying assignment. We also study several special cases of the problem, especially the reconfiguration problems for graph coloring, graph homomorphism, and their list variants. In this thesis, we study these problems from the viewpoints of polynomial-time solvability and parameterized complexity, and give several interesting boundaries of tractable and intractable cases.

1. Introduction

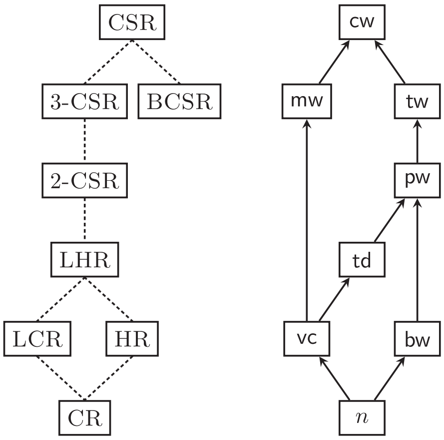
Since the 2000s, the framework of (*combinatorial*) *reconfiguration*[5) 6)] has been extensively studied in the field of theoretical computer science. This framework models several “dynamic” situations where we wish to find a step-by-step transformation between two feasible states such that all intermediate states are also feasible and each step respects a fixed reconfiguration rule. Generally, a (*combinatorial*) *reconfiguration problem* can be considered as a problem asking the reachability of two vertices in a *solution graph* defined as follows. The vertex set of a solution graph corresponds to the set of feasible states and the edge set represents a reconfiguration rule. The set of feasible states may often be defined as a set of feasible solutions for an instance of a (combinatorial) search problem. Indeed, several reconfiguration problems based on search problems are studied well, such as Boolean satisfiability reconfiguration, shortest path reconfiguration, independent set reconfiguration, vertex cover reconfiguration, and dominating set reconfiguration.

1.1 Our problems

In this thesis, we mainly study constraint satisfiability reconfiguration, and its special cases including (list) coloring reconfiguration and (list) homomorphism reconfiguration from the viewpoints of polynomial-time solvability and parameterized complexity. Due to the page limitation, we only give an informal definition of constraint satisfiability reconfiguration, which is a reconfiguration problem for the well-known constraint satisfaction problem (CSP, for short).

Let $G = (V, E)$ be a hypergraph. Let D be a set, called a *domain*; each element of D is called a *value* and we always denote by k the size of a domain. In CSP, each hyperedge $X \in E$ has a *constraint* which represents the values allowed to be assigned to the vertices in X at the same time, and we wish to find a mapping $f\colon V \to D$ which satisfies the constraints of all hyperedges in G. In constraint satisfiability reconfiguration, the vertex set of a solution graph is the set of all mappings satisfying all constraints, and two solutions (vertices in the solution graph) are adjacent if the one is obtained from the other by changing a value of a single vertex in G at a time. Then, for a given hypergraph G, a domain D of size k, a constraint for each hyperedge X, and two solutions f_s and f_t, the problem asks whether there exists a walk between f_s and f_t in the corresponding solution graph; such a walk is called a *reconfiguration sequence*. In the remainder of the thesis, we use the following abbreviations for problems we deal with:

- CSR for (original) constraint satisfiabil-



(a)　(b)

Figure 1. (a) Relationships between problems. Each dotted line between P (lower) and Q (upper) means that P is a special case of Q. (b) Relationships between graph parameters. cw, mw, tw, pw, td, vc, bw and n are the cliquewidth, the modular-width, the treewidth, the pathwidth, the tree-depth, the size of a minimum vertex cover, the bandwidth and the number of vertices of a graph, respectively. Each arrow $\alpha \to \beta$ means that α is stronger than β, that is, if α is bounded by a constant then β is also bounded by some constant.

ITY RECONFIGURATION;

- BCSR for BOOLEAN CONSTRAINT SATISFIABILITY RECONFIGURATION;
- r-CSR for r-ARY CONSTRAINT SATISFIABILITY RECONFIGURATION for each integer $r \geq 1$;
- (L)HR for (LIST) HOMOMORPHISM RECONFIGURATION; and
- (L)CR for (LIST) COLORING RECONFIGURATION.

Relationships between problems are illustrated in Figure 1(a).

In particular, COLORING RECONFIGURATION (CR) can be defined as follows. Let $G = (V, E)$ be a graph and let C be a set of k *colors*. A *k-coloring* (or simply a *vertex coloring*) of G is a mapping $f \colon V \to C$ such that $f(v) \neq f(w)$ holds for every edge $vw \in E$. In CR, the vertex set of a solution graph is the set of all k-colorings of G, and two k-colorings f and f' are adjacent in the solution graph if f' is obtained from f by changing the color assignment on a single vertex, and vice versa. Then, for a given graph G, a color set C of size k, and two k-colorings f_s and f_t of G, the problem CR asks whether there exists a walk between f_s and f_t in the corresponding solution graph. (See Figure 2 for example.) For an integer $k \geq 1$, we define k-CR as a special case of CR where $|C| = k$.

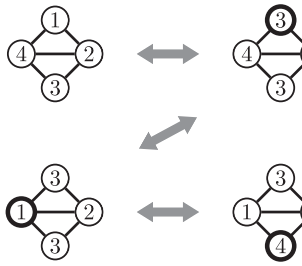


Figure 2. A reconfiguration sequence of 4-colorings. A vertex which is recolored from the immediate left 4-coloring is depicted by a thick circle.

1.2 Our contribution

In this thesis, we investigate the complexity of CSR and its spacial cases, especially 3-CSR, 2-CSR, (L)HR and (L)CR, from the viewpoints of polynomial-time solvability and parameterized complexity, and give several interesting boundaries of tractable and intractable cases.

1.2.1 Polynomial-time solvability

We first classify the complexity of the problems for each fixed size k of a domain in a given instance of CSR; recall that k corresponds to the number of colors in (L)CR. Together with known results, our results give interesting boundaries of (in)tractability as summarized in Table 1.

In order to give more detailed analyses, we also focus on the structure of an input (hyper)graph, and explore the structures which make the problems hard. We first analyze the complexities of CR and LCR from the viewpoint of graph classes. (See Figures 3 and 4.) In particular, the PSPACE-completeness of CR for chordal graphs answers the open question posed by Bonsma and Paulusma[2]. Moreover, we show the boundary of the complexity of LCR with respect to pathwidth; we give a polynomial-time algorithm for graphs with pathwidth one, while it is PSPACE-complete for graphs with pathwidth two[8]. We next investigate the complexity of more general problems, that is, (L)HR, 2-CSR, 3-CSR and CSR, for graphs with pathwidth one or two. (See Table 2.)

1.2.2 Parameterized complexity

We first show that HR parameterized by the number of vertices and LCR parameterized by the size of a minimum vertex cover are both $W[1]$-hard.

Table 1. Computational complexities with respect to the size k of a domain.

	$k \geq 4$	$k = 3$	$k = 2$
CSR	PSPACE-c.	PSPACE-c.	PSPACE-c.
3-CSR	PSPACE-c.	PSPACE-c.	PSPACE-c. [4]
2-CSR	PSPACE-c.	PSPACE-c. [Ours]	P [Ours]
LHR	PSPACE-c.	P [Ours]	P
LCR	PSPACE-c.	P [3]	P
HR	PSPACE-c.	P [7]	P
CR	PSPACE-c. [1]	P	P

Table 2. Computational complexity for graphs with pathwidth at most two.

	pw $= 2$	pw $= 1$
CSR	PSPACE-c.	PSPACE-c.
3-CSR	PSPACE-c.	PSPACE-c.
2-CSR	PSPACE-c.	PSPACE-c.
LHR	PSPACE-c.	PSPACE-c. [Ours]
LCR	PSPACE-c. [8]	P [Ours]
HR	PSPACE-c. [8]	P [8]
CR	P [Ours]	P

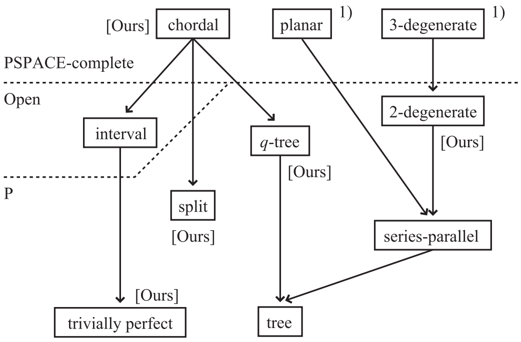


Figure 3. Known and our results for CR with respect to graph classes. Each arrow $A \to B$ represents that the graph class B is a subclass of the graph class A.

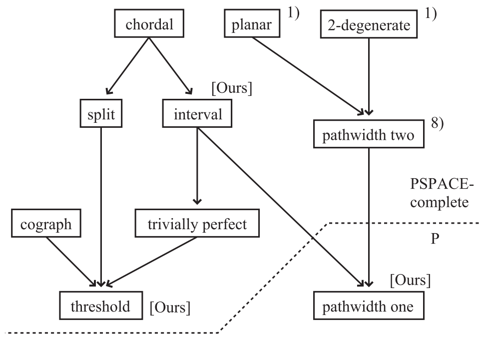


Figure 4. Known and our results for LCR with respect to graph classes.

These imply that fixed-parameter algorithms are unlikely to exist for almost all graph parameters in Figure 1(b); note that tractability (resp., intractability) result propagates downward (resp., upward). Therefore, we take as parameters k plus several graph parameters, and summarize the known and our results in Table 3.

We next consider another parameter which is not graph structural, that is, the number nb of "non-Boolean" vertices, in order to extend the analysis for $k = 2$ presented in the previous subsection. Roughly speaking, nb reflects how an instance is close to that of BCSR. The parameterized complexity regarding nb is summarized in Table 3.

Finally, we prove that 2-CSR cannot be solved in time $O^*((k+n)^{o(k+n)})$ under the exponential time hypothesis (ETH). This lower bound matches the running times of some of our algorithms.

Due to the page limitation, we omit almost all theorems and proofs, and give only one result in the next section.

2. PSPACE-completeness of CR

In this section, we prove the following theorem.

Theorem 1 *There exists a fixed constant k' such that k-CR is PSPACE-complete for chordal graphs and every $k \geq k'$.*

It is known that k-CR belongs to the complexity class PSPACE[1]. Therefore, as a proof of Theorem 1, we show that there exists a fixed constant k' such that k-CR is PSPACE-hard for chordal graphs and any $k \geq k'$, by giving a polynomial-time reduction from LCR, which is defined as follows. Let $G = (V, E)$ be a graph and let C be a set of k colors. Assume that each vertex $v \in V$ has a *list* $L(v) \subseteq C$ (*of* v). A k-coloring $f\colon V \to C$ of G is called an *L-coloring* (or a *list coloring*) if $f(v) \in L(v)$ holds for every vertex $v \in V$. Then, LCR is the reconfiguration problem in which the vertex set of a solution graph is the set of all L-colorings of G, and the edge set is the same as CR.

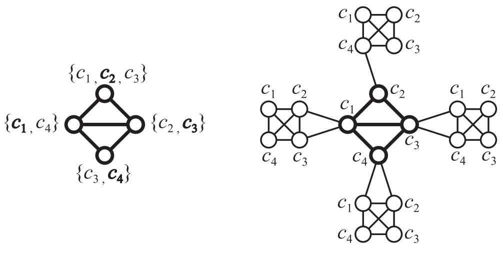
By modifying the proof of the PSPACE-completeness of LCR by Wrochna[8], we can prove the following theorem.

Theorem 2 *There exists a constant b such that* LCR *is PSPACE-complete for interval graphs even when each list is a subset of b colors.*

We then construct an instance (G, f_s, f_t) of k-

Table 3. Parameterized complexity with respect to k, graph parameters, and the number nb of non-Boolean vertices.

Parameter	$k+$ mw	$k+$ td	$k+$ vc	$k+$ bw	$k+$ nb	nb
CSR	PSPACE-c.	FPT [Ours]	FPT [Ours]	PSPACE-c.	PSPACE-c.	PSPACE-c.
3-CSR	PSPACE-c.	FPT	FPT	PSPACE-c.	PSPACE-c. [4]	PSPACE-c.
2-CSR	PSPACE-c. [Ours]	FPT	FPT	PSPACE-c.	FPT [Ours]	$W[1]$-hard, XP [Ours]
LHR	FPT [Ours]	FPT	FPT	PSPACE-c.	FPT	$W[1]$-hard, XP
LCR	FPT	FPT	FPT	PSPACE-c.	FPT	$W[1]$-hard, XP
HR	FPT	FPT [8]	FPT	PSPACE-c.	FPT	$W[1]$-hard [Ours], XP
CR	FPT	FPT	FPT	PSPACE-c. [8]	FPT	FPT [Ours]



(a) H', L, g_s　　(b) G, f_s

Figure 5. (a) A graph H', a list L and an L-coloring g_s, and (b) the constructed graph G and the k-coloring f_s.

CR from the instance (H', L, g_s, g_t) of LCR in Theorem 2.

Let $k' = |\bigcup_{u \in V(H')} L(u)| = b$ and let k be any integer at least k'. For each vertex $u \in V(H')$, we introduce a complete graph W_u of k vertices, which is called a *frozen clique gadget*. The vertices in W_u are labeled as $w_1^u, w_2^u, \ldots, w_k^u$, and each vertex w_i^u corresponds to the color c_i for each $i \in \{1, 2, \ldots, k\}$. Let $W = \bigcup_{u \in V(H')} V(W_u)$. We next add an edge between $u \in V(H')$ and $w_i^u \in V(W_u)$ if $L(u)$ does *not* contain color c_i. Note that G is chordal, because H' is chordal and the addition of frozen clique gadgets does not produce any induced cycle of length at least four.

Finally, we define f_s (resp., f_t) as a mapping obtained by extending g_s (resp., g_t) so that each vertex in W has its corresponding color. This completes the construction of the corresponding instance (G, f_s, f_t) of k-CR. This construction can be done in polynomial time.

From the construction, (H', L, g_s, g_t) is a yes-instance of LCR if and only if the corresponding instance (G, f_s, f_t) of k-CR is a yes-instance. Thus, this completes our proof of Theorem 1.

References

1) P. Bonsma and L. Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009.
2) P. Bonsma and D. Paulusma. Using contracted solution graphs for solving reconfiguration problems. In *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:15, 2016.
3) L. Cereceda, J. van den Heuvel, and M. Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011.
4) P. Gopalan, P. G. Kolaitis, E. N. Maneva, and C. H. Papadimitriou. The connectivity of Boolean satisfiability: Computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009.
5) T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12):1054–1065, 2011.
6) N. Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.
7) M. Wrochna. Homomorphism reconfiguration via homotopy. In *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 730–742, 2015.
8) M. Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018.