2版

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　３ 年　 ６ 月 １７ 日現在

機関番号：１７１０４
研究種目：研究活動スタート支援
研究期間：2019～2020
課題番号：１９Ｋ２４３４９

研究課題名（和文）A study on reconfiguration problems under Token Sliding and their applications

研究課題名（英文）A study on reconfiguration problems under Token Sliding and their applications

研究代表者

ＤｕｃＡ.　Ｈｏａｎｇ（DucA., Hoang）

九州工業大学・大学院情報工学研究院・博士研究員

研究者番号：００８４７８２４

交付決定額（研究期間全体）：（直接経費）　　2,200,000円

研究成果の概要（和文）：トークンスライディング（TS）遷移問題では、各配置はグラフGのいくつかの頂点にトークンが置かれます。２つのトークン集合が隣接するとは、占有された頂点からへの１つのトークンを隣の頂点へスライドさせることを言う。 この研究では、TSおよび他のいくつかのルールの下でこの問題のいくつかの遷移問題の変種に関して研究を開始しました。 各トークンセットは、Gのkパス頂点被覆（k-PVC）を形成します。つまり、k頂点の各パスには、Gに少なくとも1つのトークンがあります。さまざまなグラフクラスで、解くことが簡単か，もしくは難しいかを証明することに成功しました。

研究成果の学術的意義や社会的意義
When designing certain networks, one needs to put a "secured" device on each path on k vertices of the communication graph. Our results may be useful in situations where one needs to reconfigure the networks without changing its security.

研究成果の概要（英文）：In a Token Sliding (TS) reconfiguration problem, each configuration is a set of tokens placed on vertices of a graph G, and two token-sets are adjacent if one can be obtained from the other via a single token-slide from an occupied vertex to one of its neighbors. In this research, we initiated the study of some reconfiguration variants of this problem under TS and some other rules, where each token-set forms a k-path vertex cover (k-PVC) of G, i.e., each path on k vertices of G has at least one token. We succeeded in determining whether they are easy/hard to solve for different graph classes. The k-PVC concept arises when designing certain secured sensor networks. This research may be useful when we want to slightly change the network while keeping its secured property.

研究分野：Graph Algorithms

キーワード：reconfiguration problems　token sliding　k-path vertex cover　graph algorithms

1.　研究開始当初の背景

For the last decade, *reconfiguration problems* have attracted the attention from both theoretical and practical aspects of computer science [1, 2]. An instance of a reconfiguration problem consists of (1) a *description* of what a *configuration* is, and (2) a *reconfiguration rule* indicating how to "transform" one configuration into another. The question is whether there is a sequence of configurations that transforms one configuration into another such that each intermediate configuration is obtained from its predecessor by applying the reconfiguration rule exactly once. In several contexts, one may indicate a computational problem as a *source problem*, and identify each *configuration* with a *solution* of that source problem.

As an example, consider the Robot Motion problem [3], where robots and obstacles are placed in an environment, and one needs to figure out a way to move robots to their final destinations without having collision with other robots or obstacles. In a reconfiguration setting for modeling this problem, robots and obstacles placed in an environment are identified with labeled and unlabeled tokens placed on vertices of a graph. A reconfiguration rule, called the *Token Sliding* rule, can be defined by allowing a token to move only to one of its adjacent vertices. Analogously, one needs to determine if there is a reconfiguration sequence that transforms an initial configuration where the robots are placed at their sources to a target configuration where the robots are placed at their final destinations. Naturally, it is also required that the number of "sliding" operations must be minimized in the desired reconfiguration sequence.

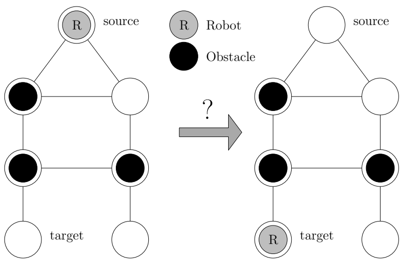

図 1　An example of modeling Robot Motion using reconfiguration.

From the theoretical viewpoint, **reconfiguration problems provide a new framework for gaining insightful knowledge of fundamental problems in the computational complexity theory**. As one may recall from the theory of computation, for the fundametal complexity classes P, NP, and PSPACE, we always have P ⊆ NP ⊆ PSPACE. Great effort has been made toward determining whether P ≠ NP, or NP ≠ PSPACE, or P ≠ PSPACE. In genral, reconfiguration problems tend to be PSPACE-hard. However, when they are restricted by some initial constraints, the complexity of the resulting problems may sometimes be in NP-hard or even in P. For a more precise example, one can refer to a variant of Robot Motion described before where no obstacles exist and all robots are identical, i.e., they are all identified by unlabeled tokens. In this case, one can design in polynomial time a plan for moving each robot to its final target. However, even when only one robot is different from all other robots, the problem is NP-hard [1]. In a different variant where all robots are identical and no edge of the input graph connects two tokens, even the problem of asking whether the reconfiguration sequence exists is indeed PSPACE-hard [4]. In short, reconfiguration problems, in some sense, "characterize" these fundamental complexity classes, and provide useful intuition for determining whether they are all different.

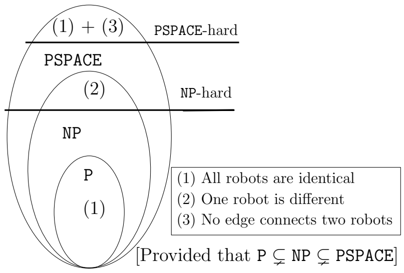

図 2　Reconfiguration "characterizes" complexity classes.

2.　研究の目的

Motivated by the importance of reconfiguration problems in both complexity theory and the modeling of Robot Motion, in this research, **we aim to draw the borderlines between the tractablility/intractability of different reconfiguration problems under Token Sliding rule**. From the theoretical viewpoint, classifying computational problems based on their tractability/intractability is one of the major subjects in the computational complexity theory. To the best of our knowledge, there is only a few number of source problems whose reconfiguration variants have been considered under the Token Sliding rule, namely Independent Set, Vertex Cover, Clique, Matching, Dominating Set, Spanning Subgraph, Induced Subgraph. As a result, **it is important to build a systematic understanding of the tractability/intractability of the reconfiguration problems that employ Token Sliding as a reconfiguration rule**. It is hoped that by achieving its primary goal, this research will not only contribute to the understanding of reconfiguration problems but also extend the current knowledge of the well-known complexity classes P, NP, and PSPACE.

3.　研究の方法

We plan to approach reconfiguration problems under Token Sliding rule from the following directions: (1) Tackle reconfiguration variants of different source problems to see which structural property of a problem makes it easy/hard to solve under Token Sliding; (2) Consider reconfiguration problems whose reconfiguration rule relates to “moving tokens on graphs” (e.g., the *Token Swapping* rule that allows only two adjacent tokens can be swapped) to understand why the problems under Token Sliding are easier/harder comparing to other “token-moving” reconfiguration rules; (3) Study different “types” of Token Sliding rule, for instance, by allowing multiple tokens to be simultaneously moved, or by restricting that only certain tokens can be moved, and so on. Toward the first direction, we aim to tackle the reconfiguration variants of some well-known NP-hard source problems. As these source problems have been well-investigated for years, we hope that some intuitive ideas for solving source problems can also be useful for understanding their reconfiguration variants. Toward the second direction, we aim to compare Token Sliding with other “similar” reconfiguration rules via characterizing the computational complexities of the corresponding reconfiguration problems that employ the same source problem. In this way, we hope to explain why it is sometimes more useful to consider Token Sliding in certain restricted conditions. Toward the third direction, we aim to understand how a slight modification in the Token Sliding rule affects the computational complexity of the corresponding reconfiguration problem. In this way, we hope to use the obtained results as a tool for efficiently modeling Robot Motion.

4.　研究成果

In a Token Sliding (TS) reconfiguration problem, each configuration is given as a set of tokens placed on vertices of a graph G, and two token-sets are adjacent if one can be obtained from the other via a single token-slide from an occupied vertex to one of its neighbors. The main task is to decide whether there is a sequence of token-slides between two given token-sets. We initiated the study of different reconfiguration variants of this problem where each token-set forms a k-path vertex cover (k-PVC) of G (i.e., a vertex-subset I such that each path on k vertices contains at least one of its members). We showed that the problems under TS and some other rules (namely, TJ, where a token can “jump” to any unoccupied vertex; and TAR(k), where each step involves either adding or removing a token such that the resulting token-set contains at least k members, for some positive integer k) are PSPACE-hard for different input graphs, including planar and bounded bandwidth graphs of maximum degree 3, bipartite graphs, and chordal graphs. On the positive side, we designed efficient algorithms for solving some variants on paths, cycle, trees. We presented these results at WALCOM 2020 [5]. The k-PVC concept arises when designing certain secured sensor networks [6]: the devices are vertices of the corresponding communication graph, two devices are adjacent if they can communicate, and one need to put a “secured” device on every k-vertex path. This research may be useful when we want to slightly change the network while keeping its secured property.

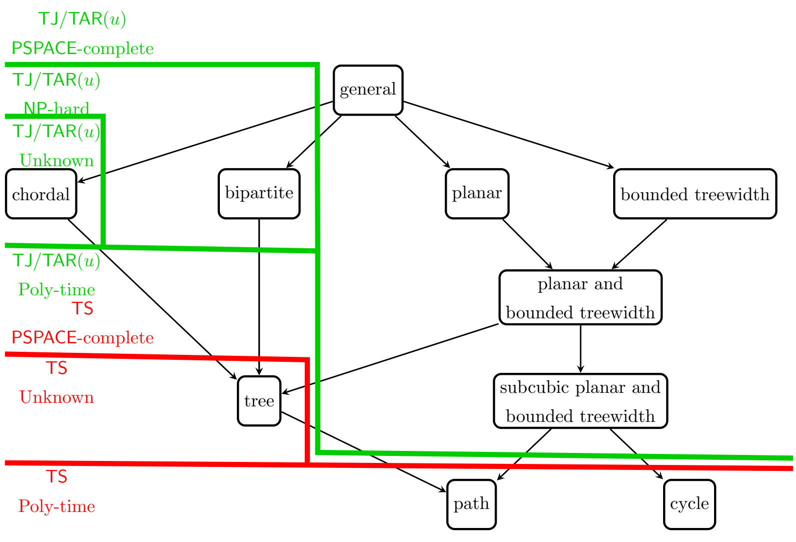

図 3　A summary of our results [5]. Each rectangle indicates a graph class, and the arrow from $\mathcal{G}$ to $\mathcal{H}$ tells us $\mathcal{H}$ is a subclass of $\mathcal{G}$.

参考文献

[1] van den Heuvel, J.: "The Complexity of Change." In: *Surveys in Combinatorics*, London Mathematical Society Lecture Note Series, vol. 409, pp. 127–160. Cambridge University Press (2013).

[2] Nishimura, N.: "Introduction to Reconfiguration." *Algorithms* 11(4) (2018). (article 52).

[3] Papadimitriou, C.H., Raghavan, P., Sudan, M., Tamaki, H.: "Motion planning on a graph." *Proceedings of SFCS 1994*, pp. 511-520, IEEE.

[4] Hearn, R.A., Demaine, E.D.: "PSPACE-Completeness of Sliding-Block Puzzles and Other Problems through the Nondeterministic Constraint Logic Model of Computation." *Theoretical Computer Science* 343(1-2), pp. 72–96 (2005).

[5] Hoang, D.A., Suzuki, A., Yagita, T.: "Reconfiguring k-path Vertex Covers." *Proceedings of WALCOM 2020*, LNCS 12049, pp. 133-145, Springer.

[6] Brešar, B., Kardoš, F., Katrenič, J., Semanišin, G.: "Minimum k-Path Vertex Cover." *Discrete Applied Mathematics* 159(12), pp. 1189–1195 (2011).

５．主な発表論文等

〔雑誌論文〕　計1件（うち査読付論文　1件／うち国際共著　1件／うちオープンアクセス　0件）

１．著者名 Hoang Duc A.、Suzuki Akira、Yagita Tsuyoshi	４．巻 LNCS 12049
２．論文標題 Reconfiguring k-path Vertex Covers	５．発行年 2020年
３．雑誌名 Proceedings of the 14th International Conference and Workshop on Algorithms and Computation, WALCOM 2020	６．最初と最後の頁 133～145
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-030-39881-1_12	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕　計1件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 Duc A. Hoang, Akira Suzuki, Tsuyoshi Yagita
２．発表標題 Reconfiguring k-Path Vertex Covers
３．学会等名 The 14th International Conference and Workshop on Algorithms and Computation, WALCOM 2020
４．発表年 2020年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

－

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関