

Research Article

On Solving the Decycling Problem in a Torus Network

Antoine Bossard 

Graduate School of Science, Kanagawa University, Tsuchiya 2946, Hiratsuka, Kanagawa, Japan 259-1293

Correspondence should be addressed to Antoine Bossard; abossard@kanagawa-u.ac.jp

Received 26 February 2021; Accepted 30 March 2021; Published 12 April 2021

Academic Editor: Yong Zhang

Copyright © 2021 Antoine Bossard. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modern supercomputers are massively parallel systems: they embody thousands of computing nodes and sometimes several millions. The torus topology has proven very popular for the interconnect of these high-performance systems. Notably, this network topology is employed by the supercomputer ranked number one in the world as of November 2020, the supercomputer Fugaku. Given the high number of compute nodes in such systems, efficient parallel processing is critical to maximise the computing performance. It is well known that cycles harm the parallel processing capacity of systems: for instance, deadlocks and starvation are two notorious issues of parallel computing that are directly linked to the presence of cycles. Hence, network decycling is an important issue, and it has been extensively discussed in the literature. We describe in this paper a decycling algorithm for the 3-dimensional k -ary torus topology and compare it with established results, both theoretically and experimentally. (This paper is a revised version of Antoine Bossard (2020)).

1. Introduction

The supercomputers of the 21st century are massively parallel systems: they embody thousands of compute nodes. Some recent devices even include several millions of nodes (e.g., 10,649,600 nodes for the Sunway TaihuLight as of November 2020's TOP500 list [1]). The interconnection of these compute nodes is thus a critical issue so as to maximise the parallel processing performance and thus the machine performance overall. Thanks to its advantageous topological properties, such as regularity, the torus network topology has proven very popular for the interconnect of modern supercomputers. For example, the supercomputer ranked number one in the world in the November 2020 TOP500 ranking, the supercomputer Fugaku built by Fujitsu and RIKEN, employs the torus topology to connect its nodes (Tofu interconnect D [2]). The IBM Blue Gene/L and Blue Gene/P, Cray Titan (Gemini interconnect [3]), and Fujitsu SORA-MA (Tofu interconnect 2 [4]) are other examples of supercomputers based on the torus topology.

It is well known that parallel processing is harmed by the presence of cycles: they are at the source of the deadlock, livelock, and starvation notorious resource allocation issues [5]. Notably, because it has important implications for parallel

processing, the decycling problem, also known as the minimum feedback vertex set problem, has been extensively addressed in the literature. Karp has shown that finding a decycling set of minimum size (i.e., an optimal decycling set) in any graph is NP-complete [6]. For instance, Fomin et al. have described an algorithm that solves this problem in any graph in $O(1.7548^n)$ time [7]. Furthermore, polynomial solutions have been described for several particular classes of graphs such as 3-regular graphs [8], convex bipartite graphs [9], permutation graphs [10], and hypercube-based networks [11, 12]. Among others, the size of an optimal decycling set (i.e., the decycling number) in the case of cubes and grids has been discussed in [13, 14] and for hypercubes in [15]. We describe in this paper a polynomial time decycling algorithm for a 3-dimensional k -ary torus network. One should note that while the case of a grid as mentioned above seems close, or at least related, to the case of the torus which we investigate hereinafter, the wrap-around edges of the torus invalidate the grid decycling approach (refer to the next section for additional details).

The rest of this paper is organised as follows. Notations, definitions, and previous results are recalled in Section 2. The decycling algorithm is presented in Section 3, including the proof of its correctness and complexity analysis.

Theoretical and empirical evaluations are conducted in Section 4. Finally, concluding remarks are given in Section 5.

2. Preliminaries

We recall in this section several notations, definitions, and previously established results. The set of the vertices of a graph G is denoted by $V(G)$, and the set of its edges by $E(G)$. A path in a graph G is a subgraph of G that is an alternating sequence of distinct vertices and edges. Such a vertex–edge sequence but whose two terminal vertices are the same vertex is called a cycle. The length of a path or cycle is its number of edges. A graph that contains no cycle is said to be acyclic and is isomorphic to a tree.

Definition 1. An n -dimensional k -ary torus, denoted by $T(n, k)$, with $n \geq 1$ and $k \geq 1$, consists in the k^n vertices induced by the set $\{0, 1, \dots, k-1\}^n$. Two vertices $u = (u_0, u_1, \dots, u_{n-1})$ and $v = (v_0, v_1, \dots, v_{n-1})$ of a $T(n, k)$ are adjacent if and only if there exists j ($0 \leq j < n$) such that $\forall i$ ($0 \leq i < n$, $i \neq j$) $u_i = v_i$ and $u_j = v_j \pm 1 \pmod{k}$.

A torus $T(n, k)$ is thus a regular graph of degree $2n$, of diameter $n \lfloor k/2 \rfloor$ and has nk^n edges. An essential torus property is next recalled.

Property 2. For a dimension δ ($0 \leq \delta < n$), a $T(n, k)$ consists in k subtori $T^{i,\delta}(n-1, k)$ ($0 \leq i < k$). Each subtorus $T^{i,\delta}(n-1, k)$ is induced by the k^{n-1} vertices $(u_0, u_1, \dots, u_{\delta-1}, i, u_{\delta+1}, \dots, u_{n-1})$ of $T(n, k)$ with u_j ($0 \leq j < n$, $j \neq \delta$) the vertex coordinate for the dimension j and i the vertex coordinate for the dimension δ .

A torus $T(2, 3)$ is shown in Figure 1(a) and its recursive structure is illustrated in Figure 1(b).

Next, previously established results are recalled.

Beineke and Vandell have established a lower bound on the size of a decycling set for any graph [16]. This result is recalled in Theorem 3 below.

Theorem 3 (see [16]). *Given a graph $G = (V, E)$ of maximum degree Δ , any decycling set S of G satisfies the following relation:*

$$|S| \geq \left\lceil \frac{|E| - |V| + 1}{\Delta - 1} \right\rceil. \quad (1)$$

We were unaware until very recently (after the publication of [17]) that Pike and Zou have shown how to calculate a decycling set of minimum size for a 2-dimensional torus in [18]. The corresponding result is recalled in Theorem 4 below.

Theorem 4 (see [18]). *In a $T(2, k)$ with $k \geq 3$, a decycling set S^k of minimum size with*

$$|S^k| = \begin{cases} \frac{3k}{2} = 6, & k = 4, \\ \left\lceil \frac{k^2 + 2}{3} \right\rceil, & \text{otherwise,} \end{cases} \quad (2)$$

can be found in $O(k^2)$ time.

3. The Case of a $T(3, k)$

We describe in this section the details of our approach to decycle a 3-dimensional k -ary torus $T(3, k)$.

3.1. Algorithm Description. We give below a constructive proof in the form of a decycling algorithm whose input is an arity k ($k \geq 1$) and which outputs a decycling set S^k .

The main idea is to consider one dimension δ to reduce $T(3, k)$ into k 2-dimensional subtori as per Property 2 and to alternate for each such subtorus the optimal decycling of a $T(2, k)$ (i.e., Theorem 4) and two other decycling methods of a $T(2, k)$ which induce a graph with no edge.

The case $k = 1$ is trivial: $T(3, 1)$ consists of one unique vertex and is thus acyclic; $S^1 = \emptyset$ is thus a decycling set for this trivial graph. A $T(3, 2)$ is isomorphic to a 3-dimensional hypercube; $S^2 = \{(0, 0, 0), (1, 1, 0), (1, 1, 1)\}$ is thus an optimal decycling set by Theorem 3. Hence, we can now assume that $k \geq 3$.

Step 1. We distinguish the two cases k even and k odd.

Case k even:

Define two decycling sets S_1^k, S_2^k in a two-dimensional k -ary torus $T(2, k)$ as follows:

$$\begin{aligned} S_1^k &= \{(i, j) \in V(T(2, k)) \mid 0 \leq i, j \leq k-1, i+j \equiv 1 \pmod{2}\}, \\ S_2^k &= \{(i, j) \in V(T(2, k)) \mid 0 \leq i, j \leq k-1, i+j \equiv 0 \pmod{2}\}. \end{aligned} \quad (3)$$

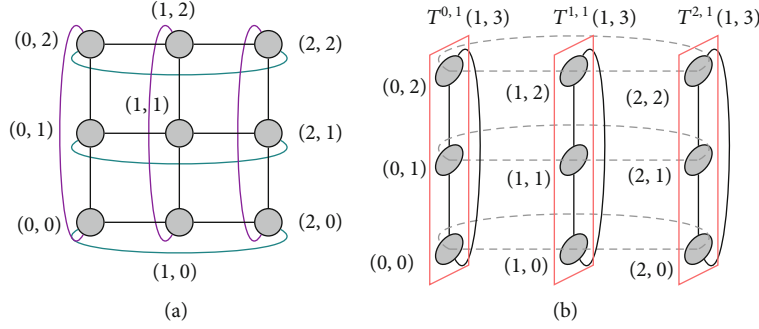
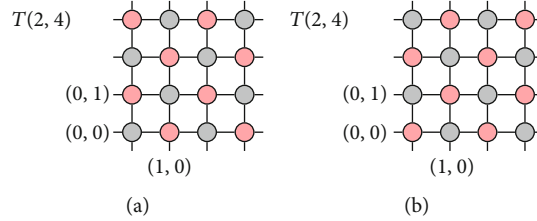
In other words, the set S_1^k is induced by the vertices of $T(2, k)$ that are taken in one particular ‘‘quincunx’’ manner, and the set S_2^k by the vertices of $T(2, k)$ that are taken in the other ‘‘quincunx’’ manner. Precisely, we have $S_2^k = V(T(2, k)) \setminus S_1^k$ and $S_1^k = V(T(2, k)) \setminus S_2^k$.

The sets S_1^k and S_2^k when $k = 4$ are illustrated in Figures 2(a) and 2(b), respectively; they consist in the red vertices.

Case k odd:

Define two decycling sets S_1^k, S_2^k in a two-dimensional k -ary torus $T(2, k)$ as follows:

$$\begin{aligned} S_1^k &= \{(i, j) \in V(T(2, k)) \mid 0 \leq i, j \leq k-1, i+j \equiv 1 \pmod{2}\} \cup \{(i, k-1) \mid 0 \leq i \leq k-1\} \cup \{(k-1, i) \mid 0 \leq i \leq k-1\}, \\ S_2^k &= \{(i, j) \in V(T(2, k)) \mid 0 \leq i, j \leq k-1, i+j \equiv 0 \pmod{2}\} \cup \{(i, k-1) \mid 0 \leq i \leq k-1\} \cup \{(k-1, i) \mid 0 \leq i \leq k-1\}. \end{aligned} \quad (4)$$


 FIGURE 1: (a) A torus $T(2,3)$. (b) A torus has a recursive structure: a $T(2,3)$ consists in three subtori $T(1,3)$.

 FIGURE 2: The sets S_1^k (a) and S_2^k (b) as defined in a $T(2,k)$ when $k=4$; they consist in the red vertices.

In other words, the set S_1^k is induced by the vertices of $T(2,k)$ that are taken in one particular “quincunx” manner and also includes the vertices of the top row and those of the right row. The sets S_1^k and S_2^k when $k=5$ are illustrated in Figures 3(a) and 3(b), respectively; they consist in the red vertices.

Step 2. Let $\delta=0$ and consider the k subtori $T^{i,\delta}(2,k)$ ($0 \leq i \leq k-1$) as per Property 2. Define S_0^k the optimal decycling set of $T(2,k)$ as induced by Theorem 4. We distinguish the three cases that are induced by the value of $k \bmod 3$.

Case $k \equiv 0 \pmod{3}$:

Decycle each subtorus $T^{i,\delta}(2,k)$ ($0 \leq i \leq k-1$) with the vertex set $S_{i \bmod 3}^k$ (see Figure 4). In this figure, where there can be edges on the third dimension between two subtori, sample edges are shown.

Case $k \equiv 1 \pmod{3}$:

Decycle each subtorus $T^{i,\delta}(2,k)$ ($0 \leq i \leq k-2$) with the vertex set $S_{i \bmod 3}^k$ and $T^{k-1,\delta}(2,k)$ with the vertex set S_2^k (see Figure 5). Again in this figure, where there can be edges on the third dimension between two subtori, sample edges are shown.

Case $k \equiv 2 \pmod{3}$:

Decycle each subtorus $T^{i,\delta}(2,k)$ ($0 \leq i \leq k-2$) with the vertex set $S_{i \bmod 3}^k$ and $T^{k-1,\delta}(2,k)$ with the vertex set $V(T^{k-1,\delta}(2,k))$ (see Figure 6). Again in this figure, where there can be edges on the third dimension between two subtori, sample edges are shown.

3.2. Correctness and Complexities. In this section, we prove the correctness of the proposed algorithm and establish its complexities.

Theorem 5. In a 3-dimensional k -ary torus $T(3,k)$ ($k \geq 1$), a decycling set S^k of 0 vertex when $k=1$, 3 vertices when $k=2$, and in the other cases with

$$|S^k| = \begin{cases} k \frac{\lceil (k^2+2)/3 \rceil + k^2}{3}, & k \equiv 0 \pmod{3}, \\ 30, & k \equiv 1 \pmod{3}, k=4, \\ (k-1) \frac{\lceil (k^2+2)/3 \rceil + k^2}{3} + \frac{k^2}{2}, & k \equiv 1 \pmod{3}, k > 4 \\ (k+1) \frac{\lceil (k^2+2)/3 \rceil + k^2}{3}, & k \equiv 2 \pmod{3}, \end{cases} \quad (5)$$

when k is even, and with

$$|S^k| = \begin{cases} k \frac{\lceil (k^2+2)/3 \rceil + k^2 + 2k - 1}{3}, & k \equiv 0 \pmod{3}, \\ (k-1) \frac{\lceil (k^2+2)/3 \rceil + k^2 + 2k - 1}{3} + \frac{k^2 - 1}{2} + k, & k \equiv 1 \pmod{3}, \\ (k-2) \frac{\lceil (k^2+2)/3 \rceil + k^2 + 2k - 1}{3} + \frac{3k^2 - 1}{2} + k, & k \equiv 2 \pmod{3}, \end{cases} \quad (6)$$

when k is odd can be found in optimal $O(k^3)$ time.

Proof. The cases induced by $k \leq 2$ are trivial; they have already been shown at the beginning of Section 3.1. So, we can assume $k \geq 3$.

By definition, the subgraph of $T(2,k)$ induced by the vertices of the set $V(T(2,k)) \setminus S_1^k$ has no edge. And similarly, the

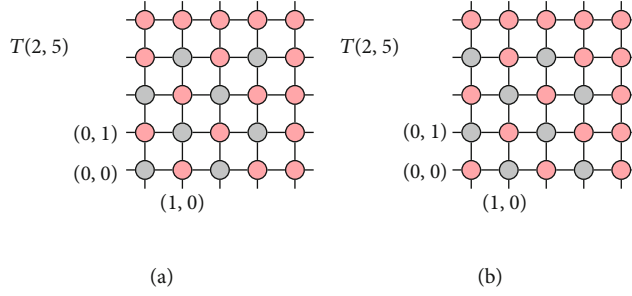


FIGURE 3: The sets S_1^k (a) and S_2^k (b) as defined in a $T(2, k)$ when $k = 5$; they consist in the red vertices.

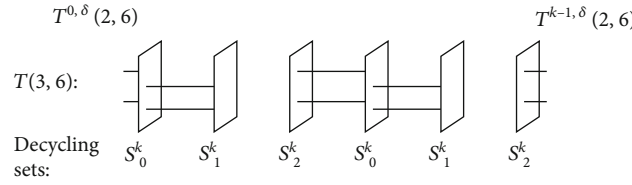


FIGURE 4: An illustration of the case $k \equiv 0 \pmod{3}$ with $k = 6$. The selected decycling set for each of the $T^{i,\delta}(2, k)$ ($0 \leq i \leq k - 1$) subtori is given below it. Where there can be edges on the third dimension between two subtori, sample edges are shown.

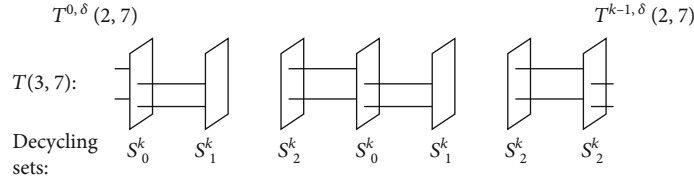


FIGURE 5: An illustration of the case $k \equiv 1 \pmod{3}$ with $k = 7$. The selected decycling set for each of the $T^{i,\delta}(2, k)$ ($0 \leq i \leq k - 1$) subtori is given below it. Where there can be edges on the third dimension between two subtori, sample edges are shown.

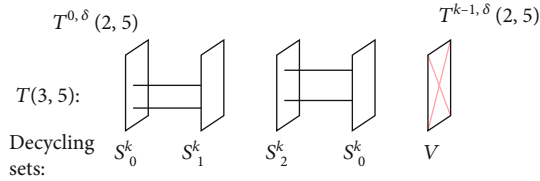


FIGURE 6: An illustration of the case $k \equiv 2 \pmod{3}$ with $k = 5$. The selected decycling set for each of the $T^{i,\delta}(2, k)$ ($0 \leq i \leq k - 1$) subtori is given below it. Where there can be edges on the third dimension between two subtori, sample edges are shown.

subgraph of $T(2, k)$ induced by the vertices of the set $V(T(2, k)) \setminus S_2^k$ has no edge either.

Hence, by definition of the algorithm, each subtorus $T^{i,\delta}(2, k)$ ($0 \leq i \leq k - 1$) is acyclic. Moreover, the only edges on the third dimension are at a vertex of a graph induced by a subtorus decycled with S_0^k . Consider two such graphs induced by a subtorus decycled with S_0^k , say the graphs that correspond to $T^{i,\delta}(2, k)$ and $T^{j,\delta}(2, k)$ where $i < j$ and $j - i$ minimal. In the case $k \equiv 0 \pmod{3}$, we have $j - i = 3$ and the greatest such j is $k - 3$. And since for every edge on the third dimension the vertex that is not in a graph induced by a sub-

torus decycled with S_0^k is inside a graph induced by a subtorus that has no edge, the resulting graph is acyclic. In the case $k \equiv 1 \pmod{3}$, we have $j - i = 3$ and the greatest such j is $k - 4$. So, again and for the same reason, the resulting graph is acyclic. In the case $k \equiv 2 \pmod{3}$, we have $j - i = 3$ and the greatest such j is $k - 2$, so there could be a path on the third dimension between a vertex of $T^{k-2,\delta}(2, k)$ (i.e., the rightmost graph induced by S_0^k) and of $T^{0,\delta}(2, k)$ (which is also induced by S_0^k). However, all the vertices of $T^{k-1,\delta}(2, k)$ are removed, so there is no such path and the resulting graph is thus acyclic.

The set S_1^k and the set S_2^k each have $k^2/2$ vertices when k is even and $(k - 1)^2/2 + (2k - 1) = (k^2 - 1)/2 + k$ when k is odd. They can thus each be calculated in $O(k^2)$ time. By Theorem 4, the set S_0^k has $3k/2$ vertices when $k = 4$ and $\lceil (k^2 + 2)/3 \rceil$ vertices otherwise, and this set can be calculated in $O(k^2)$ time.

Hence, in a $T(3, k)$, a decycling set S^k with $|S^k| = k(|S_0^k| + |S_1^k| + |S_2^k|)/3$ when $k \equiv 0 \pmod{3}$, $|S^k| = (k - 1)(|S_0^k| + |S_1^k| + |S_2^k|)/3 + |S_2^k|$ when $k \equiv 1 \pmod{3}$, and $|S^k| = (k - 2)(|S_0^k| + |S_1^k| + |S_2^k|)/3 + |S_0^k| + k^2$ when $k \equiv 2 \pmod{3}$ can be found in $O(k^3)$ time. By further distinguishing the two cases k even and k odd, we obtain the expected set sizes.

From Theorem 3, we have in a $T(3, k)$ that $|S| \geq (3k^3 - k^3 + 1)/5 = (2k^3 + 1)/5$. Therefore, $O(k^3)$ is optimal time.

4. Discussion

In this section, we discuss the obtained theoretical results and compare them with experimental data.

4.1. Comparison with the Lower Bound. We investigate in this section how close the size of the generated decycling set is to the lower bound given by Theorem 3. Figure 7 shows the values obtained from Theorem 3 and Theorem 5. Let us recall that the result of Theorem 3 is a lower bound on the size of a decycling set, and not necessarily the size of an optimal decycling set. So, the difference plotted in Figure 7 is given for reference, and it shows that the size of the obtained decycling set is promising, possibly optimal in some cases, given that it is rather close, and sometimes equal, to the lower bound of Theorem 3.

One can also notice that the size of the decycling set generated by the proposed algorithm is never smaller than the lower bound of Theorem 3. If that were the case, this would indicate a hole in the proposed algorithm.

4.2. Comparison with the Results of a Computer Experiment. We have implemented a stochastic decycling algorithm in order to compare the obtained theoretical results with those obtained experimentally. As recalled in introduction, this is an NP-complete problem; hence, the graph decycling implementation we use only approximates the size of an optimal decycling set. This implementation follows the method described in [19]. The stochastic implementation was run 1,000 times.

The values obtained from Theorem 5 and the computer experiment are shown in Table 1. The minimum size and the average size of the decycling set generated by the stochastic implementation are given. The values from Theorem 3—a lower bound on the size of a decycling set, and not necessarily the size of an optimal decycling set—are also given for reference.

From these results, it can be noticed that the proposed algorithm beats the stochastic implementation on each of all its 1,000 runs at $k \equiv 2 \pmod{4}$ and is equal or very nearly equal to it at $k \equiv 0 \pmod{4}$. In other words, when k is even, our proposal induces a smaller or nearly equal decycling set than the best decycling set found after 1,000 runs. And when k is odd, as k increases, the size difference between our proposal and the stochastic implementation continuously decreases, and at $k=9$, our proposal beats the stochastic implementation when considering the average value of its 1,000 runs.

And, of course, the complexity of the stochastic implementation is always prohibitive [19], especially compared to the worst-case time complexity of the proposal ($O(k^3)$, see Theorem 5). These are very positive results which quantitatively show the significance of the proposal. For reference, in the case $k=10$, the stochastic implementation took more than 2.5 hours to complete the 1,000 runs on a midrange computer (Intel Core i5-1035G7 CPU, 8 GB RAM).

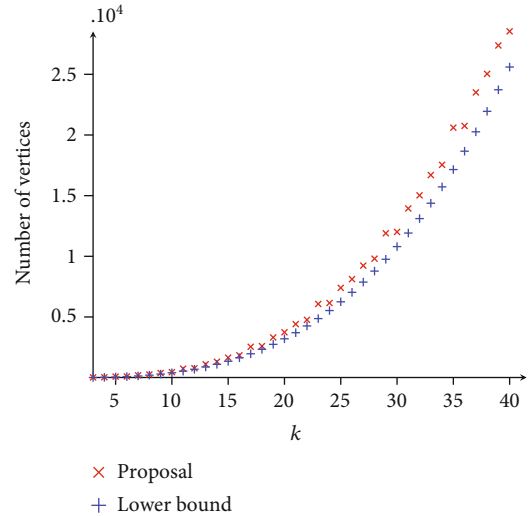


FIGURE 7: Investigating how close the size of the decycling set generated is to the lower bound of Theorem 3.

TABLE 1: Comparison of the size of the decycling set obtained with the proposed method in a $T(3, k)$ with that of the experimentally generated decycling sets. The theoretical lower bound is also given for reference.

Arity k	Proposal (Theorem 5)	Experiment (minimum size)	Experiment (average size)	Lower bound (Theorem 3)
3	18	14	14.71	11
4	30	30	33.85	26
5	85	60	65.86	51
6	98	106	113.85	87
7	189	171	180.46	138
8	258	257	269.15	205
9	378	371	383.21	292
10	452	505	526.01	401

Finally, it can also be noticed that the size of the decycling set generated by the proposed algorithm in a $T(3, k)$ is $O(k^n)$. Besides, by Theorem 4, the size of an optimal decycling set in a $T(2, k)$ is also $O(k^n)$. This is yet another positive indicator of the performance of our proposal.

5. Concluding Remarks

The torus topology is nowadays ubiquitous in supercomputing. It is the network topology of choice for the interconnect of massively parallel systems: it is for instance employed by the supercomputer ranked number one in the world as of November 2020, the supercomputer Fugaku. Besides, it is common knowledge that cycles in the network of compute nodes harm parallel processing, and this is one reason why the decycling problem—NP-complete—has been extensively addressed in the literature. We have described in this paper a decycling algorithm for a torus $T(3, k)$. Thanks to the recursive property of the torus topology, this proposal can be used

to decycle parts (i.e., subtori) of a torus of higher dimension, which as explained will consequently facilitate parallel processing. Precisely, we have given a constructive proof of a decycling set S^k for a torus $T(3, k)$ where S^k has $k(|S_0^k| + 2|S_1^k|)/3$ vertices when $k \equiv 0 \pmod{3}$, $(k-1)(|S_0^k| + 2|S_1^k|)/3 + |S_1^k|$ vertices when $k \equiv 1 \pmod{3}$, $(k-2)(|S_0^k| + 2|S_1^k|)/3 + |S_0^k| + k^2$ vertices when $k \equiv 2 \pmod{3}$ with S_0^k an optimal decycling set in $T(2, k)$, $|S_1^k| = k^2/2$ when k even, and $|S_1^k| = (k^2 - 1)/2 + k$ when k odd and can be obtained in $O(k^3)$ optimal time. We have formally evaluated the proposed algorithm and conducted evaluation experiments to compare it to conventional approaches. The obtained results have quantitatively shown the significance of the proposal.

Regarding future works, refining the proposed decycling algorithm so that the generated decycling set includes a smaller number of vertices is a first meaningful objective. Then, it will be very interesting to investigate, for instance, as explained above by using the recursive property of the torus topology, how to rely on the obtained results to produce nontrivial decycling sets for tori of higher dimensions.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The author is sincerely grateful towards Professor David Pike (Memorial University of Newfoundland, NL, Canada) for communicating the existence of [18] after he noticed the publication of [17]. This research was partly supported by a Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under grant no. 19K11887.

References

- [1] TOP500, *TOP500 list - November 2020, June 2020* February 2021, <https://www.top500.org/lists/top500/list/2020/11/>.
- [2] Y. Ajima, T. Kawashima, T. Okamoto et al., "The Tofu interconnect D," in *Proceedings of the IEEE International Conference on Cluster Computing*, pp. 646–654, Belfast, United Kingdom, September 2018.
- [3] R. Alverson, D. Roweth, and L. Kaplan, "The Gemini system interconnect," in *Proceedings of the 18th IEEE Symposium on High Performance Interconnects*, pp. 83–87, Mountain View, CA, USA, August 2010.
- [4] Y. Ajima, T. Inoue, S. Hiramoto et al., "Tofu interconnect 2: system-on-chip integration of high-performance interconnect," in *Proceedings of the 29th International Supercomputing Conference*, pp. 498–507, Leipzig, Germany, June 2014.
- [5] P. Festa, P. M. Pardalos, and M. G. C. Resende, *Feedback Set Problems*, Springer US, Boston, MA, USA, 1999.
- [6] R. M. Karp, "Reducibility among combinatorial problems," in *Proceedings of the Symposium on Complexity of Computer Computations*, pp. 85–103, Yorktown Heights, NY, USA, March 1972.
- [7] F. V. Fomin, S. Gaspers, and A. V. Pyatkin, "Finding a minimum feedback vertex set in time $O(1.7548^n)$," in *Proceedings of the International Workshop on Parameterized and Exact Computation*, vol. 4169, pp. 184–191, Zurich, Switzerland, September 2006.
- [8] D. Li and Y. Liu, "A polynomial algorithm for finding the minimum feedback vertex set of a 3-regular simple graph," *Acta Mathematica Scientia*, vol. 19, no. 4, pp. 375–381, 1999.
- [9] Y. D. Liang and M.-S. Chang, "Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs," *Acta Informatica*, vol. 34, no. 5, pp. 337–346, 1997.
- [10] Y. Daniel Liang, "On the feedback vertex set problem in permutation graphs," *Information Processing Letters*, vol. 52, no. 3, pp. 123–129, 1994.
- [11] A. Bossard, "On the decycling problem in hierarchical hypercubes," *Journal of Interconnection Networks*, vol. 14, no. 2, pp. 1350006.1–1350006.13, 2013.
- [12] A. Bossard, "The decycling problem in hierarchical cubic networks," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 293–305, 2014.
- [13] S. Bau, L. Beineke, Z. Liu, G.-M. Du, and R. Vandell, "Decycling cubes and grids," *Utilitas Mathematica*, vol. 59, 2001.
- [14] S. Bau and L. W. Beineke, "The decycling number of graphs," *The Australasian Journal of Combinatorics*, vol. 25, pp. 285–298, 2002.
- [15] D. A. Pike, "Decycling hypercubes," *Graphs and Combinatorics*, vol. 19, no. 4, pp. 547–550, 2003.
- [16] L. W. Beineke and R. C. Vandell, "Decycling graphs," *Journal of Graph Theory*, vol. 25, no. 1, pp. 59–77, 1997.
- [17] A. Bossard, "On the decycling problem in a torus," in *Proceedings of the 11th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, vol. 1362, pp. 12–21, Shenzhen, China, December 2020.
- [18] D. A. Pike and Y. Zou, "Decycling Cartesian products of two cycles," *SIAM Journal on Discrete Mathematics*, vol. 19, no. 3, pp. 651–663, 2005.
- [19] A. Bossard and K. Kaneko, "Approximating optimal decycling sets in HHC and HCN for small dimensions," in *Proceedings of the 30th International Conference on Computers and Their Applications (CATA)*, pp. 145–150, Honolulu, HI, USA, March 2015.