Malaysian Journal of Mathematical Sciences 18(3): 583–595 (2024)
https://doi.org/10.47836/mjms.18.3.08

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

Applying the Diamond Product of Graphs to the Round Robin Tournament Scheduling Problem

Rutjanisarakul, T.[1] and Sumetthapiwat, S.*[1]

[1]*Department of Social and Applied Science, College of Industrial Technology, King Mongkut's University of Technology North Bangkok, Thailand*

E-mail: supphakorn.s@cit.kmutnb.ac.th
**Corresponding author*

Received: 16 December 2023
Accepted: 30 April 2024

Abstract

The diamond product of a graph $G(V, E)$ with a graph $H(V', E')$ denoted by $G \diamond H$ is a graph whose a vertex set $V(G \diamond H)$ is a $Hom(G, H)$ and an edge set $E(G \diamond H) = \{\{f, g\} | f, g \in Hom(G, H)$ and $\{f(x), g(x)\} \in E'$, for all $x \in V(G)\}$. A round robin tournament problem involves creating a schedule where each participant plays against every other participant exactly once. This research represents the application of the diamond product of path graph and complete graph to $2n$-participants round robin tournament problem. Moreover, the research also represents an algorithm to find a solution of $2n$-participants round robin tournament problem.

Keywords: diamond product; homomorphism; graph theory; scheduling problem.

1 Introduction

Graph theory has many theoretical techniques to be used to solve a scheduling problem such as a traveling salesman problem, a schedule of a sport tournament, a schedule of machine, and etc. The objective usually is to minimize the number of traveling, the total distance of traveling, the cost of traveling, or the machine time. There are many techniques which are used to solve a scheduling problem in different conditions. In 2007, Ribeiro and Urrutia [7] studied the effective heuristics and mirrored Traveling Tournament Problem (mTTP). They presented the new GRILS-mTTP heuristic used to solve the mTTP. In 2013, Hoshino and Kawarabayashi [4] applied theory of graph to find the shortest-path and distance-optimal intra-league schedules based on Nippon Professional Baseball. The following year, Goerigk et al. [3] presented the new technique to generate a schedule by finding the graph's minimum-weight of three consecutive points or three-vertex path. In 2018, Rutjanisarakul and Jiarasuksakun [8] presented the new method called the swapping which was used in the genetic algorithm. They applied the genetic algorithm with swapping method to solve the sport tournament problem with the minimum number of traveling of all teams. In addition, a schedule of machine is also interesting. In 2022, Malhotra et al. [6] considered the flow shop scheduling and solved by using the model called "Branch and bound" with three like parallel machines at the beginning level. The result was compared to other techniques and it presented that this technique had more efficiency than other techniques. However, there is another interested operator, which is the diamond product defined by Arworn [2]. The diamond product of two graphs is the special case of homomorphism. Damnernsawat [2] and Arworn, who was her advisor, studied the diamond product operator and found theories of properties of the diamond product of Cayley graphs of groups. In 2009, Thomkeaw and Arworn [9] studied the endomorphism of book graphs and presented the properties of the endomorphism monoid. In 2010, Jiarasuksakun et al. [5] studied the diamond product of two complete bipartite graphs. They found that the diamond product of two complete bipartite graphs is also the complete bipartite graph. In addition, they found some properties of the endomorphism monoid of diamond product of two complete bipartite graphs. Although the diamond product operator is interesting, commuting graphs are also interesting. In 2018, Bhat and Sudhakara [1] considered many graphs such as trees, complete graphs, cycles and their generalized complements. The idea of partition of graph motivates to consider the properties of the diamond product.

Nevertheless, this research shows some properties of the diamond product of a path graph (P_{n-1}) with a complete graph (K_n). This research also shows the application of the diamond product of a path graph (P_{n-1}) with a complete graph (K_n). The diamond product of a path graph (P_{n-1}) with a complete graph (K_n) contains all solutions of the round robin tournament scheduling problem of n teams, where n is an even positive integer greater than 2. Moreover, we also present an algorithm for finding a subgraph of the diamond product of a path graph (P_{n-1}) with a complete graph (K_n) that is a one of solutions of a round robin sport tournament of n teams.

2 Preliminaries

2.1 Diamond product in graph

In this section, we describe the definition of the diamond product of graphs and the scheduling problem that is studied in this research.

Definition 2.1. [10] *A homomorphism of a graph $G = (V, E)$ into a graph $H(V', E')$ is a mapping $f : V \to V'$, which preserves edges: for all $x, y \in V$, if $\{x, y\} \in E$, then $\{f(x), f(y)\} \in E'$. Let $Hom(G, H))$ be the class of all homomorphisms from a graph G into a graph H.*

For example, $Hom(P_2, K_3)$ consists of 6 homomorphisms, as shown in Figure 1.

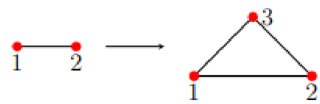

Figure 1: Finding $Hom(P_2, K_3)$.

$Hom(P_2, K_3) = \{(1\ 2), (1\ 3), (2\ 3), (2\ 1), (3\ 1), (3\ 2)\}$. However, $Hom(K_3, P_2)$ cannot be found.

Definition 2.2. [2] *The diamond product of a graph $G = (V, E)$ and a graph $H = (V', E')$ (denoted by $G \diamond H$) is a graph defined by the vertex set $V(G \diamond H) = Hom(G, H)$, where $Hom(G, H) \neq \emptyset$, and the edge set $E(G \diamond H) = \{\{f, g\} \subset Hom(G, H) | \{f(x), g(x)\} \in E' for\ all\ x \in V\}$.*

An example of graph $P_2 \diamond K_3$ is illustrated in Figure 2.

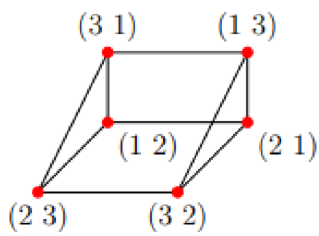

Figure 2: Graph $P_2 \diamond K_3$.

2.2 The scheduling problem

The scheduling problem is to find scheduling sequences of all teams in a tournament. There are 2 kinds of tournament that are round robin tournament and double round robin tournament. The difference between these tournaments is the number of weeks in a tournament where the number of weeks of double round robin tournament is twice as large as the number of weeks of round robin tournament.

For example, scheduling sequences of four team round robin tournament as shown in Table 1. There are three weeks in four team round robin tournament.

Table 1: An example of four team round robin tournament.

Team number	W_1	W_2	W_3
1	4	3	2
2	3	4	1
3	2	1	4
4	1	2	3

Note: W_1, W_2 and W_3 represent weeks 1, 2 and 3, respectively.

2.3 Relation between diamond product graph and scheduling

The scheduling of four team round robin tournament can be formed a complete graph with four vertices, which each vertex represents the scheduling of a team. According to the Table 1, the scheduling of team number $1, 2, 3$, and 4 are vertices $v_1 = (4\ 3\ 2)$, $v_2 = (3\ 4\ 1)$, $v_3 = (2\ 1\ 4)$, and $v_4 = (1\ 2\ 3)$, respectively. Moreover, any two vertices are adjacent if two vertices of scheduling of a team are in the same tournament. So, each vertex v_i is adjacent to each others. Thus, a formed graph is a complete graph K_4 as shown in Figure 3.

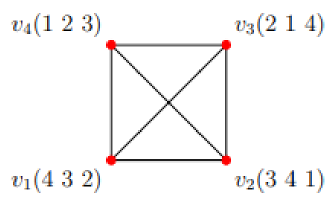

Figure 3: A graph of four team round robin tournament.

Alternatively, the graph $P_3 \diamond K_4$ is considered. The vertex set of $P_3 \diamond K_4$ is $\text{Hom}(P_3, K_4) = \{f = (f(1)\ f(2)\ f(3)) | f(i) \neq f(i+1)$, where $i = 1, 2\}$. There are 36 homomorphisms which are separated into 2 subset: repeated ($f_r = \{f | f(1) = f(3)\}$) and distinct ($f_d = \{f | f(1) \neq f(3)\}$ or $f_d = \{f | f$ is injective$\}$). Hence, $f = f_r \cup f_d$, $|f_r| = 12$ and $|f_d| = 24$. Next, we consider the vertex that is belong to the set f_d. We can choose four vertices from the set f_d by the following algorithm.

Step 1: Choose a vertex $f_1 = (f_1(1)\ f_1(2)\ f_1(3))$ in f_d, which $f_1(i) \neq 1$ for all i.
An example of f_1 is $(4\ 3\ 2)$.

Step 2: After f_1 is obtained, f_2 is chosen with conditions:

- $f_2(i) = 1$ when $f_1(i) = 2$. Then, $f_2(3) = 1$.
- $f_2(i) \neq f_1(i)$ for all i. Then, $f_2(1) = 3$ and $f_2(2) = 4$.

An example of f_2 is $(3\ 4\ 1)$.

Step 3: After f_1 and f_2 are obtained, f_3 is chosen with conditions:

- $f_3(i) = 1$ when $f_1(i) = 3$, and $f_3(i) = 2$ when $f_2(i) = 3$. Then, $f_3(1) = 2$, and $f_2(2) = 1$.
- $f_3(i) \neq f_1(i)$ and $f_3(i) \neq f_2(i)$. Then, $f_3(3) = 4$.

An example of f_3 is $(2\ 1\ 4)$.

After f_1, f_2, and f_3 are obtained, f_4 will be fixed. An example of f_4 is $(1\ 2\ 3)$.

Next, the adjacency of f_i, where $i = 1, 2, 3, 4$, is considered. The vertices f_1 and f_2 are adjacent because $\{f_1(i), f_2(i)\}$ is an edge of K_4 for all $i \in P_3$. Also, the vertices f_i and f_j are connected for all $i \neq j$. Hence, each vertex connects to each other. A graph $(G(V, E))$ with $V = \{f_1, f_2, f_3, f_4\}$ and $E = \{\{f_i, f_j\} | i \neq j\}$ is a complete graph (K_4) and also a subgraph of $P_3 \diamond K_4$ as shown in Figure 4.

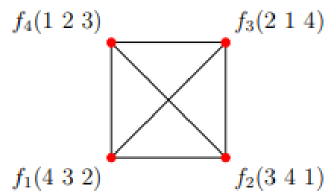

Figure 4: A subgraph of $P_3 \diamond K_4$ presenting four team round robin tournament.

Moreover, the meaning of each vertex f_i is a schedule of team i such as $f_1 = (4\ 3\ 2)$ means the team 1 having three competitions against team $4, 3, 2$, respectively.

We observe that a complete subgraph can represent a schedule of a round robin tournament because a graph in Figure 3 is as same as a graph in Figure 4.

However, a schedule of round robin tournament is not an unique. There are six different schedules of four team round robin tournament. All graphs that represent a schedule of four team round robin tournament are complete subgraph of $P_3 \diamond K_4$.

3 Results

In this section, we describe some properties of the diamond product of $P_{n-1} \diamond K_n$, which are about connectivity, the diameter, and subgraph. We also represent the application of the diamond product of $P_{n-1} \diamond K_n$, which is applied to the scheduling of round robin tournament.

Theorem 3.1. *$P_{n-1} \diamond K_n$ is a connected graph.*

Proof. Let $V(P_{n-1} \diamond K_n) = \text{Hom}\,(P_{n-1}, K_n) = \{v_i = (f_i(1)\ f_i(2)\ \dots\ f_i(n-1)) | f_i : V(P_{n-1}) \to K_n$ and $\{f_i(m), f_i(m+1)\} \in E(K_n)$, where $m = 1, 2, \dots, n-2\}$. Considering the value of n is separated into two cases:

- **Case 1:** $n = 3$, that is the smallest value of n. $V(P_2 \diamond K_3) = \{(1\ 2),\ (1\ 3),\ (2\ 1),\ (2\ 3),\ (3\ 1),\ (3\ 2)\}$. There are two possibilities of randomly choosing two vertices from $V(P_2 \diamond K_3)$. Let $v_i = (f_i(1)\ f_i(2))$ and $v_j = (f_j(1)\ f_j(2))$ be vertices in $V(P_2 \diamond K_3)$. The first case is $f_i(1) \neq f_j(1)$ and $f_i(2) \neq f_j(2)$. It means that v_i and v_j are adjacent. The second case is either $f_i(1) = f_j(1)$ or $f_i(2) = f_j(2)$. It means that v_i and v_j are not adjacent. Without loss of generality, assume that $f_i(1) = f_j(1)$. There exists a vertex $v_x = (f_x(1)\ f_x(2))$ that is adjacent to v_i and v_j. A vertex v_x can be generated by the following algorithm. Since $f_i(2) \neq f_j(2)$, $f_x(2)$ can be chosen from the set $V(K_3) - \{f_i(2),\ f_j(2)\}$. So, there is only one choice because $|V(K_3)| = 3$,

$|\{f_i(2),\ f_j(2)\}| = 2$, and $|V(K_3) - \{f_i(2),\ f_j(2)\}| = 1$. Thus, $f_x(2) = f_i(1) = f_j(1)$. Next, $f_x(1)$ can be chosen from the set $V(K_3) - \{f_i(1),\ f_j(1),\ f_x(2)\}$. There are two choices because $|V(K_3)| = 3$ and $|\{f_i(1),\ f_j(1),\ f_x(2)\}| = 1$ such that $|V(K_3) - \{f_i(1),\ f_j(1),\ f_x(2)\}| = 2$. Hence, there exists $v_x \in V(P_2 \diamond K_3)$, which is adjacent to v_i and v_j. Therefore, v_i and v_j are connected.

- **Case 2:** $n > 3$, there are two possibilities of randomly choosing two vertices from $V(P_{n-1} \diamond K_n)$. Let $v_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1))$ and $v_j = (f_j(1)\ f_j(2)\ \ldots\ f_j(n-1))$ be vertices in $V(P_{n-1} \diamond K_n)$. The first case is $f_i(k) \neq f_j(k)$, for all $k \in V(P_{n-1})$. It means that v_i and v_j are adjacent. The second case is either $f_i(k) = f_j(k)$, for some $k \in V(P_{n-1})$. It means that v_i and v_j are not adjacent. However, there exists a vertex v_x that is adjacent to v_i and v_j. A vertex v_x can be generated by the following algorithm. The first step is to find $f_x(1)$. Since v_x is adjacent to v_i and v_j, $f_x(1) \neq f_i(1)$ and $f_x(1) \neq f_j(1)$. So, $f_x(1) \in V(K_n) - \{v_i(1),\ v_j(1)\}$. The set $V(K_n) - \{f_i(1),\ f_j(1)\}$ is not an empty set because $|V(K_n)| = n$ and $1 \leq |\{f_i(1),\ f_j(1)\}| \leq 2$ such that $0 < n-2 \leq |V(K_n) - \{f_i(1),\ f_j(1)\}| \leq n-1$. So, $f_x(1)$ exists. Next, $f_x(k)$, for any $k \in V(P_{n-1}) - 1$ can be chosen from the set $V(K_n) - \{f_i(k),\ f_j(k),\ f_x(k-1)\}$. The set $V(K_n) - \{f_i(k),\ f_j(k),\ f_x(k-1)\}$ is not an empty set because $|V(K_n)| = n$ and $1 \leq |\{f_i(k),\ f_j(k),\ f_x(k-1)\}| \leq 3$ such that $0 < n-3 \leq |V(K_n) - \{f_i(k),\ f_j(k),\ f_x(k-1)\}| \leq n-1$. So, $f_x(k)$ exists for any $k \in V(P_{n-1}) - 1$. By the above steps, $v_x \in V(P_{n-1} \diamond K_n)$ can be form and v_x is adjacent to v_i and v_j for $i \neq j$. Hence, v_i and v_j are connected for $i \neq j$. Therefore, $P_{n-1} \diamond K_n$ is a connected graph.

□

Theorem 3.2. *The diam$(P_{n-1} \diamond K_n)$ is equal to 2.*

Proof. Let v_i and v_j be vertices of $P_{n-1} \diamond K_n$. $v_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1))$ and $v_j = (f_j(1)\ f_j(2)\ \ldots\ f_j(n-1))$. From Theorem 3.1, a path that connects any two vertices of $P_{n-1} \diamond K_n$ can be found. There are two possibilities of randomly choosing two vertices from $V(P_{n-1} \diamond K_n)$.

- **Case 1:** v_i is adjacent to v_j. So, the distance $(d(v_i, v_j))$ is equal to 1.
- **Case 2:** v_i is not adjacent to v_j. There exists v_x that is adjacent to v_i and v_j for $i \neq j$. So, the distance $(d(v_i, v_j))$ is equal to 2 because of the path $v_i - v_x - v_j$. Therefore, the diam$(P_{n-1} \diamond K_n)$ is equal to 2.

□

Theorem 3.3. *A complete graph K_n is a subgraph of $P_{n-1} \diamond K_n$, where n is an integer greater than 2.*

Proof. Let $P_{n-1} \diamond K_n$ be a diamond product graph which has the vertex set $V(P_{n-1} \diamond K_n) = \text{Hom}(P_{n-1}, K_n) = \{f | f : V(P_{n-1}) \to K_n \text{ and } \{f(i), f(i+1)\} \in E(K_n)$, where $i = 1, 2, \ldots, n-2\}$ and the edge set $E(P_{n-1} \diamond K_n) = \{\{f, g\} | \{f(i), g(i)\} \in E(K_n)$, where $i = 1, 2, \ldots, n-1\}$. Let G be a graph that has the vertex set $V(G) = \{f_1, f_2, \ldots, f_n\}$ and $|V(G)| = n$. Define $f_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1)) \in \text{Hom}(P_{n-1}, K_n)$. $\{f_i, f_j\}$ is an edge of a graph G if $\{f_i(k), f_j(k)\} \in E(K_n)$ or $f_i(k) \neq f_j(k)$ for all $k = 1, 2, \ldots, n-1$. Considering $k = 1$, $f_1(1),\ f_2(1),\ f_3(1),\ \ldots,\ f_n(1)$ can be formed by the permutation of $\{1,\ 2,\ \ldots,\ n\}$. Thus, $f_i(1) \neq f_j(1)$, when $i \neq j$. Considering $k = 2$, $f_1(2),\ f_2(2),\ f_3(2),\ \ldots,\ f_n(2)$ can be formed by the permutation of $\{1,\ 2,\ \ldots,\ n\}$ with the additional condition $f_i(2) \neq f_i(1)$ for $i = 1, 2, \ldots, n$. By the same idea, $f_1(k),\ f_2(k),\ \ldots,\ f_n(k)$ can be formed by the permutation of $\{1, 2, \ldots, n\}$ with the additional condition $f_i(k) \neq f_i(k-1)$

for $i = 1, 2, \ldots, n$. By the above algorithm, $f_1,\ f_2,\ \ldots,\ f_n$ are obtained and $f_i \in V(P_{n-1} \diamond K_n)$ and $\{f_i, f_j\} \in E(P_{n-1} \diamond K_n)$ for $i \neq j$ and $E(G) = \{\{f_i, f_j\} | i \neq j\}$. Thus, G is a complete graph on n vertices or K_n. Moreover, $V(G) \subset V(P_{n-1} \diamond K_n)$ and $E(G) \subset E(P_{n-1} \diamond K_n)$. Therefore, G is a complete subgraph of $P_{n-1} \diamond K_n$. □

Corollary 3.1. *K_n is the largest complete subgraph of $P_{n-1} \diamond K_n$.*

Proof. Let K_n be a complete subgraph of $P_{n-1} \diamond K_n$. Let $g \in V(P_{n-1} \diamond K_n)$ and $g \notin V(K_n)$. Since $f_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1)) \in V(K_n)$, where $i = 1, 2, \ldots, n$ and $f_1(k),\ f_2(k),\ \ldots,\ f_n(k)$ are generated by permutating the set $\{1, 2, \ldots, n\}$. Assume that G is a complete graph, where $V(G) = V(K_n) \cup g$. Since $g = (g_1\ g_2\ \ldots g_{n-1}) \in V(P_{n-1} \diamond K_n)$, $g_k \in \{1, 2, \ldots n\}$. However, $\{f_1(k),\ f_2(k),\ \ldots,\ f_n(k)\} = \{1, 2, \ldots, n\}$. Hence, $g_k = f_i(k)$ for some $i \in \{1, 2, \ldots, n\}$. Then $\{g, f_i\} \notin E(P_{n-1} \diamond K_n)$. Thus, a graph G is not a complete graph. Therefore, K_n is the largest complete subgraph of $P_{n-1} \diamond K_n$. □

Theorem 3.4. *Let G be a subgraph of $P_{n-1} \diamond K_n$, where n is an even number greater than 2. G is a schedule of n team round robin tournament if and only if G is a complete graph and satisfies following conditions:*

- *C1: $V(G) = \{f_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1)) | f_i : V(P_{n-1}) \to V(K_n) - \{i\}$ is an injective function, $f_i(k) \neq i$, for all $k = 1, 2, 3, \ldots, n-1$ and $i = 1, 2, 3, \ldots, n\}$.*
- *C2: $f_i(k) = j$ if and only if $f_j(k) = i$ for $i, j \in V(K_n)$ and $k \in V(P_{n-1})$.*

Proof. Let G be a subgraph of $P_{n-1} \diamond K_n$ and has the vertex set $V(G) = \{f_1, f_2, \ldots, f_n\}$.
($\Rightarrow$). Suppose that G is a schedule of n teams round robin tournament. Each vertex of G is a schedule of each team. $f_i = (f_i(1)\ f_i(2)\ \ldots\ f_i(n-1))$ is a sequence of teams againsting to team i and $f_i(k)$ are distinct for all $k = 1, 2, \ldots, n-1$. Thus, $f_i \in f : \{1, 2, \ldots, n-1\} \to \{1, 2, \ldots, n\} - \{i\}$ and f_i is an injective function. Moreover, $f_1(k), f_2(k), \ldots, f_n(k)$ represent games of each team in the week k. Thus, $f_i(k) = j$ if and only if $f_j(k) = i$, for all $k \in \{1, 2, \ldots, n-1\}$. Therefore, the graph G is a complete graph which satisfies conditions $C1$ and $C2$.
($\Leftarrow$). Suppose that G is a complete graph which satisfies conditions $C1$ and $C2$. Let G be a subgraph of $P_{n-1} \diamond K_n$ with $V(G) = \{f_1, f_2, \ldots, f_n\}$, where $f_i : V(P_{n-1}) \to V(K_n) - \{i\}$. f_i can be generated by the following algorithm.

The algorithm is considered into two cases that are $\frac{n}{2}$ is an even number and $\frac{n}{2}$ is an odd number:

Case: $\frac{n}{2}$ is an even number.

Step 1: Consider $k \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$. Then,

$$f_i(k) = \begin{cases} \frac{n}{2} + i + k - 1, & \text{if} \quad \frac{n}{2} + i + 1 \leq n, \\ i + k - 1, & \text{if} \quad \frac{n}{2} + i + 1 > n, \end{cases}$$

for $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$ and $f_i(k) = j$, for $i \in \left\{\frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n\right\}$ if $f_j(k) = i$ for $j \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$.

After $f_i(1), f_i(2), \ldots, f_i\left(\frac{n}{2}\right)$, for all i are obtained, we observe that

$$f_i : \left\{1, 2, \ldots, \frac{n}{2}\right\} \rightarrow \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\},$$

where $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$ and

$$f_i : \left\{1, 2, \ldots, \frac{n}{2}\right\} \rightarrow \left\{1, 2, \ldots, \frac{n}{2}\right\},$$

where $i \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$ are bijective functions.

Step 2: Consider $k \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n-1\right\}$. In case of $f_i(k)$, where $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$, $f_i(k)$ is constructed by permutating the number in the set $\left\{1, 2, \ldots, \frac{n}{2}\right\}$ with two conditions:

$$f_i(k) \neq i \text{ and } f_i(k) = j,$$

if and only if $f_j(k) = i$. In case of $f_i(k)$, where $i \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$ is also contructed by permutating the number in the set $\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n-1\right\}$ with two conditions:

$$f_i(k) \neq i \text{ and } f_i(k) = j,$$

if and only if $f_j(k) = i$. After $f_i\left(\frac{n}{2}+1\right), f_i\left(\frac{n}{2}+2\right), \ldots, f_i(n-1)$, for all i, are obtained, we observe that

$$f_i : \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n-1\right\} \rightarrow \left(\left\{1, 2, \ldots, \frac{n}{2}\right\} - \{i\}\right),$$

where $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$ and

$$f_i : \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n-1\right\} \rightarrow \left(\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\} - \{i\}\right),$$

where $i \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$ are bijective functions.

Generally, in case of step 2 will be as same as the generating of the case $\frac{n}{2}$ that is obtained before. The example shows a $f_i : V(P_3) \rightarrow V(K_4)$ constructed by this algorithm. Suppose that $n = 4$ and $k \in \{1, 2, 3\}$.

Step 1: Consider the case $k \in \{1, 2\}$, $f_i(1) = \frac{4}{2} + i$ for $i \in \{1, 2\}$ and $f_i(1) = j$, for $i \in \{3, 4\}$ if $f_j(1) = i$.

- For $k = 1$, $f_1(1) = 2 + 1 + 1 - 1 = 3$, $f_2(1) = 2 + 2 + 1 - 1 = 4$, $f_3(1) = 1$ and $f_4(1) = 2$.
- For $k = 2$, $f_1(2) = 2 + 1 + 2 - 1 = 4$, $f_2(2) = 2 + 2 - 1 = 3$, $f_3(2) = 2$ and $f_4(2) = 1$.

Step 2: Consider the case $k \in \{3\}$,

- For $i = 1, 2$, $f_3(1) = 2$ and $f_3(2) = 1$.
- For $i = 3, 4$, $f_3(3) = 4$ and $f_3(4) = 3$.

So, we can represent f_1, f_2, f_3, and f_4 as shown in Table 2:

Table 2: A result of constructed f_1, f_2, f_3, and f_4.

k	1	2	3
$f_1(k)$	3	4	2
$f_2(k)$	4	3	1
$f_3(k)$	1	2	4
$f_4(k)$	2	1	3

Since f_1, f_2, f_3, and f_4 are satisfied conditions $C1$ and $C2$, it follows that f_1, f_2, f_3, and f_4 are vertices of a complete graph that is a subgraph of $P_3 \diamond K_4$. Therefore, each f_i is the schedule of team i in 4 teams round robin tournament.

Case: $\frac{n}{2}$ is an odd number.

Step 1: Consider $k \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$ and case of i separated into two cases: $\left\{1, 2, \ldots, \frac{n}{2}\right\}$ and $\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$.

Step 1.1: Consider $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$, we add a dummy point (d) into the set $\left\{1, 2, \ldots, \frac{n}{2}\right\}$. It becomes the set $\left\{1, 2, \ldots, \frac{n}{2}, d\right\}$, which has $\frac{n}{2}+1$ elements. Since $\frac{n}{2}+1 < n$, $f_1, f_2, \ldots, f_{\frac{n}{2}}, f_d$ are obtained by this algorithm with smaller number n, then $f_1, f_2, \ldots, f_{\frac{n}{2}}, f_d$ are rearranged by a condition: $f_k(k) = d$.

Step 1.2: Consider $\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$, we apply the idea of Step 1.1 by adding a dummy point (e) into the set $\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$ to obtain $f_{\frac{n}{2}+1}, f_{\frac{n}{2}+2}, \ldots, f_n, f_e$. Then, $f_{\frac{n}{2}+1}, f_{\frac{n}{2}+2}, \ldots, f_n, f_e$ are rearranged by a condition: $f_{\frac{n}{2}+k}(k) = e$.

Step 1.3: After $f_1, f_2, \ldots, f_n$ are obtained, we have to change a value of $f_i(k) = d$ or $f_i(k) = e$ with a condition: if $f_i(k) = d$ and $f_j(k) = e$, then $f_i(k) = j$ and $f_j(k) = i$, for $i, j \in \{1, 2, \ldots, n\}$.

Step 2: Consider $k \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n-1\right\}$. Then,

$$f_i(k) = \begin{cases} i+k, & \text{if} \quad i+k \leq n, \\ i+k-\frac{n}{2}, & \text{if} \quad i+k > n, \end{cases}$$

for $i \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$ and the image of $f_i(k)$ is a subset of $\left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$. $f_i(k)$ are fixed, for $i \in \left\{\frac{n}{2}+1, \frac{n}{2}+2, \ldots, n\right\}$, and $f_i(k) = j$ if $f_j(k) = i$, where $j \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$.

The example shows a $f_i : V(P_5) \to V(K_6)$ constructed by this algorithm. Suppose that $n = 6$ and $k \in \{1, 2, 3, 4, 5\}$. The example shows a $f_i : V(P_5) \to V(K_6)$ constructed by this algorithm. Suppose that $n = 6$ and $k \in \{1, 2, 3, 4, 5\}$.

Step 1: Consider $k \in \{1, 2, 3\}$ and case of i separated into two cases: $\{1, 2, 3\}$ and $\{4, 5, 6\}$.

Step 1.1: Consider $i \in \{1, 2, 3\}$, we add a dummy point (d) into the set $\{1, 2, 3\}$. It becomes the set $\{1, 2, 3, d\}$, which has 4 elements. We apply the example above. Then, we get Table 3:

Table 3: An example of constructed f_1, f_2, f_3, and f_d.

k	1	2	3
$f_1(k)$	3	d	2
$f_2(k)$	d	3	1
$f_3(k)$	1	2	d
$f_d(k)$	2	1	3

and we rearrange f_1, f_2, f_3, f_d by a condition: $f_k(k) = d$. So, we swap the first column and the second column. Then, we get Table 4:

Table 4: A result after rearranging f_1, f_2, f_3, and f_d.

k	1	2	3
$f_1(k)$	d	3	2
$f_2(k)$	3	d	1
$f_3(k)$	2	1	d
$f_d(k)$	1	2	3

Step 1.2: Consider $\{4, 5, 6\}$, we apply the idea of Step 1.1 by adding a dummy point (e) into the set $\{4, 5, 6\}$ to obtain f_4, f_5, f_6, f_e. Then, we get Table 5:

Table 5: An example of constructed f_4, f_5, f_6, and f_e.

k	1	2	3
$f_4(k)$	6	e	5
$f_5(k)$	e	6	4
$f_6(k)$	4	5	e
$f_e(k)$	5	4	6

and we rearrange f_4, f_5, f_6, f_e by a condition: $f_{\frac{n}{2}+k}(k) = e$. So, we swap the first column and the second column. Then, we get Table 6:

Table 6: A result after rearranging f_4, f_5, f_6, and f_e.

k	1	2	3
$f_4(k)$	e	6	5
$f_5(k)$	6	e	4
$f_6(k)$	5	4	e
$f_e(k)$	4	5	6

Step 1.3: After $f_1, f_2, \ldots, f_6$ are obtained, we have to change a value of $f_i(k) = d$ or $f_i(k) = e$ with a condition: if $f_i(k) = d$ and $f_j(k) = e$, then $f_i(k) = j$ and $f_j(k) = i$, for $i, j \in \{1, 2, \ldots, 6\}$. Then, we get Table 7:

Table 7: The merge of f_1, f_2, f_3, f_4, f_5, and f_6.

k	1	2	3
$f_1(k)$	4	3	2
$f_2(k)$	3	5	1
$f_3(k)$	2	1	6
$f_4(k)$	1	6	5
$f_5(k)$	6	2	4
$f_6(k)$	5	4	3

Step 2: Consider $k \in \{4, 5\}$. Then,

$$f_i(k) = \begin{cases} i + k, & \text{if} \quad i + k \leq 6, \\ i + k - 3, & \text{if} \quad i + k > 6, \end{cases}$$

for $i \in \{1, 2, 3\}$ and we get Table 8:

Table 8: A result after adding value of $f_1(k), f_2(k), f_3(k)$, where $k = 4, 5$.

k	1	2	3	4	5
$f_1(k)$	4	3	2	5	6
$f_2(k)$	3	5	1	6	4
$f_3(k)$	2	1	6	4	5
$f_4(k)$	1	6	5		
$f_5(k)$	6	2	4		
$f_6(k)$	5	4	3		

Then, $f_i(k)$ are fixed, for $i \in \{4, 5, 6\}$, and $f_i(k) = j$ if $f_j(k) = i$, where $j \in \{1, 2, 3\}$. Then, we get Table 9:

Table 9: A result of constructed f_1, f_2, f_3, f_4, f_5, and f_6.

k	1	2	3	4	5
$f_1(k)$	4	3	2	5	6
$f_2(k)$	3	5	1	6	4
$f_3(k)$	2	1	6	4	5
$f_4(k)$	1	6	5	3	2
$f_5(k)$	6	2	4	1	3
$f_6(k)$	5	4	3	2	1

Since f_1, f_2, f_3, f_4, f_5 and f_6 are satisfied conditions $C1$ and $C2$, it follows that f_1, f_2, f_3, f_4, f_5 and f_6 are vertices of a complete graph that is a subgraph of $P_5 \diamond K_6$. Therefore, each f_i is the schedule of team i in 6 teams round robin tournament.

We observe that $f_i : V(P_{n-1}) \to V(K_n)$ obtained by this algorithm is injective, $f_i(k) \neq i$ and $f_i(k) = j$ if and only if $f_j(k) = i$ for all $k \in \{1, 2, \ldots, n-1\}$. Moreover, f_i connects to f_j if $i \neq j$. It means that $\{f_i, f_j\}$ is an edge of the graph G. So, $G(V(G), E(G))$ is a complete subgraph of $P_{n-1} \diamond K_n$.

Furthermore, $f_i \in V(G)$ represents the schedule of team i because $f_i(k) \neq i$ and f_i is an injective function. It means that a team i cannot be against to team i and a team i must be against to other teams. The meaning of $f_i(k)$ is a team i having a competition against team $f_i(k)$ at match number k. If $f_i(k) = j$, then $f_j(k) = i$ because of a competition of team i at match number k. Therefore, G is a schedule of n team round robin tournament. □

Lemma 3.1. *A graph $P_{n-1} \diamond K_n$ contains all different schedules of n teams round robin tournament.*

Proof. From Theorem 3.4, all schedule of n teams round robin tournament are complete subgraphs of $P_{n-1} \diamond K_n$. Therefore, the graph $P_{n-1} \diamond K_n$ contains all different schedules of n teams round robin tournament. □

4 Conclusions

In conclusion, a graph $P_{n-1} \diamond K_n$ is a connceted graph and it has the diameter being 2. A subgraph of the diamond product of a path graph (P_{n-1}) and a complete graph (K_n) is the one of solutions of $2n$ teams round robin tournament scheduling problem. The diamond product of graphs $(P_{n-1} \diamond K_n)$ also contains all solutions of $2n$ teams round robin tournament scheduling problem. Hence, the the diamond product of graphs $(P_{n-1} \diamond K_n)$ can be applied to find all solutions or a solution of $2n$ teams round robin tournament scheduling problem. Moreover, this paper presents an algorithm to find one of solutions of $2n$ teams round robin tournament. However, another solution can be found by permutating an algorithm solution.

Acknowledgement This research was funded by College of Industrial Technology, King Mongkut's University of Technology North Bangkok (Grant No. Res CIT0632/2023).

Conflicts of Interest The authors declare no conflict of interest.

References

[1] K. A. Bhat & G. Sudhakara (2018). Commuting graphs and their generalized complements. *Malaysian Journal of Mathematical Sciences*, *12*(1), 63–84.

[2] J. Damnernsawat (2007). *Some properties of diamond product of Cayley graphs of groups*. PhD thesis, The Graduate School Chiang Mai University, Chiang Mai, Thailand.

[3] M. Goerigk, R. Hoshino, K. Kawarabayashi & S. Westphal (2014). Solving the traveling tournament problem by packing three-vertex paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28 pp. 2271–2277. Québec, Canada. https://doi.org/10.1609/aaai.v28i1.9031.

[4] R. Hoshino & K. Kawarabayashi (2013). Graph theory and sports scheduling. *Notices of the American Mathematical Society*, *60*(6), 726–731. http://dx.doi.org/10.1090/noti1010.

[5] T. Jiarasuksakun, T. Rutjanisarakul & W. Thongjua (2010). Diamond product of two common complete bipartite graphs. *East-West Journal of Mathematics*, *12*(2), 197–205.

[6] K. Malhotra, D. Gupta, S. Goel & A. K. Tripathi (2022). Bi-objective flow shop scheduling with equipotential parallel machines. *Malaysian Journal of Mathematical Sciences*, *16*(3), 451–470. https://doi.org/10.47836/mjms.16.3.04.

[7] C. C. Ribeiro & S. Urrutia (2007). Heuristics for the mirrored traveling tournament problem. *European Journal of Operational Research*, *179*(3), 775–787. https://doi.org/10.1016/j.ejor.2005.03.061.

[8] T. Rutjanisarakul & T. Jiarasuksakun (2017). A sport tournament scheduling by genetic algorithm with swapping method. *arXiv preprint arXiv:1704.04879*, pp. 1–8. https://doi.org/10.48550/arXiv.1704.04879.

[9] J. Thomkeaw & S. Arworn (2012). Endomorphism monoid of C_{2n+1} book graphs. *Thai Journal of Mathematics*, *7*(2), 319–327.

[10] D. B. West (2001). *Introduction to graph theory*. Prentice Hall, Upper Saddle River.