Quest Journals
Journal of Research in Applied Mathematics
Volume 12 ~ Issue 8 (August 2026) pp: 40-43
ISSN (Online): 2394-0743 ISSN (Print): 2394-0735
www.questjournals.org

Research Paper

Graph Theory in Operations Research

Dr.B.K.Pedge(1)
Gramin Technical & Management Campus, Vishnupuri, Nanded

Dr.S.R.Gadhe(2)
N.W.Mahavidyalaya, Akhada Balapur Dist Hingoli.

Abstract:
Game theory is applied to problems in engineering, economics as well as war science to find the optimal way of performing certain tasks in a competitive environment.

Received 11 Aug., 2026; Revised 22 Aug., 2026; Accepted 24 Aug., 2026

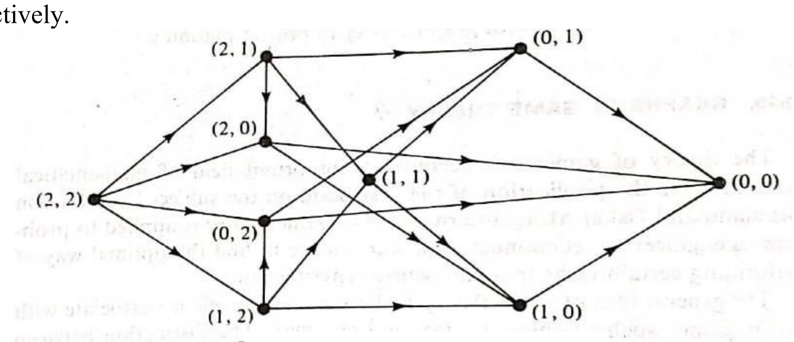
I. Introduction :

The general idea of game theory is the same as the one we associate with parlor games such as chess, bridge, and checkers. The distinction between a puzzle and a game is that in a game one plays against one or more human opponents, whereas a puzzle involves a solitary effort to solve a problem.

A game may be played between two persons, such as chess, or among more than two persons, such as poker. The former is called a two-person game and the latter an n-person game. Another classification of games is based on whether or not an element of randomness is introduced, such as by dice or cards. A third element in categorizing a game is whether or not a player has complete information on the position of a game at every move. A game such as chess, in which each player knows exactly where the game stands is called a perfect-information game. Bridge, in which one does not know what cards the other players have, is an imperfect-information game. A game is called finite if each player has a finite number of choices available at each move and the game must end after a finite number of moves. An infinite game is one in which a player chooses a move from an infinite set of moves.

We shall confine ourselves to the study of two-person, perfect-information, finite games without chance moves. A digraph is a natural representation of such a game. The vertices represent the positions (also called states) in the game and the edges represent the moves. There is a directed edge from vertex v_i, to v_j, if and only if the game can be transformed from position (state) v_i, to v_j to by a move permissible under the rules of the game. As an example, let us look at a very simple game. It is a simplified version of a game called nim.

Simplified Nim: Two piles of sticks are given and players A and B take turns, each taking any number of sticks from any one pile. The player who takes the last stick wins, and since the finite quantity of sticks will eventually be exhusted, it is obvious that the game allows no draw. As a further simplification, let us start with two piles containing two sticks each. The complete game is described by the digraph in Fig. 1-1. Each state of the game is described by an ordered pair of labels (x, y), indicating the number of sticks in the first and the second pile, respectively.



I **Fig.1.1 Simplified game of nim**

Let us observe some properties of such a game digraph (a digraph representing a two-person, perfect-information, finite game without chance moves):

1. The digraph has a unique vertex with a zero in-degree. This vertex represents the starting position in the game and is therefore called the starting vertex. Vertex (2, 2) in Fig. 1-1 is the starting vertex.
2. There are one or more vertices with zero out-degree. These correspond to the closing positions in the game, and are called the closing vertices. Vertex (0, 0) is the closing vertex in Fig. 1-1.
3. A game digraph is a connected, acyclic digraph. A directed circuit would imply that the game could go on indefinitely. (In practice, in a game such as chess, where the game may return to a state, endless matches are prevented by means of a rule that after a certain number of repetitions of the same move, the game is declared stalemated.)
4. Each directed path from the starting vertex to a closing vertex represents one complete play of the game. This path consists of edges representing the moves of the two players alternately.

The most important question in a game is the following: When and how can player choose his moves so that he is certain of winning? We shall first answer this question for the specific game in Fig. 1-1, and then generalize

Let us call a position "won" if the player who brought the game to this position can force a victory. Conversely, a position is dubbed "lost" if the player who brought the game to this position can be forced to lose. In keeping with this characterization of vertices, the closing vertex in Fig. 1-1 is to be marked as won, because the player who brought the game to this position is the winner. Having marked this vertex as won, let us use the following procedure to mark the remaining vertices as won or lost.

Mark an unmarked vertex won if all its successors are marked lost, and mark an unmarked vertex lost if at least one of its successors is marked won. (This is because it is assumed that each player is intelligent and makes the best possible move at each stage.) This results in vertices (0, 0), (1, 1), and (2,2) being marked as won and the remaining as lost. And thus the player who makes the second move has the winning strategy, since he can force his opponent to move to the vertices marked as lost.

To generalize the foregoing method of finding a winning strategy, let us introduce the concept of kernel in a digraph.

Kernel of a Digraph: A set of vertices K in a digraph G is called a kernel (or nucleus) of G if

1. No two vertices in K are joined by an edge.
2. Every vertex v not in K has an edge directed from v to some vertex in *K*.

Conditions 1 and 2 correspond respectively to definitions of an independent set and a dominating set in an undirected graph. What are some types of digraphs that have kernels?

THEOREM 1.1

Every acyclic digraph has a unique kernel.

Proof: The theorem will be proved by a constructive procedure, at the end of which all vertices forming the kernel will be painted red. Let G be the given acyclic digraph. According to **Theorem 1.2**, G must have at least one vertex with zero out-degree. Let V_i, be the set of all vertices in G with zero out-degree. Since these vertices must all be in the kernel of G, paint them red.

Next, let W_i, be the set of all those vertices in G from which there is at least one directed edge to some vertex in V_1. Clearly, no vertex in W_1, can be included in the kernel. Delete from G all vertices in W_1, (together with the edges incident on them, of course), and thus obtain subgraph (G-W_1).

Subgraph (G-W_1).is also acyclic. Let V_2 be the set of all vertices with zero out-degree in (G-W_1). Since in the original digraph G no vertex in set V_2 had a zero out-degree, every vertex in V_2 had to have at least one edge going to some vertex in W_1. Moreover, in digraph G no vertex in V_2 could have been adjacent to any vertex in V_1. Nor could any vertex in V_2 have been adjacent to any other vertex in V_2, because the out-degree of each vertex in set V_2 of (G-W_1) is zero. Thus we conclude that every vertex in V_2 must also be included in the kernel of G and therefore be painted red.

This procedure is continued till every vertex in G is either deleted or painted red. The unique set of vertices painted red constitutes the kernel of the digraph.

As an illustration, let us find the kernel in the acyclic digraph in Fig. 1.1. It is easily seen that the set of three vertices marked (0, 0), (1, 1), and (2,2) is the kernel.

Let A be the player who makes the first move in the game and B be the player who makes the second move. Assuming that the rule of the game is such that the player who is able to make the last possible move in the game is always the winner, we have the following important result.

THEOREM 1.2

In the game digraph if the starting vertex is not in the kernel K, then player A is assured of a win, and A can win by always selecting vertices in K.

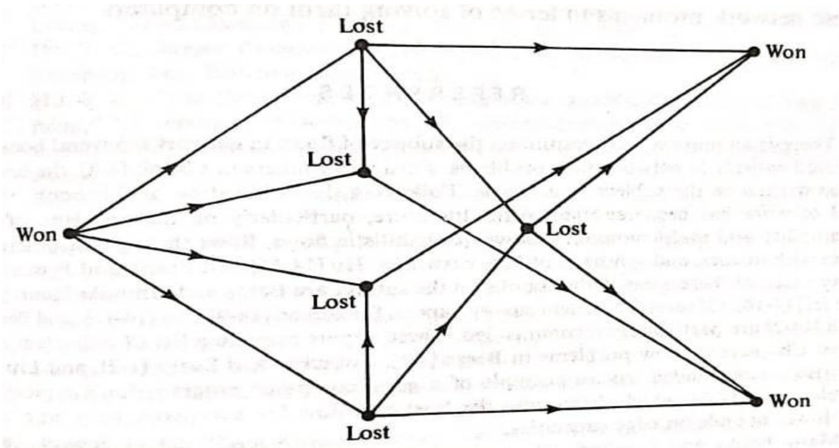
Proof: Since the starting vertex is not in set K, player A can move the game to a vertex x in K. If this vertex x is a closing vertex, A is the winner. If not, the second player B will have to move to some vertex y, which is not in the kernel K. In his next move A can take the game to some vertex in K. The game continues, with B forced to take it out of K and A bringing it back into K. Eventually, the play will be brought to a closing vertex by A, because all closing vertices are in K. Thus A wins the game.

COROLLARY

It follows from the proof of this theorem that if the starting vertex is in the kernel, the second player B has the winning strategy; and B can win by always selecting vertices in the kernel.

In the foregoing analysis the rules of the game were assumed to be such that the player who made the last possible move was always the winner. In many games (such as chess or tic-tac-toe) some of the closing vertices represent a draw and others a win. In such a game choosing vertices from the kernel will only assure a player a win or a draw,

There are also games in which the rule is such that the player who is forced to make the last move is the loser rather than the winner. From such a game digraph, if we remove all edges corresponding to the last moves, the game can be converted to the type in which the winner is the player making the last move. In other words, the game is decided at the time the second-to-the-last moves are made. For example, let us modify the nim game of Fig. 1-1 such that the player forced to take the last stick is the loser. Then erasing the last moves from the digraph, we get Fig. 1-2. The three vertices marked won constitute the kernel in this digraph. Even in this game player B has the winning strategy.



Thus, at least in theory, it is possible to construct the game digraph for any two-person, perfect-information, finite game with no chance moves, and, since the digraph is acyclic, it is also possible to obtain its kernel. Therefore, if each player plays according to his best strategy, the outcome is predetermined. It will be either a draw or a certain win for the player who makes the first move if the starting vertex is not in the kernel, or for the player who makes the second move if the starting vertex is in the kernel. In this sense, every game of this type is either "unfair" or "futile."

In reality the situation is not so bleak as it appears. In most nontrivial games such as checkers or chess, the number of positions (i.e., the vertices, in the game digraph) is so enormous that the game digraph cannot even be stored in the memory unit of any existing or contemplated computer.

This is precisely why in real problems in operations research the theory of games provides an approach rather than a complete analysis. Moreover, graph theory is applicable only to a very special but important class of games.

II. SUMMARY

We have considered three important classes of problems in combinatorial operations research: transportation problems, activity networks, and game theory. These problems can be expressed and solved elegantly as graph- theory problems involving connected and weighted (mostly acyclic) digraphs. From a practical point of view, all these problems are trivial (and so is any combinatorial problem) if the network is small. Many real-life situations, however, consist of huge networks, and therefore it is important to look at these network problems in terms of solving them on computers.

REFERENCES

[1]. BATTERSBY, A., Network Analysis for planning and Scheduling, 2nd ed., St.Martins Press, Inc., New York, 1967.
[2]. Beckenback, E.F., "Network Flow Problems, " Chapter 12 in Applied Combinatorial Mathematics (E.F. Beckenbach, ed.) John Wiley & Sons, Inc., New York, 1964.

[3]. BRAY, T.A. and C. WITZGALL, "Algorithm 336: Netflow," Comm.ACM, Vol.11, Sept.1968, 631-632; corrected in Vol.12, 1969.
[4]. BERGE, C., and A. GHOUILA-A-HOURI, Programming, Games and Transportation Networks (translated by M.Merrington and C. Ramanujacharyula), John Wiley & Sons. Inc., New York, 1965.
[5]. DIMSDALE, B., " Computer Construction of Minimal Project Networks," IBM Systems J., Vol.2, March 1963, 24-36
[6]. PLISCH, D.C., "New Results Concerning separation Theory of Graphs, ," Ph.D. Thesis, University of Wisconsin, Madison Wisc., 1970.
[7]. SAKAROVITCH, M., "The Multicommodity Flow Problem," Ph.D. Dissertation, Operation Research Centre, University of California, Berkeley, 1966.
[8]. SMITH, C.A., "Graphs and Composite Games," J.Combinatorial Theory, Vol.1, 1966, 51-81.
[9]. SCHURMANN, A., "GAN, a system for Generating and Analyzing Activity Networks," Comm. ACM, Vol.11, No.10, Oct.1968, 675-679.
[10]. WANG, H., "Games, Logic and Computers," Sci.Am., Vol.213, No.5, Nov.1965, 98-106.