Integrated Intelligent Research (IIR)
International Journal of Computing Algorithm
Volume: 03, Issue: 01 June 2014, Pages: 1-4
ISSN: 2278-2397

2 Disjoint Paths Problem in Oriented Mesh Network

Antony Xavier[1], Florence Isido. D[1] V.M.Chitra[2]
[1]Department of Mathematics, Loyola College, Chennai, India.
[2]Department of Mathematics, SKR Engineering College, Chennai, India
Email: chithu1234@yahoo.com

Abstract- Given a graph G = (V,E) and two distinct pairs of vertices (s_1, t_1) and (s_2, t_2), the 2- DPP is to find two disjoint pathsP_1 from s_1 to t_1 andP_2 from s_2 to t_2 .In this paper we define orientation in the mesh and identify the pairs of vertices in which 2 DPP is not solvable.

I. INTRODUCTION

The construction of disjoint paths in a given graph G is one of the fundamental problems in graph theory with many applications, for example in the context of routing problems, network reliability and VLSI-design. In the vertex-(edge-)disjoint paths problem, we are given a graph G and two distinct pairs of vertices (s_1, t_1) and (s_2, t_2) in *G* (which are sometimes called terminals) and we have to decide whether *G* has two vertex (edge) disjoint paths. Furthermore, we find such paths if they exist. This is certainly a central problem in algorithmic graph theory and combinatorial optimization. It has attracted attention in the contexts of transportation networks, VLSI layout and virtual circuit routing in high-speed networks or internet. A basic technical problem is to interconnect certain prescribed channels on the chip such that wires belonging to different pins do not touch each other. In this simplest form, the problem mathematically amounts to finding disjoint paths in a graph, each connecting a given pair of vertices. A mesh network is a local area network that employs one of two connection arrangements, full mesh topology or partial mesh topology. In the full mesh topology, each(workstation or other device) is connected directly to each of the others. In the partial mesh topology, some nodes are connected to all the others, but some of the nodes are connected only to those other nodes with which they exchange the most data.

A mesh network is reliable. If one node can no longer operate, all the rest can still communicate with each other, directly or through one or more intermediate nodes. Mesh networks work well when the nodes are located at scattered points that do not lie near a common line. In mesh topology, message sent to the destination can take any possible shortest,easiest route to reach its destination. Internet employs the mesh topology and the message finds its route for its destination. Router works in finding the routes for the messages andin reaching them to their destinations. The mesh derived network provides a very useful communication network pattern to implement a lot of algorithms in many parallel computing systems.

II. OVERVIEW OF THE PAPER

The problem of finding *k* vertex disjoint paths between *k* pairs of terminalsin directed non-planar graphs [2] is NP-complete for *k* = 2, and even in planargraphs [5], if *k*is not fixed. The vertex disjoint paths problem is solvable inpolynomial time in undirected graphs for any fixed *k*[9], and in directed planargraphs for any fixed *k*[10]. Shortest disjoint paths are treated in [3]. Practicallyefficient algorithms for special cases of the disjoint paths problem are surveyedin [8].Woeginger [11] presented a much simpler algorithm but also used a planar embedding of the graph under consideration.Asalready stated by Perl and Shiloach in [7], all these algorithms can be modifiedto solve the 2-V DPP on general planar graphs using a reduction of Itai [1].

For generalundirected graphs, Ohtsuki [6], Seymour, Shiloach and Thomassenindependently found the first polynomial time algorithms for the 2-V DPP.Seymour considered the decision version of the problem. The running time ofOhtsuki and Shiloach's algorithm is *O* (*nm*). By modifying Shiloach's algorithm,Khuller et.al [4] could prove a running time of $O(n^2)$ for the 2-V DPP. In this paper we have given orientation to mesh network and discussed about when the 2DPP is not solvable.

III. PRELIMINARIES

Definition

Graph:A Graph *G* is a finite non- empty set of objects called vertices together with a set of unordered pairs of distinct vertices of *G*, called edges.The vertex set and the edge set of *G* are denoted by *V* (*G*) and *E* (*G*) respectively.

Degree of a vertex: The *degree of a vertex v* in a graph *G*, denoted by $\deg_G v$, is the number of edges incident with *v*.
*Directed Graph:*A graph in which each graph edge is replaced by a directed graph edge is called a *directed graph*.A directed graph having no multiple edges or loops is called a *simpledirected graph.*

Underlying Graph: It is the graph obtained from directed graph by replacing its directed edge by undirected edge.

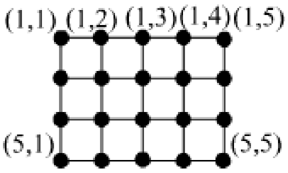
*Strongly Connected Graph:*Two vertices *x* and *y* of *D* are said to be strongly connected if there are both (*x*,*y*)-path and (*y*,*x*)-path in *D*. A digraph *D* is said to be *stronglyConnected* if every pair of vertices in *D* is strongly connected.

Mesh: Let P_n denote a path on n vertices. For $m, n \geq 2, P_m \times P_n$ isdefined as the two dimensional mesh with *m* rows and *n* columns. It is denotedby $M_{m\times n}$.

Remark 3.1 The vertex of$M_{m\times n}$ in i^{th} row and j^{th} column is denoted by(*i*,*j*),
$1 \leq i \leq m, 1 \leq j \leq n$.

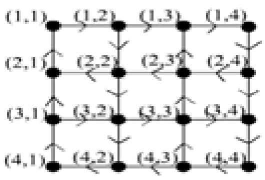
Let us introduce the orientation in the mesh $M_{n\times n}$ as follows.

International Journal of Computing Algorithm
Volume: 03, Issue: 01 June 2014, Pages: 1-4
ISSN: 2278-2397



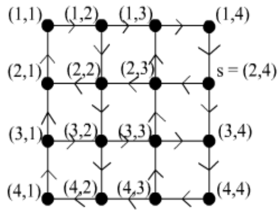
Oriented Mesh:The oriented square mesh $\overrightarrow{M_{m\times n}}$with n is even as follows

- An edge e is an outgoing edge from (i, j) to (i, j+1) when i is odd and$j$ = 1, 2, 3…n-1. and$e$ is an incoming edge from*(i, j+1)* to (i, j) when i is even and j = 1,2,3…n-1
- An edge e is an incoming edge from *(i+1, j)* to (i, j), when i= 1,2,3…n-1 and j is odd and e is an outgoing edge from (i, j) to (i+1, j) when i= 1,2,3…n-1 and j is even .



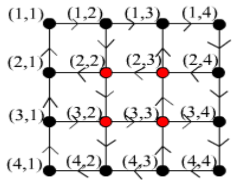
$\overrightarrow{M_{m\times n}}$

*In-degree and Out-degree of a vertex:*Let$v$ be a vertex of a graph. *Outdegree*$\delta(v)$ is the number of edges leaving$v$ *and indegree*$\rho(v)$*is* the number of edges entering$v$.



$$\delta(s) = 2 \; and \; \rho(s) = 1$$

Interior vertices:In a mesh if$\rho(v)$=$\delta(v)$ = 2 then the vertices are called *interior vertices.*



Interior vertices

Corner vertices: In a mesh if $\rho(v)$ = $\delta(v)$= 1 then the vertices are called corner vertices.

Theorem 1: The oriented mesh$\overrightarrow{M_{n\times n}}$ is strongly connected when n is even.

Proof:Let $u = (i,j)$ and$v = (i',j')$. To prove that $\overrightarrow{M_{n\times n}}$is strongly connected we have to prove that there is a path between $u$and$v$, and also between$v$and $u$for every vertex in G. Let us prove this in sixteen cases. Without loss of generality we will assume that $i < i'$ and $j < j'$

$$\boldsymbol{M_{m\times n}}$$

Case 1: Let us assume i and j are odd numbers, i' and $j'$are also odd numbers.

$$u = (i,j) \to (i,j+1) \to (i+1,j+1) \to (i+2,j+1) \to \cdots (i',j+1) \to (i',j+2) \to \cdots \to (i',j') = v$$

$$v = (i',j') \to (i'-1,j') \to (i'-1,j'-1) \to (i'-1,j'-2) \to \cdots \to (i'-1,j) \to (i'-2,j) \to \cdots \to (i,j) = u$$

Case 2: Let us assume i,jand i' are odd numbers, and j' is an even number

$$u = (i,j) \to (i,j+1) \to (i,j+2) \to \cdots (i,j') \to (i+1,j') \to \cdots \to (i',j') = v$$

$$v = (i',j') \to (i'+1,j') \to (i'+1,j'-1) \to (i'+1,j'-2) \to \cdots \to (i'+1,j) \to (i',j) \to (i'-1,j) \to (i'-2,j) \to \cdots \to (i,j)$$

Case 3: Let us assume i ,jand j' are odd numbers, i' is an even number .

$$u = (i,j) \to (i,j+1) \to (i,j+2) \to \cdots \to (i,j') \to (i,j'+1) \to (i+1,j'+1) \to (i+2,j'+1) \to \cdots \to (i',j'+1) \to (i',j') = v$$

$$v = (i',j') \to (i',j'-1) \to (i',j'-2) \to \cdots \to (i',j) \to (i'-1,j) \to (i-2,j) \to \cdots \to (i,j) = u$$

Case 4:Let us assume i, andjare odd numbers, i' andj' are even number.

$$u = (i,j) \to (i,j+1) \to (i,j+2) \to \cdots \to (i,j') \to (i+1,j') \to (i+2,j') \ldots \to (i',j') = v$$

$$v = (i',j') \to (i',j'-1) \to (i',j'-2) \to (i',j) \to (i'-1,j) \to (i'-2,j) \ldots \to (i,j) = u$$

Case 5: Let us assume i,i' and j' are odd numbers, j is an even number.

$$u = (i,j) \to (i+1,j) \to (i+2,i) \to \cdots \to (i',j) \to (i',j+1) \to (i',j+2) \to \cdots \to (i',j') = v$$

$$v = (i',j') \to (i'-1,j') \to (i'-1,j'-1) \to (i'-1,j'-2) \to (i'-1,j) \to (i'-1,j-1) \to (i'-2,j-1) \to (i'-3,j-1) \to \cdots \to (i,j-1) \to (i,j) = u.$$

Case 6: Let us assume i and i' are odd numbers, j and j' are even numbers

$$u = (i,j) \to (i,j+1) \to (i,j+2) \to (i,j') \to (i+1,j') \to (i+2,j') \to \cdots \to (i',j') = v.$$

$$v = (i',j') \to (i'+1,j') \to (i'+1,j'-1) \to (i'+1,j'-2) \to \cdots \to (i'+1,j) \to (i'+1,j-1) \to (i',j-1) \to (i'-1,j-1) \to (i'-2,j-1) \to \cdots \to (i,j-1) \to (i,j) = u.$$

International Journal of Computing Algorithm
Volume: 03, Issue: 01 June 2014, Pages: 1-4
ISSN: 2278-2397

Case 7: Let us assume i, and j' are odd numbers, i' and j are even numbers.

$$u=(i,j)\to(i,j+1)\to(i,j+2)\to\cdots\to(i,j')\to(i,j'+1)\to(i+1,j'+1)\to(i+2,j'+1)\to\cdots\to(i',j'+1)\to(i',j')=v.$$

$$v=(i',j')\to(i',j'-1)\to(i',j'-2)\to\cdots\to(i',j)\to(i',j-1)\to(i'-1,j-1)\to\to(i'-2,j-1)\to(i,j-1)\to(i,j)=u.$$

Case 8: Let us assume i is an odd number, i' ,j and j' are even numbers.

$$u=(i,j)\to(i,j+1)\to(i,j+2)\to\cdots\to(i,j')\to(i+1,j')\to(i+2,j')\to\cdots\to(i',j')=v.$$

$$v=(i',j')\to(i',j'-1)\to(i',j'-2)\to\cdots\to(i',j)\to(i',j-1)\to(i'-1,j-1)\to\to(i'-2,j-1)\to(i,j-1)\to(i,j)=u.$$

Case 9: Let us assume i is an even numberi',j and j' are odd numbers.

$$u=(i,j)\to(i-1,j)\to(i-1,j+1)\to(i,j+1)\to(i+1,j+1)\to(i+2,j+1)\to\to(i',j+1)\to(i',j+2)\to\cdots\to(i',j')=v.$$

$$v=(i',j')\to(i'-1,j')\to(i'-2,j')\to\cdots\to(\mathrm{i},j')\to(\mathrm{i},j'-1)\to(\mathrm{i},j'-2)\to\cdots\to(i,j)=u.$$

Case 10: Let us assume i, and j' are even numbers, i' and j are odd numbers

$$u=(i,j)\to(i-1,j)\to(i-1,j+1)\to(i-1,j+2)\to\cdots\to(i-1,j')\to(i,j')\to(i+1,j')\to(i+2,j')\to\cdots\to(i',j')=v.$$

$$v=(i',j')\to(i'+1,j')\to(i'+1,j'-1)\to(i'+1,j'-2)\to\cdots\to(i'+1,j)\to(i',j)\to(i'-1,j)\to(i'-2,j)\to\cdots\to(i,j)=u.$$

Case 11: Let us assume i, and i' are even numbers, j and j' are odd numbers

$$u=(i,j)\to(i-1,j)\to(i-1,j+1)\to(i-1,j+1)\to(i-1,j+2)\to\cdots\to(i-1,j')\to(i-1,j'+1)\to(i,j'+1)\to(i+1,j'+1)\to\cdots\to(i',j'+1)\to(i',j')=v.$$

$$v=(i',j')\to(i',j'-1)\to(i',j'-2)\to\cdots\to(i',j)\to(i'-1,j)\to(i'-2,j)\to\cdots\to(i,j)=u.$$

Case 12: Let us assume i, i' and j' are even numbers, j is an odd number.

$$u=(i,j)\to(i-1,j)\to(i-1,j+1)\to(i-1,j+2)\to\cdots\to(i-1,j')\to(i,j')\to(i+1,j')\to(i+2,j')\to\cdots\to(i',j')=v$$
$$v=(i',j')\to(i',j'-1)\to(i',j'-2)\to\cdots\to(i',j)\to(i'-1,j)\to(i'-2,j)\to\cdots\to(i,j)=u.$$

Case 13: Let us assume i and j are even numbers,i' and j' are odd numbers.

$$u=(i,j)\to(i+1,j)\to(i+2,i)\to\cdots\to(i',j)\to(i',j+1)\to(i',j+2)\to\cdots\to(i',j')=v.$$
$$v=(i',j')\to(i'-1,j')\to(i'-2,j')\to\cdots\to(i,j')\to(i,j'-1)\to(i,j'-2)\to\cdots\to(i,j)=u.$$

Case 14: Let us assume i,j and j' are even numbers,$i'$is an odd number.

$$u=(i,j)\to(i+1,j)\to(i+2,i)\to\cdots\to(i',j)\to(i',j+1)\to(i',j+2)\to\cdots\to(i',j')=v.$$

$$v=(i',j')\to(i'+1,j')\to(i'+1,j'-1)\to(i'+1,j'-2)\to\cdots\to(i'+1,j)\to(i',j)\to(i'-1,j-1)\to(i'-2,j-1)\to\cdots\to(i,j-1)\to(i,j)=u.$$

Case 15: Let us assume i ,j and i' are even numbers,j' is an odd number.

$$u=(i,j)\to(i+1,j-1)\to(i+2,j-1)\to(i+3,j-1)\to\cdots\to(i'-1,j-1)\to(i'-1,j)\to(i'-1,j+1)\to\cdots\to(i'-1,j')\to(i'-1,j'+1)\to(i',j'+1)\to(i',j')=v.$$

$$v=(i',j')\to(i'-1,j')\to(i'-2,j')\to\cdots\to(i,j')\to(i,j'-1)\to(i,j'-2)\to\cdots\to(i,j)=u$$

Case 16: Let us assume i ,j , $i'$and j' are even numbers.

$$u=(i,j)\to(i+1,j)\to(i+2,j)\to\cdots\to(i'-1,j)\to(i'-1,j+1)\to\cdots\to(i'-1,j')\to(i',j')=v.$$

$$v=(i',j')\to(i',j'-1)\to(i'-1,j'-1)\to(i'-2,j'-1)\to\cdots\to(i,j'-1)\to(i,j'-2)\to\cdots\to(i,j)=u.$$

It is easy to prove the following theorems. Theorem2: If $\rho(t_2)$is one and t_1 is immediate predecessor of t_2, then2-DPP is not solvable for the pair of vertices(s_1,t_1) and (s_2,t_2).

Proof: Since the in-degree of t_2 is one and$t_1$is the immediate predecessor, so it lies on the path joining s_1 and t_1 .Hence 2 DPP is not solvable.

Theorem3: If$\delta(s_1)$ is one and s_2 is immediate successorof s_1, then 2-DPP is not solvable for the pair of vertices(s_1,t_1) and(s_2,t_2).

Theorem4: Let(s_1,t_1) and(s_2,t_2).be any pair of vertices of both in - degreeand out - degree be one. If s_1 and t_1 are diagonally opposite vertices and s_2 andt_2 are the other diagonally opposite vertices, then 2-DPP is not solvable for$(s_1,t_1)$and (s_2,t_2)

Proof:Let P_1 be a path between s_1 and t_1. Since s_1 and t_1 are the corner vertices, the path will cross all the rows and columns. Delete all the vertices and edges in P_1 from G. Then the underlying graph of the$G$-P_1 will be a disconnected graph .Then$s_2$will be in one component and $t_2$will be in another component .We cannot draw a path P_2that connects $s_2$andt_2. Hence 2DPP is not solvable.

International Journal of Computing Algorithm
Volume: 03, Issue: 01 June 2014, Pages: 1-4
ISSN: 2278-2397

REFERENCES

[1] Even S, Itai A and.Shamir A, On the Complexity of timetable and multicommodity flow problems, SIAM Journal of computation, 5, (1976), 691-703.

[2] Fortune S, Hopcroft J.E. and Wyllie J. , The directed subgraph homeomorphism problem, Theoretical Computer Science,10, (1980), 111- 121

[3] Kobayashi Y. and. Sommer C, On shortest disjoint paths in planar graphs,Discrete Optimization, 7, 2010, 243-245.

[4] Khuller S. Mitchell S. G and Vazirani V. V., Processor efficient parallelalgorithms for the two disjoint paths problem and for finding a Kuratowskihomeomorph, SIAM Journal of Computation, 21, (1992), 486- 506.

[5] Lynch J. F, The equivalence of theorem proving and the interconnectionproblem, (ACM) SIGDA Networks, 5 (1975), 31-36.

[6] Ohtsuki T, The two disjoint path problem and wire routing design, Proc.Symposium on Graph Theory and Algorithms, Lecture Notes in Computer Science, 108, Springer, Berlin, (1981), 207-216.

[7] Perl Y and. Shiloach Y, Finding two disjoint paths between two pairs ofvertices in a graph, Journal of Association for Computing Machinery, 25,(1978), 1-9.

[8] Ripphausen-Lipa, H. Wagner D andWeihe K, Combinatorial optimization papers from the DIMACS Special Y ear, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 20, (1995), 295-354.

[9] Robertson N and Seymour P. D., Graph minors: XIII: The disjoint paths problem, Journal of Combinatorial Theory, Ser. B, 63, (1995), 65-110.

[10] Schrijver A., Finding k disjoint paths in a directed planar graph, SIAM Journal of Computation, 23, (1994), 780-788.

[11] Woeginger G, A simple solution to the two paths problem in planar graphs, Information Processing Letters, 36, (1990),191-192.