Canine Algorithm for Node Disjoint Paths

R. Ananthalakshmi Ammal¹(✉), P. C. Sajimon¹, and Vinod Chandra S S²

¹ Cyber Security Group, Centre for Development of Advanced Computing,
Thiruvananthapuram, India
{lakshmi, pcsaji}@cdac.in

² Computer Centre, University of Kerala, Thiruvananthapuram, India
vinod@keralauniversity.ac.in

Abstract. Node Disjoint Paths (NDP) is one of the extensively studied Graph Theory problem. In this problem, the input is a directed n vertex graph and the set of source destination pair of vertices. The goal is to find the maximum number of paths connecting each such pair, so that such discovered paths are node-disjoint. In this paper, a novel Canine Inspired Algorithm is proposed which is a bio-inspired one, based on the olfactory capabilities of canines in tracing and reaching a destination. Currently many of the existing algorithms try to identify disjoint paths in a linear manner, whereas the Canine algorithm can be executed in a concurrent manner, depending on the number of canines deployed to find the disjoint paths. The time complexity of the algorithm is estimated to be $O(n \log n)$. We hope that this algorithm finds many applications in problems related to various fields such as communication networks, scheduling and transportation and provides better results.

Keywords: Node Disjoint Paths · Canine Algorithm · Bio-inspired

1 Introduction

Disjoint path is an extensively studied fundamental routing problem of Graph Theory. Disjoint paths can be Edge Disjoint Paths (EDP) or Node Disjoint Paths (NDP) for directed or undirected graphs. EDP share no common edges or links and NDP share no common nodes or vertices. Finding NDP is more limited than EDP and if two paths are node disjoint, they are edge disjoint as well. Disjoint path finds many applications in various fields such as communication networks, design of Very Large Scale Integrated (VLSI) circuits, transportation problems and scheduling problems. Many algorithms have been published and the problem is seen as an extension of the shortest path problem.

The disjoint path problem has been shown to be NP-Complete by Karp [1], Lynch [2] and others for many variants of planar graphs and grids. We studied the problem of NDP in graphs and we propose a simple bio-inspired algorithm based on the behavior of canines to solve the problem. As part of the solution multiple paths between a source and destination are computed that do not share nodes or links.

2 Preliminaries

Let $G = (V, E)$ be a simple directed graph where V is the set of nodes and E is the set of edges. Assume that the graph G has n nodes and m edges. The source-destination vertices are denoted as $\{v_i, v_j\}$ and $V(P)$ denotes the set of nodes and $E(P)$ denotes the set of edges on the path P respectively.

For the same source destination pair $\{v_1, v_j\}$, let $P_1, P_2, P_3, \dots, P_k$ be the set of k node disjoint paths such that $E(P_i) \cap E(P_j) = \emptyset$ and $V(P_i) \cap V(P_j) = \{v_1, v_j\}$ for different $i, j \in \{1, 2, 3 \dots k\}$. An example of NDP topology is shown in Fig. 1.

Fig. 1. k Node disjoint source destination paths

In a directed graph, each node can be considered to be split up into two nodes v_{in} and v_{out} and the undirected edges (s, t) can be replaced with directed edges (s_{out}, t_{in}) and (s_{in}, t_{out}) . All then incoming edges of v will be connected to node v_{in} and all the outgoing edges will be connected to node v_{out} .

The degree of a node $d_G(v)$ in graph G is the number of edges incident on v , each loop counting as two edges. This can be considered as the sum of indegree and outdegree of the node in a directed graph.

The goal of this paper is to find for every node $v_1 \neq v_n$, a set P_1 of disjoint paths from source node v_1 to destination node v_n in the graph G in which the paths of the set P_1 are node disjoint.

3 Related Works

One of the extensively studied problem in Graph Theory is the Shortest Path problem, which is the problem of finding the path between two nodes in a graph so that the sum of the link cost is minimal. The most widely used shortest path algorithms include Dijkstra's algorithm [3] and Bellman – Ford algorithm [4, 5]. Bio inspired algorithms for optimal path computations are also in use for quite some time. The famous AntNet algorithm

based on Ant Colony Optimisation [6] is used in many telecommunication Networks for routing purposes. Smell Detection Agent (SDA) [7] based algorithm is a novel bio-inspired optimisation algorithm. The authors of this paper have also published a SDA based algorithm [8] for the shortest path problem. The disjoint path problem can be seen as an extension of the shortest path problem.

One of the earlier published algorithms for finding disjoint paths in a network was from Suurballe [9] and subsequently from Bhandari [10]. Many variants of disjoint paths algorithms such as availability based disjoint [11], maximally disjoint [12] and region based disjoint paths [13] are published. Several other constraints including the weights of the constituent edges of the paths are also added such as min-sum and min-max.

The proposed Canine algorithm is an extension of the Smell Detection Agent based algorithm published by the same authors. This agent-based algorithm can be used for finding all NDP from one node to all other nodes and is a simple and easy to implement.

4 Canine Algorithm

Canine algorithm is based on the trained behavior of the canine family in detecting smell trails. The method for identifying disjoint paths from the source node to the destination node is based on the bio-inspired analogy of trained behaviour of dogs in detecting smell trails. The olfactory mechanism of dogs can detect as well as memorize different smell signatures. Dogs also urinate in different spots to mark their territory as occupied. These two properties of dogs have been used to mark the path they undertake to reach the destination. Multiple dogs can be used for the identification of disjoint paths from a source node to the destination node using the property that one dog's territory shall not be traversed by the other dog.

Each node in the graph is considered as a smell spot, which is visited by the canine. Each smell spot is characterized by the signature related to the canine which visited the spot and the other the smell trail which is a function of the outdegree of the node. Each canine is also characterized by its unique signature value to mark the smell spots and its olfactory capability. The olfactory capability of the canine can be earmarked as the smell radius for the canine to traverse and is assumed to be proportional to the domain of the given graph which is the set of nodes and edges. The whole algorithm is based on the given graph with the set of nodes and edges. The number of canines deployed for the algorithm is equal to the maximum outdegree of the source node in the given graph. The canines deployed from each node determine the disjoint paths by traversing the unvisited highest smell trail smell spots until the destination node. This is repeated for all the nodes with their roles changing as source and destination to find all the node disjoint paths among all nodes within a given graph. The symbolic representation is given in Fig. 2.

Canine Algorithm for finding all Node Disjoint Paths in a Graph for all the nodes

Fig. 2. Symbolic representation of Canine deployment from source node

Input: Graph - $G(V,E)$; Number of nodes - n ; Maximum outdegree of node - $\Delta_{Gout}(v)$; Degree of each node - $d_G(v)$

Output: P -Set of all Node Disjoint Paths from each node to every other node in the graph $G(V,E)$

1. Initialise:
 - a. Number of smell spots = n
 - b. Maximum number of canines = $\Delta_{Gout}(v)$
 - c. For each $v_i \in V$ smell value = $d_G(v_i)$
 2. For each $v_i \in V$, do
 - a. Reset all smell spots as not visited
 - b. For each $v_j \in V$, do
 - i. Set the number of canines as $\delta_{Gout}(v_i)$
 - ii. Set signature value of each canine
 - iii. For each canine, do
 - Identify path by canine traversal through unvisited highest value smell spot until v_j
 - Populate the path set P_i for the canine which reached destination node v_j
-

5 Algorithm Analysis

The algorithm is inspired from the smell detection capability of the canine to reach the destination. The objective of the algorithm is to find all the NDP from each node to every other node in the graph. So, the major input to the algorithm is the graph with its set of nodes $G(V)$ and the set of edges $G(E)$ connecting the nodes. Consider the problem of finding the node disjoint paths from node 1 to 8, in graph of 8 nodes and 13 edges as shown in Fig. 3.

With node v_1 having an outdegree $\delta_{Gout}(v_1)$ of 3, three canine agents are to be deployed to traverse the graph and identify the disjoint paths to node v_8 . Initially all

Fig. 3. Illustrative example of an 8 node directed graph

nodes are marked as unvisited. The canine agents traverse from the outbound edges of the node through the neighbouring smell spots which are unvisited and having the highest smell trail until they reach the destination. The smell spot is marked as visited by the unique signature of the canine. The smell trail of a smell spot is proportional to the degree of the node. This process results in discovery of all disjoint paths from the source to the destination. Thus, the identified paths include $P_1 = \{1-3-6-8, 1-2-5-8, 1-4-7-8\}$ as shown in Fig. 4. As another example, the set of disjoint paths from node v_2 to node v_8 can be identified as $P_2 = \{2-3-6-8, 2-5-8\}$. In a directed graph, all the node disjoint paths can be iteratively identified from one node to all the other nodes. The neighboring nodes may be excluded from destination nodes.

Fig. 4. The three identified disjoint paths from node 1 to node 8

It has been found that the number of disjoint paths identified conforms to the Menger's theorem [14] which states that the maximum number of vertex-disjoint (s, t) paths is equal to the minimum size of an (s, t) disconnecting vertex set. In other words, the maximum number of vertex-disjoint paths from node v_1 to node v_8 is equal to the minimal number of vertices to be removed which disconnects node v_1 from node v_8 . In the example taken, the (s, t) cut disconnects the three nodes v_5, v_6 and v_7 resulting in

node v_1 in one partition and the node v_8 in the other partition. Thus, as per the theorem, the number of disjoint paths identified is also equal to three.

In the illustrated example, the path from v_8 to other nodes seems to be non-existent, as there are no directed edges available. In the case of communication networks, duplex communication over the network links exists and, in such cases, the paths can be computed. Present day communication networks are configured in such a way that no loops or cycles exist as part of the network topology.

5.1 Comparison

Many of the algorithms for identifying NDPs use Depth First Search and Breadth First Search techniques to arrive at the results. Bhandari's algorithm [10] use the min-sum disjoint path algorithm and the split node technique for directed graphs. Broder et al. [15] has used the flow techniques by eliminating the vertices. They identified that in a random graph with n nodes, we can connect at most $O(n \log d / \log n)$ pairs. The proposed Canine algorithm is simple, easy to implement and the time complexity is estimated as $O(n \log n)$ for a graph with n nodes.

6 Discussions

The NDP finds application in many real-life problems. One of the applications where this algorithm was tested was in a simulated communication network setup with 18 nodes corresponding to 18 core routers connected with multiple links among them. The knowledge of all existing NDP were essential to configure a resilient network.

Swarm Intelligence algorithms such as Ant Colony Optimization and Bee Colony optimization, are among the commonly used bio-inspired algorithms for the identification of shortest paths and disjoint paths. In such swarm intelligent algorithms, the stigmergic aspects of pheromone in the case of ants and waggle dance in the case of bees are made use of in the optimization problems. Prior to the start of the optimal path identification, many of the Swarm Intelligent algorithms requires an initial random walk or search to be performed. For example, in Ant Colony Optimization, this leaves a trail of the chemical for the other ants to follow. In the Bee Colony Optimization, the scout bees do the initial survey of the food sources. In the Canine Algorithm, the intelligent behavior of canines with their olfactory capability is applied. No prior random search is envisaged in the algorithm. This is specifically useful in cases where the network topology details are either not discovered or the network link states change dynamically.

7 Conclusions

The Canine Algorithm is an algorithm derived from the rather curious and efficient mechanisms used by canines; which could be used to solve many computational problems. The advantage is that multiple canine agents can be used concurrently to obtain a faster and better solution. This provides a better performance of the algorithm compared to the existing algorithms. As part of the future works, identification of NDP with constraints can be done, by suitably modifying the Canine Algorithm to handle the constraints.

References

1. Karp, R.M.: On the computational complexity of combinatorial problems. *Networks* **5**, 45–68 (1975)
2. Lynch, J.F.: The equivalence of theorem proving and the interconnection problem. *SIGDA Newsl.* **5**, 31–36 (1975)
3. Dijkstra, E.W.: A note on two problems in connection with graphs. *Numer. Math.* **1** (1959). <https://doi.org/10.1007/BF01386390>
4. Bellman, R.: On a routing problem. *Q. Appl. Math.* **16**, 87–90 (1958). <https://doi.org/10.1090/qam/102435>
5. Ford, L., Fulkerson, D.R.: *Flows in Networks*. Princeton University Press, Princeton (1962)
6. Ducatelle, F., Di Caro, G.A., Gambardella, L.M.: Principles and applications of swarm intelligence for adaptive routing in telecommunications networks. *Swarm Intell.* **4** (2010). <https://doi.org/10.1007/s11721-010-0040-x>
7. VinodChandra, S.S.: Smell detection agent based optimization algorithm. *J. Inst. Eng. India Ser. B.* **97**, 431–436 (2016)
8. Ananthalakshmi Ammal, R., Sajimon, P.C., Vinodchandra, S.S.: Application of smell detection agent based algorithm for optimal path identification by sdn controllers. In: Tan, Y., Takagi, H., Shi, Y., Niu, B. (eds.) *ICSI 2017. LNCS*, vol. 10386, pp. 502–510. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61833-3_53
9. Suurballe, J.W.: Disjoint paths in a network. *Networks* **4** (1974). <https://doi.org/10.1002/net.3230040204>
10. Bhandari, R.: Optimal physical diversity algorithms and survivable networks. In: *IEEE Symposium on Computers and Communications – Proceedings*, pp. 433–441. IEEE (1997). <https://doi.org/10.1109/iscc.1997.616037>
11. Yang, S., Trajanovski, S., Kuipers, F.A.: Availability based path selection. In: *Proceedings of International Workshop on Reliable Networks Design and Modeling (RNDM)* (2014)
12. Omran, M.T., Sack, J.-R., Zarrabi-Zadeh, H.: Finding paths with minimum shared edges. *J. Comb. Optim.* **26**(4), 709–722 (2012). <https://doi.org/10.1007/s10878-012-9462-2>
13. Trajanovski, S., Kuipers, F.A., Ilic, A., Crowcroft, J., Van Mieghem, P.: Finding critical regions and region-disjoint paths in a network. *IEEE/ACM Trans. Netw.* **23**, 908–921 (2015). <https://doi.org/10.1109/TNET.2014.2309253>
14. Menger, K.: Zur allgemeinen Kurventheorie. *Fundam. Math.* **10**, 96–115 (1927)
15. Broder, A.Z., Frieze, A.M., Suen, S., Upfal, E.: An efficient algorithm for the vertex-disjoint paths problem in random graphs. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 261–268 (1996)