ICICS-PCM 2003
15-18 December 2003
Singapore

Note on the Problem of Partially Link Disjoint Paths

Alexander A. Kist and Richard J. Harris
RMIT University Melbourne
BOX 2476V, Victoria 3001, Australia
kist@ieee.org, richard@catt.rmit.edu.au

Abstract

This paper discusses the problem of partially link disjoint paths in communication networks. The problem is situated in between the well-known shortest path generation and the synthesis of two disjoint paths. The contributions of this paper are twofold: Firstly, topologies and relevant graph properties are investigated and the degree of divergence is defined as a metric to distinguish the degree of commonality for the two partially disjoint paths. Secondly, heuristic approaches are introduced that solve the problem of generating partly disjoint paths.

1. Introduction

The creation of backup paths in networks is an important network design problem. These paths are needed to restore connections in the case of link failure and it is a convenient way to improve the reliability and quality of a connection. The ideal backup path for link failure would have no links in common with the original path for a connection. The second objective is to find the path with minimum cost, where the cost is a general network optimisation parameter. In certain topologies it is not possible to find two completely disjoint paths due to the network structure. In such a case it would be helpful to find the best partially disjoint path to decide if it is still a reasonable backup path. The second motivation to find partially disjoint paths is, that it is good enough to find two paths that satisfy a degree of divergence and still meet network design objectives.

Robust backup path generation is a problem that occurs in relation to all virtual path connections in communication networks. Possible applications include ATM networks, light path design and MPLS based traffic engineering. Partially disjoint path algorithms can also be used as intelligent alternatives to k-th shortest path algorithms. Where backup path generation and load balancing have been extensively investigated in the past (e.g. [1] and [2]), the problem of partial link disjointness has not been widely considered. Iwata [3] discusses a special case for partially link disjointness where shared paths are already known. Brander gives in [4] a comprehensive study between several k-th shortest path algorithms without the disjointness condition.

The problem discussed in this paper is specified as follows: Synthesise a pair of partially link-disjoint paths between an origin node and a destination node (OD-pair) in a given generic network such that the global cost of this synthesis is minimal. The problem is further specified with the assumptions that the underlying network has only links that allow a traffic flow in one direction (unidirectional links) and that the underlying network has at most one link between a node pair. The nodes are singly connected. Choudhury [5] gives references which allow the avoidance of these assumptions. The link cost is an additive positive cost value assigned to all network links. In this context the shortest path is defined as the path with the minimum global cost (cost sum of all links that are part of the path) as opposed to the definition that the shortest path is the one with the least number of hops.

The situation for *Shortest Path* (SP) and *Disjoint Paths* (DPs) will be discussed in the next section. Both problems are well known and a number of heuristics exist to address both challenges. Examples include the Dijkstra shortest path algorithm [6] and Suurballes two disjoint shortest paths algorithm [7]. The section also outlines helpful conclusions which are useful for the partially disjoint paths algorithm design. Generally, analysis in this paper follow rather intuitive approaches than strict mathematical proofs. Section 3 discusses the basic situation for metrics for disjointness. It defines values to judge the level of disjointness of two paths in the graph. Several different approaches are possible and different *Degrees of Divergence* (DoD) are defined and discussed in this section. Section 4 introduces two algorithmic approaches and the paper concludes with a summary.

2. Graph Properties

Disjointness and minimized global cost (cost sum of both paths) are the main objectives for backup paths. In a link disjoint case two paths have no links in common. A weaker condition is the partial disjointness where parts of connections are link disjoint and parts of the connections use the same hops. Two cases are of interest. In the first case the global cost of the partially link disjoint paths is better than the global cost of the two link disjoint paths and in the second case, only partially link disjoint paths exist. This section discusses topologies and properties for both, the link disjoint and the partly link disjoint

0-7803-8185-8/03/$17.00 © 2003 IEEE

case. Note that discussions in this section ignore additional paths with greater cost.

The simplest approach to generate link disjoint paths is to build a subgraph of the original network without the shortest path arcs, then find a second path in this subnetwork. However, there is a special called the *trap topology* where this approach does not work. It can be shown that for the disjoint paths a graph can be divided into two possible topologies the *trap topology* and the *shortest path topology.* A graph comprises a SP topology for a given OD-pair if the shortest path is one path of the two completely disjoint paths. In a graph with a trap topology for a given OD-pair one DP uses the root part of the SP and the other DP uses the spur part of the SP and one of the shortest path hops is neither used by the first disjoint path nor by the second disjoint path. Figure 1 depicts an example of a trap topology. The OD-pair is *a-d.* The shortest path in this example is $a-b-c-d$ (cost 3). The two best partially disjoint paths are $a-c-d$ (cost 4) and $a-b-d$ (cost 3). The first DP uses the spur part (in Figure 1 arc *cd*) of the SP and the other DP uses the root part of the SP (in Figure 1 arc *ab*) and one of the shortest path hops is neither used by the first disjoint path nor by the second disjoint path (in Figure 1 arc *bc*).

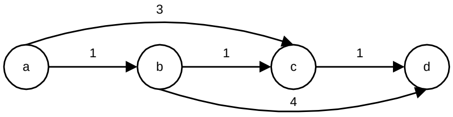

Figure 1: Trap Topology Example

The previous statements were concerned with the complete disjointness of two paths; now the case of partial disjointness is considered. All discussions use the assumption that the cost of the partially disjoint path has to be better than the global cost of disjoint paths for a reasonable solution. Otherwise there is no need to weaken the complete disjointness. As above, the situation is discussed for the shortest path and trap topology. Equation (1) states the global cost situation for a given OD-pair.

$$2 \cdot C_{SP} \leq C_{PD} \leq C_{DP} \tag{1}$$

By definition the smallest cost is given by the length of the shortest path. It is the best possible result and therefore the lower bound. The best cost for two disjoint paths is C_{DP} and therefore is the upper bound. The cost of the shortest partially disjoint path pair has to be somewhere in between these two bounds for a reasonable result.

It can be shown that in an SP topology the SP is one part of the solution. Furthermore it is not possible to reduce the cost in a trap topology by sharing hops between the two disjoint paths, except the parts that are similar to SP hops and therefore an SP topology. Figure 2 depicts an example of a trap topology with the OD-pair $a-d$ and the SP *abcd.* The two disjoint paths are $aefcd$ and *abghd.* The link that could be shared between the two paths is *gh* in this example. The partially disjoint paths are $aeghfcd$ and *abghd.* For simplicity no paths of equal cost exist in this example, but discussions can be easily adapted for this case. To improve the global cost by path sharing, the path $eghf$ has to have a smaller cost than the link ef. The second condition uses the SP: since *bc* is part of the SP the path $bghfc$ has to have a higher cost than *bc.* Combining the two conditions shows that the added cost of the paths *bc* and *eg* have to be smaller than the added cost of the paths ef, *bg* and fc (Equation (2)).

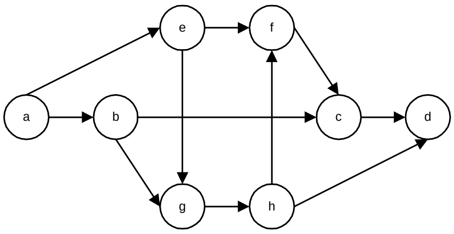

Figure 2: Trap Topology Condition

$$C_{bc} + C_{eg} < C_{ef} + C_{bg} + C_{fc} \tag{2}$$

The added cost of the two DPs $aefcd$ and *abghd* has to be smaller than the added cost of the alternative paths *abcd* and *aeghd* that this example comprises a trap topology. This yields the third condition: the cost sum of the links *bc* + *eg* has to be higher than the cost sum of the paths efc and *bg* (Equation (3)).

$$C_{bc} + C_{eg} > C_{ef} + C_{fc} + C_{bg} \tag{3}$$

Obviously Equation (2) and Equation (3), exclude each other, therefore the global cost cannot be improved by sharing non SP paths in a trap topology.

Simiarly, it can also be shown that if a trap topology is included in the best solution for two disjoint paths, it is not possible to get a better result by sharing the SP part that is not part of any of the two disjoint paths (in Figure 1 the hop *bc*). If a solution exists for the SP link that is not used by both paths it provides no improvement as long as the condition for a trap topology is meet.

3. Metrics for Disjointness

One major difficulty in solving the two partially disjoint path problem is the definition of a metric that measures the independence of the two paths. In the partially link disjoint case, common hops are used by different paths. To compare the degree of independence of the two partially disjoint paths, it is necessary to define a metric that reflects this. Two metrics for partial disjointness are proposed. The first uses a hop count as the parameter and measures the proportion of the links that are different, the second uses the cost associated with the links.

Degree of Divergence (Hop Count) A first approach involves the number of hops used by each of the two partially disjoint paths. For example, the path *abd* (Figure 1) has two hops while the path *abcd* has three hops. Hop *ab* is common to both paths. They are partially link-disjoint. The degree of disjointness can be defined by the number of common hops divided by the number of hops of the shorter of the two paths. For this example, it is 1/2. If h_A and h_B are the number of hops on two partially disjoint paths A and B, and h_{com} is the number of hops common to both then the degree of divergence (hop count) d_h is defined as Equation (4)

$$d_h^{min} = 1 - \frac{h_{com}}{min[h_A, h_B]} \tag{4}$$

This metric is only sensitive to the shorter of the two partially disjoint paths. An alternative definition is shown in Equation (5). This is sensitive to the length of both paths.

$$d_h = 1 - \frac{2 \cdot h_{com}}{h_A + h_B} \tag{5}$$

For this DoD definition the example yields $d_h = \frac{3}{5}$. The drawback in this definition is the fact that making one of the paths longer and longer using links that are not part of the other path decreases the divergence. This seems to encourage resource-wastage, but the second major objective of the path generation is to minimize the global cost. This encourages the use of as few links as possible.

Degree of Divergence (Cost) This section defines a cost based disjointness metric and its bound. As in the previous section two definitions are possible. Two paths A, B in a graph are called partially disjoint, if they have one or more links in common. The part that is used by both partially disjoint paths is called the *common path* with a cost of C_{com}. The Degree of Divergence (Cost) d_c is defined by Equation (6)

$$d_c^{min} = 1 - \frac{C_{com}}{min[C_A, C_B]} \tag{6}$$

where C_A is the cost of path A, C_B is the cost of path B and C_{com} is the cost of the links that both paths have in common. The degree of divergence has two extreme cases: If the two links A, B are completely disjoint $d_c = 100\%$. In this case $C_{com} = 0$ and the fraction is equal to zero. For two paths that use the same hops $2 \cdot C_{com} = C_A + C_B$ and the fraction equals one. The degree of divergence in this case is $d = 0\%$. For the example in Figure 1 with the O node a and the D node d the shortest path is *abcd* with a total cost of $C_{SP} = 3$. The partially disjoint path is *acd* with a cost of 4. The degree of divergence is $d_c^{min} = 1 - 1/3 = 67\%$. As before, it is also possible to define a Degree of Divergence (Equation (7)) that is sensitive to both paths.

$$d_c = 1 - \frac{2 \cdot C_{com}}{C_A + C_B} \tag{7}$$

The extreme cases are $d_p = 100\%$ and $d_p = 0\%$. For the example in Figure 1 Equation (7) yields $d_c = 1 - 2/7 = 71\%$. The next definition with reference to the SP is given in Equation (8). For a given OD-pair with two partially disjoint paths the degree of divergence with reference to the shortest path is given by the *Pseudo Degree of Divergence*

$$d_p = 1 - \frac{C_{com}}{C_{SP}} \tag{8}$$

where C_{SP} is the cost of the shortest path and C_{com} is the cost of the common part of the path. The SP in this definition is the SP for the given OD-pair, not the shorter path of the two partially disjoint paths. The bounds for this definition are the same as before. For two paths sharing all hops $d_p = 0\%$ and for two complete disjoint paths $d_p = 100\%$. The pseudo degree of divergence for the example in Figure 1 is $d_p = 1 - 1/3 = 66\%$. This result is different from the result of the first definition. If a default bound is specified for the pseudo degree of divergence this places an upper bound on the allowed cost of the common part of the path, according to Equation (8).

Discussion This section discusses the advantages and disadvantages of the different metrics: Should the metrics be based on the hop count or on the cost assigned to the links? Two lines of argumentation are possible. The first one is solely based on reliability. It can be argued that without knowing anything about probabilities of individual links failing it has to be assumed that all links have an equal probability of failure and therefore a good backup path will have the greatest proportion of disjoint links irrespective of the cost of any of the links. An additional consideration is that most paths in telecommunication networks have a limited number of hops (e.g. because of the delay). This means that several different paths with different costs would have the same degree of divergence and the choice between them would be arbitrary.

The second argument is based on the generic network optimisation link metrics. The shortest path is defined using these metrics and the disjointness is also defined in relation to them. In this context the disjointness is not necessarily concerned with the reliability of the connections but in sharing links with a small cost value. The inverse capacity ($1/c_{ij}$) is an example of a commonly used metric. Here a link with a high capacity is a short link and therefore a short common path would indicate a high capacity link.

The second question that has to be addressed is if the definitions with the minimum in the denominator (Equations (4) & (6)) are more reasonable than the definitions using the sum of both path parameters (Equations (5) & (7)). Whereas the first is sensitive to the length of the shorter of the two paths, the latter is sensitive to the length of both paths. As mentioned before, the “sum” encourages longer backup paths to minimise the DoD, but this is discouraged by the minimum global cost objective.

The degree of divergence in Equation (7) has the disadvantage that for the case where one of the two paths is very long compared it to the other path it would indicate a reasonable degree of divergence even when most of the shorter path is also used by the longer path. If it is assumed that the DoD is only used to measure the commonality of two paths and the global cost is part of the global cost objective, the definition in Equation (7) can provide suitable results. One problem of all DoD definitions is that the length of the paths is not known before the execution of a path generation algorithm. A trap topology could be part of the solution and therefore the degree of divergence on such a structure can only be calculated after both paths are found. Following this argument a successive approximation would be necessary. An algorithmic approach that is independent of the degree of divergence would be the best solution. This way the appropriate degree definition can be chosen on the actual requirements.

The *pseudo degree of divergence* can be calculated before the partially disjoint path generating algorithm is executed, but it sometimes provides an invalid result, because the whole SP is not part of the solution at all times. It provides only an upper bound. If the algorithm starts from a pseudo degree of divergence, a found solution is within the degree of divergence. If the SP is one of the two solution paths, the pseudo degree of divergence provides an optimal measurement. In a trap topology, where the SP path is not part of the solution, the shorter of the two solution paths is longer than the SP path and the pseudo degree of divergence gives only an upper bound. The algorithmic approaches have to take these facts into account. This paper provides a general overview of the problem. If the presented algorithm is used for a practical network, it is likely that the network design parameters provide more detailed objectives, e.g. the shareable paths are restricted or known before hand. In this case the algorithm can be modified to find an optimal solution for such special cases.

4. Different Approaches

This section is concerned with different algorithmic approaches to solve the problem of synthesis of two partially disjoint paths in a graph. Based on the discussion of the metric for the disjointness in Section 3, two general types of approaches appear to be likely. The first approach calculates partially disjoint path pairs in an order, where the global cost is constantly increasing from path pair to path pair, and the cost of common paths has to be constantly decreasing from calculation to calculation. This way it is possible to calculate the degree of divergence for every pair found. If the degree is not satisfied, the next path pair is generated. An approach of this kind is the k-th SP approach described in the next section. The second type of approach should be able to generate "all best" partially disjoint paths. These are partially disjoint paths that have a decreasing degree of divergence for smaller global cost paths. After the execution of the algorithm, the path pair satisfying the requested degree of divergence is chosen as a solution.

K-th SP Approach This approach uses k-th SP algorithms, which generate a set of k SPs between a given OD-pair. k-th SP algorithms are not concerned with the disjointness condition. A solution can be found by generating a number of shortest paths and comparison of their disjointness. The two parameters of interest are the disjointness and the global cost. An iterative process is necessary: Calculate the SP and the second SP, compare them and if they do not meet the termination condition, calculate the third SP and compare this path with all the others. This process continues until the termination condition is fulfilled. The terminating condition is important for the correctness of the solution. After a disjoint path pair within the degree of divergence is found during the k-th iteration, two cases are possible, viz.:

(1) The algorithm terminates if one path of the two solution paths is the SP. This is possible because the path generated with the next iteration is longer or equal than all previously calculated paths. There is no better solution possible because the SP is already part of the solution and all combinations with the new found path would have a higher global cost. It is possible that the next path has the same length as the current path. In this case it is possible that this combination has a better degree of divergence than the already found solution, but the global cost will remain the same.

(2) The shortest path is not one of the found path pairs. Here this solution provides only an upper bound. More iterations are necessary to make sure the best solution is found. The bound is that the cost of the k-th shortest path is higher than the global cost of the already found solution minus the SP cost. If, for example, the set of the path costs found at first is $\{1, 3, 4\}$ and the solution path pair is the second and the third path, the global cost of this combination is then 7. If the next path provides a solution with a cost better than $(7 - 1)$, this solution is better. When the algorithm terminates a better solution has to be found or it does not exist.

The algorithm will find the best solution with this stop condition since the best paths are compared iteratively. The approach requests no specific definition for the degree of divergence. Three disadvantages can be identified. The first problem is that k-SP algorithms are slow. The best example presented in [4] is of order $O(k \cdot n^2)$. The more important second problem is that the algorithm has to perform $k \cdot (k - 1)$ comparisons. The algorithm has to compare the new path with all previously found paths. A possible trap topology would be missed if the result is only compared with the shortest path. The third problem is the unpredictable size of k. There exists no obvious polynomial bound for k, which ensures that the disjoint paths are found. The next section discusses an approach based on a disjoint path algorithm.

DP Approach When the SP and the two DPs are known the solution for the partially disjoint path problem is found somewhere in between. This approach is based on a completely disjoint path algorithm. Suurballe [8] describes an efficient implementation of an algorithm for solving the *problem of finding two disjoint paths* in a network with a total minimum cost.

Firstly the algorithm finds the shortest path between an OD-pair. In the next step the network is transformed. A second run of an SP algorithm finds the second disjoint path. The suggested modification requires that an additional label be assigned to every network node. This label is similar to the label used by the Dijkstra algorithm, but it stores not only the path cost but also the cost of the shared path and the path itself. The label holds all best partially disjoint paths. Every time, the second run of the SP algorithm marks a SP node finally, SP nodes are processed in a forward order and partially disjoint paths are added to the label. The observations in Section 2 make this possible since only SP links provide a better global cost.

One problem is that the algorithm could finally mark nodes before all partly disjoint paths are collected and therefore part of the path is missed. This can be handled by a "step back" to the origin node and a restart of the algorithm where the node labels are not reset. The second problem is a gap topology (e.g. Figure 1 link ac has infinite cost). The node behind the gap will never be marked unless a special condition is implemented. The fact that a gap divides the network into two subnetworks can be exploited to implement a restart condition. The influence of the degree of divergence depends on the implementation of the assigned label. In general this approach is capable of calculating all best partially disjoint paths for an OD-pair. In the worst case there are $k \cdot n$ restarts necessary. If all network nodes are SP nodes, k is bounded by $(n-2)$. The expected worst case running time is bounded by $O(n^3)$ using a Dijkstra implementation. An algorithmic solution in [9] is based on this approach.

5. Conclusion

The first part of this paper outlined observations concerning the problem of synthesising partially disjoint paths in graphs. Different network topologies and their properties were discussed. The degree of divergence was introduced and several different definitions were proposed.

The second part discussed two approaches to solving the problem of the synthesis of partially disjoint paths in a generic network. The second approach seems to be the most promising one. Several reasons make this approach interesting. The expected running time is acceptable and the generation of "all best" paths allows the use of any disjointness metric without influence on the algorithmic procedure.

6. Acknowledgments

The helpful discussions with the colleagues at the Centre for Advanced Technology in Telecommunications (CATT), in particular Sayaan Choudhury, are gratefully acknowledged. The authors also would like to thank Andreas Christ (University of Applied Science, Offenburg, Germany) for his support.

References

[1] D. Torrieri, "Algorithms for finding an optimal set of short disjoint paths in a communication network," *IEEE Transactions on Communications*, vol. 40, no. 11, pp. 1698–1702, 1992.

[2] R. G. Ogier, V. Rutenburg, and N. Shacham, "Distributed algorithms for computing shortest pairs of disjoint paths," *IEEE Transaction on Information Theory*, vol. 39, no. 2, pp. 443–455, 1993.

[3] A. Iwata, R. Izmailov, and B. Sengupta, "Alternative routing methods for pnni networks with partially disjoint paths," *IEEE Globecom -New York- 1998*, vol. 1, pp. 621–626, 1998.

[4] A.W. Brander and M. C. Sinclair, "A comparative study of k-shortest path algorithms," *Proc 11th UK Performance Engineering Workshop, Liverpool*, pp. 370–379, 1996.

[5] S. Choudhury, *Approaches for Robust Virtual Path Synthesis in ATM Networks*, Ph.D. thesis, Department of Computer Systems Engineering, RMIT University, Melbourne, July 1999, Thesis submitted for the degree of Doctor of Philosophy.

[6] E.W. Dijkstra, "Note on two problem in connection with graphs," *Numerische Mathematik*, pp. 269–271, 1959.

[7] J.W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125–145, 1974.

[8] J.W. Suurballe and R. E Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325–336, 1984.

[9] A.A. Kist and R.J. Harris, *A Heuristic to Generate All Best Partially Disjoint Paths in a Communication Network*, Eighth International Conference on Communication Systems (ICCS 2002), Singapore, 25-28 November 2002.