# Optimal Design of a 2-connected Candidate Set for a Communication Network

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 1, Issue No. 2, 75–86
©The Author(s) 2012
https://journals.mirlabs.in/index.php/ijcsnt
DOI: 10.18486/ijcsnt/1.2.007
IJCSNT

## A. Q. Ansari[1], Koyel Datta Gupta[2], Maneesha Dabas[3]

### Abstract

With the rise in the use of wired and wireless network, the demand to create reliable network is also growing. The necessity of transferring various data, multimedia based crucial information across the world has drawn researchers to perform intense research on different aspect of networking. One of the prime aspects of network design is to choose the correct topology since cost and reliability of a network are mainly determined by its topology. The other network parameters like path rate, path length and load on each link varies with network topology. These factors are very critical for the network performance. Hence constructing a set of topology candidates prior to evaluation is important. To design a network topology satisfying the requisite constraints may be time consuming and as the size of the network grows subsequently the time for designing such candidate set also rises. Hence, computational time of an algorithm should be pragmatic. The paper addresses the problem of designing candidate set for a communication network so that there exists at least one disjoint path between every pair of nodes when a single link fails and the computational time is less than existing algorithms.

### Keywords

Candidate Topology; Fault Tolerant; Link Failure

**Received:** 12 May 2012; **Revised:** 08 July 2012; **Accepted:** 28 August 2012;
**Published:** 30 August 2012;

[1]Department of Electrical Engg, Jamia Millia Islamia, New Delhi, India.
[2]Dept of Computer Science & Engg., Maharaja Surajmal Inst. of Tech., New Delhi, India.
[3]Dept of Computer Science & Engg., Netaji Subhas Institute of Technology, New Delhi, India.

**Corresponding author:**
Email: aqansari@ieee.org

## Introduction

The use of network, both intranet and internet, to share important information by civil and defense organization requires robust communication network for secured communication. Robustness of a network is measured in terms of its continued ability to perform its function in the face of a failure (node or link or both). Hence "Network Topology" design should follow two criteria – Firstly the connectivity between the nodes and secondly the redundancy. A higher redundancy ensures that a network will survive higher fault rates but the cost of network also increases. Ideally, fault tolerant networks claim that each node has more than one route to reach the destination. If one route fails, another route will be used to keep data communication uninterrupted. On the other side, multiple routes can be used to transmit data concurrently. If multiple paths exist, traffic can switch quickly to an alternate path when a link or router fails. Similarly, if an adversary drops packets along a path, the traffic could be moved to an alternate path to circumvent the adversary[2]. This is particularly useful if disjoint paths are available. While generating topology candidates, it is to be checked for each candidate whether the connectivity between all pairs of nodes is satisfied at a single link failure as well as at normal operation[3].

A popular heuristic for generating a starting network is due to Steiglitz et al[7] and is called Link Deficit algorithm. This heuristic begins by numbering the nodes of a network at random or sequentially 1,2,3,…,n. In general when the Link deficit algorithm[7] is used, different numbering of nodes yields different starting networks.

Many approaches have been investigated to solve the topology design problem as an integer linear program (ILP). Ref.[1] and[9] showed heuristic approaches using a branch-and-bound method or a greedy method to solve the cost minimization problem with a constraint on the delay between nodes.

In the paper[12], the binary partition approach is applied to efficiently enumerate topology candidates maintaining connectivity at the single failure of any link. The number of candidate positions L where links can be put, is given by $L = n \times (n-1)/2$, where n is the number of nodes. All links were bidirectional and that packets were transmitted in both directions on each link[4]. The connectivity between all pairs of nodes is the minimum requirement for network topologies. By checking whether the minimum cost on the shortest route between any node pair is a finite value when the Dijkstra algorithm was applied on the topology generated by removing link from the candidate topology, and the connectivity between all pairs of nodes were tested at the single failure of link[5]. The calculation time required for this procedure was calculated as $O(n^2)$. This procedure was continued for all links, so the total calculation time of this algorithm was $O(Ln^2) = O(n^4)$ for checking the constraint for one link failure topology candidate[6].

Ref.[8] proposed and analyzed new irregular Fault-tolerant multistage interconnection network named as Irregular modified baseline network (IMABN). IMABN is dynamically re-routable and providing multiple paths of varying lengths between source and destination pairs[10].

In this paper, we consider enumerating all topologies maintaining the connectivity between all pairs of nodes at normal operation as well as at single failure of any link. The time complexity of the proposed algorithm is $O(n^3)$[11].

Section 2 presents a brief about different approaches for constructing fault tolerant network topologies. Section 3 of this paper we present the design of candidate topologies. In section 4 the proposed technique is explained with an illustrative example. The paper is concluded in section 5[13].

## Preliminaries

### *Some Important Definitions*

Reliability of a network is supported through alternative paths of transmission routes and also through the flexibility of remodeling the network on the fly. A network is reliable if it can endure node as well as link failure.

Cost of communication between two nodes in a network can defined by various metrics like distance between two nodes, no. of hops between two nodes, time of communication between two nodes[14].

A graph is 1-fault Tolerant iff there exists 2 disjoint paths between every pair of nodes in the network.

## Design of Candidate Topologies

In the proposed technique we have applied a top down approach to construct candidate set pertaining single link failure with a reduced time complexity.

In the beginning we explain the concept of designing the network topologies.

The geographical locations of the nodes are known apriori. The maximum number of links are considered as $L = n * (n - 1)/2$, n being the number of nodes. Connectivity between all nodes is treated as a prerequisite for designing the network topologies.

## Description of the Proposed Technique

The foremost thing is that given network with n nodes having connectivity between all pairs of nodes and ensures 1-fault tolerance. The proposed method is as follows

Assumptions:

1. All links considered are bidirectional.
2. All links have equal bandwidth.
3. The reliability of links are in the range 0 to 1.
4. The reliability of each node is 1.
5. The nodes are 1 hop apart from one another.

The nodes are numbered from 1 to n. The edges possible between the nodes are numbered so as to reduce the time in creating candidate set. The initial n graphs are created by adding all possible set of n-1 edges. The graphs are tested for 1-FT. Following this each graph is added with all possible combination of higher number edges and are tested for 1-FT. The graphs are marked if they are 1-FT and contain the highest number edge. The procedure of adding higher numbered edge and 1-Ft testing is continued till all the graphs are marked. All 1-Ft graphs generated in the process constitute the candidate set.

The flowchart of the method is presented (Fig1) depicting the course of action to generate an optimal 2-connected network given a set of n nodes.

## The Algorithm

Begin

Step1. In a given set of n nodes number the nodes sequentially, starting from any given node. The next node (clockwise order) corresponding to the starting node is marked as 2 and so on.

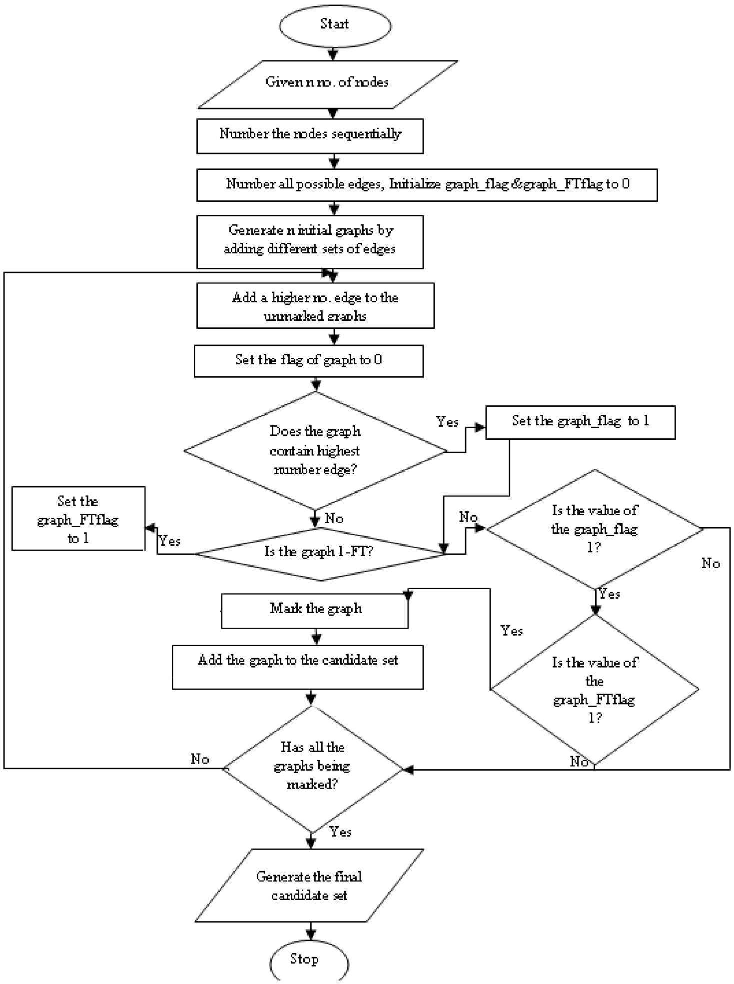


**Figure 1.** Flow diagram of the proposed method for designing optimal 2-Connected Network

Step2. All possible edges that can be present in a graph constructed using the given nodes are also numbered in the following way:

The link$(i, i+1)$ is numbered as i.

If $i = n-1$ then link$(i, i+1 \bmod n)$ is numbered as i. Where $i = 1$ to n

The link$(i, j)$ is numbered in ascending order considering minimum value for i and all possible j. Where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$; $j \neq i$ or $j \neq i+1$ or $j \neq i+1 \bmod n$ when $i = n$

Step 3. The graph_flag and graph_Ftflag are set to 0.

Step 4. To begin with n-1 edges are added to the given nodes in ascending order to generate the initial graph 1. For the remaining n-1 initial graphs n-1 edges are added each time starting with the jth edge where j = the initial graph number. The different sets of edges are added as follows:

```
1--2 , 2--3, ..... , n-1--n   for initial graph number 1
2--3, 3--4, ...... , n--1     for initial graph number 2
.
.
.
n--1, 1--2,....,n-2--n-1      for initial graph number n
```

Set the no. of edge for each node to 1.

Do

Step 5. For all the generated graphs

Do

Each graph is selected and all possible set of edges with number greater than the highest numbered edge present are added to generate different 1-fault tolerant network. Increment the no. of edges for the concerned nodes (connected by the newly added edge).

Add 1 to the degree of the nodes as an edge incident on the node is added to the graph.

Check if the graph contain highest number of edge. If yes set graph_flag to 1

To check whether the graphs generated are 1-fault tolerant or not

Do

Step 5a. if the degree$_i \geq 2 \; \forall \; i$ where $i = 1$ to n.

Repeat Step 5a until all nodes are tested. (at most n times i.e. the nodes in a graph).

Step 5b. If the graph does not pertain to 1-fault tolerance next higher edge is added else set graph_Ftflag to 1.

Step 5c. if graph_flag && graph_Ftflag ==1 Mark the graph Also mark the graph from which this graph is generated. All marked graphs will not be considered in successive steps.

Else if graph_flag==1 && graph_Ftflag=0 discard the graph go to Step 7.

Else if Graph_flag==0 && graph_Ftflag==1 go to Step 6.

Step 6. Add the 1-Ft graph to the candidate set.

Step 7. Repeat Step 5 until all graphs are marked. ($\approx n \times (n-1)$ times i.e. the no. of total unmarked graphs generated)

Step8. End.

## A Case Study

In this illustration we consider that four nodes are given and are arranged as shown in Fig.2

Now, we will construct all possible 1-fault tolerant candidate set for the above mentioned nodes using our proposed method.

By using Step1, in the given set of 4 nodes any given node at random is marked as 1. The next node (clockwise order) corresponding to the starting node is marked as 2 and so on. The numbered nodes are shown in Fig. 3.

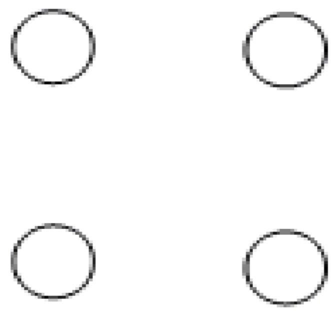

**Figure 2.** Geographical location of 4 nodes

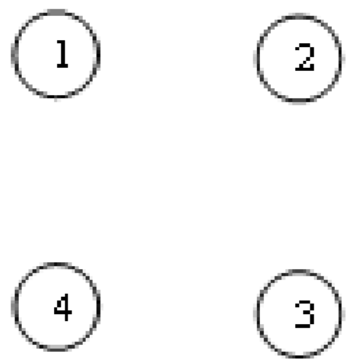


**Figure 3.** 4 nodes numbered sequentially.

With Step 2 of the proposed method, the possible edges that graph G can have using the given 4 nodes are numbered and shown in Fig. 4. Step 3 is used to initialize graph_flag and graph_FTflag to 0.

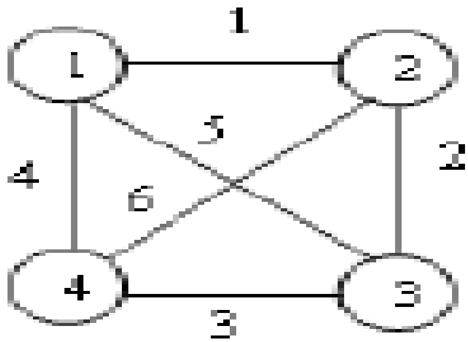


**Figure 4.** Numbering of all possible edges with numbered nodes.

With Step 4, since there are four nodes, 3 edges are added to the given nodes in ascending order to generate the initial graph 1. Then for the next initial graph the 3 edges are added starting with edge 2(between node 2 and node 3) and so on. The 4 initial graphs generated are shown in Fig. 5.

The different sets of edges are added as follows:

```
1--2, 2--3, 3--4
2--3, 3--4, 4--1
3--4, 4--1, 1--2
4--1, 1--2, 2--3
```

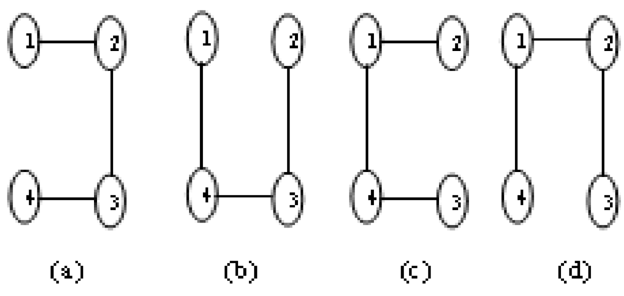


**Figure 5.** 4 initial graphs generated

Using Step5 all possible edges with number higher than the highest existing edge are added to each initial graph and the graphs generated are shown in Fig. 6. For graph in Fig. 5a edges 4,5,6 are added. Similarly for each graphs in Fig. 5b-d the edges added are 5, 6.

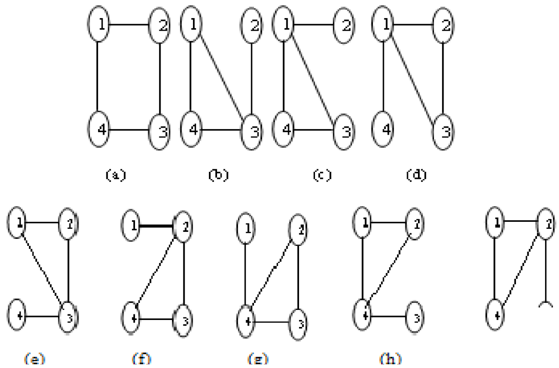


**Figure 6.** Graphs generated from initial graphs in 4a-d

Now, the graph generated as shown in Fig. 6a is 1-Fault Tolerant. The graphs in Fig. 6b-i is not 1-Fault Tolerant, so using Step 5b the graphs 6b-e are modified and shown in Fig. 7. The graphs in Fig. 6f-i are discarded.

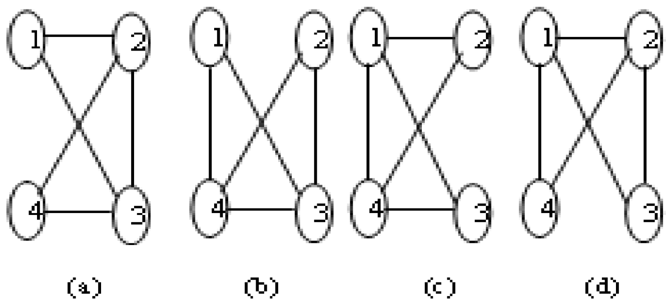


**Figure 7.** Graphs generated from graphs in figure 6b-e.

The graphs generated in Fig. 7 are 1-Fault tolerant. Since the graphs contain edge number 6 (the highest edge number) they are marked as per Step 5c. All graphs are not marked so according to Step 7, Step 5a is again visited.

Now the graph generated in Figure 6a is considered, using Step 5a, the edge number 5 is added as shown in Fig. 8.

The graph in Fig. 8 is 1-Fault tolerant. Now since all graphs are not marked Step 5 is again computed. The graph from Fig. 6a and 8 are used to generate successive graphs. The graphs generated are shown in Fig. 9.

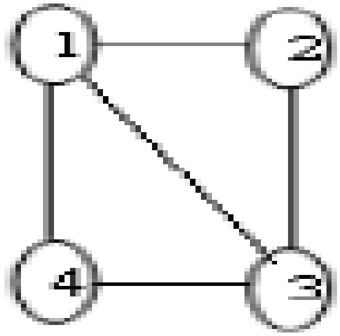


**Figure 8.** Graph generated from figure 5a by adding edge number 5.

Using Step 5a, Step 5b and Step 5c the graphs (Fig. 9) are 1-Fault Tolerant. Since both the graphs contain edge number 6 (the highest edge number), they are marked along with their respective initial graphs.

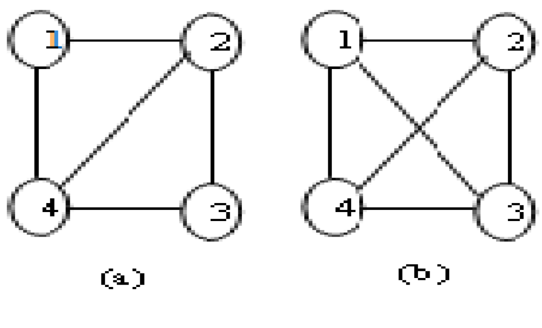


**Figure 9.** Graphs generated from figure 6a and figure 8 by adding edge number 6.

In Step 7 all graphs are marked. So Step 8 is visited. The marked graphs are the resultant candidate set for atleast 1-fault tolerant network using 4 nodes (shown in Fig. 10).

## Performance Evaluation

To evaluate the performance of the algorithm we consider n random nodes, where n= $\{4, 5, 7, 9, 10, 12, 15, 18, \text{and } 20\}$. The cost of link between each node is set to 1(for single hop distance between each pair of node). The time complexity of the proposed algorithm is $O(n^3)$ since the number of iteration is n(no. of nodes)$\times(n \times (n-1)/2)$(maximum no. of unmarked graphs generated in this technique) times

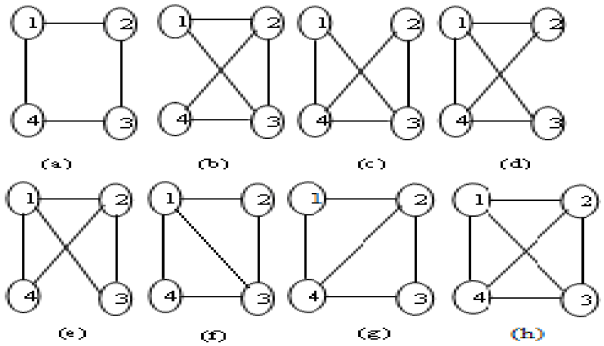


**Figure 10.** Candidate Set of 1-Fault tolerant Network using 4 nodes.

for generating a graph of n nodes and checking it for 1-fault tolerance. The computational time is less than other existing algorithm ref[12] which is $O(n^4)$ (shown in Fig. 11), n being the number of nodes.

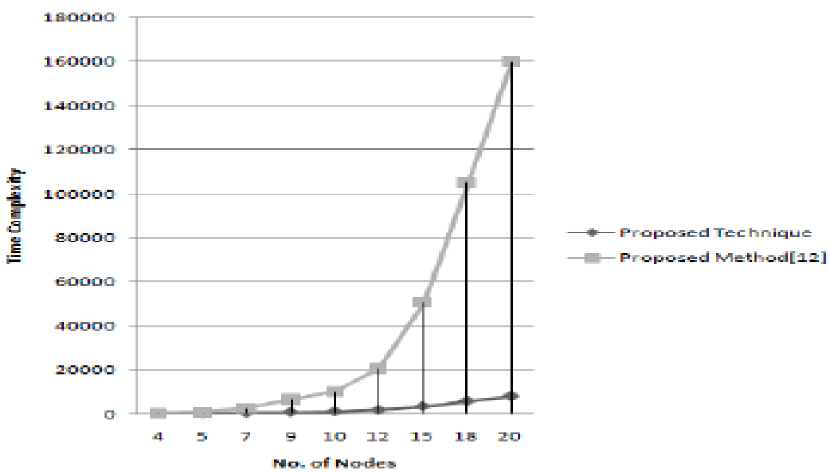


**Figure 11.** Performance Evaluation with various number of nodes

## Conclusion

The balancing act between better reliability and low cost design is essential in constructing the candidate set for a large scale network. To enumerate topology candidates in practical time in large scale and keeping the design cost low, setting a lower limit of the number of failures in link is necessary. In this paper, a linear top-down approach is proposed to specify topology candidates maintaining connectivity at a single failure as well as under normal operational condition. The time complexity of the proposed technique is $O(n^3)$ which is optimal since designing and testing the graph of n nodes for edge fault tolerance would require at least $n \times n \times (n-1)/2$ iterations. In future work, expansion of the proposed method for selecting the candidates considering parameters like path rates, and load distribution can be included. The algorithm can be upgraded for multiple fault tolerance by testing the graphs for k-fault tolerance $(1 \leq k \leq (n-2))$ and eliminating the other candidate set, however the time complexity will remain unchanged.

## References


[1] A. Gersht and R. Weihmayer, "Joint Optimization of Data Network Design and Facility Selection," IEEE J. Selec. Areas, Commun., 8, 9, pp. 1667–1681, 1990.

[2] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot, "Characterization of Failures in an IP Backbone, IEEE Infocom 2004.

[3] Bin Dai, Huabiao Lu, Zhigang Sun, Ziming Song, Yanpeng Ma and Jinshu Su, "MORT: A Technique to Improve Routing Efficiency in Fault-tolerant Multipath Routing," Fifth International Conference on Mobile Ad-hoc and Sensor Networks, pp. 430–435, 2009.

[4] B. L. Golden, E.A. Wasil, and P.T. Harker, "The Analytic Hierarchy Process," Springer-Verlag, pp. 37–58, 1989.

[5] E. C. G. Wille, M. Mellia, E. Leonardi, and M. A. Marsan, "Topological Design of Survivable IP Networks Using Metaheuristic Approaches," Lecture Notes in Computer Science, Springer-Verlag, pp. 191–206, 2005.

[6] K. Cheema, R. Aggarwal, " Design Scheme and Performance Evaluation of new Fault-Tolerant Multistage Interconnection Network", IJCSNS International Journal of Computer Science and Network Security, VOL.9 NO.9, September 2009.

[7] K. Steiglitz, P. Weiner, and D. J. Kleitman, "The Design of Minimum-Cost Survivable Networks," IEEE Trans. Circuit Theory, CT-16, 4, pp.455–460, 1969.

[8] M. Ghai, K. Cheema and Vinay Chopra, "Design and Reliability Analysis of New Fault-Tolerant Irregular Multistage Interconnection Network," International Journal of Computer Applications (0975 – 8887) Volume 10– No.1, pp. 1–5, November 2010.

[9] N. G. Chattopadhyay, T. W. Morgan, and A. Raghuram, "An Innovative Technique for Backbone Network Design, IEEE Trans. System, Man, and Cybernetics, 19, 5, pp. 1122–1132, 1989.

[10] N. Kamiyama and D. Satoh, "Network Topology Design using Analytic Hierarchy Process, IEEE ICC 2008.

[11] N. Kamiyama, R. Kawahara, and H. Yoshino, "Network Topology Design Considering Detour Traffic Caused by Link Failure, VDE Networks 2008.

[12] N. Kamiyama," Efficiently Constructing Candidate Set for Network Topology Design," IEEE ICC proceedings, 2009.

[13] T. Uno, "Enumeration Algorithms and Speeding up, 11th RAMP symposium, 1999.

[14] P. Gupta and P. R. Kumar. Capacity of wireless networks. IEEE Transactions on Information Theory, 2000.