

並列プロセッサ DAPDNA-2 を用いたリンクディスジョイント経路計算の 高速解法

木原 拓[†] 清水 翔[†] 高山[†] 荒川 豊[†] 山中 直明[†]
渡辺 昭文^{††}

[†] 慶應義塾大学理工学部情報工学科 〒223-8522 神奈川県横浜市港北区日吉 3-14-1
^{††} アイピーフレックス株式会社 〒141-0021 東京都品川区上大崎 2-27-1 サンフェリスタ目黒 6 階
E-mail: †{kihara,shimizu,gao,arakawa}@yamanaka.ics.keio.ac.jp, ††yamanaka@ics.keio.ac.jp,
†††akifu@ipflex.com

あらまし 次世代のネットワークには高い水準の信頼性が求められている。高い信頼性を実現するための技術の1つであるプロテクションでは、リンクディスジョイントな2つの経路の合計コストをできるだけ低くすることが、リソースの効率的な利用の点で重要な課題となっている。しかしながら、経路の合計コストが最も低い最適解を求めるための計算はNP困難であり、膨大な時間がかかってしまう。そこで、本稿では並列プロセッサ DAPDNA-2 を用いて、リンクディスジョイント経路計算における最適解を高速に算出する方法を提案する。提案方式では、全リンクパターンの中から制約条件を基にネットワーク中の全経路を抽出し、並列組合せ計算を行うことにより、最適解を高速に算出する。特性評価では、従来の最適解を求めるための高速化アプローチと比較して、提案方式が20倍以上高速化できることを示す。

キーワード プロテクション, リンクディスジョイント経路計算, 並列リCONFIGURABLEプロセッサ, DAPDNA-2

Fast Solution of Link Disjoint Path Algorithm on Parallel Reconfigurable Processor DAPDNA-2

Taku KIHARA[†], Sho SHIMIZU[†], Gao SHAN[†], Yutaka ARAKAWA[†], Naoaki YAMANAKA[†], and
Akifumi WATANABE^{††}

[†] Dept. of Information and Computer Science, Faculty of Science and Technology, Keio University Hiyoshi
3-14-1, Kohoku-ku, Yokohama-shi, Kanagawa, 223-8522 Japan
^{††} IPFlex Inc. Kamiosaki 2-27-1, Sun felista Meguro 6F, Shinagawa-ku, Tokyo, 141-0021 Japan
E-mail: †{kihara,shimizu,gao,arakawa}@yamanaka.ics.keio.ac.jp, ††yamanaka@ics.keio.ac.jp,
†††akifu@ipflex.com

Abstract In next generation network, a high level reliability is strongly required. In a protection, which is one of network survivable technologies, it is important to use a link-disjoint paths pair of lowest total cost in a view of network resource efficiency. However, the link-disjoint paths pair calculation is NP Complete and it takes great investment of time. In order to solve this problem, we propose a high-speed method of link-disjoint paths pair calculation based on parallel reconfigurable processor. Our proposed method calculates and finds all paths in network from all network link patterns, and then calculates all link-disjoint paths pairs and selects optimal solution. A parallel execution and data flow execution enable our propose to high-speed calculations. A two performance evaluation results show that our proposed method improves calculation time more than 20 times compared to conventional optimal link-disjoint paths pair calculation algorithm.

Key words Protection, Link-disjoint paths calculation, Parallel Reconfigurable Processor, DAPDNA-2

1. はじめに

インターネットの急速な普及に伴い、ネットワークは進化の一途をたどっている。ネットワークは年々、高速・大容量化が進み、アメリカが構想する次世代ネットワークである CORONET [1] のように、100Gbps 超高速通信が主流となる日は近い。このような超高速通信を扱うネットワークにおいては、リンクおよびノード故障による通信の遮断が引き起こす大量のデータロスが非常に大きな問題となる。そのため、プロテクションに代表される耐障害技術を用いて、ネットワークに信頼性を持たせることが必要不可欠である。

プロテクションでは通信を行う前に、故障発生時に切替えて使用するための予備経路を確保する。通常時に使用する経路をプライマリパス、故障発生時に切替えて使用する予備経路をバックアップパスと呼ぶ。耐障害性の観点からプライマリパスとバックアップパスは、互いに同じリンクを使用しない(これを互いにリンクディスジョイントであると言う)という条件を満たす必要がある。また、ネットワークリソースの効率的な利用の観点からプライマリパスとバックアップパスの2経路の合計コストをできる限り低く抑える必要もある。この2条件を満たすようなプライマリパスとバックアップパスをいかにして算出するかがプロテクションの重要な課題である。この2経路を求めるための計算はリンクディスジョイント経路計算と呼ばれ、従来から広く研究されている。従来のリンクディスジョイント経路計算は、大きく2つの方式に分けることができる。Integer Linear Programming (ILP) を用いる方式とダイクストラ法 [2] に基づくヒューリスティックを用いる方式である。しかしながら、ILP では合計コストの最も低いリンクディスジョイントな経路の組、すなわち最適解を求めることが可能だが、計算量が膨大なため計算に時間がかかってしまうという欠点があり、一方、ダイクストラ法に基づくヒューリスティックでは、計算量が少ないため解が高速に算出できるが、最適解が出せない場合があり、ネットワークリソースの利用効率が落ちてしまうという欠点がある。このように、従来のリンクディスジョイント経路計算は計算時間と解の精度の間でトレードオフであった。

そこで、本稿ではリンクディスジョイント経路計算において最適解を高速に算出する方式を提案する。提案方式では、 $2^{\text{link}} - 1$ 個のネットワーク中の全リンクパターンを列挙し、その中から始点と終点を結ぶ条件を持つパターン、すなわち正当性を持つ経路のみを的確に抽出し、全リンクディスジョイントペアを算出する。全リンクディスジョイントペアの中から合計コストの最も低いものを選ぶことによって常に最適解を導出することが可能である。さらに、提案方式を IPFlex 社 [3] が開発した並列リコンフィギュラブルプロセッサ DAPDNA-2 [4] 上に実装し、データフローおよび並列処理による、最適解の導出に必要な計算の高速化を図る。DAPDNA-2 は、32bit の専用 RISC プロセッサ (DAP) と DNA と呼ばれる並列処理エンジンの、大きく2つから構成される。DNA には PE (Processing Element) と呼ばれる演算器が、マトリックス状に合計 376 個配置されている。複数の PE を接続することでデータフローマシンを生成

し、並列演算が可能である。

本稿の残りは以下のように構成されている。2章では、現在までに研究されている様々なリンクディスジョイント経路計算の解法を紹介し、その問題点について説明する。3章では、提案方式の概要を説明と提案方式の DAPDNA-2 上への実装について説明する。続く4章では、提案方式と従来方式の実行時間と解の精度を比較する。最後に5章にて結論を述べる。

2. 関連研究

従来のリンクディスジョイント経路計算は、大きく分けて ILP を用いた方式とダイクストラ法に基づくヒューリスティックの2つの方式がある。ILP を用いた方式では、[5], [6], [7], [8] などがあるが、いずれも帯域や距離などの様々なメトリックの合計を最小化するような最適解を求めることが可能だが、いずれも数秒~数十秒単位という計算時間を要する。次世代のネットワークでは、現在以上に動的な計算要求が寄せられるため、数十ミリ秒程度で計算が実行できる必要があり、これらの方式では対応できないと考えられる。

ダイクストラ法に基づくヒューリスティックでは、[9] にある Active Path First (APF) が最も有名である。しかし、APF は最適解が見つからない場合が存在することや、そもそも解自体が見つからなくなってしまうというトラップ問題を潜在的に有することが知られている [10]。この問題を解決するために、解の候補となる経路を複数求めそれを元に計算を行う K -Shortest Path アルゴリズム [11] が考案されているが、ネットワークの規模や形状に伴い、 K の値をどのように設定するかといった具体的な事例に対する研究はされていない。その他にも、[12], [13] などのように様々なヒューリスティックや [14] のようなハイブリッド方式などが提案されているが、性能の評価が特定のネットワークのみであるなど、いずれも最適性に関する評価が足りていない。これらヒューリスティックは、早いものでは実行時間が数ミリ秒以下と、ほとんどのものが ILP と比較して大幅に実行時間を短縮しているが、解の精度に問題があるため、ネットワークリソースの利用効率が下がってしまう可能性がある。

このように、リンクディスジョイント経路計算における問題は高速性と解の精度の間のトレードオフである。次章において高速性と解の精度の両方を併せ持つ方式を提案する。

3. 提案方式

本節では、並列リコンフィギュラブルプロセッサ DAPDNA-2 を用いてリンクディスジョイント経路計算における最適解を高速に算出する方式を提案する。提案方式はビット列を用いてネットワーク中から最適解の候補を全て抜き出し、その中から最適解を選出する。

3.1 概要

提案方式は、全トポロジパターンの生成、トポロジ正当性判定、リンクディスジョイント判定、最小値検出の4つの工程

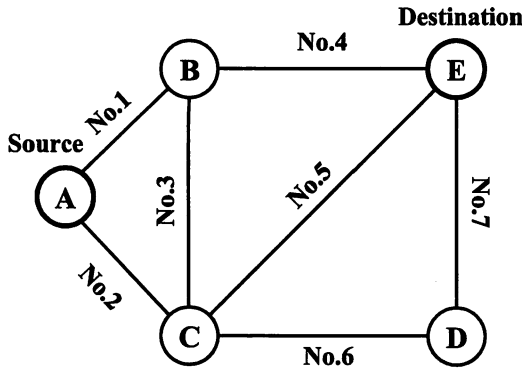


図1 7リンクネットワーク

から構成される。全トポロジパターンの生成では、ネットワーク中の全てのリンク使用パターンをビット列を用いて生成し、トポロジの正当性判定では生成したトポロジパターンの中から始点と終点を結ぶ経路を抽出する。続く、リンクディスジョイント判定では、トポロジ正当性判定において抽出した経路の中から互いにリンクディスジョイントな経路のペアを全て求める。そして、最小値検出では、全リンクディスジョイントペアの中からコストが最小になるペア、すなわち最適解を算出する。以降の節で4つの工程の詳細を説明する。

3.2 アルゴリズム詳細

3.2.1 全トポロジパターンの生成

この工程では、ネットワーク中の全トポロジパターンを生成する。トポロジパターンとは、図2に示すようにネットワーク中のどのリンクを使用したトポロジかをビット列で表したものであり、ネットワーク中のリンク数と同じデータ幅を持つ。また、トポロジパターンの各ビットは、ネットワーク中の各リンクに対応する。例えば、図1のネットワークは7リンクから構成されているため、このネットワークに対応するトポロジパターンは7ビットから構成される。7ビットは最右ビットから順にリンク No.1, No.2, ..., No.7に対応しており、0x000001はリンク1を使用するトポロジを、0x0010101はリンク1, 3, 5を使用するトポロジを表す。このトポロジパターンを Beeler 法 [15] を用いて、使用リンク数、すなわち1の数の少ないトポロジパターンの順に、 $2^{\text{link}} - 1$ 個分全て生成する。当然ながら、生成したトポロジパターンの中には始点と終点を結ぶ、所望の経路では無いものも多く含まれるため、それらを判定によって取り除く必要がある。

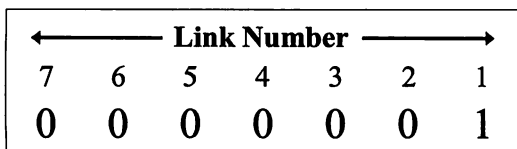


図2 トポロジパターン

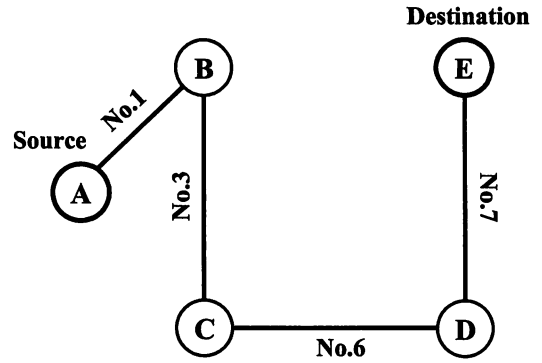


図3 正当性を持つトポロジ

3.2.2 トポロジ正当性判定

トポロジの正当性判定では、生成した各トポロジパターンが始点と終点を結ぶ経路であるかを判定する。正当性を持つトポロジであるためには、図3のトポロジのように始点、終点間が一筆書き可能である必要がある。一筆書きが可能であるためにはオイラー路 [16] の条件を満たす必要がある。提案方式では、この条件を元に正当性を持つ条件を以下のように定めた。

[条件] 始点ノードおよび終点ノードのリンク出次数が1、残りのノードのリンク出次数が全て偶数

上記の条件を満たすトポロジが正当性を持つトポロジである。そのため、各トポロジパターンにおける各ノードの出次数が条件を満たすかどうかを判定することで正当性を持つトポロジを抽出可能である。判定にはノード出次数カウンタを用いる。ノード出次数カウンタとは図4に示すようにビット列で表され、ネットワーク中のノード数と同じデータ幅を持つ。各ビットは、最右ビットから順にノード A, ノード B, ..., と、ネットワーク中の各ノードに対応する。ノード出次数カウンタは各トポロジパターンにおいて、使用リンクの両端ノードを数えることで作成される。例えば、図1のネットワークにおけるトポロジパターン 0x0100101 に対応するノード出次数カウンタは、リンク1の両端ノードであるノード A, ノード B を数えた 0x00011, 同じくリンク3の両端ノードを数えた 0x00110, リンク6の 0x01100 を加算することによって、0x01221 と求めることができる。

このようにトポロジ正当性判定では、 $2^{\text{link}} - 1$ 個の全てのトポロジパターンについて判定を行う。

Link degree of Each node				
E	D	C	B	A
0	1	2	2	1

図4 リンク出次数カウンタ

表 1 ディスジョイント判定

Path	Topology pattern
P1	0x0001001
P2	0x0010010
P3	0x0001110

3.2.3 リンクディスジョイント判定

リンクディスジョイント判定では、始点と終点を結ぶ経路の中から全てのリンクディスジョイントペアを算出する。リンクディスジョイントであるかの判定はトポロジパターン同士のビット積の結果を用いる。ある2経路がリンクディスジョイントペアである場合は、2経路のトポロジパターンのビット積の結果が0になる。逆に、ある2経路がリンクディスジョイントペアで無い場合、2経路のトポロジパターンのビット積の結果において、重複するリンク部分のビットが1になる。実際に、表1の3経路の中から全リンクディスジョイントペアを算出する例を考える。P1: 0x0001001とP2: 0x0010010は互いのビット積の結果が0x0000000になる。すなわちP1とP2はリンクディスジョイントペアである。次にP1: 0x0001001とP3: 0x0001110を調べる。P1とP3は互いのビット積の結果が0x0001000になる。これは右から4番目のビットが1であることからこの2経路はリンク4が重複していることが分かる。すなわちリンクディスジョイントペアでは無い。同様にして、P2とP3もリンク2に重複があるため、リンクディスジョイントでは無い。そのため、この3経路におけるリンクディスジョイントペアはP1とP2のみであるということが分かる。このようにして、全経路の組み合わせの中からトポロジパターンのビット積の結果が0になるものを発見することにより、全リンクディスジョイントペアを算出することが可能である。

3.2.4 最小値検出

最小値検出では、全リンクディスジョイントペアの中で合計コストが最も低いものを算出する。提案方式では、ネットワーク中の始点と終点を結ぶ全リンクディスジョイントペアを計算するため、算出された解は必然的に最適解となる。合計コストは各トポロジパターンの使用リンクのコストを用いて計算する。全てのペアの合計コストを計算し、その値を順次ピーク値検出回路にかけることにより、最適解を算出する。ここで、ピーク値検出とは、連続して入力されるデータの中から最も低い値を算出することができる回路である。

3.3 コンフィギュレーションの分割

図5に、DAPDNA-2上でのコンフィギュレーションを示す。提案方式は、3つのコンフィギュレーションを切替えてアルゴリズムを実行している。コンフィギュレーションを分割することにより、各コンフィギュレーション毎に376個の全てのPEを使用することが可能となり、提案方式の各工程における演算において高い並列度を実現している。上記の通り、コンフィギュレーションを分割した場合、DAPDNA-2上で4並列を実現することが可能である。そのため、トポロジ正当性判定やリ

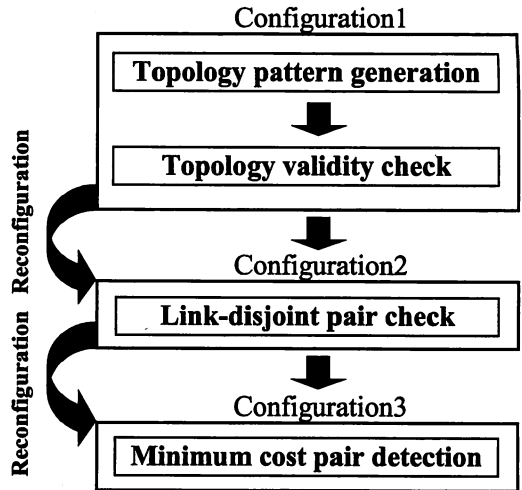


図5 コンフィギュレーションの分割

ンクディスジョイント判定のように計算頻度の非常に高い項目も高速に処理することが可能である。

4. 特性評価

特性評価では、3.節で説明した提案方式と、ILPを用いた方式[5],[6]、ダイクストラベースのヒューリスティックであるActive Path First (APF)[9]の2方式の計3方式について実行時間と解の精度についての比較を行った。提案方式はIPFlex DAPDNA-2 166MHz用評価ボードであるDAPDNA-2 EB4上に実装し、2つの従来方式は動作周波数が19倍のIntel PentiumIV 3GHzを用いた。また、様々なノード数と平均リンク次数においてランダムトポロジを発生させ、リンク数が $L = 16, 17, \dots, 32$ のトポロジを各1000個用意した。各方式の実行時間と解の精度の値は1000個のトポロジにおける平均値である。

さらに、提案方式におけるパラメータとして最大ホップ数制限 α を設定し、ネットワーク中の全トポロジパターンを生成して計算した場合と、ホップ数、すなわち使用リンク数が α 以下のトポロジパターンのみを生成して計算した場合の計算時間と解の精度の比較も行った。例えば、 $\alpha = 7$ と設定した場合には、1の数が7,6,...,1のトポロジパターンを生成し、その中から解を算出する。この時、算出した解、つまりプライマリパスとバックアップパスの組のホップ数(合計ホップ数では無いことに注意。長い方のバックアップパスのホップ数)は必ず7以下となる。このように、最大ホップ数制限 α を用いることにより、ホップ数が α という条件下での最適解(全体の最適解とは限らない)を取得することが可能である。比較においては、 $\alpha = \infty$ (最大ホップ数制限無し)と $\alpha = 2\log N$ の値を用いた。 $2\log N$ は、ランダムネットワークにおける平均バックアップパス長[17],[18]を元にして決定した値である。

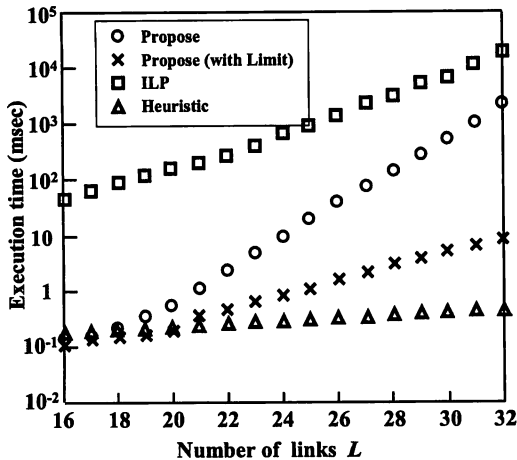


図 6 実行時間の比較

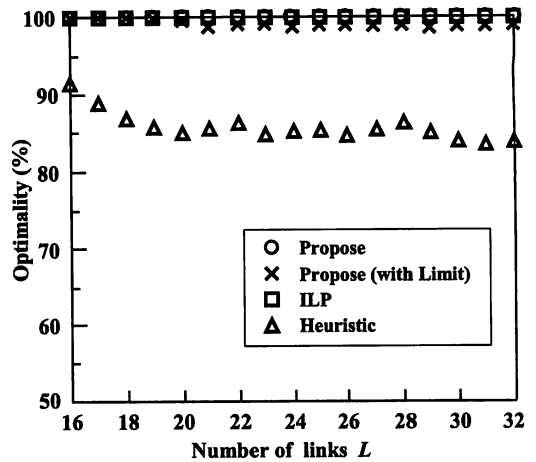


図 7 解の精度の比較

4.1 実行時間の比較

リンク数 L に対する各方式の実行時間の比較を図 6 に示す。グラフから提案方式が ILP よりも実行時間が早いことが分かる。これは、提案方式における DAPDNA-2 の並列処理とデータフローによる高速化が有効であることを示している。また、提案方式では最大ホップ数制限を設けることで、最適解として選ばれる可能性の高いトポロジデータのみを計算の対象として絞ることによりさらなる高速化が可能であることが分かる。 $L = 32$ においては、ILP と比較して、提案方式が最大ホップ数制限なしの場合で 20 倍、最大ホップ数制限ありの場合で 3000 倍高速であることが分かる。ヒューリスティックには及ばないが、最大ホップ数制限を設けることによって、提案方式とヒューリスティックの実行時間の差を 100 倍以内に近づけることが可能であることも分かる。

4.2 解の精度の比較

リンク数 L に対する各方式の解の精度の比較を図 7 に示す。グラフの縦軸は各 L における 1000 トポロジにおいて最適解を算出できた割合である。グラフより提案方式は ILP と同様、100% の割合で最適解を算出することが可能であることが分かる。これは提案方式がネットワーク中の中から全てのリンクディスジョイントペアを抽出し、その中から解を選択するためである。提案方式が ILP より早い実行時間で最適解を計算可能なことが分かる。

さらに、提案方式は最大ホップ数の制限を設けた場合でも 99% 以上の割合で最適解を求めることが可能であり、最適解を求められる割合が 90% を下回る従来のヒューリスティックと比較して非常に高い解の精度を持つことが分かる。最大ホップ数制限により、最適解として選ばれる可能性の高い経路のみを抽出して解を計算することが非常に有効であることを示している。提案方式では、最大ホップ数制限を用いることによって解の精度をほとんど落とさずにさらなる高速化を図ることも可能

である。

5. 結 論

本稿では、従来のリンクディスジョイント経路計算の問題点であった計算時間と解の精度の間のトレードオフを解決するために、並列プロセッサ DAPDNA-2 を用いてリンクディスジョイント経路計算における最適解を高速に算出する方法を提案した。提案方式はビット演算を基調とした全解探索を用いることにより、DAPDNA-2 の性能を最大限に活かして、高速に最適解を発見することを可能にした。DAPDNA-2 の評価ボード上に実装した提案方式が、動作周波数 19 倍の Pentium IV 上に実装した従来の 2 つのリンクディスジョイント計算法と比較して、20 倍早い時間で最適解を求められることを示した。さらに、提案方式は、最適解として使用される可能性の高い、ホップ数の少ない経路のみを計算の対象として絞ることによって、3000 倍早い時間で 99% 以上の割合で最適解を発見することが可能であり、従来のヒューリスティックよりも優れた性能を持つことが分かり、その有効性を示した。

謝 辞

本研究はグローバル COE プログラム「アクセス空間支援基盤技術の高度国際連携」(C12) と科研費 19360178(B) の助成を受けたものである。関係者各位に深謝する。

文 献

- [1] E.W.Dijkstra, "A note on two problems in connection with graphs," Numerische Mathematik, vol.1, pp.269-271, 1959.
- [2] E.W.Dijkstra, "A note on two problems in connection with graphs," Numerische Mathematik, vol.1, pp.269-271, 1959.
- [3] IPFlex Inc (<http://www.ipflex.com>)
- [4] Toshinori SUEYOSHI, Hideharu AMANO, Reconfigurable system, Ohmsha, Tokyo, 2005.
- [5] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximat-

- ing optimal spare capacity allocation by successive survivable routing," in Proc. IEEE INFOCOM, 2001, pp. 699-708.
- [6] D. Xu, C. Qiao, and Y. Xiong, "An ultra-fast shared path protection scheme: distributed partial information management, part II," in Proc. IEEE ICNP, Paris, France, Nov. 2002, pp. 344-353.
 - [7] G. Li et al., "Efficient distributed path selection for shared restoration connections," in Proc. IEEE INFOCOM, New York, Jun. 2002, pp.140-149.
 - [8] M.Kodialam and T.V.Lakshman, "Dynamic routing of bandwidth guranteed restoration," in Proc. IEEE INFOCOM, 2000, pp.902-911
 - [9] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125-145, 1974.
 - [10] P. Laborczi et al., "Solving asymmetrically weighted optimal or near-optimal disjoint path-pair for the survivable optical networks," presented at the 3rd Int.Workshop On Design Of Reliable Communication Networks (DRCN ' 01), Budapest, Hungary, Oct. 2001.
 - [11] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125-145, 1974.
 - [12] J.Surballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125-145, 1974.
 - [13] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999.
 - [14] Quoc V.Phung, Daryoush Habibi, Hoang N. Nguyen, and Kungmeng Lo, "K Pairs of Disjoint Paths Algorithm for Protection in WDM Optical Networks," APCC 2005, pp. 183-187, 2005.
 - [15] M.Beeler, R.W.Gosper, R.Schroeppel, HAKMEM (<http://www.inwap.com/pdp10/hbaker/hakmem/hakmem.html>)
 - [16] 小松勇作: 解析概論 [1], 広川書店.
 - [17] 増田直紀, 今野紀雄 複雑ネットワークの科学, 産業図書, 2005.
 - [18] Dahai Xu, Yang Chen, Yizhi Xiong, Chunming Qiao, and Xin He, "On the Complexity of and Algorithms for Finding the Shortest Path With a Disjoint Counterpart," *IEEEACM TRANSACTIONS ON NETWORKING*, Vol. 14, No. 1, Feb. 2006.