

遺伝的アルゴリズムによる ネットワーク信頼性向上問題解法の一手法

上田正直 松原隆 古賀義亮

防衛大学校 情報工学教室

239-8686 神奈川県横須賀市走水 1-10-20

0468-41-3810 (内線 2219)

{ masanao, matubara, koga }@cc.nda.ac.jp

あらまし

近年のコンピュータネットワークの普及に伴い、その故障が多大な社会的影響・経済的な損失をもたらすため、その防護対策として経路の多重化手法がとられている。ここでは、まずグラフ理論の分野で理論的に検討がすすめられている「枝増加問題」を紹介する。この問題は与えられたグラフについて、ある連結度 k を満たすような枝集合を加え、グラフの連結度を増大させることを目的としたものである。これは既存のネットワークに対して最小コストによって、要求する経路の多重化を行うことに適用できる。グラフ理論でのこの問題はNP困難とされている。本研究では、遺伝的アルゴリズムを用いて枝増加問題の近似解法を提案するとともに、その手法による結果を実験的に示してネットワークの最小コストによる経路多重化の課題解決の一手法を与える。

キーワード 枝増加問題, 遺伝的アルゴリズム, コンポーネント, ネットワーク信頼性

A solution for Dependable Network Composition by genetic algorithms

Masanao UEDA, Takashi MATSUBARA, Yoshiaki KOGA

Department of Computer Science, Natinal Defense Academy

239-8686 Hashirimizu 1-10-20 Yokosuka JAPAN

0468-41-3810 (ex.2219)

{ masanao, matubara, koga }@cc.nda.ac.jp

Abstract

Now a day, various activities are performed through computer networks, and troubles in the networks may cause serious social and life critical problems. The studies of “ k -edge connectivity augmentation” are presented so as to protect troubles by cutting edges, that is, cutting communication links of computer networks. But weighted edge augmentation problems are well known to be solved with non-polynomial order in the graph thory. In this paper, we present some genetic algorithms to solve the problems with experimental results toward dependable network composition.

key words edge augmentation problem, genetic algorithm, component, dependable network

1 まえがき

近年、ネットワーク接続を伴うコンピュータが様々な業種において広く普及し、ネットワークによるデータ伝送が業務遂行上きわめて緊要な役割を果たすようになってきている。それだけに、ネットワークの故障が多大な社会的・経済的損失をもたらすケースが増大し、一般的に行われている伝送経路の2重化等では必ずしも十分な故障対策とはいえない程になっている。

伝送経路をさらに多重化して故障対策の課題を扱うために、グラフ理論の k 枝増加問題を用いることができる。すなわち伝送路及び伝送路間端末をグラフの枝及び節と対応させ、伝送路の多重化を枝増加問題として扱えばよい。ここで、既存のネットワークの形、コストをどのように取り扱うかという条件により、多項式時間で解けるもの、NP問題となるものに分類される[3]。現実のネットワークに課せられた課題を取り扱うために、コストを考慮した枝増加問題は多項式時間で解けない。

そこで本研究では、現実のネットワーク構成を目的として、通信路の多重化をなるべく低いコストで実現するため、 k 枝連結度増加問題の遺伝的アルゴリズムによる解法を提案する。提案する方法では、致死遺伝子の発生を抑えるために連結度を考慮した突然変異・交叉を行い、より効率的な探索を図っている。この手法に関するプログラムを開発し様々なケースについて実行したのでその結果について述べる。

2 枝増加問題

2.1 問題の定義

$G = (V, E)$ は点集合 V と枝集合 E からなるグラフである。枝連結度とは、グラフ G の任意の枝を除去した際に、グラフが非連結となる最小の枝の数である。

枝連結度増加問題とは、“ $G = (V, E)$ の部分グラフ $G_0 = (V, E_0)$ に対し、枝連結度 k を満たすような最小の加える枝集合 Aug ($Aug \subseteq E - E_0$) を求める”問題である。ここで k 枝連結グラフとは、任意の $k - 1$ 本の枝を除去しても連結しているグラフである[1]。重みなし完全グラフ $G = (V, E)$ の部分グラフ $G_0 = (V, E_0)$ が図1のように与えられたとき、 G_0 を3枝連結グラフにする枝集合 Aug は図2のとおりである。

2.2 既往の研究

この問題は、1976年にEswaranとTarjanが最小の枝集合をグラフに加えて、2連結グラフにする問題を紹介したところに始まった[ET76]。その翌年、HsuとRamachandranにより3連結グラフに拡張される[HR91]。その後、問題の一般化(法

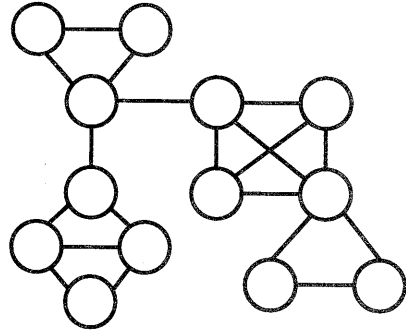


図 1: 与えられたグラフ $G_0 = (V, E_0)$

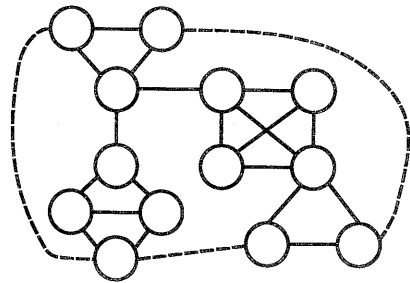


図 2: 連結度を増加させたグラフ $G'_0 = (V, E_0 \cup Aug)$

則化), アルゴリズムの高速化の研究が行われる。その後の主な成果は、以下の通りである。

- $k = 2$ かつ $G_0 = \emptyset$
 - 重み付き: NP Complete [2]
 - 重みなし: $O(V + E)$ [2]
- $k = 2$ かつ G_0 が木であるとき
 - 重み付き: NP Complete [4]
 - $O(V^2)APP(\leq 2 * optimum)$ [4]
- $k \geq 2$ かつ G_0 が木であるとき
 - 重みなし: 整数時間 [5]
- $k \geq 2$ かつ G_0 が任意のグラフ
 - 重み付き: NP Complete [6] [7]
 - 重みなし: $O(kLV^3(kV + E))$ [3]
 - (ただし, $L = \min \{k, V\}$)

実際のネットワークについて考えてみると、既存のネットワークは任意に与えられ、さらに構築の際にはコストも考慮する必要がある。このような場合、枝増加問題はNP問題になる。したがって、これを解くには近似アルゴリズムを構築する必要がある。そこで本研究では、遺伝的アルゴリズムを用いて、枝増加問題の近似アルゴリズムを提案する。

3 遺伝的アルゴリズム

3.1 あらまし

遺伝的アルゴリズムは、多量のデータから与えられた条件に適合するデータを選択する手法として用いられている [8]。一般的には選択手法が NP 問題となる場合を対象としている。データは生物学的にいえば表現型であり、データをアルゴリズムで扱えるように変換したものが生物学的にいえば遺伝子型である。遺伝的アルゴリズムでは、オペレータの操作対象が遺伝子型であり、適合度の選択は表現型に依存する。これらの操作において、優性種すなわち適合度の高い個体がより多くの子孫を残し、低い個体は淘汰されるように選択する。ここで適合度の高いものを多産にし、低いものを淘汰するような選択の方法が必要になる。その後、選択された個体の生殖の際に行われるのが交叉である。この交叉には 1 点交叉、2 点交叉、一様交叉などが挙げられる。また、この生殖の際に突然変異も生起させる。

このように選択・淘汰・生殖を繰り返すことによって、世代が進むにつれ集団全体の適合度が改善し優れた個体を作る手法が、遺伝的アルゴリズムである。

3.2 個体の選択方法

遺伝的アルゴリズムにおいて、新しい子供を生み出すときの原則は“適合度の高いものほど多産にすること”である。個体の数を N 個とすると、選択はこの N 個の個体の集合から重複を許して N 個の個体を選ぶことになる。この選択方法には、ルーレット方式、トーナメント方式などがある。

(1) ルーレット方式

適合度に比例した割合で選択する方式であり、これは適合度に比例した領域を持つルーレットを作成し、そのルーレットを回し的中した領域の個体を選択する方法である。

(2) トーナメント方式

この方法は、個体の集団の中からある数の個体をランダムに選び出し、その中で最も適合度の高いものを選択する方法である。

3.3 交叉方法

現在までに以下のような方法が提案されている。

(1) 1 点交叉

ある遺伝子座（交叉点）を境にそれぞれの親から遺伝子を受け継ぐ方法である。

(2) 複数点（ n 点）交叉

交叉点の間で交互に片方の親から遺伝子を受け継ぐ方法である。 $n = 1$ のときは 1 点交叉

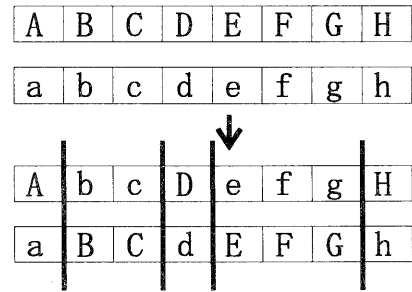


図 3: 一様交叉の例

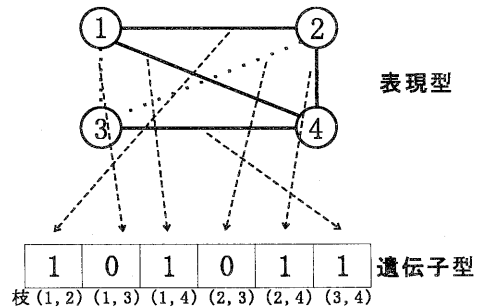


図 4: 遺伝子の設計

である。また $n = 2$ の 2 点交叉がしばしば用いられている。

(3) 一様交叉

任意個の交叉点が取れる方法である (3)。

一般に、これらの交叉方法の探索能力は、

$$1 \text{ 点交叉} < 2 \text{ 点交叉} < \text{一様交叉}$$

であるとされている。

4 提案する解法

4.1 本研究で扱うグラフ

本研究で扱うグラフは、任意のコストを表す重み付きグラフである。また当初与えられる部分グラフの連結度も任意の連結度とする。

4.2 遺伝子型の設計

遺伝子情報は、遺伝子長を $\frac{1}{2}|V|(|V|-1)$ とし、各遺伝子座には枝の有無を図 4 のようにバイナリ形式で表現する（有：1，無：0）。実際には交叉の結果によって求める枝連結度を満たさないグラフも生じる。本研究ではそのような遺伝子は致死遺伝子として扱った。ここで、各個体の適合度

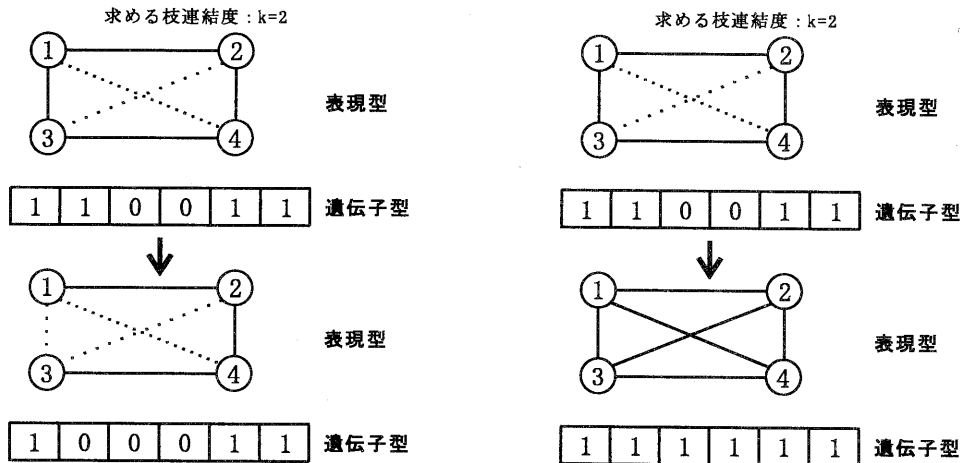


図 5: ランダムに行う突然変異

は、個体の枝連結度が求める枝連結度未満であれば 0 とした。また、求める枝連結度を満たしている場合は、世代内の個体の重みの最大値から各個体の重みを引いたものとした。

4.3 アルゴリズム

このような遺伝子型をもとに、次のようなアルゴリズムを構築した。

- step1 求める枝連結度を k とする。 $gen = 0$ とする。
- step2 個体を N 個用意し、ランダムにグラフに枝を加え、それを遺伝子情報で表現する。
- step3 それぞれのグラフの連結度 c_i 、重み w_i を計算し、その結果をもとに適合度 h_i の計算を行う。世代内の個体が 80% 以上等しくなれば終了。

$$h_i = \begin{cases} (\max_{j \in N} w_j) - w_i, & c_i \geq k \\ 0, & c_i < k \end{cases}$$

- step4 次世代の親をルーレット法により N 個選択する。
- step5 一様交叉を行い、次世代の DNA を N 個作成する。また突然変異も行わせる。
- step6 $gen := gen + 1$ とする。 $gen = 1000$ ならば終了、そうでなければ step3 へ。

4.4 アルゴリズムの改良

前述のアルゴリズムでは、この問題の特徴である連結度に対しての考慮が全くなされていないので、探索が効率的でなく局所解に陥りやすいと考える。したがって、局所解から脱するために突然変異を、また探索の効率化を図るために交叉を改良した。

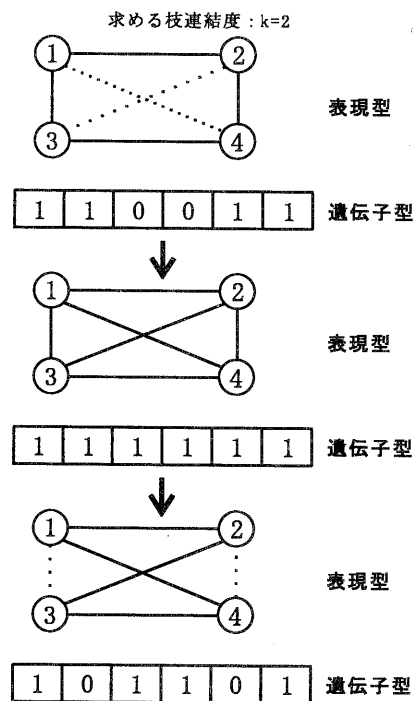


図 6: 連結度を考慮した突然変異

4.4.1 突然変異

本研究では、突然変異は世代内の個体の適合度が最も高いものに強制的に行い、局所解から脱することを図った。改良前の突然変異では、図 5 のようにランダムに遺伝子座の変異を行い、連結性を無視しているために、個体の連結度が求める連結度を満たさない可能性がある。それにより、個体が致死遺伝子になり、次世代以降に影響が小さくなると考えられる。そこで選択した親 1 に対して、以下のように突然変異を行わせた。

- step4' 適合度の最大のものを 1 つ選び、以下の手順で突然変異を行い、次世代の親として 2 つ作成する。
 - (1) n 個の枝のない遺伝子座に 1 を立てる。(ただし、 $0 \leq n \leq |E|$)
 - (2) グラフの連結度が k を下回らないようにランダムに枝を n 個まで除去する。
 さらに次世代の親をルーレット法により $N - 2$ 個選択する。

4.4.2 交叉

前述のアルゴリズムでは、交叉はランダムに行わせているが、その交叉により優良な個体が淘汰される可能性がある。特に連結度が k を下回った

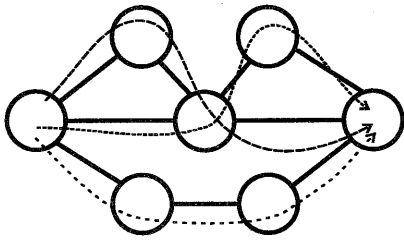


図 7: 独立した3つの経路

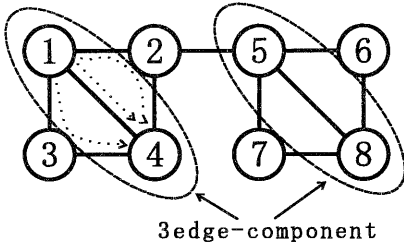


図 8: 3-コンポーネントの例

場合、適合度は0となる。それを回避するために k -コンポーネントを用いて交叉の確率を変動させた。 k -コンポーネントは、以下のような点の部分集合 $S \subseteq V$ である。

- (1) 任意の点 $u, v (u, v \in S)$ において、 u, v 間の独立した道の数が k 以上である。
- (2) 任意の点 $w (w \in V - S)$ において、 w, u_w 間の独立した道の数が k 未満となる $u_w (u_w \in S)$ が存在する。

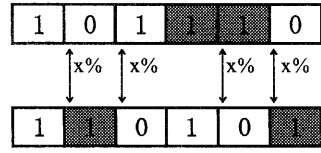
ここで独立した道の数とは、2点間の枝を共有しない道の数である(点は共有してもよい)。同

じ $k+1$ -コンポーネントに含まれる点は、少なくとも $k+1$ 連結である点の部分集合である。すなわち、 $k+1$ コンポーネントに含まれる枝を1本除去しても求める連結度 k を下回ることはないが、それ以外の枝を除去すると k を満たさなくなる。そのことから、同じ $k+1$ -コンポーネントに含まれる2点を直接結ぶ枝があるならば、その枝が交叉によって失われる確率を高く設定することにより、探索の効率化を図れると考えた。そこで、 $k+1$ -コンポーネントに含まれる枝が交叉によって失われる確率を x 、 $k+1$ -コンポーネントに含まれない枝が交叉によって失われる確率を y とした。(ただし、 $x > y$)

改良した step5 を以下に示す。

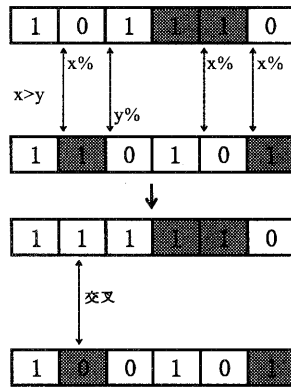
step5' 次のルールによって、次世代のDNAを N 個作成する。($x > y$)

- (1) $k+1$ -コンポーネントに含まれる枝の遺



■ コンポーネントに含まれる枝 (遺伝子座)
□ コンポーネントに含まれない枝 (遺伝子座) もしくは枝のない遺伝子座

図 9: コンポーネントを考慮していない交叉



■ $k+1$ -コンポーネントに含まれる枝 (遺伝子座)
□ $k+1$ -コンポーネントに含まれない枝 (遺伝子座) もしくは枝のない遺伝子座

図 10: コンポーネントを考慮した交叉

伝子座と枝のない遺伝子座との交叉確率を x とする。

- (2) $k+1$ -コンポーネントに含まれない枝の遺伝子座と枝のない遺伝子座との交叉確率を y とする。

5 実行結果

グラフの枝増加問題について、ここで提案した遺伝的アルゴリズムによって解が得られることを確かめるために、いくつかの異なったアルゴリズムによるプログラムを作成し、実行した。使用したグラフは、10点からなる重み付き完全グラフで、当初与えられたグラフは2枝連結グラフとした。また、目標の連結度は3とした。その実験の結果を図11から図14に示す。ここで各パラメータは $n=3, x=0.9, y=0.5$ とし、個体数は50とした。重みが少ない方、すなわち横軸の左方が良好な解が与えられていることを示している。

図11は、代表的な遺伝的アルゴリズムをほとん

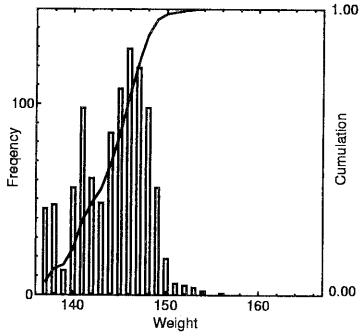


図 11: 通常の遺伝的アルゴリズムによる結果

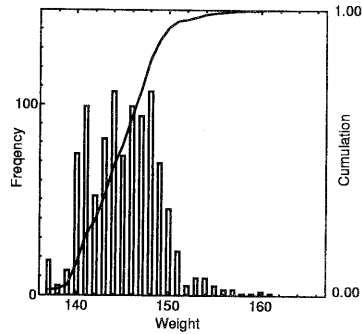


図 13: 交叉確率・突然変異の改良による結果

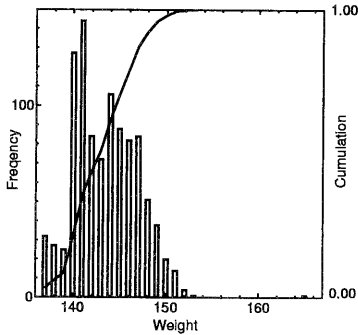


図 12: 交叉確率の改良のみによる結果

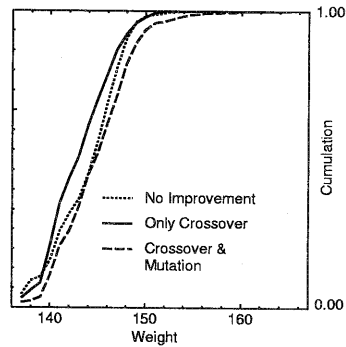


図 14: 実行結果の比較

ど改良することなしに実施した結果である。すなわち個体を選択したあと遺伝子座の一樣交叉確率をすべて0.5とし、突然変異も全く一樣にランダムに行ったものである。

図 12 は、交叉確率について改良を試みた結果である。ここで $k+1$ コンポーネントに含まれる枝の遺伝子座と枝のない遺伝子座との交叉確率を0.9とし、 $k+1$ コンポーネントに含まれない枝の遺伝子座と枝のない遺伝子座との交叉確率を0.5として行ったものである。最適解が図 11 にくらべて少なくなっているが、解の分布は最適解により近似している。

図 13 は図 12 の改良を行った上で、さらに図 6 の突然変異を加えて実行した結果である。

以上の実行結果を比較したものが図 14 である。この結果、突然変異と交叉確率に改良を加えたものは有為な結果が得られず、交叉確率だけを改良した結果は、精度の良い解が得られた。以上のことから交叉の際に連結度を考慮することで得られる解が改善されることがわかる。

6 むすび

本研究では、伝送経路を多重化して故障対策の課題を扱うため、グラフ理論の k 枝増加問題を取

り上げた。ここで枝増加問題の解法にあたって、新たに遺伝的アルゴリズムによる近似解法を提案した。さらに、遺伝的アルゴリズムに関して連結度を考慮した改良を行い、その改良の成果を明らかにするためにいくつかのグラフについて実験を行った。改良前と改良後のアルゴリズムの実行結果から、改良後の結果に解の改善がみられた。

今後の課題として、突然変異のアルゴリズムの改善、更なるアルゴリズムの改良と実際に存在するネットワークの障害対策のため速やかに解を与えるためのシステム開発があげられる。

参考文献

- [1] S. Even *Graph Algorithms*. Comput. Sci., Potomac, MD, 1979.
- [2] K. P. Wswaran and R. E. Tarjan "Augmentation Problems", *SIAMJ. Comput.*, 5:653-655, 1976.
- [3] T. Watanabe and A. Nakamura "Edge-Connectivity Augmentation Problems", *J. Comput. System Sci.*, 35:96-144, 1987.

- [4] G. N. Frederickson and J. Ja'ja "Approximation Algorithms for Several Graph Augmentation Problems", *SIAMJ. Comput.*, 10:270-283, 1981.
- [5] S. Ueno, Y. Kajitani, and H. Wada "Minimum Augmentation of Trees to a k -edge-connected graph", *Networks*, 18:19-25, 1988.
- [6] T. Watanabe and A. Nakamura "k-Edge-Connectivity Augmentation Problems", *Paper of Technical Group, IEICEJ*, AL83-90, 1984.
- [7] T. Watanabe and A. Nakamura "On Smallest Augmentation to k -edge-connect a graph", Technical Report C-20, Hiroshima Univ., 1984.
- [8] 伊庭斉志 "遺伝的アルゴリズムの基礎-GAの謎を解く-", オーム社, 東京都千代田区神田錦町, 1994.