

遺伝的アルゴリズムの分散制御型動的ルーティングへの応用

日高東潮

高井昌彰

佐藤義治

北海道大学 工学部

近年遺伝的アルゴリズム (以下 GA) の研究が進み、さまざまなモデル、またその応用が多数報告されている。しかし、その多くは最適化の対象となる解空間が固定した形式の問題であり、変化が伴うような問題への拡張はあまり見られていない。そのような問題の一つとして、パケットの転送負荷に対して動的に変化するネットワークルーティング問題が挙げられる。本稿ではネットワークに分散制御型であるという条件を加えた上で、その問題への GA の適用について考察する。はじめに、環境変化に対する追従性に優れた GA の枠組を定式化し、その上でネットワークにおけるルーティング問題を取り扱う。

Distributed Dynamic Routing with Genetic Algorithms

Toshio Hitaka, Yoshiaki Takai and Yoshiharu Sato

Faculty of Engineering, Hokkaido University, Japan

Recently various genetic algorithms (GA) have been proposed to be an effective approach to solving complex combinatorial optimization problems. In this paper, we deal with a dynamic network routing problem which is a kind of time-variant optimization problems. We first formulate yet another genetic algorithm model (CCGA : Controlled Crossover GA) which can quickly adjust to a time-variant solution space. Then we apply the proposed model to a network routing problem to minimize the total communication delay associated with a network, and we discuss some issues to implement a distributed control mechanism for the dynamic network routing.

1 はじめに

近年近似最適化手法の一つである遺伝的アルゴリズム (GA) は TSP (巡回セールスマン問題)、ジョブショップスケジューリング問題などの組み合わせ最適化問題などを対象に数多く研究がなされ、モデルが多数提唱されている。NP-完全問題など、解空間が膨大な問題や、発見的な探索手法が知られていない問題において、効果があることが実験的に確かめられている [1][2]。

しかし、枝刈り法などの発見的最適化手法と比較した場合、解空間の性質がほとんど分かっていないような問題を除けば、多くの場合発見的手法の方が探索時間が少ない。そのようなことを考慮すると、GA の探索効率が必ずしも高いとは一概に言えないとも考えられる。GA の性能を十分引き出すには、より GA の特徴をとらえ、それをうまく使う必要がある。

ここでは GA の特徴として、「解空間の情報を遺伝子の形で蓄積できる」ということを強調しておきたい。これは他の最適化手法には無い特徴である。さらにこれは「適合度関数値が時系列に対して (緩やかな) 変化をする時に適応が速い」ということを意味し、制御システムなどに応用した場合、他の手法よりもより良い制御を与えられると考えられる。

本論文では制御システムの中でも、ネットワークのパケットルーティング問題を取り上げる。近年 UNIX ネットワークは世界的規模に拡大しているが、ネットワーク内部でのパケットルーティングは現在のところ静的なルールによるものがほとんどである。しかし、静的なルーティングのままネットワーク規模を拡大していくと、ネットワーク負荷の変動に対処できず、パケット配送がスムーズにいかなくなるのが容易に推測される [3]。

ネットワークルーティング問題に GA を適用するにあたり、

1. GA の環境変化に対する追従性の向上

2. GA をネットワークルーティングに用いるための分散制御の枠組

を十分考える必要がある。本論文では、解空間の無駄を省き、環境変化に対する追従性を高めた GA である CCGA (Controlled Crossover GA) を提案し、その枠組を定式化する。さらに、本手法をネットワークの分散制御型動的ルーティングに適用することを試み、その可能性と問題点を考察する。

1.1 CCGA の基本的なアプローチ

最適化の対象となる問題の解空間を遺伝子集団にマッピングし、それに対して適合度関数値により定まる遺伝子間の優性劣性にに基づいた淘汰 (selection)、交叉 (crossover)、突然変異 (mutation) といった生物進化における 3 基本操作を繰り返し行なうことにより適合度関数に対する最大化問題を解くというのが遺伝的アルゴリズムのコンセプトである。一般的に遺伝子集団内での遺伝子の優性劣性は、集団内に占める遺伝子の割合で表現されている。

しかし、単純に解空間の探索という観点から見た場合、遺伝子集団内に同じ遺伝子が現れるのは有限の遺伝子数で探索を行なう GA の構成上、計算効率が悪いと思われる。CCGA ではこの点に注目し、通常の GA と同等の計算時間で極力遺伝子重複を防ぐことにより、解空間探索効率の向上を目指そうというものである。

また CCGA では、何らかの形で解空間情報を遺伝子操作に対して付加することを考える。ただし、問題に対して強い条件を加えると GA の特徴である一般性が無くなるので、ここでは遺伝子空間に何らかの距離を定義し、その距離に対して適合度関数値が滑らかに変化するという非常に弱い条件付けのみを行ない、それに対して有効な解空間探索が行なわれるように GA の各操作を考え、解探索の速さ、正確さを向上させている。

1.2 ルーティング問題への GA の適用可能性について

ルーティングにあたり、静的なネットワークのトポロジー情報を基にして、どの近傍ノードにパケットを転送するかを決めることも出来る。しかしネットワーク中に、同時に多数のパケットが混在している条件では、組合せ問題のようにトータルでの評価をしなければならない。ここでは、ネットワーク ルーティング問題の解決に動的ルーティング手法を用いる理由、及びその中で GA を用いる理由を述べる。

例として、6 個のノードから成る Figure.1 のようなネットワークを考える。

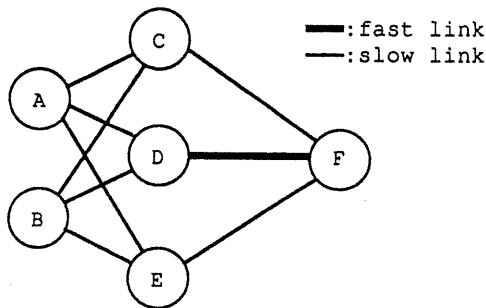


Figure.1: ネットワーク例

ここでノード A からノード F へパケットを転送する時、近傍ノード D を経由する方が速く、またノード B からノード F へパケットを転送する時も、近傍ノード D を経由する方が速いとする。このような場合、静的にパケット転送割り付けを行なうとノード D からノード F までのリンクにパケットが集中することになる。静的な手法では、ネットワークに高速なリンクがあれば、ルーティングにそのリンクを加えようとするために、高速なリンクへパケットが一斉に集中してしまう可能性がある。これは、ネットワークの規模が大きくなるほど集中の度合が増すので、無視できない問題である。このような状況を避けるためには局所的な評価だけでなく、大域的な評価が必要であり、動的な変化にも対

応できるアルゴリズムが有効である。

ルーティング問題に GA を用いるのは、「遺伝子によって解空間の情報を蓄積出来ることにより、状況変化に対しても、比較的安定した解を提供できる」という理由による。ネットワークの負荷の状況は刻一刻と変化するため、発見的手法より、変化に対する追従の点で優ることが期待される。

2 CCGA

2.1 モデルの概要

GA を構成する上で考えなければならない最も重要なことは、次の 2 点である。

1. すでに存在する遺伝子から新たに遺伝子を生成する手法
2. 遺伝子集団に対する淘汰の手法

この点を踏まえて、ここでは最初に解空間が数直線上にあるような関数の最適化における、遺伝子生成手法を考える。

CCGA は、数直線上の適当な 2 点を選び、その中点の適合度関数値と前の 2 点の適合度関数値を比較し、その値の大きな遺伝子 2 つを次の 2 点とする再帰的分割に基づいている [5]。

解空間を $[0, 1]$ の実数数直線に見立て、最初の 2 点を 0, 1 とすると、そこから生成される解は Figure.2 に示すようになる。ただし、position は数直線上の位置、class は何度目の操作によって生成された遺伝子であることを表わしている。

class	position
0	n ($n = 0, 1$)
1	$n \cdot \frac{1}{2}$ ($n = 1$)
$\vdots$	$\vdots$
t	$n \cdot \frac{1}{2^t}$ ($n = 1, 3, \dots, 2^t - 1$)

Figure.2: 遺伝子の class 分け

前述した2点から中点を作り出す手続きをGAの遺伝子集団に対する初期化とcrossoverに適用する。この際、classが異なる遺伝子同士間で遺伝子生成を行なうことにより、各遺伝子のclassより大きなclassの遺伝子が生成されることが分かっており、このことを利用して、重複無しで解空間を探索することができる。

次にCCGAにおける処理の流れについて述べる。最初にランダムに遺伝子一つを生成し(この遺伝子をseedと呼ぶ)、seedからの距離が最大の遺伝子とでclass0を形成する。その後class0の遺伝子間で前述した遺伝子生成手法によりclass1遺伝子を生成する。ここまで得られた4遺伝子に対して、再帰的に前述の遺伝子生成手法の適用を、遺伝子総数がある与えられたpopulation数に達するまで行なう。この時点で初期遺伝子集団の生成が終了する。

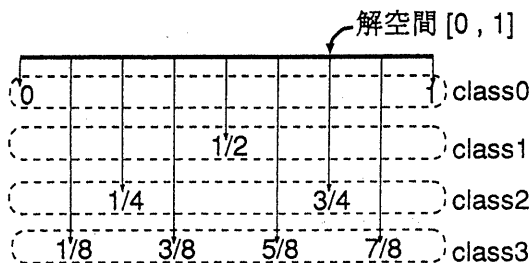


Figure.3: 遺伝子の解空間内に占める配置

その後、遺伝子が解空間中の最大classへ到達するか、もしくは何らかの収束条件を満たすまで、遺伝子集団に対してselectionとcrossoverを繰り返す。selectionの方法については、遺伝子の重複を出さないという条件以外に制約は特に無い。また、遺伝子の重複を許さないという観点からmutation操作は行なわない。

ここで、seed生成から世代を繰り返し、重複が出る直前までの一連の操作をcycleと呼ぶ。seedを変更しつつcycleを複数回実行し、各cycleから得られた近似最適解の内、最大の適応度関数値を最終的に得られる解とする。CCGAの一連

の流れをFigure.4に示す。

```
repeat
  make 'SEED'
  generate 'GENES' from 'SEED'
  while not MAX class
    selection
    crossover
  end while
until stopcriterion
```

1 cycle

Figure.4: CCGA

2.2 環境変化に対する追従性

実際に多峰性関数 $f(x; t) = \sin(x) \cdot \sin(20x + t)$ を最大化すべき目的関数として用意し、これを時間とともに変化させた時のCCGAによって得られる目的関数の挙動を従来のSimpleGA(SGA)と比較してみた。 x は変数、 t は一定世代数毎に変動するパラメータである。実験の条件は以下の通りである。

• 共通条件

- Population = 30
- Gene length = 64
- crossover時には親遺伝子の削除無し。
- t を50世代(実験(a))、100世代(実験(b))毎に0.1ずつ増加させる。

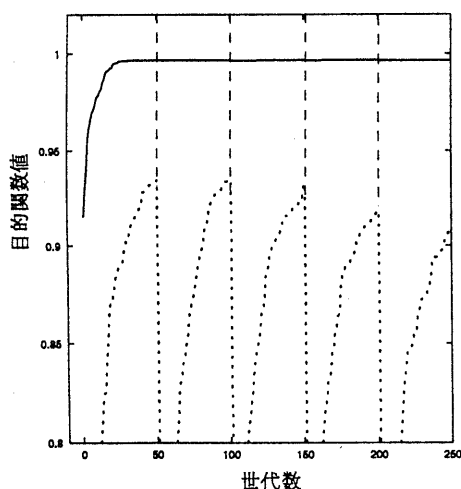
• SGAの条件

- crossover rate = 0.6
- mutation probability = 0.02
- selectionにroulette wheel selectionを用いる。

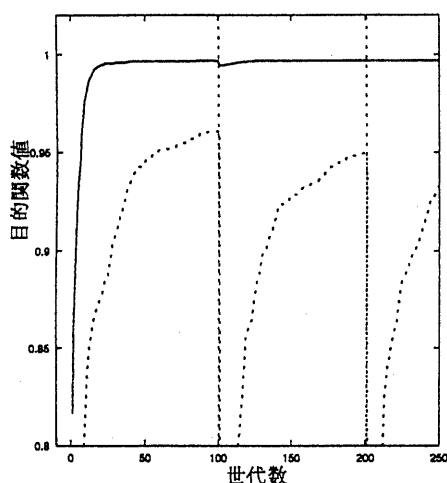
• CCGAの条件

- selectionにranking selectionを用いる。
- 1 cycleは64世代以下

Figure.5に結果を示す。ここで目的関数値は100回の平均値である。



(a) 50世代毎に変化した場合



(b) 100世代毎に変化した場合

Figure.5：時間とともに変化する目的関数に対する追従性の比較（実線 CCGA、破線 SGA）

(a),(b) いずれのグラフでも、目的関数に変化する際、その時点で得られていた最良の遺伝子の fitness が悪くなるので、目的関数値が一時的に落ち込んでいる。しかし、CCGA は、解空間情報を SGA に対してより多く確保できているために、落ち込み方も非常に少なく、また良い解への収束も速い。これより CCGA は変化に対する追従性が SGA に比べて高く、また目的関数の多峰性についても問題にならないことが分

かる [6]。

3 ネットワークルーティングへの応用

3.1 ネットワークの条件

ネットワークに対して以下のような条件を与えることにする。

- 分散型の制御を想定し、ルーティングの集中制御は行なわない。
- ネットワーク負荷変動に対して動的に対応する。
- 各ノードは原則として自身に直結したリンク上のパケットの流通量しか観測できない。
- ノードやリンクの故障・新設もネットワーク負荷の変動の一部として扱う。
- 1パケットあたりの平均転送時間を最小にすることを目的とする。

任意のネットワークに対して、グラフを整理することにより、Figure.6 に示すように自己ノード (node A)、近傍ノード (node $B_0 \cdots B_n$)、およびパケット送出先である目的ノード (node C) の位置関係を表すことができる。

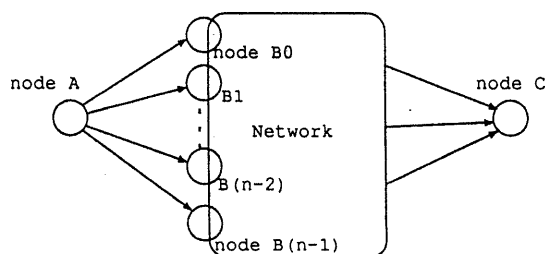


Figure.6：近傍ノードと目的ノード

ノードは送られて来たパケットに対して、どういったルーティングで目的ノードまでパケット

を転送するかを決めるのではなく、単に自身の近傍ノードのうち、目的ノードへ転送することを考えた場合の最適と思われる近傍ノードを選び、そこへパケットを転送することのみ行なう。

3.2 分散制御型動的ルーティング

ルーティングの目的は、パケットが目的ノードに達するまでの転送時間の平均を全体に渡って最小化することである。そこでパケットに、い

つ、どのノードを通過して来たかを示すタグ (通過ノード情報) を付加し、受信ノードでこの情報を獲得し、これを近傍ノードから最適な1つを選択する判断材料とする。

すなわち、ある最終目的ノードに達するのに近傍ノードの内からどれを選択すべきかを示すテーブル (ルーティングテーブル) を各ノードに用意し、ここに受信パケットから抽出した通過ノード情報をフィードバックする。

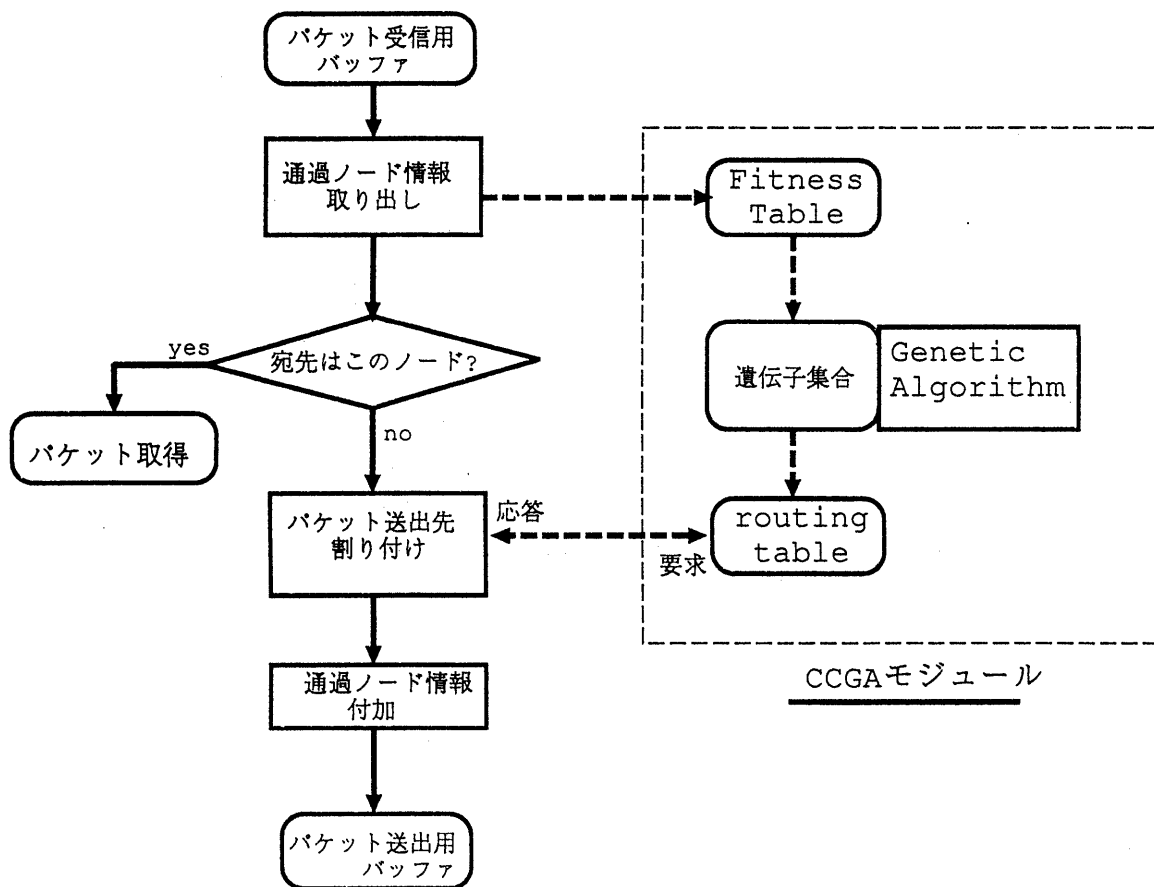


Figure.8 : ノード内のルーティング処理の流れ

ルーティングテーブルの例を Figure.7 に示す。
 ここで、ルーティングを 1 つの遺伝子として
 コーディングし、その適合度を通過ノード情報
 に基づいて決定する。

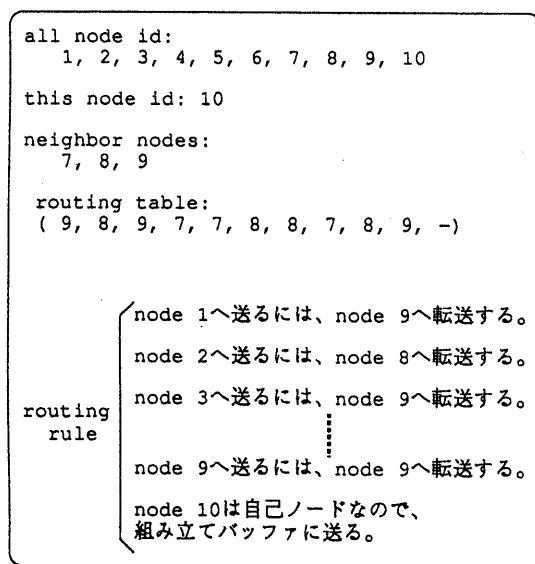


Figure.7 :ルーティングテーブル例

各ノード内のルーティング処理の流れを Figure.8 に示す。パケットに対する操作の概略は以下になる。

1. 受信バッファからパケットを取り出す。
2. パケットから通過ノード情報を引き出し、CCGA モジュールへ送る。
3. パケットの送信先をチェック。現在パケットが存在するノードが送信先である場合は、パケット組み立て用のバッファに送る。
4. CCGA モジュール部が生成したルーティングテーブルに従い、どの近傍ノードにパケットを送信するかを決定する。
5. ノードの ID と、ノードから送出される時刻をパケットの通過ノード情報に付加する。

6. 送信バッファに送り込む。

パケットは、転送に必要な以下の情報を有する。

- 目的ノード ID
- 転送データ本体
- パケットを組み立てるための情報
- 通過ノード情報

ここで通過ノード情報は、Figure.9 のような構成で、ノードを通過していく度に情報が付加されていく。

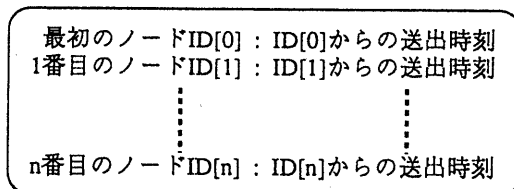


Figure.9 :通過ノード情報

リンクが故障した場合、そのノードに接続されているリンク上のパケット流通量がゼロになるため、そのリンクに対する遅延 c 評価が下がることにより、ルーティングからそのリンクが外されることで、自然に対応する。ノードが故障した場合は、そのノードに接続されているリンクがすべて故障した状況と同じ状態となる。従って、リンクが故障した場合と同じ理由により、故障したノードが接続されているリンクを含めてルーティングから外される。

リンクが新設された場合は、その両端に接続されているノードのリンク情報に対して、新設されたリンクの情報を付加することで解決する。

一方、ノードが新たに増設された場合は、事前にすべてのノードに対して新設ノードに関する情報を与えておかないと、パケットが迷走する。そのため、以下の手順を踏むことにより対処する。

- 新設ノードに接続される近傍ノードの遺伝子情報に新設ノード情報の付加を要求する。

- 全ノードは新設ノード ID と、その近傍ノード ID のアナウンスを受け、新設ノード分遺伝子を拡張する。
- 新設ノードに対するパケット転送要求が発生した際には、新設ノードの近傍ノード中最も伝送遅延が低いノードへの情報を用いて転送を行なう。

```
gene 0 : (1,3,3,2,5,1,3,2,4,5)
gene 1 : (2,4,5,1,1,3,5,2,4,5)
crossover mask : (0,1,1,1,0,1,0,0,0,1)
```

```
gene 0' : (1,4,5,1,5,3,3,2,4,5)
gene 1' : (2,3,3,2,1,1,5,2,4,5)
```

Figure.10 : crossover 例

3.3 CCGA モジュール

3.3.1 遺伝子のコーディング

あるノード j に存在するルーティングテーブルは、ノード j から目的ノード i へパケットを送るためのノード j の近傍ノード ε_i を、目的ノードの数だけ並べたもの $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ である。これを CCGA の遺伝子とする。ここで、各近傍ノード ε_i について、ノード i から ε_i を経由してノード j に到着したパケットの転送時間 d_i を割り当てることが出来る。これは、受信パケットから通過ノード情報を抽出することで可能である。これによって、遺伝子 $(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ の適合度を $\sum_{i=1}^n d_i$ の逆数で定める。

3.3.2 selection

条件としては遺伝子重複を出さないということのみである。ここでは ranking-selection を用いることとする。

3.3.3 crossover

ここでは、uniform-crossover[4] により 0,1-mask の 0 と 1 の個数が同じになるようにして中点を構成する。

親遺伝子 $gene0, gene1$ から、 $gene0', gene1'$ を mask により生成する具体例を Figure.10 に示す。近傍ノード数は 5、自身を除くすべての目的ノード数は 10 である。

4 考察

前章で述べたルーティングアルゴリズムで最短経路が構築されるかどうかを考察する。

まず、Figure.6 に示すネットワークで、ノード A から目的ノード C への最短経路を考える。ここで、経路 $B_i - C$ ($i = 0, 1, \dots, (n-1)$) はそれぞれ十分探索が行なわれており、最短経路が確保されていると仮定すれば、 $B_i - C$ ($i = 0, 1, 2, \dots, (n-1)$) に、リンク $A - B_i$ ($i = 0, 1, 2, \dots, (n-1)$) の情報を合わせ、その中での最短路を選ぶことにより、この A, C 間の最短路を確保できる。この際、ルーティングの中にループ構造が無いことが必須条件となる。この条件を達成するため、パケットの持つルーティング情報を各ノードがチェックし、ループ構造が出来るようなルーティングをしている部分を見つけた場合、ノード内の遺伝子集団に対して、そのルーティングの評価値を下げることで解決できる。

しかし、巨大なネットワークになると、転送時間情報を基にルーティングを行なっても、パケット転送時にはすでに情報が古いことになり、決定されたルーティングがもはや高速な経路でなくなっていることが考えられる。パケットにどのようなルーティングを行なうかを最初にすべて与えるような構成では、その問題が無視できない。しかし本論文で述べた手法では、通過するノード毎に近傍ノードを 1 つ選択する構成となっているため、伝送遅延による情報の歪みの影響は少ないと考えられる。

現在、シミュレータにより、小規模ネットワークの簡単な評価を行ない、パケットルーティングが問題なく行なわれ、迷走するパケットが出るようなことが無いことを確認している。

シミュレーションでは、システムが始動した時点でノード通過情報が無いので、ネットワークのトポロジー情報から最低限のルーティングを一つ与えている。ネットワークが小さければ、十分各ノードにパケットが流通し、そのためネットワーク全域の情報が得られるが、ネットワーク規模が大きくなれば、末端のノードになるほど全部のリンクの情報が行き渡らないノードが出ることも考えられる。このような情報の足りないノードは自発的に他のノードから情報を得ることの出来るような仕組みが必要となる。

5 まとめ

本研究では、環境変化に対する追従性に優れた CCGA を提案し、その有効性を時間的に変化する多峰性関数の最大化を例にとり実験的に示した。また、CCGA を用いたネットワークの動的ルーティング手法を提案した。

今回提案したルーティングの特徴は、ネットワークの観測を局所のみで行ない、それをもとに大域的な最短経路を形成するという点にある。これにより集中制御による複雑な構成を取ることなく、ネットワークを円滑に動かすことが出来る。

しかし、ノードの新設に関してはネットワーク全域に対してアナウンスを行なう必要があり、今後この点を解消して完全な分散制御を実現したいと考えている。

今後の課題としては、ノード数が多い場合のシミュレーションを行ない、遅延が大きくなった場合でも安定してルーティングが行なえることを実験により確認すること、並びに、CCGA で用いている“中点”の厳密な定義を行ない、さ

まざまな問題に対して CCGA が使えるよう一般化することが挙げられる。

また、今回のルーティングの枠組は、他の発見的手法などでも応用することができる。従って、他の手法を用いた場合との比較実験を行ない、CCGA の有効性を確認する予定である。

参考文献

- [1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison - Wesley (1989).
- [2] K. A. DeJong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems* (PhD Thesis), University of Michigan (1975).
- [3] Vijay Ahuja 著, 池田信一 訳, “コンピュータ通信ネットワーク”, 近代科学社 (1982).
- [4] Gilbert Syswerda, “Uniform Crossover in Genetic Algorithms”, *Proc. of ICGA '89*, pp.2-9 (1989).
- [5] 日高東潮, 高井昌彰, 佐藤義治, “遺伝的アルゴリズムの探索効率化に関する考察”, 第 46 回情報処理学会全国大会, 2-303 (1993).
- [6] 日高東潮, 高井昌彰, 佐藤義治, “緩やかに変化する目的関数への GA の適応性について”, 電子情報通信学会秋季大会, D-131 (1993).