

# 遺伝的アルゴリズムを用いた帯域幅割当における 分散アルゴリズムの設計 － ノード障害への対応 －

小林英博<sup>1</sup>, 棟朝雅晴<sup>2</sup>, 赤間清<sup>3</sup>, 佐藤義治<sup>4</sup>

1,4:北海道大学大学院 工学研究科 数理情報工学講座 情報解析学分野

2,3:北海道大学 情報メディア教育研究総合センター 情報メディアシステム分野

大規模ネットワークにおいて、限られたネットワーク資源を効率的に利用するためには通信帯域幅を適切に割当てることが重要である。特に高速な通信リンクは高価なため、割当ての効率化により得られる利益が大きいからである。これまで、遺伝的アルゴリズムを用いたネットワーク帯域幅割当のためのアルゴリズムとして GRA (Genetic Routing Algorithm) が提案されているが、基本的には集中型のアルゴリズムであるため、ネットワークの障害に弱く、障害が発生すると割当て自体を行うことができなくなる可能性がある。そこで、本論文ではネットワーク障害への対応を考え、GRA を分散化した D-GRA (Distributed GRA) を提案し、その障害回復機能の有効性を検証する。

## Designing a Distributed Algorithm Using Genetic Algorithms for Bandwidth Allocation － Recovery Mechanism from Node Failure －

Hidehiro Kobayashi<sup>1</sup>, Masaharu Munetomo<sup>2</sup>, Kiyoshi Akama<sup>3</sup> and Yoshiharu Sato<sup>4</sup>

1,4:Information and Data Analysis, Graduate School of Engineering,  
Hokkaido University

2,3:Center for Information and Multimedia Studies, Hokkaido University

In huge communication networks, proper allocation of bandwidths is essential to achieving effective utilization of limited network resources. High capacity communication links are costly and proper allocation of bandwidth for such links saves us more investment. To design bandwidth allocation algorithms, an algorithm called GRA (Genetic Routing Algorithm) was proposed elsewhere that employs genetic algorithm to optimize allocations. The GRA is a centralized algorithm which is considered vulnerable to network failures: it may not finish bandwidth allocation on node or link failures. In this paper, we propose a distributed algorithm for the GRA, called D-GRA (Distributed GRA) which enables us to recover from network failures. Through empirical studies, we show effectiveness of recovery mechanism.

### 1 はじめに

現在のインターネットの普及はめざましく、ネットワークを流れるトラフィック量も急激に増大している。さらにネットワークの高速化・広帯域化によりこれまでは困難であった Video-on-Demand や遠隔会議といった新たなサービスが要求されている。ところがネットワーク資源は有限であるた

め、これらのサービスを満たすためには要求帯域幅を確保するための帯域幅割当アルゴリズムが必要となる。帯域幅割当とは、目的ノードまでの経路中に要求帯域幅を確保することであり、これは割り当てる帯域幅と経路との組合せ問題とみなすことができる。この問題を効率良く解決するため遺伝的アルゴリズム GA (Genetic Algorithm) を適用した研究が行なわれている [2, 3]。これらの研究

では一つのノードが全ノードの通信要求を把握しネットワーク全体の平均遅延を最小にする集中型のアルゴリズムが提案されている。このアルゴリズムの問題点は、帯域幅割当の計算を集中型で実行するとネットワークに障害が発生した場合、割り当てが不可能となる可能性が考えられることである。そこで本研究では帯域幅割当アルゴリズムを分散化し、ノード障害へ対応した分散アルゴリズムを設計する。具体的には割当計算をノードごとに独立して実行することでノード障害に対応し、求めた割当に平均遅延による適応度を定め、適応度を他ノードと交換することでネットワーク全体の最適化を実現する。

## 2 遺伝的アルゴリズムによるネットワーク帯域幅割当

Mario Gerla らによる ATM ネットワークにおける帯域幅割当の研究 [1] では、帯域幅割当を平均パケット遅延の最小化を目的としたネットワーク最適化問題として公式化している。ATM では物理リンクを多重化した Virtual Path (VP) により通信が行なわれる。VP は途中の ATM スイッチをトンネルした論理的なパスであり、ATM ネットワーク上では VP を単位としたルーティングが可能である [6]。

帯域幅割当とは、要求された帯域幅を VP に割り当てることであるが、[1] では平均パケット遅延が最小となるように帯域幅と VP を組み合わせる手法を提案している。ここで平均パケット遅延  $T$  は待ち行列理論により、

$$T = \frac{1}{\lambda} \sum_{m=1}^M \frac{f_m}{C_m - f_m} \quad (1)$$

( $C_m$  が  $m$  番目の VP の容量、 $f_m$  が  $m$  番目の VP を流れる通信量、 $M$  が VP の総数、 $\lambda$  が平均パケット到着率、 $M$  が VP の総数) と表すことができる。

この手法では数理的計算により最適解を求めているが、ここで問題となるのはノード数の増加により最適解の探索に多くの計算量が必要となることである。この問題点を改良したものが Hong Pan らによる研究 [2] であり、最適解の探索に GA を用いて効率的な探索を行なう手法が実現されている。

上記二つの研究では平均パケット遅延の最小化を行なっているが、通信の種類によってはパケット損失確率やジッタなどを同時に最適化することが望まれる。そこで小林らにより複数の値を同時に最適化するための多目的最適化の枠組が提案されている [4]。この研究では GA を使用した最適解の探索を進展させ、各目的関数の解に対してパレート最適の概念を導入する。パレート最適の概念により複数の目的関数間でのトレードオフを決定することが可能となる。この手法により、複数の目的関数に対応した帯域幅割当を実現している。

一方、Theeb らにより動的ルーティングを行なうことで帯域幅割当を実現するための GRA (Genetic Routing Algorithm) が提案されている [3]。GRA は各通信要求に対して、それぞれの起点ノード (S) から目的ノード (D) へのルーティングテーブル中から最適な経路を選択し、ネットワーク全体の平均パケット遅延を最小化するアルゴリズムである。

S-D	0-1	0-2	...	a-b	...
Routing Table No.	$x_{01}$	$x_{02}$	...	$x_{ij}$	...

表 1: 経路の選択

例えば表 1 において  $x_{ij}$  が各 S-D 間のルーティングテーブルから選択された経路であるとすると

$$\min T(x_{01}, x_{02}, \dots, x_{ij}, \dots)$$

の式を満たす  $x_{ij}$  の組合せを GA を用いて探索する。

以上の研究の目的はある時点での通信要求に対して、グローバルに平均遅延を最小化する帯域幅割当を実現することであり、その割当計算はあるノードが集中して行なっている。しかし現実のネットワークへの実装を考えた場合、一つのノードに計算を集中させることは望ましいことではない。なぜなら、ノードやリンクに障害が発生した場合には、割り当てが不可能になることが考えられるからである。よってネットワーク障害に対応するためアルゴリズムの分散化が必要となる。

## 3 D-GRA

これまでに筆者らにより Theeb ら [3] の GRA を分散化した D-GRA (Distributed GRA) が提案されている。[4]。D-GRA では、各ノードは現在

何番目の VP の帯域幅をどれだけ使用しているかという情報を保持している。この情報を経路情報として互いに交換し、この情報に基づき各ノードごとに最適と考えられる経路を選択し、選択した経路の適応度を計算する。適応度は選択した経路のネットワーク全体での適性を評価するために使用する。しかし D-GRA ではノードやリンクは正常に動いていることが前提であり、障害が発生した場合へは未対応であった。

そこで本研究では、ノードに障害が発生した場合を考慮した分散アルゴリズム、D-GRA Ver2(以下 D-GRA2) を設計する。D-GRA2 は、いずれかのノードに障害が発生した場合でもシステムが破綻することなく割り当てを行なうことを目指し、D-GRA を改善したアルゴリズムである。ノード障害が発生した場合は平均遅延が大きくなる可能性があるが、システムの回復を優先とする。ルーティングテーブルは D-GRA2 とは独立して一定時間ごとに更新されるので、割当を実行する前にノード障害が発生していた場合には障害の発生したノードを避けることができるが、割当の実行途中でノードに障害が発生した場合には、割り当てに必要なメッセージが届かないことになる。つまりノード障害に対応するためにはメッセージが届かない場合での動作を考慮したアルゴリズムの設計が必要となる。

### 3.1 アルゴリズムの概要

D-GRA2 では各ノードで二つのプロシジャが動いている。一つは計算要求プロシジャであり、通信要求を監視して通信が終了または発生した場合に経路計算バケットをブロードキャストする。もう一つは経路計算プロシジャであり、経路計算バケットを受信すると割当計算を実行する。各ノードにおける二つのプロシジャの概要は図 1 に示す。計算要求プロシジャは通信要求の監視とパケット送信を繰り返している。ここではより複雑な動作をする経路計算プロシジャについての概要を、図 1 の流れに沿って説明する。

まず経路計算プロシジャが経路計算バケットを受信すると、ルーティングテーブルから候補とする経路を選択する。次に選択した経路と要求帯域幅の組を経路情報としてブロードキャストし、同

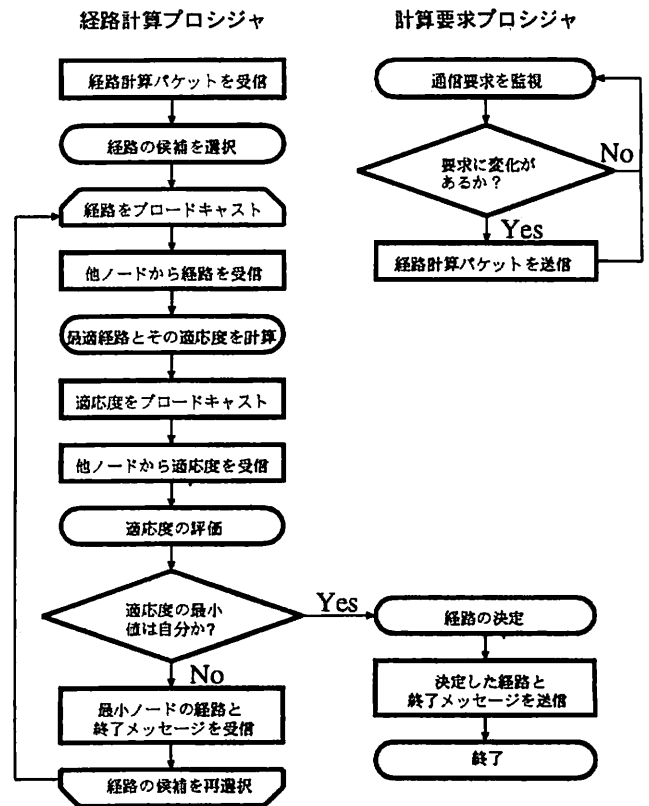


図 1: D-GRA2(各ノードごとの処理)

時に他ノードから経路情報を受信する。続いてその経路情報を元に、自ノードを起点とする最適な割り当てを GA を用いて計算し、求めた経路の適応度を計算する。求めた経路のネットワーク全体での適性を評価するため適応度をブロードキャストし、同時に他ノードから各ノードが求めた経路の適応度を受信する。全ノードから適応度を受信したら適応度の評価を行ない、最小適応度が自ノードであれば経路情報を更新し、終了メッセージを送信して計算を終了する。自ノードでなければ、適応度最小ノードが自分になるまで一連の動作を繰り返す。新たに求められた経路に基づく通信は、全ノードからの終了メッセージを受け取った場合に開始する。

### 3.2 メッセージの詳細

このアルゴリズムで交換されるメッセージは以下の 7 つである。それぞれのメッセージは ID で区別され、メッセージの発信元ノード ID とメッセージ内容が組になっている。

- MSG1: 計算要求
- MSG2: 計算開始のメッセージ
- MSG3: ブロードキャストされる経路情報
- MSG4: 他ノードから受け取る経路情報
- MSG5: ブロードキャストする適応度
- MSG6: 他ノードから受け取る適応度
- MSG7: 終了メッセージ

割り当て計算の途中にあるノードで障害が発生した場合、メッセージの到着が遅れることが考えられる。あるいは到着しないことも考えられる。つまりノード障害に対応するためにはメッセージの待ち時間と、メッセージが到着しない時にどう対応するかを定める必要がある。各メッセージはネットワークの混雑度合により遅延が発生する。D-GRA2ではネットワークの混雑度合を平均リンク使用率  $F_a$  で判断し、メッセージの待ち時間は平均 VP 容量  $C_a$  と  $F_a$  により (1) 式で求まる時間を基本の待ち時間  $T_a$  とする。

#### MSG1

MSG1は割り当ての実行を要求するメッセージ。ノードの通信要求に変化が起きた場合(新規の通信要求発生や現在の通信の終了)に、全ノードにブロードキャストする。ノードに障害が発生した場合、通信要求の監視ができないためMSG1は送信されない。割当実行中に他ノードからのMSG1を受信した場合には、計算が終了するまでこれを待ち行列に入れる。MSG1が送られてこないノードには割当を行わず、他のノードへはそのノードを経由しない経路に割り当てる。

#### MSG2

MSG2は割当計算を開始することを知らせるメッセージ。MSG1を受信するとブロードキャストする。割当計算はある程度同期的に行なわれることが望ましいのでMSG2の遅延を考え、MSG2をブロードキャストしてから  $2T_a$  経過したのち計算を開始する。

#### MSG3

MSG3は各ノードが自分の使用している経路情報を他のノードへ知らせるためのメッセージ。他

ノードから送信されるMSG3はMSG4として受信する。MSG4が返されてこないノードに対してはMSG3を再送する。

#### MSG4

他ノードから送られてくる経路情報をMSG4として受信する。割当の最適化のために重要なメッセージなので、 $3T_a$ の間到着を待ち、その間MSG3を再送する。それでもMSG4を送信してこないノードは障害が発生したと考え、そのノードへの割り当ては実行せず、他ノードに対してはそのノードを経由しない経路への割り当てを行なう。

#### MSG5

MSG5は各ノードが選択した経路の適応度を知らせるためのメッセージ。他ノードから送られてくる適応度はMSG6として受信する。MSG5を送信してから一定時間経過してもMSG6を受信しない場合はMSG5を再送する。

#### MSG6

MSG6は他ノードの選択した経路の適応度を知り、自ノードが選択した経路のネットワーク上での適性を評価するために重要なメッセージである。MSG4同様  $3T_a$  時間到着を待つ。自ノードの適応度が最も高い場合にはMSG7を送信し、そうでない場合はMSG2の送信から繰り返す。時間内にMSG6を受信できないノードがある場合は自ノードよりもそのノードの適応度の方が高いと考える事で再びMSG2の送信から繰り返す。

#### MSG7

MSG7は計算が終了した事を示すメッセージである。各ノードは計算が終了した時点でMSG7をブロードキャストする。全ノードの終了メッセージを受信するか、あるいは一定時間経過したのち通信を開始する。一定時間内にMSG7が送信されてこないノードは障害が発生していることが考えられ、開始した通信が滞る可能性がある。その場合はあらためてMSG1を送信して割り当てをやり直す。

### D-GRA2のアルゴリズム

```
proc_send_request(){
  while(true){
    if(新しい通信要求が発生 or 現在の通信の終了)
      {MSG1をブロードキャスト;}
  }
```

```

}/**end***/

proc_receive_request(){

    int    own_id, other_id[];
    int    MSG3,  MSG4[]; /* 経路情報 */
    float  MSG5,  MSG6[]; /* 適応度情報 */
    int    MSG2=0, MSG7=0; /* (0 or 1) */
    int    min_node;

    MSG2をブロードキャスト;
    while(MSG7==0){
        MSG3 = func_calc(3); /* 経路を選択 */
        <own_id,MSG3>をブロードキャスト;
        <other_id[],MSG4[]> = 他ノードから受信;

        MSG5 = func_calc(5); /* 適応度を計算 */
        <own_id,MSG5>をブロードキャスト;
        <other_id[],MSG6[]> = 他ノードから受信;

        min_node = func_evaluate(); /* 適応度の評価 */

        if(min_node == own_id){
            MSG3をブロードキャスト;
            送信実行;
            MSG7=1をブロードキャスト;
            計算終了(break);}
        else{
            MSG4[min_node] = min_nodeより経路情報を
            受信;
            min_nodeよりMSG7を受信;}
        }
    }/**end***/

func_calc(int type){
    int best_path, best_fitness;

    MSG4[]を元に経路を選択;
    for(i=0; i<5; i++){ /* 遺伝的操作 */
        選択した経路の適応度を計算;
        選択・交叉・突然変異;
    }

    best_path = 適応度を最小とする経路;
    best_fitness = その適応度

    if(type==3) return best_path;
    if(type==5) return best_fitness;
}/**end***/

func_evaluate(){
    float min_f,tmp_f;
    int node;
    min_f = MSD5,MSD6[]の中の最小値;

    for(i=0; i<ノード数; i++){
        tmp_f = MSD6[i];
        if(min_f > tmp_f[i]){

```

```

        node = i;
        return node;}
    else{
        node = own_id;
        return node;}
}/**end***/

```

## 4 シミュレーション実験

今回の実験では、まず GRA と D-GRA2での平均遅延を比較し、それぞれの利点・欠点について考察する。続いてノード障害発生イベントを生成し、D-GRA2の障害回復の機能を検証する。

### 4.1 実験条件

本研究のシミュレーション実験は、6ノードからなる ATM ネットワークを想定している。ネットワークの物理リンクは ATM スイッチにより計 11本の VP へ多重化されている。各ノードのルーティングテーブルは D-GRA2とは独立した定期的なパケット交換により自動的に更新される。発生するトラフィックは  $N(5, 1^2)$  の正規分布に平均通信量をかけた値とし、トラフィックの発生間隔はポアソン分布に従うものとする。障害が発生するのはノード 0のみとし、通信リンクその他に障害は発生しないものとする。

### 4.2 実験結果

GRA と D-GRA2の性能を、発生させる平均通信量を変化させネットワークの混雑度を変えた場合での平均パケット遅延により比較する。図 2にその結果を示す。横軸が混雑度 (%) であり、縦軸が平均パケット遅延 (ms) である。ここでの混雑度とは、全 VP 容量の合計に対する総通信量の割合である。図 2において GRA と D-GRA2(N)のグラフは全ノードが正常な場合での結果であり、D-GRA2(F)のグラフはノード 0に障害を発生させた場合での結果である。

続いてノード 0に障害を発生させた場合での実験を行なった。ノード 0に障害を発生すると GRA では割り当てそのものが全く不可能であった。しかし D-GRA2では、D-GRA2(N)のグラフより平均遅延が大きくなっているものの、システムが破

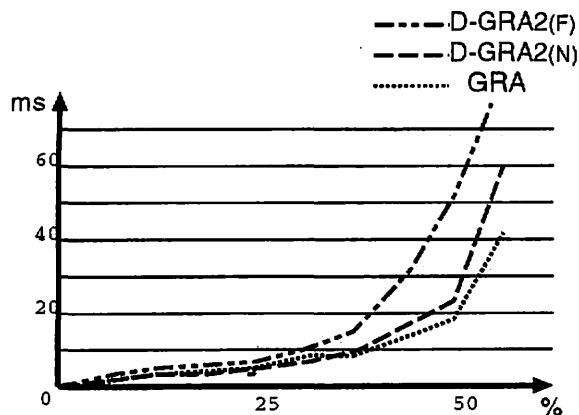


図 2: 平均パケット遅延の比較

綻する事なく割り当てを実行できている。その結果が D-GRA2(F) である。

### 4.3 考察

上記の結果より、D-GRA2 は一回の割り当てにおいて GRA に比べ、VP の平均使用率と (1) 式で求まる遅延の 8 倍の時間待たなければならない。すなわちリンク使用率が 25% では約 2.6ms、リンク使用率が 50% では約 8ms のオーバーヘッドが発生することになる。図 2 の GRA と D-GRA2(N) のグラフの差はほぼこのオーバーヘッドの分だけの遅延となっている。

最適な割当を実行するだけならば GRA が最も良いことになる。しかし GRA は一つのノードに障害が発生した場合に割り当て不可能となることが、ネットワークへの実装を考えた場合最大の問題点である。それに対し D-GRA2 はノードに障害が発生した場合にも割り当てが可能であることが示された。またネットワークの混雑が比較的小さい場合 (リンク使用率が 30% 程度まで) は GRA とほぼ同様の遅延で割り当てが可能である。D-GRA2(F) の結果が D-GRA2(N) の結果より遅延が大きくなったのは、障害が発生したノードを避けるため再割当を行なった事が原因と考えられる。

## 5 おわりに

本研究で検証した D-GRA2 は、ノードに障害が発生した場合でも機能することを実験により確認した。D-GRA2 は、特にネットワークの負荷が大

きい場合には GRA の性能には及ばないが、現実のネットワークへの実装を考えると D-GRA2 の持つ障害回復の機能が必須であると考えられる。

今回の実験ではノードに障害が発生した場合での検証のみ行なっているが、実際にはリンクを初め様々な障害が発生することが考えられる。よって今後はリンク障害やパケットロスなど様々な障害を想定し、より障害に強いアルゴリズムの設計を進めていく。さらに設計したアルゴリズムのシミュレーション実験を行ない、その有効性を示すことが課題である。

## 参考文献

- [1] Mario Gerla, Jose Augusto Suruagy Monteiro, Rodolfo Pazps: "Topology Design and Bandwidth Allocation in ATM Nets", *IEEE Journal on selected areas in communications*, Vol.7, No.8, pp.1253-1262 (1989)
- [2] Hong Pan, Irving Y. Wang: "The Bandwidth Allocation of ATM Through Genetic Algorithm", *IEEE Global Telecommunication Conference*, pp.0125-0129 (1991)
- [3] Theeb A. Al-Qahtani, Mohammed J. Abedin, Syed I. Ahson: "Dynamic Routing in Homogeneous ATM Networks Using Genetic Algorithms", *IEEE International Conference on Evolutionary Computation proceedings: IEEE World Congress on Computational Intelligence*, pp.114-119 (1998)
- [4] 小林英博, 棟朝雅晴, 佐藤義治: "遺伝的アルゴリズムを用いたネットワーク帯域幅割当", 情報処理学会研究報告, Vol99, No.94, pp.91-96 (1999)
- [5] 小林英博, 棟朝雅晴, 赤間清, 佐藤義治: "ネットワーク帯域幅割当のための分散アルゴリズム", 情報処理北海道シンポジウム 2000 講演論文集, pp.110-111 (2000)
- [6] WIDE Project 編: "インターネットオペレーション原理と実際", bit 別冊, 共立出版 (1999)