

遺伝的操作を用いた適応型ルーティング

3P-7

天野美樹 日高東潮 高井昌彰 佐藤義治
(北海道大学 工学部)

1 はじめに

現在のようにネットワークが発達し規模が拡大するにつれて、高速かつ安定した経路を確保し、通信をスムーズに行なうためのルーティングが重要となる。しかし、ネットワークの負荷状態は動的に変化するため、多く使用されている静的なルーティング法では十分に対応できない。本稿では、環境が変化する問題に対して有効なアルゴリズムの1つである遺伝的アルゴリズム [1] をベースとした遺伝的操作を用いて、リンクの負荷が変化するネットワーク通信に関し、通信遅延時間が最小になるように経路を決定する適応型ルーティング手法を提案する。

分散型とする。そのため、各ノードはシステム起動時のネットワークトポロジー情報を保持しているが、リンクの故障やノードの新設によるトポロジー変化は考慮せず、リンクの負荷変化に対応してできるだけ高速なパケット伝送経路を確保することのみを目的とする。

以下、遺伝的操作と各ノード内のアルゴリズムを示す。

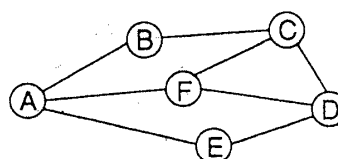


図 1: ネットワーク例

2 遺伝的操作

遺伝的アルゴリズムは1960年代に John H. Holland らによって提唱された、生物の進化と淘汰の現象をモデルとした近似最適化手法である。このアルゴリズムは文字列にコーディングされた遺伝子(解)集団に対し、基本3操作である Selection(淘汰), Crossover(交叉), Mutation(突然変異)を繰り返し施すことにより目的関数の評価値が高い解を得るものである。遺伝的アルゴリズムは (1) 遺伝子を集団で保持しているため多様性が確保できる、(2) 生物が進化し環境に適応する過程をアルゴリズム化している、という点より環境変化への適応能力が高いと考えられる。そこで本稿では、負荷変動に対応して経路を割り当てるため、Selection、Mutation を導入したルーティング手法を提案する。

3.1 経路リストに基づくルーティング

パケットの伝送要求が発生するとまず経路リスト(表1)を参照し、目的ノードへの経路候補(Route)が登録されていれば評価値(Delay)を基に採用経路を選択する。経路リストに登録されていなければ Dijkstra のアルゴリズムにより最短経路を導出し、決定した経路をパケットのヘッダに経路情報として付加した上で送信バッファへ送る。

表 1: 経路リスト

Destination	Frequency	Delay (ms)	Route
B	4	50	AB
		50	ABC
C	14	20	AFC
		40	AFDC
		60	AFD
D	7	20	ABCD
		30	AE

3 ルーティングアルゴリズム

ここで対象とするネットワークとは、特定のものではなく、いくつかのホスト(ノード)が回線(リンク)によって接続された任意のグラフ構造で表されるネットワークを想定する(図1)。通信とはネットワーク上でリンクを通してノード間でパケットを転送することである。ルーティングの集中管理は行なわず、各パケットが発生したノードで目的ノードまでの経路を決定する準

受信処理は受信バッファよりパケットを1つ取り込み、ヘッダを調べ、目的ノードが現在のノードでなければ経路情報に従って転送する。目的ノードでかつ評価パケットならば使用経路の評価値更新を行ない、基準を満たした場合、遺伝的操作を行なう。評価パケットとは、目的ノードまでのある経路に従った通信に要した

実際の遅延時間を発信元へ送り返す特別なパッケージである。評価パッケージでなければ、通信にかかった時間を計算し、評価パッケージを発信元ノードへ送り返し、パッケージを組み立てバッファへ入れる。受信バッファにパッケージが残っている間、以上の操作を繰り返す(図2)。

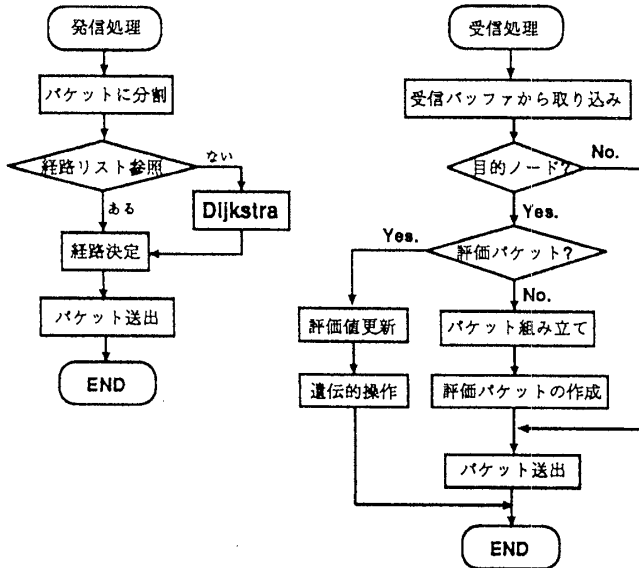


図2: ノードの処理手続きの流れ

3.2 遺伝的操作

遺伝的操作は各ノードの持つ経路リストまたは経路候補に対して施される。経路リストは目的ノード別に複数の経路候補が登録可能で、各経路候補は目的ノードから送り返される評価パッケージを参照し、実際の遅延時間から得た評価値が割り当てられている。経路候補がないノードへ伝送する場合はDijkstraのアルゴリズムを実行し、得られた経路を登録の上で使用。また、評価値更新数(Frequency)が一定値を越えた経路候補に対してMutationを行ない、部分迂回路を含んだ新たな経路候補を生成し、追加登録する。

Mutationでは、まずある経路候補から始点と終点をランダムに選択し、隣接する異なるノードを迂回ノードとして選ぶ。次に、迂回ノードから終点までの最短経路をDijkstraのアルゴリズムにより決定し、新たな部分経路とする(図3)。この操作により使用頻度の高い目的ノードへの経路候補が多くなり、特定リンクへの負荷集中を防ぐことができる。しかしMutationを繰り返すと、経路候補の増加によるノード内メモリの圧迫を招く。そこで、経路候補数の上限に達した場合にSelectionを行ない経路リストの増大を回避する。ある目的ノードに対する経路候補数が上限に達した場合、評価値が最も悪いものをMutationで生成された経路候補と交換する(Selection I)。また、経路リスト全体で規

定値を越えた場合、受信パッケージ数の少ないノードに対する経路候補を全て削除する(Selection II)。

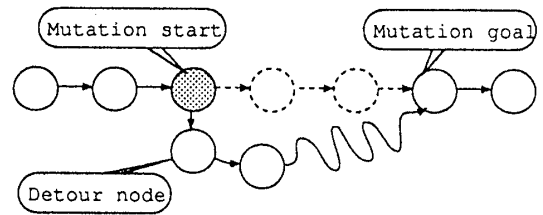


図3: Mutation

3.3 考察

提案するルーティング手法は、集中管理や負荷情報交換を行なうかわりに遺伝的操作による学習を採用した点に特徴がある。遺伝的操作で重要といわれるCrossoverを用いなかったのは、部分経路交換を行なえる場合が限られていることにより、Crossoverの構成が困難であることがあげられる。

適応型ルーティングで最も重要なのは負荷状況を知り、素早く対応して適当な転送先を決定する方法である。しかし、本手法では発信時に目的ノードまでの経路を全て決定しているため、転送途中に急激な負荷変動が生じた場合には対処不可能である。これを回避するための手段として、階層構造の導入などが考えられる。また、経路を評価するための評価パッケージが多く送信されるので、それによる通信負荷の増大が問題となる。そこで、少ない情報で評価を行なう手法の開発が今後の課題となる。

4 まとめ

本稿では遺伝的操作の導入により最適な経路の学習を行なう適応型ルーティング手法を提案した。今後の課題としては、リンクやノードの故障時の対応があげられる。

参考文献

- [1] David E. Goldberg : *Genetic Algorithms in Search, Optimization, and, Machine Learning*, Addison Wesley, 1989.
- [2] 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.
- [3] E. Minieka : "On Computing Sets of Shortest Paths in a Graph", *Communications of the ACM*, vol. 17, no. 6, June 1974.