

特別研究報告

題目

ネットワークの変動に追隨するパレート最適制御に基づく
ネットワーク省電力化手法の提案と評価

指導教員

村田 正幸 教授

報告者

秋下 耀介

2016年2月16日

大阪大学 基礎工学部 情報科学科

ネットワークの変動に追従するパレート最適制御に基づく
ネットワーク省電力化手法の提案と評価

秋下 耀介

内容梗概

近年、ストリーミング配信や、クラウドサービス等のインターネットを介したサービスの普及に伴うトラフィックの増加により、ネットワークにおける消費電力の増加は大きな課題となっている。そこで、ネットワークの消費電力を削減する手法の検討が進められている。ネットワークの消費電力を削減する手法では、ネットワークを流れるトラフィック量が時間帯により大きく異なることから、各時間帯において必要な通信性能を確保するのに必要十分なネットワークを構築し、不要なネットワークの機器やサーバをスリープさせる。これによって、通信量が多く、少数のネットワーク機器では十分な性能を確保することができない場合は、多数のネットワーク機器を動作させることにより十分な処理性能を確保し、逆に、通信量が少ない時間帯には、多くの機器をスリープさせることによって低消費電力化が可能となる。

従来、ネットワーク低消費電力化手法の検討では、性能と消費電力のトレードオフに焦点があてられており、信頼性の確保については考慮されてこなかった。しかしながら、現実のネットワークサービスでは、故障が発生した際にも、故障発生により性能低下する時間を一定以下とすることが求められる。

そこで、本報告では、短時間のトラフィック変動や故障などの環境変動に追従して、十分な通信性能、信頼性の確保と低消費電力化の3つの目的を達成するネットワーク制御手法を確立する。ネットワークの制御を行うにあたり、耐故障性を確保しようとするればオンになるノードやリンクが増加するため消費電力は増加し、性能を確保しようとしても消費電力は増加する。本報告では、これらの指標をすべて考慮した制御を実現する手法として、パレート最適解の集合（パレートフロント）を求め、そのうち、必要な性能・信頼性の制約を満たす解をネットワークに投入することにより、性能・信頼性の要件を満たす範囲内で、消費電力を最小化する。

本制御をネットワーク内の環境変動に追従して行うためには、パレートフロントの計算も環境変動に追従して行うことが必要となる。そこで、本報告では、パレートフロントの計算を高速に行う手法を検討する。本手法では、前の時刻のパレートフロントと、環境変動発生

時にパレートフロントとなる可能性の高い解を組み合わせた集合を初期解とし、進化計算により、現在の環境に合わせたパレートフロントを探索する。これにより、少ないステップで適切なパレートフロントにたどり着くことができ、高速なパレートフロントの取得が可能となる。

本報告では、提案手法をシミュレーションにより評価を行い、提案手法を用いることにより、環境変動に追従し、性能・信頼性の要件を満たしつつ、トラヒックが少ない時間帯であれば、全機器稼働状態の39%までネットワークの消費電力を低減することができることを示す。さらに、前の時刻のパレートフロントのみで初期解を生成する手法では、各ステップにおいて50世代の進化計算を行う場合では、制約条件を満たす解に到達するまで16ステップを要するような環境下においても、前の時刻のパレートフロント、環境変動発生時にパレートフロントとなる可能性の高い解を組み合わせた集合を初期解と設定する手法では、1ステップ以内に制約を満たした解に到達することができることを示す。

主な用語

ネットワーク省電力化，多目的最適化，パレート最適解，パレートフロント，進化計算

目次

1	はじめに	6
2	関連研究	8
2.1	ネットワークの省電力化	8
2.2	多目的最適化問題	8
3	パレートフロント	11
3.1	パレート最適解とパレートフロント	11
3.2	パレートフロントの導出	12
4	パレート最適制御に基づくネットワーク省電力化	15
4.1	パレート最適制御のネットワークへの適用	15
4.2	環境変動への対応	24
5	評価	28
5.1	評価環境	28
5.2	評価結果	32
6	おわりに	43
	謝辞	44
	参考文献	45

目 次

1	トポロジ例	9
2	ネットワークの省電力化	10
3	パレート最適解	11
4	非優越ソート	13
5	パレート最適制御における全体の流れ	16
6	遺伝子の構造	17
7	経路情報の遺伝子座生成	18
8	リンク情報の遺伝子座生成	19
9	経路情報のデコード	20
10	リンク情報のデコード	21
11	交叉	22
12	経路情報に関する突然変異	23
13	突然変異失敗	24
14	ネットワークへの解の投入	25
15	環境変動が起きた際の初期値の設定	27
16	FatTree 型トポロジ	28
17	総トラヒック変動パターン A	30
18	各通信におけるトラヒック変動パターン A	30
19	総トラヒック変動パターン B	31
20	各通信におけるトラヒック変動パターン B	31
21	消費電力(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 1)	34
22	性能(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 1)	35
23	耐故障性(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 1)	36
24	消費電力(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 2)	37
25	性能(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 2)	38
26	耐故障性(トラヒックパターン = A, 各時刻の世代数の設定 = パターン 2)	39
27	消費電力(トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)	40
28	性能(トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)	41
29	耐故障性(トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)	42

表 目 次

1	ネットワーク機器	28
2	進化計算におけるパラメータ	29
3	評価に用いた世代数	32

1 はじめに

近年、ストリーミング配信や、クラウドストレージのようなクラウドサービス等のインターネットを介したサービスが普及するにつれて、ネットワーク内のトラヒックは増加し続けている [1]。トラヒック増加に伴い、ネットワークの消費電力は増加の一途にあり、大きな課題となっている [2,3]。

ネットワーク全体の制御における低消費電力化は、不要なネットワーク機器やサーバをスリープさせることで実現可能である。従来、このような低消費電力化は、1日周期の通信需要変動を想定した手法が検討されてきた [4-8]。これらの手法では、通信量が多く、少数のネットワーク機器では十分な性能を確保することができない場合は、多数のネットワーク機器を動作させ、十分な処理性能を確保し、逆に、通信量が少ない時間帯には、多くの機器をスリープさせることによって低消費電力化が可能となる。

これらの従来のネットワーク省電力化手法では、信頼性の維持は考慮されていない。これらの手法では、省電力化のために、一部ノードやリンクへトラヒックを集約し、他ノード、リンクをスリープさせる。しかしながら、ノードやリンクをスリープさせることにより、冗長性が除去され、機器の故障発生時にはネットワークの接続性が確保されず、サービスの停止を招く可能性がある。それに対して、現実のネットワークサービスでは、故障が発生した際にも、故障発生により性能低下する時間を一定以下とすることが求められる。

そこで本報告では、性能・信頼性の両面から SLA (Service Level Agreement) を保証しながらも省電力化を実現するネットワーク制御を確立する。耐故障性を確保しようとするればオンになるノードやリンクが増加するため消費電力は増加し、性能を確保しようとしても消費電力は増加する、といったように、消費電力、信頼性、性能の指標は、ある指標を向上させると、別の指標が悪化してしまう密接な関係にある。本報告では、複数の指標を考慮して最適化を行うパレート最適制御をネットワークに適用する。パレート最適制御においては、多目的進化計算 [9-13] によりパレート最適解の集合 (パレートフロント) を求める。本報告では、消費電力、信頼性、性能の3つの指標についてのパレートフロントを進化計算により求める。そして、SLA を保証できる範囲内で最も低消費電力な解をネットワークに投入する。

本制御をネットワーク内の環境変動に追従して行うためには、パレートフロントの計算も環境変動に追従して行うことが必要となる。そこで、本報告では、パレートフロントの計算を高速に行う手法を検討する。本手法では、前の時刻のパレートフロントと、環境変動発生時にパレートフロントとなる可能性の高い解を組み合わせた集合を初期解とし、進化計算により、現在の環境に合わせたパレートフロントを探索する。これにより、少ないステップで適切なパレートフロントにたどり着くことを可能とする。

本報告の内容は以下の通りである．まず第 2 章では，関連研究を述べ，第 3 章でパレートフロントを求める手法について述べる．次に第 4 章でパレート最適制御のネットワークへの適用について述べ，第 5 章では本手法のシミュレーション評価の環境とシミュレーション結果を述べる．最後に，6 章で本報告のまとめと今後の課題について述べる．

2 関連研究

2.1 ネットワークの省電力化

ネットワークにおける省電力化は、ノードやリンクをスリープすることで実現される（図 1,2）。ネットワークの省電力化については、様々な手法が検討されている [4–8]。文献 [4] では、ネットワークの輻輳を避けつつ、電力消費を最小限にすることを目指している。そのためには、輻輳の回避と電力消費の最小化という、多目的最適化問題を解く必要がある。これに対して、本手法では、これらの指標の重み付きの和を最小化する混合整数線形計画問題として定式化し、定式化された混合整数計画問題を定期的に解くことにより、輻輳の回避と電力消費の最小化を達成する。文献 [5,6] においても、同様の最適化を目的とした検討がされている。特に文献 [5] では、Internet Service Provider (ISP) ネットワークにおいて、完全接続の達成と最大リンク利用率を閾値以下とする制約を保証しつつも、より多くのノードとリンクをオフにすることを目的とした整数線形計画問題を定式化し、その発見的な解法を提案している。本手法では、リンクのオフよりも、ノードのオフによる省電力化の効果が大きい。そのため、まずはネットワーク内においてオフにできるノードを探し出し、これをオフにする。その後、リンクをオフにする操作を繰り返す。文献 [7] では、冗長なトラフィックを取り除く機能を持つルータで構成されたネットワークにおいて、低消費電力化を行う手法を提案している。冗長なトラフィックを取り除く機能をルータで機能させることにより、ネットワークが提供可能な帯域を向上できる。しかしながら、本機能はルータの消費電力を増大させる [8]。そこで、文献 [7] では、本機能をどのルータで動作させるのかを、スリープするリンクの位置と合わせて決める手法を提案している。

しかしながら、これらはいずれも性能と消費電力のトレードオフに焦点が当てられており、稼働状態にあるネットワーク機器の故障等については考慮されていない。現実のネットワークにおいては、ノードの故障などが発生した際にも、故障発生により性能低下する時間を一定以下とすることが求められる。そこで本報告では、性能・信頼性の両面から SLA を保証しながらも省電力化を実現するネットワーク制御を確立する。

2.2 多目的最適化問題

本報告では、性能、消費電力、信頼性の 3 つの目的を同時に最適化する。このような多数の目的関数を最適化する問題は、多目的最適化問題と呼ばれる。多目的最適化問題では、全ての目的関数値を最小（最大）化するような解は存在せず、優劣のつけられない解（パレート最適解）を求める [9–14]。

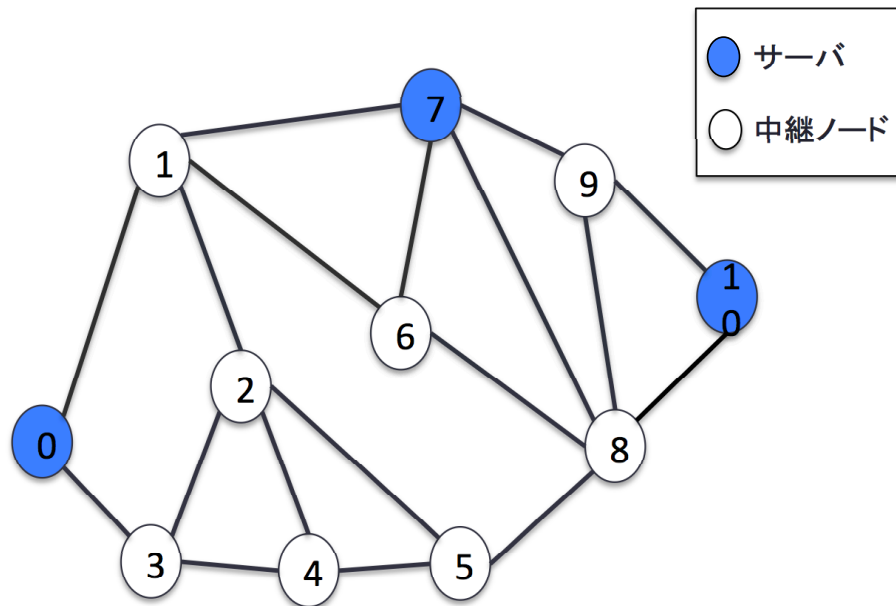


図 1: トポロジ例

多目的最適化問題は、様々な分野での適用が進められている。文献 [9] では、High Performance Computing (HPC) アプリケーションのスケジューリングにおいて、エネルギー消費、 CO_2 、プロバイダの利益、という3目的の最適化を試みている。本手法では、多目的進化計算 [11, 12] を用いることでパレートフロント（パレート最適解集合の形成する面）を導出する [10, 15]。そして、パレートフロントの中から一定の選択基準に従い選択した解を実際に投入する。文献 [9] では、シミュレーション評価により、本手法が消費電力や CO_2 排出を抑えつつも、プロバイダの利益が増えるようなスケジューリングが可能となることを示している。

文献 [13] では、再生可能エネルギーのシステムにおいて、システムの総コスト率の最小化と、システムのエクセルギー効率の最大化を多目的最適化問題として定式化している。そして、多目的進化計算でパレートフロントを導出し、パレートフロントとして存在している解集合から、エクセルギー効率と総コスト率に関する密接な式を得たことを示している。

文献 [14] では、水道システムにおいて、コストの最小化と性能の最大化の両立を図る設計問題を 12 種類準備し、各問題に対して 5 つの多目的進化計算手法を適用し、それらの解の比較を行っている。その結果、エリート主義を導入し、集団中の個体を順位づけして生存選択を行う Nondominated Sorting Genetic Algorithm-II (NSGA-II) [16] が多目的進化計算の中でも最も良い結果を出したことを明らかにしている。

以上のように、パレートフロントを求め、その中から適切な解を選択する手法が様々な分

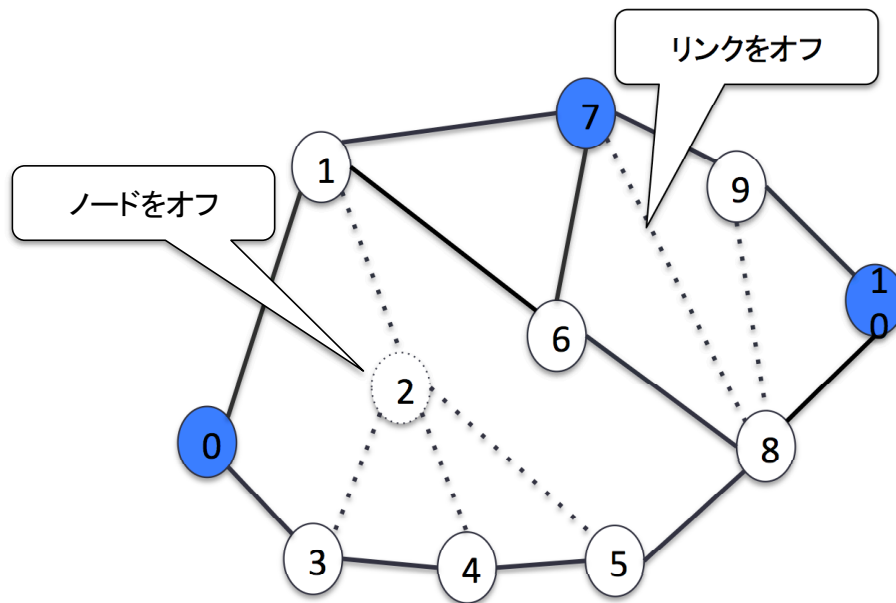


図 2: ネットワークの省電力化

野で適用されている。本報告においては、多目的最適化問題をネットワーク省電力化制御に適用し、消費電力の最小化、耐故障性の最大化、性能の最大化という3目的の多目的最適化問題を解くにあたって、パレートフロントを求め、その中からネットワークに投入する解を選出する。

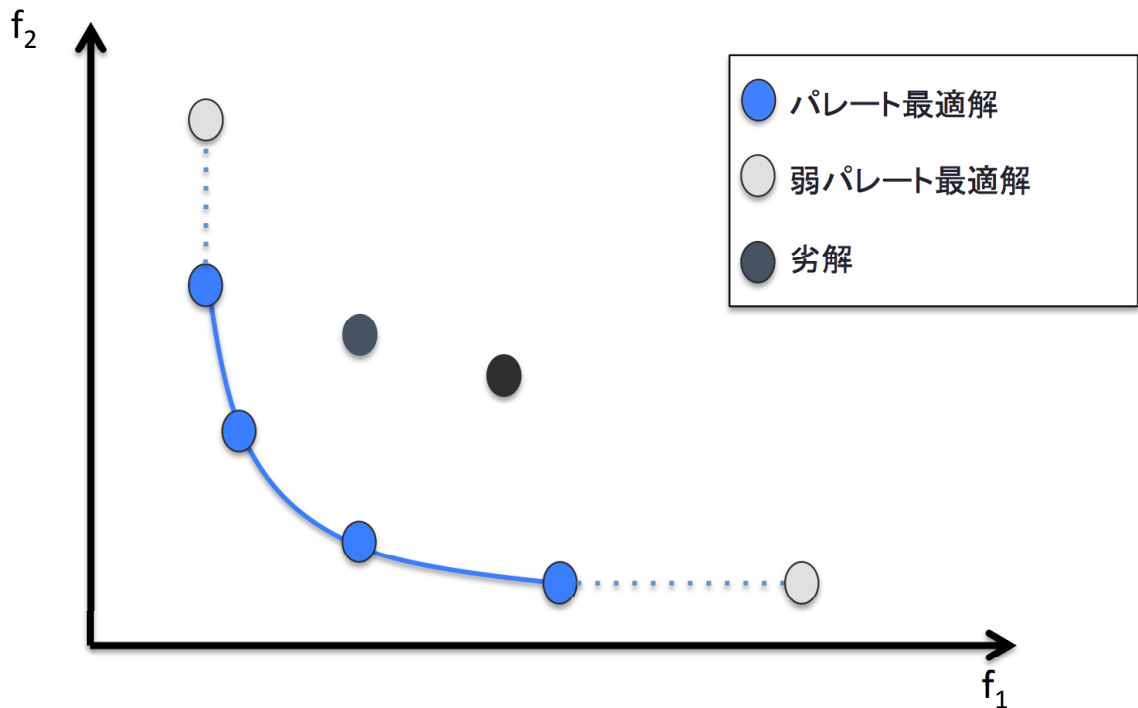


図 3: パレート最適解

3 パレートフロント

3.1 パレート最適解とパレートフロント

多目的最適化問題においては、目的関数が互いに競合しあっているため、与えられた全ての目的関数に対して完全最適解を求めることはできない。そのため、多目的最適化では「ある目的関数の値を改善するためには、少なくとも他の1つ目的関数の値を改悪せざるをえない解」を求める。図3に示されるように、 x に優越（全ての目的関数値が x より優れるまたは同じであり、かつある目的関数値においては x より優れる）する $x' \in \pi$ が存在しない解をパレート最適解、 x に強い意味で優越（全ての目的関数値が x より優れる）する $x \in \pi$ が存在しない解を弱パレート最適解と呼ぶ。すなわち、式(1)と(2)が成り立つ x が存在しないような x^* がパレート最適解であり、式(3)が成り立つ x が存在しないような x^* が弱パレート最適解である。

$$f_i(x) \leq f_i(x^*) \quad \forall i = 1, \dots, p \quad (1)$$

$$f_i(x) < f_i(x^*) \quad \exists i \in \{1, \dots, p\} \quad (2)$$

$$f_i(x) < f_i(x^*) \quad \forall i = 1, \dots, p \quad (3)$$

ただし, $f_i(x)$ は i 番目の目的関数である.

一般にパレート最適解は複数存在し, パレート最適解の集合は, 面を形成する. このパレート最適解集合が形成する面のことを, パレートフロントと呼ぶ.

3.2 パレートフロントの導出

パレート最適解は進化計算を用いることにより, 導出可能である [16,17]. パレート最適解を導出するための進化計算を多目的進化計算と呼ぶ. 多目的進化計算では生成されたある個体群に対して交叉や突然変異を行うことにより, 各個体(すなわち解)を真のパレート最適解集合に近づける. 以下にアルゴリズム全体の流れを示し, 以降で各手順の詳細を示す.

1. 初期化: N 個の個体群 P の生成
2. 評価: 個体群 P の評価, 各個体のランク付け実行, 各ランクにおける各個体の密度の評価
3. 子の生成: 遺伝子操作(選択, 交叉, 突然変異)を実行, N 個の子集団 Q を生成
4. 世代交代: $P \cup Q$ から解を選出し, N 個を P として構成
5. 終了判定: 終了条件を満たした場合終了. そうでない場合は3へ

3.2.1 初期化

ランダムに N 個の初期個体群 P の生成を行う. この個体群 P に対して遺伝子操作を行うことで新たな個体を生成していく.

3.2.2 評価

生成された個体群の評価, 非優越ソート [16] による各個体のランク付けの実行, 混雑距離の計算による各ランクにおける各解の密度の評価を行う.

非優越ソートは, 個体をランク付けする手法である. 非優越ソートでは, ある個体において比較対象となる個体とどの評価値においても優れていることを「支配している」と呼び, 各個体について, 当該個体を支配している個体数を基準としてランク付けを行う(図4). 以下に非優越ソートの手順を示す.

1. n を 1 に初期化する.

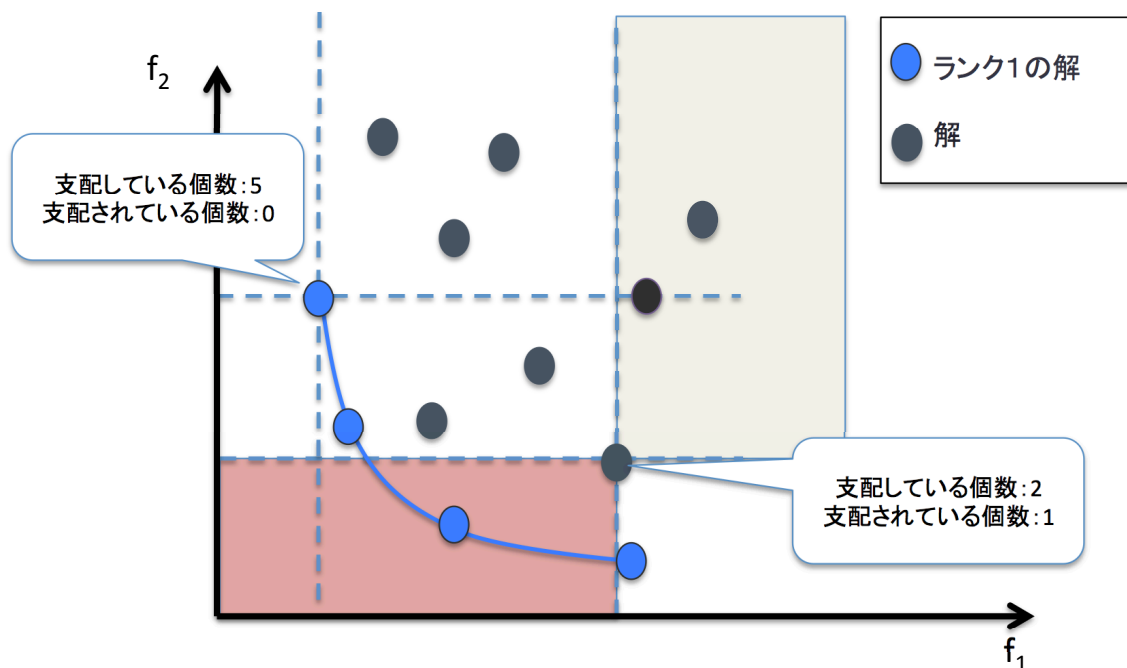


図 4: 非優越ソート

2. 各個体に対して、当該個体が支配している個体の数と、当該個体を支配している個体の数を数える
3. いずれのランクのリストにも含まれていない各個体について、当該個体を支配している個体数を調べ、その値が0の場合は、ランク n の個体のリスト F_n に加える
4. F_n 中の各個体が支配している個体について、当該個体を支配している個体の数から1を引く
5. いずれのランクのリストにも含まれていない個体数が0であれば、終了。それ以外は2へ戻る

上記の非優越ソートでは、同じランクの個体が多数生じる。同じランクの個体間での優先順位は、混雑距離 [16] を用いて確定する。混雑距離はある個体の周りにある個体の密度を示す指標であり、個体 j の混雑距離 d_j は、以下の式で定義される。

$$d_j = \begin{cases} \infty & i \text{ のいずれかの指標が全個体中最大あるいは最小値となる場合} \\ \sum_m \frac{f_m(I_m^{\text{next}}(j)) - f_m(I_m^{\text{prev}}(j))}{f_m^{\text{max}} - f_m^{\text{min}}} & \text{otherwise} \end{cases}$$

ただし、 $I_m^{\text{prev}}(j)$ は指標 m が j の次に大きい個体を示し、 $I_m^{\text{next}}(j)$ は指標 m が j よりも大きい個体のうち、もっとも指標 m が小さい個体を示す。また、 f_m^{max} は指標 m の最大値、 f_m^{min}

は指標 m の最小値である．混雑距離が大きい個体を選択することにより，個体の周りに別の個体が存在しない解を優先することができ，解の多様性を維持することができる．

3.2.3 子の生成

P に対して遺伝子操作（選択，交叉，突然変異）を実行し， N 個の個体群 Q を生成する．選択では次の世代を構成する個体を生成するために，その親となる個体を選択する．そして交叉では，選択された 2 つの親の間で遺伝子情報を一部入れ替え，子となる新しい個体を 2 つ生成する．突然変異では生成された子に対して遺伝子情報の一部を変化させる．これにより交叉だけでは得られない多様な解を得ることができる上，初期収束を防ぐことも可能となる．

3.2.4 世代交代

個体数を N 個に維持するために，個体群 P と遺伝子操作で得られた個体群 Q の計 $2N$ 個の個体から N 個の個体を選出し，次の世代の個体群 P を生成する．本世代交代を行うにあたり，まず， $P \cup Q$ に対し非優越ソートと混雑距離の計算を行うことにより，各個体を評価し，優先順位をつける．そして，優先順位が高い個体から N 番目の個体を次の世代の個体群 p に加える．

現在の世代の個体のうち，非優越ソートの結果，ランク 1 と判定された個体は，パレートフロントを形成する可能性のある個体である．そのため，他の個体とは別に，ランク 1 の個体をパレートアーカイブと呼ばれる領域に保存し，他の個体と区別できるようにする．

3.2.5 終了判定

多目的進化計算では，一定の世代数連続してパレートアーカイブ内の解が更新されない場合，または世代がある数に到達した場合，パレートフロントを求める処理を終了する．この際，パレートアーカイブに保存されていた解が求まるパレートフロントとなる．

4 パレート最適制御に基づくネットワーク省電力化

本節では、まず、多目的最適化を適用したネットワーク制御の概要を示す。そして、ネットワーク制御に用いる多目的最適化問題の具体的な解法と、実際のネットワークに投入する解の選択方法について述べる。最後に、ネットワークにおいてトラフィック変動などの環境変動が起こった際に瞬時に環境変動に対処可能な、多目的最適化にもとづくネットワーク制御手法を提案する。

4.1 パレート最適制御のネットワークへの適用

図5にパレート最適制御によるネットワークの省電力化制御の概要を示す。本制御では、集中制御コントローラは、定期的にネットワーク内のトラフィック観測情報を収集、各時刻のトラフィック状況に応じて、制御対象のネットワークを経路、機器の電源の投入状況を制御することにより、ネットワークの低消費電力化を行う。本報告においては、各時刻の制御において、集中制御コントローラは、ネットワークの性能、信頼性の条件を満たしつつ、消費電力を最小化するようなネットワーク設定を計算し、投入する。本計算をするにあたり、毎回パレートフロント解を導出し、パレートフロント解で、性能、信頼性の制約条件を満たすものうち、最も消費電力が小さい解を投入する。これにより、ネットワークの性能、信頼性の条件を満たしつつ、消費電力を最小化することが可能となる。

以下に、各時刻において集中制御コントローラが行う手順を示す。

1. 環境取得：現在の環境を取得する
2. 進化計算実行：進化計算を実行し、パレートフロントを導出する
3. 制御：2で得られたパレートフロントの中から、制約条件を満たす解をネットワークに投入する

上記の手順のうち、進化計算によるパレートフロント解の導出は3.2節に示した手法を用いる。ネットワークの省電力化問題における進化計算によるパレートフロント解の導出を行うためには、個体の遺伝子の与え方、個体の評価方法、遺伝子操作方法を定める必要がある。以下に、個体の遺伝子の与え方、個体の評価方法、遺伝子操作について述べる。

4.1.1 遺伝子の生成

本ネットワーク制御で用いる遺伝子を図6に示す。本遺伝子では、全機器間の通信の収容先の経路と、どのリンクの電源を投入するかという情報を含んでいる。本遺伝子は、この情

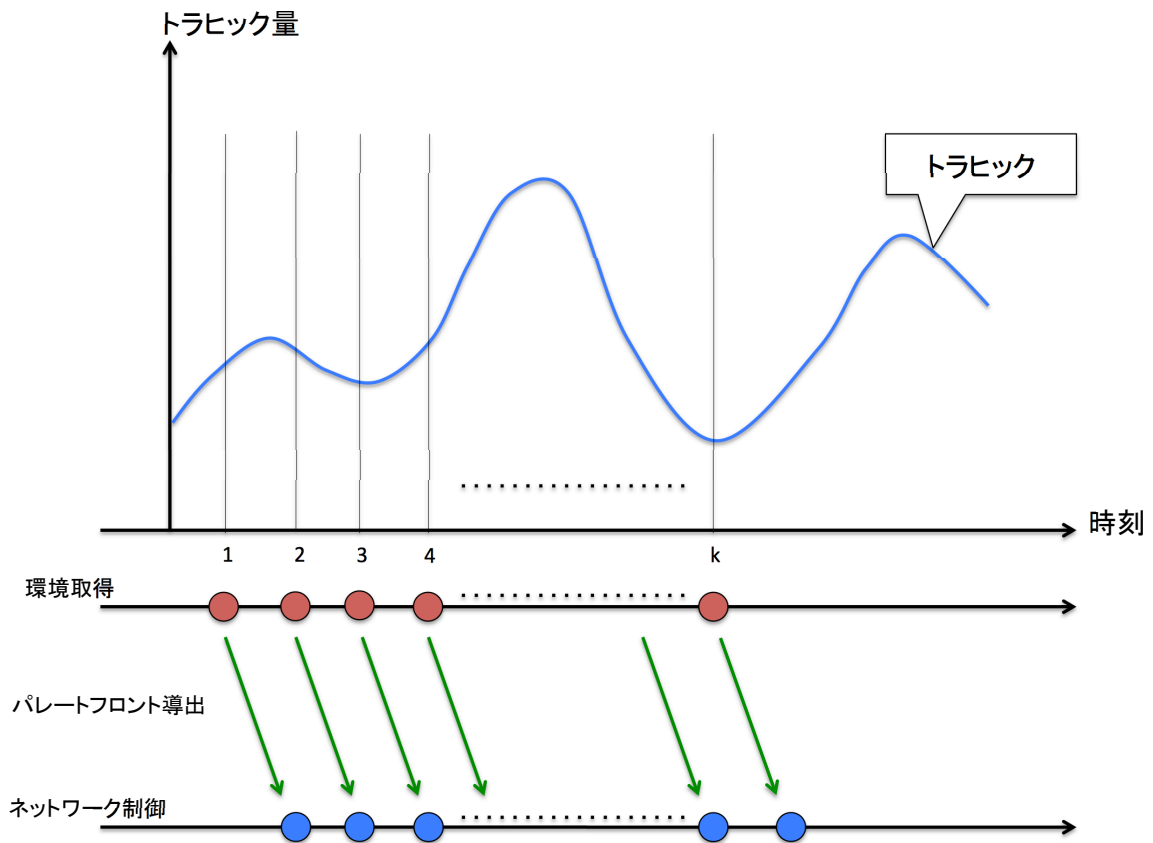


図 5: パレート最適制御における全体の流れ

報をベクトルとして表現する．そして，本遺伝子では，この2種類の情報の境界に，-1 という値を入れている．これにより，情報の境界を判別することが可能となる．

進化計算においては，初期解を与える必要がある．本報告においては，(1) ランダムに機器間の経路を生成する，(2) 生成された経路にもとづき，電源を投入するリンクの情報を生成することにより，初期解を生成する．

図 7 に経路の生成手順を示す．本手法では，まず出発点から隣接しているノードのうち一つをランダムに選択する．次に，選ばれたノードの隣接ノードを選択する．これを到着点に至るまで繰り返す．このとき同一のノードを含むループが生成された場合は，当該ループを破棄し，もう一度経路生成を行う．

次に図 8 で示されるように，生成された経路情報から，電源を投入するリンクの情報を生成する．本手順では，各経路とリンクの対応表を確認し，いずれかの経路が経由するリンクに該当する要素を 1 とする．

図 9，図 10 に示されるように，遺伝子を見れば通信が行われる各機器間の経路，および

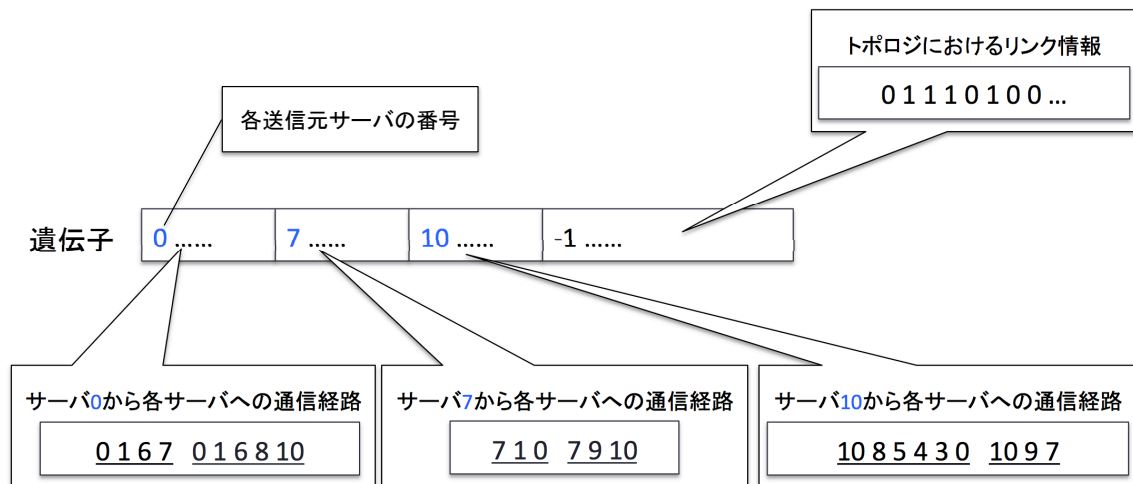


図 6: 遺伝子の構造

稼働させるノード，リンクがわかるものとなっている．

4.1.2 個体の評価

本手法では，消費電力，信頼性，処理遅延の3種類の指標をもとに各個体を評価し，非優越ソートを行うことで，各個体にランクを付ける．以下に，各指標における評価関数を示す．

消費電力 本報告では，消費電力は，電源を投入する必要があるノードの消費電力とリンクの消費電力の和をネットワークの消費電力とする．ネットワークの消費電力 $E^{\text{net}}(x)$ は，

$$E^{\text{net}}(x) = \sum_{(i,j) \in L} E^{\text{Link}} p_{i,j}(x) + \sum_{k \in V} E^{\text{Node}} p_k(x) \quad (4)$$

で求まる．ここで， L はリンクの集合， V はノードの集合， E^{Link} はリンク1本あたりの消費電力， E^{Node} はノード1つあたりの消費電力である． $p_{i,j}(x)$ は遺伝子 x において， $i-j$ 間のリンクの電源を投入する場合は1，それ以外は0となる変数であり， $p_k(x)$ はノード k の電源を投入する場合は1，それ以外は0となる変数である．

信頼性 本報告では，リンクが故障した際にも，電源をすでにオンにしている経路のみで，全機器間の接続性が確保できるようなネットワーク設定を投入する．つまり，各機器間に独

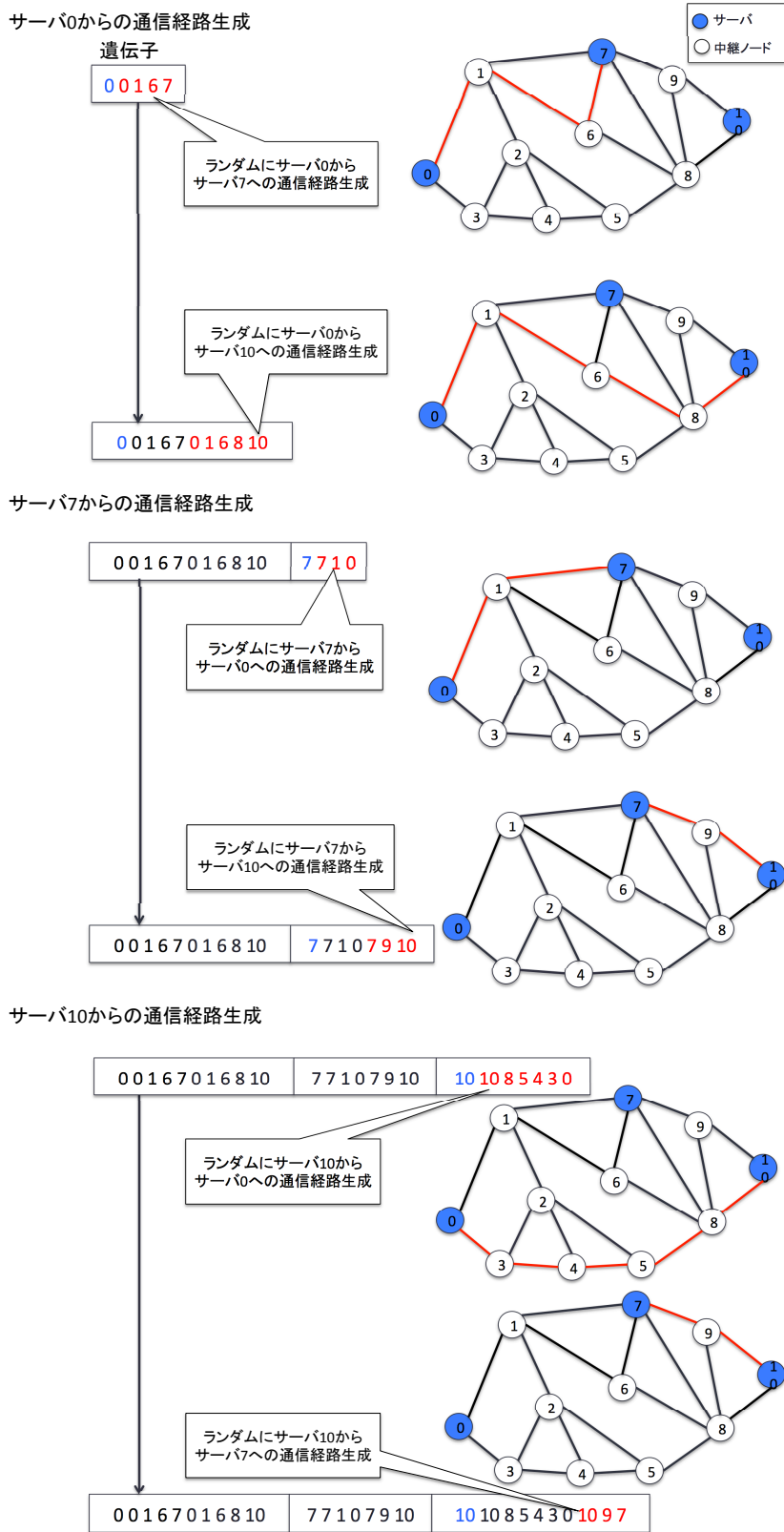


図 7: 経路情報の遺伝子座生成

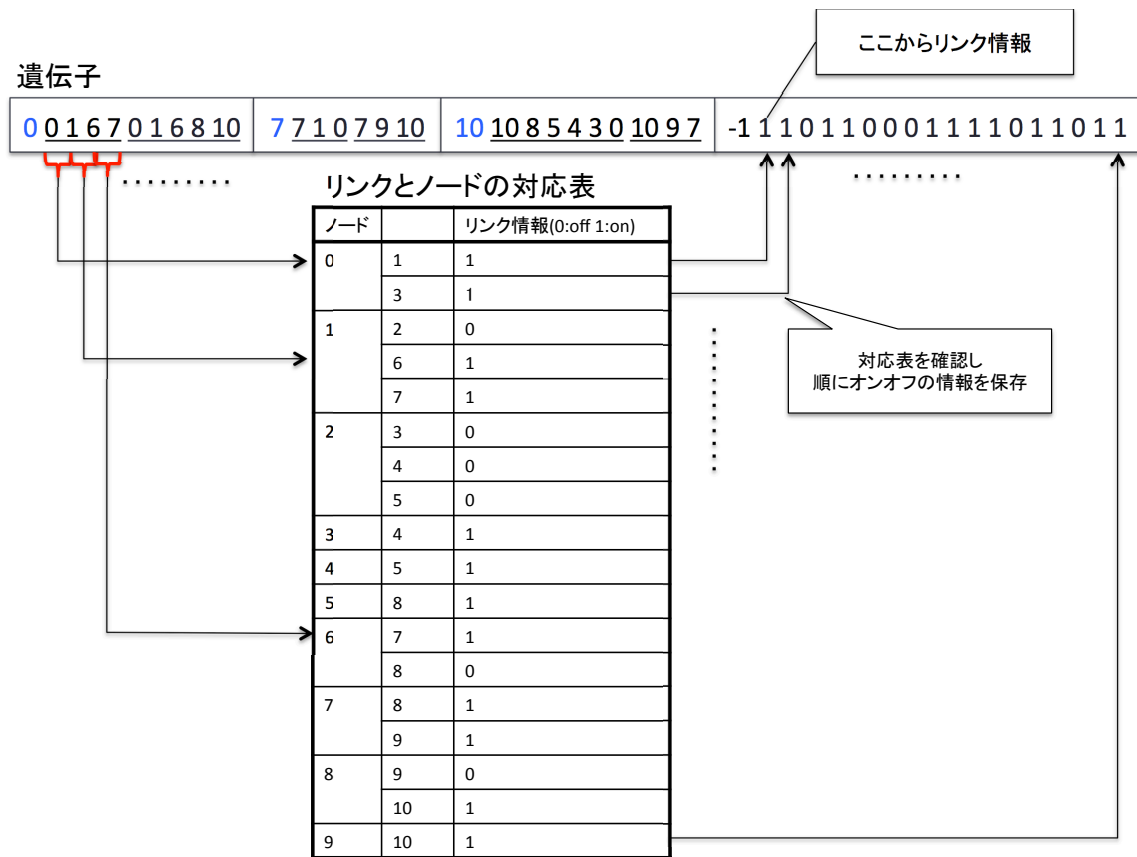


図 8: リンク情報の遺伝子座生成

立した経路が多く確保できるほど信頼性が高いと考えられる．そこで，本報告では，信頼性を以下の指標 $R(x)$ で評価する．

$$R(x) = \frac{1}{\min_{i,j} r_{i,j}(x) + \alpha \sum_{i,j} r_{i,j}(x)}$$

ただし， $r_{i,j}$ は機器 $i-j$ 間の独立な経路の数であり， α は重み変数である．本指標は，全機器間により多くの独立な経路を確保できるほど，小さな値となる．

$r_{i,j}$ は，以下の手順で求めることができる．

1. 遺伝子に含まれる電源を投入するリンクのみで構成されたトポロジ G を構築する
2. $r_{i,j} = 0$ とする
3. トポロジ G 上の経路をダイクストラ法で計算する
4. 手順 3 で経路が見つからなかった場合は，手順を終了する

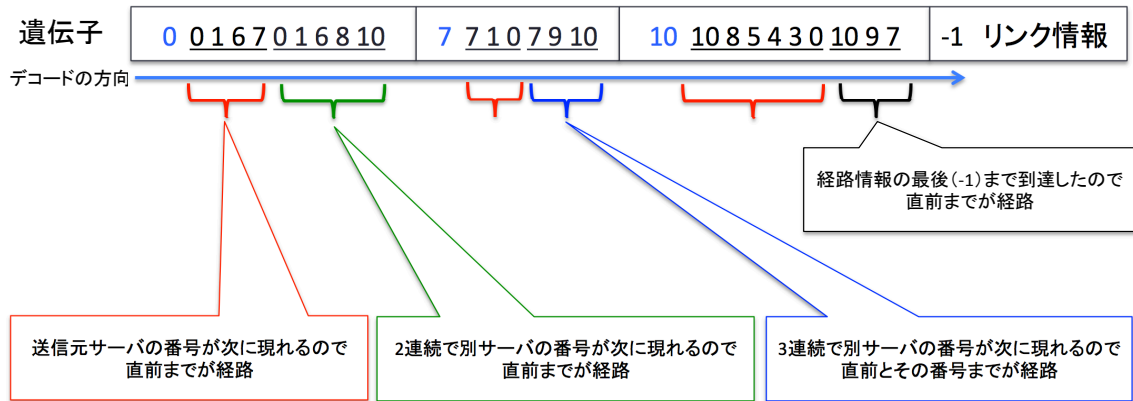


図 9: 経路情報のデコード

5. トポロジ G から手順 3 で計算された経路上のリンクを削除し, $r_{i,j}$ に 1 を加え, 手順 3 に戻る

性能 本報告では, 機器間の通信にかかる遅延を求め, その最大値を抑えることを目標とする. つまり, 性能指標として以下の式で定義される $P(x)$ を用いる.

$$P(x) = \max_{i,j} D_{i,j}(x)$$

ただし, $D_{i,j}(x)$ は遺伝子 x の経路が設定されたときの機器 $i-j$ 間の遅延である. $D_{i,j}(x)$ は以下の式で与えられる.

$$D_{i,j}(x) = \sum_{(s,d) \in p_{i,j}(x)} d_{(s,d)}(x)$$

ただし, $p_{i,j}(x)$ は遺伝子 x の経路設定における $i-j$ 間の経路に含まれるリンクの集合, $d_{(s,d)}(x)$ はリンク $s-d$ において発生する遅延である. 本報告では, 遅延は M/M/1 モデルにおいて与えられるものとした. つまり,

$$d_{(s,d)}(x) = T_s \frac{\rho_{s,d}(x)}{1 - \rho_{s,d}(x)}$$

ただし, T_s は各リンクにおけるパケットの平均処理時間であり, $\rho_{s,d}(x)$ はリンク $s-d$ の使用率である.

4.1.3 選択操作

交叉を行うための親個体を選出する. 本報告では, 親個体の候補としてパレートアーカイブから混雑距離が大きい個体を優先して選択することで解の多様性を維持する.



図 10: リンク情報のデコード

本報告では、親個体の選択は、混雑距離を評価値としたトーナメント選択により行う。トーナメント選択では、個体群の中から一定数の個体をランダムに選択し、選択された個体のうち、評価値がもっともよいものを選択する。この手順を2回行うことで、親個体を選択する。

4.1.4 交叉

交叉では、選択された2つの親の間で遺伝子情報を一部入れ替え、子となる新しい個体を2つ生成する。本報告では、交叉は、機器間の経路単位で行う。つまり、図11に示すように、交叉率に従い、いずれの機器間の経路を入れ替えるかを定める。そして、各親の決められた機器間の経路を入れ替えた子を生成する。

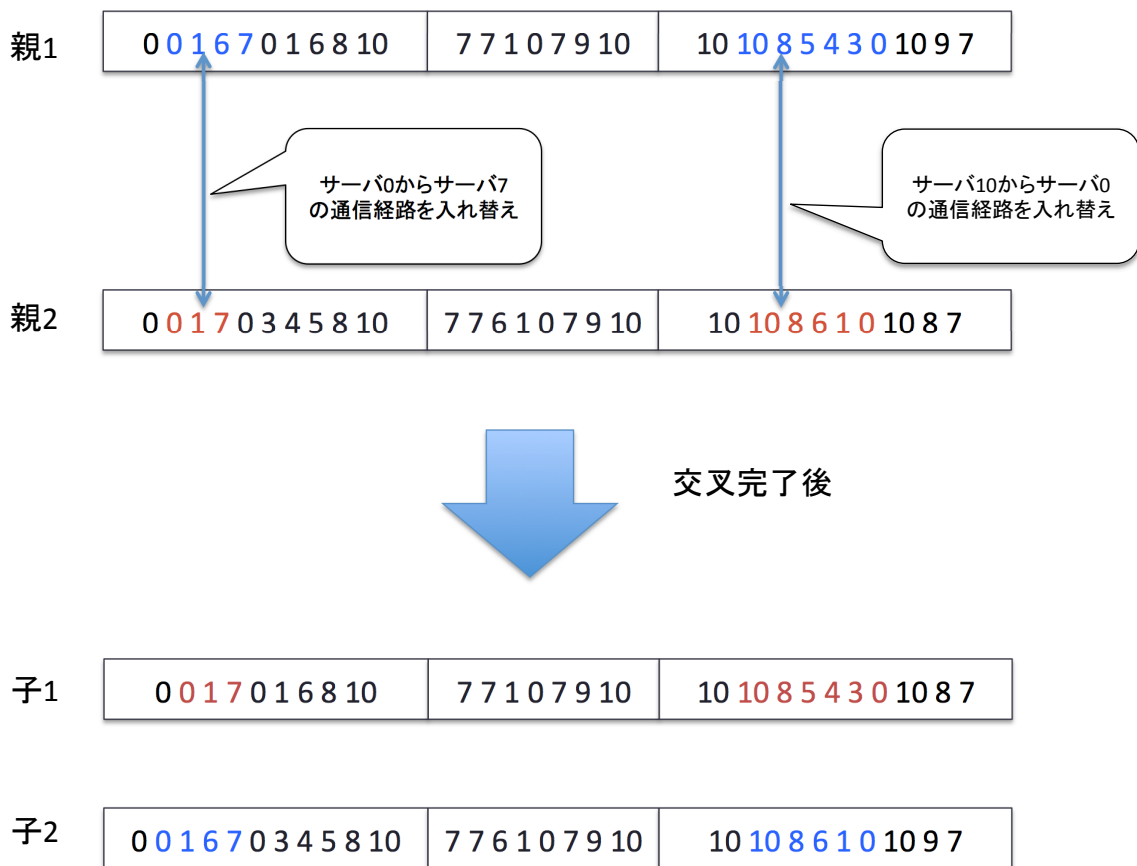


図 11: 交叉

4.1.5 突然変異

突然変異では生成された子に対して遺伝子情報の一部を変化させる。これにより交叉だけでは得られない多様な解を得ることができる。突然変異は交叉後の個体に対して行う。

本報告の手法においては、経路情報に関する突然変異と、電源投入をするリンクの情報に関する突然変異の2種類の突然変異を発生させる。

図 12 に経路情報に関する突然変異の例を示す。経路情報に関する突然変異では、まず、突然変異を発生させる機器間の組をランダムに選択する。そして、選択された機器間の経路について変異を発生させる。変異を発生させる経路では、現在の経路上のノードから、突然変異の始点となるノードをランダムに選択する。そして、選択されたノードから、宛先ノードに到達するまで、次ノードとして隣接ノードをランダムに選択し、当該経路上のノードとして加える。ただし、図 13 に示すように、ループが発生した場合は、当該ループを削除する。経路変更による突然変異が発生した場合は、当該経路に含まれるリンクの電源が投入される

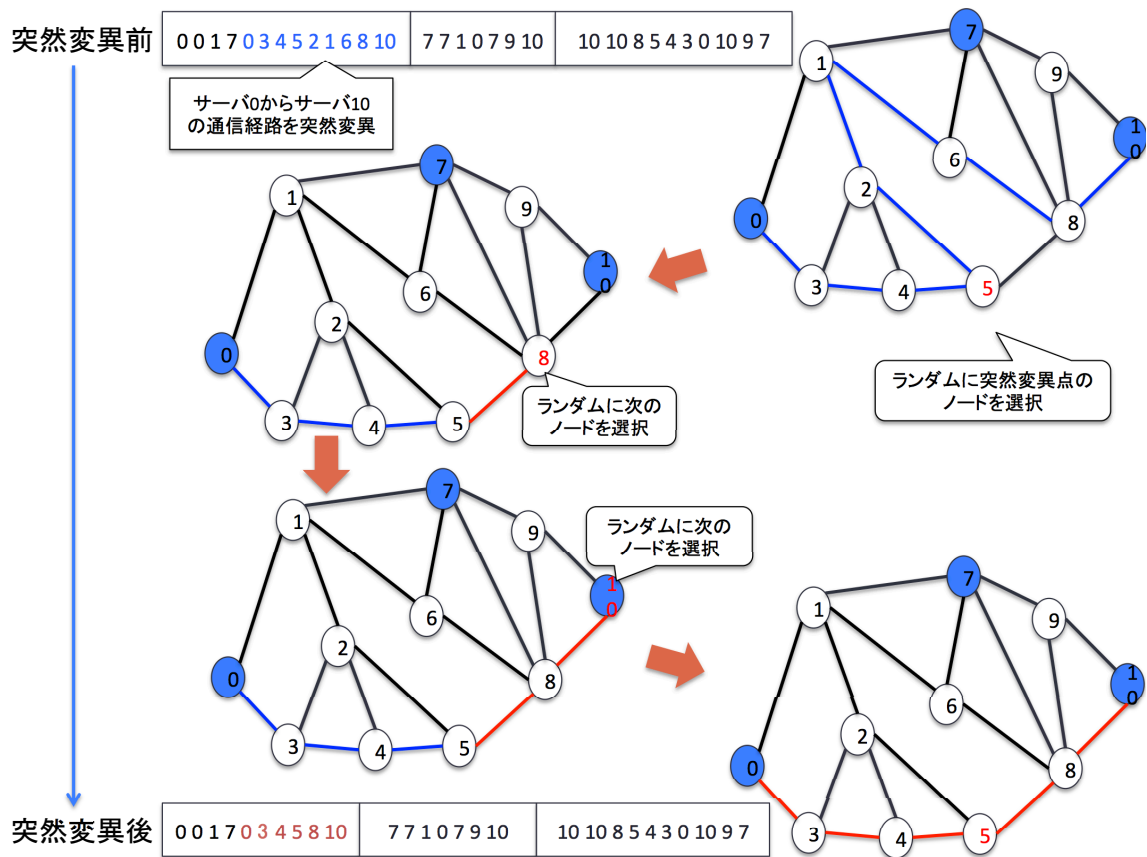


図 12: 経路情報に関する突然変異

ように、電源投入をするリンクに関する情報を修正する。

それに対して、リンク情報の突然変異においては、ランダムにノードを選択し、当該ノードの電源を落とすような変異を加える。これにより、消費電力を削減することができる解を生成する。リンク情報の突然変異は、現在の遺伝子において電源が投入されているノードから、ランダムにノードを選択する。そして、選択されたノードに接続するリンクに対応する遺伝子内の要素を0にする。その後、遺伝子中の経路のうち、電源をオフとするリンクを含んでいる経路については、初期個体生成時と同様にランダムに隣接ノードを選択することを繰り返すことにより、経路を生成する。その際に、電源をオフとなるリンクを経由しないようにすることにより、選択されたノードの電源をオフとした個体を生成する。選択されたノードの電源をオフにすることにより、機器間の接続性が失われ、経路が求まらないことも考えられる。その場合は、当該個体は死滅するものとする。

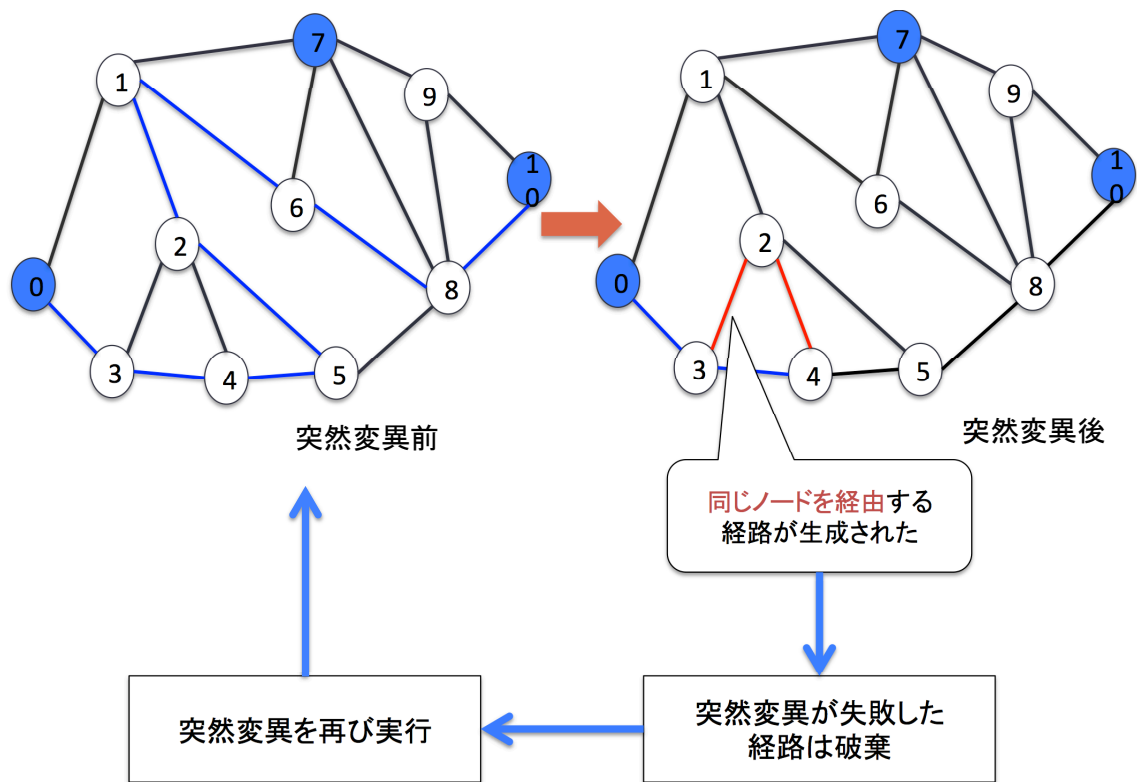


図 13: 突然変異失敗

4.1.6 ネットワークへの解の投入

進化計算により、パレートフロント解が導出された後、パレートフロント解のうち一つの解を選択し、ネットワークに投入する。本報告では、図 14 に示されるように、ネットワーク管理者が定めた制約条件を満たした解のうち、消費電力が最も低い解を用いる。本報告では、遺伝子は各サーバ間の経路情報とリンク情報を含む。そのため、まず、電源を投入する必要があるリンクの情報をもとに、電源を投入する機器、ノードを確定し、必要な機器の電源を投入する。その後、遺伝子に含まれている経路情報に従って、ネットワーク内の経路を設定する。

4.2 環境変動への対応

トラフィック変動などの環境変動が生じた場合は評価値が変化し、パレートフロントも変化する。そのため、前の時刻のパレートフロントを制御に用いることはできない。そのため、パレート最適化制御においては環境変動に合わせてパレートフロントの再探索が必要となる。しかしながら、直接パレートフロントを求めるために全ての解空間を探索するには膨大

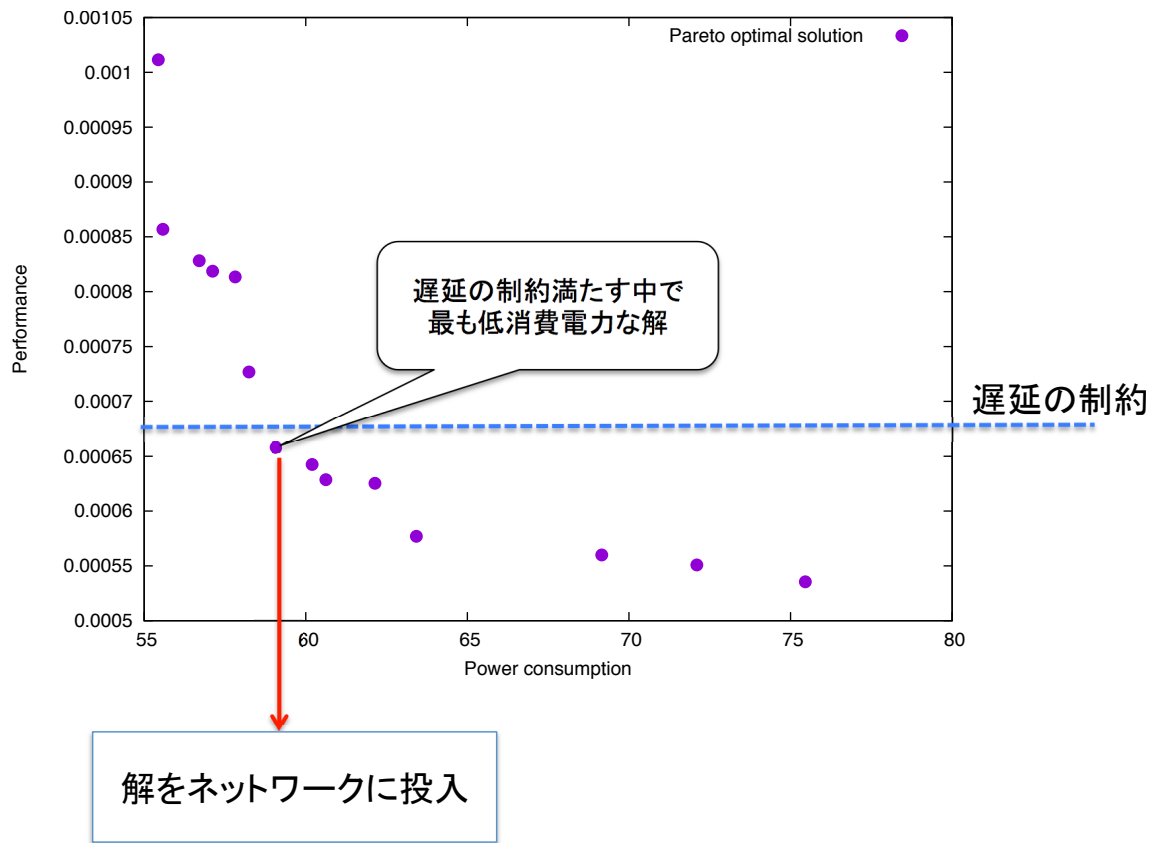


図 14: ネットワークへの解の投入

な計算時間を要する．そこで本報告では，2つのアーカイブをもとに，各時刻の初期解を生成し，当該初期解から探索を行うことで，短時間で適切な解を得る手法を提案する．

4.2.1 前の時刻のパレートアーカイブの利用

進化計算の実行により最終的に得られたパレートフロントは，パレートアーカイブに保存される．このアーカイブに保存された前の時刻のフロントの個体を進化計算の初期値として利用する．これにより，環境変動が緩やかで，現在のパレートフロントが前の時刻のパレートフロントに近い場合には，短時間で現在のパレートフロントを求めることができる．

4.2.2 多様性維持のための新たな解アーカイブの導入

大量の通信を行う機器間の組み合わせが変わるといった大きな環境変動が生じた際には，環境変動後のパレートフロントは，環境変動前のパレートフロントとは大きくかけ離れたも

のとなる．この場合，前の時刻のパレートフロントを初期解とした進化計算では，適切なパレートフロント解が求まるまで時間を要する．そこで，本報告では，前の時刻のパレートフロント解に加えて，様々な環境変動に対応可能だと考えられる多様な解を保存したアーカイブを準備する．多様な解を準備することにより，このアーカイブには，環境変動発生時にパレートフロントに近い解も含まれている可能性が高くなる．その結果，環境変動発生時にも短時間で適切な解を発見することが可能となる．

多様性維持のためのアーカイブは，パレートフロントを求める進化計算の各世代について，以下の処理を加えることにより，構築される．

1. パレートフロントを求めるための進化計算を1世代分行う
2. 現在の世代の解の集合 P と，前の時刻に保存された多様性維持のためのアーカイブ Q の和集合 $P \cup Q$ の各要素について，評価を行う
3. 評価値が小さいものから順に解アーカイブ Q に保存

上記の手順において，解の評価には，パレートフロントからの距離と，パレートフロント解，解アーカイブ内のほかの解との類似度をもとに，以下のように導出された基準値 $Ev(x)$ を用いる．

$$Ev(x) = \text{Distance}(x) \times \text{Sim}(x)$$

ただし， $\text{Distance}(x)$ はパレートフロント解からの目的関数の差， $\text{Sim}(x)$ はパレートフロント解やすでに保存済みの解アーカイブ上の解との類似度を表し，類似しているほど値が大きくなる関数である．この値が小さいものを選ぶことにより，現在のパレートフロント解に近く，信頼性と消費電力の削減の効果が高いと考える解を優先しつつ，他の解と類似していない，多様な解を解アーカイブに保存することが可能となる．

本報告では，パレートフロント解と解 x の距離 $\text{Distance}(x)$ は， x からもっとも近いパレートフロント解との距離とする．つまり，以下のように定義される．

$$\text{Distance}(x) = \min_{x' \in R_1} \sum_m (F_m(x) - F_m(x'))^2$$

また，本報告では，パレートフロント解やすでに保存済みの解アーカイブ上の解との類似度 $\text{Sim}(x)$ は，各通信の経路上の最大リンク使用率を基準として用いる． $l_{i,j}(x)$ を遺伝子 x における i - j 間の通信が経由するリンクのうち，もっとも使用率の高いリンクの使用率とする．そして，リンク使用率をもととし，遺伝子 x と x' の類似度 $S(x, x')$ を以下のように定義する．

$$S(x, x') = \frac{1}{\sqrt{\sum_{i,j} (l_{i,j}(x) - l_{i,j}(x'))^2}}$$

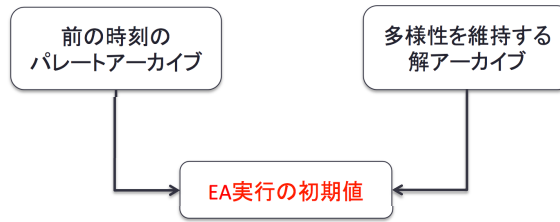


図 15: 環境変動が起きた際の初期値の設定

これにより，同一の通信が混雑するリンクを経由する場合は， $S(x, x')$ は大きな値，混雑するリンクを経由する通信が x と x' で異なる場合は， $S(x, x')$ は小さな値となる．

ここで，パレートフロント解やすでに保存済みの解の和集合 $R_1 \cup Q$ と解 x の類似度 $S(x)$ は， $x' \in R_1 \cup Q$ のうち，もっとも x と類似したものの類似度とする．

$$S(x) = \max_{x' \in R_1 \cup Q} S(x, x')$$

本報告では，類似度 $\text{Sim}(x)$ として， $\text{Sim}(x)$ の値が 0 から 1 の範囲に収まるように，以下のように正規化を行ったものとして与える．

$$\text{Sim}(x) = \frac{S(x) - \min_x S(x)}{\max_x S(x) - \min_x S(x)}$$

4.2.3 アーカイブを利用した多目的進化計算

提案手法では，図 15 に示すように，各時刻における初期解を，パレートアーカイブと多様性のための解アーカイブから以下の手順により，導出する．

1. パレートアーカイブと多様性のための解アーカイブの和集合 $P \cup Q$ に対して現在の環境での評価値を得る
2. 非優越ソートを実行し，個体のランク付けを行う
3. ランクの高い解から優先して個体数 N までを初期値とする

そして，ここで得られた解を初期値として再び 4.2 節の進化計算を実行する．これにより，前の時刻からの環境変動が小さい場合は，パレートアーカイブを初期解とすることで，適切な解を短い時間で得ることができ，環境変動が大きな場合も，多様性のための解アーカイブの解の中から，現在のパレートフロントに近い解を優先的に残した初期解を構築することができ，適切な解を短い時間で得ることができると考えられる．

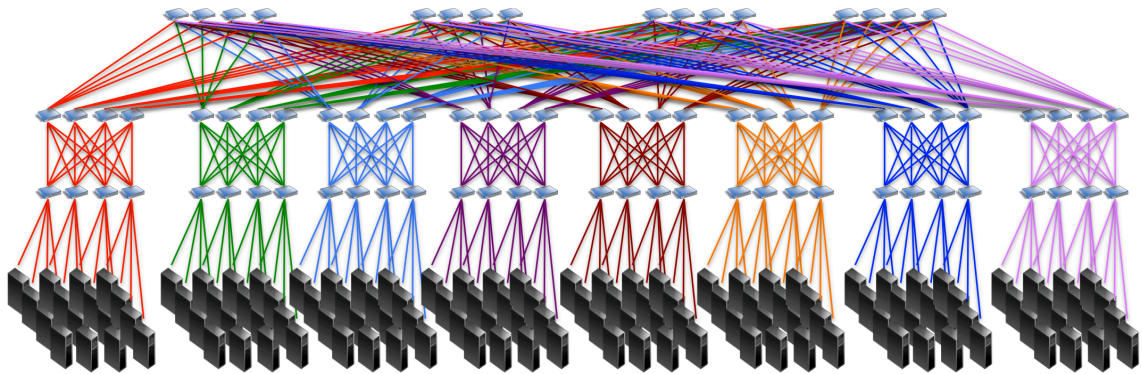


図 16: FatTree 型トポロジ

表 1: ネットワーク機器

ノードの消費電力	0.7 [kw]
リンク 1 本あたりの消費電力	0.07 [kw]
1 ポートあたりの最大伝送速度	1 [Gbps]

5 評価

5.1 評価環境

5.1.1 ネットワークトポロジ

本評価で用いたネットワーク構造を図 16 に示す．本評価では，典型的なデータセンター内ネットワーク構造である FatTree 型トポロジを用いた．本評価では，各スイッチあたりのリンク数を 8 とした FatTree 構造を構築し，Edge ノード 8 個の配下のサーバのみが電源が投入されている状況を想定した．また，ノードの消費電力はリンクの 10 倍程度であることにもとづき [4]，各機器の消費電力，処理能力は，表 1 に示される値を用いた．

5.1.2 トラフィック

本評価では，以下の 2 種類のトラフィックを用いて評価を行う．

パターン A 定常的に通信パターンの変動が起きている状況．総トラフィック量が図 17, 18 のように時間帯により大きく変化する．また，各時刻において，100 個のサーバペア間で通信が発生するものとした．本トラフィックパターンを用いることにより，トラフィックの時間変動へ提案手法が追従できることを示す．

表 2: 進化計算におけるパラメータ

母集団の個体数	交叉率	突然変異率	解アーカイブの上限個体数
30	0.5	0.5	30

パターン B 突発的なトラヒック変動が発生した状況．本トラヒックパターンでは，図 19，20 に示されるように，時刻 501 において，通信を行うサーバペアがすべて入れ替わり，トラヒック量が増える．本トラヒックパターンを用いることにより，急激な変動への提案手法の動作を確認する．

5.1.3 比較手法

本評価では，以下の 3 つの手法を比較する．

Pareto Archive and Solution Archive (PASA) 本報告で提案する，前の時刻のパレートアーカイブと多様性の確保のための解アーカイブをもとに初期解を生成し，進化計算を行うことにより，パレート最適制御を行う手法．

Pareto Archive(PA) 前の時刻のパレートアーカイブを初期解とし，進化計算を行うことにより，パレート最適制御を行う手法．

Random(R) 毎回，ランダムに初期解を生成し，進化計算を行うことにより，パレート最適制御を行う手法．

いずれの手法においても，表 2 のパラメータを用いて進化計算を行った．

本評価では，これらの手法を評価に用いることにより (1) パレート最適制御により，環境変動に追随して消費電力を抑えることができること (2) 提案手法によって初期解を調整することにより，突発的な環境変動にも対応できることを明らかにする．

いずれの手法においても，本評価では，表 2 のパラメータを用い，進化計算を行う．

上記の 3 つの手法は，各時刻において，一定世代の進化計算を行い，新たなパレートフロントを得る．ただし，R は初期解の生成に手順が必要であり，PASA は多様性確保のための解アーカイブを維持するための計算が必要となるため，同一世代数の計算であっても，PA よりも多くの計算時間を要する．そこで，本評価では，各時刻における進化計算の世代数を，

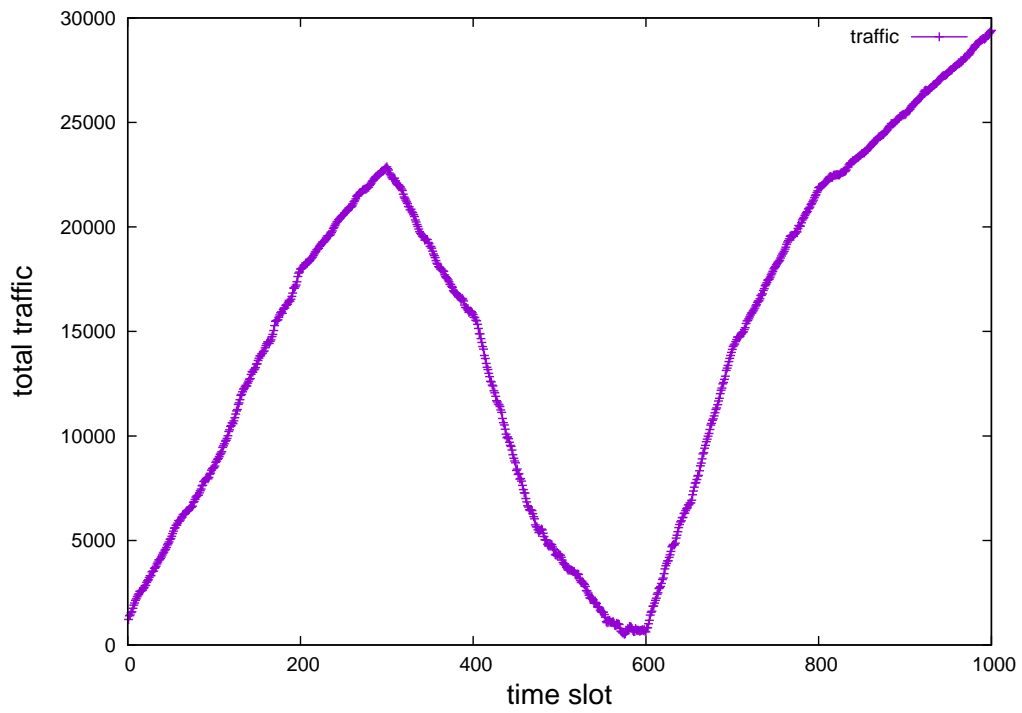


図 17: 総トラフィック変動パターン A

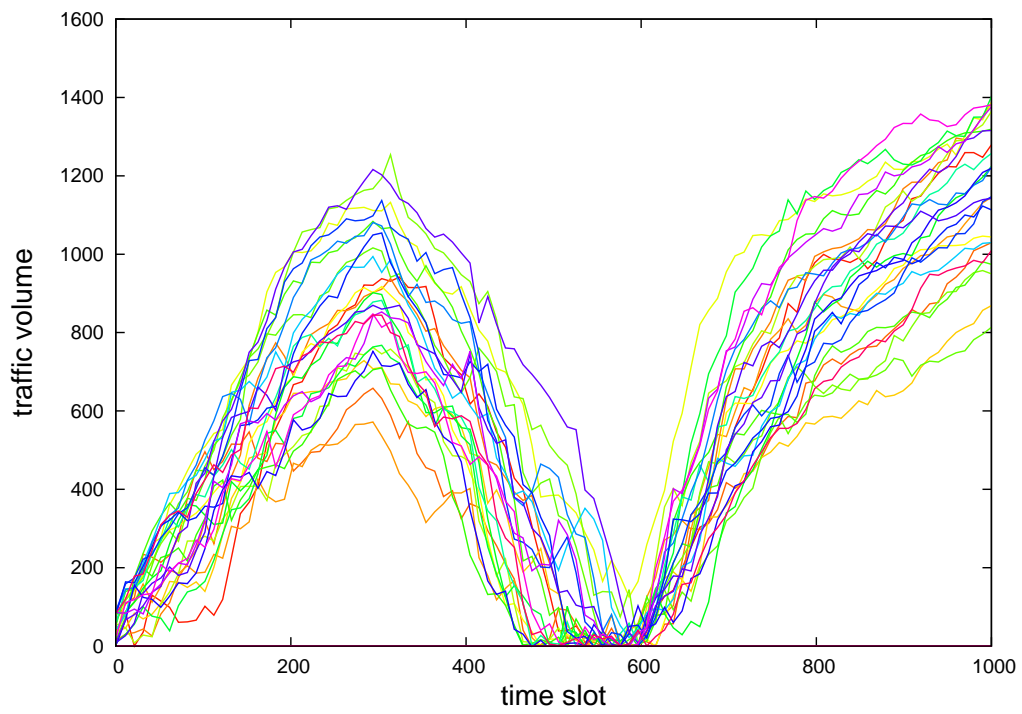


図 18: 各通信におけるトラフィック変動パターン A

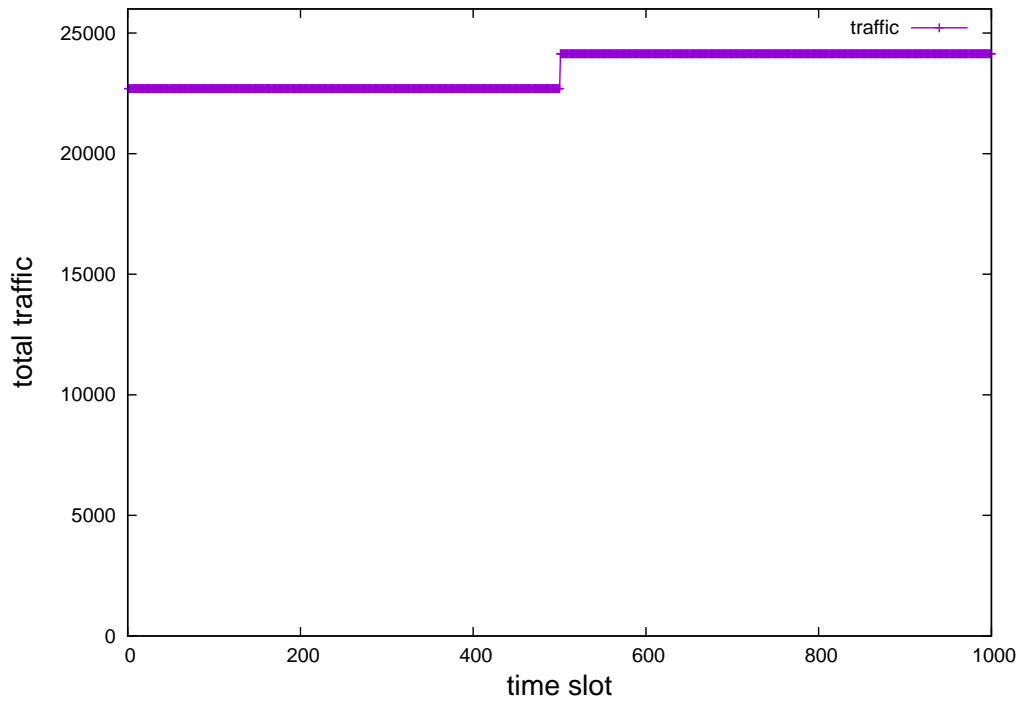


図 19: 総トラフィック変動パターン B

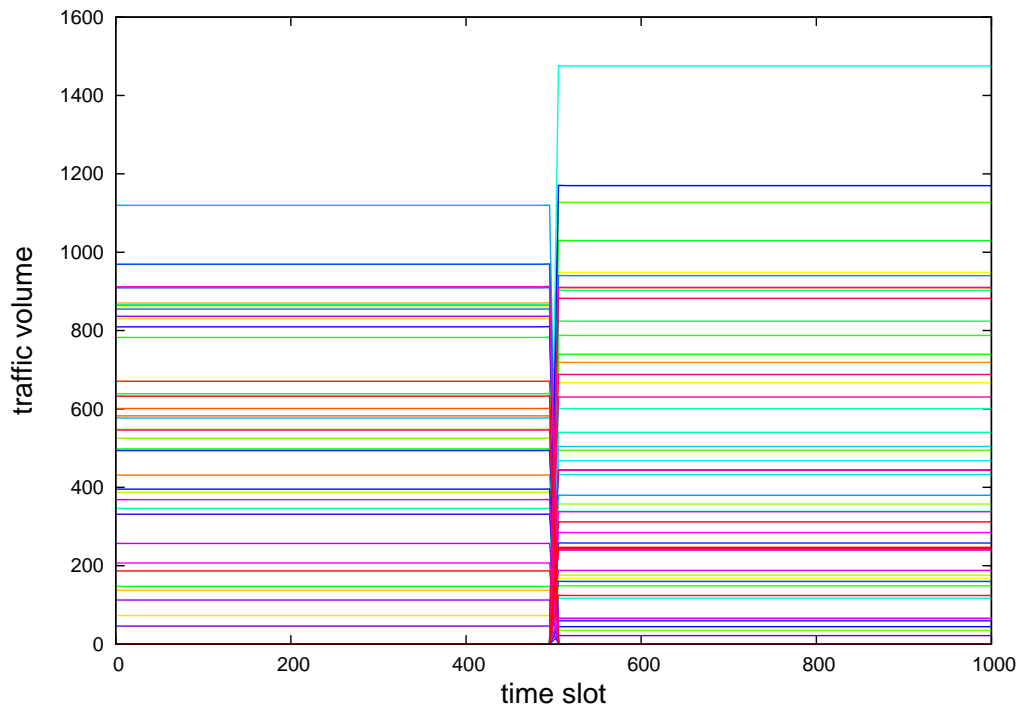


図 20: 各通信におけるトラフィック変動パターン B

表 3: 評価に用いた世代数

	PA の世代数	R の世代数	PASA の世代数
パターン 1	50	42	25
パターン 2	100	83	50

R は PA の 0.83 倍、PASA は PA の半分の世代数となるように設定した。本評価では、表 3 に示す 2 種類の世代数での評価を行った。

5.1.4 SLA

また、制御の際に達成が必要な性能条件として、全機器間の遅延を 0.00025[s] 以下にするという条件を与えた。また、耐故障性に関する制約条件として、全機器間に 2 本以上の独立な経路を確保するという条件も満たすという条件も与えた。

5.2 評価結果

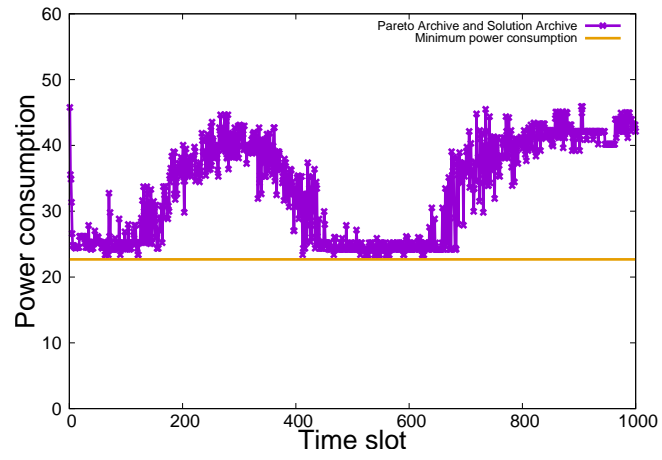
まず、変動パターン A のトラヒックを与えた場合に、各手法で達成可能な消費電力、遅延、耐故障性を確保するために確保された機器間の独立な経路の本数を調べた。

図 21 は、各時刻の進化計算の世代数をパターン 1 に設定した場合の、消費電力を示す。横軸はタイムスロット、縦軸は消費電力を示す。また、図中には、電源が投入されたサーバ間の接続性を維持するのに必要な最小限の機器の電源を投入した場合の消費電力を線で書き入れている。図より、PASA、SA を用いることにより、いずれも、トラヒック量の変動に追従して、トラヒック量の少ない時間帯には、消費電力を抑え、トラヒック量の大きな時間帯には消費電力が高くなっていることが分かる。これは、トラヒック量が少ない時間帯は、少数の機器の電源を投入するだけで、性能要件を満たすことができるのに対して、トラヒックが多い時間帯には、多くの機器の電源を投入し、トラヒックを分散させないと、輻輳が発生し、必要な制約を満たすことができないためである。PASA や PA を用いた場合は、このような性能の確保を行った上で、消費電力を抑えるような経路を発見することができており、トラヒックの変動に追従をしたネットワーク低消費電力化を実現できている。それに対して、初期解をランダムとして与えた R では、いずれの時間帯も同様の消費電力となっている。これは、ランダムな初期解を与えた場合は、42 世代までに、消費電力を削減できるような適切なパレートフロントが見つからないためである。このため、各時刻で短い世代数の進化計算によりパレートフロンを導出しつつ、環境変動に追従した低消費電力化を実現するためには、前の時刻のパレートフロントを用いることが有効であると考えられる。

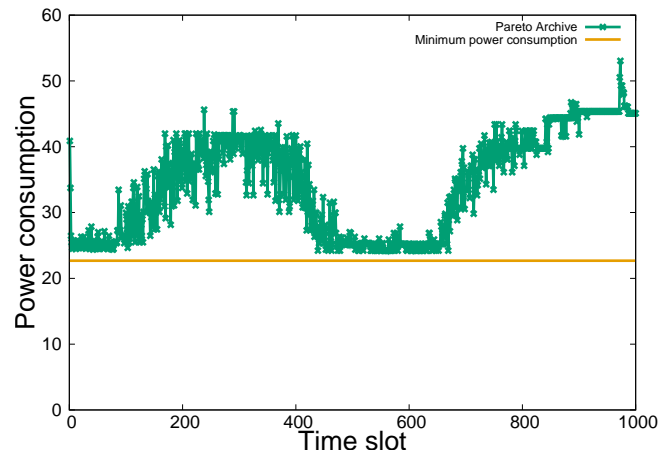
また、本制御において達成された遅延を図 22、独立な経路数を図 23 に示す。図 22 は横軸はタイムスロット、縦軸は機器間の通信遅延の最大値を示し、図 23 は横軸はタイムスロット、縦軸は、 $\min_{i,j} r_{i,j}(x) + \alpha \sum_{i,j} r_{i,j}(x)$ の値を示す。ただし、本評価においては、 α は $\frac{1}{\max_x \sum_{i,j} r_{i,j}(x)}$ として与えた。いずれの図も、本評価において与えた性能要件に対応する値に線を引いた。図より、いずれの時刻において、いずれの手法も性能、耐故障の要件を満たす解を投入することができている。これは、いずれの手法で求めたパレートフロントにおいても、性能、耐故障の要件を満たす解が存在し、本手法では、要件を満たす解からネットワーク設定を投入する解を選択するためである。

同様の評価を、各手法が各時刻において計算する進化世代数の設定をパターン 2 とした場合の結果を図 24、図 25、図 26 に示す。本結果より、進化させる世代数が増えたとしても、上述の考察が成り立つことが分かる。

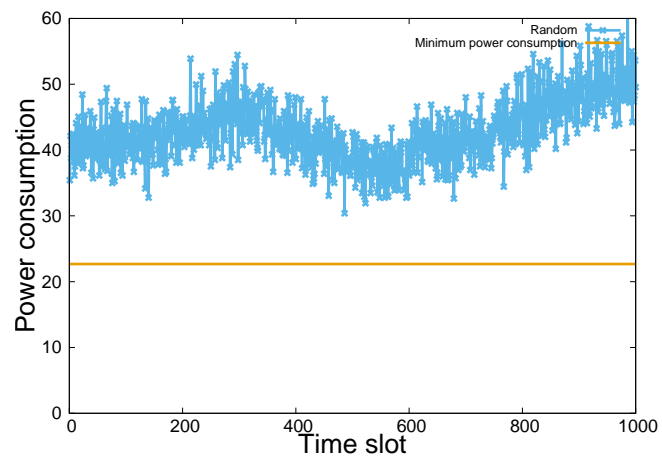
次に、トラヒックパターン B を用いて、トラヒックパターンが急激に変化した状況下において、パレート最適制御にもとづく低消費電力化手法の動作について調べる。図 27 に各時刻の消費電力、図 28 にサーバ間の遅延、図 29 に独立な経路数を示す。図より、PASA、PA とともに時刻 501 において、消費電力化が高くなっていることが分かる。これは、時刻 501 において発生した突発的な環境変動に起因するものであり、現在保持している解のうち、環境変動後においても、性能や耐故障の要件を満たす解は少ないためである。その結果、消費電力が高くなっても、性能や耐故障の要件を満たす解を投入せざるをえず、消費電力化が高くなっている。しかしながら、PASA と PA を比較すると、PASA の消費電力の増大は、PA と比べて小さいことが分かる。さらに、図 28 より、PA は遅延に対する要件を満たすことができない時間帯が存在することが分かる。これは、PA が前の時刻のパレートフロントを初期解として用いているが、トラヒックが急激に変化し、パレートアーカイブに含まれるいずれの解も真のパレートフロントから大きくかけ離れたものとなってしまったことが原因である。その結果、適切なパレートフロントを導出するまでに世代数を要し、環境変動直後の制御では、適切な解を持つパレートフロントまで到達できない。それに対して、PASA はパレートフロントに加え、多様性を維持するためのアーカイブも保持している。この多様性を維持するアーカイブから探索を開始することにより、適切なパレートフロントを短時間で発見できる。つまり、パレートフロントとは別途保存させた多様な解を初期値として用いることにより、急激な環境変動に対しても対応可能であることが分かった。



(a) PASA

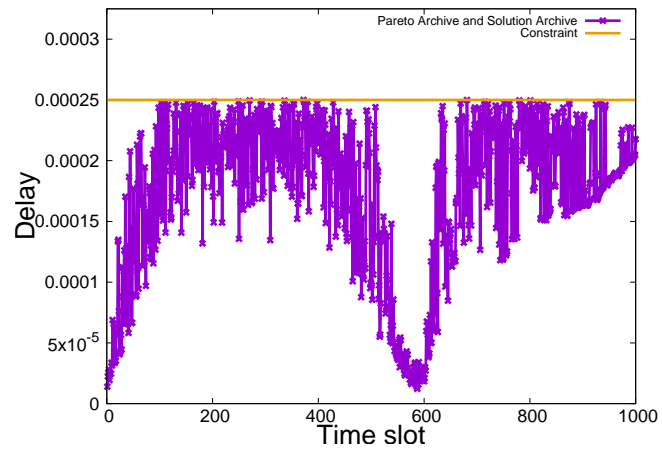


(b) PA

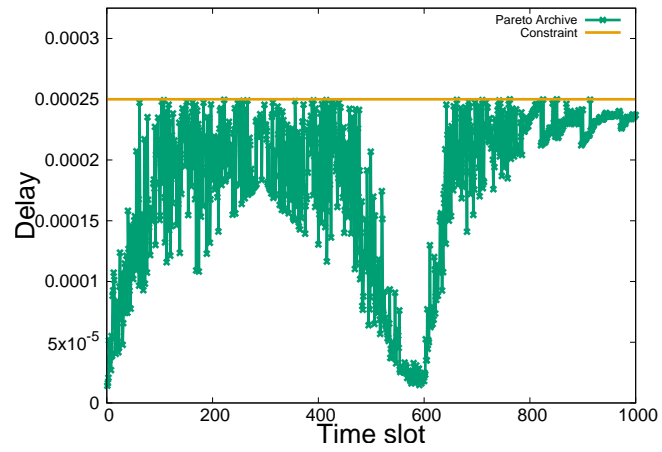


(c) R

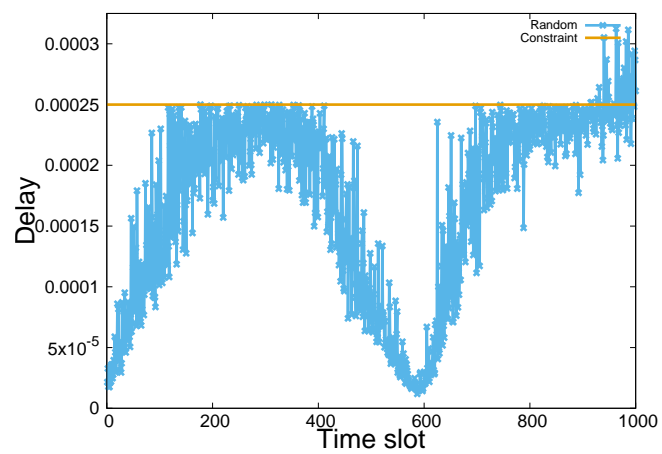
図 21: 消費電力 (トラヒックパターン = A , 各時刻の世代数の設定 = パターン 1)



(a) PASA

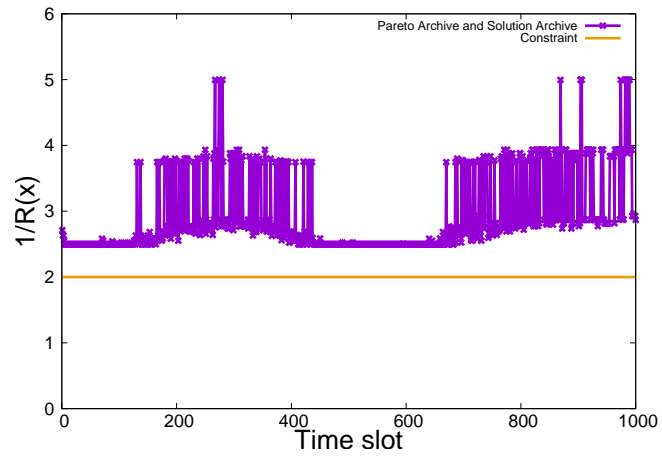


(b) PA

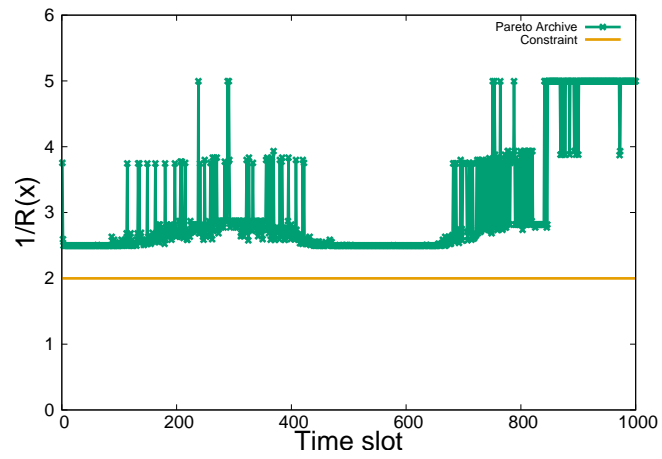


(c) R

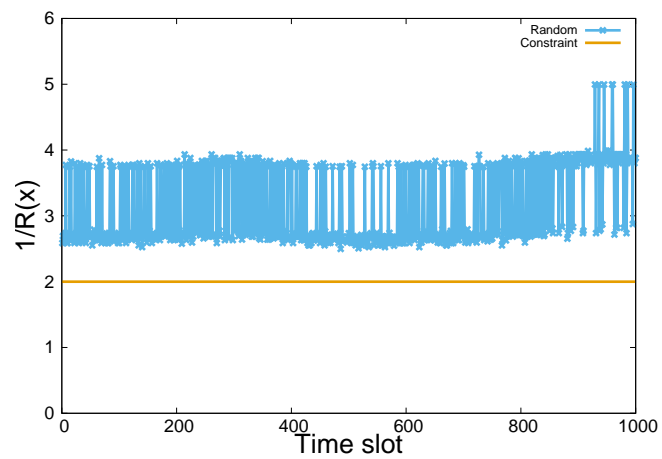
図 22: 性能 (トラヒックパターン = A, 各時刻の世代数の設定 = パターン 1)



(a) PASA

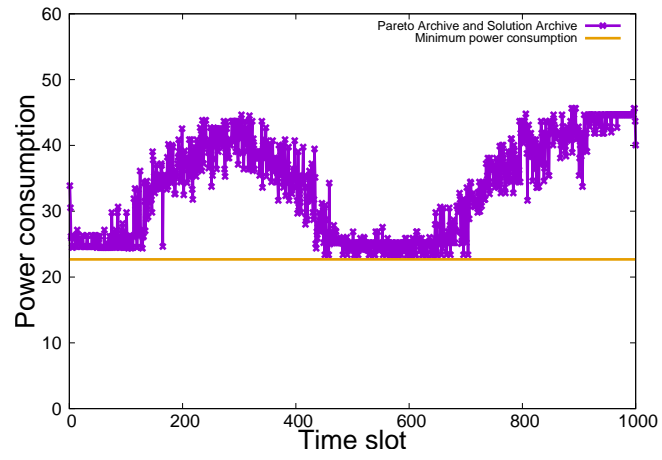


(b) PA

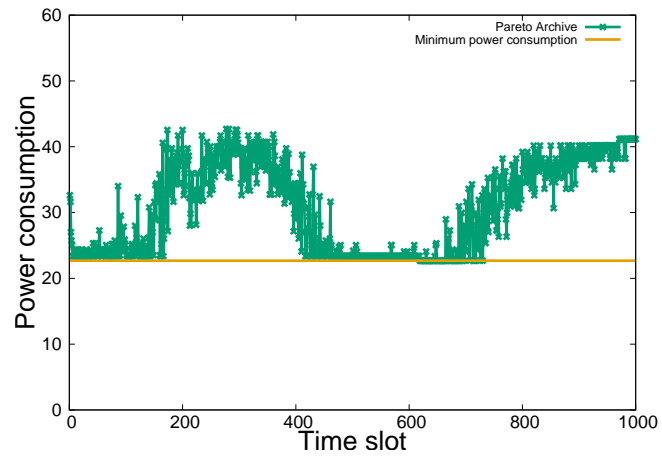


(c) R

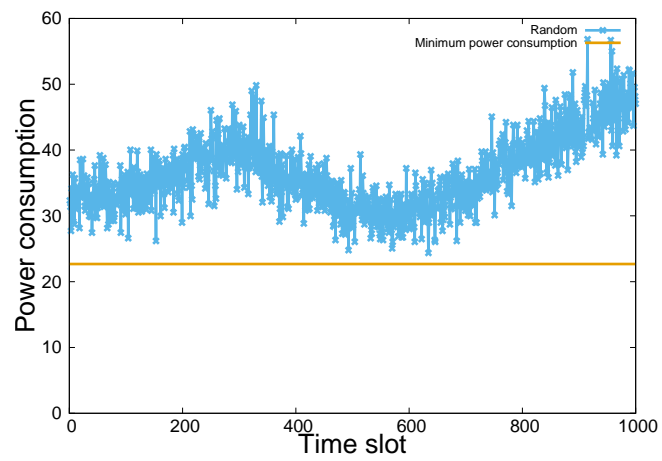
図 23: 耐故障性 (トラヒックパターン = A, 各時刻の世代数の設定 = パターン 1)



(a) PASA

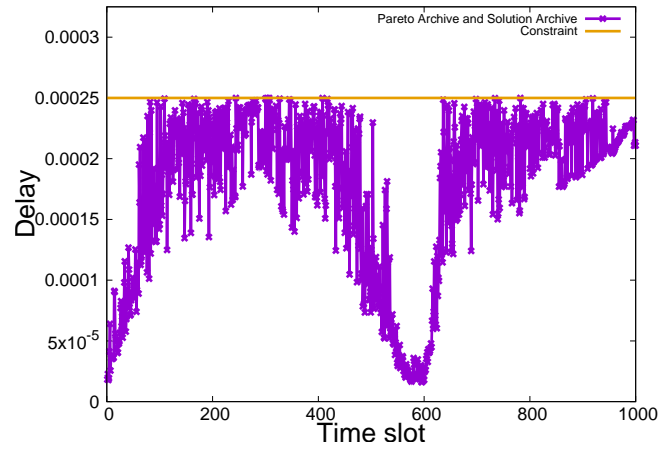


(b) PA

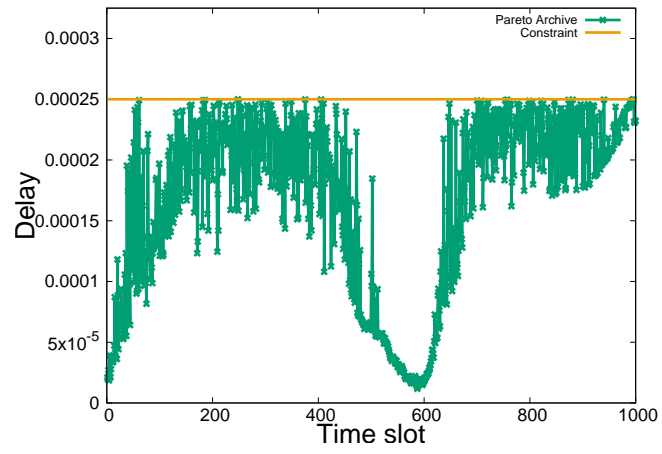


(c) R

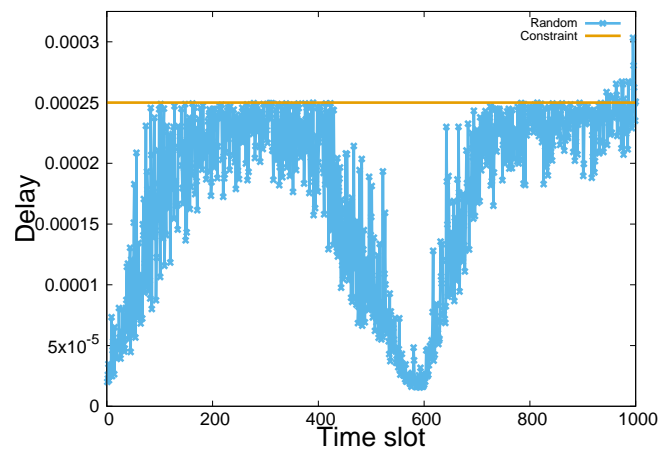
図 24: 消費電力 (トラヒックパターン = A, 各時刻の世代数の設定 = パターン 2)



(a) PASA

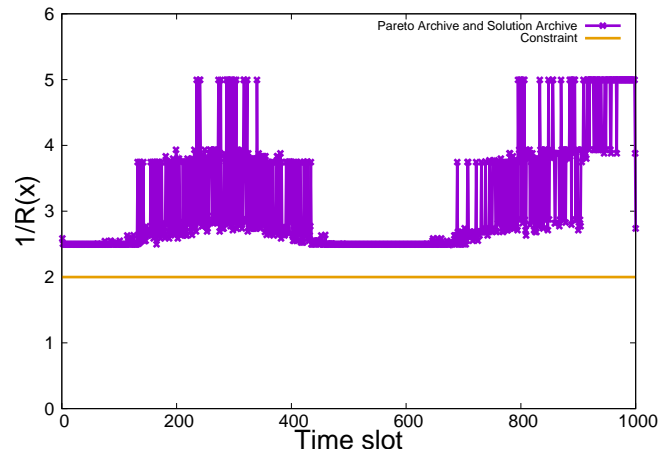


(b) PA

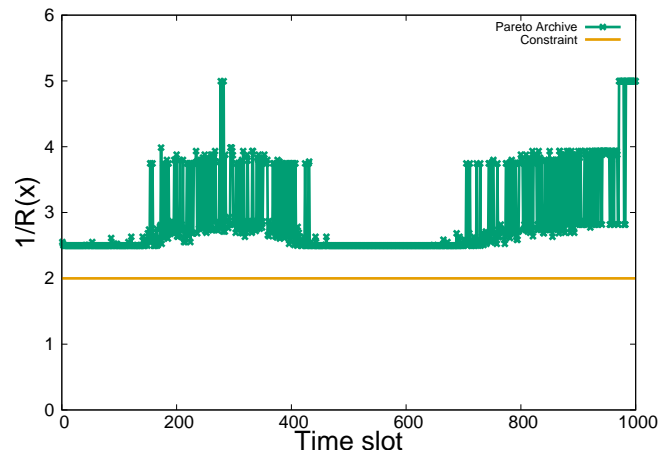


(c) R

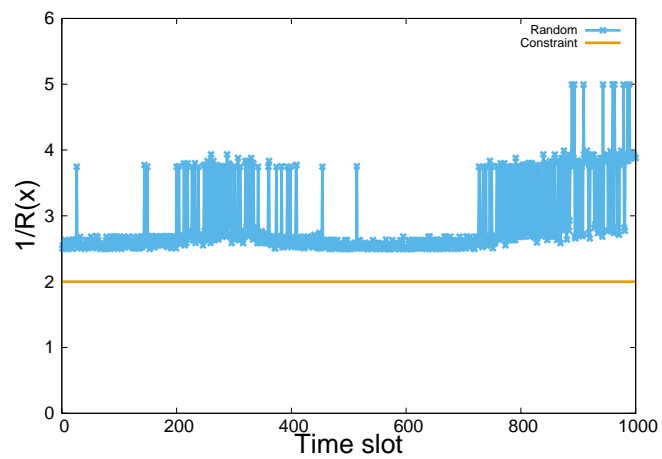
図 25: 性能 (トラヒックパターン = A, 各時刻の世代数の設定 = パターン 2)



(a) PASA

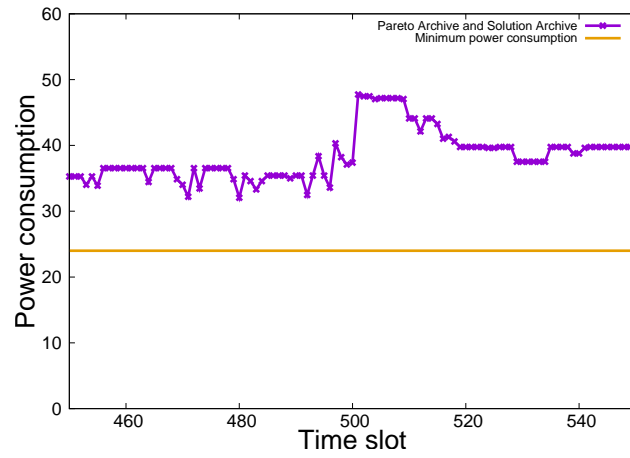


(b) PA

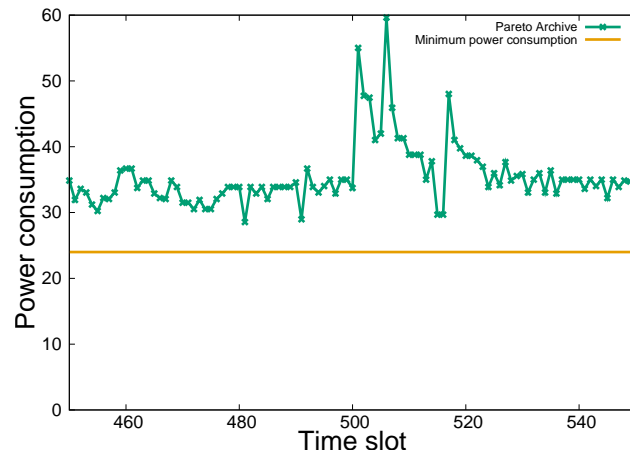


(c) R

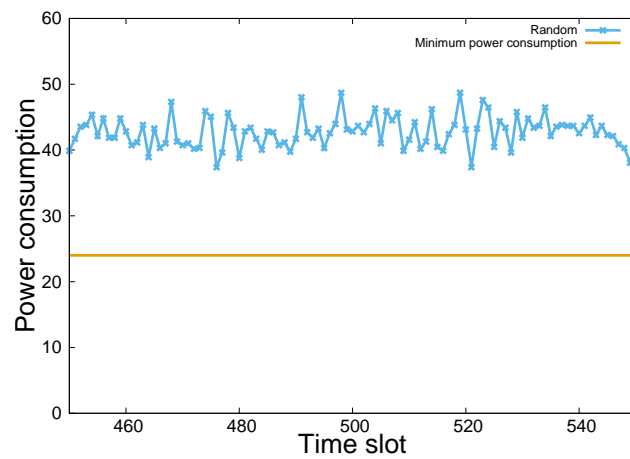
図 26: 耐故障性 (トラヒックパターン = A , 各時刻の世代数の設定 = パターン 2)



(a) PASA

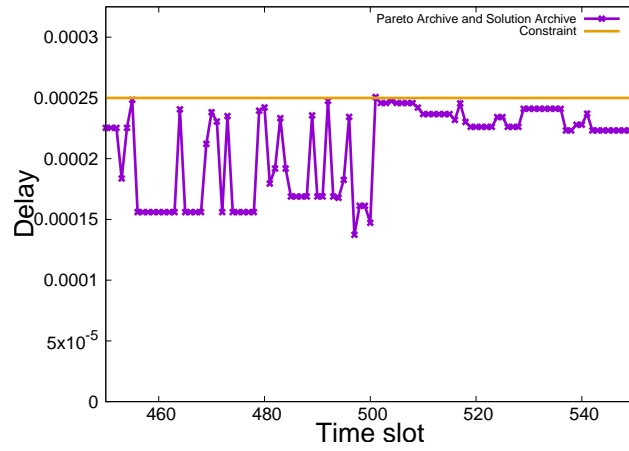


(b) PA

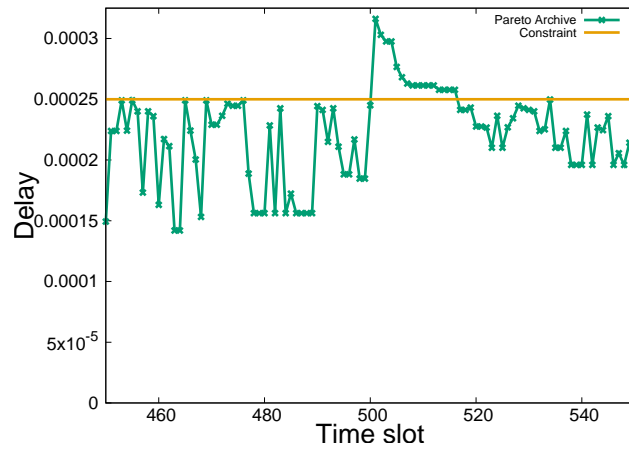


(c) R

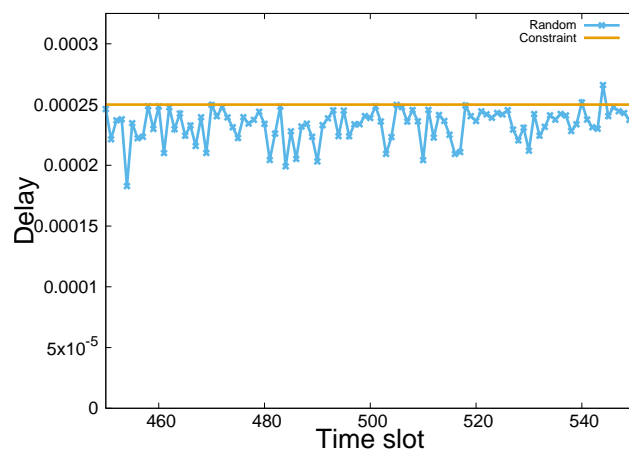
図 27: 消費電力 (トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)



(a) PASA

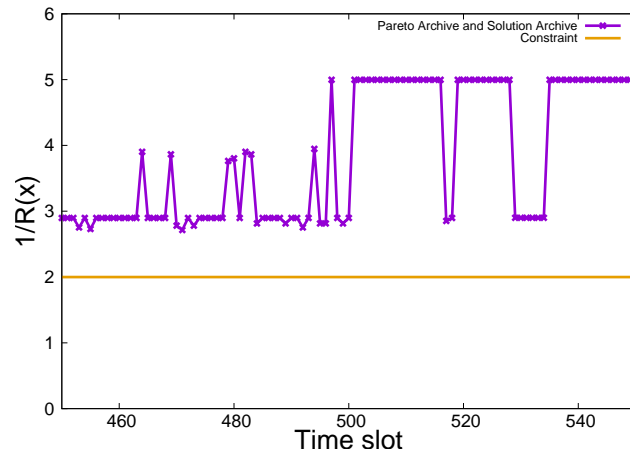


(b) PA

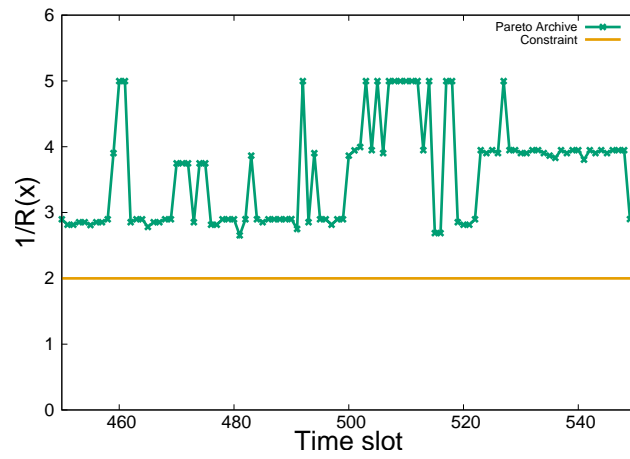


(c) R

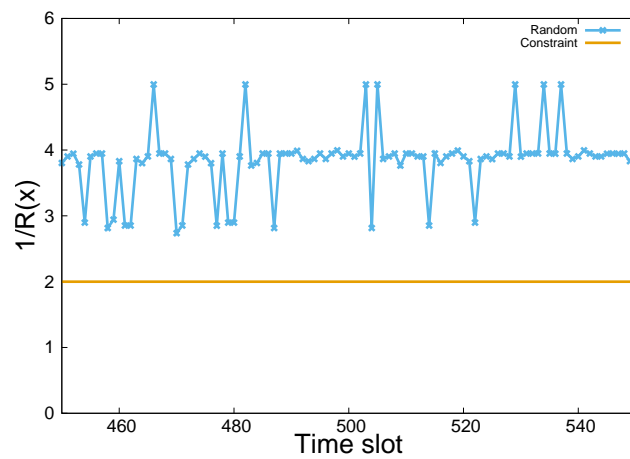
図 28: 性能 (トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)



(a) PASA



(b) PA



(c) R

図 29: 耐故障性 (トラヒックパターン = B, 各時刻の世代数の設定 = パターン 1)

6 おわりに

本報告では、短時間のトラヒック変動や故障などの環境変動に追隨して、十分な通信性能、信頼性の確保と低消費電力化の3つの目的を達成するネットワーク制御手法を確立した。ネットワークの制御を行うにあたり、耐故障性を確保しようとするればオンになるノードやリンクが増加するため消費電力は増加し、性能を確保しようとしても消費電力は増加する。提案手法では、これらの指標をすべて考慮した制御を実現する手法として、パレート最適解の集合（パレートフロント）を求め、そのうち、必要な性能・信頼性の制約を満たす解をネットワークに投入することにより、性能・信頼性の要件を満たす範囲内で、消費電力を最小化する。この制御をネットワーク内の環境変動に追隨して行うためには、パレートフロントの計算も環境変動に追隨して行うことが必要となる。そこで、本報告では、パレートフロントの計算を高速に行う手法を検討した。提案手法では、前の時刻のパレートフロントと、環境変動発生時にパレートフロントとなる可能性の高い解を組み合わせた集合を初期解とし、進化計算により、現在の環境に合わせたパレートフロントを探索する。これにより、少ないステップで適切なパレートフロントにたどり着くことができ、高速なパレートフロントの取得が可能となる。

本報告では、提案手法をシミュレーションにより評価を行い、提案手法を用いることにより、環境変動に追隨し、性能・信頼性の要件を満たしつつ、トラヒックが少ない時間帯であれば、全機器稼働状態の39%までネットワークの消費電力を低減することができることを示した。さらに、前の時刻のパレートフロントのみで初期解を生成する手法では、各ステップにおいて50世代の進化計算を行う場合では、制約条件を満たす解に到達するまで16ステップを要するような環境下においても、前の時刻のパレートフロント、環境変動発生時にパレートフロントとなる可能性の高い解を組み合わせた集合を初期解と設定する手法では、1ステップ以内に制約を満たした解に到達することができることを示した。

今後、より大規模なネットワークでの評価を行うとともに、各時刻における設定変更が必要な機器数の最小化といった他の指標も取り入れた評価を予定している。

謝辞

本報告を終えるにあたり，研究全般に関して広く御指導，御教授を頂きました大阪大学大学院情報科学研究科の村田正幸教授，ならびに研究の方針，本報告の作成に関して平素より様々な面で適切な御指導を頂きました大阪大学大学院情報科学研究科の太下裕一助教に厚く御礼申し上げます．また，研究に関して適切な御助言を多く頂きました，大阪大学大学院情報科学研究科の荒川伸一准教授，大阪大学大学院経済学研究科の小南大智助教に感謝いたします．最後に，日頃より様々な御助言と御助力を頂きました大歳達也氏，須恵匠氏，河島滉太氏，山西宏平氏をはじめとする村田研究室の皆様に御礼申し上げます．

参考文献

- [1] Cisco, “Cisco global cloud index:forecast and methodology, 2014-2019,” tech. rep., Cisco Systems Inc., Oct. 2015.
- [2] Van Heddeghem, Ward and Lambert, Sofie and Lannoo, Bart and Colle, Didier and Pickavet, Mario and Demeester, Piet, “Trends in worldwide ICT electricity consumption from 2007 to 2012,” *Computer Communications*, vol. 50, pp. 64–76, Sep. 2014.
- [3] Lambert, Sofie and Van Heddeghem, Ward and Vereecken, Willem and Lannoo, Bart and Colle, Didier and Pickavet, Mario, “Worldwide electricity consumption of communication networks,” *Optics express*, vol. 20, pp. B513–B524, Dec. 2012.
- [4] Amaldi, E. and Capone, A. and Gianoli, L. G., “Energy-aware IP traffic engineering with shortest path routing,” *Comput. Netw.*, vol. 57, pp. 1503–1517, Apr. 2013.
- [5] L. Chiaraviglio, M. Mellia, and F. Neri, “Minimizing ISP network energy cost: formulation and solutions,” *IEEE/ACM Transactions on Networking (TON)*, vol. 20, pp. 463–476, Apr. 2012.
- [6] Chiaraviglio, Luca and Mellia, Marco and Neri, Fabio, “Reducing power consumption in backbone networks,” in *Proceedings of Communications, 2009. ICC’09. IEEE International Conference on*, pp. 1–6, IEEE, Jun. 2009.
- [7] Giroire, Frédéric and Moulhierac, Joanna and Phan, Truong Khoa and Roudaut, Frédéric, “Minimization of network power consumption with redundancy elimination,” *Computer communications*, vol. 59, pp. 98–105, Mar. 2015.
- [8] Giroire, Frédéric and Moulhierac, Joanna and Phan, Truong Khoa and Roudaut, Frédéric, “Minimization of network power consumption with redundancy elimination,” *Computer communications*, vol. 7298, pp. 247–258, May 2012.
- [9] Kessaci, Yacine and Melab, Nouredine and Talbi, El-Ghazali, “A pareto-based meta-heuristic for scheduling HPC applications on a geographically distributed cloud federation,” *Cluster Computing*, vol. 16, pp. 451–468, Sep. 2013.

- [10] Battiti, Roberto and Passerini, Andrea, “Brain–computer evolutionary multiobjective optimization: a genetic algorithm adapting to the decision maker,” *Evolutionary Computation, IEEE Transactions on*, vol. 14, pp. 671–687, Sep. 2010.
- [11] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: A tutorial,” *Reliability Engineering & System Safety*, vol. 91, pp. 992–1007, Sep. 2006.
- [12] Fadaee, M and Radzi, MAM, “Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: a review,” *Renewable and Sustainable Energy Reviews*, vol. 16, pp. 3364–3369, Jun. 2012.
- [13] Ahmadi, Pouria and Dincer, Ibrahim and Rosen, Marc A, “Thermoeconomic multi-objective optimization of a novel biomass-based integrated energy system,” *Energy*, vol. 68, pp. 958–970, Apr. 2014.
- [14] Q. Wang, M. Guidolin, D. Savic, and Z. Kapelan, “Two-objective design of benchmark problems of a water distribution system via moeas: Towards the best-known approximation of the true pareto front,” *Journal of Water Resources Planning and Management*, vol. 141, p. 04014060, Jul. 2014.
- [15] T. D. Prasad and N.-S. Park, “Multiobjective genetic algorithms for design of water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 130, pp. 73–82, Dec. 2004.
- [16] Deb, Kalyanmoy and Agrawal, Samir and Pratap, Amrit and Meyarivan, Tanaka, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” *Lecture notes in computer science*, vol. 1917, pp. 849–858, Sep. 2000.
- [17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 182–197, Apr. 2002.