

パレート最適制御によるネットワーク省電力化手法

秋下 耀介[†] 大下 裕一[†] 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

E-mail: †{y-akishita,y-ohsita,murata}@ist.osaka-u.ac.jp

あらまし 近年、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おり、不要なネットワークの機器やサーバをスリープさせ、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。しかしながら、従来のネットワーク低消費電力化手法の検討では、信頼性の確保については考慮しておらず、故障が発生した際には、ネットワークの性能が低下する。そこで、本稿では、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。本稿では、すべての指標を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。また本手法では、ネットワーク内の環境変動に追従して行うために、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、高速に適切なパレートフロントの取得が可能となる。本稿では、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、全機器稼働状態の53%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。

キーワード 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

Network power saving based on pareto optimal control

Yosuke AKISHITA[†], Yuichi OHSITA[†], and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University Yamadaoka 1-5, Suita-shi, Osaka, 565-0871 Japan

E-mail: †{y-akishita,y-ohsita,murata}@ist.osaka-u.ac.jp

Abstract The power consumption of networks has been increasing as the service over the Internet becomes popular, and has become a serious problem. Many methods to reduce the power consumption by shutting down unnecessary network devices have been proposed. However, these methods do not consider the fault tolerance, and shutting down some network devices may degrade the performance of the network in case of failures. In this paper, we propose a method to control the network considering multiple objectives; performance, power consumption and fault tolerance. This method calculates the Pareto front considering the multiple objectives. Then it selects one of the solution on the Pareto front. In addition, we also propose a method to calculate the Pareto front following the environmental changes. This method calculates the Pareto front by evolving the solutions saved in the archive. By saving the solutions that may become the Pareto optimal solution in the archive, this approach can obtain the Pareto front immediately. In this paper, we evaluate the method by simulation. The results show that the method can reduce power consumption of the network by 53% following the environmental change.

Key words Network power saving , Multi-objective optimization , Pareto optimal solutions , Pareto front , Evolutionary algorithms

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フィックは増加し続けている [1]。トラフィック増加に伴い、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] .

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 [3], [4] . 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ヘトラヒックを集約し、ノードやリンクをスリープさせることにより、冗長性が除去される。そのため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。

そこで本稿では、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。本稿では、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 [5] 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。本稿では、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、本稿では、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。

本稿の構成は以下の通りである。まず第2章では、パレートフロントを求める手法について述べる。次に第3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4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、第5章で本稿のまとめと今後の課題について述べる。

2. パレートフロント

2.1 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1)。式 (1) と (2)

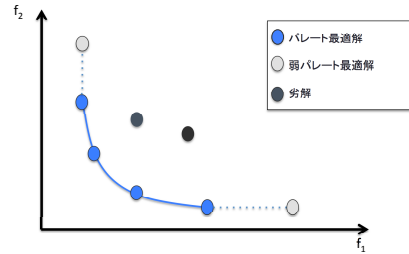


図1 パレート最適解

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、パレトフロントと呼ぶ。

2.2 パレトフロントの導出

パレ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 [6] . パレ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。進化計算では、各個体に該当する解を遺伝子としてコーディングする。以下にアルゴリズム全体の流れを示す。

(1) 初期化: N 個の個体群 P の生成。

(2) 評価: 個体群 P の評価、非優越ソートによる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計算。

(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。 P とは別に、ランク1の個体をパレトアーカイブと呼ばれる領域に保存。

(5) 終了判定: 終了条件を満たした場合終了。この際、パレトアーカイブに保存されていた解が求まるパレトフロントとなる。終了でない場合は3へ。

2.2.1 非優越ソート

上記の手順では、個体のランク付けに非優越ソート [6] を用いる。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。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。

(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

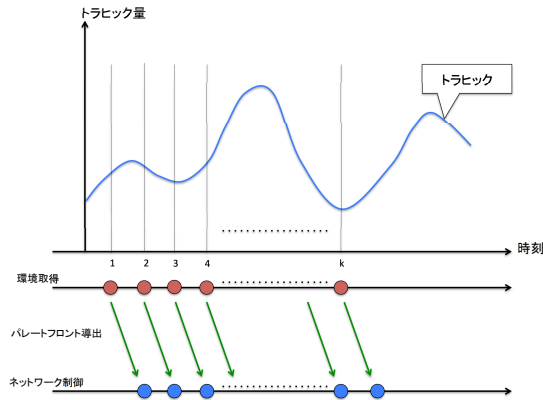


図 2 パレート最適制御における全体の流れ

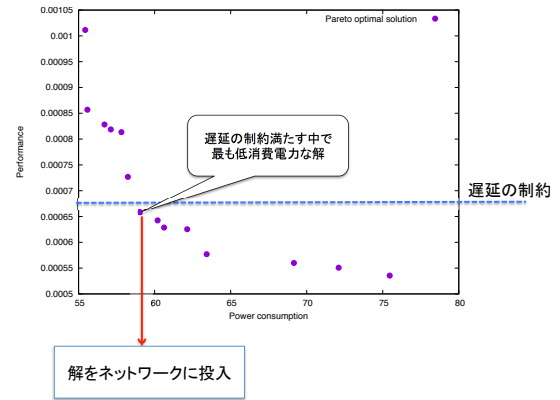


図 3 ネットワークに投入する解の選出

(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

2.2.2 混雑距離

同じランクの個体間での優先順位は、混雑距離 [6] を用いて確定する。混雑距離はある個体の周りにおけ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。

$$d_j = \begin{cases} \infty & i \text{ のいずれかの指標が最大 or 最小値} \\ \sum_m \frac{f_m(I_m^{\text{next}}(j)) - f_m(I_m^{\text{prev}}(j))}{f_m^{\text{max}} - f_m^{\text{min}}} & \text{otherwise} \end{cases}$$

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。

3. パレート最適制御に基づくネットワーク省電力化

3.1 パレート最適制御のネットワークへの適用

図 2 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ヒック観測情報を収集、各時刻のトラヒック状況に応じて、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。本計算では、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、ネットワーク管理者が定めた性能、信頼性の制約条件を満たすもののうち、最も消費電力が小さい解を投入する(図 3)。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
- (2) 進化計算実行：パレートフロントを導出
- (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

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 2.2 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、以下で、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。

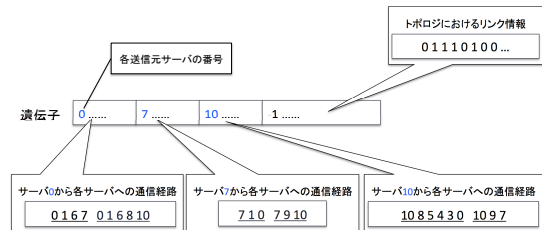


図 4 遺伝子の構造

3.1.1 遺伝子

進化計算では、各個体に該当する解を遺伝子としてコーディングする。提案手法では、ネットワーク内の経路、電源の投入状態を遺伝子としてコーディングする必要がある。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 という値を入れる。これにより、情報の境界を判別することが可能となる。本ネットワーク制御で用いる遺伝子の例を図 4 に示す。

3.1.2 個体の評価

a) 消費電力

本稿では、ネットワークの消費電力は、電源を投入する必要のあるノードの消費電力とリンクの消費電力の和とする。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。

b) 信頼性

本稿では、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、本稿では、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。

c) 性能

本稿では、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。本稿では、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。

3.1.3 遺伝子操作

本稿では、以下に定める、選択・交差、突然変異により、現在の個体群から新たな個体を生成する。

a) 選択・交差

交差を行うための親個体を、ランクと混雑距離を優先してトーナメント選択により選出する。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、評価値がもっとも良いものを選択する。

そして選択された個体で、交差を、機器間の経路単位で行う。つまり、まず交差率に従い、いずれの機器間の経路を入れ替えるかを定める。そして、決められた機器間の経路を入れ替えた子を生成する。

b) 突然変異

本稿の手法においては、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。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。その中の経路上でランダムに選択したノードから、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。

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。選択されたノードに接続するリンクを含んでいる経路は、オフとなるリンクを経由しないよう再構築され、遺伝子に組み込まれる。

3.2 環境変動への対応

トラヒック変動などの環境変動が生じた際は評価値が変化するため、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、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な計算時間を要する。そこで、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。

3.2.1 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。

3.2.2 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、本稿では、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解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解アーカイブは、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) 進化計算を1世代分行う

(2) 現在の世代の解集合 P と、前の時刻に保存された解アーカイブ 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。

本稿では、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、本稿では、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 $i-j$ 間の通信が経路するリンクのうち、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、

表 1 評価環境

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]

遺伝子 x と x' の類似度 $S(x, x')$ を以下のように定義する .

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

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, x と x' で混雑するリンクを経由する通信が異なる場合は, $S(x, x')$ 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 .

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

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 .

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

3.2.3 アーカイブを利用した多目的進化計算

提案手法では, 各時刻における初期解を, パレートアーカイブと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) ランクの高い解を優先し個体数 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 2.2 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, 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 .

4. 評価

4.1 評価環境

4.1.1 ネットワークトポロジ

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, 各スイッチあたりのリンク数を 8 とした . 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 [3], 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 .

a) トラヒック

本評価では, 図 5 のように定常的に通信パターンの変動が起きているトラヒックを用いる . これにより, トラヒックの時間変動へ提案手法が追従できることを示す .

b) 比較手法

本評価では, 以下の 3 つの場合を比較する .

多様解保持あり (w/ DS; with Diverse Solutions) 本研究

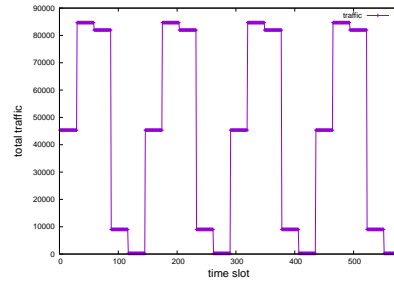


図 5 トラヒック変動パターン

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

表 3 評価に用いた世代数

w/o DS の世代数	R の世代数	w/ DS の世代数
50	49	45

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 .

多様解保持なし (w/o DS; without Diverse Solution) 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 .

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

本評価では, これらの手法を評価に用いることにより, パレート最適制御により, 環境変動に追従して消費電力を抑えることができることを明らかにする .

いずれの手法においても, 表 2 のパラメータを用い, 進化計算を行う . ただし, R は初期解の生成に手順が必要であり, w/ DS は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, w/o DS よりも多くの計算時間を要する . そこで, 表 3 に示す世代数での評価を行った .

c) SLA

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 $250[\mu s]$ 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 .

4.1.2 評価結果

図 5 の変動パターン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 6(a) において, 横軸はタイムスロット, 縦軸は消費電力を示す . w/ DS, w/o DS 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, 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, 必要

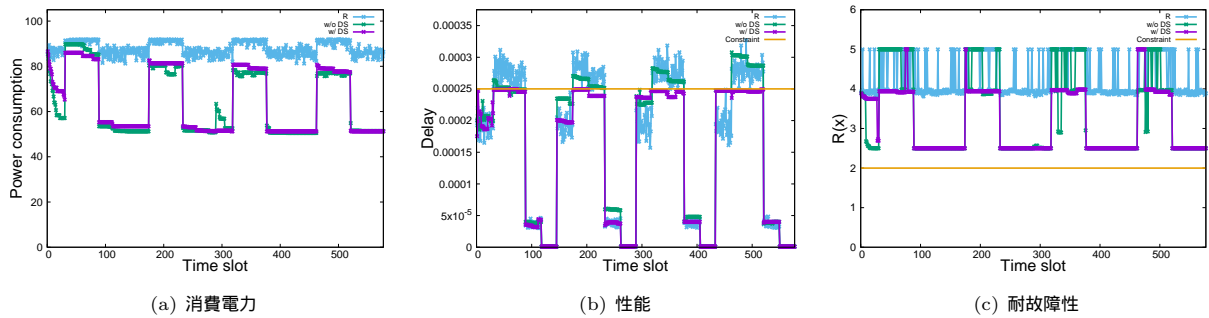


図 6 各指標ごとの評価値

な制約を満たすことができないためである。w/ DS や w/o DS を用いた場合は、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随したネットワーク低消費電力化を実現できている。しかし、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、ランダムな初期解では、49 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トを導出しつつ、環境変動に追随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 6(b)、独立な経路数を図 6(c) に示す。図 6(b) 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 6(c) 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、w/ DS を用いることで全ての時刻において、性能、耐故障の要件を満たす解を投入することができる。これは、w/ DS で求めたパレートフロントにおいて、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。w/o DS を用いた場合の時刻 175・318・461 以降の性能において、制約を満たさず、解がほとんど改善されていない様子が見られる。時刻 117 付近のような環境では、トラヒック量が 0 に近い値に近づき、ほとんどのパレートフロント解が遅延の面で優劣のない解となってしまう。これにより、その後の進化計算で世代を重ねたパレートフロント解は、トラヒック量が増加した際に輻輳してしまう解が多くなり、制約を満たせる解がほとんど残っていない状況になる。その結果、進化をしようとしても、なかなか時刻 175・318・461 のようにトラヒック量が多い時間帯の制約を満たすような解に到達できない状況に陥る。よって、w/ DS のようにパレートフロント解のみだけでなく以前省電力になったような多様な解が残ることで、高速に適切なパレートフロントを取得し、よりトラヒックの増減に追随した制御が可能になると言える。

5. おわりに

本稿では、短時間のトラヒック変動や故障などの環境変動に追随して、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。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随して行うために、本稿では、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。

本稿では、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、環境変動に追随し、性能・信頼性の要件を満たしつつ、全機器稼働状態の 53% 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。

今後、より一般的なトポロジでの評価を予定している。

謝辞 本研究の一部は、文部科学省科学費補助金基盤研究 (C) 16K00125 によっている。ここに記して謝意を表す。

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] 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.
- [4] 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.
- [5] 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.
- [6] 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.