

Osaka University

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

2017/2/10 情報ネットワーク研究会 1

Osaka University

研究背景

- インターネットを介したサービスの普及によるトラフィックの増加に伴い、**ネットワークの消費電力は増加**
- 省電力化手法
 - トラフィック量に応じて、一部のリンクにトラフィックを**集約**
 - トラフィックの流れていないノードとリンクを**スリープ**

2017/2/10 情報ネットワーク研究会 2

Osaka University

研究背景

[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.

- インターネットを介したサービスの普及によるトラフィックの増加に伴い、**ネットワークの消費電力は増加**
- 省電力化手法
 - トラフィック量に応じて、一部のリンクにトラフィックを**集約**
 - トラフィックの流れていないノードとリンクを**スリープ**

従来手法^[3]では、**トラフィックの收容のみを要件**に低消費電力ネットワークを構築

→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

性能(遅延等)
耐故障性 など

複数の要件を保証することが必要

2017/2/10 情報ネットワーク研究会 3

Osaka University

研究目的とアプローチ

- 研究目的
 - トラフィック変動や故障などの環境変動に追従して、耐故障性、性能の確保と低消費電力化のような**複数の要件を同時に達成**するネットワーク制御手法の確立
- アプローチ
 - システムに投入する設定は複数要件を考慮する**パレート最適解**として存在

一方の指標の改善を他の指標を悪化させず行うことが不可能な解

- パレート最適制御を適用
 - パレート最適解の集合 (**パレートフロント**) を導出してそのうちの1つをシステムに投入

2017/2/10 情報ネットワーク研究会 4

Osaka University

パレート最適制御によるネットワーク省電力化

- 以下の手順を各時刻で行う
 - 全通信ペアのトラフィック量を取得
 - 進化計算によりパレートフロントを導出
 - 要件を満たす解をネットワークに投入

2017/2/10 情報ネットワーク研究会 5

Osaka University

進化計算のネットワークへの適用

- 進化計算では**個体**に対して遺伝子操作を行うことで新たな**個体**を導出
 - 個体: ネットワークの設定がコーディングされた遺伝子を持つ
 - 遺伝子操作・淘汰を繰り返すことで真のパレートフロントに近づく

各送信元端末の番号
電源投入されるリンク情報

初期状態 遺伝子 0 7 10 -1

端末0から各端末への通信経路 端末7から各端末への通信経路

2017/2/10 情報ネットワーク研究会 6

Osaka University

進化計算によるパレートフロントの導出

1. 初期化
 - 個体群 P をランダムに生成
2. 評価
 - 各評価指標での評価
 - 非優越ソート: 各解の評価値に応じた自身より全ての指標において優れた解の有無によるランク付け
3. 子の生成
 - 選択→交叉→突然変異により子集団 Q を生成
4. 世代交代
 - P, Q を再評価, トーナメント法でランクと同ランク内では混雑距離を優先して P として保存
 - 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/更新

2017/2/10 情報ネットワーク研究会 7

Osaka University

混雑距離 (Crowding distance)

- 同じランク内での優先順位
- 個体 j の混雑距離 d_j

$$d_j = \begin{cases} \infty & \text{if } f_m(i) = f_m(j) \\ \sum_m \frac{f_m(i^{next}(j)) - f_m(i^{prev}(j))}{f_m^{max} - f_m^{min}} & \text{otherwise} \end{cases}$$

- $f_m(i)$: 個体 i における指標 m の評価値
- $i_m^{next}(j)$: 指標 m が個体 j の次に大きい個体
- $i_m^{prev}(j)$: 指標 m が個体 j の次に小さい個体
- f_m^{max} : 指標 m の最大値
- f_m^{min} : 指標 m の最小値

→ d_j が大きほど個体 j の周りに別の解が存在しない

混雑距離の大きい解を選択することでなるべく多様な解を残すことや状況が違つ解の選択操作が可能となる

2017/2/10 情報ネットワーク研究会 8

Osaka University

各解の評価値

- 消費電力 $E^{net}(x) = \sum_{(i,j) \in E_L} E^{Link} p_{i,j}(x) + \sum_{k \in V} E^{Node} p_k(x)$
 - 電源投入するノードとリンクの消費電力の和
 - オンにする機器が少ないほど低消費電力
- 耐故障性 $R(x) = \frac{1}{\min_{i,j} r_{i,j}(x) + \alpha \sum_{i,j} r_{i,j}(x)}$
 - 独立な経路数の最小値と経路数の総和を重み付けた値の和の逆数
 - 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
- 遅延 $P(x) = \max_{i,j} D_{i,j}(x)$
 - 各端末間の遅延の最大値
 - 通信する量の少ないリンクを経由しているほど低遅延

E^{Link} : リンク 1 本あたりの消費電力
 E^{Node} : ノード 1 つあたりの消費電力
 $p_{i,j}(x)$: 1 or 0 (i-j 間のリンク on/off)
 $p_k(x)$: 1 or 0 (ノード k on/off)
 $r_{i,j}(x)$: i-j 間の独立な経路数
 α : 重み変数
 $D_{i,j}(x)$: i-j 間の遅延
 $r_{i,j}(x)$: i-j 間の独立な経路数
 $q_{i,j}(x)$: i-j 間に含まれるリンクの集合
 $d_{(s,d)}(x)$: リンク s-d の遅延
 T_s : 各リンクの/パケットの平均処理時間
 $r_{i,j}(x)$: リンク s-d の使用率

2017/2/10 情報ネットワーク研究会 9

Osaka University

各解の評価値

- 消費電力 $E^{net}(x) = \sum_{(i,j) \in E_L} E^{Link} p_{i,j}(x) + \sum_{k \in V} E^{Node} p_k(x)$
 - リンクの消費電力の和
 - ノードの消費電力の和
- 耐故障性 $R(x) = \frac{1}{\min_{i,j} r_{i,j}(x) + \alpha \sum_{i,j} r_{i,j}(x)}$
 - 独立な経路数の最小値
 - 独立な経路数の和を 1 以内に丸める
- 遅延 $P(x) = \max_{i,j} D_{i,j}(x)$
 - 各端末間の通信の最大値
 - 経由するリンクの遅延の和
 - M/M/1 モデル

E^{Link} : リンク 1 本あたりの消費電力
 E^{Node} : ノード 1 つあたりの消費電力
 $p_{i,j}(x)$: 1 or 0 (i-j 間のリンク on/off)
 $p_k(x)$: 1 or 0 (ノード k on/off)
 $r_{i,j}(x)$: i-j 間の独立な経路数
 α : 重み変数
 $D_{i,j}(x)$: i-j 間の遅延
 $r_{i,j}(x)$: i-j 間の独立な経路数
 $q_{i,j}(x)$: i-j 間に含まれるリンクの集合
 $d_{(s,d)}(x)$: リンク s-d の遅延
 T_s : 各リンクの/パケットの平均処理時間
 $r_{i,j}(x)$: リンク s-d の使用率

2017/2/10 情報ネットワーク研究会 10

Osaka University

パレートフロント

2017/2/10 情報ネットワーク研究会 11

Osaka University

環境変動に追従したパレートフロントの導出

- ネットワーク環境が変化するとパレートフロントが変化
- 過去のパレートフロントを現在の解として使用することは不適
- 各時刻パレートフロントの計算は高速に行うことが必要

計算を高速化する初期解の設定手法を 2 つ用意

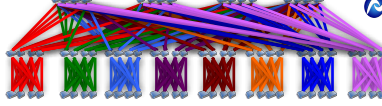
- without Diverse Solutions (w/o DS)
 - 前の時刻のパレートアーカイブを初期値とする手法
- with Diverse Solutions (w/ DS)
 - パレートアーカイブと多様性を維持する解アーカイブを初期値とする手法
 - 環境変動が大きければ以前のパレートフロントは真のパレートフロントとかけ離れたものとなる
 - 変動後でも適切なパレートフロントを高速に導出したい
 - (今のパレートフロントとの距離) × (他の解との類似度) が小さい解を進化計算における評価時に計算

適切な解を短い時間で得ることが可能

2017/2/10 情報ネットワーク研究会 12

Osaka University

評価環境



- トポロジ
 - スイッチ数 80, リンク数 256 の FatTree 型トポロジ
- トラヒック
 - ランダムに選択した機器間において通信が発生
 - あらかじめ定義した総トラヒック量内で生成
 - 通信ペアが同じ状態で定期的にトラヒック量の変動が起きている状況
- 設定した要件
 - 全機器間において遅延が 250 [μs] 以下で、2本の独立な経路を確保
- 進化計算におけるパラメータ

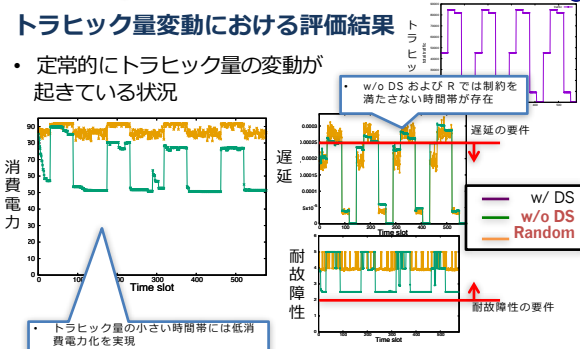
母集団の個体数	交叉率	突然変異率	解アーカイブの上限個体数
30	0.5	0.5	30

2017/2/10 情報ネットワーク研究会 13

Osaka University

トラヒック量変動における評価結果

- 定期的なトラヒック量の変動が起きている状況



- 消費電力
- 遅延
- 耐故障性

• w/o DS および R では制約を満たさない時間帯が存在

• トラヒック量の小さい時間帯には低消費電力化を実現

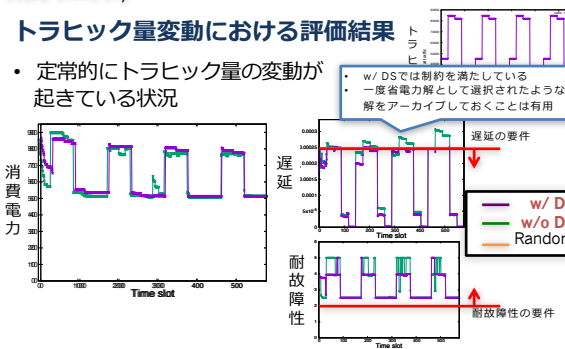
• 前の時刻のパレートフロントの利用により各時刻短い世代数の計算で変動に追隨した省電力化を実現

2017/2/10 情報ネットワーク研究会 14

Osaka University

トラヒック量変動における評価結果

- 定期的なトラヒック量の変動が起きている状況



- 消費電力
- 遅延
- 耐故障性

• w/ DSでは制約を満たしている

• 一度省電力解として選択されたような解をアーカイブしておくことは有用

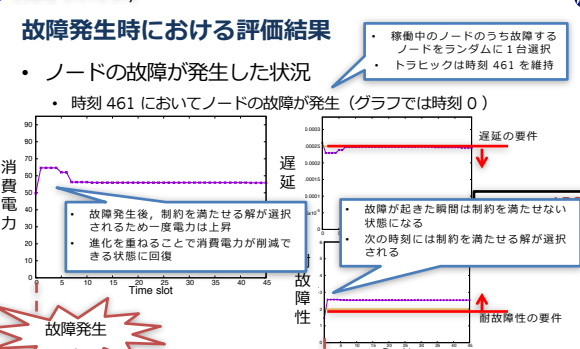
環境変動に追隨し、性能・耐故障性の要件を満たしつつ、ネットワークの消費電力を低減

2017/2/10 情報ネットワーク研究会 15

Osaka University

故障発生時における評価結果

- ノードの故障が発生した状況



- 消費電力
- 遅延
- 故障発生
- 故障
- 耐故障性

• 稼働中のノードのうち故障するノードをランダムに1台選択

• トラヒックは時刻 461 を維持

時刻 461 においてノードの故障が発生 (グラフでは時刻 0)

故障発生後、制約を満たせる解が選択されるため一度電力は上昇

進化を重ねることで消費電力が削減できる状態に回復

故障が起きた瞬間は制約を満たせない状態になる

次の時刻には制約を満たせる解が選択される

故障発生後も経路を再構築、その後消費電力が回復

2017/2/10 情報ネットワーク研究会 16

Osaka University

まとめと今後の課題

- まとめ
 - 提案手法により、環境変動に追隨し、性能・耐故障性の要件を満たしつつ、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
 - 提案手法により、故障発生時には経路を再構築し、その後世代を重ねることで消費電力が削減できる状態に回復していくことを示した
- 今後の課題
 - より一般的なネットワークでの評価

2017/2/10 情報ネットワーク研究会 17