

# 障害耐性も考慮した ネットワーク低消費電力化制御

大下 裕一

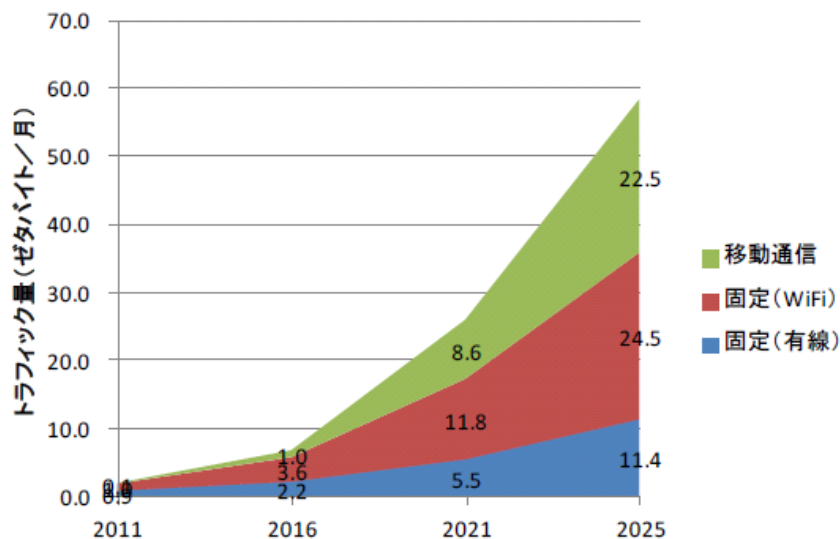
大阪大学 大学院情報科学研究科

# ネットワークの省電力化の必要性

- ネットワークトラフィックの爆発
- ネットワーク機器の消費電力の増大

## ネットワークトラフィックの爆発

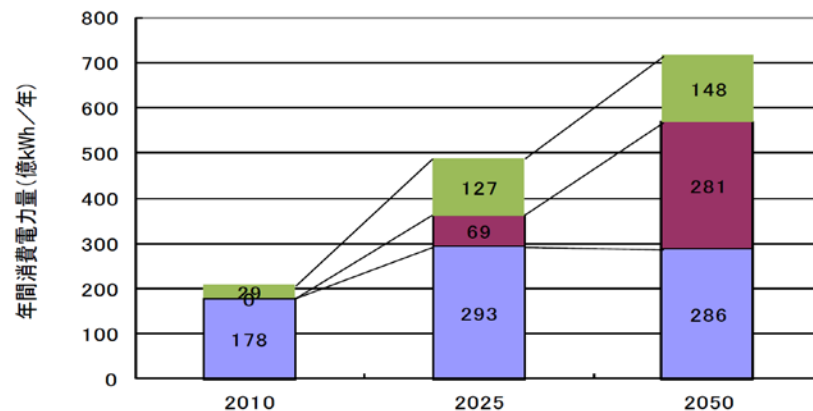
• 2025年:2011年の約30倍



## ネットワーク機器の電力も増加

• 2025年:2010年の1.7倍

### 日本のルータ・スイッチの年間消費電力量(億kWh/年)



出典：経済産業省 IT機器のエネルギー消費量に係る調査事業 報告書

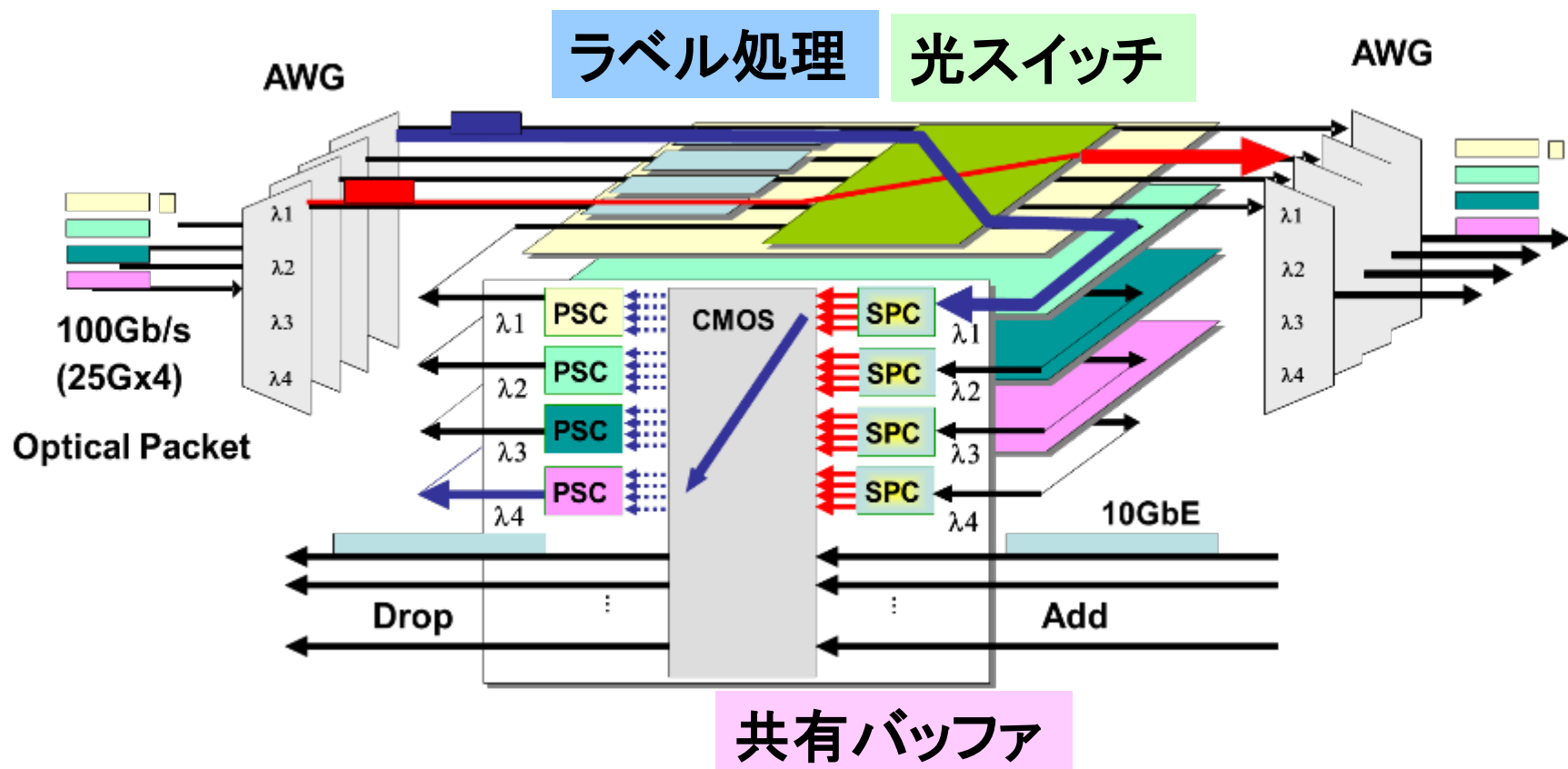
ネットワークの消費電力の削減が課題

# ネットワーク低消費電力化の方向性

- 機器自体の消費電力の削減
  - 電気スイッチの低消費電力化
    - チップに集約
  - 光通信技術の活用
    - 光回線交換
    - 光パケットスイッチ
- ネットワーク全体を通しての消費電力の削減
  - 不要な機器の電源を落とすことにより、消費電力を削減する

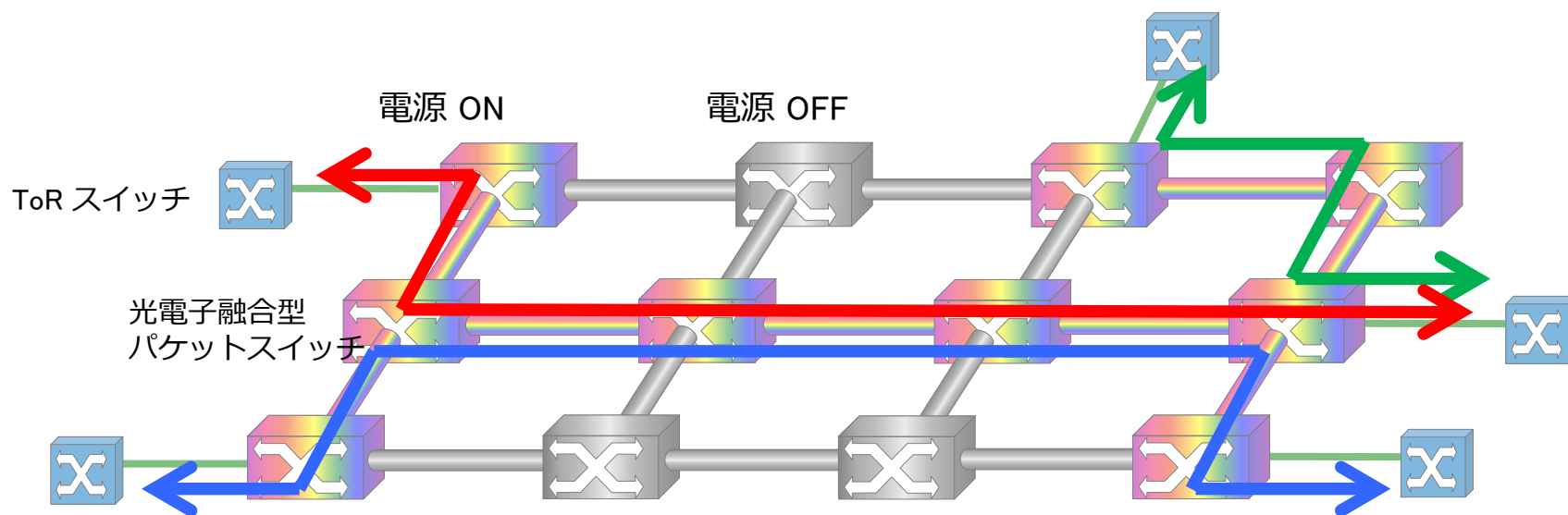
# 機器自体の消費電力の削減のとりくみ

光電子融合型パケットルータ（NTT、NEC、阪大、九大）



# 光電子融合型パケットルータネットワークの省電力化

- 必要最低限のバッファのみを用いてトラヒックを集約し、電源の投入が必要なスイッチ・バッファの総消費電力を最小化



# 低消費電力化だけでよいのか

## 省電力化制御

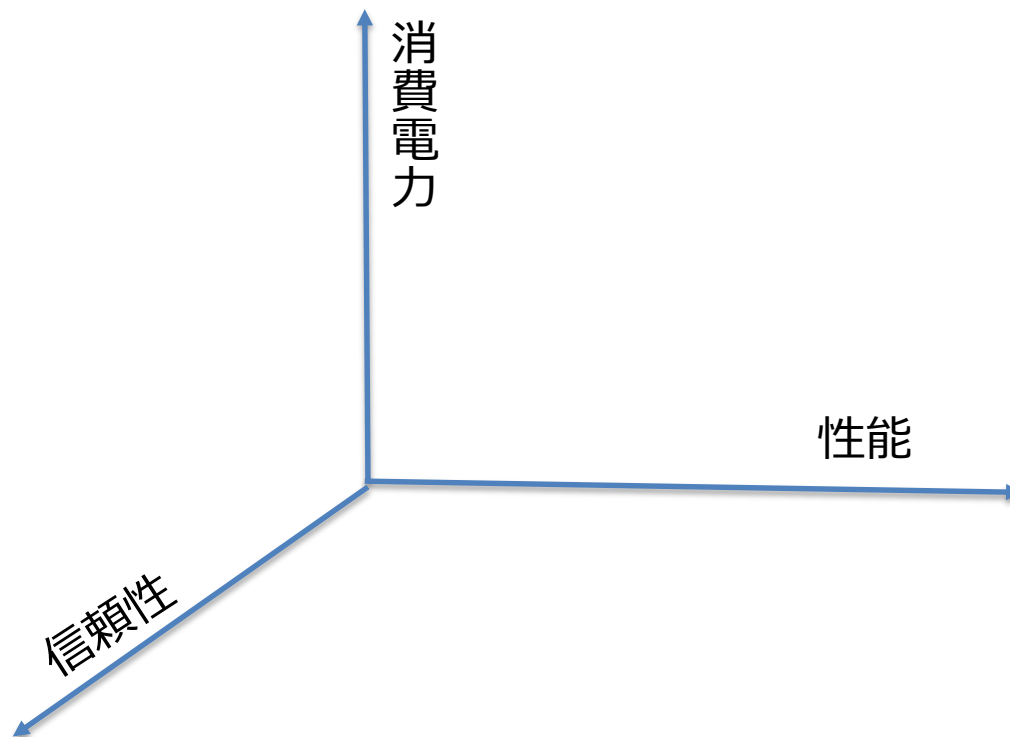
- トラヒックをネットワーク機器の一部に片寄
- 不要な機器の電源を落とす

## ネットワーク機器が故障すると…

- ネットワークが不通となる
- 回復には、機器の電源投入完了までの時間を考慮する必要がある

## 考える必要のある制御

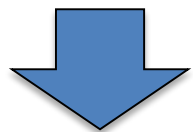
- 信頼性、消費電力、性能の3軸で考え、適切な解を選ぶことが必要



# パレート最適制御にもとづくネットワーク省電力化

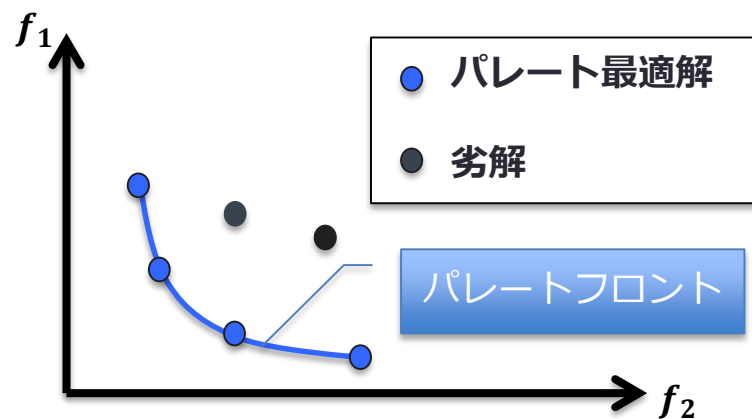
## • アプローチ

- 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



## • パレート最適制御を適用

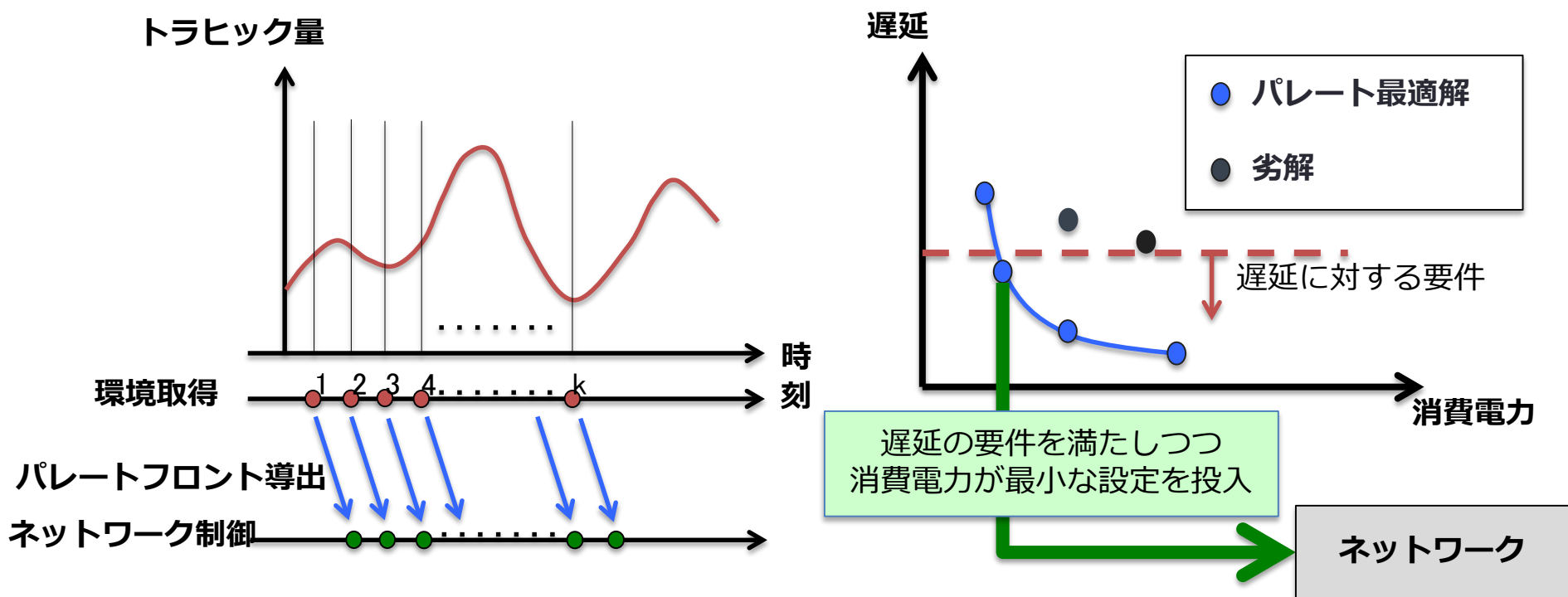
- パレート最適解の集合 (**パレートのフロント**) を導出してそのうちの1つをシステムに投入





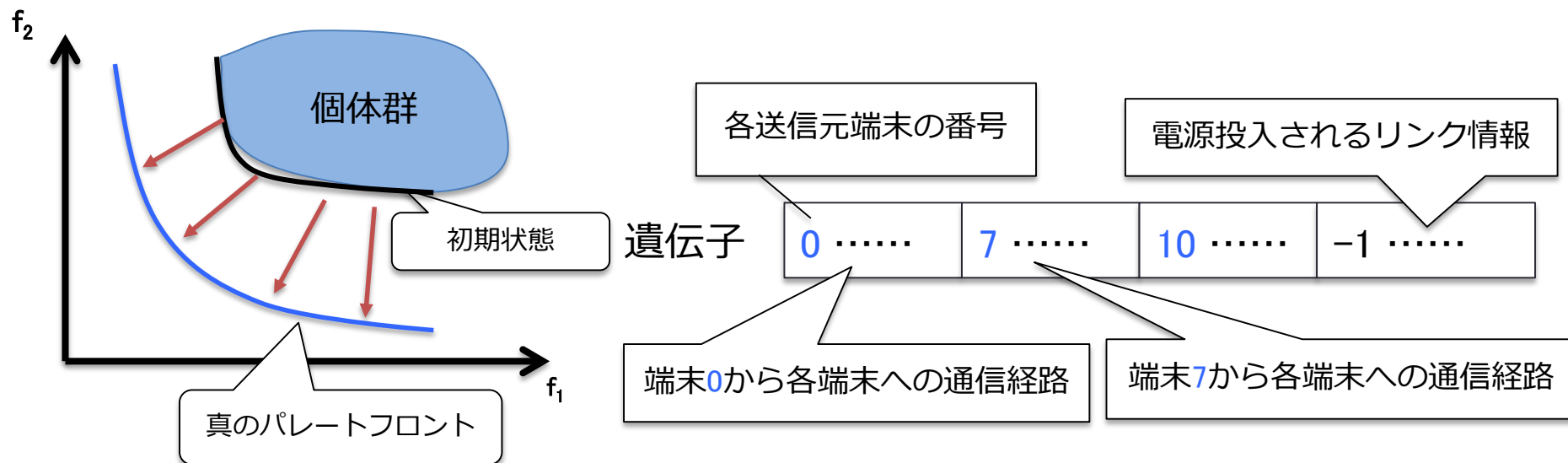
# パレート最適制御によるネットワーク省電力化

- 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



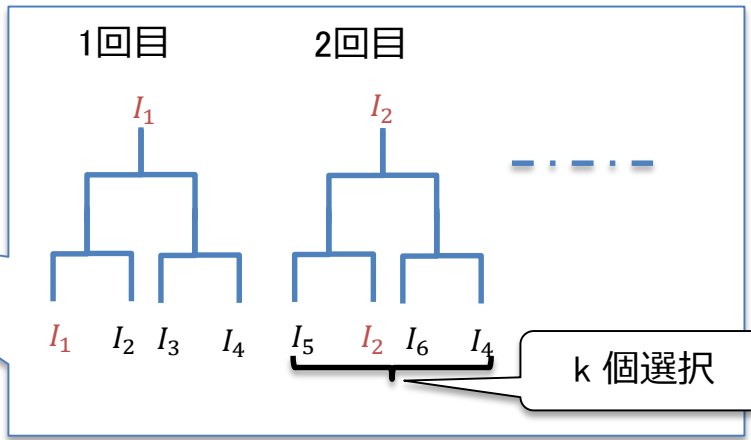
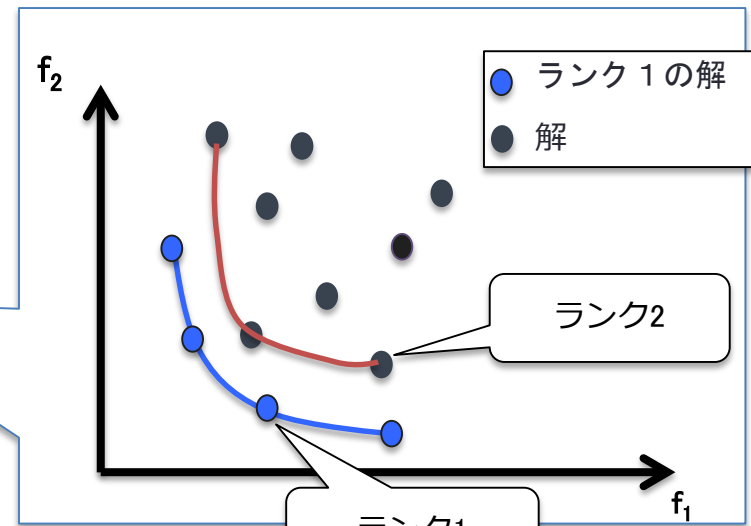
# 進化計算のネットワークへの適用

- 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保持  
→ 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



# 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 U 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



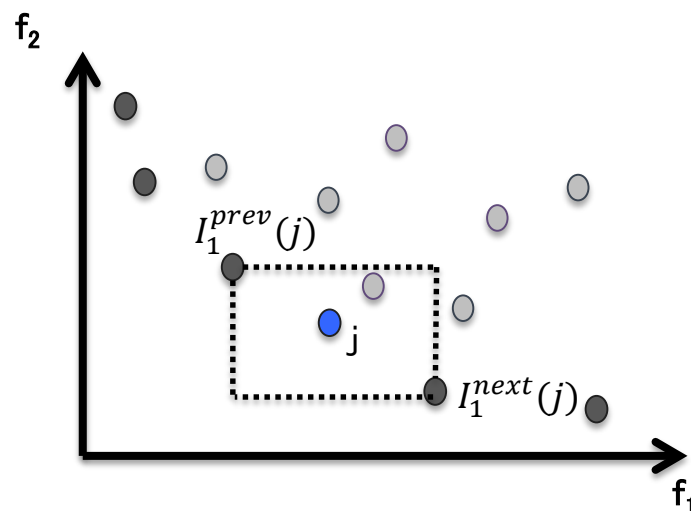
## 混雑距離 (Crowding distance)

- 同じランク内での優先順位
- 個体  $j$  の混雑距離  $d_j$

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

- $f_m(i)$  : 個体  $i$  における指標  $m$  の評価値
- $I_m^{\text{next}}(j)$  : 指標  $m$  が個体  $j$  の次に大きい個体
- $I_m^{\text{prev}}(j)$  : 指標  $m$  が個体  $j$  の次に小さい個体
- $f_m^{\text{max}}$  : 指標  $m$  の最大値
- $f_m^{\text{min}}$  : 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**う解の**選択操作**が可能となる

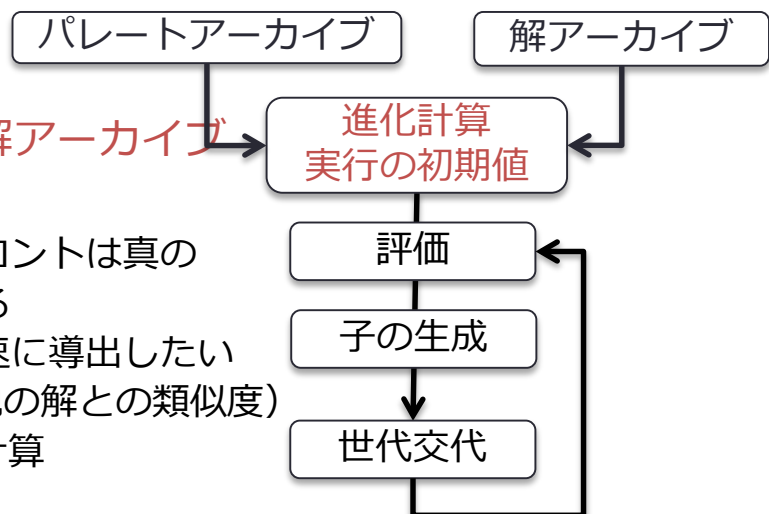
## 環境変動への追隨

- 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



## 環境変動への追隨性の向上

- **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
  - $(\text{今のパレートフロントとの距離}) \times (\text{他の解との類似度})$ が小さい解を進化計算における評価時に計算



## 各解の評価値

- 消費電力  $E^{net}(x) = \sum_{(i,j) \in L} E^{Link} p_{i,j}(x) + \sum_{k \in V} E^{Node} p_k(x)$

- 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した値の和の逆数
  - 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

$$D_{i,j}(x) = \sum_{(s,d) \in q_{i,j}(x)} d_{(s,d)}(x) \quad d_{(s,d)}(x) = T_s \frac{\rho_{s,d}(x)}{1 - \rho_{s,d}(x)}$$

$E^{Link}$  : 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  
 $\alpha$  : 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  
 $\rho_{s,d}(x)$  : リンク  $s-d$  の使用率

# 各解の評価値

- 消費電力  $E^{net}(x) = \sum_{(i,j) \in L} E^{Link} p_{i,j}(x) + \sum_{k \in V} E^{Node} p_k(x)$

リンクの消費電力の和

ノードの消費電力の和

$E^{Link}$  : 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  
 $\alpha$  : 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

- 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

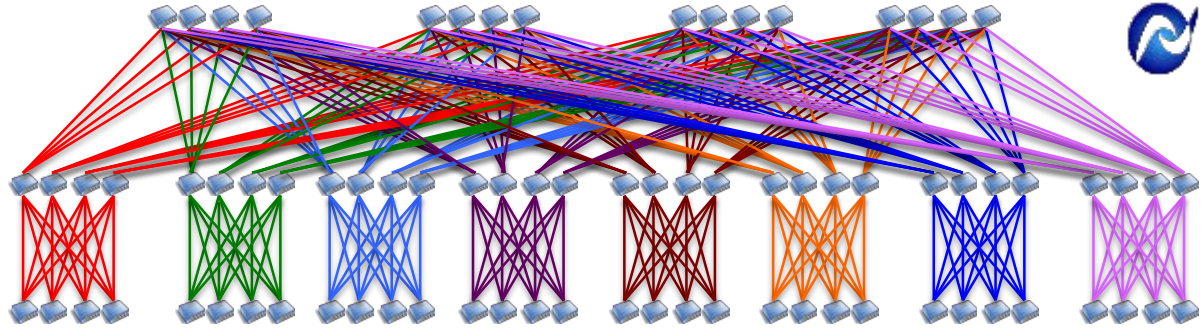
経路するリンクの遅延の和

M/M/1 モデル

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

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

## 評価環境



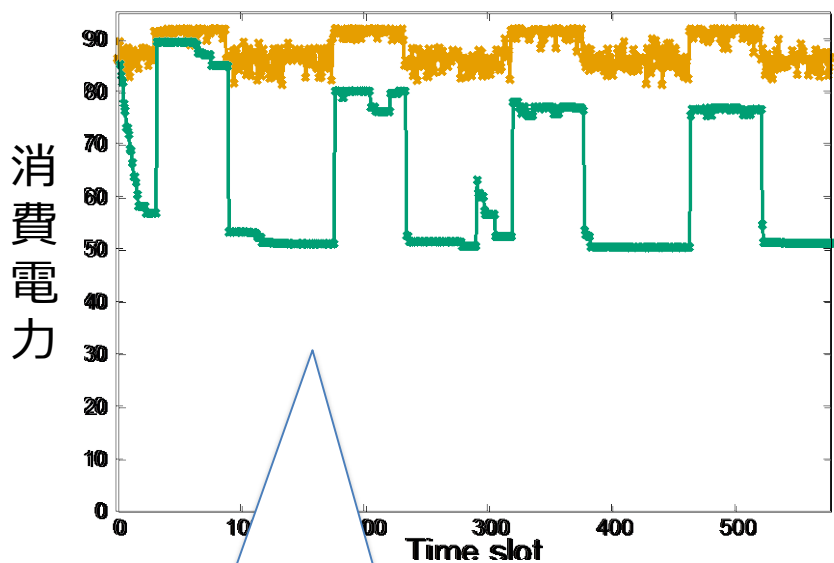
- トポロジ
  - スイッチ数 80 , リンク数 256 の FatTree 型トポロジ
- トラヒック
  - ランダムに選択した機器間において通信が発生
  - あらかじめ定義した総トラヒック量内で生成
    - 通信ペアが同じ状態で定常的にトラヒック量の変動が起きている状況
- 設定した要件
  - 全機器間において遅延が 250 [ $\mu$ s] 以下で, 2 本の独立な経路を確保
- 進化計算におけるパラメータ

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

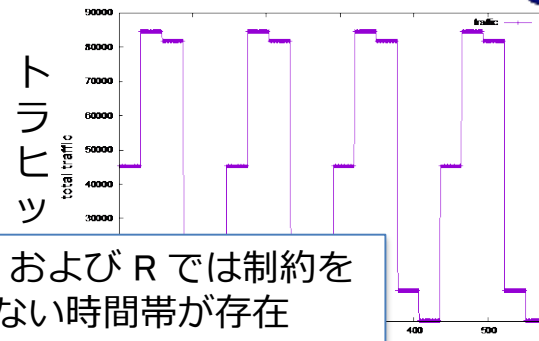


# トラヒック量変動における評価結果

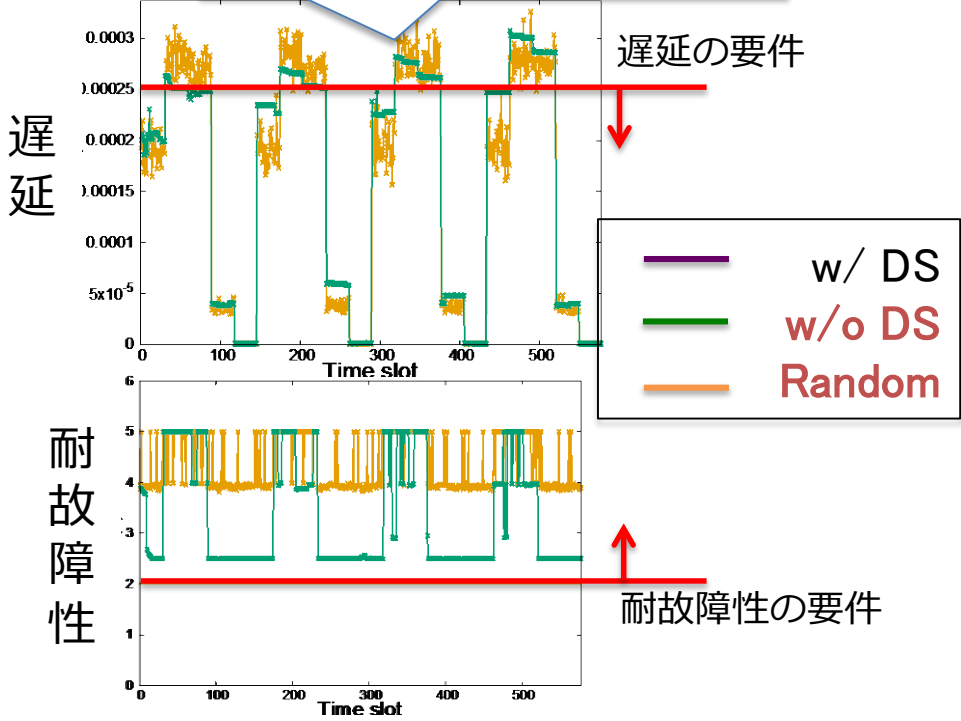
- 定常的にトラヒック量の変動が起きている状況



- 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

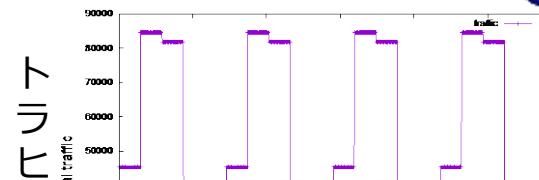


• w/o DS および R では制約を満たさない時間帯が存在

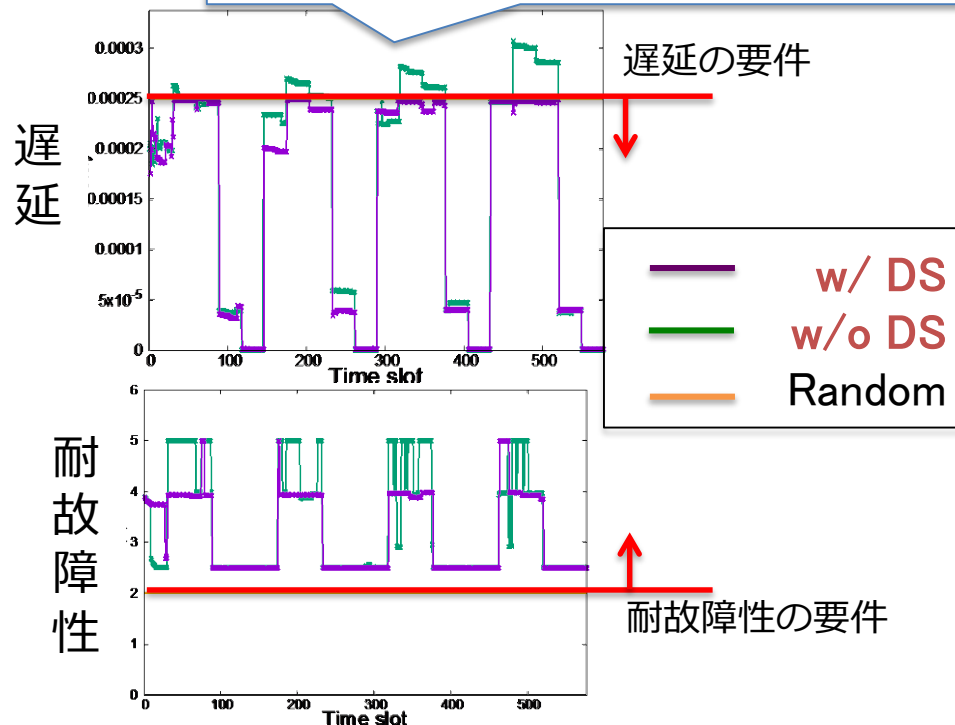
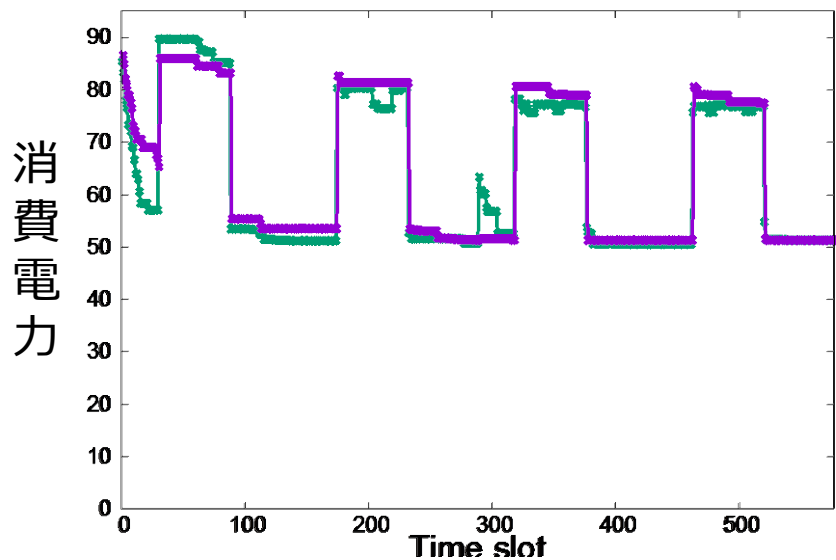


# トラフィック量変動における評価結果

- 定常的にトラフィック量の変動が起きている状況



- 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



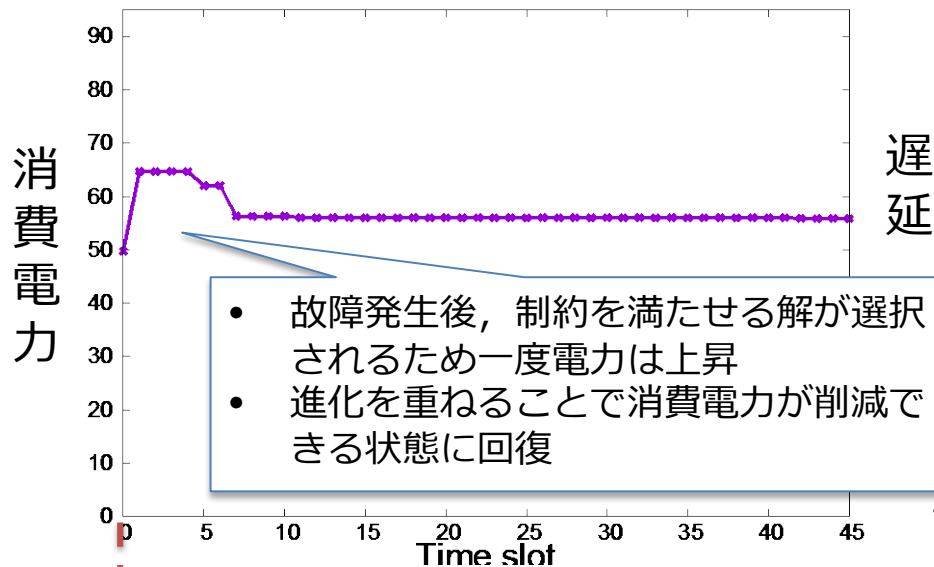
環境変動に追従し, 性能・耐故障性の要件を満たしつつ,  
ネットワークの消費電力を低減

# 故障発生時における評価結果

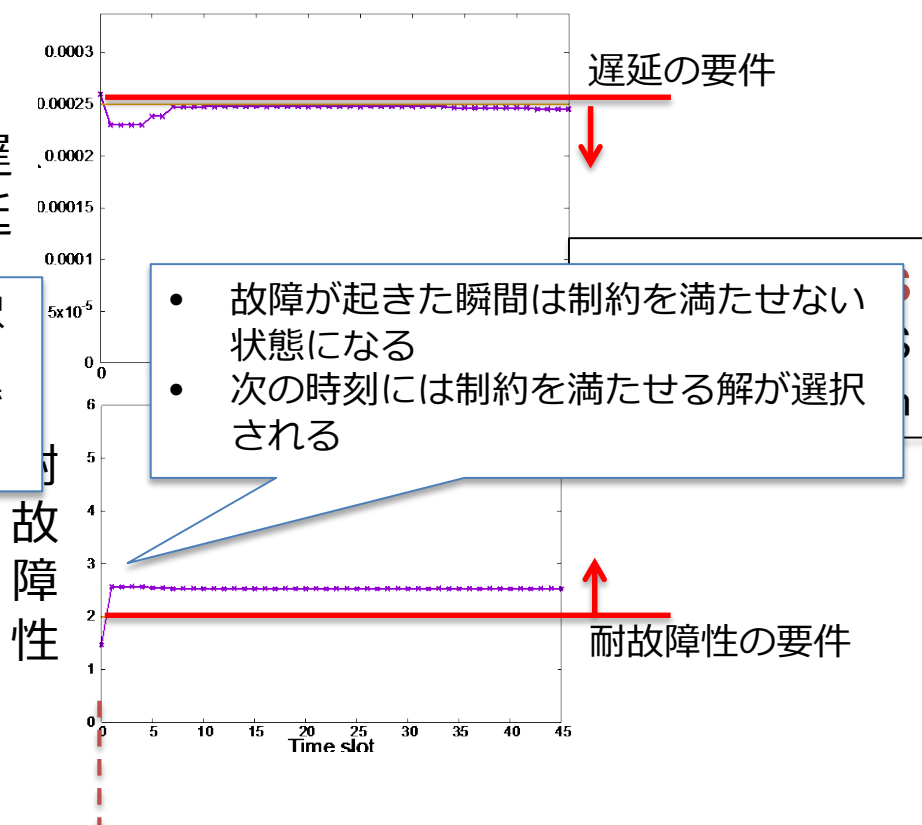
- ノードの故障が発生した状況

- 時刻 461 においてノードの故障が発生（グラフでは時刻 0）

- 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
- トラフィックは時刻 461 を維持



- 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

故障発生後にも経路を再構築，その後消費電力が回復

# まとめと今後の課題

- まとめ
  - 提案手法により、環境変動に追従し、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