

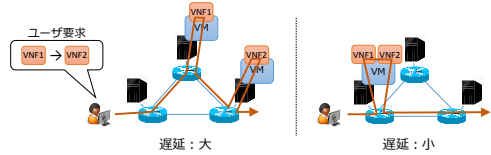
An Evolvable Approach for the Virtual Network Function Placement Problem in Varying Environments

進化適応性を備えた仮想ネットワーク機能配置手法

村田研究室 乙倉麻里

研究背景

- 仮想ネットワーク機能 (VNF)
 - Network Function Virtualization (NFV) 技術によって仮想化されたネットワーク機能
 - VNF は仮想計算機 (VM) 上で動作
- VNF 配置問題
 - VNF を実行する VM の物理計算機 (PM) への配置状態を決定



動的 VNF 配置問題

- 動的 VNF 配置問題
 - ユーザの動的な要求変更に応じて VM の配置状態を再構成
- 動的 VNF 配置問題における 2 つの要件
 - 要求変更時に必要となる配置再構成操作回数の削減
 - 配置再構成操作: VM 移動、VM サイズ変更、VM 追加、VM 削除、VNF 移動、VNF サイズ変更
 - 各要求に対しての配置状態の良さ
- 静的な VNF 配置手法を適用する際の問題点
 - 配置再構成操作回数の削減が困難
 - 従来の VNF 配置手法 [10][12] では考慮できない
 - 要求変動よりも短い時間スケールでの最適化計算が困難
 - VNF の配置最適化自体が NP 困難

[10] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On Orchestrating Virtual Network Functions in NFV," in Proceedings of 11th International Conference on Network and Service Management (Net-Service), Nov. 2015.
 [12] R. Cohen, L. Lewin-Eytan, J. Naor, and D. Raz, "Near Optimal Placement of Virtual Network Functions," in Proceedings of 2015 IEEE Conference on Computer Communications, pp. 1346-1354, Apr. 2015.

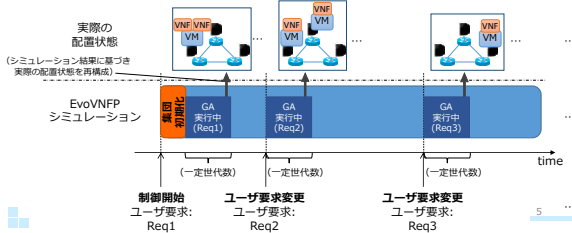
研究の目的とアプローチ

- 本研究の目的
 - 以下の特徴を有する動的 VNF 配置手法の提案
 - 配置再構成操作回数を抑制
 - 短時間で配置状態を生成可能
 - 各要求に対して良い配置状態を生成可能
- 本研究のアプローチ
 - 生物の環境変動への適応性の知見を利用
 - 生物は、環境変動の中で進化するにより、環境変動に対し少しの構造変化で適応可能な構造を獲得 [1]
 - ユーザの動的な要求変更への適応性を有する配置状態を生成

[1] N. Kahtan and U. Alon, "Spontaneous evolution of modularity and network motifs," PNAS, vol. 102, no. 39, pp. 13773-13778, Sept. 2005.

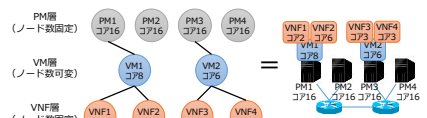
EvoVNF Placement (EvoVNF)

- 要求変更への適応性を有する配置状態を生成する動的 VNF 配置手法
 - 一定世代数の遺伝的アルゴリズム (GA) を用いて、ユーザの要求を満たす配置状態を生成
 - 配置状態生成後は次のユーザ要求変更まで実行を停止
 - ユーザの要求変更時は、集団初期化を行わずに、1 つ前の要求に対して生成されていた集団を用いて、新しい要求を満たす配置状態を生成

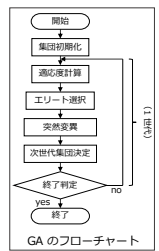


EvoVNF 内で用いられる GA の概要

- 各要求に対しての配置状態の良さに基づいて良い配置状態を選択
 - 適応度: 各要求に対しての配置状態の良さ
 - 平均ユーザ遅延が小さく、使用コア数が少ないほど大きな値
 - 個体: 3 層ネットワーク (下図左)
 - 配置状態をネットワークの接続状態で表現
 - PM 層と VM 層の接続: PM への VM の配置
 - VM 層と VNF 層の接続: VM への VNF の配置

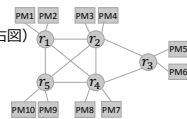


- 突然変異: ネットワークの要素を 1 つランダムに選択・変更
 - リンク接続状態、各ノードのコア数、VM 層ノード数のいずれか



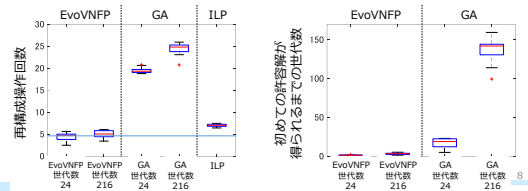
シミュレーション設定

- 物理ネットワーク
 - 5 ルータ、10 PM、各 PM は 16 コア保有 (右図)
- ユーザ処理要求
 - 3 つのユーザ処理要求
 - 始点 r_1 , 終点 r_3 , VNF1→VNF2, b_{u_1} [Mbps]
 - 始点 r_5 , 終点 r_3 , VNF2, b_{u_2} [Mbps]
 - 始点 r_4 , 終点 r_2 , VNF1→VNF3, b_{u_3} [Mbps]
 - 要求伝送速度 b_{u_1} , b_{u_2} , b_{u_3} が時間経過により変動
- 評価指標
 - 配置再構成に関する指標: 配置再構成操作回数、各実行において最初の許容解を得るまでの世代数
 - 各時点での配置状態の良さに関する指標: 使用コア数、平均ユーザ遅延
- 比較手法
 - 静的な VNF 配置手法を要求変動ごとに再実行
 - EvoVNFP で用いている GA
 - 整数計画問題 (ILP)



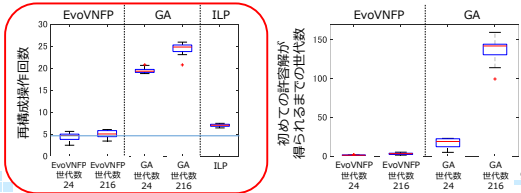
配置再構成に関する評価

- EvoVNFP は GA や ILP と比較して少ない操作回数で配置再構成が可能
- EvoVNFP では GA と比較して少ない世代数で許容解を生成可能
- EvoVNFP では、各要求に対して用いる世代数を適切に設定することにより、再構成操作回数、初めての許容解が得られるまでの世代数共に改善
 - 適切な設定により、各要求への過適応を防止



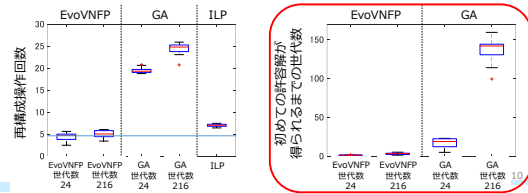
配置再構成に関する評価

- EvoVNFP は GA や ILP と比較して少ない操作回数で配置再構成が可能
- EvoVNFP では GA と比較して少ない世代数で許容解を生成可能
- EvoVNFP では、各要求に対して用いる世代数を適切に設定することにより、再構成操作回数、初めての許容解が得られるまでの世代数共に改善
 - 適切な設定により、各要求への過適応を防止



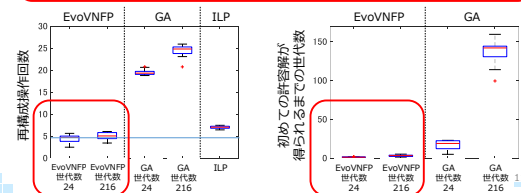
配置再構成に関する評価

- EvoVNFP は GA や ILP と比較して少ない操作回数で配置再構成が可能
- EvoVNFP では GA と比較して少ない世代数で許容解を生成可能
- EvoVNFP では、各要求に対して用いる世代数を適切に設定することにより、再構成操作回数、初めての許容解が得られるまでの世代数共に改善
 - 適切な設定により、各要求への過適応を防止



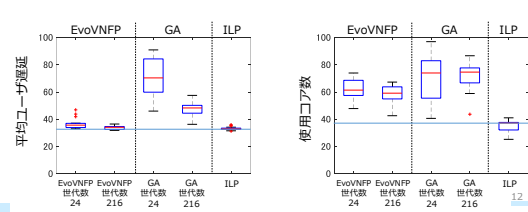
配置再構成に関する評価

- EvoVNFP は GA や ILP と比較して少ない操作回数で配置再構成が可能
- EvoVNFP では GA と比較して少ない世代数で許容解を生成可能
- EvoVNFP では、各要求に対して用いる世代数を適切に設定することにより、再構成操作回数、初めての許容解が得られるまでの世代数共に改善
 - 適切な設定により、各要求への過適応を防止



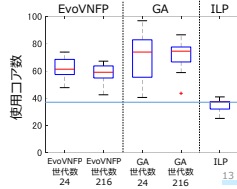
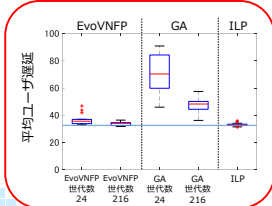
各要求に対する配置状態に関する評価

- EvoVNFP で得られる配置状態の平均ユーザ遅延は ILP とほぼ等しい
- EvoVNFP で得られる配置状態の使用コア数は ILP と比べて約2倍
 - 再構成操作回数と使用コア数はトレードオフの関係
- EvoVNFP では、各要求に対して用いる世代数を増加させることにより、平均ユーザ遅延、使用コア数共に改善



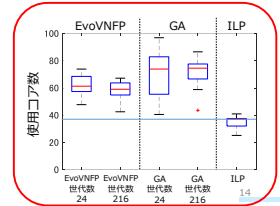
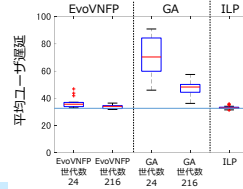
各要求に対する配置状態に関する評価

- EvoVNFP で得られる配置状態の平均ユーザ遅延は ILP とほぼ等しい
- EvoVNFP で得られる配置状態の使用コア数は ILP と比べて約2倍
 - 再構成操作回数と使用コア数はトレードオフの関係
- EvoVNFP では、各要求に対して用いる世代数を増加させることにより、平均ユーザ遅延、使用コア数共に改善



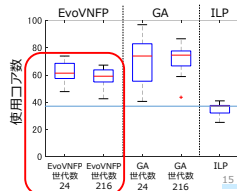
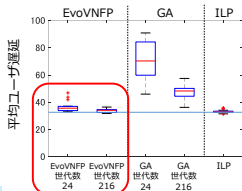
各要求に対する配置状態に関する評価

- EvoVNFP で得られる配置状態の平均ユーザ遅延は ILP とほぼ等しい
- EvoVNFP で得られる配置状態の使用コア数は ILP と比べて約2倍
 - 再構成操作回数と使用コア数はトレードオフの関係
- EvoVNFP では、各要求に対して用いる世代数を増加させることにより、平均ユーザ遅延、使用コア数共に改善



各要求に対する配置状態に関する評価

- EvoVNFP で得られる配置状態の平均ユーザ遅延は ILP とほぼ等しい
- EvoVNFP で得られる配置状態の使用コア数は ILP と比べて約2倍
 - 再構成操作回数と使用コア数はトレードオフの関係
- EvoVNFP では、各要求に対して用いる世代数を増加させることにより、平均ユーザ遅延、使用コア数共に改善



まとめと今後の課題

- まとめ
 - 動的 VNF 配置手法 EvoVNFP を提案
 - GA を用いて、各ユーザ処理要求を満たす VNF 配置状態を生成
 - ユーザ処理要求変更時は、集団を初期化せずに GA を再実行
 - シミュレーション評価により EvoVNFP の有効性を確認
 - 比較手法に対して、配置再構成に必要な操作回数を削減
 - 資源 (コア) を十分に確保した配置状態を生成
 - 比較手法に対して、配置状態の計算に必要な世代数を削減
 - 各要求に対して、最適配置状態とほぼ同等のサービス性能となる配置状態を生成
- 今後の課題
 - EvoVNFP と もう 1 つの提案手法である EvoVNFP-2 の比較
 - より大規模なネットワークを想定した検証