

品質多様性アルゴリズムを用いた5G ネットワークにおける動的ネットワークスライシング制御手法の提案

大月 天渡[†] 小南 大智[†] 大歳 達也^{††} 下西 英之^{†††} 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒565-0879 大阪府吹田市山田丘 1-5

^{††} 大阪大学 大学院経済学研究科 〒560-0043 大阪府豊中市待兼山 1-7

^{†††} 大阪大学 サイバーメディアセンター 〒560-0043 大阪府豊中市待兼山町 1-32

E-mail: †{a-otsuki,d-kominami,murata}@ist.osaka-u.ac.jp, ††t-otoshi@econ.osaka-u.ac.jp,
†††shimonishi.cmc@osaka-u.ac.jp

あらまし 5G ネットワークにおいて多様な要件を持つ種々のアプリケーションを同時にホストする上で、複数の独立した専用の仮想ネットワークを提供するネットワークスライシング技術は重要な要素である。ネットワークスライスに対する基盤資源の割り当てを最適化する問題はNSE問題として定式化されており、NP困難であるため既存研究ではヒューリスティクスが提案されている。現実的なシナリオでは複数の要件の異なるネットワークスライスを並列的に埋め込む必要があり、相互に要件を考慮しなければ全体最適は実現できない。一方で、統一的に制御する場合にはスライスの数に対して問題空間が指数的に増大するため、個別の制御が不可欠である。このため、個別に制御を行いながら全体最適化を図る必要がある。本稿では、生物進化モデルを応用した品質多様性アルゴリズムを用いて、ネットワークスライスを個別制御しながら協調的な動作によって全体最適化を実現する手法を提案する。また、計算機シミュレーションを通して既存手法との比較評価を行い、提案手法の有効性およびその性質を明らかにした。

キーワード ネットワークスライシング、NSE (Network Slice Embedding)、品質多様性アルゴリズム

1. はじめに

近年の5Gネットワークのユースケースの多様化に伴い、ネットワークに対するサービス要件も多様化している。具体的には、自動運転や遠隔医療ロボットなどの高信頼かつ低遅延が求められるURLLC (Ultra-reliable and Low Latency Communications)、動画ストリーミングなどの高データレートが求められるeMBB (enhanced Mobile Broadband)、センサネットワークやIoTなどの大規模ネットワークの同時通信に必要なmMTC (Massive Machine Type Communications) といったユースケースが存在する [1]。それぞれの要件の衝突によるサービス品質の低下を防ぐため、共有の基盤ネットワーク上の物理資源を仮想的に分割し、複数の仮想ネットワークを基盤ネットワーク上に構築するネットワークスライシング技術は最も重要な要素技術のひとつである。

複数のクライアントから様々な要件のネットワークスライスが要求される中で、ネットワークプロバイダは有限な物理資源上にそれらのスライスを配置することが求められる。より多くのスライスを限られた物理資源上に効率的に配置することはクライアント、プロバイダの双方にとって重要である。一方で、全体最適化を行うために全てのスライスを同時に最適化することは問題規模の増大化につながり、現実的な時間での求解が困難となる。そのため、スライスを個別に制御する中で、相互に連携し全体のスライス要求受容率を高めることが求められる。

5G ネットワークにおけるネットワークスライシングの制御を最適化問題として捉えるとき、様々なアルゴリズムの側面を持つが、そのうちのひとつに資源割り当て問題としての側面がある [2]。文献 [3] ではネットワークスライスの最適化を Virtual Network Embedding (VNE) 問題として定義しているが、制御の対象が基地局からコアクラウドまでとなっており、UEを含めたE2E通信となっていない。また、文献 [4] では5G ネットワーク上のE2E通信を前提としてネットワークスライスの資

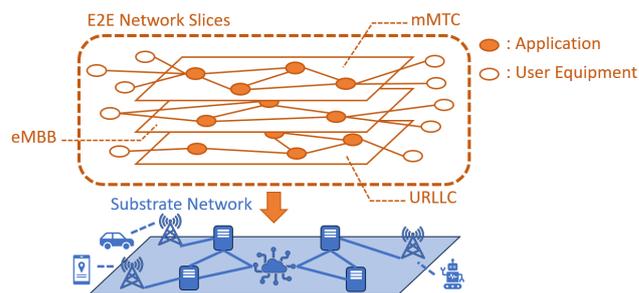


図1 5G ネットワークにおける E2E ネットワークスライシング

源割り当て制御の実装を行っているが、スライスの埋め込みは制御の対象となっておらず、また資源割り当てのアルゴリズムは最適化を前提としていない。文献 [5] は UE を含めた E2E 通信を対象とするネットワークスライスの埋め込みの最適化を、VNE 問題を拡張した NSE 問題として定義している。

NSE 問題において基盤ネットワークとネットワークスライスはそれぞれ無向グラフとしてモデル化され、基盤ネットワークを構成するノードは UE とクラウドノード、ネットワークスライスを構成するノードは UE とアプリケーションに分類される。クラウドノードとリンクは有限の資源を持ち、これを消費してアプリケーションおよび UE・アプリケーション間の通信をホストする。NSE 問題はこの資源制約下において、複数のネットワークスライスの要求に対してどの要求を受容するか、またそのスライスへの資源割り当てを決定する問題である。また、文献 [6] ではアプリケーション間の接続を含めた NSE 問題を定式化しており、本論文での NSE 問題は主にこの文献での記述に準ずる。ただし、この文献で新たに定義されている可用性や信頼性の項目は除外する。

文献 [5] では VNE 問題におけるヒューリスティクスである NodeRank 法 [7] を拡張した手法を提案しており、既存の整数

Algorithm 1 Pseudocode of MAP-Elites [8]

```

procedure MAP-ELITES
   $\mathcal{F} \leftarrow \emptyset, \mathcal{X} \leftarrow \emptyset$ 
  for  $iter \leftarrow 1, I$  do
    if  $iter < G$  then
       $x' \leftarrow \text{random\_solution}()$ 
    else
       $x \leftarrow \text{random\_selection}(\mathcal{X})$ 
       $x' \leftarrow \text{random\_variation}(x)$ 
     $b' \leftarrow \text{feature\_descriptor}(x')$ 
     $f' \leftarrow \text{fitness}(x')$ 
    if  $\mathcal{F}(b') = \text{null}$  or  $\mathcal{F}(b') < f'$  then
       $\mathcal{F}(b') \leftarrow f'$ 
       $\mathcal{X}(b') \leftarrow x'$ 
  return  $\text{feature-fitness map } (\mathcal{F} \text{ and } \mathcal{X})$ 

```

線形計画法による手法と比較して同程度のスライス要求受容率でありながら、計算時間に関して高いスケーラビリティを示している。一方で、この文献ではスライスの要件にレイテンシを含めているにもかかわらず、提案手法ではレイテンシを考慮していない。また、複数のネットワークスライスを並列的に埋め込むことを想定しているが、経時的に変化するネットワーク資源状況や他のスライスの情報をランキングに組み込んでおらず、全体最適化を考慮しないものとなっている。

本研究はスケーラビリティを確保するためネットワークスライスを個別に制御しながら、全体最適化を図る最適化手法の実現を目的とする。提案手法では品質多様性アルゴリズム、特にその一つである MAP-Elites の適応度（目標関数）の設計の自由度、および探索空間の設計の自由度の高さに着目し、レイテンシ要件や他のスライスを管理するコントローラとの連携に必要な項を適応度に組み込み、またその適応度の向上に効果的な探索空間を設計することで効率的な全体最適化を図る。

2. 関連研究

2.1 MAP-Elites Algorithm

MAP-Elites [8] は品質多様性アルゴリズムのひとつであり、高次元の探索空間に対して低次元の特徴空間で最大適応度の分布を可視化できる特徴がある。ユーザーは問題の対象から関心のある特徴を複数選択することができ、MAP-Elites はそれらを変量とする次元の特徴空間を任意の粒度で分割し、それぞれのセルで最も適応度の高い解を出力する。MAP-Elites は特徴空間のより広い領域で適応度の高い個体を並列的に計算し、その結果を元に新たな解を生成していくため、既存の進化的アルゴリズムと比較して局所解に陥りにくいことが報告されている。

MAP-Elites の疑似コードを Algorithm 1 に示す。 \mathcal{P}, \mathcal{X} はそれぞれ適応度と個体のマップを表す。マップの各セルには探索された個体の中で最も適応度の高い個体が入る。各イテレーションで個体群から無作為に選択した個体に random variation (mutation and crossover) を適用し、変異後の特徴量に対応するセル上の個体と比較、より適応度が高ければ置換することで進化を進める。

3. NSE 問題

本研究では論文 [5] [6] に倣い NSE 問題を定式化した。基盤ネットワークおよびネットワークスライス要求はそれぞれ無向グラフとしてモデル化する。無向グラフは一般に $G = (V, E)$ の順序対で表され、 V は頂点集合、 E は辺集合である。また、2 頂点 $v_i, v_j \in V$ 間の辺を $e_{i,j} = (v_i, v_j)$ と表記する。基盤ネットワークをモデル化した無向グラフを $N_S = (V_S, E_S)$ とする。ここで、基盤ネットワークの頂点は UE とクラウドノードに類別され、それぞれを V_U, V_C とすると、 $V_S = V_U \cup V_C$ である。また、 E_S は基盤ネットワーク上の有線・無線リンクを表し、UE 間の

接続は存在しない。すなわち、 $\forall e_{i,j} \in E_S (v_i \in V_C \vee v_j \in V_C)$ である。クラウドノードと有線・無線リンクはそれぞれ有限の資源を持ち、本研究ではノードの CPU リソースとメモリ容量、およびリンクのスループットとレイテンシを考慮する。 $v \in V_C$ の CPU リソースを $C(v)$ 、メモリ容量を $M(v)$ 、また $e \in E_S$ のスループットを $T(e)$ 、レイテンシを $L(e)$ とする。ただし、全てのクラウドノードはレイテンシがゼロかつスループットが無限のループ辺を持つ。これは、各クラウドノードが複数のノードによって構成され、それぞれが個別にアプリケーションをホストできる状況を想定している。

同様に、要求されるネットワークスライスをモデル化した無向グラフを $N_R = (V_R, E_R)$ とする。ネットワークスライスは基盤ネットワーク上の UE の一部と、クラウドノード上でホストされるアプリケーションで構成される。そのため、このネットワークスライスで使用する UE の集合を $V'_U \subset V_U$ 、アプリケーションの集合を V_A とすると、 $V_R = V'_U \cup V_A$ である。 E_R は UE およびアプリケーション間の仮想リンクの集合であり、UE 間の仮想リンクは存在しない。すなわち、 $\forall e_{i,j} \in E_R (v_i \in V_A \vee v_j \in V_A)$ である。ネットワークスライスのアプリケーションおよび仮想リンクはそれぞれ基盤資源に対応する資源要求量の属性を持つ。

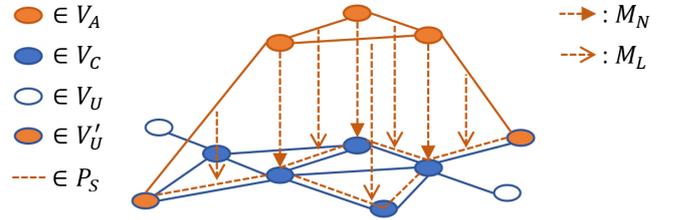


図2 NSE問題の概要図

NSE 問題では複数のネットワークスライス要求について、その受容の可否と割り当て方法を決定する。ある時点における要求された N 個のネットワークスライスの集合を \mathcal{R} とすると、その各要素 $N_R^k = (V_R^k, E_R^k) = (V'_U \cup V_A^k, E_R^k) (1 \leq k \leq N)$ について、受容の可否を以下の写像 $M_A: \mathcal{R} \rightarrow \{0, 1\}$ を用いて表す。また、受容されたネットワークスライスの番号の集合を $N_A = \{k \mid M_A(N_R^k) = 1\}$ とする。

$$M_A(N_R^k) := \begin{cases} 1 & \text{if } N_R^k \text{ is accepted} \\ 0 & \text{otherwise} \end{cases}$$

受容されたネットワークスライスの集合 $\mathcal{R}_A = \{N_R^k \in \mathcal{R} \mid M_A(N_R^k) = 1\}$ の各要素 N_R^k について、アプリケーションのクラウドノードへの割り当てを写像 $M_N^k: V_A^k \rightarrow V_C$ と表す。また、一般化のため次の写像 $M_N^k: V_R^k \rightarrow V_S$ を定義する。なお、一般的な VNE 問題におけるノード割り当てとは異なり、写像 M_N^k は単射である必要はない。

$$M_N^k(n) := \begin{cases} M_N^k(n) & (n \in V_A^k) \\ n & (n \in V'_U) \end{cases}$$

仮想リンクから基盤経路への割り当てを写像 $M_E^k: E_R^k \rightarrow P_S$ と表す。ここで、 P_S は基盤ネットワーク上の全ての経路の集合であり、経路は頂点の有限な列である。経路 p の k 番目の項（頂点）は p_k 、長さは $|p|$ と書く。また、辺 $e_{i,j}$ が経路 p 上にあることを $e_{i,j} \stackrel{\text{in}}{\sim} p$ 、経路 p 全体のレイテンシを $L_P(p)$ と表記する。すなわち、以下のように定義される。

$$e_{i,j} \stackrel{\text{in}}{\sim} p \stackrel{\text{def}}{\iff} \exists q ((p_q = v_i \wedge p_{q+1} = v_j) \vee (p_q = v_j \wedge p_{q+1} = v_i))$$

$$L_P(p) := \sum_{k=1}^{|p|-1} L((p_k, p_{k+1}))$$

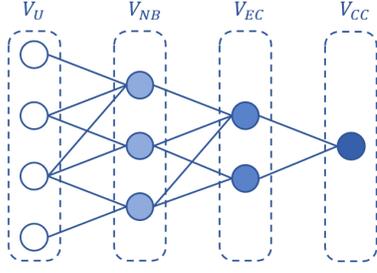


図3 層型ネットワークモデル

この時、要求されたネットワークスライスと割り当てられたネットワークスライスは同一のトポロジを持つ必要があり、この制約は条件式 (1) で表される。

$$\forall k \in \mathbb{N}_A \forall e_{i,j} \in E_R^k \left(M_L^k(e_{i,j})_1 = M_N^k(v_i) \wedge M_L^k(e_{i,j})_{|M_L^k(e_{i,j})|} = M_N^k(v_j) \right) \quad (1)$$

また、各クラウドノードはホストするアプリケーションの資源要求を満たす必要があり、この制約は条件式 (2) で表される。

$$\forall c \in V_C \forall Attr \in \{C, M\} \left(\sum_{k \in \mathbb{N}_A} \sum_{a \in V_A^k(c)} Attr(a) < Attr(c) \right) \quad (2)$$

where $V_A^k(c) = \{a \in V_A^k \mid M_N^k(a) = c\}$

同様に、各基盤リンクはその上に確立される仮想リンクの資源要求を満たす必要があり、この制約は条件式 (3) および (4) で表される。

$$\forall e_{i,j} \in E_S \left(\sum_{k \in \mathbb{N}_A} \sum_{e_r \in E_R^k(e_{i,j})} T(e_r) < T(e_{i,j}) \right) \quad (3)$$

$$\text{where } E_R^k(e_{i,j}) = \{e_r \in E_R^k \mid e_{i,j} \stackrel{\text{in}}{\sim} M_L^k(e_r)\}$$

$$\forall k \in \mathbb{N}_A \forall e_r \in E_R^k \left(L_P(M_L^k(e_r)) < L(e_r) \right) \quad (4)$$

NSE 問題は、これらの条件式を満たす写像 M_A, M_N^k, M_L^k を決定する問題として定義される。以降に本研究の評価で想定した NSE 問題に関するより詳細なモデルについて説明する。

3.1 基盤ネットワークモデル

本稿では基盤ネットワークの構造を文献 [5] [9] に倣い、UE、Node B、エッジクラウド、セントラルクラウドの4層からなる層型ネットワークとする。UE 以外はクラウドノードに分類され、順に資源量は大きくなり、基盤ネットワークに占めるノード数は小さくなる。また、リンクは隣接する層間を接続するものに限られ、同一層内の接続および隣接しない層間の接続はない。すなわち、Node B、エッジクラウド、セントラルクラウドの集合をそれぞれ V_{NB}, V_{EC}, V_{CC} とするとき、 $V_C = V_{NB} \cup V_{EC} \cup V_{CC}$ であり、 $\forall e_{i,j} \in E_S ((i \in V_U \wedge j \in V_{NB}) \vee (i \in V_{NB} \wedge j \in V_{EC}) \vee (i \in V_{EC} \wedge j \in V_{CC}))$ である。

文献 [5] では UE と Node B 間の無線接続を他のノード間の接続と同様にモデル化しているが、実際には同一の Node B に接続する UE 間では利用可能な帯域は共有されるため、基盤資源であるスループットも共有されるようにモデル化すべきである。本稿では UE 層と Node B 層の間に RAN を表現する層 V_{RAN} を追加し、同一の Node B に

接続する UE は同一の RAN を介して接続することで利用可能なスループットが共有されるように変更した。このとき、 $|V_{NB}| = |V_{RAN}| \wedge \forall v_i \in V_{NB} \exists! v_j \in V_{RAN} (e_{i,j} \in E_S)$ である。また、Node B・RAN 間のリンクは有限のスループットとゼロのレイテンシを持ち、UE・RAN 間のリンクは無限のスループットと一定のレイテンシを持つようモデル化する。

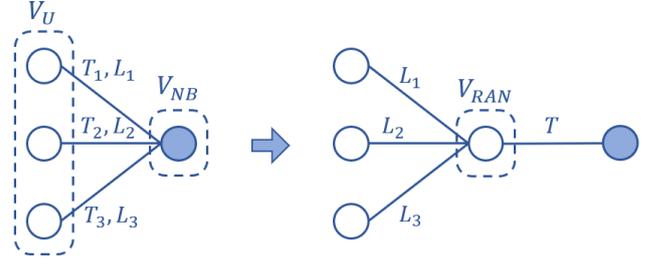


図4 RAN ノードの追加

3.2 ネットワークスライスの到着モデル

ネットワークスライス要求は各クライアントが個別に発行するものであり、それらは相互に独立したイベントであると考えられ、このような過程はポアソン過程としてモデル化することができる。各ネットワークスライスがホストされてからディアクティブートされるまでの時間を寿命と定義する。ここで、ポアソン過程においてイベントの発生間隔は指数分布に従うことが知られている。平均 n 個のネットワークスライスが並列的に要求されるようなモデルは、単位時間当たり平均 λ/n 回発生するイベントの間隔と同じ分布に従う乱数を各スライスの寿命に設定することで実現できる。したがって、各ネットワークスライスの寿命を、母数が λ/n の指数分布に従う乱数として生成する。

4. 提案手法

本稿では、NSE 問題の解である写像 M_A, M_N^k, M_L^k の最適化を MAP-Elites を用いて行う。以降では、ネットワークスライス N_R^k を制御するコントローラは C^k として記述される。

$$C^k = (\mathcal{P}^k, F^k, A^k)$$

ここで、 \mathcal{P}^k は MAP-Elites アルゴリズムにおける個体群を表し、特徴空間上の各セルが格納している個体 (占有者) の遺伝型の集合である。また、 F^k, A^k はそれぞれ \mathcal{P}^k の各セルの遺伝型に対して適応度とその遺伝型が実行可能解であるかどうかを示す値を返す写像である。ここで、実行可能解とは条件式 (1)–(4) を満たす解のことであり、遺伝型 g が実行可能解であるとき $A^k(g) = 1$ 、そうでないとき $A^k(g) = 0$ である。写像 F^k, A^k の詳細は 4.2 節で述べる。 \mathcal{P}^k はアルゴリズム 1 のイテレーションごとに変化するため、ある時刻 t におけるコントローラの状態を $C_t^k = (\mathcal{P}_t^k, F^k, A^k)$ とする。このとき、時刻 t における写像 M_A を次のように定義する。

$$M_A(N_R^k | t) := \begin{cases} 1 & \text{if } \exists i \leq t \exists g \in \mathcal{P}_i^k (A^k(g, i) = 1) \\ 0 & \text{otherwise} \end{cases}$$

この定義は時刻 t までにコントローラ C^k が実行可能解を発見していれば、ネットワークスライス N_R^k が受容されることを意味している。また、個体群 \mathcal{P}_t^k の要素である遺伝型は 4.1 に示す手続きによって M_N^k, M_L^k に対応する写像の組に変換できる。この手続きを写像 dec^k と定義すると、時刻 t における写像 M_N^k, M_L^k は以下のように決定できる。

$$M_N^k(a | t) := M_{N_t}^k(a)$$

$$M_L^k(l | t) := M_{L_t}^k(l)$$

$$(M_{N_t}^k, M_{L_t}^k) = \text{dec}^k \left(\underset{g \in A_t^k}{\text{argmax}} F^k(g, t') \right)$$

where $t' = \max\{i \leq t \mid \exists g \in \mathcal{P}_i^k(A^k(g, i) = 1)\}$,

$$A_t^k = \{g \in \mathcal{P}_t^k \mid A^k(g, t') = 1\}$$

この式は、実行可能解が個体群に存在する最近の時刻 t' における、実行可能解のうち最も適応度の高い個体の遺伝型を復号して時刻 t での解とすることを意味している。

4.1 解のエンコード

遺伝的アルゴリズムを用いて組み合わせ最適化問題を扱う上で、遺伝型の定義とそれを表現型として解に復号する方法の定義が必要である。MAP-Elites においてはパラメータを直接的に符号化して遺伝型とすることが効果的であると知られており [10]、本研究でも NSE 問題の解のパラメータを直接的に遺伝型として用いる。ネットワークスライス N_R^k の最適化におけるパラメータは、写像 M_N^k, M_L^k そのものであり、頂点 $a \in V_A^k$ と辺 $l \in E_R^k$ に対応付ける要素を表す値を並べた組を遺伝型 g とすることができる。頂点については、 $M_N^k(a_i) = c_j (a_i \in V_A^k, c_j \in V_C^k)$ であるとき、 $g_i = j$ とすることで表現する。また、辺については、経路探索の候補を各リンクの端点に対応するノード間の K 個の最短経路に絞り、何番目に短い経路を選ぶかをパラメータとして用いる。すなわち、2 頂点 c_i, c_j 間の第 h 最短経路を $p_{i,j}^h$ 、また E_R^k の各要素について一意の番号 $d \in \mathbb{N}_{\leq |E_R^k|}$ を用いて l_d と表記すると、 $M_L^k(l_d) = p_{M_N^k(a_i), M_N^k(a_j)}^h (l_d = (i, j) \in E_R^k)$ であるとき、 $g_{|V_A^k|+d} = h$ とすることで表現する。すなわち、任意の遺伝型は集合 $\mathcal{G}^k = \mathbb{N}_{\leq |V_C^k|}^{|V_A^k|} \times \mathbb{N}_{\leq K}^{|E_R^k|}$ の要素であり、遺伝的アルゴリズムの探索空間は集合 \mathcal{G}^k である。

逆に、遺伝型 $g \in \mathcal{G}^k$ を写像 M_N^k, M_L^k に復号する場合には、遺伝型 g の各要素を写像によって対応付けられる N^S の要素の表現として解釈すればよい。すなわち、写像 dec^k を以下のように定義する。

$$\text{dec}^k(g) := (\phi_g^k, \psi_g^k)$$

$$\phi_g^k(a_i) := c_{g_i}$$

$$\psi_g^k(l_d) = \psi_g^k((a_i, a_j)) := p_{\phi_g^k(a_i), \phi_g^k(a_j)}^{g_{|V_A^k|+d}}$$

ただし、写像 dec^k によって復号された写像の組 ϕ_g^k, ψ_g^k は必ずしも条件式 (1)–(4) を満たさず、そのようなものは NSE 問題の解ではない。

4.2 適応度

遺伝的アルゴリズムを用いるにあたり、最適化要件を反映するためには目的関数である適応度を適切に定義する必要がある。本手法では複数のネットワークスライスを並列的に最適化するため、コントローラ C^k において個体の評価を行う際には、他のコントローラが制御するスライスはその時点でそれらが決定した解によって埋め込まれているべきである。時刻 t における $c \in V_C$ の残余資源量 R_N および $e \in E_S$ の残余スループット R_L は以下のように定義される。

$$R_N(c, \text{Attr}, k, t) := \text{Attr}(c) - \sum_{n \in N_A^k} \sum_{a \in V_A^n(c|t)} \text{Attr}(a)$$

$$R_L(e, k, t) := T(e) - \sum_{n \in N_A^k} \sum_{e_r \in E_R^n(e|t)} T(e_r)$$

where $\text{Attr} \in \{C, M\}$, $N_A^k = \{n \mid n \neq k \wedge M_A(N_R^n | t) = 1\}$,

$$V_A^n(c | t) = \{a \in V_A^n \mid M_N^n(a | t) = c\},$$

$$E_R^n(e | t) = \{e_r \in E_R^n \mid e \stackrel{\text{in}}{\sim} M_L^n(e_r | t)\}$$

また、写像の組 $\text{dec}^k(g)$ に基づいてネットワークスライス N_R^k を埋め込んだ際の $c \in V_C$ への要求資源量 Q_N および $e \in E_S$ への要求スループット Q_L を定義する。

$$Q_N(c, \text{Attr}, k, g) := \sum_{a \in V_A^k(c|g)} \text{Attr}(a)$$

$$Q_L(e, k, g) := \sum_{e_r \in E_R^k(e|g)} T(e_r)$$

where $V_A^k(c | g) = \{a \in V_A^k \mid \phi_g^k(a) = c\}$,

$$E_R^k(e | g) = \{e_r \in E_R^k \mid e \stackrel{\text{in}}{\sim} \psi_g^k(e_r)\}$$

NSE 問題におけるもっとも基本的な要件は条件式 (1)–(4) を満たすことであり、これらの要件に違反するほど適応度が低下するように定義することで個体群は条件式を満足するように進化する。条件式 (1) は 4.1 節で述べた遺伝型の定義により満足されるため、条件式 (2)–(4) についてそのような値を定義すればよい。したがって、要求資源量に対する残余資源の不足量をペナルティ項 P^k として以下のように定義する。なお、条件式 (4) は他のスライスの埋め込みに対して独立であり最優先で満足すべき条件であるため、レイテンシに関する項が相対的に大きくなるように重み付けをするべきである。このため、定数 μ を十分に大きく設定することで、レイテンシ制約が優先的に達成されるように進化を方向づける。

$$P^k(g | t) := \sum_{X \in \{C, M\}} \sum_{c \in V_C} \max\{Q_N(c, X, k, g) - R_N(c, X, k, t), 0\}$$

$$+ \sum_{e \in E_S} \max\{Q_L(e, k, g) - R_L(e, k, t), 0\}$$

$$+ \mu \sum_{e_r \in E_R^k} \max\{L_P(\psi_g^k(e_r)) - L(e_r), 0\}$$

ここで、 $P^k(g | t) = 0$ であるとき、写像の組 $\text{dec}^k(g)$ は NSE 問題の実行可能解である。したがって、写像 A^k は以下のように定義される。

$$A^k(g, t) := \begin{cases} 1 & \text{if } P^k(g | t) = 0 \\ 0 & \text{otherwise} \end{cases}$$

また、NSE 問題における基本的な最適化要件としてスライスのホストにかかるコストの低減がある。これは資源要求量の総和を最小化することで達成されるが、アプリケーションによるクラウドノードの資源の消費量は一定であるため、仮想リンクによるスループットの消費のみを考慮すればよい。したがって、これらの要件を反映した適応度は以下のように構成できる。

$$-\rho P^k(g | t) - \sum_{e \in E_S} Q_L(e, k, g)$$

ここで、 ρ は定数でありこれを十分に大きく設定することで、個体群は制約条件の達成を優先して進化する。

しかし、この適応度の定義は他のスライスを考慮しておらず N_R^k に関する個別最適化の範疇に留まっており、全体最適化を達成できない。このため、非利己的な評価値を別に定義し適応度に組み込むことで、他のスライスを考慮した協調的な動作を実現し全体最適化を行う。

4.2.1 Load Balancing

基本的に他のスライスが受容されやすくなるためには、ボトルネックとなるような残余資源量が少ない要素を減らすことが重要である。これは、残余資源が潤沢である要素を優先的に利用することで実現できる。すなわち、スライスに割り当てた要素の残余資源量の総和を適応度に組み込めば、そのような解が優先的に選択される。したがって、写像 B^k を以下のように定

義する。

$$B^k(g | t) := \sum_{X \in \{C, M\}} \sum_{a \in V_A^k} R_N(\phi_g^k(a), X, k, t) + \sum_{e_r \in E_R^k} \sum_{e \in E_S(\psi_g^k(e_r))} R_L(e, k, t)$$

where $E_S(p) = \{e \in E_S \mid e \stackrel{\text{in}}{\sim} p\}$

4.2.2 他スライスの資源要求の考慮

$M_A(N_R^k | t) = 1$ であるときには、コントローラ C^k は他の受容されていないスライスからの需要が集中している基盤ネットワークの要素を利用しない実行可能解を探索することで、より多くのスライスが受容されることが期待できる。このとき、各コントローラにおいて最も適応度の高い個体は、ペナルティ項が小さいためその時点の資源状況に最も適合している埋め込みであり、将来的に受容される可能性が高い。したがって、このような埋め込みにおける要求資源量を、基盤ネットワークの要素の需要として設定する。すなわち、時刻 t における $c \in V_C$ および $e \in E_S$ の需要を表す値をそれぞれ D_N, D_L として以下のように定義する。

$$D_N(c, k, t) := \sum_{X \in \{C, M\}} \sum_{n \in \mathbb{N}_R^k} \sum_{a \in V_A^n(c|t)} \text{Attr}(a)$$

$$D_L(e, k, t) := T(e) - \sum_{n \in \mathbb{N}_R^k} \sum_{e_r \in E_R^n(e|t)} T(e_r)$$

$$\text{where } \mathbb{N}_R = \{n \mid M_A(N_R^n | t) = 0\},$$

$$V_A^n(c | t) = \{a \in V_A^n \mid \phi_{g_t^n}(a) = c\},$$

$$E_R^n(e | t) = \{e_r \in E_R^n \mid e \stackrel{\text{in}}{\sim} \psi_{g_t^n}(e_r)\}, g_t^n = \underset{g \in \mathcal{P}_t^n}{\text{argmax}} F^k(g, t)$$

コントローラ C^k は、これらの値が小さい基盤ネットワークの要素を利用する実行可能解を優先的に探索したく、これはスライスに割り当てた要素の需要の総和を最小化するように進化するということである。ここで、写像 D^k を以下のように定義する。

$$D^k(g | t) := \sum_{a \in V_A^k} D_N(\phi_g^k(a), k, t) + \sum_{e_r \in E_R^k} \sum_{e \in E_S(\psi_g^k(e_r))} D_L(e, k, t)$$

where $E_S(p) = \{e \in E_S \mid e \stackrel{\text{in}}{\sim} p\}$

4.2.3 協調的制御

以上で定義した項を用いて、適応度を返す写像 F^k を以下のように定義する。

$$F^k(g, t) := - \sum_{e \in E_S} Q_L(e, k, g) - \rho P^k(g | t) + \epsilon B^k(g | t) - M_A(N_R^k | t) \cdot \delta D^k(g | t)$$

ここで、 ϵ, δ は定数であり、 ϵ は第三項が他の最適化要件に影響を及ぼさないようにするために十分小さく設定すべきである。また、右辺の第四項はコントローラ C^k において実行可能解が既に発見されているときのみ有効であり、 δ は十分に大きく設定することで他のスライスを考慮した実行可能解の探索を優先させることができる。

4.3 特徴量空間の設計

MAP-Elites アルゴリズムでは解探索を特徴空間上で行うため、探索空間上に分布した優れた解を効率的に探索するためにはそれらを捉えやすい特徴空間を設計することが重要である。また、動的 NSE 問題において生じる環境変動に対して柔軟に適応するには、環境によって最適な値が異なるような特徴量を変数とすることが有効である。これを踏まえ、本手法では以下の特徴量を利用する。

表 1 比較手法

Notation	Algorithm	Parameter			
		μ	ρ	ϵ	δ
*ME-C	MAP-Elites	10^4	10^2	10^{-5}	10^2
*ME-LB				10^{-5}	0
*ME-GW				0	10^2
ME-I				0	0
GA-C	EGA	10^4	10^2	10^{-5}	10^2
GA-I				0	0
NR	RW-BFS [7]	-	-	-	-

(1) 割当ノードの総資源量

— 各アプリケーションに割り当てられたクラウドノードの物理資源量の総量。3.1 節で定義した基盤ネットワークモデルにおいてはクラウドノードの種類に応じて資源量が大きく異なるため、この特徴量はどの種類のノードを主に利用するかを意味する。そのスライスのサービス要件とその時点での物理資源利用状況によって最適なノード利用は異なると考えられるため、この特徴量において分散して解を探索することは動的 NSE において効果的である。ただし、この値を変数とする次元での解の分布は不均一であり、解探索効率の低下を招くため代理として以下の値 F_1 を用いる。ここで、基盤ノードの番号は Node B、エッジクラウド、セントラルクラウドの順に割り当てられているとする。

(2) 合計レイテンシ

— スライスの各リンクに割り当てられたパスの合計のレイテンシ。この値は主にパスのホップ長に相関を持つが、同じホップ長のネットワークについてもレイテンシの値によって区別することができ、経路選択の多様化を促す。経路選択の多様化は需要が高い資源の利用の回避などにつながるため、動的 NSE において効果的であると思われる。この値においても同様の理由で代理として以下の値 F_2 を用いる。

$$F_1(g, k) := \sum_{i=1}^{|V_A^k|} g_i, F_2(g, k) := \sum_{i=1}^{|E_R^k|} g_{|V_A^k|+i}$$

5. 評価結果

複数のネットワークスライスを個別のコントローラで制御する動的 NSE 問題について、計算機シミュレーションを通して提案手法および比較手法の評価を行った。提案手法において、単位時間あたりに 100 個の解候補を評価し、受容率および受容率の推移を比較する。なお、本研究では並列化などによるプログラムの最適化を行っていないため、実時間的な実行速度についての評価は行わないが、いずれの手法においても解候補の評価プロセスは基本的に同じであるため、実行速度において差はないと考えられる。スライス要求到着モデルのパラメータは $\lambda = 0.2, n = 40$ に設定し、一回のシミュレーション期間は 1000 単位時間とした。また、タイムアウト時間として 100 単位時間を設定しており、各コントローラにおいてこれを超過するとそのネットワークスライス要求は却下される。

評価で用いた手法の一覧を表 1 に示す。ノーテーションの前に*がついているものが提案手法である。表中のパラメータは式 4.2.3 中の定数であり、これらの値に応じて各最適化要件の影響度が変化する。Elitism-based Genetic Algorithm (EGA) は基本的な遺伝的アルゴリズムにエリート選択を用いて進化を安定させたものであり、QD アルゴリズムに対する従来の遺伝的アルゴリズムとして比較で用いる。また、手法 NR は文献 [5] で提案されているヒューリスティクスと同様に、VNE 問題のヒューリスティクスである RW-BFS [7] を本問題に適用したものである。

基盤ネットワークは 3.1 節で定義した層型基盤ネットワー

表 2 基盤ネットワークのパラメータ

	Node Type			
	V_U	V_{NB}	V_{EC}	V_{CC}
$ V $	90	30	10	1
C, M	-	[100, 200]	[200, 700]	[5000, 10000]
	Link Type			
	$E_{U,NB}$	$E_{NB,EC}$	$E_{EC,CC}$	
$ E $	[1, 3]	[2, 6]	[1, 1]	
T	[50, 80]	[80, 150]	[200, 500]	
L	[3, 7]	[3, 5]	[2, 4]	

表 3 サービスタイプごとのスライス要件

	Service Type		
	URLLC	eMBB	mMTC
$ V'_U $	[1, 10]	[1, 10]	[15, 30]
$ V_A $	[1, 5]	[1, 10]	[1, 5]
C, M	[3, 15]	[10, 40]	[1, 3]
T	[5, 15]	[10, 20]	[1, 5]
L	[10, 30]	[25, 50]	[50, 100]

表 4 受容されたリクエスト数

	ME-C	ME-LB	ME-GW	ME-I	GA-C	GA-I	NR
Mean	123.73	123.27	122.57	121.63	117.80	115.70	97
SD	3.02	3.02	3.03	2.68	3.00	3.27	-

クモデルに基づいて生成する。シミュレーションで用いた基盤ネットワークに関する変数の設定を表 2 に示す。なお、表中の $E_{A,B}$ は V_A, V_B の要素間のリンクの集合であり、 $|E|$ は V_A の各要素と接続される V_B の要素の個数を表す。また、 $E_{U,NB}$ の要素は実際には V_{RAN} を介する。

シミュレーションにおけるネットワークスライス要求に関するパラメータ設定を表 3 に示す。

一回のシミュレーションを通して受容されたネットワークスライス数を表 4 および図 5 に示す。NR 以外の手法については、同じ設定のシミュレーションを 30 回試行した結果をまとめており、またいずれのシミュレーションにおいてもネットワークスライス要求とその到着タイミングは同一である。MAP-Elites アルゴリズムを用いた手法群の平均受容個数は他の手法と比較していずれも多く、動的 NSE 問題において最も効率的に実行可能解を探索していることが分かる。また、その中でも提案手法 ME-C の平均受容個数が最も多く、非利己的な動作による協調的な制御によって最大受容個数が向上することを示している。また、単位時間ごとの平均受容個数の推移を図 6 に示す。どの時刻においても基本的に同様の結果が示されており、複数のネットワークスライスの並列最適化における MAP-Elites アルゴリズムおよび協調的な動作の有効性が確認できる。

6. おわりに

様々な要件を持つ複数のネットワークスライスを同時に扱う動的 NSE 問題について、各ネットワークスライスを協調的に動作する個別のコントローラで並列的に最適化し、全体最適化する手法を提案した。遺伝的アルゴリズムの目的関数の設計の自由度を利用し、協調的な動作を実現するための非利己的な指標を適応度に導入することで、個別のコントローラでの最適化でありながら全体最適化を可能にした。計算機シミュレーションを通して比較評価を行い、提案手法の優位性およびその最適化項ごとの性質を明らかにした。評価シナリオでの受容個数の差は顕著ではなかったが、実行可能解の発見速度の点での差を

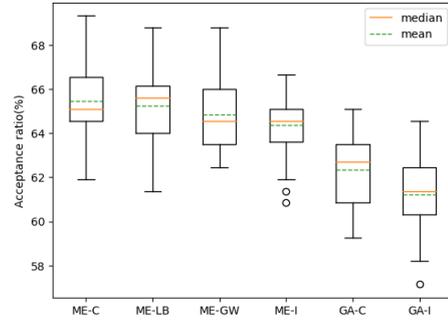


図 5 受容されたスライス数

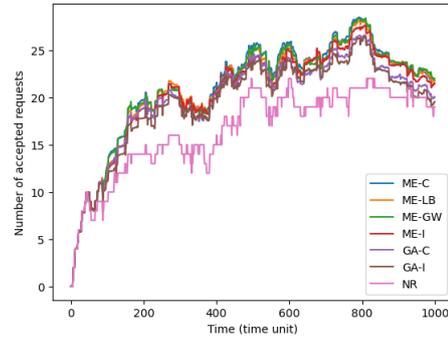


図 6 平均受容数

現在確認できており、今後、受容個数の差がより大きく現れる条件を明らかにしていく。

文 献

- [1] NGMN, 5G white paper 2, NGMN Alliance, July 2020.
- [2] F. Debbabi, R. Jmal, L.C. Fourati, and A. Ksentini, "Algorithmics and modeling aspects of network slicing in 5G and beyond network: Survey," *IEEE Access*, vol.8, pp.162748–162762, 2020.
- [3] S. Vassilaris, et al., "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol.55, no.8, pp.112–119, 2017.
- [4] H.-T. Chien, et al., "End-to-end slicing as a service with computing and communication resource allocation for multi-tenant 5G systems," *IEEE Wireless Communications*, vol.26, no.5, pp.104–112, 2019.
- [5] K. Ludwig, A. Fendt, and B. Bauer, "An efficient online heuristic for mobile network slice embedding," *Proceedings of Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp.139–143, 2020.
- [6] A. Fendt, C. Mannweiler, L.C. Schmelz, and B. Bauer, "A formal optimization model for 5G mobile network slice resource allocation," *Proceedings of IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp.101–106, 2018.
- [7] X. Cheng, et al., "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol.41, no.2, p.38–47, April 2011.
- [8] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.
- [9] P. Labrador, et al., "5G NORMA network architecture – intermediate report," Jan. 2017.
- [10] D. Tarapore, J. Clune, A. Cully, and J.-B. Mouret, "How do different encodings influence the performance of the map-elites algorithm?," *Proceedings of the Genetic and Evolutionary Computation Conference*, p.173–180, 2016.