

# SDIにおける仮想網制御の進化適応性を高める計算資源増強手法

井上 昂輝<sup>†</sup> 荒川 伸一<sup>†</sup> 村田 正幸<sup>†</sup>

<sup>†</sup> 大阪大学 情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

E-mail: †{k-inoue,arakawa,murata}@ist.osaka-u.ac.jp

**あらまし** SDI 環境構築に向けて国内外で取り組みがなされている。仮想ネットワーク要求の個数が増大に対し、変動が生じることを想定し変動に即応する制御アプローチおよび、制御の適応性を促進する物理資源増強が必要である。VNE 手法については、著者らは SDN/NFV 環境におけるゆらぎ原理にもとづく仮想網制御手法を提案し、デマンド変動への適応性を示してきた。本稿では、SDI 環境における物理計算資源の増強プランニングを検討する。生物の進化適応性に関する知見に着目し、ゆらぎ VNE 手法において、アトラクター構造の変異に伴う新環境への適応能力の最大化を図る。シミュレーション評価では、提案した増強指針でメモリ増強を施した場合、Ad-hoc に増強を行う一般的な方法と比べ、デマンド変動に対するゆらぎ VNE 制御の解発見率が上昇することを示した。

**キーワード** 物理ネットワーク増強, キャパシティプランニング, SDI, ゆらぎにもとづく仮想網制御, SDN, NFV

## An evolvable network resource planning for adaptive virtual network control in software defined infrastructure

Koki INOUE<sup>†</sup>, Shin'ichi ARAKAWA<sup>†</sup>, and Masayuki MURATA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University

1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

E-mail: †{k-inoue,arakawa,murata}@ist.osaka-u.ac.jp

**Abstract** This paper propose a strategy of physical node capacity planning in SDI. The strategy imitates the biological strategy which maintains genotypic evolvability against uncertain environmental changes. Computer simulations show that the proposed strategy improves the convergence ratio of the VNE method as compared with the general ad-hoc strategy.

**Key words** physical network design, capacity planning, SDI, Yuragi-based VNT control, SDN, NFV

### 1 はじめに

モバイル端末の普及やクラウドサービスの進展に伴い、通信ネットワークの新たな利用形態が台頭している。情報ネットワーク・システムを柔軟かつ低コストで運用管理を行うことを目的として、SDI (Software Defined Infrastructure) 環境構築に向けて国内外で取り組みがなされており、例えばマーケットへの導入が進められている IaaS (Infrastructure as a Service) は、インターネット経由で操作するサービスとして仮想化されたサーバーなどのインフラ基盤を提供する利用形態であり、顧客となる企業は必要なだけのインフラ基盤を利用し情報システムを構築および稼動することができる。コンピューターシステムにおける仮想化技術が進展し、サービスのカスタマイズ化が進められつつあるが、ネットワーク仮想化技術の検討は始まったばかりである。近年は、ネットワーク管理者が各種ネットワーク装置の設定を適宜変更する従来方式に代えて、プログラマブルなネットワーク制御を可能とする SDN (Software Defined Networking) および NFV (Network Function Virtualization) が期待され、標準化

も進められている [1,2]。SDN/NFV によって、オンデマンドの資源割り当てや、ユーザー自身によるサービスプロビジョニング、セキュアクラウドサービスなど個々のユーザーや市場の需要に即時対応することが可能となる [3]。

コンピューターシステムの仮想化技術およびネットワークの仮想化技術を導入し、基盤ネットワーク上に仮想ネットワークを埋め込む VNE (Virtual Network Embedding) 問題が広く検討されている [4]。VNE 問題は、サービス要求を VN (Virtual Network) 要求に置き換え、仮想ノードや仮想リンクに対して、基盤ネットワーク上におけるノードの仮想化計算資源や経路の帯域を割り当てる際の適切なマッピング解を求める問題である。また、仮想スライス数や各スライスにおける資源要求量が継続的な増加傾向にある状況においては、適切な物理資源の導入が必須となる。つまり、仮想ネットワークを収容するには、計算ノードにおいては仮想マシン稼働に支障が生じない CPU やメモリ量、経路中のリンクにおいてはトラフィック要求を十分に収容できるだけの帯域が具備されている必要がある。資源利用率が逼迫すると、計算処理遅延やデータ転送にかかる遅延が増大

し、仮想ネットワーク上で提供されるサービスの性能劣化に陥るためである。ネットワーク資源の増強指針として、増強時点の需要または予測に基づく将来需要に対して性能最適化を図るのが一般的である。これは、現環境における遅延性能を追求して資源増強した場合、増強時点のデマンドに対する遅延性能は最良となり、その結果、軽微な変化であれば、多少の遅延増大は生じるものの、十分に許容範囲に収まるという考え方にもとづくものである。

しかし、このようなキャパシティ設計と従来の集中管理型の最適化アプローチによる VNE では、大きなデマンド変動や環境変化への対応は困難である。仮想ネットワーク要求の個数が増大し、膨大な数のユーザーにカスタム化したサービスを提供することが求められる場合には、変動が生じることを想定し変動に即応する制御アプローチおよび、制御の適応性を促進する物理資源増強が必要である。VNE 手法については、著者らは SDN/NFV 環境におけるゆらぎ原理にもとづく仮想網制御手法(以降、ゆらぎ VNE 手法)を提案し、ネットワーク全体を詳細に把握することなく稼働する VNE 制御の実現に取り組んできた [5]。ゆらぎ原理はアトラクター選択モデルで記述され、アトラクターと呼ばれる幾つかの解候補に引き込む決定的な作用とノイズによるランダムな振る舞いを持って解空間上で良好な状態を探索し環境への適応を図る。ゆらぎ VNE 手法は、遅延の観測値を一定以下に抑えることを目的とした制御により、ネットワーク全体の詳細な情報収集のボトルネックを解決した上で、当該 VN の仮想資源デマンド量の変動に順応可能である。しかし、物理資源の増強に関する方針は検討されていない。そこで本稿では、SDI 環境における物理ネットワーク基盤のノード計算資源のキャパシティ増強プランニングを検討する。資源増強の指針として、現在の最良の性能よりも、将来にわたって良好な性能を維持することを目指す。生物が、遺伝型の変異がもたらす表現型の多様性によって未知なる環境変動に対する進化適応性を高めるという知見 [6] に着目し、ゆらぎ VNE 手法において、アトラクター構造の変異に伴う新環境への適応能力が最大化されるように物理計算資源を増強することを目的とする。

提案する増強手法には以下のような特性が期待される。

- 増強箇所を決定する際に正確なデマンド予測を必要としない。
- アトラクター構造の微小な修正で多種類の解候補へのアクセスが可能であり、適解の探索が促進され素早く適応を図ることができる。
- より多様な環境変動への適応可能性を持つ。VN 候補の多様性を増すためである。

シミュレーション評価では、物理ノードのメモリ資源を増強した上で、当該 VN のデマンド変動を与える。ゆらぎ VNE 制御を実行し、有限の制御ステップ数で多くのデマンド変動パターンに適応できることを示す。

本稿の構成は以下の通りである。2 章で対象環境である SDI のサービスモデルを紹介し、ゆらぎ VNE 手法を簡単に説明する。3 章で、進化適応性を高める資源増強手法を提案し、4 章でその性能評価について説明する。最後に 5 章でまとめと今後の課題を議論する。

## 2 SDI (Software Defined Infrastructure)

### 2.1 SDI 環境での仮想ネットワークサービス

複数の VN 要求を受け付けるシステム概要を図 1 に示す。カ

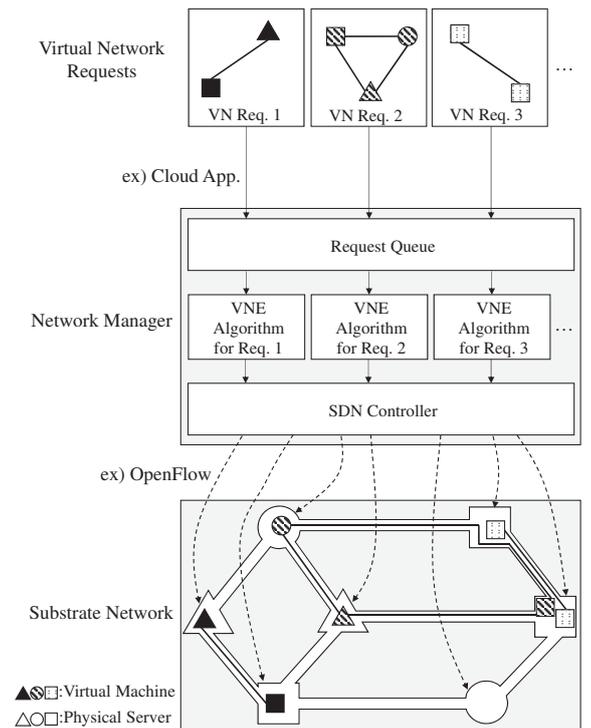


Figure 1: サービス概要

スタマーは所望の VN をクラウドアプリケーション等のインターフェースを通して要求することができる。ネットワーク管理者は VN 要求に対して何らかのアルゴリズムに従い仮想ネットワークの埋め込み位置を決定する。その結果に基づき、基盤ノードに仮想マシンを立ち上げるなどして仮想資源を割り当てる。そして、SDN コントローラーは OpenFlow 等のプロトコルを用いて基盤ネットワークのフォワーディングルールを設定し仮想ネットワークを構築する。基盤ネットワークは物理ノードと物理リンクから構成される。最も一般的な IaaS フレームワークである OpenStack では、IaaS コンポーネントとしてのノードは、CPU とメモリを備えた計算用のサーバ、スイッチング機能のみを提供するフォワーディング用のスイッチ、大容量の HDD を載せたストレージに分別される。さらに、サポートされる OS やストレージタイプなどによってノードの特性 (Features) は多様である [7]。実運用に際しては個々の特性と埋め込む仮想マシンとの整合性を厳密に考慮する必要がある。しかし、本研究ではこれらのノードの違いや特性の違い、地理的な区分等を“ノード属性”として抽象化したモデルを提案する。同様に各リンクの資源容量として帯域幅が設定される。

VN 要求は、仮想ノードと仮想リンクから構成される。仮想ネットワーク埋め込みでは、複数の VN 要求を満たすように、仮想ノードと仮想リンクを共有の物理資源ネットワーク上に配置する。仮想ノードはノード属性が一致する物理ノードのいずれかに配置する。仮想ノード間の仮想リンクに関しては、1 本または複数本の物理リンクを通るパス上にフローとして埋め込まれる。

### 2.2 ゆらぎ原理にもとづく VNE 手法

遅延やリンク利用率、その他性能指標を活性度として取り入れることで、所望の性能を達成することができる。本稿では性能指標として、最大遅延、つまり仮想ノード対地間の End-to-End 遅延のうち最大のものを用いる。ゆらぎにもとづく VNE 手法の定式化を説明する。

ノード属性  $a(= 1, 2, \dots, A)$ 、物理ノード  $m(= 1, 2, \dots, M)$  と

し、遺伝子発現量  $X = \{x_{am}\}$  と定義する。本手法は、属性  $a$  の仮想ノード要求をどの物理ノード  $m$  に埋め込むかを決定する。アトラクター選択モデルの微分方程式 (1) を適用するにあたり、遺伝子  $an(k = 1..A, m = 1..M)$  を遺伝子  $i(i = 1..AM)$  のように展開し、単一の添字  $i$  を用いた表記方法に変換する。

$$\frac{dx_i}{dt} = \alpha \left\{ \varsigma \left( \sum_j W_{ij} x_j \right) - x_i \right\} + \eta \quad (1)$$

式 (1) の第 1 項の  $\varsigma \left( \sum_j W_{ij} x_j \right) - x_i$  は、アトラクターを持つ制御構造であり、関数  $\varsigma(z)$  は式 (2) で定義されるシグモイド関数である。

$$\varsigma(z) = \tanh\left(\frac{\mu}{2}z\right) \quad (2)$$

式 (2) の  $\mu$  はシグモイド関数において 2 値が切り替わる閾値近傍における傾斜を表すパラメーターである。このアトラクターを持つ制御構造は、遺伝子の相互作用によって決まり、その相互作用は制御行列  $W_{ij}$  で表される。式 (1) の第 2 項の  $\eta$  はゆらぎ項である。

$x_i$  の発現の閾値を  $\theta$  とし、 $x_i \geq \theta$  のとき遺伝子  $i$  は発現し、 $x_i < \theta$  のとき遺伝子  $i$  は発現しない。ここで、発現/非発現を  $X_i = \text{Sign}(x_i - \theta)$  により表記する。発現状態では  $X_i = 1$ 、非発現状態では  $X_i = 0$  となる。つまり、遺伝子  $i$  に対応するノード属性  $a$  と物理ノード  $m$  について、 $X_i = 1$  のときは物理ノード  $m$  がノード属性  $a$  の仮想ノードを埋め込む候補となり、 $X_i = 0$  のときは物理ノード  $m$  に仮想ノードを埋め込まない。

仮想リンク要求それぞれに対して最短ホップ長の経路を通るようにフローを設定し仮想リンク埋め込みが完了する。ただし、ルーティングについては最短ホップ長経路以外のポリシーを用いることも可能である。

活性度の設定方法について説明する。活性度  $\alpha$  はシステムの状態の良さを表す関数でシステムからのフィードバックである。利用シーンにもよるが、一般に埋め込んだ VN において通信遅延が小さく抑えられることが望ましい。ある仮想ノード  $i$  から仮想ノード  $j$  への遅延を  $d_{ij}$  とする。そこで、仮想ノードと仮想リンクを埋め込んだ際の最大遅延  $d^{max} = \max_{i,j} \{d_{ij}\}$  を用いて式 (3) のように定義する。ただし、 $\gamma$  および  $\delta$  はパラメータである。

$$\alpha = \frac{\gamma}{1 + \exp(\delta(d^{max} - \theta))} \quad (3)$$

アトラクター構造について説明する。式 (1) 中の制御行列  $\mathbf{W}$  はアトラクター構造を表現しており、仮想ノード配置の幾つかの安定点を保持している。その安定点をアトラクターと予備、それぞれのアトラクターは  $\mathbf{x} = (x_1, \dots, x_i, \dots, x_{KN})$  と定義される。ただし、 $x_i \in \{-1, 0, 1\}$  とする。物理ノード  $n$  が属性  $k$  の仮想ノードを収容する候補となる時、 $x_{kn}$  は 1 に設定される。また、物理ノード  $n$  が属性制約により属性  $k$  の仮想ノードを収容できないとき、 $x_{kn}$  は 0 に設定される、それら以外のとき、 $x_{kn}$  は -1 に設定される。

制御行列  $\mathbf{W}$  に保持されるアトラクターの数を  $M$  とすると、アトラクター集合  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M)$  は下式によって制御行列に組み込むことができる。

$$\mathbf{W} = \mathbf{X}^+ \mathbf{X}, \quad (4)$$

ただし、 $\mathbf{X}^+$  は  $\mathbf{X}$  の擬似逆行列である。これにより、ゆらぎ微分方程式 (1) において、活性度が十分高いとき、制御変数  $\mathbf{x}$  は

保持されたアトラクターに引き込まれるように推移する。つまり、現在の状態がアトラクターの 1 つにある時、式 (1) の  $dx/dt$  は 0 に近づき、そのアトラクターに留まる振る舞いをする。

### 2.3 計算資源による制約

ゆらぎ VNE 手法は VN のマッピングを求めるが、資源利用率が逼迫し遅延が制御目標を上回る場合、もしくは、空き資源量が不足し VN を収容できない場合には活性度は 0 に近づき、解の探索が収束しない。物理資源のキャパシティプランニングを適切に行い、予め物理資源を増強しておくことで、そのような VNE 手法の制御失敗を防ぐことが本研究の目的である。本稿では、サービス性能として遅延に着目しており、計算資源としてサーバのメモリの増強を対象とする。メモリ利用率の逼迫時には、ストレージディスク上にメモリ空間を拡張する仮想メモリ機能が働くが、メモリアクセスに比してストレージへのアクセス速度は非常に遅いため計算遅延の増大が見込まれる。メモリ資源を増強することで、仮想メモリ機能の発動を抑制し遅延増大を回避する狙いである。

## 3 進化適応性を高める計算資源増強手法

SDI 環境において、利用可能な残余資源量の減少により特定の物理ノードやリンクが利用不可になるといった環境変化も考慮すべきである。なお残余資源量減少の原因として、例えば機器故障や他 VN の競合による資源枯渇が挙げられる。これらはデマンド量の微変動等よりも重篤な環境変化であり、現行の制御行列 (アトラクターセット) での適応が困難となる恐れがある。ゆらぎ VNE 手法で制御行列の変異により新環境への適応を図るが、その際に新しい適解を発見できるよう、VNE 制御の進化適応性 (“Evolvability”) が高められていることが重要である。そこで、生物が遺伝型進化の過程で進化適応性 (Evolvability) の高い遺伝型 ( $W_{ij}$  に相当) を残すことにより、大きな環境変化に対して環境適応度 (Fitness) を素早く、かつより優れた値に回復する知見 [6] に着想を得る。

### 3.1 生物の環境変動に対する進化適応戦略

遺伝型の進化の課程で大きな環境変動が起こった際に、生物は表現型の分散を増加させることによって、Fitness の早期回復が可能になる。表現型の分散は遺伝型  $J_{ij}$  の変化、ノイズ項  $\eta$  の 2 つの要因によってもたらされる。特に、環境変化直後に遺伝子変化 (Genetic variation) による表現型の分散  $V_g$  が小さく表現型が十分にばらつかない場合、環境適応の回復が難航する。一方、 $V_g$  が十分大きく多様な表現型が出現する場合、新環境に適した形質の獲得が促進され、遺伝型の進化適応性 (Genotypic evolvability) が向上する。つまり、 $V_g$  は進化適応性を示す指標とみなされる。

以下に生物進化のモデル、分散の定義を記す。文献 [6] は  $N$  個の個体からなる個体群の変異、選択淘汰の過程をモデル化している。各個体は  $M$  個の遺伝子で構成され、それぞれの遺伝子  $i(i \in \{1..M\})$  は発現量  $x_i$  を持つ。図 2 に示す発現ダイナミクスによる遺伝子発現量の決定、そして環境適応度に基づく変異、選択淘汰のフローを 1 世代とする。ただし、ここでは個体  $n$  の、発現ダイナミクス  $l$  回目の試行における遺伝子発現量を  $x_i^{n,l}$  と表記する。遺伝子  $i$  は自身の発現量が閾値  $\theta_i$  より大きい場合発現し、小さい場合は発現しない。遺伝子発現パターンである表現型は  $\mathbf{X}^{n,l} = \{X_i^{n,l} | X_i^{n,l} = \text{Sign}(x_i^{n,l} - \theta_i), i = 1, \dots, M\}$  のように定義される。ただし  $\text{Sign}(x)$  は符号関数 (サイン関数、シグナム関数) であり、 $x > 0$  のとき 1、 $x < 0$  のとき -1 として

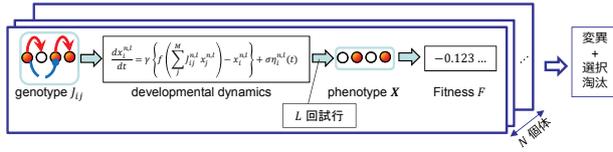


Figure 2: 生物進化モデル 1 世代の動作

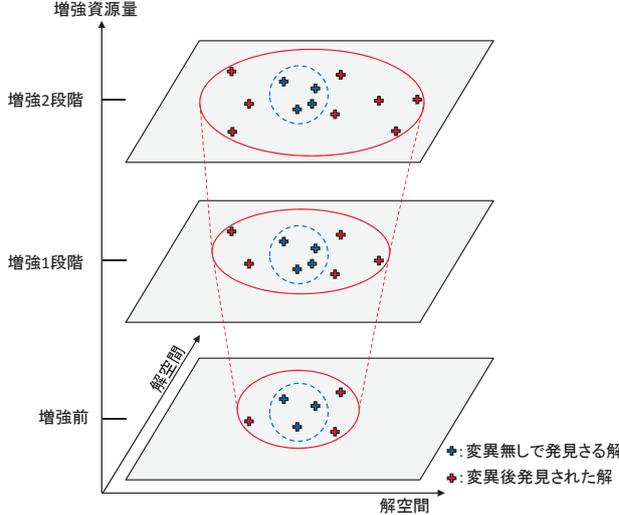


Figure 3: Evolvability の概念図

$x = 0$  のとき 0 を返す。そして、表現型を引数とする Fitness 関数  $F(\mathbf{X})$  により環境適度  $F^{n,l} = F(\mathbf{X}^{n,l})$  が算出される。このとき、遺伝型の変異に係る遺伝子  $i$  の分散  $V_g(i)$  は下式のように表される。

$$V_g(i) = \frac{1}{N} \sum_{n=1}^N (\overline{X_i^n} - \langle \overline{X_i} \rangle)^2 \quad (5)$$

また、 $\overline{X_i^n} = \frac{1}{L} \sum_{l=1}^L X_i^{n,l}$  は  $L$  回試行の平均を表し、 $\langle \overline{X_i} \rangle = \frac{1}{N} \sum_{n=1}^N \overline{X_i^n}$  は個体間の平均を表す。同様に、表現型の環境適度の分散  $V_g$  は以下のように定義される。

$$V_g = \frac{1}{N} \sum_{n=1}^N (F^{n,*} - \langle F^{*,*} \rangle)^2 \quad (6)$$

### 3.2 生物戦略の応用

図 3 にゆらぎ VNE 制御の Evolvability の概念図を示す。図中のマーク (+) は環境変動に応じて発見される解 (VNE) を表している。厳密には初期アトラクターセットに含まれる解と、それらの変異によって初めて発見される解に分けられるが、より広範な探索範囲をもって多種類の解に到達できる場合に Evolvability が高いと言える。つまり、図中の楕円 (実線) の面積を Evolvability を表す指標とみなすこともできる。資源増強を施し Evolvability が高められる場合、解空間内で発見される解の種類が増える。以上を踏まえ、ここでは SDI 環境における Evolvability を次のように解釈する：“制御行列の変異を伴うゆらぎ制御が、新しい解 (VN 配置) を発見できる能力。また、解探索の阻害にならぬよう物理資源が備わっていること。”<sup>(注1)</sup>

本稿の目的は、当初の制御行列では適応困難な環境変動に対処する方策として、制御行列の再構成による新たな環境への適応を促すように物理容量をプランニングすることである。制御行

列の軽微な更新によってより多種類の VN 配置が使われ得る状態になるよう物理ネットワークを設計しておくことで、そのような事態に適応しやすくなる。そこで、遺伝型変異に係る表現型の多様性指標を導入する。ただし、文献 [6] で定義されている  $V_g(i)$  や  $V_g$  をそのまま用いた場合、それぞれは制御変数  $x_i$  やネットワーク性能 (遅延値など) の分散として記述されるため、仮想資源配置の多様性そのものを表す指標と乖離している。そこで、多種類の VN 配置が使われ得ることを表すために、エントロピーによる表現  $H(G)$  を定義する。制御行列  $k \in \{1..K\}$  を用いたオフラインシミュレーションによって得られる制御変数ベクトル  $\mathbf{X}^k = \{X_i^k | i = 1..AM\}$  に対して、その仮想ネットワーク配置を  $g_k = \sum_{i=1}^{AM} 2^{(i-1)} X_i^k$  のスカラー値として識別する。計  $K$  個の制御行列によるオフラインシミュレーション試行により形成される  $K$  個の VN 配置  $\{g_k | k = 1..K\}$  を母集団として、ある VNT  $g$  の出現回数を  $|g|$  とする。 $|g|$  は  $g_k = g$  となる  $k$  の個数である。このとき、その VNT 出現確率を  $p(g) = \frac{|g|}{K}$  とする。VNT 出現確率  $p(g)$  のエントロピーを VNT の多様性を表す指標  $V_g$  に代わって指標  $H(G)$  を下式に定義する。

$$H(G) = - \sum_{g \in \{0..2^i-1\}} p(g) \log_2 p(g) \quad (7)$$

### 3.3 計算資源増強手法

物理 NW 基盤中のどこのリソース、すなわち CPU やメモリなどの計算資源量を増強すべきか、を求める際の方針として、生物が環境適応および可塑性を獲得するための戦略に倣い、 $H(G)$  の増加可能性をオフラインシミュレーションにより試算し、それらが大きくなる箇所に資源を追加する増強方針を考案する。 $H(G)$  を大きくしておくことで、制御行列の軽微な更新に関わらず、より多種類の VN 配置それぞれの使われる機会が増し、新環境に適したアトラクターを構築しやすくなる。

資源追加位置を決定するための ADD アルゴリズムに基づくオフラインシミュレーションの手順を以下に示す。設計時点現在でゆらぎ制御が用いている制御行列を  $\mathbf{W}$  とする。

**Step 1** 物理ネットワーク中のノード  $\forall n \in \{1, \dots, N\}$  に対して以下の手順を行い、当該ノードに資源追加した場合の  $H(G)$  を求め、その値を参考に資源増強するノードを決定する。

**Step 1.1** シミュレーション上で当該ノードに資源を一時的に追加する。

**Step 1.2** 以下のシミュレーションを行い、 $H(G)$  を取得する。

**Step 1.2.1** 制御行列を  $K$  個に変異させ、 $\mathbf{W}^k (k = 1, \dots, K)$  を生成する。ここで変異とは、制御行列に記憶されているアトラクターの一部をランダムに更新することを指す。

**Step 1.2.2** 制御行列  $\mathbf{W}^k$  を用いて、想定デマンド変動に対してゆらぎ原理にもとづく VNE 手法を実行する。 $k = 1, \dots, K$  まで実行し、遺伝型の変化がもたらす多様性の指標  $H(G)$  を算出する。

**Step 2** 資源を仮増強した際に  $H(G)$  が最も大きい値を示した物理ノード  $n$  に資源追加する。

**Step 3** 既定の資源追加量を追加し終わったら終了。そうでなければ Step 1 に戻る。

オフラインシミュレーション Step 1.2.1 では、制御行列  $\mathbf{W}$  を  $K$  個に変異させる (ただし、重複を認める)。ここで、変異とはアトラクターとして記憶している仮想ネットワーク配置の 1 つをランダムに選択し、その配置状態を部分的に変更する。本稿では、仮想スライス  $n$  の属性  $a$  を埋め込む物理ノードを  $m$

(注1) : 生物進化における Evolvability は次のように定義される [8] : 遺伝型変異による探索において新環境に適した表現型を発現できる可能性。ひいては、遺伝型変異や環境変化に応じて発現する表現型が多種類であること。

から  $m'$  に変更することとする。更に、それぞれの制御行列  $k$  に対して想定デマンド変動を与え、発現ダイナミクス試行を行う。制御行列  $k$  における仮想ネットワークのマッピング位置の決定状態を  $\mathbf{X}^k = \{\text{Sign}(x_i^k) | i = 1..AM\}$  とし、決定状態の出現頻度にもとづきエントロピー  $H(G)$  を求める。

## 4 性能評価

資源増強の効果としてゆらぎ VNE 制御の解探索能力を評価する。ゆらぎ VNE 制御は活性度の低下をトリガーとして制御を開始するため、性能指標 (i.e., 遅延性能) が劣化するシチュエーションを生成し評価する必要がある。そこで、当該 VN およびバックグラウンド VNs に加えて新規の VNs 要求を受理・収容する過程における物理ネットワーク資源使用率の増加、その弊害による当該 VN の遅延性能劣化を考慮する。新規に到着する VN リクエストをヒューリスティック VNE 手法 (Greedy VNE) により収容していき、その都度のゆらぎ制御の収束率を評価する。

### 4.1 評価環境

図 4 に示すトポロジーを対象に評価を行う。台形の破線枠はノード属性の区分を表している。各物理ノードの初期キャパシティをノード次数  $\times 30$  とする。また、物理リンクキャパシティは中心性指標  $\times 30$  とする。ここで中心性指標は Edge Betweenness Centrality  $C_B(e) = \sum_{s,t \in V} \frac{\sigma(s,t|e)}{\sigma(s,t)}$  を用いた。ただし、本評価では経路を一意に固定しているため  $\sigma(s,t) = 1$ 、 $\sigma(s,t|e) = \{0, 1\}$  となる。バックグラウンド VN として各ノード、リンクは一様乱数  $[30 : 50]$  だけ使用されている。

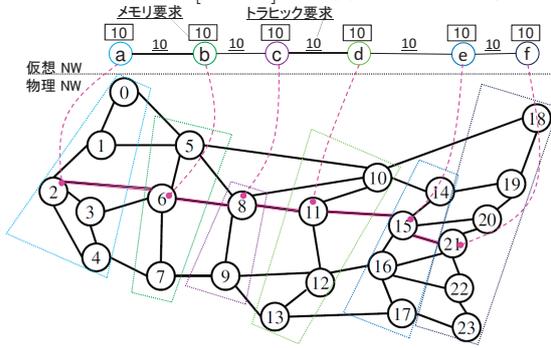


Figure 4: 評価対象ネットワーク

### 4.2 比較手法

比較手法として Ad-hoc 増強手法を用いる。当手法は、ヒューリスティックな VNE 手法の一つである Greedy アルゴリズムにもとづく仮想ノード配置手法 [4] を用いることで予測需要変動に対する性能期待値を高める。Greedy な VNE 手法は設計時点のデマンド要求および空き資源量を用いて、ノード利用率とリンク利用率を小さくするように仮想ノード配置を行う VNE 手法である。Ad-hoc 増強手法は以下の ADD アルゴリズムに従う。

**Step 1** 物理ネットワーク中のノード  $\forall n \in \{1, \dots, N\}$  に対して以下の手順を行い、当該ノードに資源追加した場合のエンド間遅延遅延期待値  $d_*^{(n)}$  を求め、その値を参考に資源増強するノードを決定する。

**Step 1.1** シミュレーション上で当該ノードに資源を一時的に追加する。

**Step 1.2** 以下のシミュレーションを行い、 $d_*^{(n)}$  を求める。

**Step 1.2.1** 想定デマンド変動  $f(f = 1, \dots, F)$  に対して、Greedy VNE(ヒューリスティック VNE 手法) を実行し、出力として最大エンド間遅延を  $d_f^{(n)}$  得る。

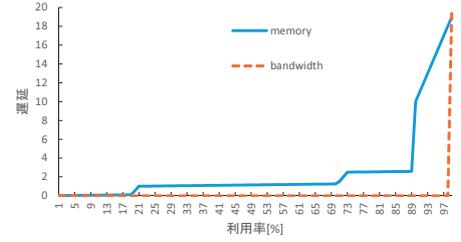


Figure 5: 遅延プロファイル

**Step 1.2.2** 遅延性能指標として、最大エンド間遅延の平均  $d_*^{(n)} = \sum_{t=1}^F d_f^{(n)} / F$  を求める。

**Step 2** 最良 (最小) の  $d_*^{(n)}$  を示したノード  $n$  に資源追加する。

**Step 3** 既定の資源追加量を追加し終わったら終了。そうでなければ Step 1 に戻る。

### 4.3 シミュレーション評価

既定の資源追加量は、メモリ資源 20 を計 4 ノードに追加するとした。提案手法である H(G) 増強はノード  $\{13, 7, 23, 23\}$ 、比較手法である Ad-hoc 増強はノード  $\{11, 11, 15, 10\}$  と求めた。それぞれの増強後ネットワークにおいて、ゆらぎ VNE 手法を実行し適応性を評価する。新規 VNs の収容過程における、デマンド変動に対するゆらぎ VNE 手法の収束率を求める。

#### 4.3.1 VNE 手法の収束判定

ゆらぎ制御を最大 400 ステップ実行する。ゆらぎ制御の活性度の閾値として、最大遅延 (各仮想リンクのエンド間遅延のうち最大のもの) 10 以下を制御目標とする。エンド間遅延は観測値として取得するが、本シミュレーションにおいては下式を用いて算出している。

$$d_{ij} = w_m \sum_{n \in R_{ij}} d_n^m + w_b \sum_{l \in L_{ij}} d_l^b \quad (8)$$

ここで、 $R_{ij}$  はノード  $i$  から  $j$  に至る経路中の物理ノード集合、 $L_{ij}$  は物理リンク集合とする。 $d_n^m$  はノード  $n$  で消費されるメモリの利用率に起因する遅延、 $d_l^b$  はリンク  $l$  の利用率にもとづく遅延を表す。本稿では、 $w_m = w_b = 1$  とし、各ノード、リンクの資源利用率にかかる遅延を図 5 に示す遅延プロファイルで定義する。

ゆらぎ制御の成功判定は、「活性度 0.5 以上の状態で 10 ステップ連続留まる」or「活性度 0.9 以上の状態になる」とする。400 ステップ費やしても収束しなかった場合は制御失敗とする。ただし、40 ステップ経過時点で収束していない場合は、現行の制御行列での適応が困難な環境変動が起こったと判断するとし、以降の 20 ステップ毎に制御行列の変異操作を行う。具体的な操作処理は以下の通りである。

**Step.1** アトラクターとして制御行列に投入されている VNT の中からランダムに 1 つの VNT を選択する。

**Step.2** その VNT を構成する仮想ノードの中からランダムに 1 つの仮想ノードを選択する。

**Step.3** その仮想ノードを配置する先の物理ノードを、ノード属性制約を満たす中からランダムに選び直す。

**Step.4** 以上の処理で得られた VNT を新たなアトラクターとして置き換え、制御行列を再構成する。

新規 VN を以下のように生成し、Greedy VNE 手法により逐次収容していく。

- 6 ノード、6 属性のチェイントポロジーとする。
- 各仮想ノードのメモリ資源要求  $m_i$  は、一様乱数  $[1, 4]$  に属性重み  $w$  を乗算したものとす。  $w$  は各属性の物理ノー

ド数に比例した重みであり、属性  $c$  の重みが 1 になるように正規化している。

- 各仮想リンクの資源要求量  $t_{ij}$  は両端の仮想ノード要求量に応じて  $t_{ij} = (m_i + m_j)/2$  とする。

新規 VN の収容過程において、以下で定義される仮想ネットワーク要求のデマンド変動を 100 パターン生成し、ゆらぎ VNE の収束率を求める。

- 各仮想ノードのメモリ資源要求量:  $m_i \rightarrow m'_i$ 
  - $m_i^{exp} := m_i + U[0, \Delta]$  ( $\Delta$  は増加見込み)
  - $m_i^{tmp} := m_i^{exp} + N[0, (C.V. \times m_i^{exp})^2]$  ( $C.V.$  は変動係数)
  - $m'_i := m_i^{tmp} \times \frac{\sum m_i^{exp}}{\sum m_i^{tmp}}$  (正規化)
- 各仮想リンク上のトラフィックデマンド:  $t_{ij} \rightarrow t'_{ij}$ 
  - $t'_{ij} = (m'_i + m'_j)/2$

#### 4.3.2 結果

評価結果を図 6 に示す。横軸は新規 VN の到着数 (収容数)、縦軸はゆらぎ VNE 制御の収束率である。図 6(a) は  $\Delta = 10, C.V. = 0.1$  のデマンド変動に対する収束率を示している。新規 VN の到着数が 20 まで増えても、ゆらぎ VNE 手法により両増強ともほぼ 100% 収束している。図 6(b) や図 6(c) では、それぞれ  $\Delta = 20, 30$  の変動に対する評価であるが、新規 VN の収容が蓄積するにつれ収束率に低下が見られる。図 6(c) に着目すると、VN 到着数 13, 14 時点で、HG 増強は収束率 85% を維持しているのに対して、Ad-hoc 増強では収束率が更に低下している。VN 到着数 15 以降では両増強とも大きく収束率を低下させるが、Ad-hoc 増強での収束率が 50% となる時点で、HG 増強は収束率を 70% に向上させている。HG を高める増強の結果として、限られた制御ステップ内で、より多くのデマンドパターンに対して適した解を発見できている。

## 5 まとめと今後の課題

本稿では、SDI 環境において予測困難なデマンド変動に対処するための物理ネットワーク基盤のノード計算資源のキャパシティ増強プランニングを検討し、生物が未知なる環境変動に対してアトラクター構造を更新しながら適応性を高める進化適応性の知見を応用した計算資源増強手法を提案した。計算シミュレーション評価では、提案した増強指針で物理ネットワーク中のメモリ増強を施した場合、Ad-hoc に増強を行う一般的な方法と比べ、ネットワーク中の空き資源量が他 VN の収容により枯渇していく過程において、デマンド変動に対するゆらぎ VNE 制御の解発見率が上昇することを示した。

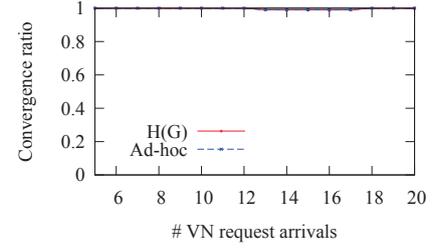
本稿では 1 つの VN のデマンド変動に注目して資源増強の効果を示した。今後の課題として、他の VN に対しても本提案に従って資源増強を行った場合にも同様に適応性の向上が得られるかを検証する。また、制御行列の定義の都合により、当該 VN 要求のトポロジー変更を考慮できていない。制御行列の変異操作を工夫する等により、VN 要求のノード数やトポロジー変更のようなデマンド変動にも対応可能にする方法を検討する。

## 謝辞

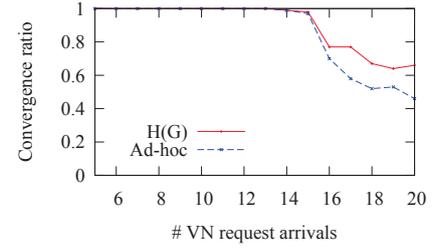
本研究は JSPS 科研費 (A) (No. JP15H01682) の助成を受けたものです。

## References

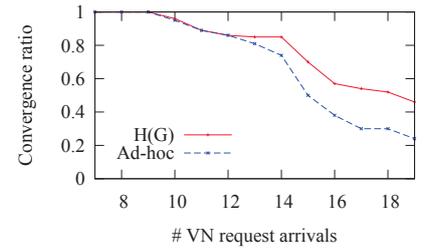
[1] P. Bhaumik, S. Zhang, P. Chowdhury, S. S. Lee, J. Lee, and B. Mukherjee, "Software-defined optical networks (SDONs): A survey," *Photonic Network Communications*, vol. 28, pp. 4–18, June



(a)  $\Delta = 10, C.V. = 0.1$



(b)  $\Delta = 20, C.V. = 0.1$



(c)  $\Delta = 30, C.V. = 0.1$

Figure 6: 新規 VN の収容過程におけるゆらぎ VNE 制御収束率

2014.

[2] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Communications Magazine*, vol. 51, pp. 36–43, July 2013.

[3] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future Internet," *Computer Networks*, vol. 75, Part A, pp. 453–471, Dec. 2014.

[4] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 17–29, Mar. 2008.

[5] K. Inoue, S. Arakawa, S. Imai, T. Katagiri, and M. Murata, "Adaptive VNE method based on Yuragi principle for software defined infrastructure," in *Proceedings of IEEE HPSR*, pp. 191–196, June 2016.

[6] K. Kaneko, "Evolution of robustness and plasticity under environmental fluctuation: Formulation in terms of phenotypic variances," *Journal of Statistical Physics*, vol. 148, pp. 687–705, Sept. 2012.

[7] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, "A novel approach to virtual networks embedding for SDN management and orchestration," in *Proceedings of IEEE NOMS*, pp. 1–7, May 2014.

[8] A. Wagner, "Robustness and evolvability: a paradox resolved," in *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 275, pp. 91–100, Jan. 2008.