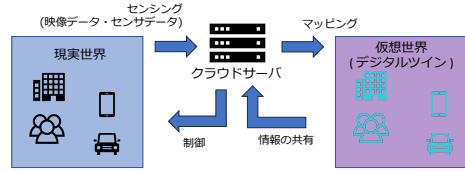


# Adaptability evaluation of distributed video analysis system in dynamic environments based on gene regulatory networks

大阪大学 大学院情報科学研究科  
情報ネットワーク学専攻 村田研究室  
井上 誠志郎

## デジタルツインの活用

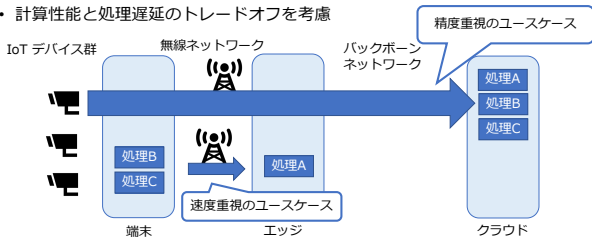
- 現実世界と仮想世界を高度に統合するデジタルツイン技術への期待
  - センシングデータをクラウド上のAIで分析して仮想世界にマッピング
  - 現実のモノやシステムの継続的な監視と最適化が可能
- トラフィック量の増大によるネットワーク負荷への影響が問題視<sup>[4]</sup>
  - デジタルツインの普及に伴いリアルタイムの映像分析処理が大量に発生
    - 消費電力の急増
    - リアルタイム性に支障



[4] D.M. Bot 'in-Sanabria, A.-S. Mihaita, R.E. Peimbert-Garc 'ia, M.A. Ram 'irez-Moreno, R.A. Ram 'irez-Mendoza, and J.d.J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," Remote Sens., vol.14, no.6, pp.1-25, March 2022.

## エッジサーバによる負荷分散

- エッジコンピューティング技術<sup>[5]</sup>
  - データ発生源の近くに配置されたエッジサーバに処理を移行
  - ネットワーク全体のトラフィック量や消費電力を軽減
- 効率的な運用には動的制御が必要
  - 求められるサービス品質やシステム利用状況を考慮して動的に処理を割り当てる必要がある
  - 計算性能と処理遅延のトレードオフを考慮



[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet of Things J., vol.3, no.5, pp.637-646, Oct. 2016.

## 研究の目的とアプローチ

### 研究の目的

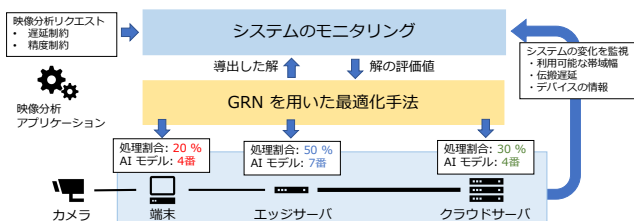
- 遺伝子制御ネットワーク (GRN: Gene Regulatory Network) を利用した分散型映像分析システムの動的環境における適応性の評価

### アプローチ

- GRN の数値モデルによる遺伝的アルゴリズムの拡張手法の提案
  - 生物の遺伝子間の相互作用が形成するネットワーク構造
  - 進化によって過去の環境を記憶した構造を獲得する
- 拡張方式に基づいた分散型映像分析システムの動的制御
  - GRN が記憶した環境を想起することで適応性が向上することを示す

## 映像分析システムの最適化モジュール

- アプリケーションは二つの性能要件を設定して映像分析処理を要求
  - 処理遅延: 映像分析にかかる E2E (end-to-end) 遅延
  - 分析精度: 各デバイスにおける精度の処理割合による加重平均
- 映像分析に利用するデバイス (端末/エッジ/クラウド) の設定を制御
  - 処理割合: デバイス毎に割り当てる処理の量を決定
  - AI モデル: 精度や処理効率が異なるモデルから選択
- リクエストに応じてシステム状態を監視しながら消費電力を最適化



## 組合せ最適化問題の定式化

- 制約条件: 処理遅延  $T_s(t)$  と分析精度  $A_s$

$$T_s(t) = \sum_{d \in D^s} T_s^d(t) + \sum_{n \in N^s} T_n^s(t) \leq T_s^{max}, \forall s \in S$$

デバイスの処理遅延 ネットワークの伝送遅延

$$A_s = \sum_{d \in D^s} A_s^d W_s^d / |D^s| \geq A_s^{min}, \forall s \in S$$

デバイスにおける分析精度の加重平均

- 目的関数: システム全体の消費電力  $E(t)$

$$E(t) = \sum_{d \in D^s} E^d(t) + \sum_{n \in N^s} E^n(t)$$

デバイスの電力消費 ネットワークの電力消費

- 個体の適合度  $F$

$$F = -E - \alpha \sum_{s \in S} (T_s^{max} - T_s) - \beta \sum_{s \in S} (A_s - A_s^{min})$$

消費電力 処理遅延のペナルティ項 分析精度のペナルティ項

- 実測値に基づいた推定モデル<sup>[11]</sup>を利用してネットワークの伝送遅延と消費電力・デバイスの消費電力の算出

[11] H. Shimonishi, M. Murata, G. Hasegawa, and N. Techarasmitkul, "Energy optimization of distributed video processing system using genetic algorithm with bayesian attractor model," in Proceedings of 2023 IEEE 9th International Conference on Network Softwarization (NetSoft), Jun. 2023, pp. 35-43.

分散処理の処理割合とAIモデルの選択を符号化

- セッション  $s$  に関する制御情報を配列で表現
 
$$[W_s^d \forall d \in D^s, M_s^d \forall d \in D^s]$$

$D^s$ : セッション  $s$  が利用するデバイスの集合 (端末/エッジ/クラウド)  
 $W_s^d$ : デバイス  $d$  に割り当てる処理量の重み  
 $M_s^d$ : デバイス  $d$  で稼働する映像分析 AI の種類
- 配列の値を2進数表記に変換した配列を最適化の対象として扱う

遺伝型 [010 101 011 100 111 100]

[ 2 5 3 4 7 4 ]

$W_s^d$ : 処理量  $M_s^d$ : AIの種類

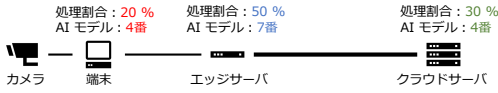


図: 符号化の例

生物の進化を模倣して組合せ最適化問題の解を探索する手法

- 環境変動後の再探索にかかる時間が課題
  - 進化が進むにつれて多様性が失われる
  - 1度の変異での個体の変化が小さいため、収束状態からの再探索に時間を要する

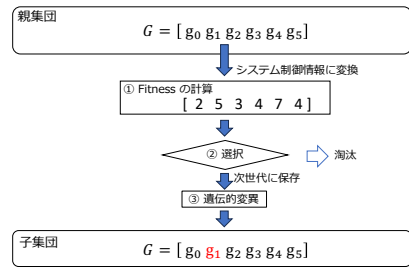


図: 遺伝的アルゴリズムの流れ

- 生物の遺伝子間の相互作用が形成するネットワーク構造
- GRN の特性
  - 遺伝型の小さな変異が表現型を大きく変異させる
  - 特定の表現型を発現しやすい構造に進化する (過去の環境の記憶)
- 動的なシステム制御における GRN への期待
  - 表現型の大きな変異を環境変動時の制御方法の切り替えに利用
  - 適した制御方法を記憶することで既知の環境に高速に適応可能

GRN の数理モデル

遺伝型  $G$ : 遺伝子発現の初期パターン  
 $G = [g_0, g_1, \dots, g_n] \quad (-1 \leq g_k \leq 1)$

GRN の隣接行列  $B$ : 遺伝子間の正負の相互作用を表現  
 $B = \begin{bmatrix} b_{00} & \dots & b_{0n} \\ \vdots & \ddots & \vdots \\ b_{n0} & \dots & b_{nn} \end{bmatrix}$

表現型  $P = [p_0, p_1, \dots, p_n]$ : 遺伝型と隣接行列の乗算により算出  

$$p_i(0) = g_i$$

$$p_i(k+1) = p_i(k) + \gamma_1 \sigma \left( \sum_{j=0}^n b_{ij} p_j(k) \right) - \gamma_2 p_i(k)$$

パラメータ  $k^*$  による  $P(k^*)$  が表現型となる

意味	設定
$\gamma_1$ 相互作用の影響の大きさ	1.0
$\gamma_2$ 発現パターンの減衰率	0.2
$\sigma$ シグモイド関数	tanh
$k^*$ 発現過程の計算回数	10

GA の拡張方法

- Fitness の評価には GRN モデルの発現過程を経て得られた表現型を利用
  - 適した表現型を記憶した GRN を利用することで、有利な解が導出されやすくなる
- 遺伝的変異を遺伝型または表現型のいずれかに適用
  - 遺伝型の変異: 発現過程を経て多くの表現型を変異させ記憶の想起を促す
  - 表現型の変異: 想起した表現型の周辺を探索する

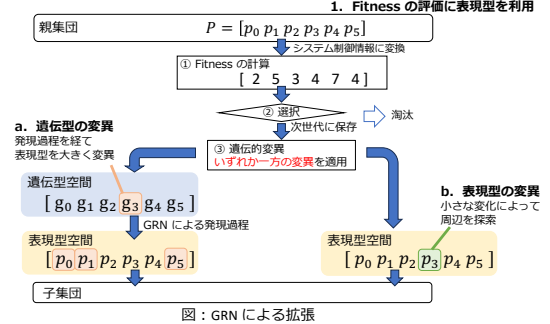


図: GRN による拡張

- 既存の学習アルゴリズム<sup>[14]</sup>を利用して表現型を学習
  - 同時に複数の記憶を保持することができる
- GRN の学習設定
  - 三つのアプリケーションの性能要求を既知の環境として事前に設定
  - それぞれの性能要求における有利な解を選択してGRNに学習
    - GA を複数回実行して見つかった最適解を選択 (セッション数: 4)

表: 性能要求と学習させた表現型 (セッション数: 4)

要求	セッション	遅延制約: s	精度制約: %	記憶させた表現型
A	1	0.05115	52.828	[7,0,0,3,0,5]
	2	0.10679	29.842	[0,0,7,1,5,1]
	3	0.06512	55.647	[0,0,7,2,3,7]
	4	0.09143	30.241	[0,0,5,1,1,5]
B	1	0.12866	38.833	[0,0,5,3,2,4]
	2	0.14595	46.161	[0,0,6,1,4,5]
	3	0.04433	34.157	[6,0,0,4,3,3]
	4	0.09747	46.974	[0,0,6,1,2,5]
C	1	0.04957	50.332	[7,0,0,4,2,5]
	2	0.08061	27.674	[0,0,6,3,5,0]
	3	0.07007	35.058	[0,0,5,0,6,4]
	4	0.13463	55.633	[0,0,7,0,1,6]

[14] R. A. Watson, G. P. Wagner, M. Pavlicev, D. M. Weirich, and R. Mills, "THE EVOLUTION OF PHENOTYPIC CORRELATIONS AND "DEVELOPMENTAL MEMORY ", Evolution, vol. 68, no. 4, pp. 1124–1138, 04 2014. [Online]. Available: https://doi.org/10.1111/evo.12337

- GRN を用いる場合と用いない場合で3通りの環境変動への適応性を評価
  - 評価1: アプリケーションが切り替わる場合
    - 大きな環境変動への適応性を評価
    - アプリケーションは GRN が事前に学習したものを周期的に切り替わる
  - 評価2: 帯域幅が変動する場合
    - 小さな環境変動への適応性を評価
  - 評価3: 学習時に想定していないアプリケーションにおいて帯域幅が変動する場合
    - 未知の環境の場合に適応性がどのように変化するか検証
- 適応性の指標
  - 環境変動の発生から基準の解の発見に要する時間
    - 評価1: GRN に記憶させた最適解の発見
    - 評価2, 3: 制約条件を満たす解の発見 (帯域の変動によって解空間が変化するため基準の解を変更している)
  - 環境変動の発生から 10 秒後と 180 秒後の解の適合度の高さ (評価2, 3)

表: ネットワークの設定

	アクセスネットワーク	バックボーンネットワーク
帯域幅 (変動を想定しない場合)	10~50 Mbps (25 Mbps)	10~500 Mbps (250 Mbps)
伝搬遅延	10 ms	10 ms
1ビットの送信電力	$193 \times 10^{-9}$ W	$60 \times 10^{-9}$ W
1フレームあたりのビット数	470 Kbit	

表: デバイスの設定

	端末	エッジサーバ	クラウドサーバ
CPU	Core i7-8700T	Core i9-10940X	Xeon GOLD 6226R×2
GPU	GeForce GTX1070	RTX A5000	Tesla T4×2

表: 選択可能な映像分析 AI モデルと性能

Model	Accuracy: A%	FLOPS: 0 <sup>th</sup> (1)	M%
YOLOv3-tiny	33.1 %	5.6 B	{0,1,2}
YOLOv3	55.3 %	65.9 B	{3,4,5}
YOLOv3-spp	60.6 %	141.5 B	{6,7}

## 評価 1 の結果: アプリケーションが変化する状況

13

- GRNによる記憶の想起によって高速に解が発見されることを確認
  - 準最適解を発見するまでの時間に大きな差
- 環境変動の発生から次の変動までの60秒以内に準最適解を発見した回数はシステム規模の拡大とともに大きく差が開いた
  - 大規模システムにおける提案手法の有効性が期待される

表: 環境変動発生から準最適解の発見までの時間

	GA	GRN
平均値	49.2 秒	0.21 秒
中央値	60 秒	0.06 秒

表: 環境変動の発生から60秒以内に準最適解を発見した回数

セッション数	GA	GRN
1	90/90	90/90
2	74/90	90/90
4	19/90	90/90

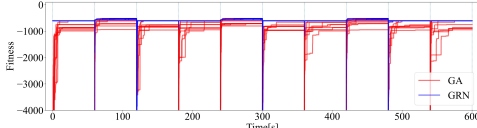


図: 60秒ごとの環境変動を10セット実施した場合の適合度の推移 (セッション数は4)

## 評価 2 の結果: 帯域幅が変動する状況

14

- 提案手法が制約条件を満たす解を短時間で発見しており、適応性が改善
- 環境変動の発生から一定時間経過時の適合度の比較においてもGRNが優位

表: 環境変動発生から制約条件を満たす解の発見時間

	GA	GRN
平均値	119.94 秒	91.86 秒
中央値	180 秒	179.49 秒

表: 環境変動発生から一定時間経過時の適合度の比較

経過時間	GAが優位	GRNが優位	同じ
10 秒	7 / 50	32 / 50	11 / 50
180 秒	10 / 50	24 / 50	16 / 50

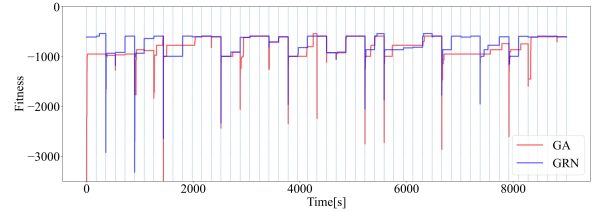


図: 提案手法(GRN)およびGAの適合度の推移  
 ・帯域幅を180秒の時間間隔で49回ランダムな値に変更  
 ・性能要求は事前に設定した要求Aで固定

## まとめと今後の展開

15

### まとめ

- GRNの数理モデルを用いたGAの拡張方式を提案
- 三つの環境変動を想定して適応性を評価
  - 既知の環境下において環境変動に対する適応性が向上
  - 未知の環境においても解の構造が捉えられれば適応性が向上
  - 記憶の想起のために解探索性能が低下

### 今後の展開

- さらに複雑な環境変動への適応性を評価
- 環境変動に合わせてGRNが自律的に学習する方式を検討
  - GRNの学習過程を最適化方式の進化に組み込む

## 評価結果: 未知のアプリケーションが利用される状況

16

- いずれの評価指標においても、GRNよりGAの適応性が高い
- GAが多くの帯域幅に共通で適合度が高い解を発見しているがGRNは発見できていない場合が多い
  - 提案手法では表現型の変異を適用する個体数がGAよりも少ないため親集団の個体の周辺を探索する性能が低下した
- シミュレーション開始~5度目の環境変動まではGRNが高速に適応
  - 環境変動から10秒経過時と180秒経過時の適合度の比較においてGRNが優位
  - 記憶した複数の表現型が最適化問題で共通の解構造を捉えたためと考えられる

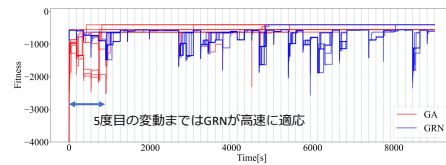


図: 提案手法(GRN)およびGAの適合度の推移。  
 ・帯域幅を180秒の時間間隔で49回ランダムな値に変更  
 ・性能要求にはGRNに学習させたものと異なる要求値を設定

表: 環境変動発生から制約条件を満たす解の発見時間

	GA	GRN
平均値	19.502 秒	72 秒
中央値	0.051 秒	0.891 秒

表: 環境変動発生から一定時間経過時の適合度の比較

経過時間	GAが優位	GRNが優位
10 秒	331 / 500	118 / 500
180 秒	331 / 500	63 / 500