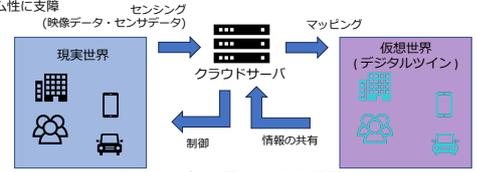


# 遺伝子制御ネットワークを用いた分散型映像分析システムの消費電力最適化方式

大阪大学 大学院情報科学研究科 村田研究室  
井上 誠志郎, 山内 雅明, 小南 大智, 下西 英之, 村田 正幸  
s-inoue@ist.osaka-u.ac.jp

## 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
  - 全ての人やモノがつながり様々な情報が共有される
  - 現実のモノやシステムの継続的な監視と最適化が可能
- デジタルツインの構築
  - 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
  - デジタルツイン上の情報に基づき現実世界を制御
- 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

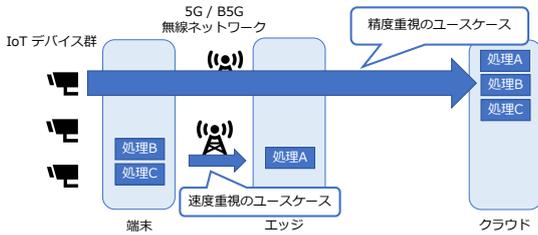


図：サイバー・フィジカルシステム

[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 ap- plications: 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
- 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

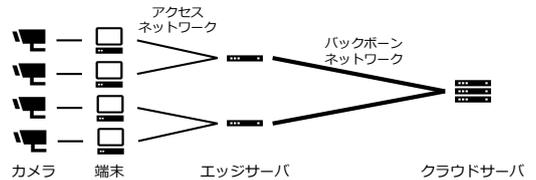


図：映像分析システムの動的な最適化

[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.

## 分散型映像分析システム最適化への取り組み

- システム消費電力の動的な最適化を組合せ最適化問題に定式化<sup>[8]</sup>
  - 制約条件：処理遅延 / 分析精度
  - 目的関数：システム全体の消費電力
  - 制御対象：処理の分散割合 / 利用するAIモデル
- 遺伝的アルゴリズムによる動的な最適解の探索
  - アプリケーションごとの性能要件の変化
  - 最適化対象のシステムが大規模



図：システム構成例

[8] H. Shimonishi, M. Murata, G. Hasegawa, and N. Techasantikul, "Energy optimization of distributed video processing system using genetic algorithm with bayesian attractor model," Proc. of IEEE 9th Int. Conf. Netw. Softwarization (NetSoft), pp.35-43, June 2023.

## 遺伝的アルゴリズム (Genetic Algorithm : GA)

- 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
  - 解の評価値により適応度を決定し、適応度の高い個体を集団に残す
- 遺伝的操作を加えて新たな個体を生成
  - 選択：集団から親となる個体を決定
  - 交叉：親個体の遺伝子に操作を加えて新たな個体を生成
  - 突然変異：新たに生まれた個体にランダムな変化を加える
- GA の課題
  - 解の多様性が維持されない
    - 初期収束：局所解に収束する
    - 環境変動の影響：変動後の解探索に時間がかかる
  - 研究対象のシステムは環境変動が頻発するため多様性を維持する必要がある



図：遺伝的アルゴリズムによる解探索

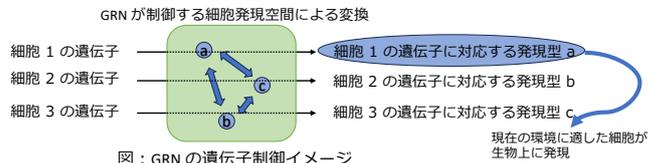
## 研究の目的とアプローチ

### 研究の目的

- 分散型映像分析システムの環境変動に対して高速に適応可能な最適化手法の提案と評価

### アプローチ

- 遺伝子制御ネットワーク (GRN: Gene Regulatory Network) の数理モデルを応用した手法の提案
  - 分子進化の中立説 (自然淘汰とは無関係な突然変異が多様性を維持) を応用
  - GRN がネットワーク構造に過去の環境を記憶する特性
  - 記憶した環境の想起により高速な適応を実現

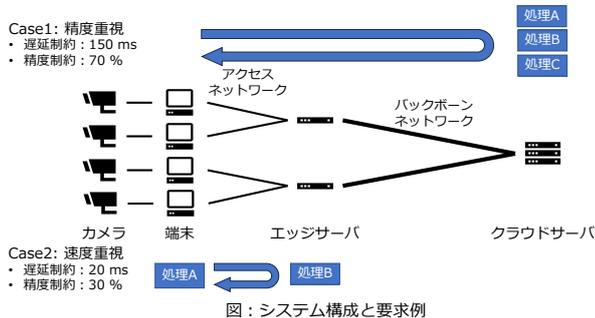


図：GRN の遺伝子制御イメージ

## 分散型映像分析システム<sup>[8]</sup>

7

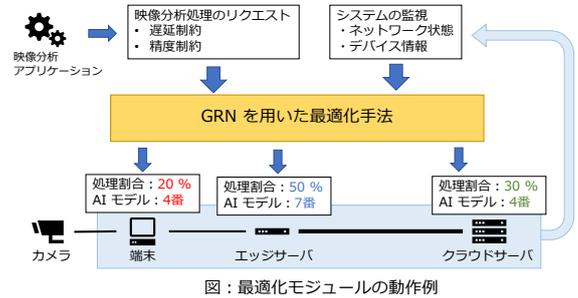
- 端末・エッジサーバ・クラウドサーバから構成されるシステム
- アプリケーションは二つの性能要件を設定して映像分析処理を要求
  - 処理遅延：映像分析にかかる E2E (end-to-end) 遅延
  - 分析精度：各デバイスにおける精度の処理割合による加重平均



## システム最適化モジュール

8

- 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
- 映像分析処理のリクエストに応じて動的に最適化



## 組合せ最適化問題の定式化

9

- 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

- 個体の適応度  $F$

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

処理遅延のパナルティ項    分析精度のパナルティ項

$S$ : セッションの集合  
 $T_s^d, T_n^s$ : セッション  $s$  におけるデバイス  $d$  とネットワーク  $n$  の処理遅延  
 $T_s^{max}$ : アプリケーションが設定した処理遅延の上限値  
 $A_s^d$ : 映像分析 AI  $M_s^d$  の分析精度  
 $A_s^{min}$ : アプリケーションが設定した分析精度の下限値  
 $D^s$ : 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  $d$  に割り当てる処理割合の重み  
 $M_s^d$ : デバイス  $d$  で稼働する映像分析 AI の種類

## 解の符号化

10

- 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する対象として扱う

最適化対象 [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



## 提案手法

11

- GA における個体の評価に GRN を応用
  - 標準的な GA では遺伝型を直接システム制御方法に変換して評価
  - GRN の発現過程を経て遺伝型を表現型へ変換
    - 記憶の影響で有利な表現型が現れやすくなる

標準的な GA



GRN を応用した GA



図：遺伝型から解への変換過程

## GRN の数理モデル<sup>[10]</sup>

12

- 個体は遺伝型と GRN の隣接行列を保持
  - 遺伝型  $G$ : 遺伝子発現の初期パターン  
 $G = [g_0, g_1, \dots, g_n] (-1 \leq g_k \leq 1)$
  - GRN の隣接行列  $B$ : 遺伝子間の正負の相互作用を表現  
 $B = \begin{bmatrix} b_{00} & \dots & b_{0n} \\ \vdots & \ddots & \vdots \\ b_{n0} & \dots & b_{nn} \end{bmatrix}$

- 遺伝型と隣接行列の乗算により表現型  $P(t) = [p_0(t), p_1(t), \dots, p_n(t)]$  を算出

$$\begin{cases} p_i(0) = g_i \\ p_i(t) = p_i(t) + \gamma_1 \sigma \left( \sum_{j=0}^n b_{ij} p_j(t) \right) - \gamma_2 p_i(t) \end{cases}$$

- パラメータ  $t^*$  による  $P(t^*)$  を用いて適応度を算出

表: 定数、変数の定義	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
$t^*$	発現過程の計算回数	10

[10] A. Wagner, "Does evolutionary plasticity evolve?," Evolution, vol.50, no.3, pp.1008-1023, June 1996.

## 提案手法の評価

13

- GRN による記憶と最適化問題における環境変動後の新しい解の発見に要する時間の関係性を検証
- 評価方式
  - 事前に GRN に表現型を記憶
    - シミュレーション環境における解を学習
  - 記憶を持つ GRN が探索に要する時間を評価
    - GRN を用いない標準的な GA (SGA: Simple GA) と比較
- シミュレーション設定
  - 小規模のシステムを想定して環境を設定
    - デバイス数: 端末・エッジ・クラウドそれぞれ1つずつ
    - セッション数: 1
  - 環境変動の設定
    - 一定の時間間隔で処理要求がA~Cに変化
      - 処理遅延: 20-150 ms の範囲から設定
      - 分析精度: 20-70 % の範囲から設定

要求	遅延制約: s	精度制約: %	SGA による解 記憶させた表現型
A	0.08914	59.245 %	[0,0,7,**,6-7] [0,0,7,0,0,7]
B	0.03414	24.271 %	[6,0,0,-2,*,*] [6,0,0,2,0,0]
C	0.07214	58.271 %	[0,7,0,*6-7,*] [0,7,0,0,7,0]

## GRN の事前学習

14

- シミュレーションで用いる三つの環境における解を記憶
  - 標準的な GA (SGA: Simple GA) で求めた解
    - 007007, 600200, 070070
- 表現型の発見に偏りを確認
  - ランダム生成した 1万個の遺伝型から発現する表現型の分布を評価
  - 29.9 % の確率で記憶させた表現型が発現
    - 表現型の組み合わせ総数は  $2^{18}$  であり高確率と言える

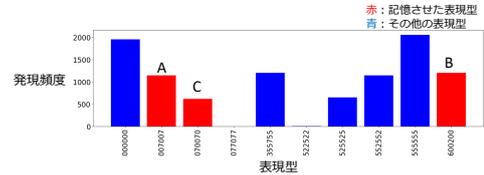


図: ランダム生成した遺伝型から発現する表現型の分布

[9] R.A. Watson, G.P. Wagner, M. Pavlicev, D.M. Weinreich, and R. Mills. "The evolution of phenotypic correlations and "developmental memory", " Evolution, vol.68, no.4, pp.1124-1138, Dec. 2014

## 解の発見に要する時間の比較

15

- 変動後の環境で新しい解の発見に要した時間を評価
  - 世代数による評価
  - 計算時間による評価
- GRN を用いた場合は継続的かつ安定的に発見
  - わずかな世代で解を発見
  - 解の発見までの計算時間にばらつきが少なく安定的
  - 計算時間の平均値に約 8 倍の差を確認
    - GRN: 32 ms
    - SGA: 254 ms

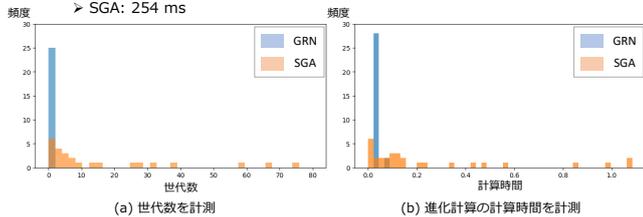


図: 環境変動直後から解の発見に要する時間の分布

## まとめと今後の展開

16

- まとめ
  - 生物の適応進化メカニズムを応用することで、システム的环境条件の変動に対して適応力の高い最適化方式を提案
  - 提案手法による適応力の向上を検証
    - 環境変動に対して高速に解が発見されることを確認
    - 継続的かつ安定的に解の発見が可能であることを確認
      - 計算時間の平均値に約 8 倍の差を確認
- 今後の展開
  - 問題が大規模な場合の適応可能性の検証
  - 環境変動に合わせてGRNを自律的に学習させる手法の検討
    - GRN の学習過程を最適化方式の進化に組み込む
  - 未知の環境への適応力の検証
    - 過去の記憶の組み合わせにより新たな表現型を生み出す GRN の特性を活用

## 二つの表現型の記憶

17

- シミュレーションで用いる二つの環境における解を記憶
  - 標準的な GA (SGA: Simple GA) で求めた解
    - A: 007007
    - B: 600200
- 表現型の発見に偏りを確認
  - ランダム生成した 1万個の遺伝型から発現する表現型の分布を評価
  - 約 50 % の確率で記憶させた表現型が発現

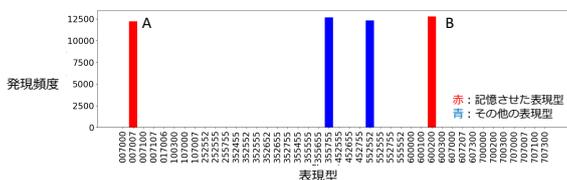
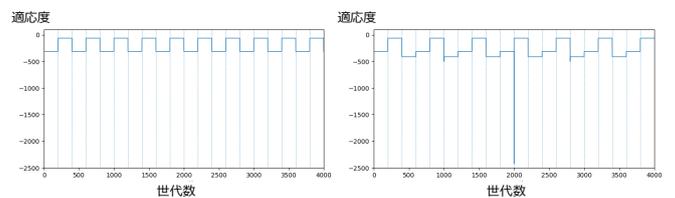


図: ランダム生成した遺伝型から発現する表現型の分布

## GRNを適用したGAにおける適応度の推移

18

- 200 世代ごとに交互に要求を切り替えた場合の適応度の推移を評価
- 変動後の環境における最適解を即座に発見
- 三つの環境で切り替える場合は時間を要するケースも確認
  - 記憶の数が増えたことにより最適解を発見する確率が低下

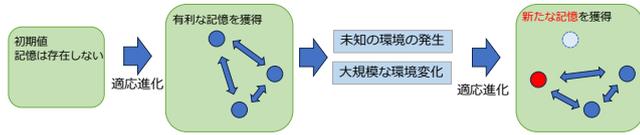


(a): 要求 A・B で切り替え

(b): 要求 A・B・C で切り替え

図: シミュレーションにおける適応度の推移

- GRN を用いた手法への期待
  - 自律的な記憶の獲得が可能であるため事前学習や再学習が不要
  - 既存の記憶の一般化によって未知の環境に適応可能
- GRN の欠点
  - 進化計算に時間がかかる
    - システム規模の拡大に対応できない可能性がある
  - 安定性に問題
    - 記憶が消滅する可能性がある
    - 環境への適応を保証できない



図：GRN の自律的な進化