

パラメータセットを用いるシミュレーションを行う 研究者支援システムの構築と適用

青木悠祐, 前田太陽, 村田忠彦



文部科学大臣認定 共同利用・共同研究拠点
関西大学ソシオネットワーク戦略研究機構
関西大学政策グリッドコンピューティング実験センター
(文部科学省私立大学社会連携研究推進拠点)

Policy Grid Computing Laboratory,
The Research Institute for Socionetwork Strategies,
Joint Usage / Research Center, MEXT, Japan
Kansai University
Suita, Osaka 564-8680, Japan
URL: <https://www.pglab.kansai-u.ac.jp/>
<http://www.kansai-u.ac.jp/riss/>
e-mail: pglab@ml.kandai.jp
tel. 06-6368-1228
fax. 06-6330-3304

関西大学政策グリッドコンピューティング実験センターからのお願い

本ディスカッションペーパーシリーズを転載、引用、参照されたい場合には、ご面倒ですが、弊センター（pglab@ml.kandai.jp）宛にご連絡いただきますようお願い申し上げます。

Attention from Policy Grid Computing Laboratory, Kansai University

Please reprint, cite or quote WITH consulting Kansai University Policy Grid Computing Laboratory (pglab@ml.kandai.jp).

パラメータセットを用いるシミュレーションを行う 研究者支援システムの構築と適用

青木悠祐¹, 前田太陽², 村田忠彦^{2,3}

Building of Support System for Development of Simulation Using Set of Parameters

Yusuke Aoki¹, Taiyo Maeda², Tadahiko Murata^{2,3}

概要

パラメータセットを用いるシミュレーションとは、確率変数を含む確率的シミュレーションや確率変数を含まない決定的シミュレーションにおいて、様々なパラメータを組合せて実行する必要があるシミュレーションである。パラメータセットを用いるシミュレーションを効率的に開発するためには、計算実行時間、結果の整理や比較作業時間の効率化が必要不可欠となる。そこで本論文では、これらの時間的な効率化を目的とし、パラメータセットを用いるシミュレーションを行う研究者支援システムを構築し、被験者実験を行った。その結果、システムを用いた場合と用いなかった場合の全体の作業時間は、2372.5秒から437.2秒となり、約1/5の時間的な効率化が行えた。

Abstract

Many researchers who research simulations using sets of parameter have to run their programs many times to compare developing algorithms and find better sets of parameter for their programs. We focus on stochastic simulations such as reinforcement learning algorithms because these simulations have much more sets of parameter than other simulations. In order to reduce their working time, we build a problem solving environments (PSE) system to assist them. Our system has three sub-systems: a distributed computing system, a data management system and a graph generation system. We conduct experiments with human subjects. As a result, user obtained five times speed-up for his work time through executions, storing results and comparing algorithms.

キーワード：問題解決環境，強化学習，分散計算システム，データ管理システム，
グラフ生成システム

Keywords: PSE system, reinforcement learning algorithms, distributed computing system,
data management system, graph generation system

1 関西大学大学院 総合情報学研究科 Graduate School of Informatics, Kansai University

2 関西大学 政策グリッドコンピューティング実験センター Policy Grid Computing Laboratory, Kansai University

3 関西大学 総合情報学部 Faculty of Informatics, Kansai University

1. はじめに

パラメータセットを用いるシミュレーションとは、確率変数を含む確率的シミュレーションや確率変数を含まない決定的シミュレーションにおいて、様々なパラメータを組合せて実行する必要があるシミュレーションである。ここでパラメータセットとは、シミュレーションを実行していく上で必要となる変数のことである。例えば、本論文では遺伝的アルゴリズム (Genetic Algorithms: GA) [1]と強化学習アルゴリズム[2, 3]を組合せたアルゴリズムをシミュレーションに用いる。そこで用いる GA のパラメータは、初期個体数、個体選択の方法、交叉の方法、交叉率、変異率、世代数などである。また、強化学習アルゴリズムのパラメータは、報酬値、学習率、割引率、学習回数などである。これらのアルゴリズムのパラメータ以外にシミュレーションの対象となる問題のパラメータとして、空間の大きさ、観測すべきデータの選択などがある。また、これらのパラメータセットは、シミュレーションプログラムの設計者が設定するため、数や組合せは多数におよぶ[4, 5]。本論文のシミュレーションの場合、アルゴリズムの設計者は、これらのパラメータセットを変更することで、アルゴリズムの頑健性の確認や他のアルゴリズムとの比較を行う必要がある。

本論文において構築するシステムは、これらのシミュレーションプログラムに用いられるアルゴリズムの開発を行う研究者やアルゴリズムを検証する研究者を対象とする。パラメータセットを用いるシミュレーションには、効率的に開発を進める上でいくつかの問題があり、複数回の計算の実行作業、複数回のシミュレーションの結果の比較作業の2つの問題に着目する。1つ目の問題は、以下の要因によって発生する。パラメータセットを用いるシミュレーションは、入力パラメータやアルゴリズムによってシミュレーション結果が異なる。アルゴリズムを変更した際、それに対しても入力パラメータを調整する。また、確率変数を含む確率的シミュレーションの場合、統計的にシミュレーション結果を検証するために異なる乱数系列を用いてシミュレーションを行う必要がある。これらによって、入力パラメータ、アルゴリズム、乱数系列、それぞれの組合せで複数回の計算を実行しなければならない。2つ目の問題は、計算によって得られたシミュレーション結果から時間ステップごとの細かい推移を確認する必要があるために発生する。単に数値データの比較だけではシミュレーション結果の検証を行うことは困難であるため、数値データをグラフ化することで時間ステップごとの細かい推移を比較し、入力パラメータやアルゴリズムの検証を行わなければならない。

パラメータセットを用いるシミュレーションを効率的に開発するためには、計算実行時間、結果の整理や比較作業時間の効率化が必要不可欠となる。これらの解決方法として次の3つの支援を行う必要がある。それは、複数回の計算実行を効率的に行うための分散計算、データの管理を行いやすくするための自動登録機能を持ったデータ管理、結果の細か

い推移の確認や比較を行うためのグラフの自動生成である。これら3つの機能を持ち、これらの専門知識を意識することなく本来のパラメータセットを用いるシミュレーションの開発に専念できるシステムがあればパラメータセットを用いるシミュレーションの開発の促進につながる。

すでに開発されているバイオ[6]、流体[7-9]、構造解析[10, 11]、創薬[12]などのための問題解決環境 (Problem Solving Environments: PSE) [13]は、扱うデータや検証方法が異なるため、これらのシミュレーションを開発する研究者に対して特有の支援システムが必要となる。そこで、パラメータセットを用いるシミュレーションを行う研究者支援システムを構築するために、他の分野の支援システムの特徴と機能を挙げる。バイオの分野では、分散計算によってシミュレーションを実行している。さらに、シミュレーション結果をデータベース (Database: DB) へ登録し、シミュレーション結果の比較までをシステム化することで、ゲノムの膨大なデータの解析や管理を行っている[6]。流体の分野では、PIV-WL を利用した実験、FLUENT[7]を利用したシミュレーションの実行、それらの条件や数値データなどを DB へ登録し、実験とシミュレーション結果の可視化までをシステム化することで、時間的な効率化を行っている[8, 9]。構造解析の分野では、ADVENTURE[10]を利用し、可視化までをシステム化している[11]。これらのシステムは、分散計算と DB により効果を得ている。

一方、パラメータセットを用いるシミュレーションの開発を行う研究者は、開発する際、アルゴリズムの改変によって、複数のプログラムを実行する。そのため、他の分野の支援システムで用いられている分散計算や DB などの機能に加え、シミュレーションプログラム、入力パラメータ、統計的にアルゴリズムを検証するために用いる乱数系列、最終のシミュレーション結果だけでなく時間ステップごとのシミュレーションの途中経過、これらを得た全ての入力パラメータなどを後から効率的に参照できることが重要となる。

そこで本論文では、パラメータセットを用いるシミュレーションの開発に要する計算実行、データ管理の作業、シミュレーションの結果を比較するための作業、それぞれの時間的な効率化を目的とする。各分野での PSE システムを参考に、パラメータセットを用いるシミュレーションを行う研究者支援システムを構築する[14-16]。

2.で本システムに適用するためにパラメータセットを用いるシミュレーションの開発を行い、その結果と考察を述べ、3.で本システムを構築するためにパラメータセットを用いるシミュレーションの開発の特徴とその効率化の方法について議論し、本システムの概要を述べる。4.で本論文の 2.で開発したパラメータセットを用いるシミュレーションを適用事例としてシステムの評価を行う。5.でシステムの汎用化について議論し、6.でまとめと今後の課題とする。

2. パラメータセットを用いるシミュレーション

本論文では、パラメータセットを用いるシミュレーションの中でも特に、社会科学において対象となる現象の再現や、現象を解明する1つの手段として注目されているマルチエージェントシミュレーション (Multi-Agent Simulation: MAS) に着目する。MAS は、実データを基に医療機関選択を行うシミュレーション[17]や政治学における政党支持のシミュレーション[18, 19]などに用いられている。本論文では、構築した支援システムに対して、様々なパラメータの調整や統計的な考察が必要となる GA と強化学習アルゴリズムを用いたマルチエージェントの協調行動シミュレーションを適用する。

そこで、本章ではマルチエージェント強化学習の概要と目的について述べ、シミュレーションに用いたアルゴリズムについて述べる。また、適用したシミュレーションの入力パラメータについて述べた後、協調行動シミュレーションの結果を示す。

2.1 マルチエージェント強化学習

近年、マルチエージェント強化学習について様々な方法が検討されている[4]。強化学習アルゴリズムは、エージェントの協調行動を獲得する手段として用いられる。強化学習アルゴリズムは、演繹的知識を必要とせず、環境から報酬を受け取ることでエージェントの行動を決定できる代表的な手法である。これを用い、制御対象に目標を設定しておくだけで、目標までの過程を意識することなく目標を達成することが可能となる。

強化学習アルゴリズムには、様々な手法がある。この代表的な手法である Profit Sharing[20-22]は、学習主体であるエージェントが、目標到達までに用いたルールをルール集合としてエピソード単位で学習していく手法である。また、Q 学習[23]は、エージェントが取りえる状態と行動を2次元のテーブルとして作成し、エージェントの状態と行動からテーブル内の値を学習していく手法である。それぞれの学習手法は、入力パラメータとして、エージェントが存在する環境の情報、エージェントが学習する回数、値の更新に用いる学習に対する重みなどを用いる。新たな強化学習手法の枠組みとして伊藤ら[24-27]は、遺伝的アルゴリズムにより動的に状態行動空間を生成し、生成された部分的な状態行動空間に対して学習を行う QDSEGA (Q-learning with Dynamic Structuring of Exploration Space Based on Genetic Algorithm) を提案している。これまでに QDSEGA は、マニピュレータ制御[24]、多足ロボット制御[25]、ホモジーニアスなマルチエージェント制御[26]、ヘテロジーニアスなマルチエージェント制御[27]に適用されている。

QDSEGA に関する従来手法では、単純な交叉が用いられていたが、近傍交叉を用いることにより、多足ロボットの制御問題に対し、より少ない探索回数で行動を獲得できることを示した[28, 29]。近傍交叉[30]は交叉対象を表現型空間の距離の近いものに限定すること

により，表現型空間での異なる形質を持つ個体同士の交叉により引き起こされる進化的停滞を防ぐことを目的としている．本論文では，各エージェントの能力が同等なホモジニアスなマルチエージェントの協調行動シミュレーションに対し，近傍交叉を用いた QDSEGA，近傍交叉を行うタイミングを閾値パラメータにより調整した QDSEGA，削除アルゴリズムを導入した QDSEGA を用いて高性能な Q-table をコンパクトな状態行動空間により実現できることを示す[31-34].

2.2 QDSEGA

QDSEGA は Q 学習を用いた行動獲得ダイナミクスと遺伝的アルゴリズムを用いた行動空間生成ダイナミクスから構成されている．行動獲得ダイナミクスにおいて，Q 学習で用いられる Q-table は実行可能な行動と遷移可能な状態から生成される．QDSEGA において，Q-table に用いられる行動は GA の個体群により生成され，行動状態空間が探索される．Q 学習では，生成された Q-table を使い，経験を通して学習が行われる．行動空間生成ダイナミクスにおいて，学習により更新された Q-table から各個体の適応度を計算し，遺伝的操作によって新たな個体群を生成する．QDSEGA の流れを図 1 に示す．

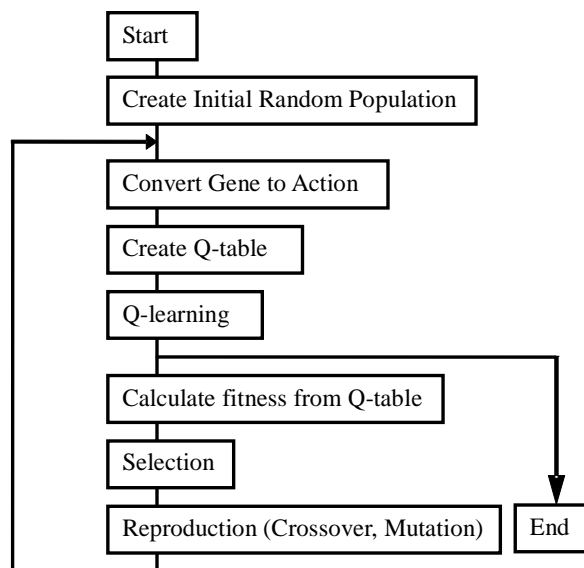


図 1 QDSEGA の流れ

2.2.1 行動獲得ダイナミクス

行動獲得ダイナミクスに用いられる Q 学習は Q-table を用いて行われる．Q-table に用いられる行動は GA の個体群から構成される．しかし，同一の個体が存在した場合，Q-table が冗長となる．そこで，同一の個体は一つの個体とみなし，行動を生成する．従って，世

代を重ねるにつれ、タスク達成に必要な大きさまで Q-table が縮小されていく。Q-table を生成する上で、同一の行動を持つ個体の削除の例を図 2 に示す。Q-table に用いられる状態は、行動が行われた目標状態とするため、行動と同一のものとして設定される。Q-table の更新は以下の式で行う。

$$Q(s, a) \leftarrow Q(s, a) + \alpha \{r(s, a) + \gamma \max_{a'} Q(s', a')\} \quad (1)$$

ここで、 $Q(s, a)$ は状態 s と行動 a の Q 値を示し、 r は報酬、 α は学習率、 γ は割引率を示している。

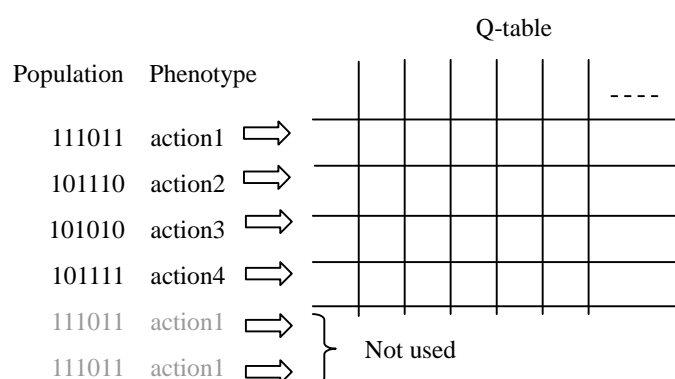


図 2 同一の行動を持つ個体の削除

2.2.2 行動空間生成ダイナミクス

行動獲得ダイナミクスによって得た Q 値を基に、Q 値の規格化、Q 値による適応度、行動の使用頻度による適応度を計算し、最終的な個体の適応度を計算する。Q 値の規格化は以下の 2 つの理由より行われる。規格化を行わず Q 値を適応度とした場合、報酬に近い状態で実行された行動が高く評価され、報酬にいたるまでの行動が淘汰されてしまい、タスク達成が困難となってしまう危険性がある。また、Q 学習において、Q 値は負の値を取る可能性があるが、GA における適応度は正の値である必要がある。規格化は以下の式により行われる。

IF $Q(s, a) \geq 0$

$$\text{THEN } Q'(s, a) = \frac{1-p}{V_{\max}(s)} Q(s, a) + p \quad (2)$$

IF $Q(s, a) < 0$

$$\text{THEN } Q'(s, a) = -\frac{1-p}{V_{\min}(s)} Q(s, a) + p \quad (3)$$

ただし,

$$V_{\max}(s) = \max_{a'}(Q(s, a')) \quad (4)$$

$$V_{\min}(s) = \min_{a'}(Q(s, a')) \quad (5)$$

また, $p(0 \leq p \leq 1)$ は報酬とペナルティの比率を表す定数である. 規格化を行った $Q'(s, a_i)$ に対し, 行動を固定し, 状態に対して降順でソートを行い $Q'_s(s, a_i)$ とする. 降順でソートを行うことにより, $Q'_s(1, a_i)$ は $Q'(s, a_i)$ における最大値を示し, $Q'_s(N_s, a_i)$ は $Q'(s, a_i)$ の最小値を示す. ここで N_s は Q-table における状態の総数である. ソートを行った Q 値から, 各行動の Q 値による適応度を以下の式によって計算する.

$$fit_Q(a_i) = \sum_{j=1}^{N_s} (w_j \frac{\sum_{k=1}^j Q'_s(k, a_i)}{j}) \quad (6)$$

ここで, w_j は特殊な行動と, 一般的な行動の重みを表す定数である. 次に, 使用頻度による適応度を計算する. Q 値による適応度では, 報酬に近い状態で実行された行動の価値が高く評価され, 重要な行動であっても評価されない. そこで, 重要な行動は複数の状態から高い頻度で使用されることに着目し, 以下の式によって使用頻度による適応度を計算する.

$$fit_u(a_i) = \frac{N_u(a_i)}{\sum_{j=1}^{N_a} N_u(a_j)} \quad (7)$$

ここで, N_a は Q 学習における全行動数, $N_u(a_i)$ は Q 学習における a_i の使用回数を示している. 最後に, 各個体に対する適応度を以下の式によって計算する.

$$fit(a_i) = fit_Q(a_i) + k_f \cdot fit_u(a_i) \quad (8)$$

ここで, k_f は Q 値による適応度 $fit_Q(a_i)$ と使用頻度による適応度 $fit_u(a_i)$ を調整する定数である. 適応度 $fit(a_i)$ を用い, 遺伝的操作により次世代の行動を決定する. 次世代の行動を生成する際, 親個体と子個体の交換は行わない. 従って, 遺伝的操作によって個体数は増加する. しかし, 選択によって個体数は一定数まで減少するため, 増加し続けることはない.

2.2.3 近傍交叉による交叉と削除アルゴリズム

近傍限定による交叉[30]とは, 表現型空間における異なる形質を持つ個体同士の交叉により引き起こされる進化的停滞を防ぐために, 交叉対象を表現型空間の距離の近い順に k 個選び出すというものである. 本論文では, 個体に対し, 同じ遺伝子座において, 同一の

値を取っている数によって、近傍を定義する。図3の例では近傍4となる。過去の研究において、近傍交叉を用いた QDSEGA を多足ロボットの制御へ適用し、学習の高速化が行われた[28, 29]。本論文では、近傍交叉を用い QDSEGA をマルチエージェントの協調行動シミュレーションへ適用する。さらに近傍交叉を適用するタイミングに関するパラメータを閾値として設定することで、近傍交叉を用いるタイミングを調整し、Q-table に用いられる行動数を削減する。

また、削除アルゴリズムとは、Q-table をよりコンパクトにするために、目標達成に不必要と考えられる行動を次世代に引き継がないというものである。本論文では、行動獲得ダイナミクスに用いられる Q 学習において、目標達成に用いられなかった行動に対して適応度 0 を設定し、遺伝的操作によって選択されないように設定する。目標達成に用いられた行動に対しては従来どおり Q 値と使用頻度から適応度を計算する。学習初期において削除アルゴリズムを適用した場合、すべての行動が削除されてしまう危険性があるため、削除アルゴリズムを適用するタイミングに関するパラメータを閾値として導入することで、Q-table に用いられる行動数の削除を行う。本論文で用いた問題では、学習効率を示す GAIN がある閾値を超えたときに削除アルゴリズムを適用する。GAIN については、後述する。

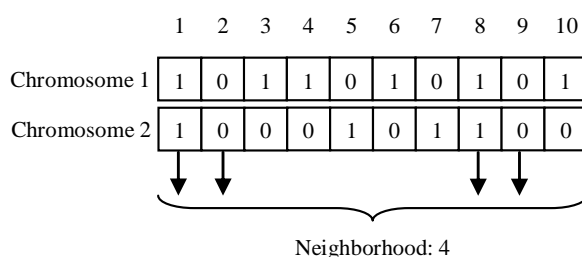


図3 近傍の定義

2.3 シミュレーション

本論文では、図4に示す荷物運搬タスクを用いてシミュレーションを行う。環境は5×5の25のセルからなり、各セルには番号が割り当てられている。セル0、セル4にそれぞれ5体のエージェントが配置され、荷物1 (L1) がセル7に、荷物2 (L2) がセル22に配置されている。この状態を初期状態とする。タスクの目的はL1をゴールへ運ぶことである。エージェントが、荷物を移動するためには2体以上のエージェントが必要である。ここで、エージェントは1ステップに上下左右に1セル移動可能であるが、L1は上下方向のみ、L2は左右方向のみ移動可能である。全エージェントが一つのセルに存在することはできるが、荷物が存在するセルには、他の荷物およびエージェントが進入することはできない。また、ゴールには扉が存在し、スイッチを押さなければ、扉は開かない。従って、L1をゴールへ

運ぶためには、L2 を L1 の進路から排除し、スイッチを押して扉を開け、L1 をゴールへ運ぶ必要がある。

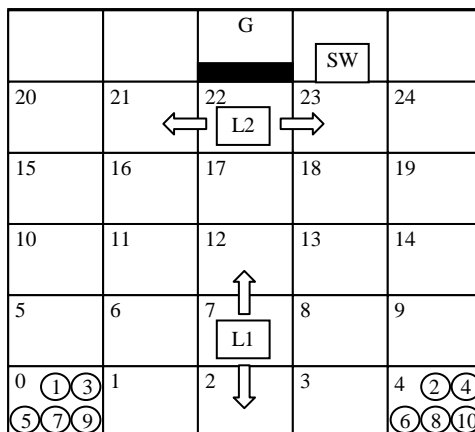


図4 荷物運搬タスク

2.3.1 コーディング

各個体はエージェント数と同一の遺伝子座からなる。各遺伝子座が取る値は、各エージェントが存在するセルの番号である。従って、初期状態である図4における遺伝子は図5となる。エージェントの移動は以下のルールによって行われる。

<R1 : エージェントの移動ルール>

IF $x(i) \neq x_i$

THEN $x(i+1) = x(i) - \text{sgn}(x(i) - x_i) \Delta x$ and $y(i+1) = y(i)$

ELSE IF $y(i) \neq y_i$

THEN $y(i+1) = y(i) - \text{sgn}(y(i) - y_i) \Delta y$ and $x(i+1) = x(i)$

ELSE $x(i+1) = x(i)$ and $y(i+1) = y(i)$

<R2 : 衝突回避ルール>

IF 障害物がエージェントの進行方向に存在する

THEN IF 障害物が荷物である

THEN R3 を用いる

ELSE 止まる

ELSE R1 を用いて移動

<R3 : 荷物の移動ルール>

IF 荷物が進行方向に存在する

THEN 移動する方向に荷物を押す

ELSE R1 を用いて移動

<R4 : ドアの操作ルール>

IF 移動が終了した場所にスイッチが存在する

THEN ドアを開ける

ELSE 何もしない

	1	2	3	4	5	6	7	8	9	10
Chromosome	0	4	0	4	0	4	0	4	0	4

図5 荷物運搬タスクの初期状態の遺伝子

2.3.2 報酬とペナルティ

Q 学習時に与えられる報酬とペナルティは、以下の条件によって与えられる。L1 を上方へ移動した場合に 20, L2 を L1 の進路から排除した場合に 20, L1 をゴールへ運びタスクを達成した場合に 200 の報酬を与え、初期状態へ戻す。また、L1 を下方へ移動した場合に -20, L2 を L1 の進路へ移動した場合に -20, Q 学習によって与えられた行動によって目標状態へ移動できなかった場合に -10 のペナルティを与え、初期状態へ戻す。

2.3.3 入力パラメータ

シミュレーションに用いる入力パラメータを表 1 と表 2 に示す。前述した近傍交叉を用いる場合の近傍については、 k_{sim} を用いる。よって、 $k_{sim}=0$ は、従来の交叉と同様であり、 $k_{sim}=10$ は、同一の個体のみを交叉対象とすることを示す。本論文では、 $k_{sim}=0, 2, 4, 6, 8$ を用いてシミュレーションを行った。

表 1 GA に用いる入力パラメータ

The number of individuals	300
Selection	Roulette selection
Type of crossover	Uniform crossover
The probability of crossover	0.2
Type of mutation	Change the value among valid cell number
The probability of mutation	0.001
The number of generations	100
Weights	$K_f=200, w_1=0.5, w_N=0.5, w_i=0(i=2,\dots, N_{s-1})$

表 2 Q 学習に用いる入力パラメータ

Reward	When “L1” reaches the goal, $r = 100$
	When “L1” moves up or “L2” is removed, $r = 20$
	When “L1” moves down or “L2” blocks the course of “L1”, $r = -20$
	When any agent can not move to the target position, $r = -10$
Learning rate	$\alpha = 0.9$
Discount rate	$\gamma = 0.1$
ε -greedy action selection	10% random action
The number of trials of each learning dynamics	10,000

2.4 シミュレーションの結果

荷物運搬タスクに対し、以下の4つのアルゴリズムを用いてシミュレーションを行い、その結果を示す。使用するアルゴリズムは、オリジナルの QDSEGA[26]、近傍交叉を用いた QDSEGA、閾値と共に近傍交叉を用いた QDSEGA[31, 32]、閾値と共に近傍交叉と削除アルゴリズムを用いた QDSEGA[33, 34]である。比較には、2つの指標を用いる。

指標として用いる GAIN は、各世代における Q 学習の 10,000 ステップの学習のうち最終 100 ステップ間の学習の報酬とペナルティの総和の平均である。前述したルールよりエージェントの動きは横方向へ移動した後、縦方向へ移動するため、タスクは最小で2回の状態遷移により達成される。1回目の状態遷移において、L1 を上へ移動し報酬 20、2回目の状態遷移において、L2 を横へ移動、L1 を上へ移動、L1 をゴールへ運び報酬 240 を獲得する、従って合計の最大獲得報酬は 260 となり、この2つの学習ステップにおける平均 GAIN は 130 となる。なお、近傍交叉および削除アルゴリズムの閾値を、GAIN が 100 を超えるときとした。この時、ゴールへ到達できる行動が獲得されているため、本シミュレーションでは、この値を用いた時の結果を示す。次に、指標として用いる行動数は各世代における Q-table の行動数を示している。これら2つの指標を用いて乱数系列を 10 回変更し、平均した値を用いる。

図 6 と図 7 に、従来手法と近傍交叉を用いた QDSEGA の GAIN と行動数の推移を示す。近傍交叉のみを用いた場合、図 6 の結果より、近傍 2 に関しては伊藤らによる従来手法[26]と比較し、高い GAIN を得るための状態行動空間が獲得されていることがわかるが、近傍 2 以外の近傍交叉においては、従来手法と比較し、報酬を得るための行動が獲得されてい

ない。しかし、図7の結果より、Q-tableを構成する状態行動空間が縮小されていることがわかる。シミュレーションにより、近傍交叉を直接用いた場合、状態行動空間は縮小できたものの、近傍2以外の近傍交叉においては性能を悪化させる結果になったため、近傍交叉を行うタイミングを閾値パラメータにより調整する。

図8と図9に、従来手法と閾値と共に近傍交叉を用いたQDSEGAのGAINと行動数の推移を示す。近傍交叉を適用するタイミングを閾値によって調整した場合、図8の結果より、全ての近傍交叉において、従来手法と同等のGAINが獲得されていることがわかる。また、図9の結果より、近傍交叉のみを用いた場合と同じくQ-tableを構成する状態行動空間が縮小されていることがわかる。このことより、高性能なQ-tableをコンパクトな状態行動空間により実現できた。さらに、よりコンパクトなQ-tableを実現するため、削除アルゴリズムを導入する。図10と図11に、従来手法、削除アルゴリズム、閾値と共に削除アルゴリズムと近傍交叉を用いたQDSEGAのGAINと行動数の推移を示す。図8と図10の結果より、閾値付き近傍交叉や従来手法と比較し高いGAINを得ていることがわかる。また、図9と図11の結果より、閾値付き近傍交叉で最も行動数を削減することのできた近傍8と比較して、削除アルゴリズムを導入した場合、全てにおいて行動数を削減することができている。表3に図9の最終世代における行動数、表4に図11の最終世代における行動数を示す。また、これらの結果から得られたエージェントの協調行動の例を図12と図13に示す。

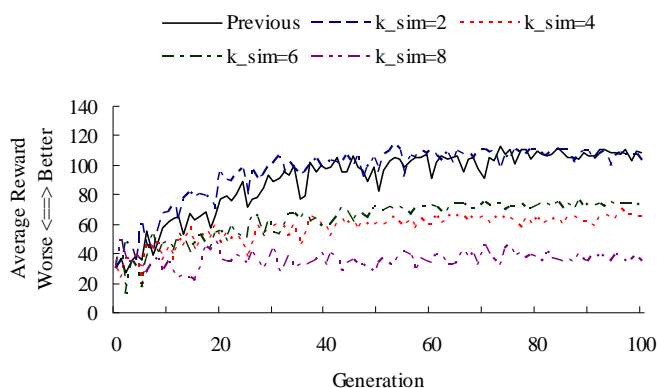


図6 近傍交叉を用いた場合のGAIN

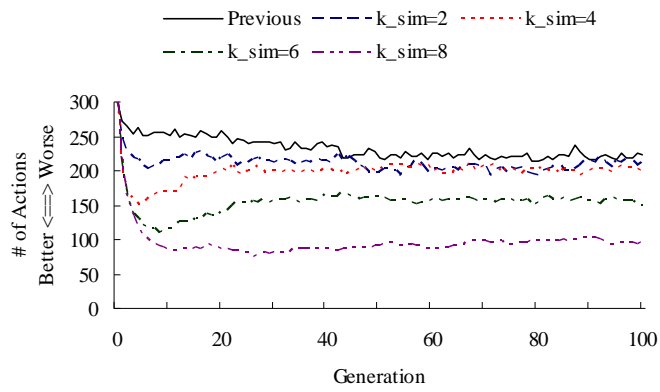


図7 近傍交叉を用いた場合の行動数

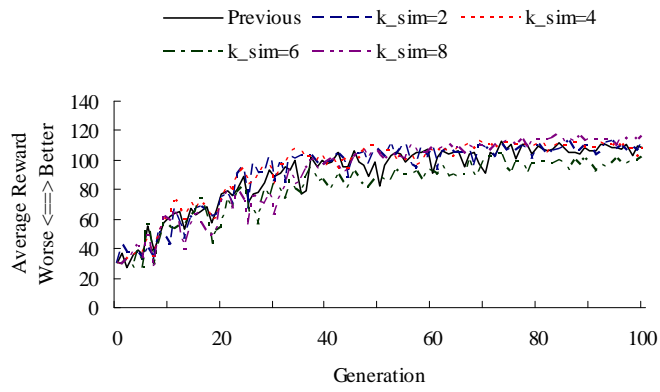


図8 閾値と共に近傍交叉を用いた場合の GAIN

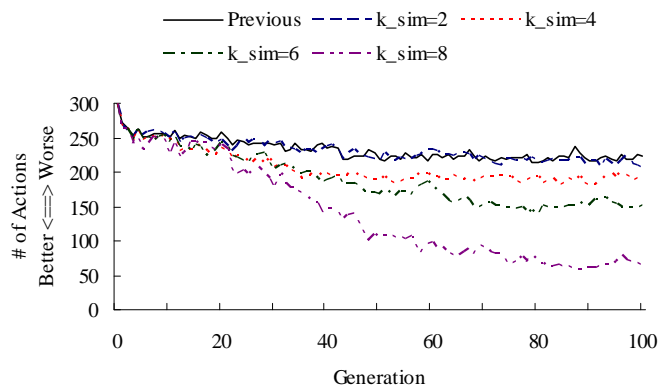


図9 閾値と共に近傍交叉を用いた場合の行動数

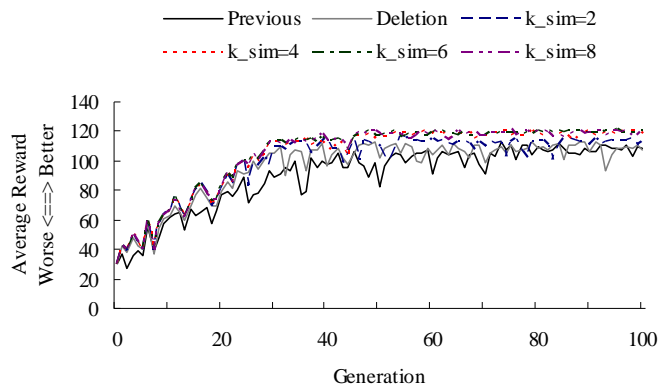


図 10 閾値と共に近傍交叉と削除アルゴリズムを用いた場合の GAIN

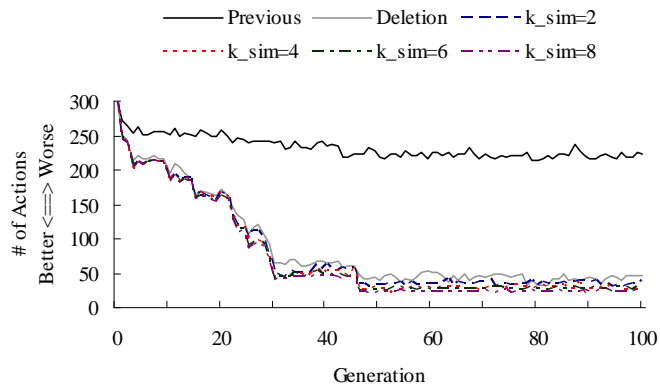


図 11 閾値と共に近傍交叉と削除アルゴリズムを用いた場合の行動数

表 3 閾値と共に近傍交叉を用いた場合の最終世代における行動数

Without Deletion Algorithm	Previous	2	4	6	8
# of actions	222.2	210.8	196.9	149.2	96.5

表 4 閾値と共に近傍交叉と削除アルゴリズムを用いた場合の最終世代における行動数

With Deletion Algorithm	Previous	2	4	6	8
# of actions	46.6	39.0	28.0	30.2	25.1

	1	2	3	4	5	6	7	8	9	10
Initial State	0	4	0	4	0	4	0	4	0	4
Transition	23	1	15	7	12	3	24	12	16	19
Final State	21	15	23	22	16	22	16	7	16	21

図 12 最終的に得られたエージェントの行動を示す遺伝子

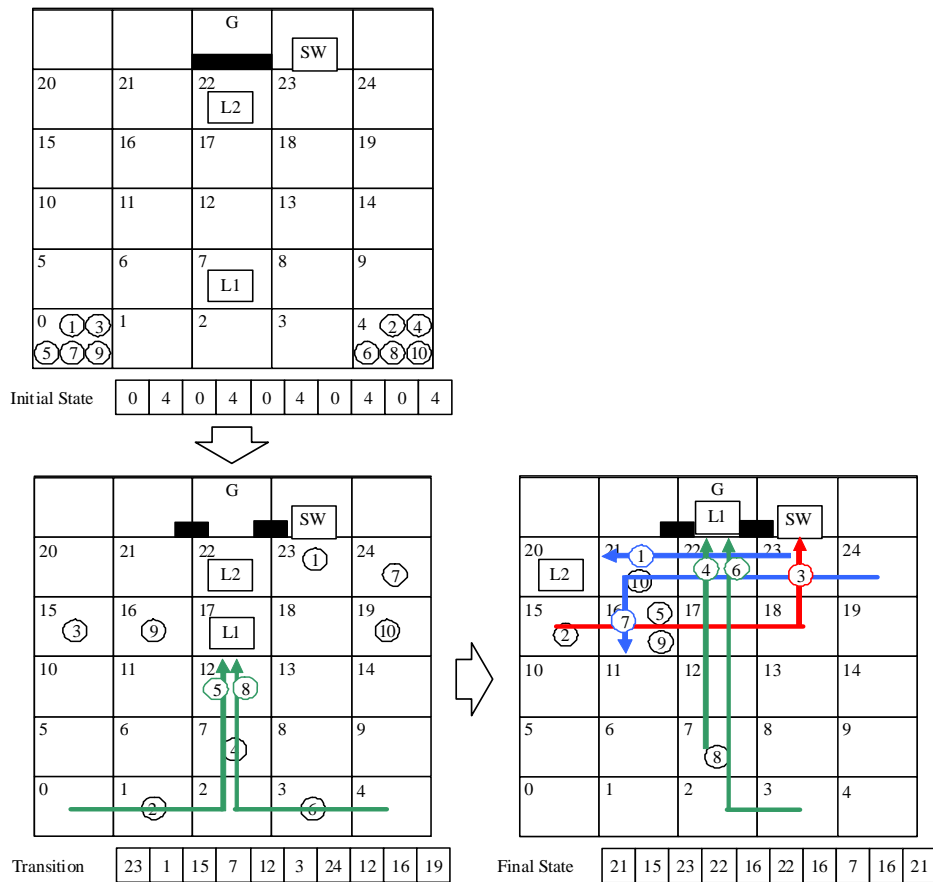


図 13 最終的に得られたエージェントの行動

2.5 シミュレーションの考察

本章において、ホモジーニアスなマルチエージェントシミュレーションに対して近傍交叉を用いた QDSEGA の有効性と削除アルゴリズムを用いた QDSEGA の有効性を示した。近傍交叉のみを用いた場合、2 近傍以外の近傍交叉が有効に働いていないのは、似た個体を交叉対象とするため、探索が進まないことが原因であると考えられる。また、閾値により交叉のタイミングを調整し、ある程度有効な行動を確保できた後に局所を探索するため、近傍数に関係なく、有効な行動系列を獲得でき、Q-table の縮小も行うことができたと考え

られる。

削除アルゴリズムを用いた場合、他のアルゴリズムと比較して行動数をより削減することに成功した。また、高い GAIN も得ることに成功した。これは、目標達成に不必要な行動を削除するという操作によって、目標達成に必要な行動のみを残すことができ、それにより GAIN の増加にもつながったと考えられる。近傍交叉を用いた場合と比較し、近傍交叉の影響が顕著に見えることはなかったが、削除アルゴリズムを導入した場合でも、必要な行動の中から、さらに局所を探索し、有効な行動を作るため、従来手法を削除アルゴリズムと共に用いた場合でも、性能が向上したと考えられる。

3. パラメータセットを用いるシミュレーションを行う研究者支援システム

本システムで対象とするパラメータセットを用いるシミュレーションを行う研究者は、シミュレーションの対象となる問題を分析しアルゴリズムを開発し、シミュレーションプログラムを作成する。研究者は、シミュレーションプログラムに対してパラメータを設定するため、その値は離散値を用いる。これらの研究者は、他のアルゴリズムとの比較や有効性の検証のために様々なパラメータを変更してシミュレーションを行う。しかし、全てのパラメータの組合せを用いてシミュレーションを実行することは、組合せ数に伴うシミュレーションの実行時間、パラメータやシミュレーション結果の管理作業の観点から現実的ではない。そこで、比較に用いないパラメータや有効性を検証するために用いないパラメータを固定することで、パラメータの設定を少なくするようにアルゴリズムやシミュレーションプログラムを設計する。しかし、パラメータの設定を少なくするようにアルゴリズムやシミュレーションプログラムを設計したとしても、アルゴリズムの変更やパラメータの組合せに伴い以下の要因により支援が必要となる。

2.のような確率的シミュレーションを行う場合、研究者はアルゴリズム開発だけでなく、シミュレーションプログラムの作成、シミュレーションプログラムの実行、用いたパラメータの管理、シミュレーション結果のグラフ化など、アルゴリズム開発を行うだけでなく様々な作業が伴う。研究者は、提案したアルゴリズムから 2.5 のような考察を行うために、表 1 や表 2 のようなパラメータを組合せて関連研究の論文や資料と同様の傾向を持つ結果を出し、さらに図 6-図 11 のように、提案したアルゴリズムの比較を行わなければならない。これらの作業の流れを図 14 に示す。パラメータを用いるシミュレーションを行う際、研究者は、シミュレーションプログラム、コマンドラインやファイルによる入力パラメータ、統計的な結果を得るための乱数系列を組合せて複数回の計算を行う。その後、研究者はシミュレーション結果を収集し、その結果と入力パラメータや乱数系列の対応関係をまとめる。シミュレーション結果を数値データのみから時間ステップごとの細かい推移の確認や比較を行うことは、困難であるために、数値データを図 6-図 11 のようにグラフ化す

る。

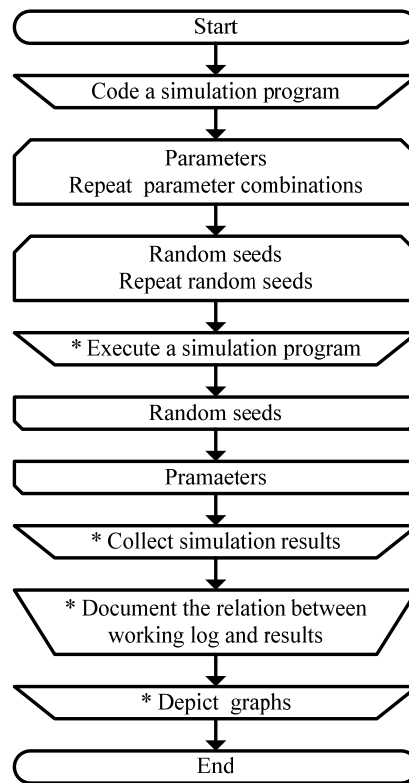
研究者は、図 14 の作業を 1 つのシミュレーションプログラムに対して行った後、異なるシミュレーションプログラムに対しても同様の作業を行う。その後、シミュレーション結果のグラフから細かい推移の確認や比較を行う。研究者は、より良いアルゴリズムの開発のためにシミュレーションプログラムを変更し、これらの作業を繰り返す。これらのことから本論文では、計算実行、入力パラメータやシミュレーション結果の管理、グラフ化の作業に着目し自動化を行う。自動化を行う部分を図 14 中*によって示す。

適用した 2. のシミュレーションでは、図 1 の流れが 1 回の実行となる。1 回の実行における表 1 や表 2 のパラメータや乱数系列の指定は、シミュレーションの初期条件として設定できるため、計算は独立して行える。そのため、分散計算を用いて計算実行時間の短縮が可能となる。

データ管理において研究者は、表 1 と表 2 のパラメータなどからディレクトリ名やファイル名を付けることを行う。また、2.2.3 のように新たなアルゴリズムを導入した場合、それに対しても同様の作業を行う。これらの作業を人手によって行う場合、データ数が増えるに従い、データの紛失、ファイル名の付け間違いなどにより、重複したシミュレーションや作業が行われる可能性がある。そこで、システムによって DB 内に、シミュレーションプログラム、入力パラメータ、乱数系列、シミュレーション結果などのデータと、それらの対応関係を自動的に登録することでデータ管理の作業時間の短縮が可能となる。

シミュレーション結果を比較するための作業は、2.4 の結果の提示や 2.5 の考察において必要となる。シミュレーション結果の数値データのみから時間ステップごとの細かい推移の確認や比較を行うことが困難であるため、研究者はスプレッドシートなどを用いて図 6- 図 11 のようなグラフを作成する。その際、グラフ化されたシミュレーション結果の比較だけでなく、シミュレーションプログラムや入力パラメータなども同時に比較しなければならない。そこで、ユーザが行わなければならないグラフ化の作業をシステムで自動化し、DB と連携することでシミュレーション結果の比較作業時間の短縮が可能となる。

パラメータセットを用いるシミュレーションを行う研究者にとって、アルゴリズムの比較や有効性の検証のために、パラメータの組合せを検討することは必要不可欠である。そのため、これらの作業の支援が必要となる。



[*] is a work which a system is able to automate.

図 14 確率的シミュレーションを行う研究者の作業

3.1 システムの概要

本システムは、分散計算システム、データ管理システム、グラフ生成システムの3つのサブシステムから構成される。ユーザがシステムを専門的な知識を用いることなく利用できることやシステムの導入の煩雑さを考慮して、PHP と Adobe Flash を用いた Web ベースのシステムとした[35, 36]。なお、研究者支援における支援には、プログラムの作成支援、データの解析支援、論文執筆支援などが考えられるが、本論文では、プログラムの実行支援、データの管理支援、グラフ化支援の3つを行うことで研究者の作業支援を行う。そのため、想定するユーザは、自分自身でプログラムを書くことができるパラメータセットを用いるシミュレーションプログラム開発者とパラメータセットを用いるシミュレーションプログラム利用者とした。実行可能なシミュレーションプログラムは、コマンドライン上で実行可能な exe 形式のファイルと Java で作成された jar 形式のファイルとした。シミュレーション結果は、標準出力にコンマ区切りの csv 形式の数値データとして出力されるものとする。

3.2 分散計算システム

分散計算システムでは、InstantPSE[37]の一部を利用し、分散計算の実行支援機能を実現した。分散計算に利用する計算機は、Windows がインストールされた計算機とした。分散計算システムは、1つのバッチファイルの実行を1つの計算ジョブとして計算機に割り当てることで、ユーザのジョブ実行を代行する。図15に分散計算システムの構成を示す。

システム構築者は、システム導入時に以下の作業を1度だけ行う必要がある。システム構築者は、サーバ側で計算機として用いるPCのIPアドレスを分散計算システム内のJavaモジュールを用いて登録する。その際、計算機として用いるPCに対してもJavaモジュールを導入しておく。その後、計算機として用いるPCに対してシミュレーションに必要なプログラムやファイルをアクセス可能な状態にするために、シミュレーションプログラム格納用ディレクトリ、入力パラメータに用いるファイル格納用ディレクトリ、バッチファイル格納用ディレクトリ、シミュレーション結果格納用ディレクトリを共有フォルダとして設定する。なお、ユーザはこれらの作業を意識する必要はない。

ユーザは、シミュレーションプログラムを実行ファイルとしてシステム内に登録する。登録する際、ユーザ側の計算機で実行されるFlashからサーバ側へファイルや付随する情報を送信するために、FlashからPOSTによってPHPへ通信を行いそれらの情報をサーバ側へ送信する。この登録は後述するデータ管理システムにより実行される。次に、ユーザは登録されたシミュレーションプログラムに対し、入力パラメータとして値の範囲やファイル名、乱数系列を設定する。設定後、シミュレーションプログラムの登録と同様にFlashからPOSTによってPHPへ通信を行い、データ管理システムが登録を行う。分散計算システム内のPHPモジュールは、登録されたシミュレーションプログラムと入力パラメータや乱数系列の組合せから自動的にバッチファイルを生成する。

バッチファイルは、組合せの数だけ生成される。バッチファイル内に記述される内容は、1回のシミュレーション実行に必要な情報と実行が完了した際のパイプ処理のみである。バッチファイル格納用ディレクトリは、分散計算システム内のJavaモジュールによって常に監視が行われている。新規にバッチファイルが生成されると、分散計算システム内のJavaモジュールはそれを検知し、計算機として用いるPCに対してジョブを割り当て、各計算機で実行されているJavaモジュールによってジョブを実行する。

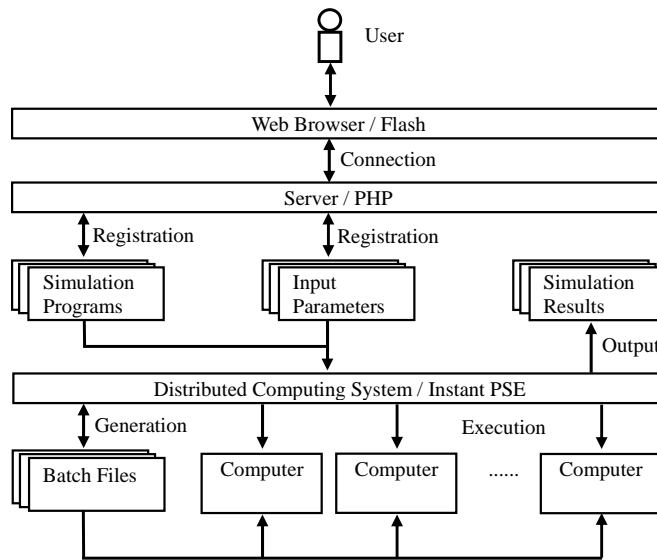


図 15 分散計算システム

3.3 データ管理システム

データ管理システムでは、管理対象となるシミュレーションプログラム、入力パラメータ、乱数系列、シミュレーション結果とそれらの対応関係を登録するために PHP, Java, MySQL を用いてデータ管理のための自動登録を実現した。入力パラメータやシミュレーション結果がファイルであった場合、共有フォルダを用いているため、それらのディレクトリ名とファイル名を DB に登録することとした。図 16 にデータ管理システムの構成を示す。

データ管理システムは、ユーザが分散計算システムの Flash を用いてシミュレーションプログラムを登録する際、データ管理システム内の PHP モジュールから MySQL のクエリを実行し DB 内のシミュレーションプログラム用のテーブルへ登録する。登録されたシミュレーションプログラムには、主キーとして ID を設定する。また、ユーザが入力パラメータや乱数系列を登録した際、データ管理システム内の PHP モジュールから MySQL のクエリを実行し DB 内に全組合せを入力パラメータ用のテーブルへ登録する。登録する際、1つの組合せを1つのレコードとして登録し、シミュレーションプログラムと同様に、主キーとして ID を設定する。分散計算システムは、これらの主キーを用いてバッチファイルを生成する。

データ管理システム内の Java モジュールは、常にシミュレーション結果格納用ディレクトリを監視している。そのため、データ管理システムに登録されたシミュレーションプログラム、入力パラメータ、乱数系列を用いて分散計算システムが実行を終了した後、シミュレーション結果が出力されることで、データ管理システムはシミュレーション実行の終

了を検知する。シミュレーションの実行の終了を検知するとデータ管理システム内の Java モジュールは、MySQL のクエリを実行し、シミュレーション結果をシミュレーション結果用のテーブルへ登録する。その後、それぞれの主キーを用いて、シミュレーションプログラムと入力パラメータのリレーションテーブルへ対応関係を登録する。また、シミュレーションプログラムとシミュレーション結果、入力パラメータとシミュレーション結果も同様にそれぞれのリレーションテーブルへ対応関係を登録する。これらによって、シミュレーションプログラムから実行したことのあるパラメータセットの検索や、シミュレーション結果からパラメータセットの検索などが可能となる。

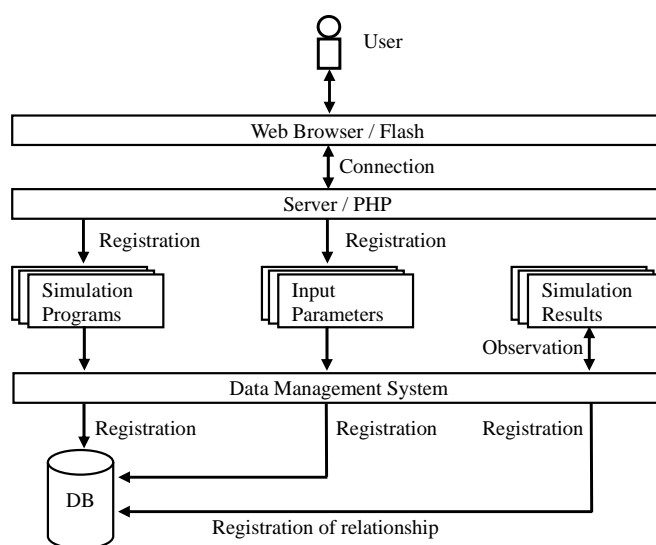


図 16 データ管理システム

3.4 グラフ生成システム

グラフ生成システムでは、シミュレーション結果の時間ステップごとの推移を提示する。グラフのサムネイル画像の生成と、ユーザの操作からグラフの表示とその重ね合わせを自動的に行う。グラフ化の対象となる結果は、先に述べた csv 形式のファイルとした。図 17 にグラフ生成システムの構成を示す。

グラフのサムネイル画像の自動生成には、グラフ生成ツールである gnuplot を使い、データ管理システムと同様に、グラフ生成システム内の Java モジュールを用いてシミュレーション結果が出力されるディレクトリを監視することで実現した。システムは、シミュレーション結果が出力されると gnuplot と連携し、ユーザにサムネイル画像を提示する。その際、グラフを生成するために必要となる gnuplot 用のグラフ記述ファイルの生成とその実行をグラフ生成システム内の Java モジュールで処理し、サムネイル画像のディレクトリ名とファイル名を Java モジュールから MySQL のクエリを実行することで DB へ登録する。

また、サムネイル画像についても主キーを設定し、シミュレーション結果の対応関係をリレーションテーブルへ登録する。シミュレーション結果は、すでにデータ管理システムによってシミュレーションプログラムや入力パラメータと対応関係を持っている。そのためユーザは、大まかなシミュレーション結果の傾向がわかるサムネイル画像から、これらを参照することも可能となる。

ユーザが選択したシミュレーション結果の詳細を重ね合わせて比較できる機能を **Flash** と **PHP** を用いて実現した。ユーザは、結果の詳細を比較したい場合、サムネイル画像、プログラム名、入力パラメータなどの情報から、データを選択する。ユーザが選択操作を行うと、サムネイル画像と対応関係があるシミュレーション結果を **Flash** から **POST** によって **PHP** へ通信し **PHP** から **MySQL** のクエリを実行し、シミュレーション結果のファイル名を取得する。その後、**csv** 形式のファイルを文字列操作によって各列に分解し、**PHP** から **POST** によって **Flash** へ通信し、**Flash** はグラフ化を行う。また、その数値データも同時に表示する。ユーザは、同様の手順を繰り返すことによって、複数のグラフを重ね合わせて比較することができる。図 18 は、グラフ生成システムが計算実行後に、ユーザに提供する **GUI** のスクリーンショットである。

グラフ生成システムは、計算実行時の入力パラメータ、乱数系列を文字列として出力し、計算結果をサムネイル画像として計算ごとにユーザへ提示する（図 18 の左上パネル）。ユーザがこれらの情報を基に左上パネル内の入力パラメータやサムネイル画像が表示されているエリアを選択すると、システムは結果の数値データを表示する（図 18 の右上パネル）。また、ユーザが **LINE CHART** ボタンを押すと、システムは結果のグラフを表示する（図 18 の下パネル）。その後、ユーザが左上パネル内の異なるエリアを選択することで、システムは結果の数値データを更新する（図 18 の右上パネル）。この際、ユーザが **ADD LINE CHART** ボタンを押すと、システムは結果のグラフを重ねて表示する（図 18 の下パネル）。ここで表示されているグラフは、左から **Q** 学習に用いられる行動数、目標達成回数、**GAIN** の値である。なお、図 18 のスクリーンショットに示されている上部の他の 2 つのタブは、それぞれシステムへのプログラムの登録や検索、プログラムの引数の入力を行う **GUI** である。

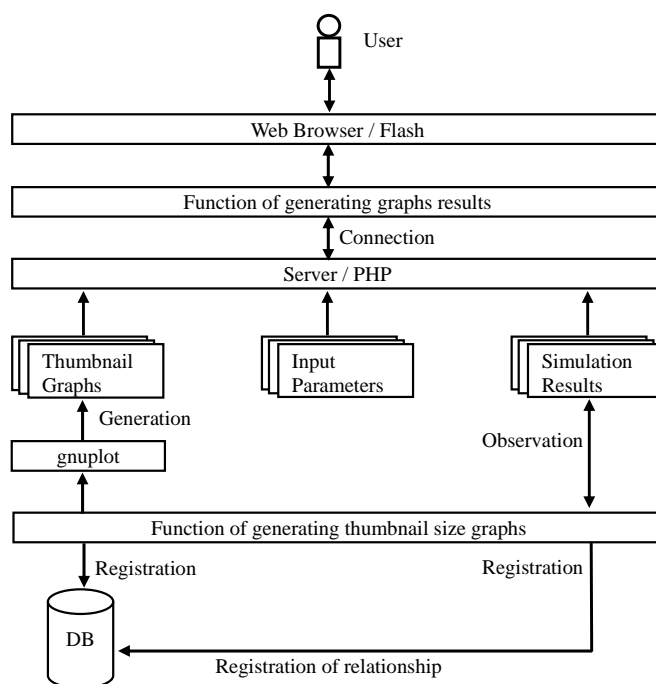


図 17 グラフ生成システム

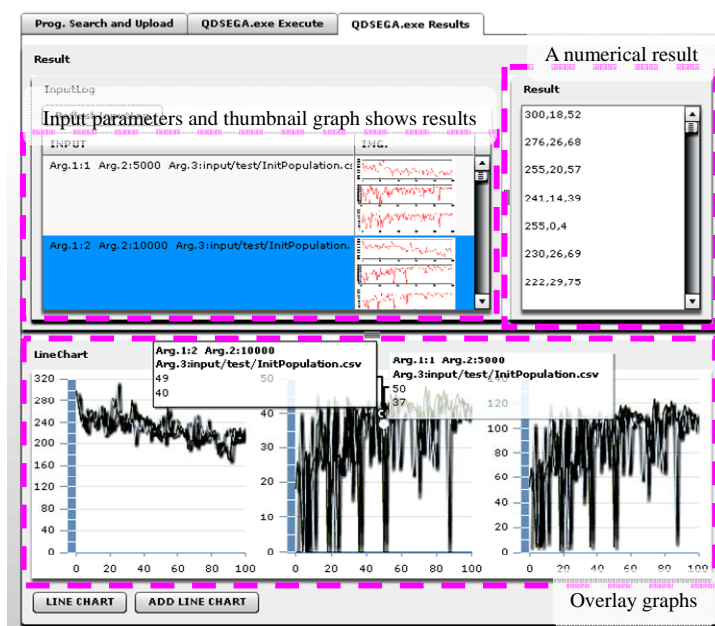


図 18 グラフ生成システムの GUI

4. 適用事例と結果

本システムに前述した QDSEGA[26](以下, 適用事例 1) と近傍交叉を用いた QDSEGA[31, 32] (以下, 適用事例 2), 削除アルゴリズムと近傍交叉を用いた QDSEGA[33, 34] (以下,

適用事例 3) の 3 種類のアプローチを用いたパラメータセットを用いるシミュレーションのプログラムを用いる。本論文ではシステムの評価のため、表 1 と表 2 より各アプローチを比較する際に重要となる乱数系列、Q 学習に用いる学習回数、近傍数 k_{sim} をパラメータとして用いて 220 ケースを適用事例とした。

適用事例 1 は、引数を 2 つ取る。引数 1 は、乱数系列を変更するための SEED 値を指定し、0~9 の 1 刻みの 10 通りとする。引数 2 は、QDSEGA 内の Q 学習に用いる学習回数を指定し、10000, 20000 とする。学習回数による学習効率の差をみるためにこれらの値を用いた。よって適用事例 1 では、20 ケースとなる。

適用事例 2, 適用事例 3 は、引数を 3 つ取る。引数 1, 引数 2 は適用事例 1 と同様の値を用いる。引数 3 は、近傍交叉の近傍数 k_{sim} を指定し、0~8 の 2 刻みの 5 通りとする。よって適用事例 2, 適用事例 3 では、それぞれ 100 ケースとなる。これらの適用事例から合計 220 ケースを実行する。

これらの 220 ケースを用いて 1 つ目のサブシステムである分散計算システムによる計算実行時間の評価を行う。シミュレーション結果から 15 個のファイルを用いて 2 つ目のサブシステムであるデータ管理システムによるデータ管理の作業時間の評価を行う。そして、220 ケースのシミュレーション結果から 15 個のファイルを用いて 3 つ目のサブシステムであるグラフ生成システムによるシミュレーション結果の比較作業時間の評価を行う。最後に、各サブシステムによって得られた評価からシステム全体の評価を行う。

本論文では、被験者による比較作業において、手法の比較を考慮し 3 つの適用事例をシステムへ適用した。また、被験者への負担を考慮し、手作業で行うデータ管理の作業と比較作業では、220 ケース中 15 個のファイルを用いた。

適用事例に対して、計算実行時間の評価として分散計算による時間の測定、データ管理の作業時間の評価として被験者の手作業によるデータの登録とシステムによる DB へのデータの登録を行い、その作業時間を測定した。さらに、シミュレーション結果の比較作業時間の評価として被験者に対して利用実験を行い、その作業時間を測定した。

計算実行時間の評価は、シミュレーション計算資源をアルゴリズムの研究者が利用している環境で行った。計算機の性能を表 5 に示し、図 19 に上記の 220 ケースの計算実行時間を示す。計算結果から 1 台で 8952 秒かかっていた計算が、8 台では 1152 秒で計算可能となった。

データ管理の作業時間の評価は、被験者の手作業によるシミュレーション結果の登録とデータ管理システムによる DB への自動登録を行い、その作業時間の比較によって行う。被験者は、強化学習アルゴリズムの開発者であり、適用事例のシミュレーション結果に慣れた被験者を 1 名、適用事例のシミュレーション結果のフォーマットを知らない被験者 4 名とした。被験者の手作業によるシミュレーション結果の登録は、被験者らの DB の知識の差を考慮し、シミュレーション結果の数値データを Microsoft Excel へ記録することとし

た。被験者による登録は、1 ファイル 1 登録の作業を 15 回行い、その平均時間を 1 ファイルあたりの作業時間として測定した。データ管理システムでの DB への登録は、被験者による登録と同様に、1 ファイル 1 登録の作業を 15 回行い、その平均時間を 1 ファイルあたりの作業時間として測定した。作業の概要を図 20 に示す。

各被験者の作業時間の平均とシステムの登録時間の平均を表 6 に示す。表 6 の被験者 1 が適用事例のシミュレーション結果に慣れた被験者である。被験者の手作業による作業時間の平均は、17.78 秒であり、データ管理システムでの 1 ファイルあたりの作業時間は 0.13 秒となった。

シミュレーション結果の比較作業時間の評価は、適用事例のシミュレーション結果のフォーマットを知らない被験者 5 名に対して行った。適用事例のシミュレーション結果は、時間ステップごとに、適用事例内の Q 学習に用いられる学習効率を示す評価関数の値が出力される。ユーザの作業は、学習効率が最も高い値で収束が早い結果が得られたアルゴリズムと入力パラメータの組合せを見つけ出すまでとした。作業の概要を図 21 に示す。被験者に対しグラフ生成システムを用いない場合と、用いた場合の比較作業時間の比較によって行う。

グラフ生成システムを用いない場合、上記の 220 ケースのシミュレーションからランダムに抽出した 15 個のシミュレーション結果の数値データを被験者へ提示した。その後、一般的なグラフ作成ソフトとして用いられている Microsoft Excel のグラフウィザード機能を用いて、学習効率が最も高い値で収束が早いものを選択するまでの時間を比較作業時間とした。グラフ生成システムを用いた場合、自動で生成されるグラフとグラフの重ね合わせ機能を用い、目的の組合せを選択するまでの時間を比較作業時間とした。シミュレーション結果の比較によって、アルゴリズムに対する最適なデータセットの比較作業時間を表 7 に示す。

グラフ生成システムを用いない場合と用いた場合で、最も作業時間の差が大きかった被験者 1 では、システムを用いない場合は 1815.9 秒、システムを用いた場合は 314.1 秒となった。また、全被験者でのグラフ生成システムを用いない場合の平均は 1495.5 秒、グラフ生成システムを用いた場合の平均は 356.8 秒となった。

最後に、各サブシステムによって得られた評価からシステム全体の評価を行う。その評価は、システムを用いない場合と、用いた場合の平均経過時間の比較により行う。平均経過時間は、システムを用いない場合、計算機 1 台を用いた計算実行時間、手作業によるデータ管理の作業時間の平均、手作業によるシミュレーション結果の比較作業時間の平均の合計とする。システムを用いた場合、分散計算システムにより計算機 8 台を用いた計算実行時間、データ管理システムによるデータ管理の作業時間、グラフ生成システムを用いたシミュレーション結果の比較作業時間の平均の合計である。また、シミュレーション結果の比較作業時間とファイルのデータ管理の作業の 15 個の計測結果と合わせるため、15 ケ

ースの分散計算時間を算出し比較する。図 22 と表 8 に平均経過時間を示す。

適用事例の 15 ケースに対する、それぞれのサブシステムによって短縮された時間を示す。計算実行時間は 610.3 秒から 78.5 秒、データ管理の作業時間は 266.7 秒から 1.9 秒、シミュレーション結果の比較作業時間は 1495.5 秒から 356.8 秒に短縮され、比較作業において最も時間的な効率が得られた。短縮された時間の割合は、計算実行時間が約 1/8、データ管理の作業時間が約 1/140、シミュレーション結果の比較作業時間が約 1/4 となり、データ管理の作業時間の短縮の割合が最も高かった。このことから、データ管理の作業や比較作業の効率化が重要であることがわかる。短縮の割合が最も高かったデータ管理の作業は、他の短縮された時間と比較して短縮された時間そのものは、264.8 秒と他の短縮時間（531.8 秒、1139.7 秒）よりも少ないが、手作業でのデータ管理がより煩雑な場合、時間短縮が短縮の割合から期待できる。また、このデータ管理の自動化を行わなければ、シミュレーション結果の比較作業での効率化は実現できない。

システム全体をみた場合、平均経過時間は、2372.5 秒から 437.2 秒となり、約 1/5 となった。この結果は、計算実行、データ管理の作業、比較作業を全て実現した結果である。仮に、計算時間のみを効率化した場合、平均経過時間は最大で約 4/5、比較時間のみを効率化した場合、最大で約 1/2 にしかならない。このことから、サブシステムの組合せが重要であるといえる。

表 5 用いた PC のスペック

CPU	Pentiumu4 – 3.4GHz
Main Memory	2G byte
Network	Gigabit Ethernet
OS	Windows XP pro SP2

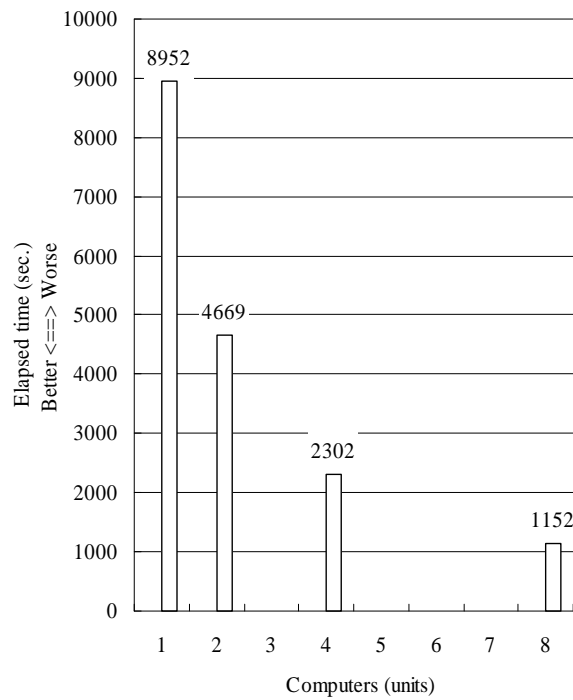


図 19 計算時間

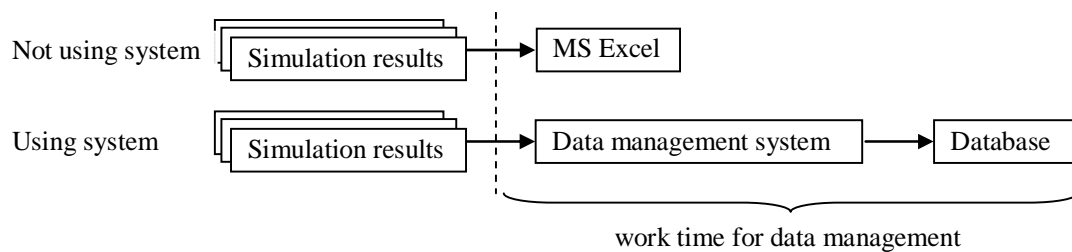


図 20 データ管理作業の概要

表 6 データ管理の作業時間

Examinee ID	Average work time (sec.)
Examinee 1	14.3
Examinee 2	13.8
Examinee 3	19.4
Examinee 4	20.5
Examinee 5	20.9
Data management System	0.13

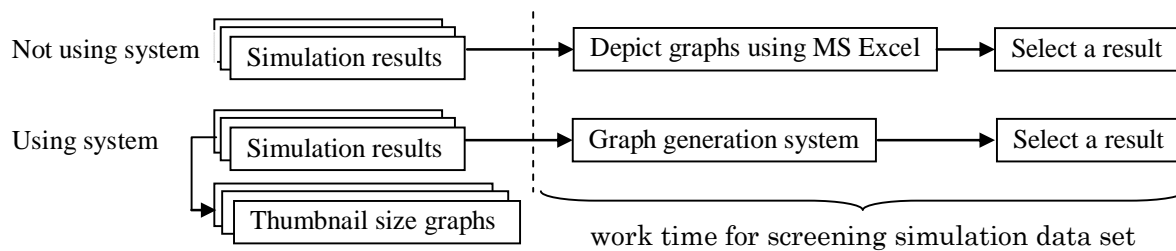


図 21 シミュレーション結果の比較作業の概要

表 7 シミュレーション結果比較の作業時間

Examinee ID	Not using system (sec.)	Using system (sec.)
Examinee 1	1815.9	314.1
Examinee 2	1055.7	442.6
Examinee 3	1508.0	582.4
Examinee 4	1696.9	269.4
Examinee 5	1401.2	175.6

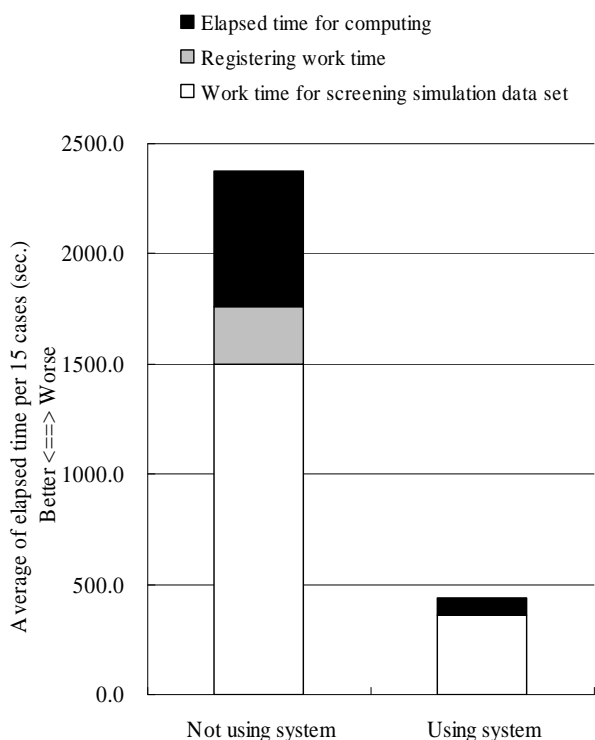


図 22 15 ケースにおける平均経過時間

表 8 15 ケースにおける平均経過時間

Measurement category	Not using system (sec.)	Using system (sec.)
Elapsed time for computing	610.3	78.5
Registering work time	266.7	1.9
Work time for screening simulation data set	1495.5	356.8
Total time	2372.5	437.2

5. システムの汎用化

これらの結果を踏まえ、より汎用的にパラメータセットを用いるシミュレーションへ適用するために、本システムのサブシステムである分散計算システムとグラフ生成システムの汎用化を行う。本論文で構築したシステムは、パラメータセットを用いるシミュレーションへの機能の組合せとその評価について議論した。しかし、DB への依存度が高く、各サブシステムが DB によって拘束されている状態である。そこで、他のパラメータセットを用いるシミュレーションへの適用を行うために、サブシステムを切り離すことで汎用化を行う。

ユーザの利用を考慮した場合、手軽に Web 上でシステムを利用したいユーザと、研究データをデータ漏洩の危険性がある Web 上でシステムを利用したくないユーザがいると考えられる。そこで、より幅広いユーザが利用できるようにデスクトップアプリケーションとして構築する。

5.1 分散計算システムの汎用化

分散計算システムの主な変更点を以下に述べる。システム上からユーザのプログラムの登録を行わず、ユーザが実行したいプログラムをローカルディスクから選択する。また、ユーザが分散計算に用いる計算機を指定する。汎用化した分散計算システムのスクリーンショットを図 23 と図 24 に示す。分散計算システムを用いて、子育てに関するアンケートからの医療機関選択ルールの抽出と同定[17]に適用した結果、1 台の計算機を用いた場合の実行時間 16 時間 20 分から 4 台の計算機を用いた場合の実行時間 4 時間 24 分となった。

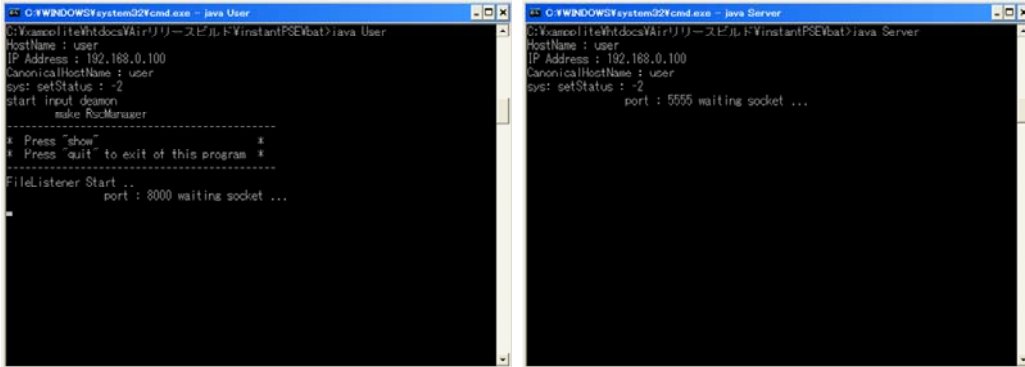


図 23 計算機の登録

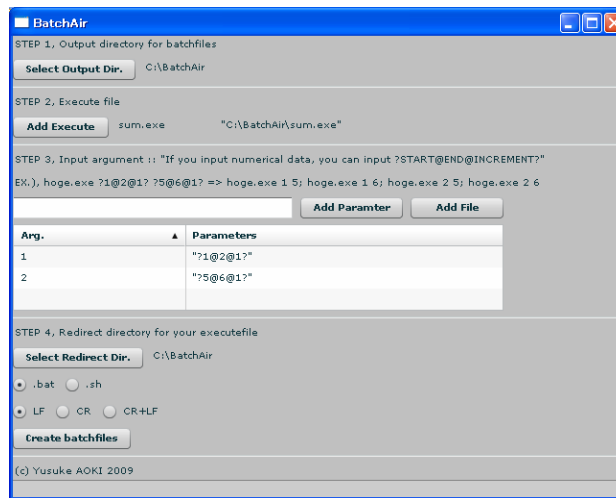


図 24 プログラムの選択と入力パラメータの入力

5.2 グラフ生成システムの汎用化

グラフ生成システムの主な変更点を以下に述べる。DB からのサムネイル画像やグラフ生成を行わず、ユーザがグラフ化を行いたい結果ファイルが格納されたディレクトリを選択する。また、時系列のみを判断するパラメータセットを用いるシミュレーション以外に対応するため、折れ線グラフだけでなく他のグラフにも対応させた。さらに、DB を用いることなく結果を検索するためにユーザの教示点に対する近接度に基づく検索機能を追加した。汎用化したグラフ生成システムのスクリーンショットを図 25 に示す。

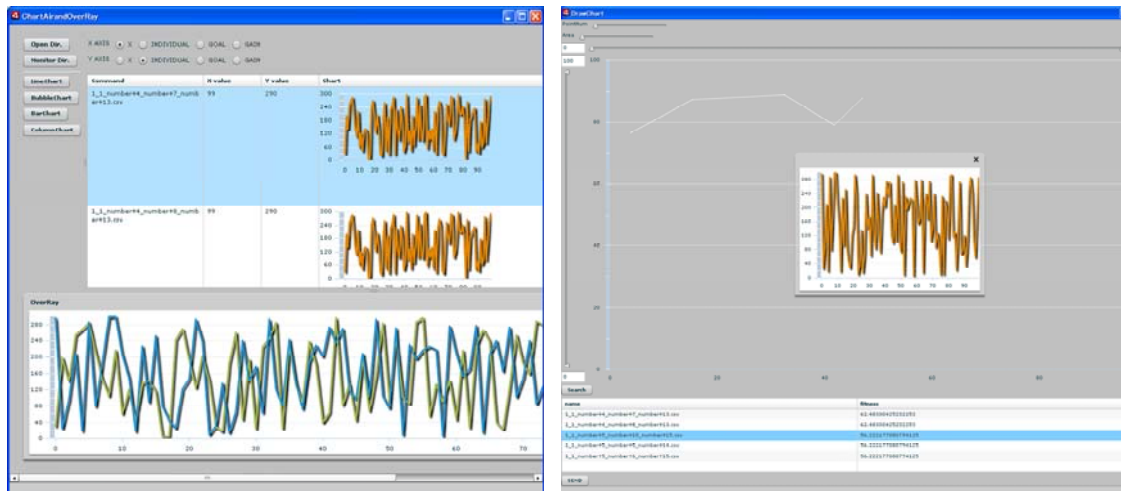


図 25 汎用化したグラフ生成システム

6. まとめと今後の課題

本論文では、パラメータセットを用いるシミュレーションを効率的に開発するために、パラメータセットを用いるシミュレーションの開発を行い、それに対する問題点を洗い出し、パラメータセットを用いるシミュレーションを行う研究者支援システムを構築した。適用事例において、サブシステムごとの評価とシステム全体の評価を行った。入力パラメータ、アルゴリズム、乱数系列、それぞれの組合せで複数回の計算を実行する必要があるユーザは、計算実行において、8 台の計算機でも分散計算による効果が得られた。また、手作業によって行ってきたデータ管理を、システムによって自動的に行うことで作業時間を短縮することができた。さらに、シミュレーション結果の比較作業によって最適なアルゴリズムとデータセットの組合せを選び出す際も、グラフの自動生成による効果が得られた。アルゴリズム開発において、これらのシステムを構成する要素はどれも必要不可欠であった。

これらの結果より、GA や強化学習アルゴリズムのような入力パラメータとアルゴリズムの変更を必要とするパラメータセットを用いるシミュレーションを効率的に開発する場合、本システムが持つサブシステムの組合せが有効であるといえる。本論文で構築したシステムによる計算実行時間やユーザの作業時間の短縮は、本来の目的であるパラメータセットを用いるシミュレーション開発の促進につながると期待できる。汎用化については、分散計算システムを切り離し、他の適用事例に適用したことで計算時間の効率化が行えることが分かった。

今後の課題として、多くのシミュレーションへの適用を目指すために、サブシステムごとに汎用化を行い、各サブシステムの評価を行う必要がある。また、ファイルフォーマットなどにより制限を受けるデータ管理システムの汎用化は行っていないが、今後、データ

管理システムの汎用化を行っていく必要がある。それに伴い、他のシミュレーションにおいて、どのサブシステムが効果的であり、どのサブシステムの組合せが有効であるか、また新たに必要な機能やサブシステムを探っていく必要がある。

謝辞

本研究の一部は、文部科学省社会連携研究推進事業（平成 17 年度～平成 21 年度）による私学助成を得て行われた。

参考文献

- [1] 伊庭齊志, 遺伝的アルゴリズム, 医学出版, 2002.
- [2] Richard S. Sutton, Reinforcement Learning Architectures for Animats, Proc. of the International Workshop on the Simulation of Adaptive Behavior: From Animals to Animats, MIT Press, pp.288-296, 1991.
- [3] Richard S. Sutton and Andrew G. Barto, 三上貞芳, 皆川雅章, 強化学習, 森北出版, 2000.
- [4] 荒井幸代, 宮崎和光, 小林重信, マルチエージェント強化学習の方法論 -Q-Learning と Profit Sharing による接近-, 人工知能学会論文誌, Vol.13, No.4, pp.609-618, 1998.
- [5] 畝見達夫, 実例に基づく強化学習法, 人工知能学会論文誌, Vol.7, No.4, pp.697-707, 1992.
- [6] 笠口隆重, 梅舘典也, 井上智文, 苫米地功, 牧野徹, 田子精男, グリッドでのゲノム解析に向けた問題解決環境の開発, 日本計算工学会論文集, Vol.2005, 20050030, 2005.
- [7] FLUENT, <http://ansys.jp/products/fluent/>
- [8] 中井純, 庄司朱澄, 門岡良昌, 田子精男, 流体研究者のための問題解決環境 Fluid e-Laboratory の構築, 計算工学講演会論文集, Vol.11, pp.553-556, 2006.
- [9] 田子精男, 大西尚樹, 門岡良昌, グリッドを利用した PIV 仮想研究所の構築, 雑誌 FUJITSU, Vol.55, No.2, pp.127-132, 2004.
- [10] ADVENTURE, <http://adventure.sys.t.u-tokyo.ac.jp/>
- [11] 塩谷隆二, 萩野正雄, 河合浩志, 三好昭生, 大規模解析のための web ベース CAE システムの開発, 日本計算工学会論文集, Vol.2007, 20070020, 2007.
- [12] Takashi Maeno, Susumu Date, Yoshiyuki Kido and Shinji Shimojo, Analytic Space Management for Drug Design Application, IPSJ Journal, Vol.41, No.6, pp.1234-1245, 2000.
- [13] John R. Rice and Ronald F. Boisvert, From Scientific Software Libraries to Problem-Solving Environments, Computing in Science and Engineering, Vol.3, No.3, pp.44-53, 1996.
- [14] 前田太陽, 青木悠祐, 村田忠彦, マルチエージェントシミュレーションのためのデータ管理機能を持った分散計算システムの開発, 計算工学講演会論文集, Vol.13, pp.987-990, 2008.
- [15] 青木悠祐, 前田太陽, 村田忠彦, マルチエージェントシミュレーションのためのデータ比較支援システムの開発, 第 11 回問題解決環境ワークショップ, pp.47-50, 2008.

- [16] 青木悠祐, 前田太陽, 村田忠彦, 強化学習アルゴリズム研究者の開発作業を支援する PSE システムの構築, 日本計算工学会論文集, Vol. 2009, 20090017, pp.1-7, 2009.
- [17] 十倉伸太郎, 村田忠彦, 松原光也, 子育てに関するアンケートからの医療機関選択ルールの抽出と同等, PG Lab ディスカッションペーパーシリーズ, Vol.27, pp.1-12, 2008.
- [18] 中村悦大, 茶本悠介, 村田忠彦, 中選挙区下での選挙競争を理解する, シミュレーションによるひとつの検討, 社会科学の実験アプローチ (河野勝・西條辰義編, 勁草書房), pp.85-114, 2007.
- [19] 中村悦大, 茶本悠介, 村田忠彦, 名取良太, 選挙制度と政党の目標に関するマルチエージェントシミュレーションの実践, 政策グリッドコンピューティングとマルチエージェントシミュレーション (村田忠彦・鶴飼康東編, 多賀出版), pp.119-141, 2008.
- [20] Peter Stone and Richard S. Sutton, Scaling Reinforcement Learning toward RoboCup Soccer, Proc. of International Conference on Machine Learning 2001, pp.537-544, 2001.
- [21] Peter Stone, Richard S. Sutton and Gregory Kuhlmann, Reinforcement Learning for RoboCup Soccer Keepaway, Adaptive Behavior, Vol.13, No.3, pp.165-188, 2005.
- [22] John J. Grefenstette, Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, Machine Learning, Vol.3, pp.225-245, 1988.
- [23] Christopher J. C. H. Watkins and Peter Dayan, Technical Note: Q-Learning, Machine Learning, Vol.8, pp.55-68, 1992.
- [24] 伊藤一之, 松野文俊, GAにより探索空間の動的生成を行う Q 学習, 人工知能学会論文誌, Vol.16, No.6, pp.510-520, 2001.
- [25] 伊藤一之, 松野文俊, QDSEGA による多足ロボットの歩行運動の獲得, 人工知能学会論文誌, Vol.17, No.4B, pp.363-372, 2002.
- [26] Kazuyuki Ito and Akio Gofuku, Hybrid autonomous control for multi mobile robots, Advanced Robotics, Vol.18, No.1, pp.83-99, 2004.
- [27] Kazuyuki Ito, Akio Gofuku, Hybrid autonomous control for heterogeneous multi-agent system, Proc. of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, pp.2500-2505, 2003.
- [28] Tadahiko Murata and Masatoshi Yamaguchi, Neighboring Crossover to Improve GA-Based Q-Learning Method for Multi-Legged Robot Control, Proc. of Genetic and Evolutionary Computation Conference 2005, pp.145-146, 2005.
- [29] Tadahiko Murata and Masatoshi Yamaguchi, Multi-Legged Robot Control Using GA-Based Q-Learning Method With Neighboring Crossover, Frontiers in Evolutionary Robotics, IN-TECH, pp.341-152, 2008.
- [30] 高橋治, 木村周平, 小林重信, 交叉敵突然変異による適応的近傍探索-騙しのある多峰性関数の最適化-, 人工知能学会論文誌, Vol.16, No.2C, pp.175-184, 2001.
- [31] 青木悠祐, 村田忠彦, マルチエージェントの行動制御のための QDSEGA における近傍交叉の適用法, 第 23 回ファジィ・システム・シンポジウム講演論文集, pp.615-618, 2007.
- [32] Tadahiko Murata and Yusuke Aoki, Developing Control Table for Multiple Agents Using GA-Based

- Q-Learning with Neighboring Crossover, Proc. of IEEE Congress on Evolutionary Computation: CEC 2007, pp.1462-1467, 2007.
- [33] 青木悠祐, 村田忠彦, QDSEGA を用いたマルチエージェント行動制御における Q テーブルの最小化, 日本知能情報ファジィ学会 ECOmp 研究部会研究発表会, 4 pages, 2008.
- [34] Tadahiko Murata and Yusuke Aoki, GA-Based Q-Learning to Develop Compact Control Table for Multiple Agents, Evolutionary Computation, IN-TECH, 14 pages, 2010. (in printing)
- [35] 前田太陽, 田子精男, 協調型問題解決環境システムのためのフレームワークシステム, 日本計算工学会論文集, Vol.2006, 2006009, 2006.
- [36] Shigeo Kawata, Takayuki Teramoto, Hideyuki Sugiura, Yuichi Saitoh and Yoshikazu Hayase, Mathematical Modeling Support in a Distributed Problem Solving Environment for Scientific Computing, Proc. of the Second IEEE International Conference on e-Science and Grid Computing, pp.98-104, 2006.
- [37] 武田真, 前田太陽, 梅舘典也, 加藤佳則, 田子精男, LAN 内の分散計算のための問題解決環境 “Instant PSE” , 計算工学講演会論文集, Vol.12, pp.695-698, 2007.