

# 巡回セールスマン問題に対する 並列コンサルタント誘導型探索アルゴリズム

榎原 博之<sup>1,a)</sup> 中山 弘基<sup>1,†1</sup> 飯田 修平<sup>1</sup> 長辻 亮太<sup>1</sup>

受付日 2015年4月3日, 採録日 2015年10月2日

**概要:** 近年, メタヒューリスティクスは組合せ最適化問題を解く手法として多くの研究が行われている。最近の研究では, コンサルタント誘導型探索 (CGS) と呼ばれる新しいメタヒューリスティクスが提案されている。本研究では, CGS を用いた巡回セールスマン問題 (TSP) に対する並列アルゴリズムを提案する。アルゴリズムの並列化では, CGS における仮想人間をそれぞれの計算機の各プロセッサコアに割り当てることで効率良く解の探索を行う。また, 仮想人間の集団を複数のサブ集団に分割し, 各サブ集団どうしで仮想人間の移住を行う島モデルを CGS に取り入れる。10 台の計算機を用いた性能評価実験を行い, 都市数が 5,000 の TSPLIB のベンチマーク問題例に対して 5%未満の誤差率を達成することを示す。

**キーワード:** コンサルタント誘導型探索, PC クラスタ, 並列処理, メタヒューリスティクス, 組合せ最適化問題

## Parallel Consultant-guided Search Algorithm for Traveling Salesman Problem

HIROYUKI EBARA<sup>1,a)</sup> KOKI NAKAYAMA<sup>1,†1</sup> SHUHEI IIDA<sup>1</sup> RYOTA NAGATSUJI<sup>1</sup>

Received: April 3, 2015, Accepted: October 2, 2015

**Abstract:** Metaheuristic algorithms have been studied as a method for solving combinatorial optimization problems. Recently, the Consultant-Guided Search (CGS) for solving the Traveling Salesman Problem (TSP) has been proposed. In this paper, we propose a parallel method which assigns virtual consultants and virtual clients of the CGS to processes of computers, and calculates an approximation solution effectively for the TSP. In addition, we introduce the island model to increase the diversity of the solution. We execute computer experiments with the benchmark instances (TSPLIB) by 10 quad-core computers. Our algorithm provides a solution with less than 5% error rate for problem instances of 5,000 cities.

**Keywords:** consultant-guided search, PC cluster, parallel processing, metaheuristics, combinatorial optimization problem

### 1. まえがき

近年, 計算機の性能と現代社会における計算機への依存度は年々増加している。また量子力学や物理的なシミュレーションなど, 計算機に要求される計算量も上昇している。このような大規模な計算を行う場合, 計算対象を計算

機での処理が可能な形に定式化し, 組合せ最適化問題として扱うことがある。組合せ最適化問題とは, 与えられた制約条件を満たす組合せ集合の中から目的関数の評価を最大あるいは最小にする組合せを求める問題である。具体例として, 巡回セールスマン問題やナップサック問題, 2次割当て問題などがあげられる。

組合せ最適化問題の解法には大きく分けて厳密解法と近似解法の2種類がある。問題の規模が大きくなると現実的な時間で厳密解を求めることが困難となるため, 現実的な時間と計算能力である程度良い解を求めることができる近

<sup>1</sup> 関西大学  
Kansai University, Suita, Osaka 564-8680, Japan

<sup>†1</sup> 現在, 株式会社日立システムズ  
Presently with Hitachi Systems, Ltd.

<sup>a)</sup> ebara@kansai-u.ac.jp

似解法が多く用いられている。最近では、近似解法と様々な戦略を組み合わせるより精度の高い解を求める枠組みであるメタヒューリスティクスの研究がさかんに行われている。

メタヒューリスティクスは、初期解から探索戦略に従って局所探索を繰り返し、一定の条件を満たすと探索を終了する。代表的な探索戦略には、遺伝的アルゴリズム (GA) [1], シミュレーテッド・アニーリング法 (SA) [1], アントコロニー最適化 (ACO) [1], [2], ビーコロニー最適化 (ABC) [1] などがある。また、これらの探索戦略を用いてより高速で精度の高い探索を実現する方法として並列化がある。並列化モデルには様々な種類が存在し、それぞれ負荷分散の方法が異なる。その中でも島モデル [3] は並列化効率が高いことで知られる。

本研究では、コンサルタント誘導型探索 (CGS) における 2 種類の仮想人間であるコンサルタントとクライアントを 1 人ずつ各計算機の 1 つのプロセッサコアに割り当て、島モデルを取り入れた、巡回セールスマン問題に対する並列化計算モデルを提案する。提案手法では、従来の CGS と異なり、1 人の仮想人間がコンサルタントかクライアントのどちらか一方のみを演じることで探索を進める。仮想人間を 1 人ずつ各計算機の 1 つのプロセッサコアに割り当てることで、コンサルタントを演じる仮想人間は戦略の構築と更新とクライアントへのアドバイスのみを行い、クライアントを演じる仮想人間はコンサルタントを選択して解の構築のみを行う。そのため、従来法と違ってクライアントとコンサルタントの人数は同数である必要はない。また、島モデルを取り入れ、全体の仮想人間集団をいくつかの島に分散させる。島内のコンサルタントは同一島内のクライアントに対してアドバイスを提供する。まず、評価実験により最適なコンサルタントとクライアントの人数比を求める。提案手法を用いることで、著者らが文献 [8] で提案した CGS と ACS を組み合わせた並列ハイブリッドアルゴリズムや逐次処理の CGS を単純に並列化したプロセッサコア間の通信が発生しない手法と比較して、より高い探索性能を実現することを目指す。

本論文は、以下の 7 章で構成される。2 章では、巡回セールスマン問題について述べる。3 章では、メタヒューリスティクスの 1 つである CGS のアルゴリズムについて述べる。4 章では、関連研究について述べる。5 章では、提案手法について述べる。6 章では、提案手法を並列システムに実装し、実際に動作させた結果についての性能評価と考察を行う。最後に 7 章では、本論文のまとめと今後の課題について述べる。

## 2. 巡回セールスマン問題

巡回セールスマン問題 (TSP) とは、都市の集合が与えられた際にすべての都市を 1 度だけ訪問して出発点の都市

に戻ってくるハミルトン閉路の中で最短の経路長を持つものを求める問題である。また、NP 困難である組合せ最適化問題として知られ、都市数が増加するに従って巡回経路の組合せは指数関数的に増加する。

$n$  個の都市の集合  $V = \{1, 2, \dots, n\}$  と都市  $i$  と都市  $j$  間の移動コストを  $C_{ij}$  とすると、目的関数  $f(x)$  は以下のように数式化される。

$$f(x) = \sum_{k=1}^{n-1} C_{x(k)x(k+1)} + C_{x(n)x(1)} \quad (1)$$

$x(k) = i$  は  $k$  番目に訪れる都市が  $i$  であることを表す。式 (1) で示される目的関数  $f(x)$  を最小化する問題が TSP である。

## 3. コンサルタント誘導型探索

### 3.1 概要

コンサルタント誘導型探索 (CGS) とは、2010 年に Serban Iordache が提案した新しいメタヒューリスティクス [4] であり、現実の人間がコンサルタントから受けたアドバイスをもとに方針を決定する行動を模倣したアルゴリズムである。現実の人間は非常に複雑な行動をとるが、CGS では単純な規則にのみ従う仮想的な人間を用いる。CGS における仮想人間は 2 つの役割を持ち、各反復において両方の役割を同時に演じながら行動する。上記の 2 つの役割はコンサルタントとクライアントと呼ばれる。

コンサルタントの役割は解を構築する際のアドバイスをクライアントに提供することである。探索開始時にコンサルタントはこのアドバイスのもととなる解を構築する。コンサルタントが構築した解は CGS において戦略と呼ばれる。

クライアントの役割はコンサルタントのアドバイスをもとに解を構築することである。クライアントは各反復において 1 人のコンサルタントを選択し、そのコンサルタントの戦略に従って解を構築する。

また、仮想人間には 2 種類のモード、*sabbatical mode* と *normal mode* があり、それぞれ異なった行動をとる。以下に各モードについての詳しい説明を示す。

### 3.2 Sabbatical Mode

*sabbatical mode* の仮想人間は、クライアントとして動作せず、コンサルタントとして戦略となる解を構築し、その経路を保存する。また、このモードではクライアントへのアドバイスは提供しない。解の構築には都市間の距離情報を利用した擬似確率的方法 (pseudorandom proportional rule) を用いる。各コンサルタントは以下の式に従って都市  $j$  を選択し、解の構築を行う。

$$j = \begin{cases} \operatorname{argmin}_{l \in N_{i,k}} \{d_{il}\}, & \text{if } a \leq a_0 \\ J, & \text{otherwise} \end{cases} \quad (2)$$

ここで、 $\mathcal{N}_{i,k}$  は都市  $i$  に滞在する仮想人間  $k$  の実行可能な近傍を表す。ここで、実行可能な近傍とは都市  $i$  と隣接している都市の中で未訪問の都市の集合である。 $d_{il}$  は都市  $i$  と都市  $l$  の間の距離を表す。 $a$  は 0 以上 1 以下の一様乱数であり、 $a_0$  は 0 以上 1 以下のパラメータである。 $J$  は以下の式によって与えられる確率分布から選択された都市となる。

$$p_{ij,k} = \frac{(1/d_{ij})^\beta}{\sum_{l \in \mathcal{N}_{i,k}} (1/d_{il})^\beta} \quad (3)$$

$\beta$  は都市間の距離の影響を決定するパラメータを表す。

*sabbatical mode* では戦略の構築はあらかじめ指定した回数だけ繰り返され、構築された戦略候補の中で最も良いものを戦略として保存する。また、探索開始時はすべての仮想人間が *sabbatical mode* に属する。戦略が設定された後、*normal mode* へ移行する。

### 3.3 Normal Mode

*normal mode* の仮想人間は、クライアントとしてコンサルタントからのアドバイスをもとに解を構築する。同時に、コンサルタントとして別のクライアントにアドバイスする。

解の構築にあたって、はじめに *normal mode* のコンサルタントを 1 人だけ選択する。コンサルタントの選択は、各コンサルタントの持つ戦略の経路距離の逆数によって与えられる *personal preference*、および各コンサルタントの評判を表す *reputation* の 2 つの値が影響する。この *reputation* は値が大きいほど良いアドバイスを提供したコンサルタントであることを示す。この値は *normal mode* に属するコンサルタントのみに与えられる。コンサルタント  $k$  は以下の式によって与えられる確率によってクライアントに選択される。

$$p_k = \frac{reputation_k^\alpha \cdot personal\ preference_k^\gamma}{\sum_{c \in C} reputation_c^\alpha \cdot personal\ preference_c^\gamma} \quad (4)$$

ここで、 $C$  は選択可能なコンサルタントを表す。すなわち、*normal mode* に属する仮想人間の集合を表す。また、 $\alpha$  はコンサルタントの *reputation* の影響を決定するパラメータ、 $\gamma$  は *personal preference* の影響を決定するパラメータを表す。

次に、クライアントは選択したコンサルタントが持つ戦略に基づいて解を作成する。クライアントの解の構築にも擬似確率的方法を用いる。各クライアントは以下の式に従って都市  $j$  を選択し、解の構築を行う。

$$j = \begin{cases} v, & \text{if } v \neq null \wedge q \leq q_0 \\ argmin_{l \in \mathcal{N}_{i,k}} \{d_{il}\}, & \text{if } (v = null \vee q > q_0) \wedge b \leq b_0 \\ J, & \text{otherwise} \end{cases} \quad (5)$$

ここで、 $b, q$  は 0 以上 1 以下の一様乱数である。 $b_0, q_0$  は 0 以上 1 以下のパラメータを表す。 $J$  は *sabbatical mode* と同様で式 (3) から選択された都市である。 $v$  はコンサル

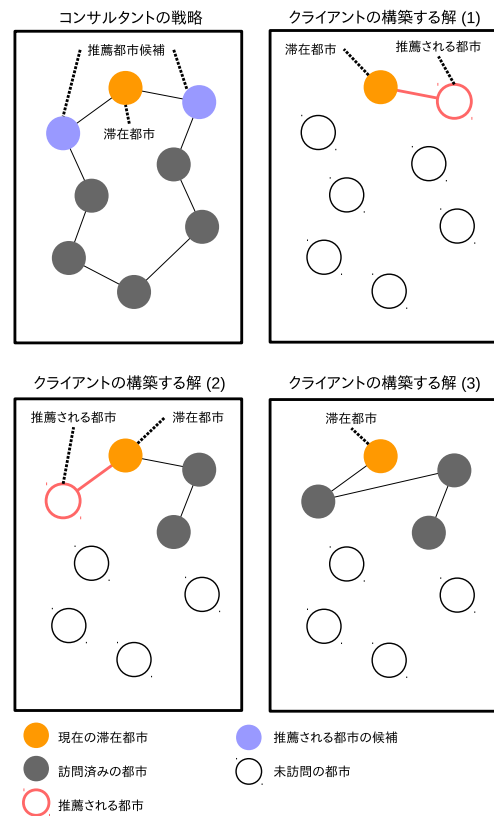


図 1 コンサルタントの推薦する都市の決定方法

Fig. 1 Method of determining the city recommended by consultants.

ントからアドバイスとして推薦された都市を表す。この都市はコンサルタントの持つ戦略から選択される。以下に推薦する都市の選択方法を示す。

はじめに、クライアントが現在滞在する都市  $i$  を、選択したコンサルタントの戦略から探す。次に、その都市と前後に連続している都市を推薦する都市の候補としてあげる。そして、候補の中から以下の場合に分けて推薦する都市を決定する (図 1)。

- (1) 前後に連続する都市の両方ともにクライアントが未訪問の場合、都市  $i$  から都市間の距離が短い都市を推薦する。
- (2) 前後に連続する都市の片方のみクライアントが未訪問の場合、未訪問の都市を推薦する。
- (3) 前後に連続する都市の両方ともにクライアントが訪問済みの場合、コンサルタントは都市を推薦できない。クライアントは式 (2) の都市選択方法で次に訪問する都市を決定する。

### 3.4 コンサルタントの戦略と reputation の更新

*normal mode* に属するクライアントの解構築と *sabbatical mode* に属するコンサルタントの戦略構築が終了した後、*normal mode* に属するコンサルタントの戦略と *reputation* を更新する。クライアントに選択されたコンサルタントの

reputation は、そのクライアントの構築した解によって変動する。もし、構築された解がコンサルタントの戦略よりも良い場合はこれを「成功」と呼び、以下の式のように成功した回数だけコンサルタント  $k$  の reputation に加算される。

$$reputation_k \leftarrow reputation_k + success\ count \quad (6)$$

ここで、success count は成功した回数を表す。また、この「成功」が発生した場合、現在のコンサルタントの戦略を成功したクライアントの解に置き換える。

さらに、構築された解が現時点での最良解であった場合は、以下の式に従ってコンサルタント  $k$  の reputation に追加でボーナスが加算される。

$$reputation_k \leftarrow reputation_k + bonus \quad (7)$$

ここで、bonus はあらかじめ設定されたボーナスの加算値を表す。また、コンサルタントの reputation は増加するだけでなく、各反復において以下の式に従って減少する。reputation の減少は実際のコンサルタントの評判が時間の経過によって風化することを模している。

$$reputation_k \leftarrow reputation_k(1 - r) \quad (8)$$

ここで、 $r$  は fading rate と呼ばれる減少の割合を表す。fading rate は、reputation が高い順にコンサルタントを列挙したときの上位数人の過去  $w$  回で成功した回数の和を用いて以下の式により決定する。計算に用いるコンサルタントの人数は fading ranks というパラメータ値で表される。

$$r = r_0 \left( 1 + \frac{s_w}{\sqrt{1 + (\frac{s_w}{f})^2}} \right) \quad (9)$$

ここで、 $s_w$  は成功した回数の和を表し、 $r_0$  は減少率に関するパラメータを表す。また、 $f$  は以下の式により与えられる。

$$f = \left( \frac{1}{r_0} - 1 \right) \left( 1 - \frac{1}{\sqrt{k_w}} \right) \quad (10)$$

$k_w$  は成功しなかった場合の減少率よりも、成功した回数が非常に多い場合の減少率がどれほど大きいかを表すパラメータである。

最後に、コンサルタント  $k$  の reputation があらかじめ設定された最大値を超過していた場合、以下の式に従って reputation を最大値に設定する。

$$reputation_k \leftarrow \max\ reputation \quad (11)$$

$\max\ reputation$  は reputation の最大値を表す。

### 3.5 モードの更新

コンサルタントの戦略と reputation の更新後、コンサル

タントのモードを更新する。sabbatical mode で一定回数の反復を経て戦略が設定された場合は normal mode へ移行する。移行の際にコンサルタントの reputation を初期値に設定する。また、normal mode において reputation があらかじめ設定された最低値を下回った場合は、sabbatical mode へ移行する。これを sabbatical leave と呼び、発生した場合は現在の戦略を捨て、sabbatical mode で新たな戦略を構築する。

### 3.6 アルゴリズム

以下に CGS のアルゴリズムを示す。また、CGS のフローチャートを図 2 に示す。

- (1) 初期化  
仮想人間を作成し、sabbatical mode に設定する。
- (2) 探索  
sabbatical mode であれば、式 (2) に従って戦略の構築を行う。normal mode ならば、式 (4) に従ってコンサルタントを選択し、式 (5) に従って解の構築を行う。
- (3) 戦略、reputation の更新  
クライアントが成功した場合、クライアントが選択したコンサルタントの戦略を更新し、成功回数を加算する。また、式 (6)~(11) に従って各コンサルタントの reputation を更新する。
- (4) モードの更新  
sabbatical mode であり、戦略をあらかじめ決められた回数だけ構築した場合は normal mode へ移行する。normal mode であり、コンサルタントの reputation があらかじめ設定された最低値を下回った場合は、現在の戦略を捨て sabbatical mode へと移行する。
- (5) 終了判定  
指定の終了条件を満たした場合、探索を終了する。

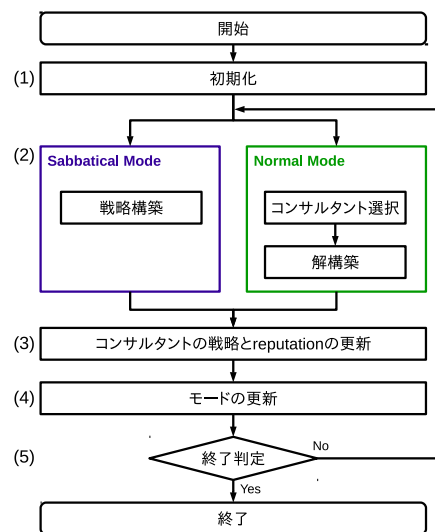


図 2 CGS のフローチャート  
Fig. 2 Flowchart of CGS.

## 4. 関連研究

Iordache の研究 [5] では、TSP を解くことに特化させた CGS-TSP を提案している。CGS-TSP とは、CGS に局所探索として 3-opt が組み込まれたものである。また、提案手法を改良した CGS-TSP-C では、コンサルタントがクライアントへ提供するアドバイスに *confidence* という概念を取り入れる。この概念は、クライアントが現在滞在する都市からコンサルタントの推薦する都市への枝ごとに *confidence* の強さを重みとして付与し、その強さによってクライアントがアドバイスを受け入れる確率変動するということである。この枝の重みは、コンサルタントの戦略が更新されるごとに変動する。各反復における更新の際に、すべての枝の集合からコンサルタントの戦略の経路を構成するすべての枝に対して *confidence* の強さを 1 ずつ加算する。以下の式によってクライアントがコンサルタントのアドバイスを受ける確率に影響するパラメータである  $q_0$  が与えられる。

$$q_0 = \begin{cases} q_{max} + s \cdot \frac{q_{max} - q_{min}}{s_{max}}, & \text{if } s < s_{max} \\ q_{max} & , \text{ otherwise} \end{cases} \quad (12)$$

ここで、 $s$  は推薦する枝に対する *confidence* の強さの値を表す。また、 $q_{min}$ 、 $q_{max}$ 、 $s_{max}$  はそれぞれパラメータを表す。otherwise が選択された場合は、その枝に対する *confidence* の強さの値を 0 に設定する。

性能評価実験では、提案手法である CGS-TSP と CGS-TSP-C、比較に用いる Ant Colony System (ACS) [6]、Max-Min Ant System (MMAS) [7] を用いて評価を行い、局所探索を取り入れた CGS の有効性を示した。さらに、CGS-TSP にコンサルタントのアドバイスに対する *confidence* の強さを取り入れた CGS-TSP-C によって結果を改善した。これらの結果から局所探索を組み合わせた場合であっても、CGS は ACS、MMAS と競い合うことが可能であることを報告した。

CGS と ACS を組み合わせた並列ハイブリッドアルゴリズムを提案する著者らの研究 [8] では、ACS における蟻のフェロモン情報を CGS の仮想人間であるコンサルタントの戦略に引き継ぎ、情報の共有を行う手法を示した。この手法は、ACS と CGS の 2 段階で構成され ACS での探索 (フェーズ 1) を行った後に CGS での探索 (フェーズ 2) を行う。フェーズ 2 では、フェーズ 1 の ACS によって得られた解をコンサルタントの戦略に設定して CGS が実行される。また同時にフェーズ 1 で得られたフェロモン情報を引き継ぎ、その情報の更新が随時実行される。フェーズ 2 において終了条件が満たされた場合、再びフェーズ 1 に戻る。そして、並列化では全体のフェロモン情報の更新時に、各計算ノードが持つ最良解の共有を行う。評価実験では、提案手法と単純並列 CGS、ACS を比較し、すべてに

おいて提案手法が優れた結果となった。この結果は周期的に最良解の共有を行い、解の多様性が増加したためと推測される。また、集中的な探索を行う ACS の解をコンサルタントの戦略に用いることで、クライアントに的確なアドバイスを提供できるものと考えられる。

Deepanandhini らの研究 [9] では、ジョブショップスケジューリング問題 (JSP) に CGS を適用し、効率的な探索が実行可能であることを示した。JSP とは、順序関係のある複数の作業を複数の機械で処理する際に、全体の処理時間が最小になるように機械の稼働スケジュールを決定する問題である。JSP のベンチマーク問題例を用いて行った動作実験では、いくつかの問題において最適解に到達することができたと報告している。また、最適解に至らなかった問題でも実行可能解に至ったことを確認している。今後は、他のアルゴリズムと組み合わせることで、さらなる性能の向上が期待されると報告している。

他の CGS に関する研究では、2 次割当て問題 (QAP) に CGS を適用した Iordache の研究 [10] がある。

Jovanovic らの研究 [11] では、ACO の並列化に鳥モデルを取り入れた並列アルゴリズムを提案している。ACO における蟻のエージェントを複数のコロニーに分割し、接続されたコロニーどうして解の上書き、交換を行う。各コロニーが形成するトポロジは 4 種類あり、すべての鳥と接続し解の上書きを行う完全結合型、最良解が最も悪かったコロニーと接続し解の上書きを行う部分交換型、指定したコロニーに対して解の上書きを行う固定リング型、隣接したコロニーと解の交換を行う双方向リング型を実装している。提案手法を最小重み点被覆問題に適用し、並列化にともなう各トポロジの性能を比較したところ、特に優れていたトポロジは双方向リング型と固定リング型であることを報告した。最良解を上書きせずに交換を行う双方向リング型では特に良い性能を示した。以上のことから、トポロジによっては最良解を上書きするよりも交換の方が良い結果になると報告している。

Arnaldo らの研究 [12] では、問題の構造と鳥モデルのトポロジの相関性を調査している。比較するトポロジは固定リング型、スター型、完全結合型の 3 種類である。Multi-Skill Call Center Problem (MSCCP) に鳥モデル型並列 GA を適用して得られた結果では、固定リング型が最も良い結果を示した。固定リング型では鳥数が増加するほど他のトポロジよりも性能が向上し、多様性の維持も可能であると報告している。

## 5. 提案手法

### 5.1 概要

通常の CGS では、1 人の仮想人間がコンサルタントとクライアントの両方を同時に演じることで探索を進める。提案手法では通常の CGS と異なり、1 人の仮想人間がコンサ

ルタントかクライアントのどちらか一方のみを演じることで探索を進める。したがって、コンサルタントを演じる仮想人間は戦略の構築と更新とクライアントへのアドバイスのみを行い、クライアントを演じる仮想人間はコンサルタントを選択して解の構築のみを行う。このような仮想人間を1人ずつ各計算機の1つのプロセッサコアに割り当てることでコンサルタントはコンサルタントの役割に、クライアントはクライアントの役割に集中させる。そのため、クライアントとコンサルタントの人数は同数である必要はない。本研究では、最適なコンサルタントとクライアントの人数比を実験により求め、以降の実験ではその人数比で実験する。また、島モデルを取り入れ全体の仮想人間集団をいくつかの島に分散させる。そして、島の中のコンサルタントは同一の島に属するクライアントに対してアドバイスを提供する。提案手法を用いることで、文献[8]で提案されている ACCGS (Ant Colony Consultant-Guided Search) や逐次処理の CGS を単純に並列化したものと比較して、より良い精度の解を短時間で提供することを目指す。

## 5.2 並列化

並列化モデルとして、各計算機の1つのプロセッサコアに対して1人の仮想人間を割り当てる。割り当てられた仮想人間は、コンサルタントかクライアントのどちらか一方を演じながら、他のプロセッサコアに割り当てられた仮想人間と協調して解の探索を行う。この並列化モデルでは、逐次処理の CGS を複数の計算機で実行する場合の総仮想人間数よりも仮想人間数が少なくなるため仮想人間1人あたりの探索性能が向上する。関連研究で用いられた CGS の仮想人間数は、Iordache の研究 [4] では、 $3 + 1,400/n$  となっている。 $n$  は TSP 問題の都市数を表す。著者らの研究 [8] では 20 人から 30 人となっている。また、CGS ではコンサルタントとクライアントの人数が同数であったが、提案手法ではコンサルタントとクライアントの人数を可変にすることができる。たとえば、コンサルタント数をクライアント数より少なく設定することで、コンサルタントの戦略を集中的に改善して探索を進めることができると考えられる。反対に、コンサルタント数をクライアント数より多く設定することで、コンサルタントの戦略を多様にすることもできると考えられる。

6章の評価実験では、管理サーバ1台、計算ノード(クワッドコア)10台でPCクラスタを構成して実験を行う。最初に、最適なコンサルタントとクライアントの人数比を求める実験を行う。計算ノードの総プロセッサコア数が40となっているため仮想人間の合計を40人とする。複数の問題例に対して、コンサルタントとクライアントの人数比を変更して最適な人数比を求める。

また、この並列化モデルでは仮想人間が割り当てられたプロセッサコアどうしで通信が発生する。通信が必要な動

作は、クライアントがコンサルタントを選択する動作と、クライアントの構築した解をコンサルタントが成功か成功でないかを判定する動作の2つである。クライアントがコンサルタントを選択する動作では、コンサルタントの戦略と *reputation* を全クライアントに送信する必要がある。クライアントの構築した解に対してコンサルタントが成功か成功でないかを判定する動作では、クライアントが構築した解を選択したコンサルタントに送信する必要がある。以上あげた2つの通信動作を実行して探索を進める。

## 5.3 島モデル

提案手法に島モデルと呼ばれる並列化モデルを取り入れる。1つのプロセッサコアに割り当てた1人の仮想人間で構成される母集団を複数のサブ母集団(島)に分割する。そして、その島の中でコンサルタントはクライアントにアドバイスを提供し、クライアントはコンサルタントのアドバイスをもとに解の構築を行う。島内のクライアントはその島特有のパラメータを用いて解を構築する。島ごとに異なるパラメータを付与することで解の多様性を増加させる。また、島内のコンサルタントは一定周期ごとに戦略と *reputation* を保持しつつ別の島へ移住する。異なる環境の島へ移住することによって、コンサルタントは多種多様なクライアントが構築する解を戦略に反映することができる。移住するコンサルタントは島内のコンサルタントからランダムに選択される。

本実験では島数を4に設定する。コア単位で仮想人間を割り当てている関係で仮想人間数は40となり、島内の仮想人間数を同一にするために島数は40の約数である必要がある。また、コンサルタントはランダムに少なくとも1人が移住するため最低でも2人必要である。島数の候補としては、2, 4, 8が考えられる。島数2では、島モデルとして島数が少なすぎる。島数8では、各島の仮想人間数が5人となり、コンサルタント2、クライアント3となる。クライアントが少なすぎるため、コンサルタントとクライアントの人数比を柔軟に対応できない。これらの理由から島数4を選択する。

島のトポロジに関して、島数4では複雑なトポロジを採用することはできない。さらに、文献[11]にも比較的単純なリング型が有効であると述べられているため、予備実験として、固定リング型、双方向リング型、交換リング型について評価した結果、固定リング型を選択する。固定リング型とは移住先を固定したリング状のトポロジであり、移住動作のたびに移住先が変化することはない。図3に固定リング型トポロジの移住動作を示す。

## 5.4 アルゴリズム

以下に本論文における提案手法のアルゴリズムを示す。また、下記に列挙する一連の動作のフローチャートを図4

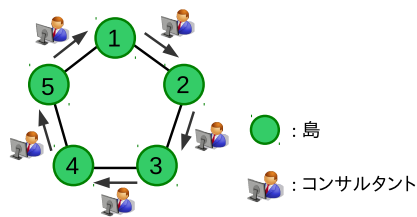


図 3 固定リング型トポロジの移住動作

Fig. 3 Migration of ring topology.

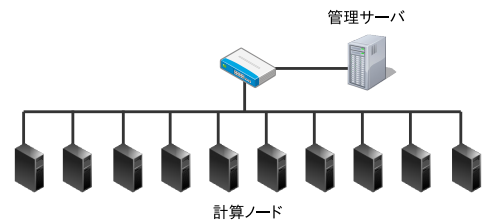


図 5 実験環境

Fig. 5 Experiment system.

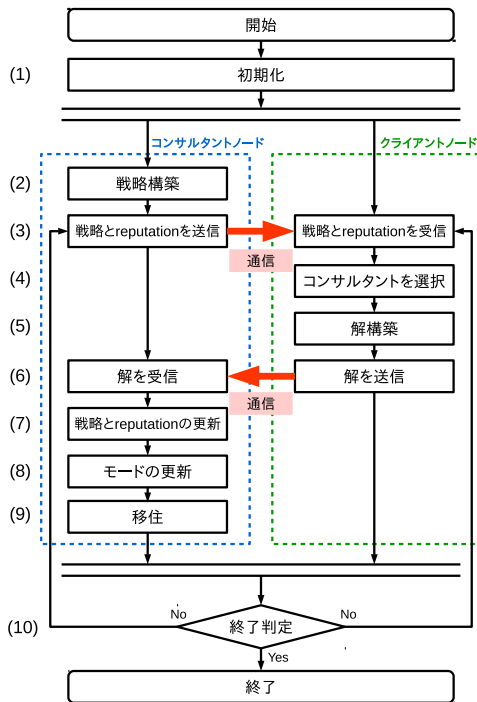


図 4 提案手法のフローチャート

Fig. 4 Flowchart of the proposed method.

に示す。

(1) 初期化

探索全体で使用する変数の初期化と仮想人間の作成を行う。

(2) 戦略の構築

コンサルタントノードは式 (2) に従って戦略を構築する。

(3) 戦略と reputation の送受信

各コンサルタントノードは戦略と reputation をすべてのクライアントノードへ送信する。クライアントノードは各コンサルタントノードの戦略と reputation を受信する。

(4) コンサルタント選択

クライアントノードは各コンサルタントノードの戦略と reputation からアドバイスを提供してもらうコンサルタントノードを式 (4) に従って選択する。

(5) 解の構築

クライアントノードは選択したコンサルタントノードの戦略をもとに式 (5) に従って解の構築を行う。

(6) 解の送受信

クライアントノードは構築した解を選択したコンサルタントノードへ送信する。コンサルタントノードは選択された各クライアントノードから解を受信する。

(7) 戦略と reputation の更新

クライアントノードから受信した解が現在の戦略より優れた解だった場合、その解をコンサルタントノードの戦略に設定し、成功回数を加算する。また、各コンサルタントノードの reputation の更新を式 (6)~(11) に従って行う。

(8) モードの更新

コンサルタントノードの reputation が最小値を下回っていた場合、現在の戦略を破棄して式 (2) に従い新しい戦略を構築する。そして reputation を初期値に設定する。

(9) 島移住

一定の反復回数ごとにコンサルタントノードが属する島を次に訪れる島に変更する。移住するコンサルタントはランダムに決定される。

(10) 終了判定

終了条件を満たす場合は全ノードの探索を終了する。終了条件を満たさない場合は (3) に戻る。

6. 評価実験

6.1 実験環境

本実験には PC クラスタコンソーシアム [14] の SCore を用いて MPI 環境を構築した並列システムを用いる。この並列システムにより 1 台の管理サーバと、10 台の計算ノードが同一 LAN 内で接続されており、互いに MPI 通信を行うことができる。図 5 に本実験で利用した並列システム、表 1 にその性能を示す。

6.2 実験方法

本実験に使用する TSP の問題例は、TSP のベンチマーク問題例を配布している TSPLIB [13] から入手した。実験には、それぞれ都市数が 575 である rat575, 783 である rat783, 1,002 である pr1002, 1,060 である u1060, 2,152 である u2152, 2,392 である pr2392, 3,038 である pcb3038, 5,915 である r15915 の 8 題を使用する。問題例の選択は、

表 1 管理サーバと計算ノードの構成

Table 1 Performance of server and calculation nodes.

	管理サーバ	計算ノード
CPU	Intel Xeon E5606 @2.13 GHz × 4	Intel Core i5-2400 @3.10 GHz × 4
Memory	8 GB	8 GB
OS	CentOS 5.10	CentOS 5.10
MPI	SCore 7.0.1	SCore 7.0.1
台数	1	10
コア数	4	40

比較のため著者らの論文 [8] で採用した問題例を本論文でも採用する。さらに、5,000 都市程度の問題例として、都市数 5,915 である rl5915 を無作為に追加で採用する。TSP の実装の都合でユークリッド型の問題例のみ採用する。また、各問題例ごとに解の収束性を考慮してそれぞれの探索時間を設定する。探索時間は、都市数が 2,000 未満の問題例は 2 時間、都市数が 2,000 以上で 3,000 未満の問題例は 3 時間、都市数が 3,000 以上で 5,000 未満の問題例は 5 時間、都市数が 5,000 以上の問題例は 12 時間とする。試行回数はすべて 10 回である。

提案手法の有効性を示すために 4 つの実験を行う。

1 つ目の実験では、コンサルタントとクライアントの人数比を分析するため、人数比を変えて実験を行う (実験 1)。この実験により、最適な人数比を求め、以降の実験ではその値を採用する。

2 つ目の実験では、提案手法の効果を確認するため、性能評価実験を行う (実験 2)。提案手法と比較するために、文献 [8] の提案手法である ACCGS と単純並列 CGS の結果を用いる。文献 [8] の計算機環境は、本論文のものと同一である。問題例 rl5915 に関しては、同一環境、同一パラメータ値で実験して得られた結果である。この 3 種類のアルゴリズムを用いて比較を行い、提案手法の並列化の有効性を測定する。

3 つ目の実験では、計算ノード数を評価するために、計算ノード数を 5 台に減らした場合の性能を評価する (実験 3)。仮想人間数を 20 とし、島数が 4 の場合と 2 の場合を実験する。

4 つ目の実験では、島モデルの効果と収束性を確認するため、提案手法、移住を実行しない提案手法、単純並列 CGS の 3 つのアルゴリズムを用いて比較実験を行う (実験 4)。移住を実行しない提案手法とは、提案手法から移住に関する処理を省いたものである。これらのアルゴリズムを用いて移住が探索にどれほど影響するのかを測定する。

### 6.3 実装方法

提案手法の実装方法は、コア単位で仮想人間を割り当てている関係で、島内の仮想人間数は 10 となり、クアッドコアの計算ノード 2.5 台 (2 台の計算機と CPU 内に 4 つあ

表 2 提案手法のパラメータ

Table 2 Parameters of the proposed method.

パラメータ項目名	値
$M$	40
$\beta$	5
$a_0$	0.95
$r_0$	$5.0 \times 10^{-7}$
$k_w$	5
$w$	750
<i>fading ranks</i>	2
<i>Init reputation</i>	25
<i>Max reputation</i>	250
<i>Min reputation</i>	5
<i>Bonus</i>	5

表 3 島モデルのパラメータ

Table 3 Parameters of island model.

パラメータ項目名	島 1, 3 での値	島 2, 4 での値
$b_0$	0.98	0.6
$q_0$	0.99	0.99
$\alpha$	3	7
$\gamma$	7	3

るコアのうち 2 つ) で 1 つの島を担当している。

本実験で比較対象としている文献 [8] の ACCGS と単純並列 CGS は、島モデルを採用せず、10 台の計算ノードでは各々逐次処理で実行される。仮想人間数は計算ノードごとに 20 から 30 の範囲でランダムに決められている。ACCGS では、フェロモン情報を計算ノード単位で保存し、計算ノード間でフェロモン情報の共有を行う。さらに、解情報も計算ノード間で共有する。単純並列 CGS では、計算ノード間でいっさい情報共有がなく、通信は発生しない。各計算ノードが導出した解の中で最も良い解を単純並列 CGS の解とする。

### 6.4 実験パラメータ

各実験で用いる提案手法のパラメータを表 2、表 3 に、ACCGS と単純並列 CGS のパラメータをそれぞれ表 4、表 5 に示す。

表 2 では、CGS のアルゴリズムに用いるパラメータ値を示している。 $M$  は仮想人間数を表す。1 人の仮想人間を 1 つのプロセッサコアに割り当てるため 40 人の仮想人間を用いる。 $\beta$  は、都市選択における擬似確率的方法の距離の重みを表す。 $a_0$  は、コンサルタントの戦略構築において最も近い都市を選択する確率を表す。 $r_0$ ,  $k_w$ ,  $w$ , *fading ranks* は、それぞれコンサルタントの *reputation* が減少する割合に影響するパラメータである。*Init reputation*, *Max reputation*, *Min reputation* は、それぞれコンサルタントの *reputation* の初期値、最大値、最小値を表す。*Bonus* は、最も良い解を求めたクライアントに対してアドバイスをし



表 4 ACCGS のパラメータ

Table 4 Parameters of ACCGS.

パラメータ項目名	値
$M$	20~30
$\alpha$	7
$\beta$	12
$\gamma$	7
$a_0$	0.9
$b_0$	0.98
$q_0$	0.98
$r_0$	$3.0 \times 10^{-7}$
$k_w$	3
$w$	1,000
<i>fading ranks</i>	3
<i>Init reputation</i>	50
<i>Max reputation</i>	100
<i>Min reputation</i>	1
<i>Bonus</i>	10
<i>lph</i>	0.10
<i>gph</i>	0.10

表 5 単純並列 CGS のパラメータ

Table 5 Parameters of simple parallel CGS.

パラメータ項目名	値
$M$	20~30
$\alpha$	7
$\beta$	12
$\gamma$	7
$a_0$	0.9
$b_0$	0.98
$q_0$	0.98
$r_0$	$3.0 \times 10^{-7}$
$k_w$	3
$w$	1,000
<i>fading ranks</i>	3
<i>Init reputation</i>	50
<i>Max reputation</i>	100
<i>Min reputation</i>	1
<i>Bonus</i>	10

たコンサルタントの *reputation* に加算する値を表す。パラメータ値は、文献 [8] で採用したパラメータ値を参考に、予備実験において 250~2,000 程度の都市数の問題例を中心に数例用いて良い結果が得られた値に決定している。

提案手法では、4つの環境が異なる島を用いて探索を行う。移住を行うのは、コンサルタントを演じる仮想人間のみである。クライアントが解を 500 回構築するごとに 1 人のコンサルタントが異なる島へ移住する。

表 3 では、提案手法における各島のクライアントのパラメータ値を示している。 $b_0$  は、クライアントの解構築における最も近い都市を選択する確率を表す。 $q_0$  は、コンサルタントのアドバイスに従う確率を表す。 $\alpha$  は、コンサルタントを選択する確率を計算する際に利用するコンサルタン

表 6 人数比の特性

Table 6 Characteristics of virtual person ratio.

問題	人数比	平均値	平均誤差率 [%]	最小値	最小誤差率 [%]
rat783	4 : 36	<b>8,984</b>	<b>2.02</b>	8,963	1.78
	8 : 32	8,985	2.04	<b>8,933</b>	<b>1.44</b>
	20 : 20	9,020	2.43	8,966	1.82
	32 : 8	9,521	8.12	9,445	7.26
u1060	4 : 36	228,881	2.14	<b>227,217</b>	<b>1.39</b>
	8 : 32	<b>228,542</b>	<b>1.98</b>	228,043	1.76
	20 : 20	235,308	5.00	233,739	4.30
	32 : 8	251,701	12.32	248,794	11.02
pr2392	4 : 36	392,013	3.70	<b>388,897</b>	<b>2.87</b>
	8 : 32	<b>391,807</b>	<b>3.64</b>	390,522	3.30
	20 : 20	419,303	10.92	412,732	9.18
	32 : 8	436,019	15.34	429,581	13.64

トの *reputation* の重みを示す。 $\gamma$  は、コンサルタントを選択する確率を計算する際に利用する戦略の経路距離の逆数の重みを示す。鳥モデルのパラメータ値は予備実験で主要と思われるパラメータを変更し、比較的良かったものの中から 2 種類選択している。

表 4 では、ACCGS におけるパラメータ値を示している。*lph* は、局所的なフェロモン情報の更新におけるフェロモンの上昇率を示す。*gph* は、全体のフェロモン情報の更新におけるフェロモンの上昇率を示す。パラメータ値は、文献 [8] と同じものである。

表 5 では、単純並列 CGS におけるパラメータ値を示している。パラメータ値は、文献 [8] と同じものである。

### 6.5 評価方法

実験の評価には誤差率を用いる。誤差率は以下の式によって与えられ、TSPLIB [13] で公開されている既知の最適解と実験結果から導くことができる。

$$Error\ rate = \frac{Solution - Optimal\ solution}{Optimal\ solution} \quad (13)$$

ここで、*Error rate* は誤差率、*Solution* は実験結果、*Optimal solution* は最適解を表す。

### 6.6 実験 1 の実験結果

rat783, u1060, pr2392 の問題例に対して、コンサルタントとクライアントの人数比を変えて実験を行った。その結果を表 6 に示す。

表 6 からすべての問題例において、コンサルタント数をクライアント数と同等あるいは多くすると誤差率が悪くなることが分かった。このことから、戦略を多様にするよりもクライアントを増やし、より多くの探索を行う方がよいことが分かる。さらに、コンサルタントとクライアントの人数比を 8 : 32 と 4 : 36 にした場合で比較すると、最小値においては人数比 4 : 36 が良い誤差率を示すこともある

表 7 実験結果

Table 7 Results of experiment.

問題	アルゴリズム	平均値	平均誤差率 [%]	最小値	最小誤差率 [%]
rat575	提案手法	6,884	1.64	6,865	1.36
	単純並列 CGS [8]	6,861	1.30	6,850	1.14
	ACCGS [8]	<b>6,853</b>	<b>1.18</b>	<b>6,828</b>	<b>0.82</b>
rat783	提案手法	8,985	2.04	8,933	1.44
	単純並列 CGS [8]	8,995	2.14	8,949	1.62
	ACCGS [8]	<b>8,920</b>	<b>1.30</b>	<b>8,883</b>	<b>0.87</b>
pr1002	提案手法	263,934	1.88	262,976	1.52
	単純並列 CGS [8]	264,797	2.22	264,132	1.96
	ACCGS [8]	<b>261,909</b>	<b>1.11</b>	<b>260,981</b>	<b>0.75</b>
u1060	提案手法	228,542	1.98	228,043	1.76
	単純並列 CGS [8]	229,125	2.25	227,799	1.65
	ACCGS [8]	<b>228,114</b>	<b>1.79</b>	<b>227,334</b>	<b>1.45</b>
u2152	提案手法	66,686	3.78	66,472	3.45
	単純並列 CGS [8]	67,447	4.97	66,762	3.90
	ACCGS [8]	<b>65,622</b>	<b>2.22</b>	<b>65,381</b>	<b>1.88</b>
pr2392	提案手法	391,807	3.64	390,522	3.30
	単純並列 CGS [8]	405,769	7.34	402,507	6.47
	ACCGS [8]	<b>386,073</b>	<b>2.13</b>	<b>384,499</b>	<b>1.67</b>
pcb3038	提案手法	<b>143,707</b>	<b>4.36</b>	<b>143,265</b>	<b>4.05</b>
	単純並列 CGS [8]	158,333	14.99	157,919	14.69
	ACCGS [8]	156,662	13.78	155,358	12.83
rl5915	提案手法	<b>589,230</b>	<b>4.19</b>	<b>586,607</b>	<b>3.73</b>
	単純並列 CGS	618,836	9.42	616,764	9.06
	ACCGS	619,172	9.49	614,895	8.73

が、平均値ではおおむね人数比 8 : 32 が優れている\*1。人数比 4 : 36 の場合戦略が少なく良い戦略に巡り会った場合は良い誤差率を示す場合もあるが、バラツキが激しく不安定であると考えられる。以上の結果から、コンサルタントとクライアントの最適な人数比は 8 : 32 と評価し、以降の実験ではその値を採用する。

### 6.7 実験 2 の実験結果

提案手法の効果を確認するため、文献 [8] の提案手法である ACCGS と単純並列 CGS の結果と比較する。都市数 575 から 5,915 の 8 題の問題例に適用して得られた結果を表 7 に示す。

表 7 の実験結果から提案手法と単純並列 CGS を比較すると、都市数の大きい問題例であるほど提案手法の方が単純並列 CGS よりも優れていることが分かる。特に都市数が 3,000 以上の問題例では結果の差が顕著に現れている。この差は並列化モデルの違いに起因していると考えられる。単純並列 CGS ではプロセッサコアごとに逐次処理の CGS が実行されるが、提案手法では 1 つのプロセッサコアに 1 人の仮想人間を割り当てて探索を行う。また、CGS の仮想人間はコンサルタントとクライアントを同時に演じ

\*1 rat783 の場合は平均値において 0.02% だけ 4 : 36 が優れている。

表 8 計算ノード数 5 台の実験結果

Table 8 Experimental results in 5 computing nodes.

問題	計算ノード数	鳥数	平均値	平均誤差率 [%]	最小値	最小誤差率 [%]
rat783	10	4	<b>8,985</b>	<b>2.04</b>	<b>8,933</b>	<b>1.44</b>
	5	4	8,998	2.18	8,955	1.69
	5	2	9,025	2.49	8,988	2.07
u1060	10	4	<b>228,542</b>	<b>1.98</b>	<b>228,043</b>	<b>1.76</b>
	5	4	230,060	2.66	228,257	1.86
	5	2	229,902	2.59	229,060	2.22
pr2392	10	4	<b>391,807</b>	<b>3.64</b>	390,522	3.30
	5	4	393,241	4.02	<b>390,086</b>	<b>3.19</b>
	5	2	393,605	4.12	391,984	3.69

るが、提案手法ではコンサルタントの役割を与えられた仮想人間はコンサルタントのみを演じ、クライアントの役割を与えられた仮想人間はクライアントのみを演じ続ける。そのため CGS では不可能だったコンサルタントとクライアントの人数調整を提案手法では可能にしている。実験 1 の結果から分かるように、クライアントの人数をコンサルタントの人数より多めにすることで解の探索の役割を持つクライアントに比重を置いている。これは同時にクライアントに割り当てるプロセッサコア数をコンサルタントに割り当てるプロセッサコア数よりも多くすることを表している。これらの違いが都市数の多い問題例で探索性能の差が大きく現れた要因であると考えられる。

次に、提案手法と ACCGS の結果を比較すると、pcb3038 と rl5915 では提案手法の方が優れ、それより小さい問題例では ACCGS の方が優れていることが分かる。ACCGS は CGS と ACS をうまく組み合わせることで単純並列 CGS の結果を大きく改善しているが、pcb3038 と rl5915 ではあまり改善されていない。一方、提案手法は並列化手法を工夫することで都市数の多い問題例に対する探索性能を大幅に改善している。このことから都市数の多い問題では、ハイブリッド型 CGS よりも並列化手法を工夫する方がより探索性能を向上させることができると考えられる。

提案手法は 3,000 都市以上の問題例では優れた結果が得られたが、1,000 都市以下の小さな問題例では他の手法より優れているとはいえない。著者らはこの理由について、鳥モデルは解の多様性の向上に優れているが、小さな問題例では多様性が大きすぎて解が収束しないのではないかと推測している。詳細は不明のため今後の課題としたい。

### 6.8 実験 3 の実験結果

rat783, u1060, pr2392 の問題例に対して、計算ノード数を 5 台に減らして実験を行った。その結果を表 8 に示す。この実験では計算ノード数 5 台の場合の仮想人間数は 20 人とする。コンサルタントとクライアントの人数比は実験 1 の結果から 8 : 32 とし、コンサルタント数 4、クライ

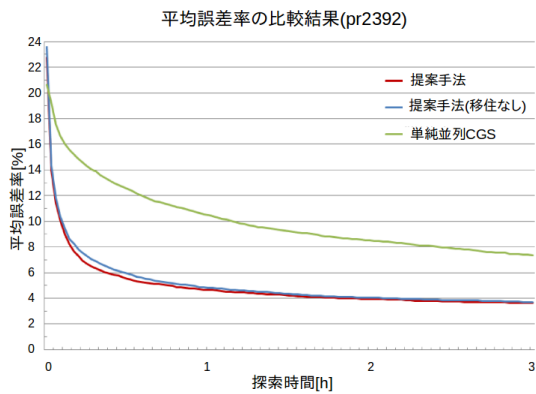


図 6 比較結果 (pr2392)  
Fig. 6 Experimental result of pr2392.

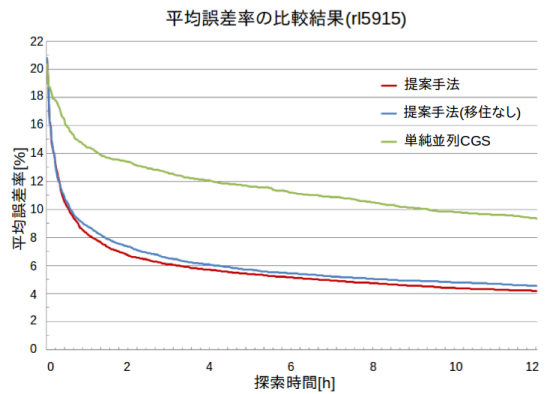


図 8 比較結果 (rl5915)  
Fig. 8 Experimental result of rl5915.

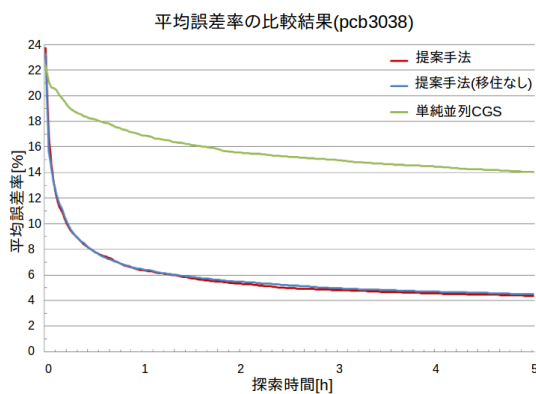


図 7 比較結果 (pcb3038)  
Fig. 7 Experimental result of pcb3038.

手法の探索において都市数が多いほど移住による効果を受けていることが分かる。したがって、移住は比較的規模の大きい問題に対して効果的だと考えられる。

## 7. まとめ

本研究では、CGS の仮想人間をプロセッサコアに割り当て、島モデルを取り入れた並列化手法を提案した。提案した並列化手法は多くの TSP 問題に対して良い解を導出し、特に都市数の多い問題に対して高い探索性能を発揮し、3,038 都市と 5,915 都市の問題例では既存のハイブリッド型 CGS を大幅に改善した。また、島モデルは都市数が多くなると有効であることを示した。

今後の課題として、CGS に他のメタヒューリスティクスを取り入れることがあげられる。たとえば、文献 [8] と同様にフェロモン情報を CGS におけるコンサルタントの戦略に引き継ぐことや、遺伝的アルゴリズムで実行される交差と突然変異を CGS におけるコンサルタントの戦略に適用することが考えられる。また、提案手法が小さな問題例で良い結果を得られない理由を解明し、小さな問題例でも良い結果が得られるように提案手法を改善したい。

謝辞 本研究の一部は、関西大学先端科学技術推進機構「非常時緊急救命避難支援のための情報通信技術開発」研究グループ予算によるものである。

## 参考文献

- [1] El-Ghazali, T.: *Metaheuristics: From Design to Implementation*, Wiley Publishing (2009).
- [2] Dorigo, M.: Ant colony optimization: A new metaheuristic, *Proc. Congress on Evolutionary Computation*, pp.1470-1477 (1999).
- [3] Ruciski, M., Izzo, D. and Biscani, F.: On the impact of the migration topology on the island model, *Parallel Computing*, Vol.36, No.1011, pp.555-571 (2010). *Parallel Architectures and Bioinspired Algorithms*.
- [4] Iordache, S.: Consultant-guided search - A new metaheuristic for combinatorial optimization problems, *GECCO 2010: Proc. 12th Genetic and Evolutionary Computation Conference*, pp.225-232 (2010).

アント数 16 とする。

表 8 から分かるように、10 台で仮想人間数 40 人の場合に比べ、5 台で仮想人間数 20 人の場合は悪い結果となっている。計算パワーが半減しているため、結果が悪くなるのは当然だが、コンサルタント数が少ないため解のバラツキも激しい。

## 6.9 実験 4 の実験結果

提案手法と移住を実行しない提案手法を pr2392, pcb3038, rl5915 に適用して得られた結果から作成したグラフをそれぞれ図 6, 図 7, 図 8 に示す。参考として、単純並列 CGS の計算結果も載せている。この実験では、移住による探索への影響と解の収束速度に関する比較を行うため都市数の大きい 2,000 都市以上の問題例に各アルゴリズムを適用する。

提案手法と移住を実行しない提案手法の移住による効果を比較すると pr2392, pcb3038 ではほとんど差がみられないが、rl5915 では差がわずかにみられる。それぞれの探索終了時の平均誤差率を比較すると、移住を行う方が pr2392 では 0.04% 優れ、pcb3038 では 0.12% 優れ、rl5915 では 0.35% 優れる結果となった。それぞれの結果から提案

- [5] Iordache, S.: Consultant-guided search algorithms with local search for the traveling salesman problem, *Parallel Problem Solving from Nature, PPSN XI*, Vol.6239, pp.81-90 (2010).
- [6] Dorigo, M. and Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evolutionary Computation*, pp.53-66 (1997).
- [7] Stützle, T. and Hoos, H.H.: Max-min ant system, *Future Gener. Comput. Syst.*, Vol.16, No.9, pp.889-914 (2000).
- [8] Ebara, H., Hiranuma, Y. and Nakayama, K.: Hybrid consultant-guided search for the traveling salesperson problem, *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, Vol.97, No.8, pp.1728-1738 (2014).
- [9] Deepanandhini, D. and Amudha, T.: Solving job shop scheduling problems with consultant guided search metaheuristics, *International Journal of Software and Web Sciences (IJSWS)*, Vol.1, pp.1-6 (2013).
- [10] Iordache, S.: Consultant-guided search algorithms for the quadratic assignment problem, *Hybrid Metaheuristics*, Vol.6373, pp.148-159 (2010).
- [11] Jovanovic, R., Tuba, M. and Simian, D.: Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem, *W. Trans. on Comp.*, Vol.9, No.1, pp.83-92 (2010).
- [12] Arnaldo, I., Contreras, I., Milln-Ruiz, D., Hidalgo, J.I. and Krasnogor, N.: Matching island topologies to problem structure in parallel evolutionary algorithms, *Soft Computing*, Vol.17, No.7, pp.1209-1225 (2013).
- [13] TSPLIB, available from (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>) (accessed 2015-08-10).
- [14] PC Cluster Consortium, available from (<http://www.pcluster.org/>) (accessed 2015-08-10).



飯田 修平

1991年生。2014年関西大学システム理工学部電気電子情報工学科卒業。同年より同大学大学院理工学研究科博士課程前期課程システムデザイン専攻在籍。機械学習を用いた巡回セールスマン問題の解法の研究に従事。



長辻 亮太

1993年生。2015年関西大学システム理工学部電気電子情報工学科卒業。同年より同大学大学院理工学研究科博士課程前期課程システムデザイン専攻在籍。コンサルタント誘導型探索を用いた巡回セールスマン問題の解法の研究

に従事。



榎原 博之 (正会員)

1982年大阪大学工学部通信工学科卒業。1987年同大学大学院博士(通信)課程修了。同年大阪大学工学部助手。1994年関西大学工学部専任講師となり、現在、准教授。組合せ最適化問題、計算幾何学、並列アルゴリズム等の研究に従事。工学博士。電子情報通信学会、IEEE、ACM各会員。

研究に従事。工学博士。電子情報通信学会、IEEE、ACM各会員。



中山 弘基

1990年生。2013年関西大学システム理工学部電気電子情報工学科卒業。2015年同大学大学院理工学研究科博士課程前期課程システムデザイン専攻修了。在学中はコンサルタント誘導型探索を用いた巡回セールスマン問題の

解法の研究に従事。現在は株式会社日立システムズに勤務。