

粒子群最適化による適応パラメータチューニングの効果

長辻 亮太^{1,a)} 飯田 修平¹ 榎原 博之^{2,b)}

受付日 2015年9月3日, 再受付日 2015年10月19日,
採録日 2015年11月16日

概要: 組合せ最適化問題などの複雑な問題に対して, メタヒューリスティクス手法は有効であり, その性能は近年飛躍的に向上している. その中でも粒子群最適化 (Particle Swarm Optimization: PSO) を用いた群知能 (Swarm Intelligence: SI) の適応パラメータチューニングは強力な最適化能力を示す. 本研究では, 複数の SI アルゴリズムに PSO を用いた適応パラメータチューニングを施し, それらを比較することで PSO 適応パラメータチューニングによって強化された SI の最適化能力を評価する. さらに, 重要なパラメータ項目の考察, ならびに固定パラメータと PSO 適応パラメータチューニングを比較し, PSO 適応パラメータチューニングの効果を分析する.

キーワード: コンサルタント誘導型探索, 巡回セールスマン問題, 粒子群最適化, 蟻コロニー最適化, パラメータチューニング

Effects of Adaptive Parameter Tuning by Particle Swarm Optimization

RYOUTA NAGATSUJI^{1,a)} SYUUHEI IIDA¹ HIROYUKI EBARA^{2,b)}

Received: September 3, 2015, Revised: October 19, 2015,
Accepted: November 16, 2015

Abstract: Meta-heuristics is available for the combinatorial optimization problem and its ability has been developed. The Swarm Intelligence (SI) with the adaptive parameter tuning by the Particle Swarm Optimization algorithm shows powerful results. In this study, some SI algorithms apply the adaptive parameter tuning by the PSO algorithm. We perform simulation experiments, and evaluate the ability of SI enhanced by the PSO. In addition, we consider the important parameters for some instances, and analyze its effectiveness, comparing the enhanced SI with the SI algorithm set the static parameters.

Keywords: Consultant Guided Search (CGS), Traveling Salesman Problem (TSP), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), parameter tuning

1. はじめに

現代社会において金融問題や空力デザインなど様々な分野で最適化問題が重要になってきており, 大規模な最適化

問題を計算機で解く需要はますます高まっている. 計算機で最適化問題を解くには, 問題の定式化は不可欠である. 制約条件 $x \in A$ と目的関数 $f(x)$ が与えられ $f(x)$ を最小・最大にできるような x を求める問題を定式化できる. 特に x が離散値の場合, 組合せ最適化問題と呼ばれ, ナップサック問題, 2 次割当て問題などが含まれる. 組合せ最適化問題の解法は 2 通りに大別され, 最適解を厳密に求める厳密解法と厳密解でないことを許し近似的に解を求める近似解法が含まれる. 組合せ最適化問題に含まれる問題には巡回セールスマン問題や充足可能性問題などの NP 困難な問題が含まれている. NP 困難な問題は問題例のサイズの多項式時間で厳密解を見つけるアルゴリズムを考案することが

¹ 関西大学大学院理工学研究科システム理工学専攻電気電子情報工学分野アルゴリズム工学研究室

Graduate School of Science and Engineering, Systems Science Department, Electrical and Electronic Information Engineering, Kansai University, Algorithm Engineering Lab, Suita, Osaka 564-8680, Japan

² 関西大学システム理工学部電気電子情報工学科

Department of Electrical and Electronic Engineering, Faculty of Engineering Science, Kansai University, Suita, Osaka 564-8680, Japan

^{a)} ryoutafunphy@gmail.com

^{b)} ebara@kansai-u.ac.jp

難しい。NP 困難な問題に対するアプローチとして、現実的な計算時間・計算資源である程度良い解を求めることができる近似解法が多く用いられる。近似解法には様々な戦略を組み合わせ高精度な解を求め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を用いて局所探索を反復的に行い、終了条件を満たすまで計算を実行する。代表的なメタヒューリスティクスには蟻コロニー最適化 (Ant Colony Optimization: ACO) [1] や遺伝的アルゴリズム (Genetic Algorithm: GA) [2] などが含まれる。メタヒューリスティクスアルゴリズムの改善は文献 [3], [4], [5], [6] のような戦略構築手法の改善だけでなく、パラメータの設定手法も重要で数多く提案されている。パラメータの設定には実行中に動的に変更を加えるものの [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] がある。

PSO は昆虫や鳥の群れが良さそうな場所を見つけると、そこに群がる性質をモデルとしたメタヒューリスティクスである。このアルゴリズムは、粒子 (Particle) と呼ばれる個体の集合を用いて解を探索する。一般的に、Particle は問題の要素と同じ次元のユークリッド空間を飛翔し、解を探索し、移動速度に比例した抵抗力により局所解に収束する。

本研究では、蟻コロニー最適化 (Ant Colony Optimization: ACO) [1] と粒子群最適化 (Particle Swarm Optimization: PSO) を用いた適応パラメータ設定手法 [15] を参考に、その他の SI アルゴリズムについて同様な手法で適応パラメータチューニングを行うことで、PSO を用いた適応パラメータチューニングによって強化された SI アルゴリズムの性能を分析する。文献 [15] の手法では、各個体ごとに独立して存在する ACO パラメータを PSO の解空間に割り当て、PSO が慣性定数 1、すなわち抵抗力 0 の ACO パラメータ空間を粒子のエネルギーが減衰することなく振動的に探索する。この手法は簡単な差分方程式によって実現されるので、任意の次元への拡張が可能である。

本研究では、ACO, ACS, CGS, の 3 つの解法を扱う。戦略構築時にそれぞれ 2, 3, 6 個のパラメータ値が主に使用される。それらすべてのパラメータを PSO の適応パラメータチューニングの対象とするために 2, 3, 6 次元の高次元パラメータ空間を採用する。計算機実験を行うことで、PSO 適応パラメータチューニングによって強化された SI アルゴリズムと本来のアルゴリズムの性能を比較する。また、戦略構築時に参照されるパラメータのうち、PSO 適応パラメータチューニングを行い、性能向上に寄与するパラメータ項目について考察を行う。

2. 巡回セールスマン問題とそのアルゴリズム

2.1 巡回セールスマン問題

巡回セールスマン問題 (Traveling Salesman Problem: TSP) は都市と呼ばれる頂点の集合と三角不等式を満足する都市間の移動コストがグラフとして与えられたときに、すべての都市を必ず一巡し、開始地点である都市に戻ってくる巡回路中で総移動コストが最小となるものを探索する問題である。この問題の解 x は都市の集合 V と移動コスト $C(v_1, v_2)$ を用いて式 (1), (2) に定式化できる。

$$A(x) = \{i, j \in [0, n-1] \wedge i \neq j \wedge x_i \neq x_j \wedge x_i, x_j \in V \wedge |V| > 0\} \quad (1)$$

$$\min_x f(x) = \min_x \sum_{i \in [0, n-1]} C(x_i, x_{(i+1) \bmod n}) \quad (2)$$

また、この問題の問題例は TSPLIB [23] より配布されており、そこから問題例の入手を行う。

この章で述べるアルゴリズムは現在訪問中の都市の情報を用いて、候補の都市集合の中から次の都市を選択する方針をとる。ゆえに、候補の集合から選択することで解の構築が可能な問題に対しては単純な応用が可能と考えられる。

2.2 コンサルタント誘導型探索 (CGS)

コンサルタント誘導型探索 (Consultant Guided Search: CGS) [24] は人間が何らかの意思決定を行う際にコンサルタントに助言を求める行動をモデルとしたメタヒューリスティクスである。このアルゴリズムは仮想人間 (Virtual Person) と呼ばれる個体の集合を用いて、解を探索する。初期状態において仮想人間は *Mode* が *Sabbatical* に設定されており、最初は単独で解を探索する。一定回数を重ねると *Normal* に遷移し、コンサルタントおよびクライアントと呼ばれる役割を担う。*Mode* が *Sabbatical* である場合、式 (3) より、他の仮想人間に助言を求めずに最近傍を基とした戦略構築法を用いる。

$$v_n = \begin{cases} \arg \min_{i \in V_R} C(v_c, i) & (\text{random}() \leq a_0) \\ \arg \text{prob} C(v_c, i)^{-\beta_{cgs}} & (\text{otherwise}) \end{cases} \quad (3)$$

Normal である場合、*Mode* が *Normal* である他の仮想人間を評判 *Rep* と評価値の逆数で与えられる *Pre* を基にコンサルタント c を式 (4) で選び、クライアントの立場で助言を求め解を式 (5) で構築する。

$$c = \arg \text{prob}_{i \in \{p | p \in P, \text{Mode}(p) = \text{Normal}\}} \text{Rep}(i)^{\alpha_{cgs}} \text{Pre}(i)^{\gamma} \quad (4)$$

$$v_n = \begin{cases} advice(c, v_c, V_R) & (\exists c \wedge random() \leq q_{0,cgs}) \\ \arg \min_{i \in V_R} C(v_c, i) & (random() \leq b_0) \\ \arg \text{prob} C(v_c, i)^{-\beta_{cgs}} & (\text{otherwise}) \end{cases} \quad (5)$$

コンサルタント c がクライアントに対して都市を助言する方法はコンサルタント c が持つ解から現在訪問中の都市を見つけ出し、前後の都市の中から近い方を返す。両方の都市が訪問済みであれば、未訪問都市の中から現在の都市からの距離を用いて確率的に選択する。仮想人間の *Mode* は CGS 実行中に交互に遷移する。*Mode* が *Sabbatical* であるときは一定回数 C_S で *Normal* へ、*Normal* であるときは *Rep* が閾値 R_T を下回るまで継続され、下回った際に *Sabbatical* へ遷移する。*Normal* において、*Rep* はイテレーションごとに減衰し、最良解または助言の成功により加算される。*Rep* の減衰率は *FadingRate* と呼ばれ、式 (6), (7), (8) で与えられる。*Rank* は *Mode* が *Normal* である仮想人間が保持している解の総距離が短い順に仮想人間を並べたときに得られる順位である。

$$FadingRate = r_0 \left(1 + \frac{s}{\sqrt{1 + (\frac{s}{f})^2}} \right) \quad (6)$$

$$s = \sum_{i \in \{x | x \in P, Mode(x) = Normal, Rank(x) \leq R_F\}} success(i, w) \quad (7)$$

$$f = \left(\frac{1}{r_0} - 1 \right) \left(1 - \frac{1}{w \sqrt{k_w}} \right) \quad (8)$$

2.3 蟻コロニー最適化 (ACO)

蟻コロニー (Ant Colony Optimization: ACO) [1] は蟻の採餌行動をモデルとしたメタヒューリスティクスである。人工蟻 (Artificial Ant) と呼ばれる個体の集合を用いて、解の探索を行う。TSP において、次の都市の選択確率を式 (9) でフェロモン τ と呼ばれる変数で計算する。この値は解を構築した蟻によって強化され、無条件で揮発し、弱化する。

$$v_n = \arg \text{prob} \tau(v_c, i)^{\alpha_{aco}} C(v_c, i)^{-\beta_{aco}} \quad (9)$$

ACO のエージェントはフェロモンをグラフ上に散布し、ACO アルゴリズムはグラフ上のすべてのフェロモンを弱化する。

2.4 蟻コロニーシステム (ACS)

蟻コロニーシステム (Ant Colony System: ACS) [25] は ACO アルゴリズムを改良したものであり、解構築方法とフェロモンの散布方法が異なる。構築方法の異なる点は一定の確率で重み付けられたフェロモン強度と都市間の近さ

の積が最大になるような都市を選択する戦略を式 (10) を用いて確率的に適用する。フェロモンの散布は最良解のエージェントのみが行う。

$$v_n = \begin{cases} \arg \max_{i \in V_R} \tau(v_c, i)^{\alpha_{acs}} C(v_c, i)^{-\beta_{acs}} & (random() \leq q_{0,acs}) \\ \arg \text{prob} \tau(v_c, i)^{\alpha_{acs}} C(v_c, i)^{-\beta_{acs}} & (\text{otherwise}) \end{cases} \quad (10)$$

3. 提案手法

Mostafa Mahi らの手法 [15] は ACO の各個体に保持されているパラメータである $param(i, \alpha_{aco}, \beta_{aco})$ を PSO の個体の位置 $z_i = param(i, \alpha_{aco}, \beta_{aco})$ として割り当て、ACO の個体が持つ解の評価値を用いて PSO の $z_{i,best}, z_{best}$ として ACO の個体のパラメータを選ぶ。ACO の個体が評価される際に、その個体が保有しているベストな解を更新した場合、そのパラメータをその個体に保持させ、それを $z_{i,best}$ とする。また、システム上の全体ベストを更新した個体が保持しているパラメータを ACO が保持し、それを z_{best} とする。これらの個体のベストとシステム上のベストな解に対応するパラメータを用いて、PSO により個体のパラメータを用いて更新する。そして、最良解に 3opt 法を適用する。この方法は ACO-PSO-3opt と呼ばれ、PSO の適応パラメータ設定を行いながら ACO で解を探索し、最後に 3opt で近傍探索を行う。この手法は高い性能を示したものの、最後の 3opt による効果が大きいので、ハイブリッド手法 ACO-PSO の性能が不明である。

図 1 は一般的な SI アルゴリズムのフローに PSO による適応パラメータチューニングを追加したものを言語統合クエリ [26] の定義に基づいて記述したものである。戦略および解を構築し、評価を行い、パラメータを更新し、これを実行条件を満たす間実行し続ける。パラメータの保持は評価の際に行われ、個体 i が自身が持つ解を上回る解を構築した場合 $z_{i,best}$ として保持され、最良解を上回る解を算出した場合は z_{best} としてそれぞれ個体のパラメータを PSO の最良解として用いる。PSO でパラメータを更新する手法は図 1 に示すようなタイミングで PSO を 1 イテレーション実行する。これは式 (11) に $z_i = param(i)$ を代入し、1 回だけ計算を行うことを表す。また、式 (11) はベクトルの 2 階差分方程式であり、任意の次元への拡張が可能である。

$$\Delta^2 z_i = c_1 r_1 (z_{i,best} - z_i) + c_2 r_2 (z_{best} - z_i) \quad (11)$$

本研究では、この手法を CGS と ACS に適用する。ただし、CGS と ACS には多くのパラメータが存在し、その中でも戦略構築に関与しているパラメータについて適応パラメータチューニングを行う。この提案手法を CGS-PSO

- (1) 個体群生成
 $initialize(S)$
- (2) 戦略および解の構築
 $foreach(S, i \rightarrow buildStrategy(i))$
 $foreach(S, i \rightarrow setSolution(strategy(i)))$
- (3) 個体の最良パラメータの更新
 $foreach(where(S, i \rightarrow updatedSolution(i)), i \rightarrow$
 $z_{i,best} \leftarrow param(i))$
- (4) 最良解および最良パラメータの更新
 $a \leftarrow min(S, i \rightarrow f(solution(i)))$
 $if f(solution(a)) < f(solution(a_{best}))$
 $a_{best} \leftarrow a$
 $z_{best} \leftarrow param(a_{best})$
- (5) PSO 適応パラメータチューニング
 $foreach(select(S, i \rightarrow param(i)), z_i \rightarrow z_i \leftarrow$
 $nextState(\Delta^2 z_i = c_1 r_1(z_{i,best} - z_i) + c_2 r_2(z_{best} - z_i)))$
- (6) 終了条件
終了するならば手続き (7) へ、そうでなければ (2) へジャンプする。
- (7) 終了

図 1 PSO 適応パラメータチューニング手法

Fig. 1 The method of adaptive parameter tuning by using PSO.

表 1 CGS(-PSO) パラメータ項目の説明

Table 1 The description of CGS(-PSO) parameters.

項目	説明
R_T	<i>Sabbatical</i> に遷移する <i>Rep</i> の閾値
R_{max}	<i>Rep</i> の最大値
R_B	コンサルタント活動成功時に加算される <i>Rep</i>
k_w	<i>FadingRate</i> を算出するために必要な定数
r_0	<i>Rep</i> の減衰率
α_{cgs}	コンサルタント選択確率の <i>Rep</i> の重み
γ	コンサルタント選択確率の <i>Pre</i> の重み
$q_{0,cgs}$	コンサルタント活用率
b_0	<i>Normal</i> 時の最近傍利用率
β_{cgs}	確率の最近傍を制御する変数
a_0	<i>Sabbatical</i> 時の最近傍利用率
C_s	<i>Sabbatical</i> 期間の長さ
R_d	<i>Rep</i> の初期値
R_F	<i>FadingRank</i>
w	<i>FadingRate</i> を算出するために必要な定数
$c_{1,cgs}$	PSO 弾性係数 (各個体最良解)
$c_{2,cgs}$	PSO 弾性係数 (最良解)
K_{min}	パラメータ初期値の最小倍率
K_{max}	パラメータ初期値の最大倍率
N_{CGS}	個体数

と ACS-PSO と呼ぶことにする。式 (12), (13) に CGS と ACS の PSO の適応パラメータチューニング対象を示す。

$$param_{CGS}(i, a_0, \beta_{cgs}, \alpha_{cgs}, \gamma, q_{0,cgs}, b_0) \quad (12)$$

$$param_{ACS}(i, \alpha_{acs}, \beta_{acs}, q_{0,acs}) \quad (13)$$

表 2 ACO(-PSO) パラメータ項目の説明

Table 2 The description of ACO(-PSO) parameters.

項目	説明
α_{aco}	フェロモン依存度
β_{aco}	近傍依存度
Q_{aco}	散布フェロモン係数
ρ_{aco}	フェロモン減衰率
$\tau_{max,aco}$	フェロモンの最大値
$\tau_{min,aco}$	フェロモンの最小値
$\tau_{init,aco}$	フェロモンの初期値
$c_{1,aco}$	PSO 弾性係数 (各個体最良解)
$c_{2,aco}$	PSO 弾性係数 (最良解)
$\alpha_{max,aco}$	α パラメータ初期値の最小値
$\alpha_{min,aco}$	α パラメータ初期値の最大値
$\beta_{max,aco}$	β パラメータ初期値の最小値
$\beta_{min,aco}$	β パラメータ初期値の最大値
N_{ACO}	ACO アルゴリズムの個体数
N_{ACOPSO}	ACO-PSO アルゴリズムの個体数

表 3 ACS(-PSO) パラメータ

Table 3 The description of ACS(-PSO) parameters.

項目	説明
α_{acs}	フェロモン依存度
β_{acs}	近傍依存度
Q_{acs}	散布フェロモン係数
$q_{0,acs}$	最大択確率戦略利用率
ρ_{acs}	フェロモン減衰率
$\tau_{max,acs}$	フェロモンの最大値
$\tau_{min,acs}$	フェロモンの最小値
$\tau_{init,acs}$	フェロモンの初期値
$c_{1,acs}$	PSO 弾性係数 (各個体最良解)
$c_{2,acs}$	PSO 弾性係数 (最良解)
$\alpha_{max,acs}$	α パラメータ初期値の最小値
$\alpha_{min,acs}$	α パラメータ初期値の最大値
$\beta_{max,acs}$	β パラメータ初期値の最小値
$\beta_{min,acs}$	β パラメータ初期値の最大値
$q_{0,min}$	q_0 パラメータ初期値の最小値
$q_{0,max}$	q_0 パラメータ初期値の最大値
N_{ACS}	ACS アルゴリズムの個体数
N_{ACSPSO}	ACS-PSO アルゴリズムの個体数

これらのパラメータはすべて戦略構築に関与しており、各個体が保有しており、次元はそれぞれ 6, 3 次元である。各パラメータを要素として持つベクトルを拡張された PSO の個体に対応付け、適応パラメータチューニングを行う。計算機実験を実施することで、PSO 適応パラメータチューニングによって強化された SI アルゴリズムの性能を評価することができる。

各アルゴリズムで使用されるパラメータ項目についての解説を表 1, 表 2, 表 3 に示す。また、最も戦略構築パラメータが多い CGS について、重要なパラメータ項目を考察することで、戦略構築方法に類似点のある蟻コロニーのアルゴリズムにも応用が効くと考えられる。

4. 計算機実験

本研究では、Mostafa Mahi らのパラメータ適応手法 [15] を CGS と ACS に適用する．この手法は 3 章に記述されており，適応パラメータチューニングの対象である SI アルゴリズムのパラメータを実行中に書き換えるものである．本章では実際に CGS-PSO の性能と PSO チューニング対象の検証や，CGS-PSO や ACS-PSO，ACO-PSO の性能実験，すなわち PSO チューニングの汎用性と最適化能力についての計算機実験の結果を示す．ただし，計算機実験に使用したベンチマーク問題例は TSPLIB [23] である．実験環境を表 4 に示す．

4.1 適応パラメータチューニングの効果

SI アルゴリズムにおける戦略構築パラメータ項目と PSO チューニング対象項目の関係を調べるために，最も高次元なチューニング対象を持つ CGS の PSO チューニング対象をすべての組合せで計算機実験を行う．すなわち， 2^6 通りの組合せについて，各組合せで 5 回分の平均値を計測する計算機実験を実施し，それを 3 つ以上の問題例について実施し， $2^6 \times 5 \times 3$ 項目以上の実験を行う．その際，チューニング対象を表す変数を式 (14) のように 2 進数で定義する．

$$Tuning_{PSO,CGS}(a_0, \beta_{cgs}, \alpha_{cgs}, \gamma, q_{0,cgs}, b_0) = 000000 \quad (14)$$

式 (14) は上の桁から順番に対象のパラメータをチューニングするかどうかを表すフラグが格納される．たとえば， α_{cgs} パラメータのみをチューニング対象とするならば $Tuning_{PSO,CGS}(a_0, \beta_{cgs}, \alpha_{cgs}, \gamma, q_{0,cgs}, b_0) = 001000$ とする．

使用した CGS パラメータを表 5 に示す．ただし， $Tuning_{PSO,CGS}$ により，適応パラメータチューニングの対象となったものは $[K_{min}, K_{max}]$ の範囲の乱数をかけて初期値とする．これらのパラメータは参考文献 [27] を参考に決定した．使用した問題例は d493, d657, pr1002 であり，実行時間はそれぞれ 500 s, 600 s, 800 s とした．実験結果を平均誤差率でソートしたグラフを図 2，図 3，図 4 に示す．すべての問題例において $Tuning_{PSO,CGS} = 111111$ の結果は上位 1 位の結果と 0.8% 以下となった．ゆえに，すべてのパラメータを PSO チューニング対象とすることで，問題例に対する汎用性が得られると考えられる．このこと

表 4 実験環境

Table 4 The environment.

項目	スペック
CPU	Intel E5-2620v2 × 2 (24 threads)
MEM	32 GB
HDD	1 TB
CHIP-SET	C600

から，SI アルゴリズムにおいて戦略構築時に参照されるすべてのパラメータをチューニング対象にすると任意の問題例に適應できると予測できる．すなわち，ACS，ACO も同様にすべてのパラメータをチューニング対象とすることで強力な汎用性と最適化能力が得られると考えられる．また，パラメータ $q_{0,cgs}$ のチューニングフラグを境目に急激に誤差率が改善された．

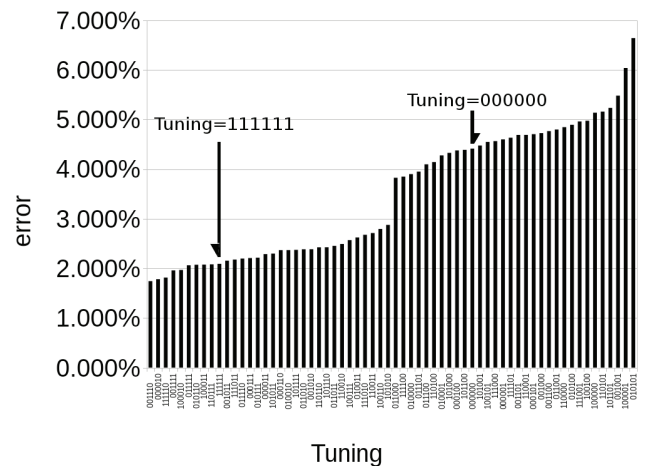
4.2 改悪パラメータの耐性

4.1 節では，本研究で取り扱う SI アルゴリズムとして最も高次元なチューニング対象を持つアルゴリズムである CGS について，PSO によるチューニング特性を確認し

表 5 CGS(-PSO) パラメータ

Table 5 Parameters of CGS(-PSO).

項目	値
R_T	1
R_{max}	40
R_B	8
k_w	3
r_0	0.0000003
α_{cgs}	7
γ	7
$q_{0,cgs}$	0.9
b_0	0.98
β_{cgs}	12
a_0	0.9
C_s	100
R_d	6
R_F	3
w	1000
$c_{1,cgs}$	0.0002
$c_{2,cgs}$	0.005
K_{min}	0.9
K_{max}	2
N_{CGS}	10

図 2 実験結果：d493- $Tuning_{PSO,CGS}$ 特性Fig. 2 Results: d493- $Tuning_{PSO,CGS}$ characteristics.

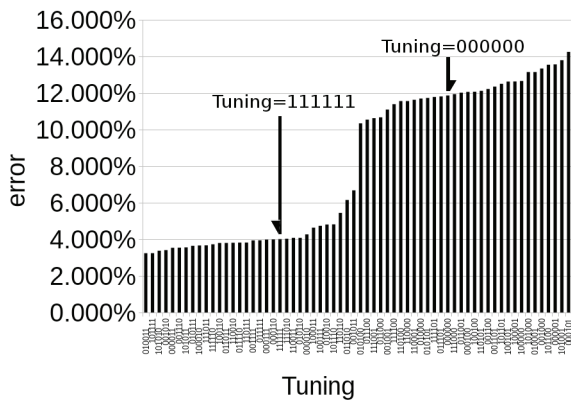


図 3 実験結果：d657- $Tuning_{PSO,CGS}$ 特性

Fig. 3 Results: d657- $Tuning_{PSO,CGS}$ characteristics.

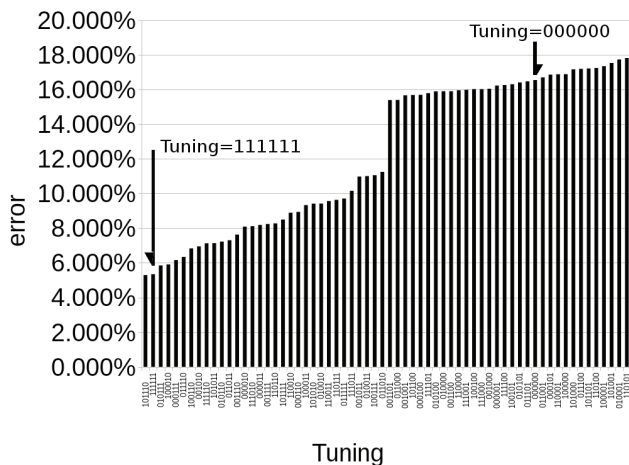


図 4 実験結果：pr1002- $Tuning_{PSO,CGS}$ 特性

Fig. 4 Results: pr1002- $Tuning_{PSO,CGS}$ characteristics.

た．本節では故意に改悪された個々のパラメータを初期値として PSO 適応パラメータチューニングの効果について確認を行う．パラメータ項目を 1 つ選択し，選択された項目に対して改悪パラメータを与える．その項目に限ってパラメータチューニングを行い，適応パラメータチューニングを行わない CGS と比較することで，改悪パラメータへの効果を確認する．なお，パラメータの改悪は表 5 に示したのから，改悪対象を 0.25 倍して使用するものとし，実験結果は bad として命名する．実行時間は 4.1 節と同じである．また，対象の問題例は d493, d657, pr1002, u1432, d2103 であり，実行時間をそれぞれ 500s, 600s, 800s, 6000s, 15000s とし，各パラメータ項目の PSO 適応パラメータチューニングの特性についてより詳しく検証する．なお，計算機実験は 5 回の平均値とする．

実験結果をグラフにしたものを図 5, 図 6, 図 7, 図 8, 図 9 に示す．実験結果より，改悪されたパラメータに対しても PSO 適応パラメータチューニングが有効であること， $q_{0,CGS}$ と β_{CGS} パラメータへのチューニングが特に有効であることが判明した．すなわち， $q_{0,CGS}, \beta_{CGS}$ についてパラメータチューニングを行うと，CGS は高性能化するこ

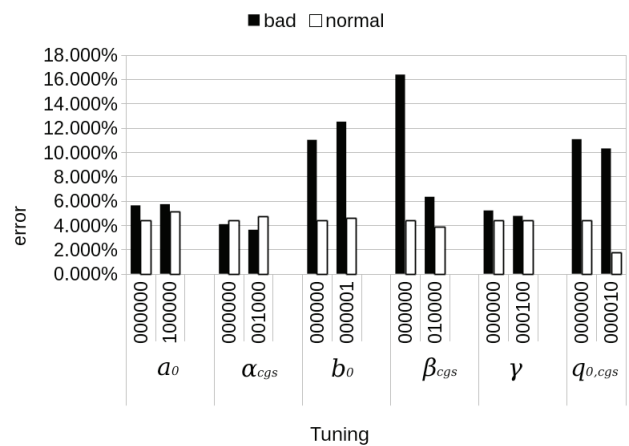


図 5 実験結果：d493- $Tuning_{PSO,CGS}, Bad$ 特性

Fig. 5 Results: d493- $Tuning_{PSO,CGS}, Bad$ characteristics.

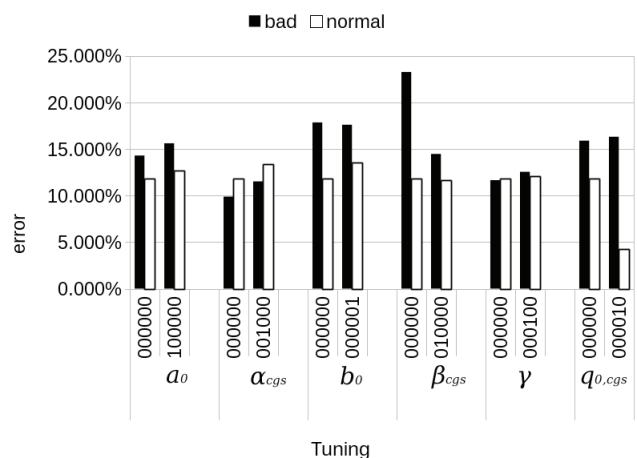


図 6 実験結果：d657- $Tuning_{PSO,CGS}, Bad$ 特性

Fig. 6 Results: d657- $Tuning_{PSO,CGS}, Bad$ characteristics.

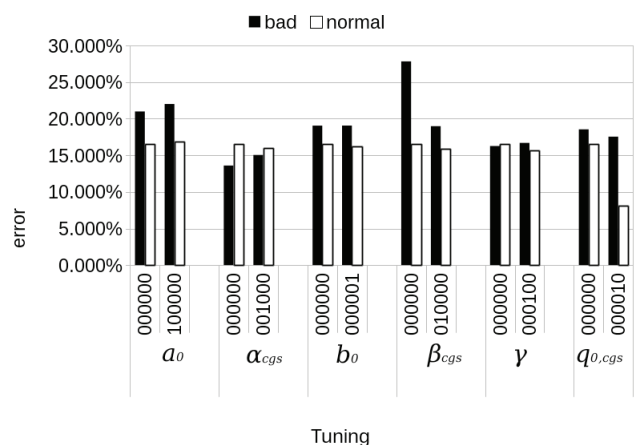


図 7 実験結果：pr1002- $Tuning_{PSO,CGS}, Bad$ 特性

Fig. 7 Results: pr1002- $Tuning_{PSO,CGS}, Bad$ characteristics.

とが明らかになった．その他のパラメータについては各問題例ごとに性質が異なる結果となった．たとえば，図 9 の改悪された b_0 パラメータへの PSO 適応パラメータチューニングの効果は数%誤差率を下げる結果となったが，他の問題例ではその傾向はみることができなかった．ゆえに，

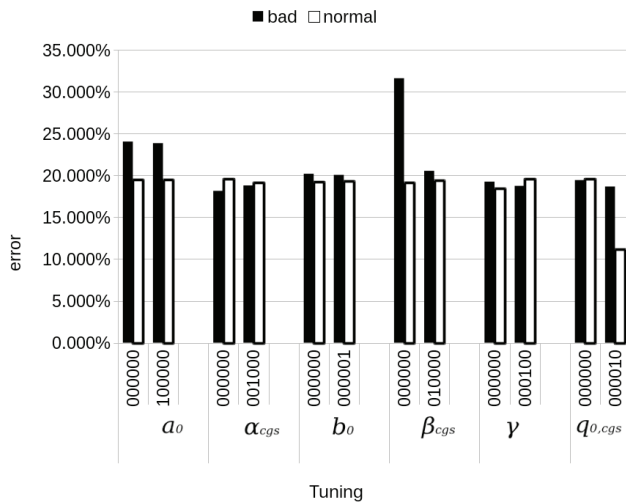


図 8 実験結果：u1432- $Tuning_{PSO,CGS}$, Bad 特性

Fig. 8 Results: u1432- $Tuning_{PSO,CGS}$, Bad characteristics.

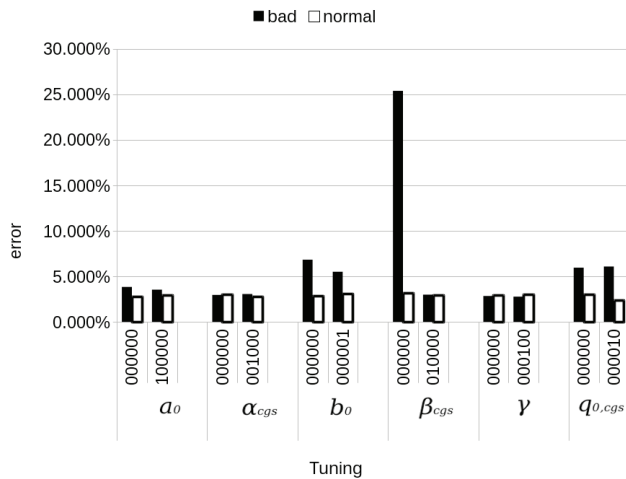


図 9 実験結果：d2103- $Tuning_{PSO,CGS}$, Bad 特性

Fig. 9 Results: d2103- $Tuning_{PSO,CGS}$, Bad characteristics.

$q_{0,cgs}$, β_{cgs} を PSO 適応パラメータチューニングの対象とすることが有効であることが示された。また、初期パラメータが悪くても適応パラメータチューニングを行うことで十分良い解が得られることが分かった。

4.3 固定パラメータの特性

4.2 節より、CGS-PSO は $q_{0,cgs}$, β_{cgs} について適応パラメータチューニングを行うと高性能化することが判明したが、CGS についてはパラメータ $q_{0,cgs}$, β_{cgs} の値が表 5 より 1 種類しか与えられておらず、それが最適ではない可能性が十分に考えられる。すなわち、CGS について多くの $q_{0,cgs}$, β_{cgs} の値で計算機実験を行うことで、CGS-PSO に勝る結果が発生する可能性があるため、それらを計測することで PSO 適応パラメータチューニングの強力な最適化能力を検証する。実験結果を表 6、表 7、表 8 に示す。

d493 の最小平均誤差率は $(q_{0,cgs}, \beta_{cgs}) = (0.9, 30)$ で 3.157%, d657 の最小平均誤差率 $(0.9, 20)$ で 8.080%, pr1002

表 6 実験結果：d493-CGS- $q_{0,cgs}$, β_{cgs} 特性

Table 6 Results: d493-CGS- $q_{0,cgs}$, β_{cgs} characteristics.

CGS	誤差率 [%]	$q_{0,cgs}$					
		0.5	0.6	0.7	0.8	0.9	1
β_{cgs}	10	10.445	12.774	10.042	12.531	5.451	17.125
	20	12.362	11.416	9.288	8.465	5.120	17.330
	30	11.782	11.231	9.497	7.545	3.157	17.128
	40	9.802	13.779	10.539	10.194	4.277	17.130
	50	10.237	9.079	10.817	6.068	3.957	16.719
	60	12.428	11.388	12.054	7.182	6.471	17.653
CGSPSO	Tuning=010010	2.370					

表 7 実験結果：d657-CGS- $q_{0,cgs}$, β_{cgs} 特性

Table 7 Results: d657-CGS- $q_{0,cgs}$, β_{cgs} characteristics.

CGS	誤差率 [%]	$q_{0,cgs}$					
		0.5	0.6	0.7	0.8	0.9	1
β_{cgs}	10	16.777	18.531	16.769	16.611	16.121	20.876
	20	16.278	15.843	16.323	15.722	8.080	21.631
	30	17.439	16.166	16.601	16.252	12.852	21.263
	40	16.129	16.540	15.411	16.000	11.167	19.782
	50	17.415	16.773	16.560	15.360	9.861	22.121
	60	16.401	15.806	20.020	18.202	9.585	19.658
CGSPSO	Tuning=010010	4.809					

表 8 実験結果：pr1002-CGS- $q_{0,cgs}$, β_{cgs} 特性

Table 8 Results: pr1002-CGS- $q_{0,cgs}$, β_{cgs} characteristics.

CGS	誤差率 [%]	$q_{0,cgs}$					
		0.5	0.6	0.7	0.8	0.9	1
β_{cgs}	10	17.498	18.420	18.518	17.075	15.951	18.247
	20	16.882	16.691	16.802	18.672	15.391	17.988
	30	18.201	17.763	18.396	15.127	15.981	18.154
	40	17.744	17.860	17.019	15.568	15.015	18.227
	50	18.293	16.823	17.982	15.281	14.295	18.347
	60	17.187	18.620	16.628	17.169	16.978	19.560
CGSPSO	Tuning=010010	9.422					

は $(0.9, 50)$ で 14.295%であった。 β_{cgs} パラメータに関しては、特定の値で性能が良くなるとはいえなかったが、他のパラメータと相関がある可能性も十分に考えられる。CGS は $q_{0,cgs}$ が 0.9 で性能が良くなる傾向がみられたが、すべての問題例の誤差率の最小値が CGS-PSO($Tuning_{PSO,CGS} = 010010$) よりも性能が劣ることが明らかとなった。ゆえに、どのような優れたパラメータを CGS に与えることができたとしても、改悪されたあるいは適切なパラメータに設定された CGS-PSO には性能で勝ることはなく、PSO 適応パラメータチューニングの強力さが示された。本研究で使用している PSO の粒子は抵抗値 0 のため減衰することなくパラメータ空間を飛翔し続けるので、パラメータを一定値へと収束させ固定し続けるよりも、PSO により良好なパラメータ周りを振動させておく方が高性能であることが示された。ゆえに、振動系のパラメータチューニングの方が SI には適していると推測できる。

4.4 適応パラメータチューニングの性能

4.3 節までの内容では、標準的な SI アルゴリズムとして CGS を用いて試験を行い、CGS-PSO は最良にパラメータ設定された CGS よりも高性能であることが示された。本節では、ACO, ACS, CGS のアルゴリズムに PSO 適応

表 9 ACO(-PSO) パラメータ
Table 9 Parameters of ACO(-PSO).

項目	値
α_{aco}	1
β_{aco}	5
Q_{aco}	1
ρ_{aco}	0.3
$\tau_{max,aco}$	18
$\tau_{min,aco}$	1
$\tau_{init,aco}$	10
$c_{1,aco}$	0.0002
$c_{2,aco}$	0.005
$\alpha_{max,aco}$	8
$\alpha_{min,aco}$	-8
$\beta_{max,aco}$	8
$\beta_{min,aco}$	-8
N_{ACO}	min(sizeof(問題例),200)
N_{ACOPSO}	10

表 10 ACS(-PSO) パラメータ
Table 10 Parameters of ACS(-PSO).

項目	値
α_{acs}	1
β_{acs}	2
Q_{acs}	0.1
$q_{0,acs}$	0.9
ρ_{acs}	0.1
$\tau_{max,acs}$	8
$\tau_{min,acs}$	0.0001
$\tau_{init,acs}$	0.005
$c_{1,acs}$	0.0002
$c_{2,acs}$	0.005
$\alpha_{max,acs}$	8
$\alpha_{min,acs}$	-8
$\beta_{max,acs}$	8
$\beta_{min,acs}$	-8
$q_{0,min}$	0
$q_{0,max}$	1
N_{ACS}	min(sizeof(問題例),200)
N_{ACSPSO}	10

パラメータチューニングを施した ACO-PSO, ACS-PSO, CGS-PSO と非適応アルゴリズムを TSPLIB ベンチマークで比較する。なお、実験は無作為に選択した TSPLIB の問題例 10 個に対して、非適応アルゴリズムと PSO 適応パラメータチューニングのアルゴリズムの 3×2 種類のアルゴリズムを各 10 回の計算機実験を行い平均値を計測する。実行時間は ACO に合わせて長くし、実験結果の表 11 に示す。ACO(-PSO), ACS(-PSO) の計算機実験に使用したパラメータを表 9, 表 10 に示す。ただし、蟻コロニーアルゴリズムは適応パラメータチューニング対象であるパラメータの初期値は最小値と最大値の範囲から乱数で定める。PSO 適応パラメータチューニングにおける蟻コロニー

表 11 実験結果：PSO 適応パラメータチューニング
Table 11 Results: adaptive parameter tuning - PSO.

問題例	時間 [s]	手法	平均値	平均誤差率 [%]	最小値	最小誤差率 [%]
eil101	30	ACO	770.0	22.417	753	19.714
eil101	30	ACOPSO	709.5	12.798	694	10.334
eil101	30	ACS	648.3	3.068	633	0.636
eil101	30	ACSPSO	676.0	7.472	661	5.087
eil101	30	CGS	634.9	0.938	629	0.000
eil101	30	CGSPSO	637.2	1.304	630	0.159
d198	60	ACO	19472.0	23.397	19164	21.445
d198	60	ACOPSO	18317.7	16.082	18054	14.411
d198	60	ACS	17789.8	12.736	17470	10.710
d198	60	ACSPSO	17693.8	12.128	17118	8.479
d198	60	CGS	15911.3	0.832	15868	0.558
d198	60	CGSPSO	15918.9	0.880	15834	0.342
a280	120	ACO	3475.9	34.777	3368	30.593
a280	120	ACOPSO	3093.9	19.965	3056	18.496
a280	120	ACS	3106.5	20.454	2994	16.092
a280	120	ACSPSO	2855.1	10.706	2747	6.514
a280	120	CGS	2612.7	1.307	2588	0.349
a280	120	CGSPSO	2613.8	1.349	2585	0.233
d657	1500	ACO	69527.4	42.148	68491	40.029
d657	1500	ACOPSO	61313.3	25.354	60399	23.485
d657	1500	ACS	66113.1	35.167	64987	32.865
d657	1500	ACSPSO	58375.6	19.348	55881	14.248
d657	1500	CGS	52536.3	7.410	51683	5.665
d657	1500	CGSPSO	50300.9	2.840	50014	2.253
rat783	1800	ACO	12628.2	43.404	12499	41.937
rat783	1800	ACOPSO	11060.2	25.598	11004	24.960
rat783	1800	ACS	12177.1	38.282	11967	35.896
rat783	1800	ACSPSO	10250.3	16.401	9956	13.059
rat783	1800	CGS	9773.5	10.987	9485	7.711
rat783	1800	CGSPSO	9022.1	2.454	8963	1.783
pr1002	2100	ACO	382141.4	47.519	377859	45.866
pr1002	2100	ACOPSO	330229.3	27.480	325872	25.797
pr1002	2100	ACS	371006.4	43.221	358972	38.575
pr1002	2100	ACSPSO	309690.2	19.551	300364	15.951
pr1002	2100	CGS	294986.1	13.874	289427	11.728
pr1002	2100	CGSPSO	268531.7	3.662	266404	2.841
u1432	6000	ACO	227029.2	48.414	224835	46.980
u1432	6000	ACOPSO	196811.6	28.660	195287	27.664
u1432	6000	ACS	232595.1	52.053	229314	49.908
u1432	6000	ACSPSO	182390.9	19.233	179233	17.169
u1432	6000	CGS	182097.4	19.041	180077	17.720
u1432	6000	CGSPSO	159444.9	4.233	158610	3.687
d2103	15000	ACO	123694.5	53.753	122745	52.573
d2103	15000	ACOPSO	99575.1	23.773	98815	22.828
d2103	15000	ACS	131709.1	63.715	128916	60.244
d2103	15000	ACSPSO	86767.5	7.853	86219	7.171
d2103	15000	CGS	82530.9	2.587	82370	2.387
d2103	15000	CGSPSO	82081.6	2.028	81750	1.616
f3795	27000	ACO	46190.8	60.541	45162	56.965
f3795	27000	ACOPSO	38697.3	34.496	38284	33.060
f3795	27000	ACS	46788.3	62.617	45553	58.324
f3795	27000	ACSPSO	33580.9	16.714	32633	13.419
f3795	27000	CGS	33553.7	16.619	33109	15.074
f3795	27000	CGSPSO	30410.2	5.694	29991	4.237
rl5934	45000	ACO	901594.0	62.144	892246	60.463
rl5934	45000	ACOPSO	740475.4	33.168	735265	32.231
rl5934	45000	ACS	1125774.1	102.461	1109475	99.530
rl5934	45000	ACSPSO	663385.2	19.304	658124	18.358
rl5934	45000	CGS	661922.3	19.041	659244	18.559
rl5934	45000	CGSPSO	599915.1	7.890	583906	5.011

に用いられる蟻数は参考文献 [15] によると、10 体が最良であると示されているので、PSO 適応パラメータチューニングの場合のみ 10 体とする。すべての実験結果を表 11 に示す。実験結果より、PSO 適応パラメータチューニングを施したアルゴリズムがきわめて高い性能を示した。サイズの大きい問題例に対しては、CGS-PSO は非常に良い解を算出しているといえる。高次元 PSO 適応パラメータチューニングと CGS と PSO の相性が両方作用したことが

表 12 実験結果：文献 [15] との比較

Table 12 Results: comparison with Ref. [15].

問題例	時間 [s]	手法	平均値	平均誤差率 [%]	最小値	最小誤差率 [%]
eil101	30	ACO	770.0	22.417	753	19.714
eil101	30	ACOPSO	709.5	12.798	694	10.334
eil101	30	ACS	648.3	3.068	633	0.636
eil101	30	ACSPSO	676.0	7.472	661	5.087
eil101	30	CGS	634.9	0.938	629	0.000
eil101	30	CGSPSO	637.2	1.304	630	0.159
eil101	302.15	ACO-PSO-3opt [15]	632.7	0.588	631	0.318

あげられる。CGS のアルゴリズムを考慮すると戦略構築方法が *Mode* が *Sabbatical* と *Normal* の 2 通りあり、それぞれで使用するパラメータが独立して存在するので、各戦略が独立して最適化され結果として強力な最適化能力を示したと考えられる。

また、文献 [15] から問題例 eil101 の実験結果を参照し、比較を行ったものを表 12 に示す。平均誤差率は ACO-PSO-3opt の方が性能が良い結果が出たが、3opt の効果によるものが大きいと考えられる。なぜなら、ACO-PSO-3opt の最小誤差率が大きいので、文献 [15] の ACO-PSO により最適解を含まない局所解へ誘導され、3opt はその局所解を脱出することができなかったからであると考えられる。

4.5 適応パラメータチューニングの動的特性

4.4 節まで PSO 適応パラメータチューニングの有効性について述べた。本節では PSO 適応パラメータチューニングの動的特性、すなわち、各イテレーションにおけるパラメータの振舞い、および個体群が保持している解のばらつきを誤差率の推移とともに計測することで PSO 適応パラメータチューニングがなぜ良いのかを考察する。測定する対象は CGS であり、計測対象は 4.2 節で示した重要なパラメータ $q_{0,cgs}$, β_{cgs} と誤差率の推移、解のばらつきである。解のばらつきの評価は TSPLIB にて配布されている最適解 x_{opt} の情報を用いて、式 (15) で示される定義に基づき、個体が持っている順回路と最適解を比較し、一致している枝の本数を求め、その標準偏差を解のばらつきとする。

$$M_{opt}(x) = count([0, n-1], i \rightarrow (x_i, x_{(i+1)modn}) \in x_{opt}) \quad (15)$$

使用した問題例は pr1002, pr2392 であり、実行時間はそれぞれ 2100s, 18000s である。PSO 適応パラメータチューニングを行わない状態とすべてのパラメータ項目を PSO 適応パラメータチューニングの対象とした結果を pr1002 を図 10, 図 11, pr2392 を図 12, 図 13 に示す。これらのグラフはパラメータチューニングを行っている PSO が参照している最良パラメータ z_{best} の β_{cgs} , $q_{0,cgs}$ と CGS の個体群が保持している解のばらつきである $SD[M_{opt}(x)]$, CGS-PSO システムが算出したその時点での最小誤差率をプロットしたものである。グラフでの凡例は q_0 (緑) が $q_{0,cgs}$, β (黄) が β_{cgs} , $error$ (青) が最小誤差率, SD (橙)

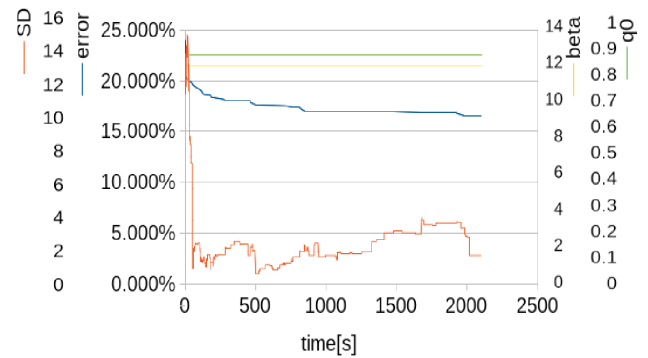


図 10 実験結果：pr1002-Tuning = 000000 動的特性

Fig. 10 Results: pr1002-Tuning = 000000 dynamic characteristics.

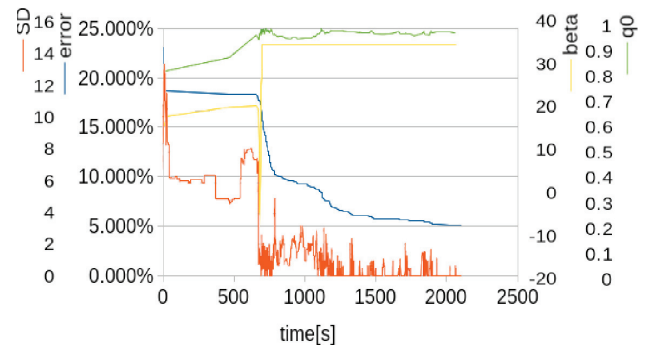


図 11 実験結果：pr1002-Tuning = 111111 動的特性

Fig. 11 Results: pr1002-Tuning = 111111 dynamic characteristics.

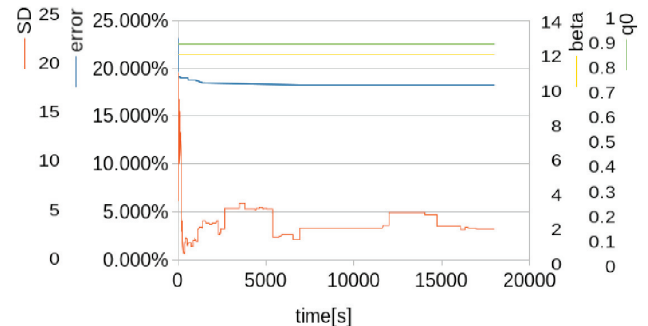


図 12 実験結果：pr2392-Tuning = 000000 動的特性

Fig. 12 Results: pr2392-Tuning = 000000 dynamic characteristics.

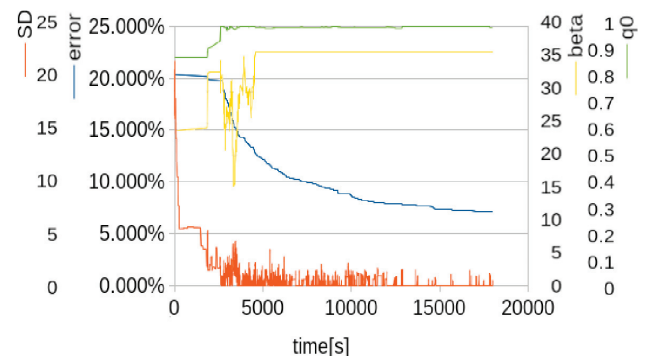


図 13 実験結果：pr2392-Tuning = 111111 動的特性

Fig. 13 Results: pr2392-Tuning = 111111 dynamic characteristics.

が $SD[M_{opt}(x)]$ に対応している．実験結果より，PSO 適応パラメータチューニングが施されていない方は解のばらつきが大きく誤差率の収束も遅い．PSO 適応パラメータチューニングが施された方は $q_{0,cgs}$ パラメータが 1 に近づくとき急激に解のばらつきが小さくなり，誤差率の収束も速くなった．解のばらつきが 0 になった後もインパルス的に発生していることが分かる．インパルスのなばらつきの発生は β_{cgs} の振動部分でよく見ることができ，その後は発生間隔が大きくなっていった．これは仮想人間の *Mode* が *Sabbatical* に遷移したため個体間の誘導探索が行われなくなったことが要因であると考えられる．まとめると，CGS-PSO アルゴリズムは $q_{0,cgs}$ の最適化により，個体間の解のばらつきをなくして局所探索性能を引き上げ，仮想人間の *Sabbatical* 状態と β_{cgs} への振動的なパラメータチューニングを用いて，インパルス的に解のばらつきを与えることにより，解の多様性に貢献したと考えられる．

5. 関連研究

Mostafa Mahi らの研究 [15] では，TSP を解くために ACO に PSO を用いたパラメータチューニングを施す手法を提案している．ACO-PSO-3opt と呼ばれるこの手法は蟻コロニー最適化アルゴリズムのパラメータをチューニングするために粒子群最適化を利用し，アルゴリズム終了後に 3opt を適用している．ACO の個体が戦略を構築し，すべての個体が戦略構築を終了した後に，個体が持つ解の評価値を利用して PSO によりパラメータを 1 単位時間分更新を行う．ACO-PSO-3opt は ACO よりも高い性能を示すことが報告されている．この手法はパラメータ最適化に個体が持つ評価値をそのまま利用するので，目的関数が凸でない場合や離散的であっても，最適化が可能である．機構が単純で反復演算を行うアルゴリズムを強化することが単純であるので，新規に提案された PSO アルゴリズムを利用すること [9], [10] や，パラメータチューニングが難しいアルゴリズムへの適用が期待される．

それ以外の動的パラメータチューニングの研究には，PSO を用いて数値最適化問題を効率良く解くものがある．Teruyoshi Yamaguchi らの研究 [9] は PSO アルゴリズムで数値最適化問題を解く際に，PSO の運動方程式における各個体の加速に用いられる弾性係数を各個体特有の独立したパラメータ変数とみなし，その変数ベクトルに質量を与え移動速度に比例した抵抗力のある無重力空間に初期速度を与え放出する．それらの質点すなわち PSO パラメータの c_1, c_2 に相当するものは次第にエネルギーを失い，収束することでパラメータチューニングを行う．Nobuhiro Iwasaki らの研究 [10] は粒子群の平均速度情報を用いて，任意の粒子に発生する慣性係数 w を小さい値に収束させることでパラメータチューニングを行う．これらの関連研究はアルゴリズム実行中にパラメータを書き換えている．しか

し，関連研究 [15] はパラメータ変数がパラメータ空間上を減衰することなく振動し続けるのに対し，関連研究 [9], [10] は最適値へと収束していく違いがある．

著者らの研究 [27] では，CGS のハイブリッド手法として ACS におけるフェロモン情報を CGS の個体の 1 種であるコンサルタントの戦略に引き継ぎ，情報の共有を行って並列計算を行う手法を示した．ACS で探索を行った後，CGS で探索を行う 2 フェーズからこのアルゴリズムは構築されている．ACS で探索を行い，それによって得られた解をコンサルタントの解に設定して，CGS が実行される．この際，フェーズ 1 で得られたフェロモン情報を引き継ぎ，その情報が更新される．並列環境では，全体のフェロモン情報が更新される際に，すべての計算ノードで最良解の共有を行う．この手法は単純並列 CGS，ACS よりも性能が高いことが報告されている．

6. まとめ

PSO によるパラメータチューニングは非常に高い最適化能力を示した．4.1 節では戦略構築に使用されるパラメータをすべてチューニング対象にすることで，問題例に対する非常に高い汎用性が得られることが明らかになった．4.2 節では改悪パラメータの耐性に関する性能が明らかとなり，特に β_{cgs} と $q_{0,cgs}$ パラメータがそれぞれ CGS の性能を左右することが明らかとなった．4.3 節では CGS に多くの $\beta_{cgs}, q_{0,cgs}$ パラメータを与え，その中で最も良かった解を CGS-PSO の解と比較することで，CGS-PSO の強力な最適化能力を示した．4.4 節では 3 種類の SI アルゴリズムに PSO による適応パラメータチューニングが有効であることを示し，アルゴリズムに対する汎用性の高さを示した．特に，CGS-PSO は大規模な問題例に対しても良い結果を示した．4.5 節では CGS-PSO の $\beta_{cgs}, q_{0,cgs}$ と誤差率と解のばらつきの推移の動的特性を計測した． z_{best} の $q_{0,cgs}$ パラメータが 1 に近づき解のばらつきを抑え，局所探索性能を引き上げ， β_{cgs} の振動と仮想人間の *Mode* の *Sabbatical* により，解のばらつきがインパルス的に発生し様々な領域を効率良く探索していると考えられる．

PSO 適応パラメータチューニングはアルゴリズム自体が PSO で非常に単純ではあるが，既存の SI アルゴリズムに対して高い最適化能力を与えることができた．これは戦略構築に関連するすべてのパラメータをチューニング対象とするという非常に簡単な手法で強力な最適化能力と汎用性を得ることができることを示した．さらに，静的なパラメータチューニングはいかなる最良なものを与えることができたとしても，PSO 適応パラメータチューニングの最適化能力には勝ることがないことを示し，任意のアルゴリズムに対する最適化の可能性を十分に示した実験結果であるといえる．また，CGS において PSO 適応パラメータチューニングが特に有効であった $q_{0,cgs}, \beta_{cgs}$ に対応するパ

ラメータ, たとえば ACS における $q_{0,acs}, \beta_{acs}$, をパラメータチューニングの対象とすることで, 他のアルゴリズムに対して PSO 適応パラメータチューニングが性能向上に大きく寄与することが期待される。

コンサルタント誘導型探索は蟻コロニーアルゴリズムと解の構築手法の基本方針が似ている面がある。TSP のように解の構築に解の要素の候補から選択することで構築を行う点であり, このように基本方針が似ているならば, 蟻コロニーアルゴリズムの応用手法をコンサルタント誘導型探索にも適用可能であると考えられる。蟻コロニーアルゴリズムは混合変数最適化問題 [28] やルーティングとタスキング問題 [29], パターンマッチング [30] などへの応用例が近年報告されている。CGS も巡回セールスマン問題 [31] 以外にジョブショップスケジューリング問題 [32], 2 次割当問題 [33] への応用例が報告されている。応用先の問題および SI アルゴリズムにおいても, 本研究が示した PSO 適応パラメータチューニングの効果は個体群に最適化された探索性能を与えることができると考えられる。

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

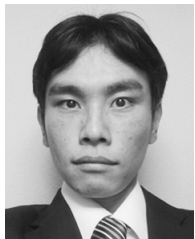
- [1] Dorigo, M.: Optimization, Learning and Natural Algorithms, Ph.D. Thesis, Politecnico di Milano, Italy (1992).
- [2] El-Ghazali, T.: *Metaheuristics: From Design to Implementation*, Wiley Publishing (2009).
- [3] Zhang, L., Wang, L. and Zheng, D.-Z.: An adaptive genetic algorithm with multiple operators for flowshop scheduling, *Int. J. Adv. Manuf. Technol.*, Vol.2006, No.27, pp.580-587 (2004).
- [4] Tang, J., Lim, M.H. and Ong, Y.S.: Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems, *Soft Comput.*, Vol.2007, No.11, pp.873-888 (2007).
- [5] Geng, X., Chen, Z., Yang, W., Shi, D. and Zhao, K.: Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search, *Applied Soft Computing*, Vol.11, pp.3680-3689 (2011).
- [6] Xiao, W. and Dunford, W.G.: A Modified Adaptive Hill Climbing MPPT Method for Photovoltaic Power Systems, *2004 35th Annual IEEE Power Electronics Specialists Conference* (2004).
- [7] Zhu, K.Q.: *A Diversity-controlling Adaptive Genetic Algorithm for the Vehicle Routing Problem with Time Windows*, National University of Singapore (2003).
- [8] Hasançebi, O., Erdal, F. and Saka, M.P.: Adaptive Harmony Search Method for Structural Optimization, *Journal of Structural Engineering*, Vol.136 No.4 (2010).
- [9] Yamaguchi, T. and Yasuda, K.: Adaptive Particle Swarm Optimization, *2006 IEEE International Conference on Systems, Man, and Cybernetics* (2006).
- [10] Iwasaki, N., Yasuda, K. and Ueno, G.: Dynamic Parameter Tuning of Particle Swarm Optimization, *Trans. Electrical and Electronic Engineering IEEJ*, Vol.2006, No.1, pp.35-363 (2006).
- [11] Wong, K.Y.: Komarudin, Parameter Tuning for Ant Colony Optimization: A Review, *Proc. International Conference on Computer and Communication Engineering 2008* (2008).
- [12] Eiben, A.E., Hinterding, R. and Michalewicz, Z.: Parameter Control in Evolutionary Algorithms, *IEEE Trans. Evolutionary Computation*, Vol.3, No.2 (1999).
- [13] Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C. and Paredes, F.: Parameter tuning of a choice-function based hyperheuristic using Particle Swarm Optimization, *Expert Systems with Applications*, Vol.40, pp.1690-1695 (2013).
- [14] Jackson, W.G., Ozcan, E. and John, R.I.: *Fuzzy Adaptive Parameter Control of a Late Acceptance Hyperheuristic*, ASAP Research Group, School of Computer Science, University of Nottingham Jubilee Campus, UK (2014).
- [15] Mahi, M., Baykan, O.K. and Kodaz, H.: A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem, *Applied Soft Computing*, Vol.30, pp.484-490 (2015).
- [16] Kumar, V., Chhabra, J.K. and Kumar, D.: Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems, *Journal of Computational Science*, No.5, No.2014, pp.144-155 (2013).
- [17] Niknam, T., Azizipanah-Abarghooee, R. and Roosta, A.: Reserve Constrained Dynamic Economic Dispatch: A New Fast Self-Adaptive Modified Firefly Algorithm, *IEEE Systems Journal*, Vol.6, No.4 (Dec. 2012).
- [18] Ponz-Tienda, J.L., Yepes, V., Pellicer, E. and Moreno-Flores, J.: The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm, *Automation in Construction*, Vol.29, pp.161-172 (2013).
- [19] Xu, G.: An adaptive parameter tuning of particle swarm optimization algorithm, *Applied Mathematics and Computation*, Vol.219, pp.4560-4569 (2013).
- [20] Olivas, F., Valdez, F. and Castillo, O.: Dynamic parameter adaptation in Ant Colony Optimization using a fuzzy system for TSP problems, *16th World Congress of the International Fuzzy Systems Association (IFSA), 9th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)* (2015).
- [21] Lee, M.A. and Takagi, H.: Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques, *Proc. 5th Int'l Conf. on Genetic Algorithms (ICGA '93)*, Urbana-Champaign, IL, pp.76-83 (1993).
- [22] Alaya-Feki, A.B.H., Sayrac, B. and Moulines, E.: Semi dynamic parameter tuning for optimized opportunistic spectrum access, 978-1-4244-1722-3/08/©2008, *IEEE* (2008).
- [23] TSPLIB, available from (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>) (accessed 2015-08-29).
- [24] Iordache, S.: Consultant-Guided Search — A New Metaheuristic for Combinatorial Optimization Problems, *GECCO'10*, July 7-11, Portland, Oregon, USA (2010).
- [25] Dorigo, M. and Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evolutionary Computation*, Vol.1, pp.53-66 (1997).
- [26] Microsoft: Language-Integrated Query: LINQ, available from (<https://msdn.microsoft.com/ja-jp/library/bb397926.aspx>) (accessed 2015-10-09).

- [27] 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).
- [28] Liao, T., Socha, K., Montes de Oca, M.A., Stutzle, T. and Dorigo, M.: Ant Colony Optimization for Mixed-Variable Optimization Problems, *IEEE Trans. Evolutionary Computation*, Vol.18, No.4 (2014).
- [29] Zaza, T. and Richards, A.: Ant Colony Optimization for Routing and Tasking Problems for Teams of UAVs, *UKACC International Conference on Control 9th-11th* (2014).
- [30] Sreeja, N.K. and Sankar, A.: Pattern Matching based Classification using Ant Colony Optimization based Feature Selection, *Applied Soft Computing*, Vol.31, pp.91-102 (2015).
- [31] Iordache, S.: Consultant-guided search algorithms with local search for the traveling sales man problem, *Parallel Problem Solving from Nature, PPSN XI*, Vol.6239, pp.81-90 (2010).
- [32] Deepanandhini, D. and Amudha, T.: Solving job shop scheduling problems with consultant guided search metaheuristics, *International Journal of Software and Web Sciences*, Vol.1, No.1, pp.1-6 (2013).
- [33] Iordache, S.: Consultant-guided search algorithms for the quadratic assignment problem, *Hybrid Metaheuristics*, Vol.6373, pp.148-159 (2010).



榎原 博之 (正会員)

1982 年大阪大学工学部通信科卒業。
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。工博。IEEE, ACM 各会員。



長辻 亮太

1993 年生。2015 年 3 月関西大学シ
ステム理工学部電気電子情報工学科卒
業。同年 4 月関西大学大学院理工学研
究科システム理工学専攻電気電子情報
工学分野アルゴリズム工学研究室に配
属。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への取り組みに従事。



飯田 修平

1991 年生。2014 年 3 月関西大学シ
ステム理工学部電気電子情報工学科卒
業。同年 4 月関西大学大学院理工学研
究科システム理工学専攻電気電子情報
工学分野アルゴリズム工学研究室に配
属。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。