

# 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<sup>1,a)</sup> 飯田 修平<sup>1</sup> 榎原 博之<sup>2,b)</sup>

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、それらと比較することで 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、固定パラメータチューニングと比較し、PSO 適応パラメータチューニングの効果を分析する。

## 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、解空間と制約条件と目的関数を与え定式化する。解空間が離散であるものは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近似解法と厳密解法 2 通りに大別される。この問題は解空間のサイズが巨大であるため、厳密解法で探索することは現実的ではない。ゆえに、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ている。近似解法に含まれるメタヒューリスティクスの研究が盛んに行われている。

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] ~ [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 [7] ~ [19] 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。

文献 [12] の手法では、各個体毎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。それら全てのパラメータを 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。

## 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

<sup>1</sup> 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,564-8680,Japan

<sup>2</sup> 関西大学 システム理工学部 電気電子情報個学科  
Kansai University, Faculty of Engineering Science, Department of Electrical and Electronic Engineering,564-8680,Japan

a) ryoutafunphy@gmail.com

b) ebara@kansai-u.ac.jp

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) に定式化できる．

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

## 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)[22] 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，コンサルタント及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 (2) 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．

$$v_n = \begin{cases} \arg \min_{i \in V_R} C(v_c, i) & (\text{random}() \leq a_0) \\ \arg \text{prob} C(v_c, i)^{-\beta} & (\text{otherwise}) \end{cases} \quad (2)$$

*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 *Rep* と *Pre* を基に確率的に選び，クライアントの立場で助言を求め解を式 (3) で構築する．

$$v_n = \begin{cases} \text{advice}(c, v_c, V_R) & (\exists c \wedge \text{random}() \leq q_0) \\ \arg \min_{i \in V_R} C(v_c, i) & (\text{random}() \leq b_0) \\ \arg \text{prob} C(v_c, i)^{-\beta} & (\text{otherwise}) \end{cases} \quad (3)$$

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* はイテレーション毎に減衰し，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 (4)，(5)，(6) 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時に得られる順位である．

$$\text{FadingRate} = r_0 \left( 1 + \frac{s}{\sqrt{1 + (\frac{s}{f})^2}} \right) \quad (4)$$

$$s = \sum_{i \in \{x | x \in P, \text{Mode}(x) = \text{Normal}, \text{Rank}(x) \leq R_F\}} \text{success}(i, w) \quad (5)$$

$$f = \left( \frac{1}{r_0} - 1 \right) \left( 1 - \frac{1}{w \sqrt{k_w}} \right) \quad (6)$$

## 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 (7) でフェロモン  $\tau$  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，弱体化される．

$$v_n = \arg \text{prob}_{i \in V_R} \tau(v_c, i)^\alpha C(v_c, i)^{-\beta} \quad (7)$$

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体化させる．

## 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)[21] 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 (8) 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．

$$v_n = \begin{cases} \arg \max_{i \in V_R} \tau(v_c, i)^\alpha C(v_c, i)^{-\beta} & (\text{random}() \leq q_0) \\ \arg \text{prob}_{i \in V_R} \tau(v_c, i)^\alpha C(v_c, i)^{-\beta} & (\text{otherwise}) \end{cases} \quad (8)$$

## 3. 提案手法

Mostafa Mahi らの手法 [12] は ACO の各個体に保持されているパラメータである  $\text{param}(i, \alpha, \beta)$  を PSO の個体の位置  $x_i = \text{param}(i, \alpha, \beta)$  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  $x_{i, \text{best}}, x_{\text{best}}$  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  $x_{i, \text{best}}$  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  $x_{\text{best}}$  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．

一般的な 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を追加する手法を説明する．戦略及び解を構築し，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  $x_{i,best}$  として保持され，最良解を上回る解を算出した場合は  $x_{best}$  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．式 (9) に  $x_i = param(i)$  を代入し，1 回だけ計算を行うことでパラメータを更新する．また，式 (9) 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．

$$\Delta^2 x_i = c_1 r_1 (x_{i,best} - x_i) + c_2 r_2 (x_{best} - x_i) \quad (9)$$

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 と ACS-PSO と呼ぶことにする．式 (10)，(11) 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．

$$param_{CGS}(i, a_0, \beta, \alpha, \gamma, q_0, b_0) \quad (10)$$

$$param_{ACS}(i, \alpha, \beta, q_0) \quad (11)$$

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．式 (10) が示す CGS パラメータについて説明する． $a_0$  が *Sabbatical* 状態における最近傍選択率， $\beta$  が確率的近傍選択率パラメータ， $\alpha, \gamma$  がコンサルタント選択率における *Rep, Pre* 依存度の重み， $q_0, b_0$  がそれぞれ *Normal* におけるコンサルタント利用率と最近傍選択率を表す．式 (11) が示す ACS パラメータは， $\alpha, \beta$  がそれぞれフェロモン・都市の近さ依存度の重み， $q_0$  が前述の確率が最大になる都市を選択する確率である．

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．

#### 4. 計算機実験

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[20] である．

表 1 実験環境

Table 1 the environment

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

表 2 CGS(-PSO) パラメータ

Table 2 parameters of CGS(-PSO)

項目	値	説明
$R_T$	1	<i>Sabbatical</i> に遷移する <i>Rep</i> の閾値
$Rep_{max}$	40	<i>Rep</i> の最大値
$Bonus$	8	コンサルタント活動成功時に加算される <i>Rep</i>
$k_w$	3	<i>FadingRate</i> を算出するために必要な定数
$r_0$	0.000003	<i>Rep</i> の減衰率
$\alpha$	7	コンサルタント選択率の <i>Rep</i> の重み
$\gamma$	7	コンサルタント選択率の <i>Pre</i> の重み
$q_0$	0.9	コンサルタント活用率
$b_0$	0.98	<i>Normal</i> 時の最近傍利用率
$\beta$	12	確率的最近傍を制御する変数
$a_0$	0.9	<i>Sabbatical</i> 時の最近傍利用率
$C_s$	100	<i>Sabbatical</i> 期間の長さ
$R_d$	6	<i>Rep</i> の初期値
$R_F$	3	<i>FadingRank</i>
$w$	1000	<i>FadingRate</i> を算出するために必要な定数
$c_1$	0.0002	PSO 弾性係数 (各個体最良解)
$c_2$	0.005	PSO 弾性係数 (最良解)
$min\ random$	0.9	パラメータ初期値の最小倍率
$max\ random$	2	パラメータ初期値の最大倍率
$No. Person$	10	個体数

実験環境を表 1 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み合わせで計算機実験を行う．その際，チューニング対象を表す変数を式 (12) のように 2 進数で定義する．

$$Tuning_{PSO,CGS}(a_0, \beta, \alpha, \gamma, q_0, b_0) = 000000 \quad (12)$$

式 (12) 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．

使用した CGS パラメータを表 2 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 [ $min\ random, max\ random$ ] の範囲の乱数をかけて初期値とした．使用した問題例は d493,d657,pr1002 であり，実行時間はそれぞれ 500s,600s,800s とした．実験結果を平均誤差率でソートしたグラフ (d657) を図 1 に示す．上位 3 位の  $Tuning_{PSO,CGS}$  の論理和を取ると 111111 となり，全てのパラメータを 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．このことから，SI アルゴリズムに

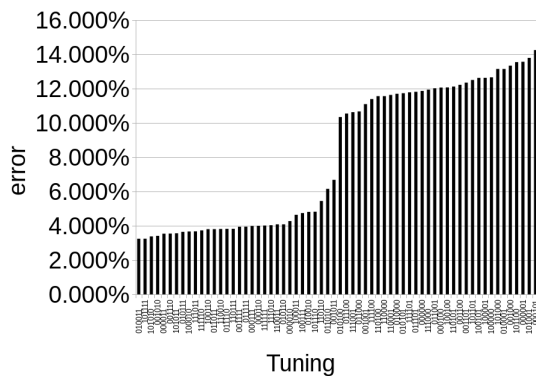


図 1 実験結果：d657- $Tuning_{PSO,CGS}$  特性  
Fig. 1 results:d657- $Tuning_{PSO,CGS}$  characteristics

において戦略構築時に参照される全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応できると予測できる。ゆえに、ACS, ACO も同様に全てのパラメータをチューニング対象とする。また、パラメータ  $q_0$  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

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 2 に示したもから、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。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 (d657) にしたものを図 2 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、 $q_0$  と  $\beta$  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、 $q_0, \beta$  についてパラメータチューニングを行うと、CGS は高性能化することが明らかになった。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わかった。

#### 4.3 固定パラメータの特性

4.2 節より、CGS-PSO は  $q_0, \beta$  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  $q_0, \beta$  の値が表 2 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。多くの  $q_0, \beta$  を CGS に与え得られた最小誤差率と CGS-PSO と比較することで、CGS-PSO の最適化能力を確認する。実験結果を表 3 に示す。 $\beta$  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

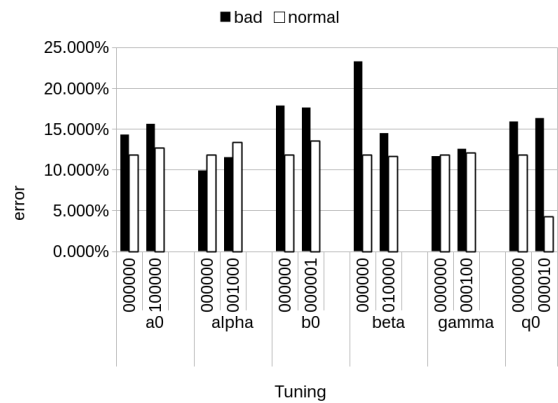


図 2 実験結果：d657- $Tuning_{PSO,CGS,Bad}$  特性  
Fig. 2 results:d657- $Tuning_{PSO,CGS,Bad}$  characteristics

表 3 実験結果：最適 ( $q_0, \beta$ )-CGS  
Table 3 results:the CGS gave optimal ( $q_0, \beta$ )

問題例	手法	パラメータ	誤差率 [%]
d493	CGS	$(q_0, \beta) = (0.9, 30)$	<b>3.157</b>
d493	CGSPSO	Tuning=010010	<b>2.370</b>
d657	CGS	$(q_0, \beta) = (0.9, 20)$	<b>8.080</b>
d657	CGSPSO	Tuning=010010	<b>4.809</b>
pr1002	CGS	$(q_0, \beta) = (0.9, 50)$	<b>14.295</b>
pr1002	CGSPSO	Tuning=010010	<b>9.422</b>

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。全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無く、PSO 適応パラメータチューニングの強力が示された。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。各個体のパラメータが振動することは各個体が構築する戦略及び解に大きな多様性を絶えず与え続けることができるため、どの時点においても広域探索性能は下がらないと考えられる。

#### 4.4 適応パラメータチューニングの性能

本節では、ACO, ACS, CGS のアルゴリズムに PSO 適応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比較する。なお、実験は無作為に選択した問題例 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を各 10 回の計算機実験を行い平均値を計測する。実行

表 4 ACO,ACS(-PSO) パラメータ  
Table 4 parameters of ACO,ACS(-PSO)

項目	ACS	ACO	説明
$\alpha$	1	1	フェロモン依存度
$\beta$	2	5	近傍依存度
$Q$	0.1	1	散布フェロモン係数
$q_0$	0.9		最大採確率戦略利用率
$\rho$	0.1	0.3	フェロモン減衰率
$\tau_{max}$	8	18	フェロモンの最大値
$\tau_{min}$	0.0001	1	フェロモンの最小値
$\tau_{init}$	0.005	10	フェロモンの初期値
$c_1$	0.0002	0.0002	PSO 弾性係数 (各個体最良解)
$c_2$	0.005	0.005	PSO 弾性係数 (最良解)
$\alpha_{max}$	8	8	$\alpha$ パラメータ初期値の最小値
$\alpha_{min}$	-8	-8	$\alpha$ パラメータ初期値の最大値
$\beta_{max}$	8	8	$\beta$ パラメータ初期値の最小値
$\beta_{min}$	-8	-8	$\beta$ パラメータ初期値の最大値
$q_{0,min}$	0		$q_0$ パラメータ初期値の最小値
$q_{0,max}$	1		$q_0$ パラメータ初期値の最大値
$No.Ant(ACS)$	min(n,200)	左と同じ	個体数
$No.Ant(ACSPSO)$	10	10	ACS-PSO アルゴリズムの個体数

表 5 実験結果: PSO 適応パラメータチューニング  
Table 5 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	<b>634.9</b>	<b>0.938</b>	<b>629</b>	<b>0.000</b>
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	<b>15911.3</b>	<b>0.832</b>	15868	0.558
d198	60	CGSPSO	15918.9	0.880	<b>15834</b>	<b>0.342</b>
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	<b>2612.7</b>	<b>1.307</b>	2588	0.349
a280	120	CGSPSO	2613.8	1.349	<b>2585</b>	<b>0.233</b>
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	<b>50300.9</b>	<b>2.840</b>	<b>50014</b>	<b>2.253</b>
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	<b>9022.1</b>	<b>2.454</b>	<b>8963</b>	<b>1.783</b>
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	<b>268531.7</b>	<b>3.662</b>	<b>266404</b>	<b>2.841</b>
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	<b>159444.9</b>	<b>4.233</b>	<b>158610</b>	<b>3.687</b>
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	<b>82081.6</b>	<b>2.028</b>	<b>81750</b>	<b>1.616</b>
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	<b>30410.2</b>	<b>5.694</b>	<b>29991</b>	<b>4.237</b>
r15934	45000	ACO	901594.0	62.144	892246	60.463
r15934	45000	ACOPSO	740475.4	33.168	735265	32.231
r15934	45000	ACS	1125774.1	102.461	1109475	99.530
r15934	45000	ACSPSO	663385.2	19.304	658124	18.358
r15934	45000	CGS	661922.3	19.041	659244	18.559
r15934	45000	CGSPSO	<b>599915.1</b>	<b>7.890</b>	<b>583906</b>	<b>5.011</b>

時間は ACO に合わせて長くし、実験結果の表 5 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 4 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に用いられる蟻数は参考文献 [12] によると、10 体が最適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。全ての実験結果を表 5 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極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挙げられる。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、それぞれで使用され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 [12] 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 6 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 [12] 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。

## 5. 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全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  $\beta$  と  $q_0$  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  $\beta, q_0$  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

表 6 実験結果：文献 [12] との比較  
Table 6 results:comparison with the reference[12]

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	<b>629</b>	<b>0.000</b>
eil101	30	CGSPSO	637.2	1.304	630	0.159
eil101	302.15	ACO-PSO-3opt[12]	<b>632.7</b>	<b>0.588</b>	631	0.318

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。

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全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す。さらに、静的なパラメータチューニングはいかなる最適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。

## 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] T. El-Ghazali, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009

[3] Liang Zhang, Ling Wang, Da-Zhong Zheng, An adaptive genetic algorithm with multiple operators for flowshop scheduling, Int J Adv Manuf Technol (2006) 27: 580-587, 2004

[4] Jing Tang, Meng Hiot Lim, Yew Soon Ong, Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems, Soft Comput (2007) 11:873-888, 2007

[5] Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, Kai Zhao, Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search, Applied Soft Computing 11 (2011) 3680-3689, 2011

[6] Weidong Xiao, William G. Dunford, A Modified Adaptive Hill Climbing MPPT Method for Photovoltaic Power Systems, 2004 35th Annual IEEE Power Electronics Specialists Conference, 2004

[7] Kenny Q. Zhu, A Diversity-controlling Adaptive Genetic Algorithm for the Vehicle Routing Problem with Time Windows, National University of Singapore, 2003

[8] Ouzhan Hasanebi, Ferhat Erdal, Mehmet Polat Saka, Adaptive Harmony Search Method for Structural Op-

timization, Journal of Structural Engineering Vol.136 No.4, 2010

[9] Teruyoshi Yamaguchi, Keiichiro Yasuda, Adaptive Particle Swarm Optimization 2006 IEEE International Conference on Systems, Man, and Cybernetics, 2006

[10] Nobuhiro Iwasaki, Keiichiro Yasuda Genki Ueno, Dynamic Parameter Tuning of Particle Swarm Optimization TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING IEEJ Trans 2006; 1: 35-363, 2006

[11] Warren G. Jackson, Ender Ozcan, Robert I. John, Fuzzy Adaptive Parameter Control of a Late Acceptance Hyper-heuristic ASAP Research Group School of Computer Science University of Nottingham Jubilee Campus Wollaton Road Nottingham, NG8 1BB, UK, 2014

[12] Mostafa Mahi, Omer Kaan Baykan, Halife Kodaz, A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem, Applied Soft Computing 30 (2015) 484-490, 2015

[13] Vijay Kumar, Jitender Kumar Chhabra, Dinesh Kumar, Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems, Journal of Computational Science 5 (2014) 144-155, 2013

[14] Taher Niknam, Rasoul Azizipanah-Abarghooee, and Alireza Roosta, Reserve Constrained Dynamic Economic Dispatch: A New Fast Self-Adaptive Modified Firefly Algorithm, IEEE SYSTEMS JOURNAL, VOL. 6, NO. 4, DECEMBER 2012, 2012

[15] Jose Luis Ponz-Tienda, Vctor Yepes, Eugenio Pellicer, Joaquin Moreno-Flores, The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm, Automation in Construction 29 (2013) 161-172, 2013

[16] Gang Xu, An adaptive parameter tuning of particle swarm optimization algorithm, Applied Mathematics and Computation 219 (2013) 4560-4569, 2013

[17] Frumen Olivas, Fevrier Valdez, Oscar Castillo, 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

[18] Michael A. LEE, Hideyuki TAKAGI, Dynamic Control of Genetic Algorithms using Fuzzy Logic Techniques, the Proceeding of 5th Int'l Conf. on Genetic Algorithms (ICGA'93), Urbana-Champaign, IL, pp.76-83, 1993

[19] Afef Ben Hadj Alaya-Feki, Berna Sayrac, Eric Moulines, Semi dynamic parameter tuning for optimized opportunistic spectrum access, 978-1-4244-1722-3/08/ 2008 IEEE, 2008

[20] TSPLIB, <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>, (accessed 2015-8-29)

[21] Dorigo, M., Gambardella, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem, Evolutionary Computation, IEEE Transactions on, vol.1, pp.53-66 1997

[22] Serban Iordache, Consultant-Guided Search - A New Metaheuristic for Combinatorial Optimization Problems, GECCO '10, July 7-11, 2010, Portland, Oregon, USA, 2010