

粒子群最適化パラメータチューニングを用いた 高速ハイブリッド手法

長辻 亮太^{1,a)} 裕 一稀^{1,b)} 榎原 博之^{2,c)}

受付日 2016年12月22日, 採録日 2017年7月4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 Particle Swarm Optimization (PSO) を Consultant Guided Search (CGS) の適応パラメータチューニングに用いた, CGS-PSO ハイブリッド手法は強力な最適化能力を示した. 本研究では, CGS-PSO ハイブリッド手法をさらにエンハンスメントするために, プロセス間でパラメータを共有させる並列化手法を提案する. TSPLIB の問題例に対するベンチマークにより, 収束速度と誤差率について性能評価を行う. その結果, 提案手法は収束速度の観点から優れた手法であることを示す.

キーワード: コンサルタント誘導型探索, 粒子群最適化, リン・カーニハン法, 巡回セールスマン問題, 適応パラメータチューニング, ハイブリッド並列手法

Fast Hybrid Method with Particle Swarm Optimization Parameter Tuning

RYOUTA NAGATSUJI^{1,a)} KAZUKI HAZAMA^{1,b)} HIROYUKI EBARA^{2,c)}

Received: December 22, 2016, Accepted: July 4, 2017

Abstract: Meta-heuristics is available for the combinatorial optimization problem and its ability has been developed. Especially, the Consultant Guided Search–Particle Swarm Optimization (CGS-PSO) hybrid method has showed powerful performance. CGS-PSO hybrid method tunes parameters of CGS algorithm with PSO algorithm adaptively. In this study, in order to enhance the CGS-PSO hybrid method, we propose the new parallel parameter tuning method with sharing parameters between processes. We benchmark the proposed method in the convergence speed and the error rate for instances of TSPLIB. As a result, we show the proposed method is superior in terms of the convergence speed.

Keywords: Consultant Guided Search (CGS), Particle Swarm Optimization (PSO), Lin-Kernighan (LK), Traveling Salesman Problem (TSP), adaptive parameter tuning, hybrid parallel method

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

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

a) ryoutafunphy@gmail.com

b) sukima2353@gmail.com

c) ebara@kansai-u.ac.jp

1. はじめに

近年, 力学や空力, 金融, 運輸などの分野において, 組合せ最適化問題を計算機を用いて解く需要が高まっている. 問題を計算機を用いて最適化するには, 最適化問題への定式化が必要である. たとえば, スケジュール最適化を一般化した Job-shop Scheduling Problem (JSP) [2], 配置とフロー最適化を一般化した Quadratic Assignment Problem (QAP) [3] などが組合せ最適化問題として存在する.

組合せ最適化問題には, NP 困難な問題が多い. NP 困

難な問題は、問題例のサイズの多項式時間で解を与えるアルゴリズムを見つけることが難しい。ゆえに、近似解法としてのメタヒューリスティクスがさかんに研究されている。Simulated Annealing (SA) [4] アルゴリズムや Swarm Intelligence (SI) に分類されるものとして Ant Colony Optimization (ACO) [5] や Genetic Algorithm (GA) [6], Consultant Guided Search (CGS) [7] などがメタヒューリスティクスに含まれる。近似解法には最適解が必要でないケースで高速に求解できるという特徴があり、NP 困難な組合せ最適化問題に対しても有効である。

メタヒューリスティクスでは、最良解の改善や高速化などのエンハンスメントを施すためにハイブリッド化や並列化などが用いられる。ハイブリッド手法は複数のアルゴリズムを組み合わせ、既存のメタヒューリスティクスをエンハンスメントする手法である。たとえば、ACO の各個体が保持しているパラメータを Particle Swarm Optimization (PSO) [8] の粒子に対応させ、パラメータ粒子をパラメータ空間で飛翔させることで ACO の探索能力を最適化する ACO-PSO-3opt [9] などが存在する。この手法は、ACO の局所解に陥りやすいという問題点を PSO を用いて問題例ごとに適応パラメータチューニングさせることで対応し、最後に 3opt 局所探索を適用することでエンハンスメントしている。このように、互いの弱点を補い合いより強力な戦略を構築するために、ハイブリッド手法が研究されている。

Kurdi の研究 [10] は GA アルゴリズムをエンハンスメントするために、Tabu Search (TS) による局所探索を適用し、Island Model による並列化を施している。この手法は、Hybrid Island Model Genetic Algorithm (HIMG) と呼ばれ、文献 [10] で Job-shop Scheduling Problem (JSP) を用いて過去に提案された良いアルゴリズムと比較された。HIMG は過去に提案された良いアルゴリズムに対して、最適解発見率の観点から効果的な結果を示した。Shi らの研究 [11] は、SI アルゴリズムに属する PSO と GA を組み合わせたハイブリッド手法を提案している。PSO と GA は、それぞれの個体が解を生成する手続きを複数回数繰り返す。解を生成する際、PSO と GA は独立しており、互いに依存性がない。その後、PSO の個体と GA の個体が持つ解をそれぞれ交換し、再度探索を繰り返し行う。この研究 [11] は、一般的な PSO アルゴリズムに対してより効果的に最適化できることが示された。これらのほかにも、優れたメタヒューリスティクス手法は数多く存在する。たとえば、猫の群れる習性をモデルとした Cat Swarm Optimization [12]、雑草が生い茂る習性をモデルとした Invasive Weed Optimization [13] などがある。

本研究では、メタヒューリスティクスのさらなる高性能化を目的とし、ハイブリッド化と並列化を両方駆使した手法を提案する。具体的には、CGS の戦略パラメータを PSO を用いて、ACO-PSO-3opt [9] と同様に適応パラメータ

チューニングを行い、高速に動作する 2opt を CGS-PSO の個体である各 Virtual Person の戦略に適用する並列手法を構築し、Traveling Salesman Problem (TSP) でベンチマークを行う。最後に Lin-Kernighan (LK) 法 [14] による局所探索を適用することで最良解を改善する。短時間で精度の良い近似解を求めることが本研究の目的である。

2. Traveling Salesman Problem と既存のハイブリッド手法

Traveling Salesman Problem (TSP) は、都市と呼ばれる頂点の集合と任意の 2 都市間に与えられた移動コストがグラフとして与えられたときに、すべての都市を一巡し、開始地点の都市に戻ってくる巡回路の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。以下のように定式化することができる。

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

$$x_i \in V \quad (i \in [0, n-1])$$

x は求める巡回路であり、 C は 2 都市間の移動コスト行列、 V は都市の集合、 n は都市数を表している。また、TSP の問題例は TSPLIB [15] より配布されており、ここから問題例を入手する。

また、2opt とは巡回路の中から任意に 2 本の枝を選び、巡回路となる枝交換で解が良くなる場合に交換する局所探索法である。3opt は、巡回路の中から任意に 3 本の枝を選び、巡回路となる枝交換 (複数存在する) で解が良くなる場合に交換する局所探索法である。いずれの方法もこれ以上交換しても解が良くならなくなった場合に終了する。

2.1 ACO-PSO-3opt ハイブリッド手法

Mahi ら [9] は ACO アルゴリズムをエンハンスメントするために、PSO 適応パラメータチューニングを施し、最良解に 3opt 局所探索を適用するハイブリッド手法を研究している。ハイブリッド化手法は ACO の各個体が保持しているパラメータを PSO の粒子の位置に対応付け、パラメータ空間を粒子が飛翔することにより、動的に ACO の最適化能力を最適化する。そして、最後に一度最良解に 3opt 局所探索を適用する。文献 [9] では、このアルゴリズムを TSP でベンチマークしている。都市数の小さい問題例でしかベンチマークされていなかったが、ACO に対してきわめて高い最適化能力と収束速度を示した。

ACO の戦略は式 (2) によって、次の訪問先の都市が決定される。

$$v_n = \arg \operatorname{prob} \tau(v_c, i)^{\alpha_{aco}} C(v_c, i)^{-\beta_{aco}} \quad (2)$$

$$i \in V_R$$

式 (2) の v_n , v_c はそれぞれ次の訪問都市、現在の都市、 τ は 2 都市間に設置されたフェロモン量、 α_{aco} , β_{aco} は ACO

の個体が保持しているパラメータである。すべての個体が戦略の構築を終えた後、解の更新を行い、その評価値を用いて PSO による適応パラメータチューニングを行う。式 (3) により、PSO パラメータ空間に ACO の個体が保持しているパラメータが対応付けられる。

$$z_{aco,i} = (\alpha_{aco,i}, \beta_{aco,i}) \tag{3}$$

そして、式 (4) により、パラメータチューニングが行われる。

$$\begin{aligned} \Delta^2 z_{aco,i} = & r_1 c_1 (z_{aco,i,best} - z_{aco,i}) \\ & + r_2 c_2 (z_{aco,best} - z_{aco,i}) \end{aligned} \tag{4}$$

式 (4) の $z_{aco,i}$ は ACO の個体 i が保持しているパラメータ、 $z_{aco,i,best}$ は個体 i の自身の最良解に対応するパラメータ、 $z_{aco,best}$ はシステム全体での最良解に対応するパラメータである。そして、終了条件を満たすまで ACO-PSO を実行し、最後に 1 度 3opt 局所探索を行う。

2.2 CGS-PSO ハイブリッド手法

Consultant Guided Search-Particle Swarm Optimization (CGS-PSO) は、著者らが文献 [1] で PSO を用いた適応パラメータチューニングの効果を詳しく検証するために開発したハイブリッドアルゴリズムである。CGS-PSO は、TSP を用いてベンチマークされ、CGS-TSP [7] よりもはるかに高性能であることが示された。その最適化能力は、CGS の重要なパラメータである q_0 、 β の多くの組合せを実行し、CGS の最も良い性能がでるパラメータ設定を上回った。特に、TSPLIB [15] の大きな都市数の問題例に対して有効に働いた。

CGS-PSO は、CGS の各個体が保持しているパラメータを PSO で適応パラメータチューニングを行うものであり、ACO-PSO-3opt [9] と同様にパラメータチューニングを行う。式 (5) により、CGS の各個体に保持されているパラメータが PSO の個体位置に対応づけられる。

$$z_i = (a_{0,i}, \beta_i, \alpha_i, \gamma_i, q_{0,i}, b_{0,i}) \tag{5}$$

式 (5) の右辺は、CGS の個体 i が保持しているパラメータをベクトル化したものである。

CGS-PSO は高い最適化能力を示したが、CGS-TSP のパラメータ空間が 6 次元であるため良好なパラメータの発見までに時間がかかり、収束速度に若干のばらつきがみられた。本研究では、CGS-PSO の弱点であった良好なパラメータ空間の発見に時間がかかる点を解決するための並列化手法を提案する。

3. 提案手法

CGS-PSO をさらに高速化するための並列手法を提案する。CGS-PSO は、文献 [1] で比較的高速な収束速度である

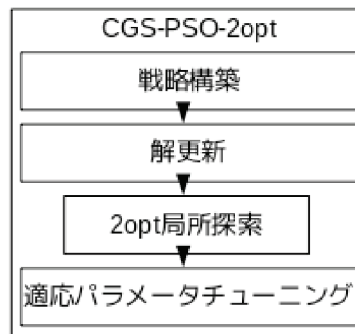


図 1 CGS-PSO-2opt ハイブリッド手法
Fig. 1 CGS-PSO-2opt hybrid method.

ことが示されている。しかしながら、良好なパラメータ空間を発見するまでに時間がかかっている。したがって、本論文では良好なパラメータ空間を並列処理で早く見つけることで、CGS-PSO の本来の性能を引き出す方法を考える。

提案する並列処理は、個体の集合を実行する CGS のプロセスを複数個並列に実行する方法である。本提案手法では、複数のプロセスを用いて並列に良好なパラメータを探索する。

各プロセスに CGS-PSO を設置する。この個体は CGS-PSO [1] と同じものである。そして、この個体には収束速度を速めるために適応パラメータチューニングの直前にすべての個体の解に対して、2opt 局所探索法が適用される。この手法を CGS-PSO-2opt ハイブリッド手法^{*1}と呼ぶ。CGS-PSO-2opt ハイブリッド手法は、適応パラメータチューニングのタイミングが図 1 のタイミングとなる。このタイミングで適応パラメータチューニングを行うことにより、2opt 局所探索に適した解を CGS-PSO が算出すると考えられる。本提案手法は、良好なパラメータのみをプロセス間で共有する。共有タイミングはプロセス上で最良解が更新された際、それに対応するパラメータをランダムに選んだ他の 1 つのプロセスへと複製させる。また、この複製するパラメータを Q と呼ぶ。パラメータベクトル Q は、式 (5) に示されるように、6 つのパラメータをベクトル化したものである。最後に、最良解に対して LK 法 [14] を用いた局所探索を行う。パラメータベクトル Q と LK 局所探索を加えた手法を提案手法 Q-CGS-PSO-2opt-LK と呼ぶ。パラメータベクトル Q の複製と LK 局所探索のタイミングを図 2 に示す。また、このパラメータベクトル Q は式 (6) のように適応パラメータチューニングの戦略として使用される。

$$\begin{aligned} \Delta^2 z_i = & r_1 c_1 (z_{i,best} - z_i) + r_2 c_2 (z_{best} - z_i) \\ & + r_3 c_3 (Q - z_i) \end{aligned} \tag{6}$$

Q-CGS-PSO-2opt-LK ハイブリッド並列手法は並列に良好なパラメータを探索することによって、CGS-PSO 本

^{*1} CGS-PSO [1] の Virtual Person の戦略に 2opt を適用するもの

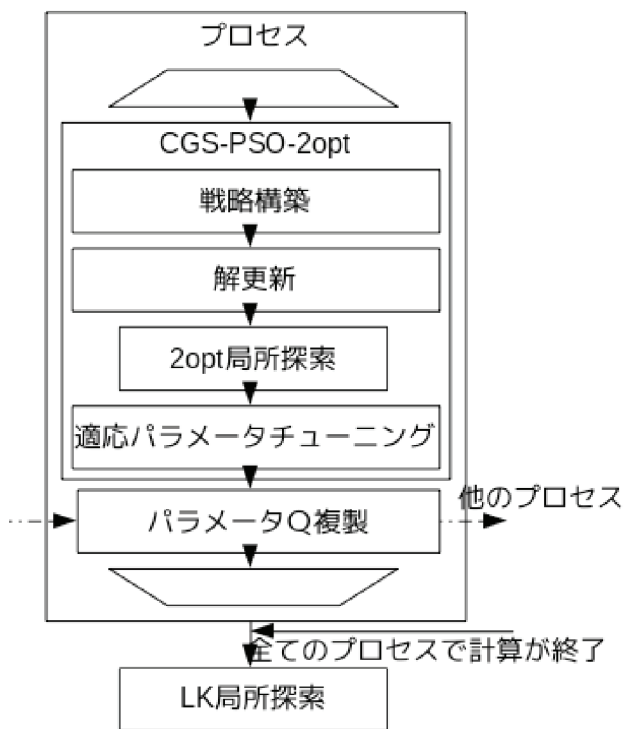


図2 Q-CGS-PSO-2opt-LK ハイブリッド並列手法
Fig. 2 Q-CGS-PSO-2opt-LK hybrid parallel method.

来の性能を引き出しつつ並列に探索することで高速な収束性能が期待できる。さらに、LK 局所探索は長大な実行時間がかかるが、本提案手法は LK 局所探索にかかる時間も短縮できると考えられる。なぜなら、並列処理である Q-CGS-PSO-2opt ハイブリッド手法*2を用いることで、LK 法の局所解に近い解を高速に求めることができるからである。それにより、LK 法は局所解に到達するために少しの計算量で済み、短時間で終了すると考えられる。

パラメータベクトル Q の複製は計算の高速化のために非同期で行われる。図3を用いて、非同期複製について説明する。各プロセスにはスレッド main と sub が独立して実行されている。スレッド main では Q-CGS-PSO-2opt が実行されており、スレッド sub ではパラメータベクトル Q の複製待ちを行っている。スレッド sub で複製が完了した際、パラメータベクトル Q を更新する手続きを生成し、それをスレッド main で実行させることで複製中も計算を停止させずに、Q-CGS-PSO-2opt を実行させることができる。提案手法 Q-CGS-PSO-2opt-LK は、Q-CGS-PSO-2opt ハイブリッド並列手法が算出した最良解に対して LK 局所探索をすでに計算が終了している別の1つのプロセスで行う。Q-CGS-PSO-2opt ハイブリッド並列手法がすべて終了した際に LK 局所探索を実行するので、計算機リソース上ではプロセスの追加は必要ない。

*2 提案手法において、最後に LK 法を用いた局所探索を行わないものを Q-CGS-PSO-2opt と呼ぶ。

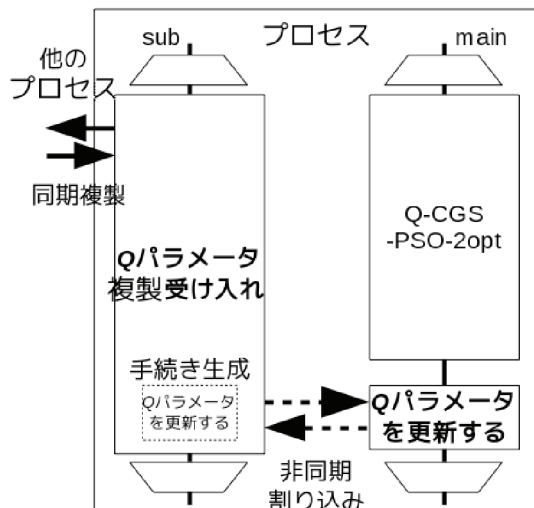


図3 パラメータベクトル Q 非同期複製方法
Fig. 3 The method that the parameter Q sharing asynchronously.

表1 計算機環境
Table 1 Computational environments.

項目	値
OS	CentOS 6.5
CPU	Intel-Xeon E5-2620v2 x2
クロック数	2.1 GHz
スレッド数	24
チップセット	C600
メモリ	DDR3-1600 ECC registered 16 GBx8 : 128 GB
MPI	Intel MPI Library for Linux, 5.0 Update 2
コンパイラ	icpc (ICC) 15.0.1 20141023

4. 実験結果

本実験では、提案手法の性能をベンチマークするために TSP LIB [15] の問題例を利用し、同等のリソースを投入した並列ランダム多スタート LK 法と比較する。ベンチマークを行う計算機環境を表1に示す。提案手法のパラメータを表2に示す。表2の各パラメータは文献[1]を参考に決定し、プロセス数 M は計算機に投入できるリソースを参考に決定する。プロセス数 M を変化(減少)させた場合の結果は付録 A.1 に掲載する。また、パラメータベクトル Q の複製先はランダムとし、複製はそのプロセスで最良解の更新が起きたときに行う。パラメータベクトル Q の共有回数と改善回数については付録 A.2 に掲載する。計算機実験は、1つの実験項目につき10回試行し実験結果を集計する。誤差率は一般的な定義である式(7)により定める。式(7)の $f(x_{opt})$ は、TSP LIB によって提供される最適解の評価値である。また、表3に提案手法の Q-CGS-PSO-2opt ハイブリッド並列手法の実行時間を示す。表3に示す Q-CGS-PSO-2opt ハイブリッド並列手法

表 2 提案手法のパラメータ
Table 2 Parameters of the proposed method.

パラメータ	説明	値
R_T	Sabbatical 移行評判閾値	1
R_{max}	評判の最大値	40
R_B	コンサルタント評判加算値	8
k_w	評判制御パラメータ	3
r_0	評判減衰率	3.0×10^{-7}
α	コンサルタント選択 α パラメータ	7
γ	コンサルタント選択 γ パラメータ	7
q_0	アドバイス組込み率	0.9
b_0	Normal モード最近傍都市組込み率	0.98
β	近傍都市選択 β パラメータ	12
a_0	Sabbatical モード最近傍都市組込み率	0.9
C_S	Sabbatical 繰り返し数	100
R_I	評判の初期値	6
N_F	Fading Rank	3
w	評判制御パラメータ	1,000
c_1	自己ベストスチフネス	0.0002
c_2	プロセス最良スチフネス	0.003
c_3	システム最良スチフネス	0.006
η	速度抵抗係数	0
K_{min}	パラメータ初期位置最小値係数	0.5
K_{max}	パラメータ初期位置最大値係数	2
$K_{z,max}$	パラメータ位置最大値係数	3
V_{max}	パラメータ初期速度最大係数	0.002
N	1 プロセスあたりの個体数	10
M	プロセス数	20

の実行時間は、予備実験において Q-CGS-PSO-2opt ハイブリッド並列手法の収束の具合から決定した。収束の具合は付録 A.3 を根拠としている。

$$\eta = \frac{f(x) - f(x_{opt})}{f(x_{opt})} \quad (7)$$

4.1 閾値到達時間による速度評価

本節では、提案手法の収束速度を評価するために誤差率に閾値を設け、閾値よりも等しいまたは下回った際の時間を計測する。これにより、所定の誤差率を持つ解を得るために必要な計算時間の平均値と提案手法の収束速度が分かる。表 4 に各閾値と時間を示す。表 4 の列名は、提案平均が提案手法の平均値、LK 平均が並列ランダム多スタート LK 法の平均値、提案最小が提案手法の最小値、LK 最小が並列ランダム多スタート LK 法の最小値に対応する。計測の都合上 1 秒を切った場合、0 秒に換算されることと並列ランダム多スタート LK 法が計算機資源不足により、問題例 rl11849 と usa13509, brd14051 で実行できなかった。また、誤差率に閾値を設けて時間を計測すると、Q-CGS-PSO-2opt ハイブリッド並列手法の部分でその誤差率に到達してしまうこともある。

実験結果より、多くの問題例で提案手法が速いことが判

表 3 CGS の実行時間
Table 3 Running time of CGS.

問題例	実行時間 [s]
dsj1000	5
pr1002	5
si1032	5
u1060	5
vm1084	5
pcb1173	5
d1291	5
rl1304	5
rl1323	5
nrv1379	5
fl1400	10
u1432	10
fl1577	10
d1655	10
vm1748	10
u1817	10
rl1889	10
d2103	10
u2152	10
u2319	10
pr2392	30
pcb3038	30
fl3795	60
fn14461	60
rl5915	260
rl5934	260
pla7397	300
rl11849	800
usa13509	5,000
brd14051	5,000

明した。誤差率 5%において、提案手法が並列ランダム多スタート LK 法よりも早く収束している。その傾向は、誤差率 8%においてもあてはまり、ほとんどの問題例と誤差率閾値の組合せにおいて提案手法が高速に動作することを示している。特に、問題例 d2103 の閾値 5%は提案手法が最も LK 法と比較して高速に動作しており平均値において、その到達時間の比をとると 18.9 倍である。誤差率 5%については LK 法以上に高速に動作し、数個の問題例で LK 法より若干遅い結果となった。誤差率 10%については LK 法との差が縮まり、収束速度の観点から提案手法に近づいている。しかしながら、誤差率 5%の比較的良い解を高速で求めることができる提案手法 Q-CGS-PSO-2opt-LK の方が収束速度が速いことは明らかである。ゆえに、提案手法は並列ランダム多スタート LK 法よりも早く良い解を見つけることができる。その傾向は最小値にもあてはまり、5%, 8%において提案手法が LK 法よりも早く収束している。誤差率 5%の問題例 d2103 の最小値において、14.5 倍高速に収束されていることが判明した。平均値と同様に誤

表 4 実験結果：閾値 5%, 8%, 10%到達時間 [s]
Table 4 Results: reach times in 5%, 8%, 10%.

問題例/閾値	提案平均	LK 平均	提案最小	LK 最小	提案平均	LK 平均	提案最小	LK 最小	提案平均	LK 平均	提案最小	LK 最小
	5%	5%	5%	5%	8%	8%	8%	8%	10%	10%	10%	10%
dsj1000	7.9	7.0	7.0	5.0	1.7	2.2	1.0	1.0	1.5	1.1	1.0	0.0
pr1002	6.7	8.9	6.0	6.0	1.5	2.5	1.0	1.0	1.5	1.2	1.0	1.0
si1032	0.2	0.9	0.0	0.0	0.2	0.9	0.0	0.0	0.2	0.9	0.0	0.0
u1060	7.0	7.8	7.0	6.0	2.1	2.5	1.0	1.0	2.1	1.2	1.0	0.0
vm1084	6.7	9.3	4.0	6.0	1.9	2.3	1.0	2.0	1.9	1.2	1.0	1.0
pcb1173	9.7	13.0	8.0	8.0	3.5	4.7	2.0	4.0	2.2	2.4	2.0	1.0
d1291	6.2	30.0	3.0	19.0	3.0	8.4	3.0	5.0	3.0	3.7	3.0	2.0
rl1304	10.0	20.4	8.0	15.0	3.0	5.4	2.0	4.0	2.5	2.1	2.0	1.0
rl1323	7.5	17.0	7.0	9.0	2.7	4.3	2.0	2.0	2.7	2.1	2.0	1.0
nrv1379	8.5	12.5	7.0	10.0	3.2	3.9	3.0	3.0	3.2	1.9	3.0	1.0
fl1400	5.7	4.3	5.0	3.0	5.5	2.3	5.0	2.0	5.5	2.1	5.0	2.0
u1432	13.6	11.2	13.0	7.0	3.4	4.6	3.0	4.0	3.3	3.1	3.0	2.0
fl1577	6.1	22.6	5.0	8.0	5.7	5.1	5.0	2.0	5.7	3.3	5.0	2.0
d1655	15.5	50.8	13.0	34.0	5.0	15.4	5.0	11.0	5.0	8.1	5.0	4.0
vm1748	18.5	44.0	15.0	27.0	5.4	11.3	5.0	5.0	5.4	4.1	5.0	3.0
u1817	20.6	121.1	14.0	92.0	5.8	37.6	5.0	30.0	5.8	20.6	5.0	17.0
rl1889	30.1	76.3	23.0	59.0	6.7	20.0	6.0	7.0	6.4	7.4	6.0	3.0
d2103	8.7	164.5	8.0	116.0	8.6	66.5	8.0	51.0	2.5	36.9	0.0	25.0
u2152	43.5	170.0	31.0	123.0	8.7	62.1	8.0	58.0	8.4	32.1	8.0	28.0
u2319	9.7	5.0	9.0	4.0	9.6	3.8	9.0	3.0	9.6	3.8	9.0	3.0
pr2392	57.4	160.5	50.0	91.0	12.7	56.0	11.0	39.0	11.8	25.0	11.0	20.0
pcb3038	103.0	242.0	74.0	202.0	20.4	85.4	18.0	69.0	19.0	39.7	18.0	29.0
fl3795	64.9	393.6	38.0	269.0	43.9	43.4	38.0	26.0	43.9	23.2	38.0	17.0
fl14461	187.3	492.7	159.0	419.0	45.5	143.7	43.0	103.0	45.5	46.5	43.0	30.0
rl5915	631.2	2,848.4	542.0	2,296.0	75.9	877.2	68.0	653.0	74.1	334.3	68.0	192.0
rl5934	818.5	2,892.7	543.0	2,432.0	78.3	982.4	74.0	711.0	76.5	495.0	68.0	336.0
pla7397	646.9	2,129.4	549.0	1,725.0	148.8	592.6	143.0	315.0	148.0	193.3	143.0	119.0
rl11849	4,647.5		3,533.0		376.5		356.0		375.5		356.0	
usa13509	8,909.9		7,859.0		619.0		528.0		601.0		528.0	
brd14051	7,722.0		7,352.0		570.4		550.0		570.4		550.0	

差率 10%については LK 法との差が縮まる結果となった。ゆえに、提案手法は平均値と最小値ともに LK 法よりも誤差率 5%の比較的良い解に対する収束が高速であることが示された。

4.2 総実行時間によるメタ戦略評価

本節では、総実行時間を評価することにより Q-CGS-PSO-2opt-LK ハイブリッド並列手法がどれだけ並列ランダム多スタート LK 法の時間を短縮できるのかを評価する。これにより、メタ戦略を用いる場合の局所探索にかかる時間の短縮具合を評価する。総実行時間とは提案手法と並列ランダム多スタート LK 法が終了するまでの時間である。提案手法は Q-CGS-PSO-2opt 並列ハイブリッド手法が開始されてから LK 局所探索が終了するまでの時間を計測する。並列ランダム多スタート LK 法は提案手法におけるプロセス数 20 個の初期解を局所探索する際、20 並列の LK 局所探索がすべて終了するまでの時間を計測する。表 5 に

総実行時間の平均値と最小値を示す。

総実行時間が並列ランダム多スタート LK 法と比較して提案手法が短いことが分かる。並列ランダム多スタート LK 法と比較すると、提案手法の総実行時間の増加が比較的緩やかであることが分かる。これらのことから、提案手法の Q-CGS-PSO-2opt ハイブリッド並列手法が LK 法を用いた局所探索の時間を削減できていると考えられる。特に、今回の実験では、計算機リソースの関係上並列ランダム多スタート LK 法を実行できなかった問題例 rl11849 と usa13509, brd14051 を除き提案手法は総実行時間を LK 法に対して低減することに成功している。さらに、問題例 d2103 において提案手法の総実行時間が並列ランダム多スタート LK 法に対して 8.9 倍高速に動作していることが分かる。4.1 節でも述べたように、問題例 d2103 は提案手法に有利な都市配置をしていると考えられる。しかしながら、有利な都市配列でないと考えられる問題例 u2319 においても 3.4 倍提案手法が速く動作している。ゆえに、提案手法

表 5 実験結果：総実行時間 [s]
Table 5 Results: total running time [s].

問題例	提案平均	LK 平均	提案最小	LK 最小
dsj1000	20.4	102.2	18.0	89.0
pr1002	20.0	104.8	15.0	92.0
si1032	11.5	33.0	10.0	29.0
u1060	22.5	119.0	16.0	104.0
vm1084	22.0	121.2	17.0	108.0
pcb1173	31.7	172.1	22.0	156.0
d1291	28.2	185.5	20.0	158.0
rl1304	30.1	206.6	23.0	177.0
rl1323	32.7	216.0	27.0	194.0
nrv1379	42.9	285.4	35.0	260.0
fl1400	49.9	256.4	36.0	232.0
u1432	45.3	216.8	35.0	191.0
fl1577	59.2	423.8	43.0	380.0
d1655	75.1	479.3	61.0	450.0
vm1748	95.9	634.1	72.0	539.0
u1817	83.5	626.7	70.0	557.0
rl1889	110.9	726.5	83.0	610.0
d2103	123.0	1,093.5	69.0	908.0
u2152	178.2	1,129.1	126.0	1,001.0
u2319	175.4	587.9	149.0	538.0
pr2392	306.8	1,778.8	242.0	1,484.0
pcb3038	666.5	4,014.9	567.0	3,616.0
fl3795	1,293.4	7,650.7	931.0	6,581.0
fnl4461	2,534.2	14,070.7	1,879.0	12,452.0
rl5915	4,286.3	29,271.5	3,588.0	26,789.0
rl5934	4,721.2	29,875.8	3,656.0	25,375.0
pla7397	10,661.5	63,921.4	7,128.0	52,494.0
rl11849	46,791.2		35,321.0	
usa13509	105,516.3		70,906.0	
brd14051	88,020.9		65,954.0	

Q-CGS-PSO-2opt-LK ハイブリッド並列手法のメタ戦略により、既存のアルゴリズムである CGS-PSO と LK 法の両者の性能を向上させることに成功しているといえる。

4.3 最終誤差率による精度評価

本節では、最終誤差率を比較することでアルゴリズムの精度を検証する。さらに、実験結果 4.2 節の結果と合わせることで速さと精度も考察する。表 6 に最終誤差率を示す。最終誤差率は並列ランダム多スタート LK 法に対して劣っていることが判明した。しかしながら、その差は小さい。並列ランダム多スタート LK 法に対する相対精度は式 (8) を用いて計算し、表 6 に示す。

$$\mu = \frac{\eta_p + 1}{\eta_{LK} + 1} \quad (8)$$

η_p は提案手法の誤差率を、 η_{LK} は並列ランダム多スタート LK 法の誤差率を表す。参考として、PSO 適応パラメータチューニングの効果を確認するために開発した CGS-PSO ハイブリッド手法の結果を文献 [1] から引用して表 6 に示

す。ただし、この手法は並列手法ではなく、プロセスは 1 つである。また、実行時間も異なっている。さらに、提案手法では解更新後に 2opt を実施している関係でパラメータも一部異なっている。

提案手法の並列ランダム多スタート LK 法に対する相対精度は平均誤差率から計算したものが最大値 101.562%、最小値 99.497% となった。また、最小誤差率から計算したものが最大値 101.239%、最小値 99.376% となった。さらに、実験結果 4.2 節の総実行時間の結果より、提案手法が並列ランダム多スタート LK 法に対して非常に高速に動作することが分かっている。これらのことから、提案手法は同等の計算機リソースを投入した並列ランダム多スタート LK 法に対しておおよそ同等の誤差率を持つ解をより短時間で見つけることに成功しているといえる。特に、表 6 と 4.2 節より、問題例 d2103 は並列ランダム多スタート LK 法より平均で 8.9 倍、総実行時間の最小値で 13.2 倍高速に動作し、平均値の相対精度 99.497%、最小値の相対精度 99.376% の解を生成することができる。その他の問題例においても、並列ランダム多スタート LK 法よりも非常に高速で、並列ランダム多スタート LK 法の最終誤差率を持つ解の平均に対して 99.497% から 101.562% というほぼ同等の結果を算出できることが判明した。ゆえに、提案手法は LK 法の実行時間を短縮することも可能であるといえる。また、文献 [1] 引用の CGS-PSO の結果と比較すると CGS-PSO ハイブリッド手法から Q-CGS-PSO-2opt-LK ハイブリッド並列手法への最終誤差率の改善が確認された。問題例 d2103 の平均と最小誤差率はほぼ差がないことから、問題例 d2103 は CGS にとって有利な都市配置であることが分かる。特に、都市数が 3,795 以上の問題例については LK 法の局所探索が有効に働いていることが分かる。ゆえに、提案手法 Q-CGS-PSO-2opt-LK ハイブリッド並列手法は CGS-PSO をさらに高性能化しているといえる。

4.4 考察

これらの結果より、すべての問題例で提案手法が並列ランダム多スタート LK 法に対して非常に速く動作することが判明した。さらに、解の質も並列ランダム多スタート LK 法に対してほぼ同等の品質の精度を持つ解を生成できた。これは、Q-CGS-PSO-2opt ハイブリッド並列手法がパラメータベクトル Q の非同期複製により CPU を有効に使い、さらに、CGS-PSO-2opt ハイブリッド手法に速やかに良好なパラメータが伝達されるからであると考えられる。並列ランダム多スタート LK 法よりも提案手法が高速に動作する理由は 3 章で述べた考察どおり、提案手法の Q-CGS-PSO-2opt-LK ハイブリッド並列手法の Q-CGS-PSO-2opt ハイブリッド並列手法が高速に LK 法の局所解に近い解を算出するためであると考えられる。

表 6 実験結果：最終誤差率
Table 6 Results: final error rate.

問題例	平均値 (引用)			平均誤差率		最小値 (引用)			最小誤差率	
	CGS-PSO [1]	提案手法	LK 法	CGS-PSO [1]	提案手法	LK 法	CGS-PSO [1]	提案手法	LK 法	相対精度
dsj1000		3.041%	1.921%	101.099%		2.100%	1.328%		100.762%	
pr1002	3.662%	2.727%	2.254%	100.463%	2.841%	1.912%	1.834%		100.077%	
si1032		0.171%	0.127%	100.044%		0.032%	0.047%		99.985%	
u1060		2.549%	1.707%	100.828%		1.580%	1.420%		100.158%	
vm1084		2.608%	1.628%	100.964%		1.420%	1.183%		100.233%	
pcb1173		2.897%	1.673%	101.204%		2.079%	1.401%		100.669%	
d1291		2.591%	2.609%	99.983%		1.592%	1.504%		100.087%	
rl1304		3.586%	1.993%	101.562%		2.670%	1.413%		101.239%	
rl1323		2.893%	1.532%	101.341%		1.690%	0.938%		100.745%	
nrw1379		2.333%	1.482%	100.838%		1.732%	1.326%		100.401%	
fl1400		1.593%	1.173%	100.415%		1.028%	0.964%		100.064%	
u1432	4.233%	1.621%	1.019%	100.596%	3.687%	1.207%	0.804%		100.399%	
fl1577		1.219%	1.076%	100.142%		0.769%	0.679%		100.089%	
d1655		2.733%	2.148%	100.573%		2.223%	1.659%		100.554%	
vm1748		2.741%	1.873%	100.852%		2.041%	1.399%		100.633%	
u1817		3.477%	3.361%	100.112%		2.708%	2.825%		99.886%	
rl1889		3.711%	2.574%	101.108%		2.482%	1.927%		100.544%	
d2103	2.028%	1.976%	2.492%	99.497%	1.616%	1.080%	1.715%		99.376%	
u2152		3.720%	3.065%	100.636%		2.714%	2.599%		100.112%	
u2319		0.426%	0.341%	100.086%		0.357%	0.322%		100.035%	
pr2392		3.003%	2.618%	100.375%		2.360%	1.952%		100.400%	
pcb3038		2.756%	1.982%	100.759%		2.282%	1.807%		100.467%	
fl3795	5.694%	1.909%	2.052%	99.860%	4.237%	1.491%	1.585%		99.908%	
fl4461		2.205%	1.717%	100.480%		2.078%	1.627%		100.444%	
rl5915		2.768%	2.282%	100.476%		2.251%	1.904%		100.341%	
rl5934	7.890%	3.369%	2.573%	100.777%	5.011%	2.312%	2.257%		100.054%	
pla7397		2.101%	1.617%	100.476%		1.626%	1.451%		100.172%	
rl11849		3.016%				2.742%				
usa13509		2.742%				2.391%				
brd14051		2.337%				2.131%				

しかしながら、提案手法 Q-CGS-PSO-2opt-LK ハイブリッド並列手法は最終的に LK 法による局所探索を行うので、LK 法の局所探索に起因する計算時間の増加は避けられない。1.3 万都市程度の問題例までは 30 時間もあれば計算が終了し、2.5 時間程度で品質が誤差率 5% の解を得ることができた。しかし、数万都市の問題例になると急激に計算時間が増大してしまう。ゆえに、最後に適用する局所探索法そのものに対して並列化手法を適用し高速化する必要がある。

また、MPI によるマルチプロセスやスレッディングによる並列化だけでなく、Single Instruction Multiple Data (SIMD) 命令を利用したベクトル演算による並列化手法についても新規提案する必要がある。

5. まとめ

本研究では、CGS-PSO ハイブリッド手法の性能を向上させるために並列処理を用いたパラメータチューニングと

局所探索を適用した提案手法 Q-CGS-PSO-2opt-LK ハイブリッド並列手法を提案した。また、計算機実験により提案手法の性能評価を行った。

実験結果より、同等の計算機リソースを投入した並列ランダム多スタート LK 法よりも非常に高速に動作することと、最終誤差率の品質が並列ランダム多スタート LK 法と同等の精度を持つ解であることが判明した。しかしながら、1.3 万都市程度までは比較的高速に解くことができたが、数万都市の問題例になると LK 法の特徴である長大な実行時間が現れてしまうと考えられる。

今後の課題として、さらなる高速化のために SIMD 命令を用いたベクトル演算による高速化やパラメータベクトル Q のより強力な複製戦略、最後の局所探索の並列化、より大きな計算機資源を持つ環境でのベンチマークなどがあげられる。本研究では、数万都市から急激に計算時間が増大するという予想を立てたが、より強力なアーキテクチャとアルゴリズムを両方駆使することで数万都市でもより高

速に動作するアルゴリズムを得ることができると考えられる。

参考文献

[1] 長辻亮太, 飯田修平, 榎原博之: 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, 情報処理学会論文誌数理モデル化と応用, Vol.9, No.1, pp.1–12 (2016).

[2] Graham, R.L.: Bounds for Certain Multiprocessing Anomalies, *The Bell System Technical Journal*, Vol.XLV, No.9 (1966).

[3] Klincewicz, J.G. and Rajan, A.: Using GRASP to solve the component grouping problem, Technical Report, AT&T Bell Laboratories, Holmdel, NJ (1992).

[4] Aarts, E., Korst, J. and Michiels, W.: Simulated Annealing, Springer Science+Business Media New York, pp.265–285 (2014).

[5] Dorigo, M.: Optimization, Learning and Natural Algorithms, Ph.D. Thesis, Politecnico di Milano, Italy (1992).

[6] Zhang, L., Wang, L. and Zheng, D.-Z.: An adaptive genetic algorithm with multiple operators for flowshop scheduling, *Int. J. Adv. Manuf. Technol.*, Vol.27, No.5-6, pp.580–587 (2006).

[7] 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).

[8] Gass, W.: *Particle Swarm Optimization*, Chapter 3, p.45, Springer-Verlag Berlin Heidelberg (2014).

[9] 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).

[10] Kurdi, M.: A new hybrid island model genetic algorithm for job shop scheduling Problem, *Computers & Industrial Engineering*, Vol.88, pp.273–283 (2015).

[11] Shi, X.H., Lu, Y.H., Zhou, C.G., Lee, H.P., Lin, W.Z. and Liang, Y.C.: Hybrid evolutionary algorithms based on PSO and GA, *CEC '03*, pp.2393–2399, IEEE (2003).

[12] Chu, S.-C., Tsai, P.-W. and Pan, J.-S. *Cat Swarm Optimization*, pp.854–858, Springer-Verlag Berlin Heidelberg (2006).

[13] Zhou, Y., Luo, Q., Chen, H., He, A. and Wu, J.: A discrete invasive weed optimization algorithm for solving traveling salesman problem, *Neurocomputing*, Vol.151, No.3, pp.1227–1236 (2015).

[14] Lin, S. and Kernighan, B.W.: An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Oper. Res.*, Vol.21, pp.498–516 (1973).

[15] TSPLIB, available from (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>).

付 録

A.1 プロセス数 M を変化させた場合

本研究では, 実験で利用した計算機のリソース (スレッド数) から, プロセス数 M を 20 とした. プロセス数と性能の関係を調べるために, dsj1000, pcb3038, pla7397 の 3 つの問題例に対して, プロセス数 M を 5, 10, 20 と変化させた場合の実験を行う. それぞれ誤差率が 5%, 8%, 10% の到達時間 [s] について試行回数 10 回の平均値を表 A.1 に示す. 表 A.1 より, プロセス数 M が多くなるとパラメータベクトル Q の伝搬速度が遅くなり, 結果として計算時間が長くなるのが分かる. $M = 20$ では伝搬速度の低下により結果が悪くなっている. さらに, パラメータベクトル Q の共有の有効性も $M = 10$ 程度で十分だと思われる.

A.2 パラメータベクトル Q の共有回数と改善回数

パラメータベクトル Q の共有の効果を調べるために, dsj1000, pcb3038, pla7397 の 3 つの問題例に対して, パラメータベクトル Q の共有回数と改善回数に関する実験を行う. 共有回数は, 1 回の試行で各プロセスがパラメータベクトル Q を共有する回数の合計である. 改善回数は, 共有後に解の改善があった回数を表す. プロセス数 M は 20 である. 共有回数と改善回数の試行回数 10 回の平均値を表 A.2 に示す. 問題例ごと (都市数ごと) に共有回数は異なり, 必ずしも都市数の多い問題例が共有回数が多いとはいえなかった. また, 解の改善は後半に多く起こる傾向があり, パラメータベクトル Q の共有は, 計算処理の後半に効果を発揮していると考えられる.

表 A.1 プロセス数 M を変化させた場合の結果

Table A.1 Results when the number of processes M is changed.

問題例	M	平均値				到達時間 [s]			
		閾値	5%	8%	10%	最小値	到達時間 [s]		
			5%	8%	10%	閾値	5%	8%	10%
dsj1000	5	6.9	1.4	1.4	1.4	6.3	1.3	1.3	1.3
	10	7.0	1.5	1.5	1.5	6.1	1.5	1.5	1.5
	20	7.9	1.7	1.5	1.5	7.0	1.0	1.0	1.0
pcb3038	5	83.6	21.3	17.4	60.9	16.7	16.7	16.7	16.7
	10	75.2	17.9	17.6	17.6	70.0	16.5	16.5	16.5
	20	103.0	20.4	19.0	19.0	74.0	18.0	18.0	18.0
pla7397	5	135.6	135.6	596.4	131.2	131.2	475.5	475.5	475.5
	10	134.9	134.9	594.9	135.2	135.2	614.4	614.4	614.4
	20	146.9	146.9	620.5	143.6	143.6	510.2	510.2	510.2

A.3 Q-CGS-PSO-2opt 手法の収束時間

本実験において、Q-CGS-PSO-2opt ハイブリッド並列手法に実行時間をパラメータとして与える。Q-CGS-PSO-2opt ハイブリッド並列手法が十分収束する実行時間を与えることが必要であるため、予備実験として本実験とは別の実験で表 3 に示す実行時間の 2 倍の実行時間を与え、収束の具合を確認する。表 A.3 に実行時間とそのときの Q-CGS-PSO-2opt ハイブリッド並列手法の誤差率を示す。表 A.3 の Δ 平均と Δ 最小は時間 0 から時間 1 への誤差率の変位である。時間 0 が本実験で使用されたパラメータで

表 A.2 パラメータベクトル Q の共有回数と解の改善回数

Table A.2 The number of sharing times and improvement times of parameter vector Q .

問題例	共有回数	改善回数
dsj1000	39.9	13.9
pcb3038	25.5	4.0
pla7397	38.5	14.1

表 A.3 Q-CGS-PSO-2opt ハイブリッド並列手法の収束具合

Table A.3 Convergence of the Q-CGS-PSO-2opt hybrid parallel method.

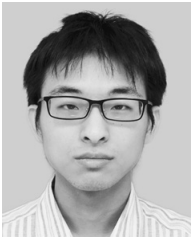
問題例	時間 0 [s]	平均値	最小値	時間 1 [s]	平均値	最小値	Δ 平均	Δ 最小
dsj1000	5	6.576%	6.300%	10	6.576%	6.300%	0.000%	0.000%
pr1002	5	5.866%	5.466%	10	5.866%	5.466%	0.000%	0.000%
si1032	5	1.648%	1.526%	10	1.618%	1.514%	-0.030%	-0.012%
u1060	5	5.937%	5.247%	10	5.937%	5.247%	0.000%	0.000%
vm1084	5	5.657%	5.456%	10	5.657%	5.456%	0.000%	0.000%
pcb1173	5	7.459%	7.057%	10	7.459%	7.057%	0.000%	0.000%
d1291	5	5.679%	5.274%	10	5.604%	5.085%	-0.075%	-0.189%
rl1304	5	6.594%	5.931%	10	6.594%	5.931%	0.000%	0.000%
rl1323	5	5.844%	5.362%	10	5.791%	5.362%	-0.053%	0.000%
nrv1379	5	5.930%	5.345%	10	5.819%	5.344%	-0.111%	0.000%
fl1400	10	3.320%	2.743%	20	3.318%	2.743%	-0.002%	0.000%
u1432	10	7.015%	6.654%	20	7.015%	6.654%	0.000%	0.000%
fl1577	10	3.777%	2.769%	20	3.632%	2.769%	-0.145%	0.000%
d1655	10	5.915%	4.761%	20	5.906%	4.761%	-0.010%	0.000%
vm1748	10	6.476%	6.015%	20	6.461%	6.015%	-0.015%	0.000%
u1817	10	6.402%	5.794%	20	5.971%	5.752%	-0.431%	-0.042%
rl1889	10	6.847%	5.767%	20	6.489%	5.767%	-0.357%	0.000%
d2103	10	3.400%	2.671%	20	2.698%	1.768%	-0.702%	-0.904%
u2152	10	7.477%	6.720%	20	6.699%	6.216%	-0.778%	-0.504%
u2319	10	4.712%	4.448%	20	4.372%	4.241%	-0.340%	-0.207%
pr2392	30	7.010%	6.549%	60	7.010%	6.549%	0.000%	0.000%
pcb3038	30	7.722%	7.239%	60	7.486%	7.118%	-0.236%	-0.121%
fl3795	60	4.585%	3.761%	120	3.952%	3.580%	-0.633%	-0.181%
fnl4461	60	6.817%	6.545%	120	6.661%	6.488%	-0.156%	-0.057%
rl5915	260	7.101%	6.538%	520	7.101%	6.538%	0.000%	0.000%
rl5934	260	7.044%	6.389%	520	7.044%	6.389%	0.000%	0.000%
pla7397	300	6.828%	6.666%	600	6.794%	6.666%	-0.034%	0.000%
rl11849	800	7.416%	7.106%	1,600	7.273%	7.079%	-0.143%	-0.026%
usa13509	5,000	7.473%	7.257%	10,000	7.473%	7.257%	0.000%	0.000%
brd14051	5,000	6.935%	6.697%	10,000	6.935%	6.697%	0.000%	0.000%

ある。時間 1 が予備実験で使用されたパラメータである。すべての問題例において時間 0 から時間 1 にかけての最小と平均誤差率の変位の大きさがすべて 1%未満であるのでほぼ収束していると考えられる。これらのことから、本実験の Q-CGS-PSO-2opt ハイブリッド並列手法の実行時間を決定した。



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 月同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。2017 年 3 月同大学院修了。同年 4 月よりウイングアーク 1st 株式会社勤務。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。



碓 一稀

1995年生。2017年4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月同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。巡回セールスマン問題の解法に関

する研究に従事。



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、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。IEEE, ACM 各会員。