THE IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS (JAPANESE EDITION)

EiC D 電子情報通信学会論文誌 情報・システム

VOL. J103-D NO. 5
MAY 2020

情報・システムソサイエティ
一般社団法人 電子情報通信学会
THE INFORMATION AND SYSTEMS SOCIETY
THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS

CMA-ESにおける高次元・悪条件最適化のための確率的次元選択手法*

清水　洸希[†a)]　　小宮山純平[††b)]　　豊田　正史[†††c)]

Stochastic Dimensional Selection for CMA-ES to Optimize High-Dimensional and Ill-Conditioned Functions*

Hiroki SHIMIZU[†a)], Junpei KOMIYAMA[††b)], and Masashi TOYODA[†††c)]

あらまし　本論文では，高次元かつ悪条件な目的関数の最適化を目的とした，CMA-ES において目的関数の次元を任意の次元の組に確率的に分割し一組ずつ更新する手法を提案する．CMA-ES は，進化戦略に分散共分散行列を導入し，変数間の依存性を考慮した最適化を可能としたアルゴリズムで，ブラックボックス最適化問題において優れた性能を発揮している．一方で，100,000 次元の Ellipsoid 関数のような高次元かつ悪条件な関数の最適化においては反復を重ねても，目的関数値の初期値からの減少が見られないことが報告されている [1]．本論文では，CMA-ES の変数の一つであるステップサイズが，目的関数が悪条件である場合に，分散共分散行列の適応を妨げていることが，原因の一つであることを実験を通して確認し，解決のため前述の手法を提案した．提案手法は，悪条件関数の最適化における条件数の良化とステップサイズのベクトル化の二点を達成し，高次元かつ悪条件な関数の最適化において，従来の CMA-ES よりも優れた結果を得た．

キーワード　進化戦略，進化計算，最適化，機械学習

1. まえがき

最適化問題とは，与えられた目的関数に対して，それを最小化あるいは最大化する変数の組を求める問題である．特に目的関数が明示的に与えられておらず，入力に対する出力のみが得られるような場合をブラックボックス最適化と呼ぶ．

ブラックボックス最適化は実世界においても，鉄道のダイヤグラム作成や航空機の翼の設計等多数の応用が存在する．実世界のブラックボックス最適化において想定される目的関数の特徴として，悪条件性，多峰性，変数間依存性が指摘されている [2], [3]．また，近年では，ニューラルネットワークのような数百万次元に及ぶ非常に高次元なモデルも主流であり，高次元空間への適応も重大な課題の一つである．

CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [4] は，ブラックボックス最適化において最も優れたアルゴリズムの一つとして知られている．CMA-ES は，進化戦略 (ES) に分散共分散行列を導入したアルゴリズムで，多変量正規分布 $\mathcal{N}(\boldsymbol{m}, \sigma^2\boldsymbol{C})$ に基づいて個体を多数生成し，より良い目的関数値を得た個体を用いて，分布の変数 (平均 $\boldsymbol{m}$，分散共分散行列 $\boldsymbol{C}$ 及びステップサイズ σ) の最適化を行うことで，解を探索する．ここでステップサイズは，探索地点と最適解との距離に応じて増減するスカラーで，分散共分散行列の補正を行い探索速度を高める役割をもつ．CMA-ES のもつ大きな特徴として，分散共分散行列の適応による変数間の依存性を考慮した個体生成，個体を用いることによる広範囲な探索，ステップサイズによる解の探索効率の向上が挙げられる．個体数やステップサイズについては，rank-one update と rank-μ update の組み合わせによる個体数の増加への

† 東京大学大学院情報理工学系研究科，東京都
Graduate School of Information Science and Technology, The University of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo, 113–8654 Japan
†† ニューヨーク大学レナード・N・スターン・スクール，米国
New York University Stern School of Business, Kaufman Management Center 8–81, 44 West 4th Street, New York, NY 10012, USA
††† 東京大学生産技術研究所，東京都
Institute of Industrial Science, The University of Tokyo, 4–6–1 Komaba, Meguro-ku, Tokyo, 153–8505 Japan
a) E-mail: shimizu@tkl.iis.u-tokyo.ac.jp
b) E-mail: junpei.komiyama@stern.nyu.edu
c) E-mail: toyoda@tkl.iis.u-tokyo.ac.jp
* 本論文は，データ工学研究専門委員会推薦論文である．
DOI:10.14923/transinfj.2019DET0003

対応，CSA (Cumulative Step-size Adaptation) を用いた悪条件性 (変数間のスケールの差) を考慮したステップサイズの更新といった様々な提案がなされている．以上のことから，CMA-ES は一般に，前述の悪条件性，多峰性，変数間依存性をもつ目的関数に対して有効であるとされている [5].

CMA-ES の課題として，目的関数の次元数 d に対し，時間計算量が $O(d^3)$，空間計算量が $O(d^2)$ であることから，高次元な最適化問題への適用が困難な点が挙げられる．計算量については，分散共分散行列を対角成分に限定することで双方の計算量を $O(d)$ に低減した sep-CMA-ES [6] をはじめとして，さまざまな手法が提案されており，特に sep-CMA-ES はその計算量の小ささにもかかわらず，目的関数に変数間の依存性がない場合は，通常の CMA-ES よりも高速に解を発見することができる．これらの計算量を改善したアルゴリズムの貢献により 10,000 次元程度までの最適化問題への拡張を可能としたが，それ以上に高次元な場合では依然として課題が存在する．Loshchilov は，原因については言及していないものの実験を通して，sep-CMA-ES を用いた 100,000 次元の Ellipsoid 関数の最適化では反復を重ねても，目的関数値の初期値からの減少が見られないことを指摘している [1]．本論文では実験を通して，この問題の原因が，目的関数が高次元なだけではなく悪条件性をもつ場合に，分散共分散行列の固有値の平均とステップサイズの増減の振る舞いが一致せずに大きく乖離し，探索効率が低下しているためであることを確かめた．

この問題を解決するため，本論文では CMA-ES において確率的に次元を選択する手法を提案する．提案手法では，目的関数の次元を任意の次元の組に確率的に分割する．そして世代ごとに組を選択し，その選択された次元について，平均，分散共分散行列，ステップサイズといった変数の更新を行う．この次元選択により，悪条件な関数の最適化における条件数の良化とステップサイズのベクトル化の二点を達成することができ，より高次元かつ悪条件な関数に適した最適化が可能となる．

実験では，高次元の Ellipsoid 関数や Star 型 Rosenbrock 関数といった悪条件性が強い関数を用い，提案手法が従来の CMA-ES よりも探索の速さや解の優劣の観点から優れることを確認した．

以下，**2.** で関連研究について説明し，次に **3.** で既存の CMA-ES を用いた実験を通して前述の問題の原因について検討を行う．そして，**4.** でその問題を解決するため，確率的に次元を選択する CMA-ES を提案し，**5.** で実験を通してその評価を行う．最後に **6.** でまとめを述べる．

2. 関連研究

本章では，CMA-ES のアルゴリズムの概要と，目的関数の特徴に応じた CMA-ES の改良の研究について述べる．

2.1 CMA-ES

本節では，CMA-ES のうち最も一般的とされる $(\mu/\mu_w,\lambda)$-CMA-ES について概要を述べたのち，分散共分散行列を対角成分に限定することで時間・空間計算量を低減した sep-CMA-ES について述べる．

2.1.1 $(\mu/\mu_w,\lambda)$-CMA-ES

CMA-ES は，一般の進化戦略とは異なり，解候補を直接保持せず，多変量正規分布 $\mathcal{N}(\boldsymbol{m}^{(t)},{\sigma^{(t)}}^2\boldsymbol{C}^{(t)})$ により解候補の集団を生成し，目的関数を用いた評価で高評価を得た候補から平均 $\boldsymbol{m}^{(t)}$ と分散共分散行列 $\boldsymbol{C}^{(t)}$ を更新することで解の探索を行う．ここで t はステップ数とする．

目的関数 $f(\boldsymbol{x})\in\mathbb{R}$, $\boldsymbol{x}\in\mathbb{R}^d$ を最小化する場合の，具体的なアルゴリズムについて以下に記す．はじめに，変数の初期化を行う．平均 $\boldsymbol{m}^{(0)}\in\mathbb{R}^d$，分散共分散行列 $\boldsymbol{C}^{(0)}\in\mathbb{R}^{d\times d}$，ステップサイズ $\sigma^{(0)}\in\mathbb{R}$ をそれぞれ探索領域に応じて決める．また，分散共分散行列とステップサイズのそれぞれの進化パス $\boldsymbol{p}_{\boldsymbol{c}}^{(0)}\in\mathbb{R}^d$ 及び $\boldsymbol{p}_{\boldsymbol{\sigma}}^{(0)}\in\mathbb{R}^d$ を $\mathbf{0}$ とする．そして，あらかじめ定めた終了条件を満たすまで以下のステップを繰り返す．

[Step 1.] 式 (1) のとおり，正規分布 $\mathcal{N}(\mathbf{0}\in\mathbb{R}^d,\boldsymbol{I}\in\mathbb{R}^{d\times d})$ から，λ 個の個体 $\boldsymbol{z}\in\mathbb{R}^{\lambda\times d}$ を生成する．次に式 (2) のとおり，$\boldsymbol{C}^{(\boldsymbol{t})}$ を平方分解し，$\boldsymbol{z}^{(t)}$ との行列積から $\boldsymbol{y}^{(t)}\in\mathbb{R}^{\lambda\times d}$ を求める．そして，式 (3) のとおり，$\sigma^{(t)}$ を $\boldsymbol{y}^{(t)}$ の各要素にかけ，平均 $\boldsymbol{m}^{(t)}$ との和から解集団 $\boldsymbol{x}^{(t)}\in\mathbb{R}^{\lambda\times d}$ を生成する．

$$\boldsymbol{z}^{(t)}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}), \tag{1}$$

$$\boldsymbol{y}^{(t)}=\boldsymbol{z}^{(t)}\sqrt{\boldsymbol{C}^{(t)}}, \tag{2}$$

$$\boldsymbol{x}^{(t)}=\boldsymbol{m}^{(t)}+\sigma^{(t)}\boldsymbol{y}^{(t)}. \tag{3}$$

[Step 2.] Step 1. で生成した解集団 $\boldsymbol{x}^{(t)}$ の各個体について $f(\boldsymbol{x}_i),(i=1,...,\lambda)$ から評価値を計算し，昇順に $\boldsymbol{x}^{(t)}$, $\boldsymbol{y}^{(t)}$, $\boldsymbol{z}^{(t)}$ を個体 (λ) の軸について並び替える．

[Step 3.] 各個体の順位についての重み $\boldsymbol{w}\in\mathbb{R}^\lambda$ を用いて，式 (4), (5) のとおり $\boldsymbol{w}$ と $\boldsymbol{x}^{(t)}$, $\boldsymbol{y}^{(t)}$ の個

体の軸についての内積を取り，$\boldsymbol{dy}^{(t)}$, $\boldsymbol{dz}^{(t)} \in \mathbb{R}^d$ を求める．ここで，重み $\boldsymbol{w}$ は，$1 < \mu < \lambda$ として，$w_1 \geq \cdots \geq w_\mu > 0$, $w_{\mu+1}, \cdots, w_\lambda = 0$, $\|\boldsymbol{w}\|_1 = 0$ を満たすものとする．

$$\boldsymbol{dy}^{(t)} = \sum_i^\lambda w_i \boldsymbol{y_i}^{(t)}, \tag{4}$$

$$\boldsymbol{dz}^{(t)} = \sum_i^\lambda w_i \boldsymbol{z_i}^{(t)}. \tag{5}$$

[Step 4.] Step 3. で求めた $\boldsymbol{dy}^{(t)}$, $\boldsymbol{dz}^{(t)}$ を用いて式 (6), (7), (8) のとおり進化パスを更新する．ここで，$\mu_w = \frac{1}{\|\boldsymbol{w}\|}$, $\chi_d = \mathbb{E}[\|\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\|] \simeq \sqrt{d}(1 - \frac{1}{4d} + \frac{1}{21d^2}) \in \mathbb{R}$ とし，$c_\sigma, c_c \in \mathbb{R}$ を各進化パスの学習率とする．

$$h_\sigma^{(t+1)} = \begin{cases} 1 & \|\boldsymbol{p_\sigma}^{(t+1)}\| < (1.4 + \frac{2}{n+1})\chi_d \\ 0 & else \end{cases}, \tag{6}$$

$$\boldsymbol{p_\sigma}^{(t+1)} = (1 - c_\sigma)\boldsymbol{p_\sigma}^{(t)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_w}\boldsymbol{dz}, \tag{7}$$

$$\boldsymbol{p_c}^{(t+1)} = (1 - c_c)\boldsymbol{p_c}^{(t)} + h_\sigma^{(t+1)}\sqrt{c_c(2 - c_c)\mu_w}\boldsymbol{dy}. \tag{8}$$

[Step 5.] Step 4. で求めた進化パスを用いて式 (9), (10), (11) のとおり多変量正規分布の変数の更新を行う．ここで，$OP(\cdot) \in \mathbb{R}^{n \times n}$ は入力ベクトルとそれ自身の直積行列とする．また，$\eta_m \in R$ を平均 $\boldsymbol{m}$ の学習率，η_{c_1}, $\eta_{c_\mu} \in R$ をそれぞれ分散共分散行列 $\boldsymbol{C}$ の rank-one update (式 (11) 右辺二項目), rank-μ update (式 (11) 右辺三項目) の学習率とし，$d_\sigma \in R$ をステップサイズの減衰率とする．

$$\boldsymbol{m}^{(t+1)} = \boldsymbol{m}^{(t)} + \eta_m \sigma^{(t)} \boldsymbol{dy}^{(t)}, \tag{9}$$

$$\sigma^{(t+1)} = \sigma^{(t)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|\boldsymbol{p_\sigma}^{(t+1)}\|}{\chi_d} - 1\right)\right), \tag{10}$$

$$\boldsymbol{C}^{(t+1)} = \boldsymbol{C}^{(t)} + \eta_{c_1}\left(OP(\boldsymbol{p_c}^{(t+1)}) - \boldsymbol{C}^{(t)}\right) + \eta_{c_\mu}\sum_i^\lambda w_i \left(OP(\boldsymbol{y_i^{(t)}}) - \boldsymbol{C}^{(t)}\right). \tag{11}$$

以上の Step 1.∼Step 5. を繰り返すことで，平均ベクトル $\boldsymbol{m}$ が解に，ステップサイズ σ が 0 に，分散共分散行列 $\boldsymbol{C}$ が $\boldsymbol{0}$ に収束することで，多変量正規分布がデルタ関数に収束し，最適解が求まる．

2. 1. 2　sep-CMA-ES

sep-CMA-ES は目的関数が高次元である場合に，時間・空間計算量の点から CMA-ES の適用が困難であるという問題を解決するために，CMA-ES における分散共分散行列を対角成分に限定したアルゴリズムである．sep-CMA-ES では，変数間の依存関係を考慮しない代わりに，時間計算量・空間計算量はともに $O(d)$ となる．

また，sep-CMA-ES では，分散共分散行列の学習率を増大させており，目的関数の入力次元を d として，rank-one update, rank-μ update ともに $(d+2)/3$ を係数にかける．

2. 2　ステップサイズの役割

CMA-ES では，多変量正規分布における分散をステップサイズ σ と分散共分散行列 $\boldsymbol{C}$ に分割して表現している．ステップサイズの更新には，式 (10) のとおり，ステップサイズの進化パスのノルム $\|\boldsymbol{p_\sigma}\|$ と，多変量正規分布のノルムの期待値 χ_d の比が用いられ，$\|\boldsymbol{p_\sigma}\| > \chi_d$ の場合，ステップサイズは増大し，$\|\boldsymbol{p_\sigma}\| < \chi_d$ の場合，減少する．ステップサイズの進化パス $\boldsymbol{p_\sigma}$ は，式 (7) のとおり，良い評価を得た解候補の生成の基になった乱数 $\boldsymbol{z}^{(t)}$ の重み付き和 $\boldsymbol{dz}^{(t)}$ を累積した値である．つまり，現在の探索域が最適解から離れている場合，$\|\boldsymbol{p_\sigma}\|$ は増大し，最適解に近い場合は減少する傾向にある．したがって，ステップサイズは探索地点と最適解との間の距離に応じて χ_d をしきい値として増減することで，探索速度の向上を図る役割をもっていると言える．

2. 3　CMA-ES の改良

CMA-ES の改良は多く存在し，目的関数の特徴に応じて設計される．目的関数の特徴は主に，変数間依存性 (Separability)，悪条件性 (Ill-Conditioning)，多峰性・大谷構造 (Multimodality) に分類される [2], [3].

変数間依存性とは，目的関数の変数が分離可能であるか否かを指す．全ての変数について分離可能であるならば，全ての変数を独立に最適化しても最適値に到達することができるが，そうでなければ変数間の依存関係を考慮する必要性がある．CMA-ES は分散共分散行列行列を用いることで解決を図っているが，高次元 (large-scale) の場合には時間・空間計算量が重大な問題となる．前述の sep-CMA-ES や，LM-CMA [1], VD-CMA [7] といった研究では，分散共分散行列のうち保持する成分を制限することで，時間・空間計算量を削減しつつ，変数間の依存関係を考慮した最適化を可能としている．

悪条件性とは，目的関数における各次元のスケールの差が非常に大きい状態を指し，目的関数のヘッセ

表 1 実験に用いたベンチマーク関数．$\boldsymbol{R}$ はランダムに生成した $d \times d$ の直交行列である．

目的関数	Definition
Sphere	$f_{sph}(\boldsymbol{x}) = \sum_{i=1}^{d} x_i^2$
Ellipsoid	$f_{ell}(\boldsymbol{x}) = \sum_{i=1}^{d} (1000^{\frac{i-1}{d-1}} x_i)^2$
Rotated Ellipsoid	$f_{ellrot}(\boldsymbol{x}) = f_{ell}(\boldsymbol{Rx})$
Star Rosenbrock	$f_{s-ros}(\boldsymbol{x}) = \sum_{i=2}^{d} \left(100(x_1 - x_i^2)^2 + (1 - x_i)^2\right)$

行列の最大固有値と最小固有値の比で定義される条件数によって定量的に測ることができる．本論文で扱う Ellipsoid 関数のような凸二次関数の場合では，各変数の係数がヘッセ行列の固有値に相当する．現在の CMA-ES のステップサイズ更新手法の主流であり，前述の $(\mu/\mu_w, \lambda)$-CMA-ES にも採用されている Cumulative Step-size Adaptation (CSA) [8], [9] は，ステップサイズを進化パスのマハラノビス距離を用いて更新することで，悪条件性に対する頑健性の向上を図っている．

多峰性･大谷構造とは，目的関数が局所最適解をもつ状態 (non-convex な状態) を指す．個体数の増加 [10]，再スタートごとに個体数を変化させる再スタート戦略 [11]，最適化の度合いに応じて世代ごとに個体数を変化させる Population Size Adaptation (PSA) [12] が提案されている．

従来の研究では，いずれも目的関数の次元数にかかわらずスカラーのステップサイズを用いているが，高次元下においてもスカラーのステップサイズで全ての次元について補正を行うことは不合理であると言える．また，目的関数の条件数そのものを良化させることができれば，悪条件な関数においてもより高速な探索が実現可能であると考えられる．そこで本研究では，目的関数の条件数の良化とステップサイズのベクトル化の二点に着目し，その実現を図る．

3. 予備実験：高次元かつ悪条件な関数の最適化における CMA-ES の挙動

本章では，高次元かつ悪条件な目的関数における CMA-ES の挙動，すなわち，良条件な目的関数の場合と比較してどのような性能低下や変数の推移の変化が現れるかを確認する．はじめに，良条件かつ変数間依存性のない Sphere 関数を用いた実験を行い，その結果を他の目的関数に対してのベースラインとする．次に，悪条件な関数として，Ellipsoid 関数及び Star 型 Rosenbrock 関数を用い，Sphere 関数の結果との比較を行う．一般的によく知られる Chain 型の Rosenbrock 関数は隣り合う変数間の依存性が強い関数であるが，Star 型の場合は，一つ目の変数とそれ以降の変数との間に強い依存性をもつ [13] ことから，一つ目とそれ以降の変数について依存の度合いが異なり，高次元な場合については悪条件な関数とみなすことができる．CMA-ES におけるハイパーパラメタは文献 [14] に示されている推奨値を使用した．また，本論文中で用いるベンチマーク関数を表 1 に示した．

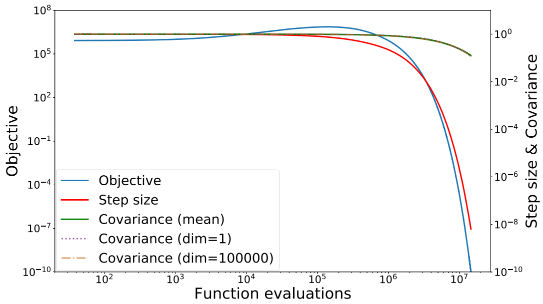
3.1 Sphere 関数を用いた実験

はじめに，sep-CMA-ES を用い，100,000 次元の Sphere 関数の最適化を行う．Sphere 関数は，良条件かつ変数間の依存性を有さない関数であり，最適化におけるベースラインとなる関数である．初期値については，$\boldsymbol{m}^{(0)} = U(-5, 5)$, $\boldsymbol{C}^{(0)} = \boldsymbol{I}$, $\sigma^{(0)} = 1.0$, $\lambda = 4 + 3\lfloor \ln(d) \rfloor$ と設定し，以降の全ての実験で同一の設定を用いた．また，目的関数の目標値は 10^{-10}，最大評価回数は $\lambda \times 10^7$ と設定した．

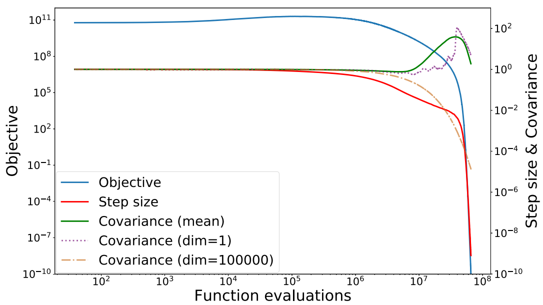
評価回数 (Function evaluations) に対する目的関数の値 (Objective)，ステップサイズ (Sigma)，分散共分散行列の固有値 (Covariance) の変化を，図 1 (a) に示した．分散共分散行列の固有値については，全ての平均の値 (Covariance (mean))，一つ目の変数に対応する値 (Covariance (dim=1)), 100,000 番目の変数に対応する値 (Covariance (dim=100,000)) を示した．なお，sep-CMA-ES における分散共分散行列は対角行列であるので，その固有値は対角成分に一致する．図より，約 1.5×10^7 回の評価回数で目標値に到達し，ステップサイズ及び分散共分散行列の固有値の値は単調に減少していることが伺える．また，分散共分散行列の固有値が次元別に示したものと平均を示したものとでおおよそ同じ値を取りつつ推移していることから，Sphere 関数の良条件性を確認することができる．

3.2 Ellipsoid 関数を用いた実験

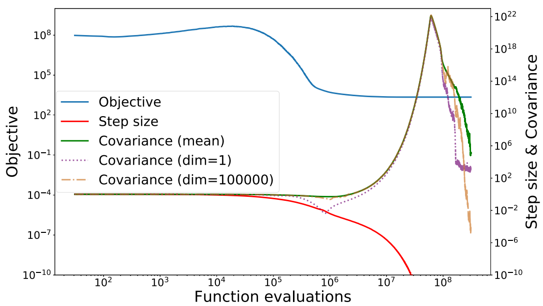
本節では，sep-CMA-ES を用い，100,000 次元の Ellipsoid 関数の最適化を行い，CMA-ES の高次元空間かつ悪条件下での振る舞いについて確認を行う．Ellipsoid 関数は，変数間の依存性は有さないが，悪条



(a) 100,000 次元の Sphere 関数



(b) 100,000 次元の Ellipsoid 関数



(c) 10,000 次元の Star 型 Rosenbrock 関数

図 1 ベンチマーク関数の最適化における sep-CMA-ES の各変数の推移．

件性をもつ (条件数 10^6) 関数である．目的関数以外の実験設定については，前節と同様に扱った．

評価回数 (Function evaluations) に対する目的関数の値 (Objective)，ステップサイズ (Sigma)，分散共分散行列の固有値 (Covariance) の変化を図 1 (b) に示した．また，前節と同様に，分散共分散行列の固有値について各次元の値の推移と平均値の推移を示した．図より，約 6.5×10^7 回の評価回数で目標値に到達しており，前節の Sphere 関数の結果と比較して，約 5.0×10^7 回程度多くなった．目的関数の値の減少は，Sphere 関数の場合と Ellipsoid 関数の場合のいずれも約 1.0×10^6 回付近から始まっており，Ellipsoid 関数の方が評価回数あたりの最適化速度が遅いことがわかる．また，Sphere 関数の場合では，ステップサイズと分散共分散行列の固有値が共に単調に減少しているのに対し，Ellipsoid 関数の場合では，分散共分散行列の固有値の平均 (Covariance (mean)) については，スケールの大きな次元の固有値 (Covariance (dim=100,000)) への適応に応じて上昇したのち減少に転じており，ステップサイズについては単調減少ではあるものの，分散共分散行列の固有値の平均の上昇中は緩やかであり，スケールの大きな次元の適応後は急速に減少している．つまり，スケールの大小にかかわらず単一のステップサイズを適用しているため，スケールの小さな次元についてはステップサイズが過大となり，スケールの大きな次元については過小となっている．その結果として，分散共分散行列の固有値の平均とステップサイズの増減の振る舞いの乖離が，Sphere 関数の場合と比較して拡大していると考えられる．この乖離が Sphere 関数の場合と比較して Ellipsoid 関数の最適化速度を下げていると推測され，ステップサイズをより各次元のスケールの大きさに適応させ，この乖離を抑制することでこの差を軽減することができると考えられる．

3.3 Star 型 Rosenbrock 関数を用いた実験

本節では，sep-CMA-ES を用い，10,000 次元の Star 型 Rosenbrock 関数の最適化を行う．Star 型 Rosenbrock 関数は，多峰性 (局所最適解を有する)，変数間の依存性をもち，次元数に比例して悪条件性を増す関数である．Star 型 Rosenbrock 関数では，一つ目の変数が二つ目以降の全ての変数との間に依存関係をもつのに対して，二つ目以降の変数は一つ目の変数との間のみ依存関係をもつため，一つ目とそれ以降の変数との間でスケールの差が生じ，高次元になるほど悪条件性が高まる (二次元の場合では，スケールの差が生じないため良条件な関数となる)．

前節と同様に結果を，図 1 (c) に示した．目標値には到達せず，最大評価回数時の目的関数の値は約 2252.2 となった．ステップサイズに注目すると，1.0×10^6 回の評価回数付近で，一つ目の変数に対応する固有値とともに収束に向かってしまっている．その後，一つ目の固有値も上昇に転じ，全ての固有値の平均が大きく上昇しているのに対して，ステップサイズは収束してしまい，最適化が緩やかになっていることがわかる．Star 型 Rosenbrock 関数は一つ目の変数の影響が非常に大きく，一つ目の固有値の収束がステップサイズに

大きな影響を及ぼしたと考えられる．

4. 提案手法

本章では，予備実験の結果を受け，高次元かつ悪条件な関数の最適化に適したアルゴリズムを提案し，その特徴について述べる．

前章における予備実験を通して，従来の CMA-ES では高次元かつ悪条件な関数の最適化において，良条件な Sphere 関数と比較して最適化速度の大きな低下が発生し，その原因の一つが分散共分散行列の固有値の平均とステップサイズの増減の振る舞いにおける乖離であることが確認された．本研究では，確率的な次元選択を提案し，目的関数の条件数の低減とステップサイズのベクトル化の二点を通して前述の問題の解決を図る．

以下，**4.1** において具体的な提案手法について，**4.2** において提案手法の特徴及び前述の問題に対しての効果について述べる．

4.1 更新次元の確率的選択

本節では，CMA-ES において各世代で更新する次元を確率的に選択するアルゴリズムを提案する．提案手法では，目的関数の入力次元をインデックス化し，そのインデックスの要素をランダムに並び替え，あらかじめ指定した次元長の組に分割する．そして，分割された組のインデックスに指定された次元について，CMA-ES の各変数から区分行列及びベクトルを抽出し，解候補を生成する．変数の更新についても，指定された次元についてのみ行う．インデックスの最後の要素まで参照を終えたら，再びインデックスをランダムに並び替える．この手法は CMA-ES と sep-CMA-ES の双方に適用でき，アルゴリズムの変更点は同様なため，CMA-ES への適用について以下の項で述べる．

4.1.1 CMA-ES への適用

最小化したい目的関数を $f(\boldsymbol{x}) \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^d$ とする．はじめに，平均 $\boldsymbol{m}^{(0)} \in \mathbb{R}^d$，分散共分散行列 $\boldsymbol{C}^{(0)} \in \mathbb{R}^{d\times d}$，ベクトル化されたステップサイズ $\boldsymbol{\sigma}^{(0)} \in \mathbb{R}^n$ をそれぞれ探索領域に応じて決め，分散共分散行列とステップサイズのそれぞれの進化パスを $\boldsymbol{p_c}^{(0)} \in \mathbb{R}^d$, $\boldsymbol{p_\sigma}^{(0)} \in \mathbb{R}^d = \boldsymbol{0}$ と初期化する．次に，目的関数の入力次元長のインデックスベクトル $\boldsymbol{\iota} \in \mathbb{R}^d = (0, 1, ..., d-1)$ を新たに導入する．インデックスベクトルは，あらかじめ要素をランダムに並び替え，インデックスの参照を終えた要素の終端地点を $e \in \mathbb{R} = 0$ と初期化し，各反復で選択する次元数を $s \in \mathbb{R}$ とする．そして，あらかじめ定めた終了条件を満たすまで以下のステップを繰り返す．

[Step 1.] $\boldsymbol{\iota}$ から，前世代での終了地点 e から $e+s$ までを取り出し，当世代でのインデックスベクトル $\boldsymbol{\iota'} \in \mathbb{R}^s$ とする．ただし，$e+s > d$ の場合，e から d までのベクトル $\boldsymbol{\iota'} \in \mathbb{R}^{d-e}$ とし，$\boldsymbol{\iota}$ の要素を全てランダムに並び替え，$e = 0$ とする．

[Step 2.] 生成した当世代でのインデックスベクトル $\boldsymbol{\iota'}$ に基づいて，各変数から部分ベクトル及び部分行列を生成する．具体的には，平均 $\boldsymbol{m'} \in \mathbb{R}^s$，分散共分散行列 $\boldsymbol{C'} \in \mathbb{R}^{s\times s}$，ステップサイズ $\boldsymbol{\sigma'} \in \mathbb{R}^s$，進化パス $\boldsymbol{p_c'} \in \mathbb{R}^s$, $\boldsymbol{p_\sigma'} \in \mathbb{R}^s$ となる．分散共分散行列については，指定されたインデックスの対角成分及びそれらの共起成分からなる対称行列を抽出する．

[Step 3.] 正規分布 $\mathcal{N}(\boldsymbol{0} \in \mathbb{R}^s, \boldsymbol{I} \in \mathbb{R}^{s\times s})$ から，λ 個の個体 $\boldsymbol{z'}^{(t)} \in \mathbb{R}^{\lambda\times s}$ を生成する．$\boldsymbol{C'}$ を平方分解し，$\boldsymbol{z'}^{(t)}$ との行列積から $\boldsymbol{y'}^{(t)} \in \mathbb{R}^{\lambda\times s} = \boldsymbol{z'}^{(t)}\sqrt{\boldsymbol{C'}}$ を求める．そして，$\boldsymbol{\sigma'}^{(t)}$ を $\boldsymbol{y'}^{(t)}$ の各要素にかけ，平均 $\boldsymbol{m}^{(t)}$ の $\boldsymbol{\iota'}$ の要素との和から解集団 $\boldsymbol{x}^{(t)} \in \mathbb{R}^{\lambda\times d} = \boldsymbol{m}^{(t)} + \boldsymbol{\sigma'}^{(t)} \circ \boldsymbol{y'}^{(t)}$ を生成する (ここで $\circ$ は二つのベクトル間の要素積を取る演算子とする)．このとき，$\boldsymbol{x}^{(t)}$ は通常の CMA-ES と同じ次元の行列となるが，$\boldsymbol{\iota'}$ 以外のインデックスの次元については，各個体全て同じ値 ($\boldsymbol{m}^{(t)}$ の値) を取る．

[Step 4.] 通常の CMA-ES と同様に，目的関数を用いて各個体を評価・順位づけし，昇順に並び替える．また，同様に重み $\boldsymbol{w} \in \mathbb{R}^\lambda$ との個体の軸 (λ の軸) についての内積 $\boldsymbol{dy'}^{(t)}$, $\boldsymbol{dz'}^{(t)} \in \mathbb{R}^s$ を求める．

[Step 5.] Step 4. で求めた $\boldsymbol{dy'}^{(t)}$, $\boldsymbol{dz'}^{(t)}$ をもとに，進化パスを更新する．このとき，$\boldsymbol{\iota'}$ で指定された要素のみ更新を行い，その他の要素については元の値を維持する．

[Step 6.] 進化パスをもとに平均 $\boldsymbol{m}^{(t)}$，分散共分散行列 $\boldsymbol{C}^{(t)}$，ステップサイズベクトル $\boldsymbol{\sigma}^{(t)}$ の更新を行う．ここでも，Step 4. と同様に $\boldsymbol{\iota'}$ で指定された要素のみ更新を行い，その他の要素については元の値を維持する．ステップサイズの更新値は，**2.** 式 (10) にあるとおりスカラー値であり，提案手法においても，同一反復中での更新値は $\boldsymbol{\iota'}$ の次元について同一となる．しかし，提案手法では，更新に用いられる $\boldsymbol{p_\sigma'}$ の次元の組が毎回異なることから，反復を重ねることでステップサイズベクトルの各要素は異なる値を取る．

以上の Step 1.~Step 6. を通常の CMA-ES と同様に終了条件を満たすまで繰り返す．sep-CMA-ES につ

いても分散共分散行列の対角成分から $\boldsymbol{\iota}'$ に対応する次元を抽出することで CMA-ES の場合と同様に適用することができる．

4.2 提案手法の特徴

進化計算において，最適化を行う変数を選択する手法は，Zhang らの OpenAI ES [15] でも用いられており，目的関数が高次元である場合に，集団サイズを維持した状態で更新する次元を減らすことができ，分割された次元の更新のタイミングを揃えることで並列化も可能なため，有効であるとされている．

本研究の提案手法では，次元のインデックスを最後まで参照した際に，インデックスベクトルの要素をランダムに並び替えることで，選択された次元に含まれる次元の組が毎回変化する．これは目的関数がブラックボックスである場合に，変数間の依存性や条件数が未知であるため，次元の組を変化させることで，確率的に依存関係をもつ変数を同じ組に含ませることが目的である．また，インデックスベクトルを先頭から順番に参照していくことで，全ての次元が均一に選択される．これは，選択される回数にばらつきが発生した場合に，選択されなかった次元の共起成分が維持され続け，世代を重ねるごとに対角成分に対して大きな値をとって最適化を妨げることを防ぐためである．

更新次元の確率的な選択により，提案手法は，悪条件関数における条件数の低減，ステップサイズのベクトル表現の二つの点から性能の向上を図っている．

提案手法では，各反復で選択した一部の変数のみ更新を行うことから，悪条件性はそれらの選択された変数における条件数によって変化する．具体的には，Ellipsoid 関数の場合，従来の条件数は 10^6 であるが，各ステップで 100 次元分の変数をランダムに選択すると，その選択された次元における条件数の期待値はおよそ 7.7×10^5 となり，従来よりも良条件下で最適化を行うことができる．

ステップサイズは，式 (10) のとおり進化パス $\boldsymbol{p_\sigma}$ のノルムを用いて更新することから，従来の CMA-ES ではスカラーで表すことしかできない (ベクトル化しても全ての成分が同じ値をもつ)．しかし提案手法では，次元の選択を行うことで，一反復中に選択された次元における更新値は同じ値となるが，各反復で選択される次元の組が変化する．そのため，ステップサイズをベクトル化することで，反復を重ねるごとにステップサイズの各成分が各次元に適応した値に更新される．これにより，従来の CMA-ES では，目的関数の各次元のスケールが大きく異なっていた場合に，分散共分散行列の固有値の平均とステップサイズの増減の振る舞いの乖離が大きいという問題が生じていたが，提案手法では，分散共分散行列を補正し大域的な収束を早めるというステップサイズの本来の性質を維持しつつ，より局所的な次元間のスケールの差を考慮することが可能となる．

最後に，時間・空間計算量について考える．空間計算量については元の分散共分散行列の値を保持するため次元数 d に対して，$O(d^2)$ (sep-CMA-ES の場合は $O(d)$) で変わらない．一方で，時間計算量については，分散共分散行列の固有値分解について，選択した次元のオーダーで行うことができる．

5. 評 価 実 験

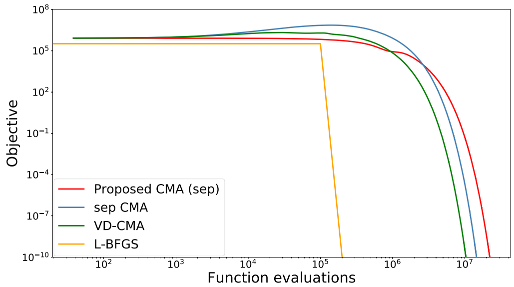
本章では，提案手法の高次元かつ悪条件な関数の最適化における性能を確認する．CMA-ES におけるハイパーパラメタは予備実験と同様に文献 [14] に示されている推奨値を使用した．ただし，ハイパーパラメタの引数となる次元の値については，目的関数の入力次元数 d ではなく，各世代ごとに選択された次元数 s を用いる．また，sep-CMA-ES への適用の場合は，sep-CMA-ES と同様に分散共分散行列の rank-one update, rank-μ update 双方の学習率に $(s+2)/3$ の係数をかける．また，本実験で用いた計算機環境を表 2 に示した．

5.1 高次元かつ悪条件な関数を用いた比較

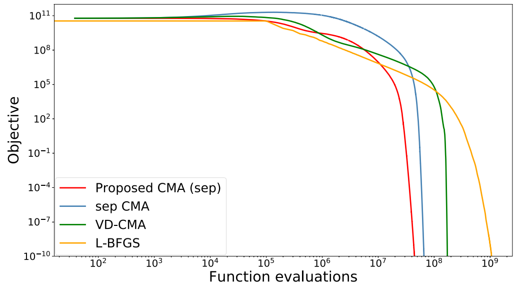
本節では，提案手法，既存の CMA-ES，L-BFGS 法との性能の比較を行う．既存の CMA-ES として，sep-CMA-ES 及び VD-CMA を用いた．VD-CMA は，分散共分散行列を対角行列とベクトルを用いた分解を行うことで，空間計算量の削減を図るアルゴリズムで，高次元空間下においても変数間の依存関係を考慮した最適化を行うことができる．L-BFGS 法は，勾配法をベースとしたアルゴリズムの中で最も優れたものの一つとして知られている．しかし，同手法は，勾配 (目的関数の一階導関数) を必要とするため，本来はブラックボックス最適化に用いることはできない．そのため，本実験では Loshcilov の実験 [16] と同様に，勾配を既

表 2　計算機環境

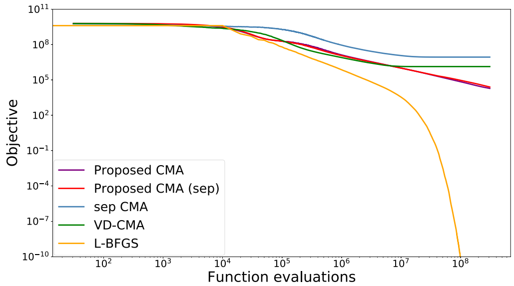
CPU	Intel Xeon E5-2680 v4 2.40GHz
RAM	256GB
GPU	NVIDIA Quadro P6000
GPU 環境	Cuda9.0 Cudnn7.2 CuPy4.3.0
OS	Ubuntu 16.04.3 LTS



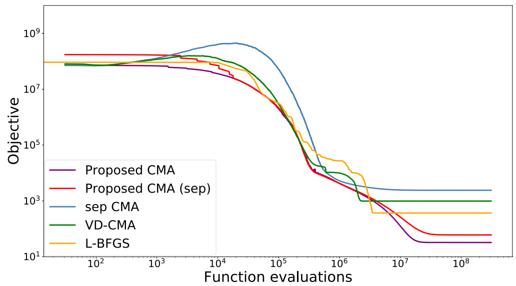
(a) 100,000 次元の Sphere 関数



(b) 100,000 次元の Ellipsoid 関数



(c) 10,000 次元の Rotated Ellipsoid 関数



(d) 10,000 次元の Star 型 Rosenbrock 関数

図 2　ベンチマーク関数の最適化における各アルゴリズムの比較.

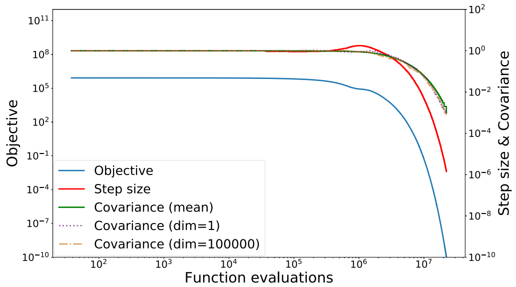
知の情報とする代わりに，一反復あたりの評価回数について目的関数の評価回数に各次元における導関数の計算回数を加えた回数とした．また，初期値，目標値，最大評価回数等の設定については予備実験と同様とし，提案手法における各反復の選択次元数は 100 とした．最適化を行うベンチマーク関数については，予備実験で用いた三つの関数に加えて，Rotated Ellipsoid 関数を用いた．Rotated Ellipsoid 関数は，Ellipsoid 関数の入力を直交行列を用いて変換する関数で，通常の Ellipsoid 関数と同様に悪条件な関数であることに加え，変数間の依存性をもつ．

各アルゴリズムの評価回数 (Function evaluations) に対する目的関数の値 (Objective) の推移を，Sphere 関数，Ellipsoid 関数，Rotated Ellipsoid 関数，Star 型 Rosenbrock 関数について，それぞれ図 2 (a), (b), (c), (d) に示した．また，各関数について提案手法 (sep) における，評価回数に対するステップサイズと分散共分散行列の固有値の推移を，Sphere 関数，Ellipsoid 関数，Rotated Ellipsoid 関数，Star 型 Rosenbrock 関数について図 3 (a), (b), (c), (d) に示した．

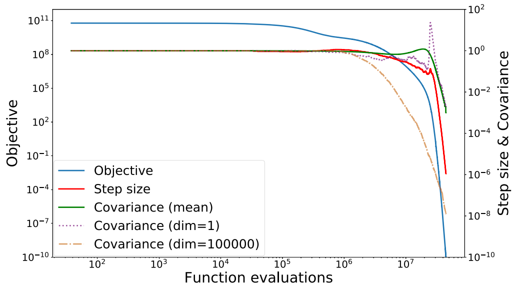
はじめに，Sphere 関数では，L-BFGS 法が最も少ない評価回数で目標値に到達し，提案手法は約 2.5×10^7 回で最も多くの評価回数を必要とした．提案手法は，一反復あたりに更新を行う次元数を選択するため，良条件な関数では優位性を示すことはできなかった．

次に，Ellipsoid 関数では，約 4.5×10^7 回と最も少ない評価回数で目標値に到達し，sep-CMA-ES よりも約 2.0×10^7 回程度評価回数が少なくなった．各変数について見ると，予備実験における sep-CMA-ES の場合と比較して，分散共分散行列の固有値とステップサイズの推移間の乖離が小さく抑えられており，乖離の抑制が結果に影響したと考えられる．

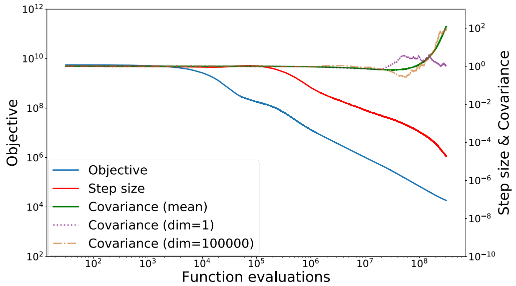
Rotated Ellipsoid 関数では，L-BFGS 法のみ最適値に到達し，CMA-ES アルゴリズムはいずれも設定した最大評価回数の範囲内では到達しなかった．図 2 (c) より，sep-CMA-ES と VD-CMA は，最適化が収束しているのに対して，提案手法は目的関数値が減少過程のうちに最大評価回数に達しており，評価回数を増やすことでより最適な値に到達可能であることが見込まれる．また，変数間の依存性をもつ関数の最適化であるにもかかわらず，提案手法において，依存性を考慮する手法 (図 2 (c) 中紫線) と，分散共分散行列を対角成分に限定し依存性をしない手法 (図 2 (c) 中赤線) との間に大きな差は見られなかった．これは，入力変数の Rotation に用いる直交行列の各成分の値が，高次元になるにつれて小さくなるためであると考えられ



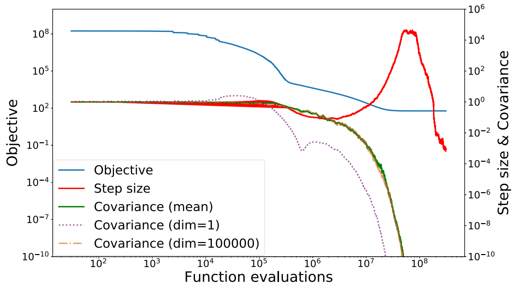
(a) 100,000 次元の Sphere 関数



(b) 100,000 次元の Ellipsoid 関数



(c) 10,000 次元の Rotated Ellipsoid 関数



(d) 10,000 次元の Star 型 Rosenbrock 関数

図 3　ベンチマーク関数の最適化における提案手法 (sep) の各変数の推移．

る．Rotation を用いる関数において，次元の増大につれて変数間の依存性による影響が小さくなる現象は，秋本らによる実験 [7] においても見受けられる．

最後に，Star 型 Rosenbrock 関数では，いずれのアルゴリズムも目標値には到達しなかったが，提案手法が最大評価回数時に目的関数の値が約 59.76 となり最小の値に到達した．こちらの場合でも，ステップサイズと分散共分散行列の固有値の乖離が，sep-CMA-ES よりも小さくなったことが原因であると考えられる．

5.2　選択次元の大きさについての比較

本節では，提案手法の選択次元の大きさの違いについての性能の比較を行う．100,000 次元の Ellipsoid 関数と 10,000 次元の Star 型 Rosenbrock 関数に対して，選択次元を 10, 100, 1,000 と変えて実験を行った．Ellipsoid 関数に対しては sep 型の (分散共分散行列を対角成分に限定する) 提案手法を用い，Star 型 Rosenbrock 関数については通常の (分散共分散行列の全ての要素を考慮する) 提案手法を用いた．目標値は 10^{-10} とした．

評価回数あたりの目的関数の値を図 4 に示す．Ellipsoid 関数では，選択次元を 100 としたものが最も少ない評価回数で目標値に到達した．一方で，Star 型 Rosenbrock 関数では，目標値に到達したものはなかったが，選択次元を 10 としたものが最もよい値を得た．以上の結果より，更なる検証が必要ではあるが，目的関数の入力次元数に対して 0.1%程度の選択次元が適しているという推測が得られた．

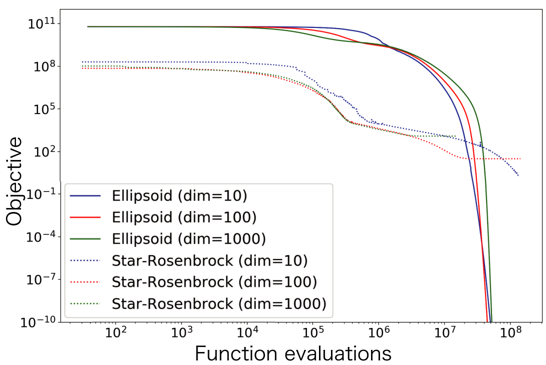
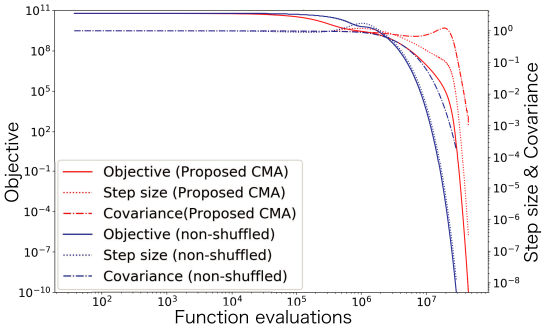


図 4　選択次元数を変化させた場合の評価回数ごとの目的関数の値の推移．

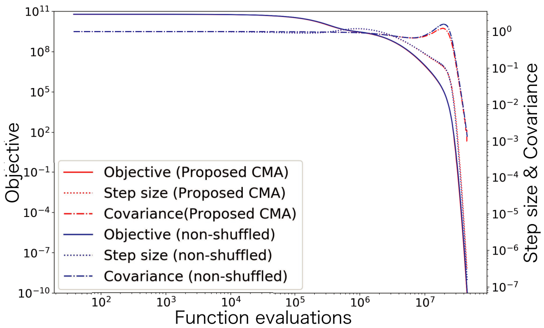
5.3　次元の選択方法についての比較

本節では，提案手法における次元の選択方法について，ランダムに選択する場合と，選択する次元の組を固定する場合とで比較を行う．後者については，各次元について先頭から 100 次元ずつ選択を行った．

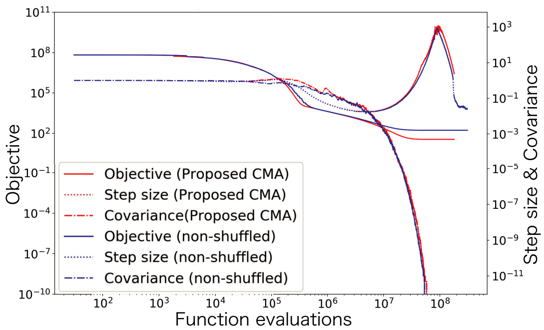
はじめに，Ellipsoid 関数について，評価回数に対



(a) 100,000 次元の Ellipsoid 関数



(b) 係数を入れ替えた 100,000 次元の Ellipsoid 関数



(c) 10,000 次元の Star 型 Rosenbrock 関数

図 5　ベンチマーク関数の最適化における次元の選択方法の比較．

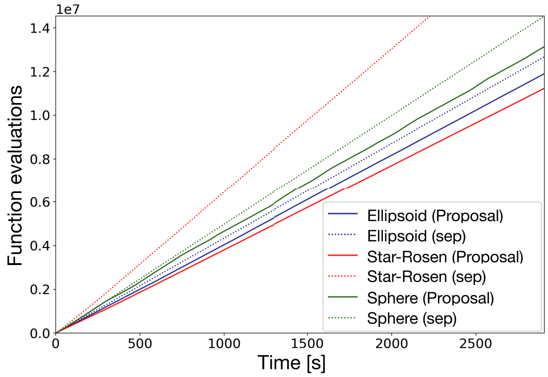


図 6　実時間に対する関数評価回数の推移．

する目的関数値，分散共分散行列の固有値の平均，ステップサイズの推移を，図 5 (a) に示した．図より，ランダムに選択する場合が約 4.5×10^7 回で目標値に到達したのに対し，組を固定した場合では，約 2.0×10^7 回で到達した．Ellipsoid 関数では，各次元について単調にスケールが増大するため，順番に次元を固定した場合，スケールが近い次元同士が同じ組に含まれることが影響したと考えられる．

次に，各次元について係数の順番をランダムに入れ替えた Ellipsoid 関数について，評価回数に対する目的関数値，分散共分散行列の固有値の平均，ステップサイズの推移を，図 5 (b) に示した．前述のとおり，従来の Ellipsoid 関数は各次元について単調にスケールが変化するため，順番を入れ替えることでその影響の確認を行う．図より，どちらの選択方法の場合でもほぼ同様の結果が得られた．Ellipsoid 関数の係数をランダムに入れ替えたことにより，次元の組を固定した場合でも，ランダムに選択する場合と同等の状態になったと考えられる．

最後に，Star 型 Rosenbrock 関数について，評価回数に対する目的関数値，分散共分散行列の固有値の平均，ステップサイズの推移を，図 5 (c) に示した．図より，どちらの場合も目標値に到達できなかったが，ランダムに選択を行ったものが，最大評価回数において，より良い値に到達することができた．Star 型 Rosenbrock 関数では，一つ目の変数とそれ以降の変数との間に依存関係があるため，変数の組にそれらが同時に含まれることが望ましいと考えられる．ランダムに選択を行う場合は，確率的に依存関係のある変数同士が含まれる次元の選択が実現されるのに対し，固定した場合は，はじめの選択に含まれなかった変数については実現されないため，この点が結果に影響を与えたと考えられる．

以上の結果から，変数の選択方法について，目的関数における変数間のスケールの近さや依存関係が既知の場合は，それらを考慮することで最適化性能を高めることが可能であると言える．一方で，目的関数がブラックボックスである場合ではランダムな選択が妥当であると考えられる．

5.4 計算時間についての比較

本節では，計算時間について提案手法と sep-CMA-ES の性能比較を行う．目的関数には，100,000 次元の Ellipsoid 関数，10,000 次元の Star 型 Rosenbrock 関数，100,000 次元の Sphere 関数を用いた．実時間に対する関数評価回数の推移を図 6 に示した．

図より，いずれの関数についても，提案手法がより短い時間で関数評価を行えており，計算量についても一定の有効性を示せていると考えられる．

6. む す び

本論文では，はじめに，高次元かつ悪条件下における既存の CMA-ES の問題について，予備実験を通してその原因の一つが，分散共分散行列の固有値の平均とステップサイズの増減の振る舞いにおける乖離であることを明らかにした．そして，その問題を解決するため，CMA-ES が最適化を行う次元を各世代でランダムに選択し，選択された次元について，ステップサイズをはじめとする CMA-ES の変数を更新するアルゴリズムを提案し，その評価を行った．

提案手法は，確率的に次元を選択することで，目的関数の条件数の良化とステップサイズのベクトル化の二点を達成し，既存手法における前述の課題を解決した．性能評価では，Ellipsoid 関数や Star 型 Rosenbrock 関数のような悪条件な目的関数を用いた実験において，従来の CMA-ES より少ない評価回数で優れた解に到達できることを明らかにした．一方で，Sphere 関数のように良条件な関数に対しては，世代ごとに更新する変数を選択していることから，目標値への到達までの評価回数で従来の CMA-ES を下回った．また，次元の選択方法については，目的関数がブラックボックスである場合はランダムな選択が最も妥当であることを示した．

今後の展望として，提案手法の性質についての更なる検証，次元ごとのスケールの近さや依存関係に動的に適応する次元選択方法の提案，本手法のニューラルネットワークの学習への適用等が挙げられる．特にニューラルネットワークの学習は，層ごとの役割が異なるため悪条件最適化であると考えられ，また多層なネットワークが一般的であり非常に高次元であることから，本手法の特性が活かされやすいと考えられる．

謝辞 本研究は, JSPS 科研費 16H02905, 17K12736 の助成及び JST CREST JPMJCR19A4 の支援を受けたものである．

文 献

[1] I. Loshchilov, “A computationally efficient limited memory CMA-ES for large scale optimization,” Proc. 2014 Conference on Genetic and Evolutionary Computation - GECCO ’14, pp.397–404, ACM Press, 2014.

[2] N. Hansen, S. Finck, R. Ros, and A. Auger, “Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions,” Technical Report RR-6829, INRIA, 2009.

[3] 坂井節子，高濱徹行，“最適化手法における関数評価回数の削減手法—ポテンシャルモデルに基づく比較推定法の提案,” 数理解析研究所講究録，vol.1548, no.7, pp.61–70, 2007.

[4] N. Hansen, “The CMA Evolution Strategy: A Tutorial,” arXiv:1604.00772, 2016.

[5] 秋本洋平, “Evolution Strategies による連続最適化 CMA-ES の設計原理と理論的基盤,” 進化計算の時代特集号, vol.60, pp.292–297, 2016.

[6] R. Ros and N. Hansen, “A simple modification in CMA-ES achieving linear time and space complexity,” Parallel Problem Solving from Nature – PPSN X, vol.5199, pp.296–305, Springer Berlin Heidelberg, 2008.

[7] Y. Akimoto, A. Auger, and N. Hansen, “Comparison-based natural gradient optimization in high dimension,” Proc. 2014 Conference on Genetic and Evolutionary Computation - GECCO ’14, pp.373–380, ACM Press, 2014.

[8] A. Ostermeier, A. Gawelczyk, and N. Hansen, “Step-size adaptation based on non-local use of selection information,” Parallel Problem Solving from Nature — PPSN III, vol.866, pp.189–198, 1994.

[9] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” Evolutionary Computation, vol.9, no.2, pp.159–195, 2001.

[10] N. Hansen and S. Kern, “Evaluating the CMA evolution strategy on multimodal test functions,” Parallel Problem Solving from Nature - PPSN VIII, vol.3242, pp.282–291, Springer Berlin Heidelberg, 2004.

[11] A. Auger and N. Hansen, “A restart cma evolution strategy with increasing population size,” 2005 IEEE Congress on Evolutionary Computation, vol.2, pp.1769–1776, IEEE, 2005.

[12] K. Nishida and Y. Akimoto, “PSA-CMA-ES: CMA-ES with population size adaptation,” Proc. Genetic and Evolutionary Computation Conference on - GECCO ’18, pp.865–872, ACM Press, 2018.

[13] S. Kobayashi, “The frontiers of real-coded genetic algorithms,” Trans. Japanese Society for Artificial Intelligence, vol.24, pp.147–162, 2009.

[14] N. Hansen and A. Auger, “Principled design of continuous stochastic search: From theory to practice,” Theory and Principled Methods for the Design of

Metaheuristics, pp.145–180, 2014.
[15] "On the relationship between the openAI evolution strategy and stochastic gradient descent," arXiv:1712.06564, 2017.
[16] I. Loshchilov, "LM-CMA: An alternative to L-BFGS for large-scale black box optimization," Evolutionary Computation, vol.25, no.1, pp.143–171, 2017.

（2019 年 6 月 28 日受付，11 月 4 日再受付，
2020 年 1 月 31 日早期公開）

清水　洸希

東京大学大学院情報理工学系研究科電子情報学専攻博士課程在学中．2017 年明治大学理工学部電気電子生命学科卒業．2019 年東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了．

小宮山純平

2009 年 4 月から 2012 年 6 月まで (株) ドワンゴにソフトウェア・エンジニアとして勤務．2012 年 10 月より東京大学大学院情報理工学研究科数理情報学専攻に入学，2016 年 3 月に博士（情報理工学）を取得．2016 年 4 月から 2019 年 8 月まで東京大学生産技術研究所助教．2019 年 9 月より現職．機械学習・データマイニング分野の研究が専門．2015 年 IBISML 研究会賞 (IEICE TC-IBISML Research Award) を受賞．

豊田　正史　（正員）

東京大学生産技術研究所教授．1994 年東京工業大学理工学部情報科学科卒業．1996 年同大学大学院情報理工学研究科修士課程修了．1999 年同大学大学院情報理工学研究科博士過程修了．博士 (理学)．同年，科学技術振興事業団計算科学技術研究員．ウェブマイニング，ユーザインターフェース，ビジュアルプログラミングに興味をもつ．ACM, IEEE CS，日本ソフトウェア科学会各会員．