

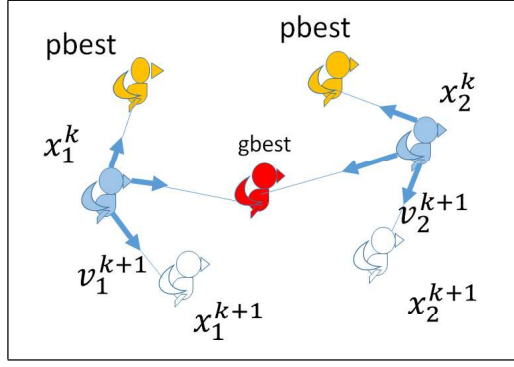


図 4.1: PSO の探索模式図

$$x_d^{k+1} = x_d^k + v_d^{k+1} \quad (4.1)$$

$$v_d^{k+1} = wv_d^k + c_1r_1(x_{db}^k - x_d^k) + c_2r_2(x_g^k - x_d^k) \quad (4.2)$$

ここで, PSO の探索模式図及び速度と位置の更新式より, pbest に向かう $c_1r_1(x_{db}^k - x_d^k)$, gbest に向かう $c_2r_2(x_g^k - x_d^k)$, これまでの進行方向へ向かう wv_d^k の 3 つのベクトルを合成して速度ベクトル v_i^{k+1} を決定し, それを元に次に移動する位置 x_d^{k+1} 決定する.

PSO の探索式はランダム要素を含み, 同時に最良解情報である pbest と gbest が探索に伴い変化するという時変性を有している [40]. このままの形では理論解析が困難であるので, 一つの Particle に着目し, 一次元の位置 x と速度 v について考え, さらに pbest と gbest を一つの点に縮約した簡略モデルが提案されている [39]. この簡略モデルは, 確定的な線形時不変システムとして表現されており, その安定性を示す.

Particle i に注目すると速度ベクトル v^{k+1} は以下の式のように変形できる (式 (4.3), 式 (4.4)). ステップ幅 ϕ は二つの一様乱数を足し合わせたものであり, 最小値 0, 最大値 $C_1 + C_2$, 平均 $\frac{C_1+C_2}{2}$ の分布に従う.

$$V^{k+1} = wv^k + \phi(P - x^k) \quad (4.3)$$

$$P = \frac{\phi_1 pbest^k + \phi_2 gbest^k}{\phi_1 + \phi_2} \quad (4.4)$$

ここで, $\phi = \phi_1 + \phi_2$, $\phi_1 = C_1 rand$, $\phi_2 = C_2 rand$, さらに $y^k = p - x^k$ とおくと, 式 (4.5) のように表せる. また $\phi = (C_1 + C_2)/2$ と見なすと固有値 λ は式 (4.6) のように表せる. よって λ が 1 を境にシステムの特徴が安定・不安定 (収束・発散) に変化することが分かる.

$$\begin{bmatrix} v^{k+1} \\ y^{k+1} \end{bmatrix} = \begin{bmatrix} w & \phi \\ -w & 1 - \phi \end{bmatrix} \begin{bmatrix} v^k \\ y^k \end{bmatrix} \quad (4.5)$$

$$\lambda = \frac{w + 1 - \phi \pm \sqrt{(w + 1 - \phi)^2 - 4w}}{2} \quad (4.6)$$

連続型 PSO アルゴリズム (Continuous Particle Swarm Optimization; CPSO) について述べる.

ベクトル y と $\text{sgn}(y)$ の要素によって与えられる対角要素を持つ対角行列を $\text{diag}[y]$ とする. y の σ 関数を表す. として $\text{sgn}(y) = 1$ if $y > 0$ の場合は, $\text{sgn}(y) = -1$ if $y < 0$.

したがって, 正の定数であると仮定すると, 最小化のために X の進化を近似することが提案される. また CPSO の安定性解析も議論されている [41].

状態変数 X, V, X_{db} はベクトルではなく, 以前に定義された適切な次元の行列であるため, 上記の表記法は標準状態空間表記法ではない. また以下に CPSO の位置と速度の更新式 (式 (4.7), 式 (4.8)) と, アルゴリズムについて示す. 問題解決の実行可能領域を考え, 行列による連続時間 PSO 動力学を示す (式 (4.9), 式 (4.10), 式 (4.11)).

CPSO アルゴリズム

- 1. X, V とパラメータ α, β, γ と a の初期値を設定する.
- 2. X_{db}, X_{gb} の初期値を導出する.
- 3. $\dot{V}$ を計算して, V を更新する.
- 4. X を更新して X_{db}, X_{gb} を評価する.
- 5. 収束すると仮定した場合は終了しそれ以外の場合は 3 から繰り返す.

$$\dot{X} = V \quad (4.7)$$

$$\dot{V} = -\alpha V + \beta(X_{db} - X) + \gamma(X_{gb} - X) \quad (4.8)$$

$$\dot{X}_{db} = a(X - X_{db})[I_n + \text{diag}[\text{sgn}(F(X_{gb}) - F(X))]] \quad (4.9)$$

$$\dot{X}_{gb} = X_{db} Q_j \quad (4.10)$$

$$j = \arg \inf_{0 \leq i \leq n} (f(x_{db})) \quad (4.11)$$

オリジナルの PSO アルゴリズムに含まれる恣意性を少なくし, より効率的かつ高精度な探索を実現するために勾配法による速度評価を導入している [42] [43]. 運動性素子が自分の置かれた近くの環境を知覚してより適合度の高い空間座標を獲得するために, 以下のようなセンシング・アルゴリズムを搭載する.

勾配によるスケーリングパラメータの導入を行う. 粒子が投入された探索空間 (ξ, η) には問題に応じた 目的関数 Q が定義されており, 粒子はその最大値か最小値を探索するものとする. 現時間ステップ k における粒子の位置座標を (ξ_k, η_k) とし, その座標における目的関数の値を Q_k として, 粒子の移動に伴う目的関数の変化に注目すると次のような目的関数の離散的な勾配 α が得られる.

勾配 α を, $v_i^{k+1} = \beta^k v_i^k$, $\beta^k = \alpha^{k-1/\alpha^k}$ と置くことでランドスケープに合わせた調整が可能となる. つまり, 最適点が遠いと思うなら早く, 近いと感じるならば遅く移動する. よってオリジナル PSO より精密な探索が実施できる.

本節では CPSO に勾配情報の要素を加えた勾配 PSO について解説する． PSO の応用法である CPSO の応用法であり， X, Y の二つの行列に加えて Z を加えかつ，いくつかのパラメータを与えて再急降下法を用いる． 以下は Z を表す (式 4.12)，

勾配 PSO アルゴリズム

- 1. X, V とパラメータ α, β, γ と a の初期値を設定する．
- 2. X_{db}, X_{gb}, X_μ から Z の初期値を導出する．
- 3. $\dot{Z}$ を計算して, Z を更新する．
- 4. $\dot{V}$ を計算して, V を更新する．
- 5. X を更新して X_{db}, X_{gb} と Z を評価する．
- 6. X_μ を更新する．
- 7. 収束すると仮定した場合は終了しそれ以外の場合は 3 から繰り返す．

$$Z = \beta(X_{db} - X) + \gamma(X_{gb}T - X) + \gamma(X_\mu) \quad (4.12)$$

以下の条件を考慮する (式 (4.13)). X_0 は初期位置行列である．

$$X = X_0 + \int_0^t V(s)ds \quad (4.13)$$

よって勾配 PSO の更新式を以下に示す． ここでは X, V, X_{db}, X_{gb}, T の次元は簡略化のため省略する．

$$\dot{X} = V \quad (4.14)$$

$$\dot{V} = -\alpha V + Z \quad (4.15)$$

$$\dot{Z} = \beta(\dot{X}_{db} - \dot{X}) + (X_{gb}\dot{T} - \dot{X}) + \delta(\dot{X}_\mu) \quad (4.16)$$

$$\dot{X}_{db} = a(X - X_{db})[I_n + \text{diag}[\text{sgn}(F(X_{gb}) - F(X))]] \quad (4.17)$$

$$\dot{X}_{gb} = X_{db}Q_j, j = \arg \inf_{0 < i \leq n} (f(x_{db_i})) \quad (4.18)$$

$\alpha, \beta, \gamma, \delta$ などの実数は， PSO と勾配情報を調整するために重み付けするパラメータである． X_μ はニューラルネットワークのダイナミクスに由来する新しい行列である． X_μ は以下で定義する (式 (4.19)) ．

$$x_{\mu i} = -C \sum_{i=1}^n n \frac{\partial f(y_i(t))}{\partial y_i} \frac{\partial \varphi(x(t))}{\partial x_i} \quad (4.19)$$

$$\ddot{x}_i = -\alpha x_i(t) + z_i(t) \quad (4.20)$$

$$y_i(t) = \varphi(x_i(t)) \quad (4.21)$$

したがって z_i^k は Z のベクトルである．次に，差分法を適用する．理論的な分析の観点から，PSO の $\dot{V}$ と $\dot{x}_i$ は同等のものとみなす． $\beta(\dot{X}_{db} - \dot{X}) + \gamma(\dot{X}_{gb}T - \dot{X})$ は PSO の速度を制御する． $\delta(\dot{X}_\mu)$ は勾配情報を制御する．

PSO が有するグローバル探索，ニューラルネットワークが持つ局所最急降下法などがある．連続時間モデルでは，PSO とニューラルネットワークの組み合わせの理論的アルゴリズムが考慮されるが，分散モデルによって数値シミュレーションが行われる．サンプリング時間の設定は，係数 (β, γ, δ) の値によって変化する．したがって， X_μ を計算して取得する．

§ 4.2 制約がある場合の PSO

前節までは PSO に勾配情報を加える説明を行った．本節ではそこに制約条件を加える．連続時間 PSO アルゴリズムについて述べる．

PSO の更新式を力学系モデルとみなし，その連続化を試みると，

$$\frac{dx^p(t)}{dt} = c \int_0^t e^{-a(t-\tau)} [F^p(x^p(\tau), \tau) + C(x^p(\tau), \tau)] d\tau \quad (4.22)$$

$$\frac{d^2x^p(t)}{dt^2} + a \frac{dx^p(t)}{dt} = c [F^p(x^p(t), t) + C(x^p(t), t)] \quad (4.23)$$

またそれぞれの関数 F^p, C は以下のようになる．

$$F^p(x^p, t) = c_1(x^p(T^p(t)) - x^p) \quad (4.24)$$

$$C^p(x^p, t) = c_2(x^{Q(t)}(T^o(t)) - x^p) \quad (4.25)$$

また，2 階微分方程式で表される連続時間系モデルの状態変数表現を， $u^p(t) = x^p(t)$ ， $v^p(t) = du^p(t)/dt + au^p(t)$ とおいて導入すると，離散時間系に対応した連続系の内部状態表現モデル，

$$du^p(t)/dt = -au^p(t) + v^p(t) \quad (4.26)$$

$$dv^p(t)/dt = c[F^p(u^p(t), t) + C(u^p(t), t)] \quad (4.27)$$

を得ることができる．

次に提案手法であるハイブリッド PSO について解説する．PSO の応用法である連続時間 PSO アルゴリズムの応用法であり，勾配情報を加えることにより，精密な探索を行うことを狙いとしている．

(式 (4.22), 式 (4.23)) に勾配情報を加えるとそのモデルは，

$$\frac{dx^p(t)}{dt} = c \int_0^t e^{-a(t-\tau)} [F^p(x^p(\tau), \tau) + C(x^p(\tau), \tau) - \nabla E(x^p(\tau), \tau)] d\tau \quad (4.28)$$

$$\frac{d^2x^p(t)}{dt^2} + a \frac{dx^p(t)}{dt} = c [F^p(x^p(t), t) + C(x^p(t), t) - \nabla E(x^p(t), t)] \quad (4.29)$$

またそれぞれの関数は以下ようになる．

$$F^p(x^p, t) = c_1(x^p(T^p(t) - x^p)) \quad (4.30)$$

$$C^p(x^p, t) = c_2(x^{Q(t)}(T^o(t) - x^p)) \quad (4.31)$$

$$\nabla E(x^p, t) = c_3 \frac{\partial E(x^p, t)}{\partial x^p} \quad (4.32)$$

解説したままのモデルでは無制約なので，制約条件に対応したモデルである上下限制約連続時間 PSO モデルを以下の式に示す．上下限制約付最適化問題

$$\min E(x) \quad (4.33)$$

$$\text{subj. } top_i \leq x_i \leq q_i, i = 1, \dots, n \quad (4.34)$$

を直接解くために，この上下限制約領域内に問題の変数を変換して無制約化した新たな変数空間に無制約 PSO モデルを適用した「変数変換モデル」を導入する．非線形変数変換モデルを作成するために，

$$x_i = f_i(y_i) = \frac{q_i + p_i \exp(-y_i)}{1 + \exp(-y_i)} \quad (4.35)$$

とおく．この変換式を制約条件付き問題に代入して変数 x を消去すると，

$$\min E(f(y)) \quad (4.36)$$

を得ることができる．よって式 (4.29) に対応させると，

$$\frac{d^2 y^p(t)}{dt^2} + a \frac{dy^p(t)}{dt} = c[F^p(y^p(t), t) + C(y^p(t), t) - \nabla E(y^p(t), t)] \quad (4.37)$$

またそれぞれの関数は以下ようになる．

$$F^p(y^p, t) = c_1(y^p(T^p(t) - y^p)) \quad (4.38)$$

$$C^p(y^p, t) = c_2(y^{Q(t)}(T^o(t) - y^p)) \quad (4.39)$$

$$\nabla E(y^p, t) = c_3 \frac{\partial E(y^p, t)}{\partial y^p} \quad (4.40)$$

次にプログラムへの実装を考えた時に，連続式のままではプログラムに実装することが難しいので，オイラー法を用いて連続式を離散化し非線形変数変換モデルの離散化 PSO を作成．それぞれに対応する式を以下に示す．

$$u^p(k+1) = (1 - a\Delta T)u^p(k) + \Delta T v^p(k) \quad (4.41)$$

$$v^p(k+1) = v^p(k) + c\Delta T[F^p(u^p(k), k) + C(u^p(k), k) - \nabla E(u^p(k), k)] \quad (4.42)$$

$$F^p(k, k) = c_1(u^p(l^p(k)) - u^p(k)) \quad (4.43)$$

$$C^p(k, k) = c_2(u^{Q(k)}(l^o(k)) - u^p(k)) \quad (4.44)$$

$$\nabla E(k, k) = c_3 \frac{\partial E(k, t)}{\partial k} \quad (4.45)$$

$$l^p(k) = \text{argmin}(E(x^p(l)) | l = 0, \dots, k) \quad (4.46)$$

$$(Q(k), l^o(k)) = \text{argmin}(E(x^q(l)) | q = 1, 2, \dots, P, l = 0, 1, \dots, k) \quad (4.47)$$

$$x_i^p(k) = f_i(u_i^p(k)) = \frac{q_i + p_i \exp(-u_i^p(k))}{1 + \exp(-u_i^p(k))}, i = 1, \dots, n \quad (4.48)$$

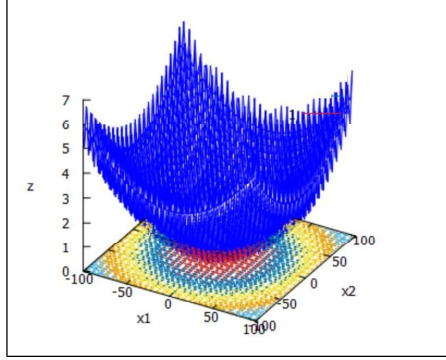


図 4.2: Griewank³⁰

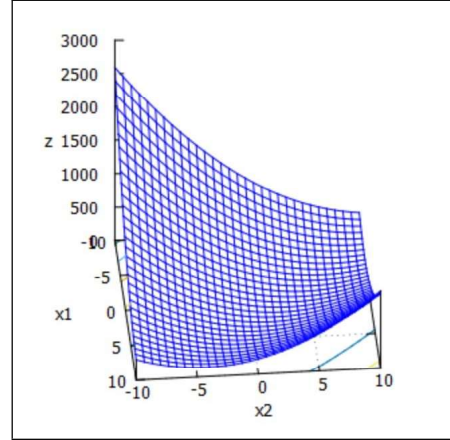


図 4.3: Booth function³¹

提案手法の有効性を示すため多峰性ならびに非常に多くの局所解をもつ評価関数として Griewank function³⁰(図 4.2 参照)(式 (4.49), 式 (4.50)), 単峰性を持つ関数として Booth function³¹(図 4.3 参照)(式 (4.51), 式 (4.52)) を用いて数値実験を行う。

$$f(x_1, \dots, x_n) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (4.49)$$

$$-600 \leq x_i \leq 600, \quad f_{\min}(0, \dots, 0) = 0 \quad (4.50)$$

$$f(x_1, x_2) = (x_1 + 2x_2 + -7)^2 + (2x_1 + x_2 - 5)^2 \quad (4.51)$$

$$-10 \leq x_1, x_2 \leq 10, \quad f_{\min}(1, -3) = 0 \quad (4.52)$$

以下に従来法と提案手法を比較した物を示す(図 4.4, 4.5 参照)。各パラメータの値は N (粒子の数) = 10, $C=1.0$, $c_1 \cdot c_2=1.4$, $c_3=0.1$, $a=1.0$, $\Delta T = 0.9$ というように与えた。また試行回数は 100 とした。Griewank と Booth の結果を比較すると Griewank の場合では従来法と提案手法ともに最適値に収束していない。一方 Booth の結果は従来法では最適値に収束していないのに対して、提案手法では最適値に試行回数 40 以内に収束している。

よって提案手法は多峰性をもつ関数の場合にあまり有効ではないが、単峰性をもつ関数の場合には有効に働くことを示すことができた。

また Booth function の実行時に粒子の数を $N=100$ に変更した場合の結果(図 4.6 参照)も示すとともに、制約条件として Booth function の実行時に $-10 < x < -5$, $-10 < y < -5$ の制約条件を付けくわえたものの結果(図 4.7)も示す。この制約条件の範囲を付け加えた場合の最適値の座標は図 4.3 より $(-5, -5)$ になる。そのときの最適値は式 (4.53) より 884 となり、図 4.7 と一致し、制約条件が働いていることがわかる。

$$\begin{aligned} f(-5, -5) &= (-5 - 10 - 7)^2 + (-10 - 5 - 5)^2 \\ &= 884 \end{aligned} \quad (4.53)$$

³⁰<https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda> griewank-function

³¹<https://qiita.com/tomitomi3/items/d4318bf7afbc1c835dda> booth-function