勾配情報にもとづく局所探索を組み込んだ ハイブリッド粒子群最適化

富山県立大学工学部電子情報工学科　　　　　　指導教員：奥原浩之
1515050　山本聖也

1　はじめに

粒子群最適化 (Particle Swarm Optimization; PSO) は，群の中の固体（粒子）が持つ最良の情報とそのグループの最適値から過去の探索から考慮した確率的最適化手法であり， ケネディ[1] が社会的行動に基づいて開発した並列進化計算技術である． 社会的方法と計算方法の両方を扱う PSO に関する標準的な研究がある [2].

近年，コンピュータサイエンスの発展は，ハードウェアとソフトウェアの有効性が顕著に表れている．その中で大規模問題の最適化の重要性はますます高めっている．ソーシャルネットワークサービスの登場により，ログやパスの問題も大規模になっている．最新のコンピュータでこれらの問題を解決するには時間がかかってしまう．

本研究では数ステップでもっとも最適な解が見つかる新しいハイブリッド動的システムを提供する．そこで連続 PSO アルゴリズムに勾配法を組み込み，定式化することができる．そこで，提案した手法の有効性を示すことが本研究の目的となっている．

2　PSO の概要

2.1　PSO アルゴリズム

PSO は群をなして移動する生物の行動を模範したアルゴリズムである．群をなす生物をモデル化し，粒子は最適化問題における候補解を示している．PSO は群の中の粒子がもつ最良の情報 (pbest) とその集団の最適値 (gbest) から過去の探索を考慮し，さらにその集団の各粒子の位置および速度を更新することによって計算される．以下に PSO の解説を示す (図 1 参照).

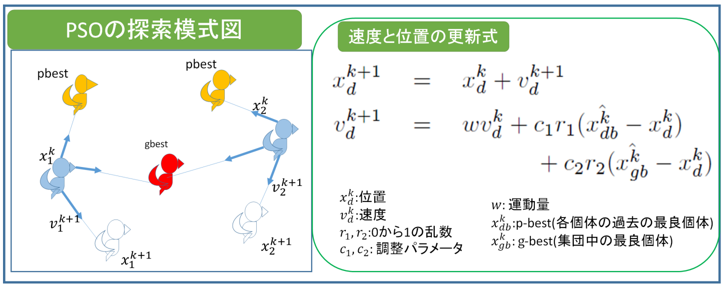

図 1　PSO の概要

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 $c_2r_2(x_{db}^k - {x_d}^k)$，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 [3]. 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 [2]. 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 (図 2 参照).

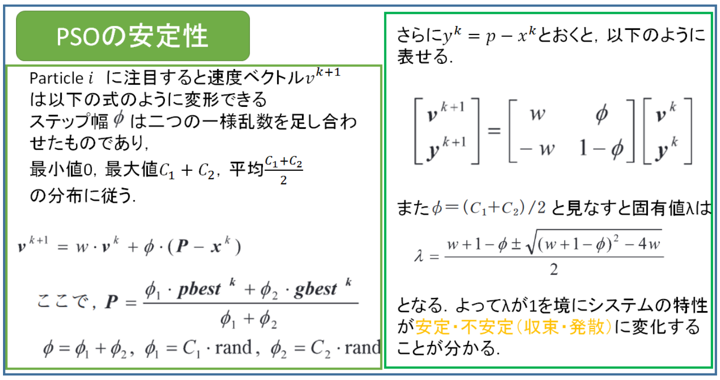

図 2　PSO の安定性

2.2　連続時間 PSO アルゴリズム

連続時間 PSO アルゴリズムについて述べる．
連続時間 PSO で用いるベクトルおよび行列を以下のように定義する (図 3 参照).

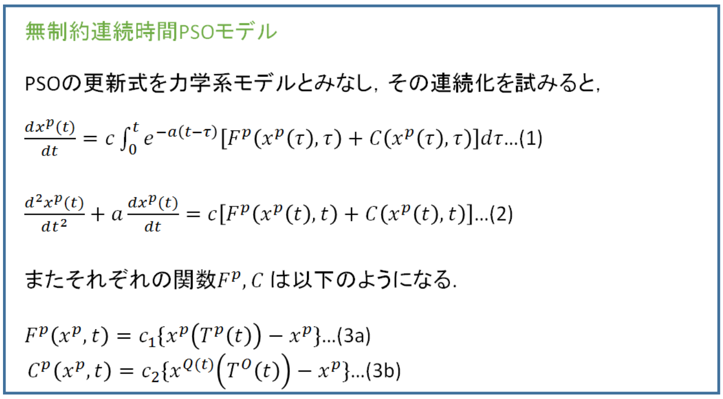

図 3　連続時間 PSO モデル

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，
$du^p(t)/dt = -au^p(t) + v^p(t)$
$dv^p(t)/dt = c[F^p(u^p(t),t) + C(u^p(t),t))]$
を得ることができる．

2.3　PSO の探索能力の向上

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 [5, 6]. 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，以下のようなセンサリング・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 $(\xi k, \eta k)$ 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．

3　ハイブリッド PSO の提案

本節では提案手法であるハイブリッド PSO について解説する（図 4 参照）．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．

無制約連続時間PSOモデル

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，

$$\frac{dx^p(t)}{dt} = c\int_0^t e^{-a(t-\tau)}[F^p(x^p(\tau),\tau) + C(x^p(\tau),\tau) - \nabla E(x^p(\tau),\tau)]d\tau \ldots(1)$$

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

またそれぞれの関数$F^p, C, \nabla E$ は以下のようになる．

$$F^p(x^p,t) = c_1\{x^p(T^p(t)) - x^p\} \ldots(3a)$$
$$C^p(x^p,t) = c_2\{x^{Q(t)}(T^O(t)) - x^p\} \ldots(3b)$$
$$\nabla E(x^p,t) = c_3\frac{\partial E(x^p,t)}{\partial x^p} \ldots(3c)$$

図 4　ハイブリッド PSO の解説

図 4 で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を以下に示す (図 5 参照)．

上下限制約連続時間PSOモデル

非線形変数変換モデルを作成するために，

$$x_i = f_i(y_i) = \frac{q_i + p_i\exp(-y_i)}{1+\exp(-y_i)} \ldots(4)$$

とおく．
この変換式を制約条件付問題に代入して変数 x を消去すると，

$$\min_y E(f(y)) \ldots(5)$$

が得られる，よって(2)式に対応すると，

$$\frac{d^2y^p(t)}{dt^2} + a\frac{dy^p(t)}{dt} = c[F^p(y^p(t),t) + C(y^p(t),t) - \nabla E(y^p(t),t)] \ldots(5)$$

$$F^p(y^p,t) = c_1\{y^p(T^p(t)) - y^p\} \ldots(6a)$$
$$C^p(y^p,t) = c_2\left\{y^{Q(t)}(T^O(t)) - y^p\right\} \ldots(6b)$$
$$\nabla E(y^p,t) = c_3\frac{\partial E(y^p,t)}{\partial y^p} \ldots(6c)$$

図 5　制約条件

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，それぞれに対応する式を以下に示す (図 6 参照)．

非線形変数変換モデルの離散化PSOモデル

上下限制約モデルからオイラー法で離散化し，対応する式も加える．

$$u^p(k+1) = (1-a\Delta T)u^p(k) + \Delta T v^p(k) \ldots(7a)$$
$$v^p(k+1) = v^p(k) + c\Delta T[F^p(u^p(k),k) + C(u^p(k),k) - \nabla E(u^p(k),k)] \ldots(7b)$$
$$F^p(k,k) = c_1\{u^p(l^p(k)) - u^p(k)\} \ldots(7c)$$
$$C^p(k,k) = c_2\{u^{Q(k)}(l^O(k)) - u^p(k)\} \ldots(7d)$$
$$\nabla E(k,k) = c_3\frac{\partial E(k,t)}{\partial k} \ldots(7e)$$
$$l^p(k) = \underset{l}{\mathrm{argmin}}\{E(x^p(l)) | l = 0,\ldots,k\} \ldots(7f)$$
$$\left(Q(k), l^O(k)\right) = \arg\min_{(q,l)}\left\{E\left(x^q(l)\right) \middle| q = 1,2,\ldots,P, l = 0,1,\ldots,k\right\} \ldots(7f)$$
$$x_i^p(k) = f_i\left(u_i^p(k)\right) = \frac{q_i + p_i\exp(-u_i^p(k))}{1+\exp(-u_i^p(k))} \quad i = i,\ldots,n \ldots(7g)$$

図 6　離散化

4　数値実験ならびに考察

今回は一般的な PSO の数値実験を，評価関数として Griewank を用いて行う（図 7 参照）．粒子をランダムに生成した初期状態とループ回数 100 の場合で比較すると，(0, 0, 0) に向けて収束に向かってるように見える．しかしループ回数を重ねても収束しない．

また別の点に収束してしまう場合もあり，PSO はランダム性を含んだプログラムではあるが，安定性に欠ける結果が出た．またパラメータの値は任意で与えるものなので最適のパラメータを決定することは実際のアプリケーションでは難しい．

PSOの実行

評価関数：$f_{Griewank} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{\sqrt{i}}{x_i}\right) + 1$

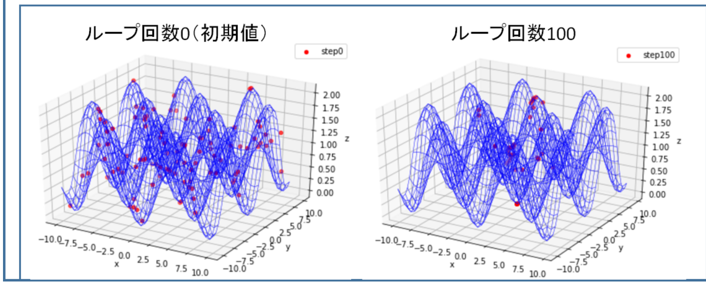

図 7　PSO の実行結果

5　おわりに

本研究では PSO とその応用手法について解説し，その定式化を行った．また，PSO の簡単な数値実験を行った．今後の課題は，IoT のセンサにより収集されたビッグデータからの意思決定に提案手法を適応，また定式化ができたのでプログラムを C 言語で実装する．

参考文献

[1] J. Kennedy, R. C. Eberhart, “Particle swarm optimization”, *IEEE Conf. On Neural Networks, IV, Piscataway, NJ*, pp. 1942-1948 (1995).

[2] J. Kennedy, R.C. Eberhart, Y. Shi, “Swarm intelligence”, Morgan Kaumann Publishers, San Fran-cisco, CA, pp. 1942-1948, 2001.

[3] 石亀敦司, 安田恵一郎, “群れの知能：Particle Swarm Optimization”, 知能と情報（日本知能情報ファジィ学会誌）, Vol. 20, No. 6 , pp. 829-839（2008）.

[4] H. M. Emara and H. A. Abdel Fattah, “Continuous swarm optimization technique with stability analysis”, *Proceeding of the 2004 American Control Conference*, 2811-2817 (2004).

[5] Ryuzaburo SUGINO, Anan National College of Technology: “Numirical Performance of PSO Algorithm Using by Gradient Method”,

[6] M. Jiang, Y. P. Luo and S. Y. Yang, “Stochastic convergenceanalysis and parameter selection of the standard particle swarm optimization algorithm”, *Information Processing Letters* vol. 102, No. 1, pp. 8-16 (2007).