

## 群知能アルゴリズムを応用した脚歩行ロボットの研究 Research of the leg walking robot adapting a swarm intelligence algorithm

○佐藤雄貴<sup>1</sup>, 奥山寿康<sup>1</sup>, 猪原知俊<sup>2</sup>, 鈴木 虹太<sup>3</sup>, 入江寿弘<sup>4</sup>Yuki Sato<sup>1</sup>, Hisayasu Okuyama<sup>1</sup>, Tomotoshi Inohara<sup>2</sup>, Kouta Suzuki<sup>3</sup>, Toshihiro Irie<sup>4</sup>

In recent years, various algorithms are studied and it is used for a numerical optimization problem. Then, we use algorithms for optimization of a walk of a leg mechanism robot. We examined last year whether only an ABC algorithm would be applicable. In addition, we also consider this year employment of the particle swarm optimization algorithm announced by J.Kennedy and R.Eberhart in 1995.

### 1. はじめに

近年様々なアルゴリズムが研究され数値最適化問題に使用されている。その中でも本研究では脚機構ロボットの歩行の最適化に使用する。昨年はABCアルゴリズムを応用可能か検討したが、今年度はそれに加え、J. Kennedy 氏と R. Eberhart 氏が 1995 年に発表した粒子群最適化(PSO: Particle Swarm Optimization)アルゴリズムの運用を検討する。

### 2. 歩行方法

本研究では以下の①～⑥の動作を繰り返すことで歩行を試してみる。Figure 1 にその流れを示す。

- ① 重心を右に移動し、左足を持ち上げ半歩前に出す
- ② 左足をさらに半歩前に出す
- ③ 右足を半歩後ろに下げ左足を着地させる
- ④ 重心を左に移動し、右足を持ち上げ半歩前に出す
- ⑤ 右足を更に半歩前に出す
- ⑥ 左足を半歩後ろに下げ右足を着地させる

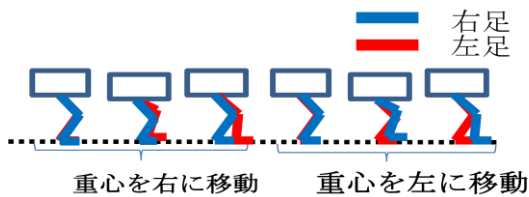


Figure 1. Walking pattern

この歩行動作は以下の値によって定まる。

[機体の質量], [歩幅], [歩行周期], [左右の重心移動量], [足を持ち上げる高さ]

それぞれのパラメータの最適な組み合わせを実現した値を探索し、歩行動作を安定化することがシミュレーションの本旨である。

### 3. 最適化手法

このシミュレーションでは歩行パラメータの組み合わせが重要となる。そこで PSO アルゴリズムという群知能アルゴリズムを用いて最適化を試みる。これは自然界の鳥や魚の群れの振る舞いを模倣した最適化アルゴリズムであり、数値最適化問題などに使用される。鳥や魚は仲間が良い餌場を見つけると、次第にそこに集まって周辺を探索する。同時に、各個体が探索した中での良い場所も覚えており、その周辺に戻る動きも見せる。最も良い場所は一箇所しかない反面、比較的良好な場所は多数存在する場合もある。また良い場所の周辺は良い場所である可能性が高い。これらの各個体を粒子で置き換え、アルゴリズムとして表現したものである。以下に PSO アルゴリズムの手順を示す。

制約条件を満たす粒子を既定の群サイズ分ランダムに生成し、目的関数により群の各粒子の評価値を求める。群の各粒子の評価値を用い、粒子を設計変数空間内の新たな位置に再配置する。ここで再度、目的関数により群の各粒子の評価値を求める。以上から⑤探索の過程で得た群と各粒子の最良解を求め、定めた終了条件に基づき、終了を判定する。条件を満たしていなければ、に戻り繰り返す。

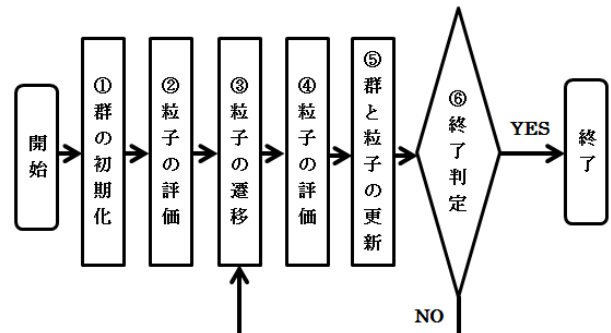


Figure 2. The procedure of PSO algorithm

1 : 日本大学・学部・精機 2 : 日本大学・院 (前)・精機 3 : 日本大学・院 (前)・医福

4 : 日本大学・教員・精機

また、PSO の粒子についての更新方程式は以下のようになる。

$$\begin{aligned} \mathbf{v}(t+1) &= (\mathbf{w} \times \mathbf{v}(t)) + (c1 \times r1 \times (\mathbf{p}(t) - \mathbf{x}(t))) \\ &+ (c2 \times r2 \times (\mathbf{g}(t) - \mathbf{x}(t))) \end{aligned} \quad (1)$$

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t+1) \quad (2)$$

- $\mathbf{v}$  : 粒子の速度
- $\mathbf{x}$  : 粒子の現在位置
- $\mathbf{p}$  : これまで検出した中で最適な粒子の位置
- $\mathbf{g}$  : 群中の粒子によって検出された中で最適な位置
- $w$  : 慣性質量
- $c1$  : 粒子のローカル質量
- $c2$  : 群のグローバル質量
- $r1$  : 粒子の確率変数
- $r2$  : 群の確率変数

#### 4. PSO による最適化の留意点

関数  $f = \sin(x_0) + \sin(x_1)$  ( $-4 \leq x \leq 4$ ) の解が取り得る値を三次元空間に図示したものを Figure 3 に示す。この関数の最小値を探索するプログラムを実際に動かして最適化を試みたところ、収束する値として様々な値を取った。このことは探索が比較的良好な値の周辺に集中してしまう場合があり、探索範囲のランダム性を確保できなくなってしまうために起こる。この時の解を局所解と呼ぶが、その対応として、粒子の数、粒子への評価の基準、探索範囲について調整することで局所解に陥る頻度を緩和することができた。

ただし、この場合は仮の関数についての探索を行ったものであり、解が自明であったことから局所解に対する方策を考えるまでに至ったものである。しかし、実際には未知の解を探索するため、局所解としての認識を得ずに探索を実行しなければならない。従って、探索する関数の解の形状についての認識を深め、適切な設定を施した粒子により試行を繰り返すことが望ましい。

例えば、単調な関数の探索では、局所解が存在しないので粒子の数を多くする必要は無い。多峰性の関数の場合、局所解に陥らないために、多くのポイントを探索する必要があるため粒子の数を多くする必要がある。

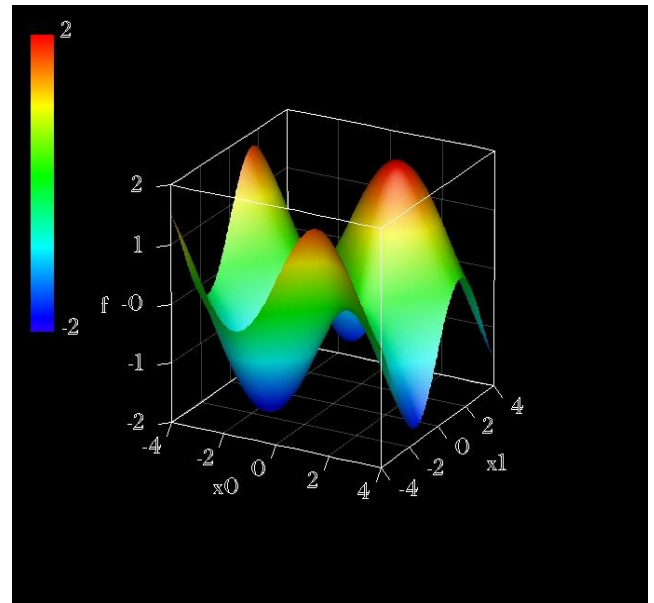


Figure3. The solution of  $f = \sin(x_0) + \sin(x_1)$  ( $-4 \leq x \leq 4$ )  
 ※f の値が大きいほど赤くなり、小さいほど青くなる。

Table 1. The solution by the search which changed the number of particles

粒子の数	2	3	4	5	6	7
$f(x_0, x_1)$	最適解	局所解	局所解	局所解	局所解	局所解
粒子の数	8	9	10	20	30	40
$f(x_0, x_1)$	最適解	最適解	最適解	最適解	最適解	最適解
粒子の数	50	60	70	80	90	100
$f(x_0, x_1)$	最適解	最適解	最適解	最適解	最適解	最適解

PSO の粒子の数を変化させて関数  $f = \sin(x_0) + \sin(x_1)$  ( $-4 \leq x \leq 4$ ) の最小値を探索した結果を上 Table 1 に示した。最適解は探索した関数により自明で  $f = -2.0$  であるが、Table 1 より粒子の数が少ないと局所解に陥る頻度が高いことが見て取れる。粒子の数を増やしていくと、安定して最適解を導いた。

#### 5. まとめ

PSO は探索する関数によって粒子の数や評価、探索範囲を設定する必要がある。このことを踏まえ、歩行動作に必要な値を組み合わせた関数について正しく認識し、適切に PSO アルゴリズムを運用していくことが重要である。

#### 6. 参考文献

[1] James Kennedy and Russell Eberhart : Particle Swarm Optimization. Proceedings of IEEE the International Conference on Neural Networks, 1995