舞鶴工業高等専門学校 電子制御工学科　卒業研究論文 (平成 30 年 2 月 ○○ 日提出)

○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○に関する研究

穂本 創一 (**5069**)

平成 30 年度

卒 業 研 究 論 文

論題： ○○○○○○○○○○○○○○○○○○○○ ○○○○○○○○○○○に関する研究

指導教員	伊藤 稔 教授

舞鶴工業高等専門学校 電子制御工学科	
提出者	穂本 創一 (8101)
提出日	平成 31 年 2 月 ○○ 日

Department of Control Engineering,
National Institute of Technology, Maizuru College

○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○に関する研究

指導教員　：伊藤 稔 教授

提出者　　：穂本 創一 (8101)

平成 30 年 2 月 ○○ 日

舞鶴工業高等専門学校 電子制御工学科

論 文 要 旨

ここには論文要旨を書きます.

目 次

第 1 章　はじめに

最適化の分野において 1980 年代後半に “メタヒューリスティックス” と呼ばれる新たなパラダイムが生まれた．この分野は，コンピュータの性能の飛躍的な向上に伴い計算可能な量がかくだいしたこと，実在するシステムの大規模化・複雑化によって実用上十分な解を求めることが可能な近似に対する要求が高まったことから注目を浴びた．

メタヒューリスティクスの特徴として，物理現象，生物現象，あるいは生物の集団・社会現象などを類推し提案されたものが多く代表的な例として，遺伝的アルゴリズム（Genetic Algorithm:GA），アントコロニー最適化（AntColony Optimization），Particle Swarm Op-timization（PSO），差分進化（Differential evolution:DE）が挙げられる．　メタヒューリスティクスので対象とするのは，最適化もんだいである．最適化問題では，D 次元の実数ベクトル $x = (x_1, \cdots, x_D)^T$ について，目的関数 $f(\boldsymbol{x}) \in R$ を最小化（あるいは最大化）する最適解 $\boldsymbol{x}$ を求める．工学分野において現れる実問題は，関数最適化に定式化される．目的関数の勾配情報や概形が単峰性か多峰性か，変数分離の可否などの事前情報が与えられることなく目標関数 $f(\boldsymbol{x})$ の近似解を求める．このような問題を，black-box-optimization と呼ぶ．そのため，複数の解を保持している関数の現在の解がどの程度良い解なのかを評価できれば，解の探索が可能となる．

第 2 章　差分進化

2.1　進化計算

進化計算（Evolutionary Computation）は，対象とする問題を疑似的に生物個体とみなし，その集団を用いて解探索を行う．たとえば，遺伝的アルゴリズムでは個体の遺伝子を多変数関数の変数の状態と解釈し，

(1)　遺伝子をそれぞれの変数に代入し，生物の個体を生成する．

(2)　それらの生物の評価を行う．

(3)　評価の高いものを生存させ，個体間での交叉を行う．さらに，突然変異を発生させ次世代の染色体とする．

という，手順を繰り返す．進化計算では（3）の交叉と突然変異をどのような手法を用いて行うかによって名称が異なる．それぞれの技法でもパラメータが存在し，パラメータによって探索速度や短峰性，多峰性の探索性能に変化が生じる．　一般に進化計算では集団に属する個体各々の適用度によって評価を行い，この値のみを用いて探索する．

2.2　理論

差分進化（Differential Evolution:DE）は R.Storn と K.Price によって 1995 年に提案された実数空間における最適化アルゴリズムである．DE は非線形問題，微分不可能な問題，非凸問題，多峰性問題など様々な最適化問題に適応されている．DE の評価は進化計算の主要な国際会議の 1 つである IEEE Congress on Evolutionary Computation（CEC）にて良好な性能を示している．

一方，DE の弱点としてはほかの最適化アルゴリズムと同様に万能でない．そのため，関数の特性によってパラメータの影響を受ける．

DE はいくつかの手法が提案されており，それぞれ突然変異の部分に用いる数式や交叉の方法に違いがある．DE は DE/base/num/cross という形式で表記され，base は交叉の基本となる親の選択方法を示している．best であればその世代で最も評価の高い個体のベクトルを用いて交叉を行う．rand であれば集団の中からランダムに抽出して交叉する．num は基本ベクトルに変化を加える差分ベクトルの数を表現している．cross は子を生成するための交叉方法を示している．binomial crossover（bin）では一定の確率で前回の値をそのまま引き継ぐ．exponential crossover（exp）では指数関数的に交叉を発生させる確率を減少させて前回の値を引き継ぐ．

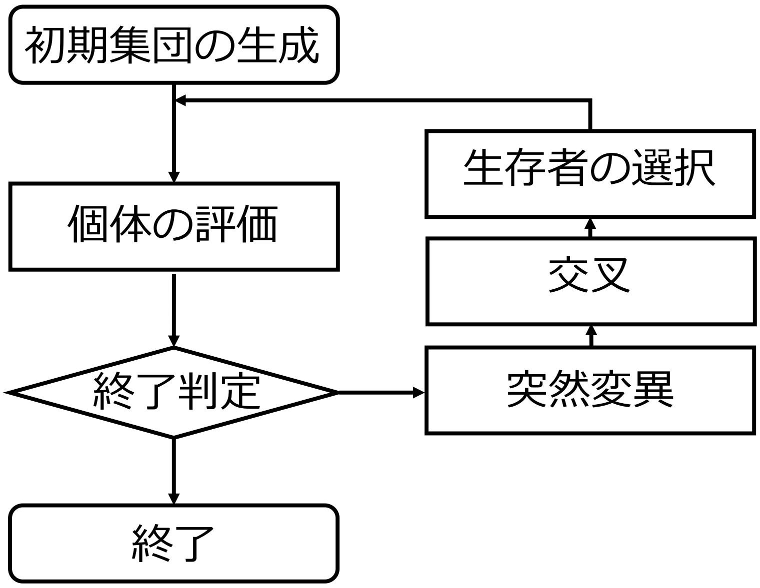

図 **2.1**　DE のアルゴリズムのフローチャート

2.3　アルゴリズム

2.3.1　初期集団の生成

第 3 章　参考文献，その他

参考文献[2)–4)] です．参考文献 2) です．

丸文字やリターンキーは ①， $\boxed{\hookleftarrow}$ ， $\hookleftarrow$ のようにして書けます．

参考文献

1) 島ほか：非線形システム制御論，コロナ社 (1997)

2) 川田，島津，井上：Hamilton-Jacobi 方程式に基づく非線形 $\mathcal{H}_\infty$ 制御の近似実現，システム制御情報学会論文誌，Vol. 11，No. 7，pp. 401–410 (1998)

3) A. J. van der Schaft: $\mathcal{L}_2$-gain Analysis of Nonlinear Systems and Nonlinear State Feedback $\mathcal{H}_\infty$ Control, *IEEE Trans. Automat. Contr.*, Vol. AC-37, No. 6, pp. 770–784 (1992)

4) 中村：二次安定化による倒立振子システムのロバスト制御に関する研究，立命館大学理工学部卒業論文 (1997)

謝 辞

謝辞はここに書きます.

付 録

A.1　付録の式番号

付録の式番号は

$$\int_0^\infty \| \boldsymbol{x}(t) \|^2 dt < \infty \tag{A.1}$$

のように区別してください.

A.2　付録の図番号，表番号

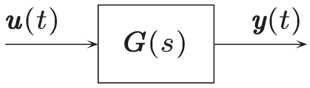

図 **A.1**　線形システム

付録の図番号や表番号も図 **A.1** のように本文と区別してください.

プログラムリストは**リスト A.1** のように書きます.

リスト A.1 (prog.m：フルビッツの安定判別法)

```
%% prog.m

clear
format compact

syms s
syms kP kI kD real
syms a0 a1 b0 real

P = b0/(s^2 + a1*s + a0);
C = (kD*s^2 + kP*s + kI)/s;

[Np Dp] = numden(P);
[Nc Dc] = numden(C);

Delta = Dp*Dc + Np*Nc;
Delta = collect(Delta,s)
alpha = coeffs(Delta,s);

N = length(alpha);
n = N - 1;

% ===== 条件 A ======================
disp('----- 条件 A：a_i > 0 ------')
for i = 1:N
  str = ['a', num2str(i-1), '= alpha(i)'];
  eval(str)
end

```

```
cond1 = ' ';
for i = 1:N
  if i == 1
    cond1 = strcat(cond1,['simplify(a' num2str(i-1) '> 0)']);
  else
    cond1 = strcat(cond1,[' & simplify(a' num2str(i-1) '> 0)']);
  end
end

% ===== 条件 B" ======================
for i = 1:n
  for j = 1:n
    k = (N - 1) + (i - 1) - 2*(j - 1);

    if k >= 1 & k <= N
      H(i,j) = alpha(k);
    else
      H(i,j) = 0;
    end
  end
end

disp('----- 行列 H ------')
H

if mod(n,2) == 0  % 次数: n = 2*k
  i_min = 3;  i_max = n - 1;
else              % 次数: n = 2*k + 1
  i_min = 2;  i_max = n - 1;
end

disp('----- 条件 B"：H_i > 0 ------')
for i = i_min:2:i_max
  str = ['H', num2str(i), '= det(H(1:i,1:i))'];
  eval(str)
end

cond2 = ' ';
for i = i_min:2:i_max
  if i == i_min
    cond2 = strcat(cond2,['simplify(H' num2str(i) '> 0)']);
  else
    cond2 = strcat(cond2,[' & simplify(H' num2str(i) '> 0)']);
  end
end

% ===== 安定条件 ======================
disp('----- 安定条件 ------')
simplify(eval(cond1) & eval(cond2))
```