5K-1

遺伝的プログラミングにおける多段階探索交叉の検証

細川 長洋 † 花田 良子 † 棟安 実治 †
† 関西大学システム理工学部

1 はじめに

グラフ構造など組合せ的な構造を持つ最適化問題を遺伝的アルゴリズムのような進化計算で解く際には両親の形質を受け継ぐような問題固有の構造，性質を利用した交叉の設計が重要であり，これまで多くの手法が提案されてきた．交叉 deterministic Multi-Step Cross over Fusion (dMSXF) [1] は近傍探索のメカニズムを利用した形質遺伝に優れた交叉手法であり，巡回セールスマン問題など種々のグラフ問題において非常に強力な探索性能を有している．本研究では交叉 dMSXF を遺伝的プログラミング (Genetic Programming: GP) [2] における交叉に採用し，両親の部分木構造受け継いだ多様な子個体を生成する．関数同定問題に適用し，その解探索性能を検証する．

2 dMSXF

池田らによって提案された dMSXF [1] は親 p_1 から親 p_2 に向けて局所探索を行うことで，両親の形質の受け継ぎ方が多様な子個体群を生成する．dMSXF のアルゴリズムを以下に示す．親 p_1, p_2 から生成される子個体群を $C(p_1, p_2)$ と表す．また図 1 に dMSXF の探索の様子を示す．

【dMSXF のアルゴリズム】

Step 0 p_1，p_2 を両親，その子個体群 $C(p_1, p_2) = \phi$ とする．

Step 1 探索初期点 $x_1 = p_1$，k=1 とし，x_1 を $C(p_1, p_2)$ の要素として加える．

Step 2 ステップ k における探索点 x_k の近傍個体を μ 個生成し，その集合を $N(x_k)$ とする．ただし，$N(x_k)$ のすべての近傍解 $y_i (0 < i < \mu)$ はかならず $d(y_i, p_2) < d(x_k, p_2)$ を満たさなければならない．

Step 3 $N(x_k)$ の中で最も良い解 y を選択する．$x_{k+1} = y$ とし，x_{k+1} を $C(p_1, p_2)$ の要素として加える．

Step 4 $k = k + 1$ とし，$k = k_{max}$ あるいは x_k が p_2 に等しくなれば終了．そうでなければ，Step 2 にもどる．

Step 2 において，暫定解 x_k から生成する近傍の解 y_i を解 x_k よりも p_2 に近い個体に制限し，x_{k+1} が x_k よりも劣っていたとしても必ず探索を進めることで解遷移を決定的に行う．dMSXF で必要なパラメータはステップ数 k_{max}，1 ステップの近傍数 μ であり，最大で $k_{max} \times \mu$ 個の個体が生成される．

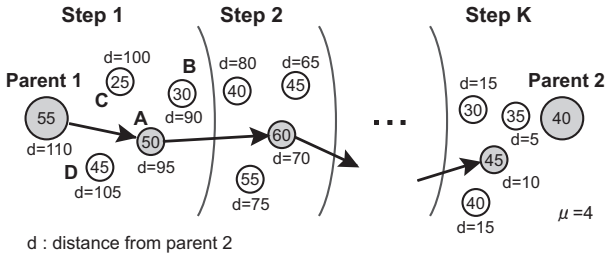

図 1: dMSXF の探索の様子

3 GP における dMSXF

dMSXF を GP に応用する際には，木構造における距離尺度および近傍解の生成方法を定める必要がある．以下にそれぞれの定義を示す．

3.1 2 つの木の間の距離

dMSXF を木構造を扱う GP に応用する際には 2 つに木の間において距離を定義する必要がある．本研究では比較する木を u，v としたとき，u，v においてノードの種類を無視した最大共通部分 $Com(u, v)$ を抜き出し [3]，そのなかでノードの種類が異なる数を $N_{com}(u, v)$ とする．また，共通部分として抜き出されなかった部分木のノード $U_v(u)$，$V_v(v)$ の数をそれぞれ $|U_v(u)|$，$|V_v(v)|$ と表し，u，v 間の距離 $d(u, v)$ を次のように定義した．

$$d(u, v) = N_{com}(u, v) + \frac{|U_v(u)| + |V_v(v)|}{2} \qquad (1)$$

3.2 近傍解の生成方法

dMSXF のステップ k における暫定解 x_k の近傍解 y_i を，$N_{com}(u, v)$，$U_v(u)$，$V_v(v)$ を用いて次の 3 つの処理を確率的に行うことで生成する．これらの操作をすることにより，dMSXF の局所探索の過程で p_2 に似た形質を持つ木を生成することができる．

- $N_{com}(x_k, p_2)$ の内の 1 つのノードを p_2 から x_k にコピー
- $U_{p_2}(x_k)$ におけるノードを 1 つ x_k から削除
- $V_{p_2}(p_2)$ におけるノードを 1 つ x_k に追加

Analysis of effectiveness of multi-step crossover fusion on genetic programming
† Nagahiro HOSOKAWA
† Yoshiko HANADA (hanada@kansai-u.ac.jp)
† Mitsuji MUNEYASU (muneyasu@kansai-u.ac.jp)
Faculty of Engineering Science, Kansai University (†)
3-3-35 Yamate-cho, Suita, Osaka 564-8680 Japan

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.

4 関数同定問題

本研究では，交叉の性能を評価するため，非終端ノードを {+，-. ×，/}，終端ノードを $\{x\}$ とする単純な連続関数の同定問題に適用する．num_x 個の入力点について，各点における誤差の総和を目的関数とし，最小化することで関数を同定する．ここでは，なだらかなランドスケープで同定が比較的簡単な関数 f_1，および同定が比較的難しい多峰の関数 f_2 の 2 種を用いる．いずれも x の定義域は [-1, 1] とし，入力点は区間内に均等に 21 点配置した．

(1) $f_1(x) = x^4 + x^3 + x^2 + x$

(2) $f_2(x) = x^3 + x^{-3} + x^2 + x$

5 数値実験

関数同定問題を用いて，交叉 dMSXF と 1 点交叉 (1X) の解探索性能を比較検証する．1X では生成する子個体を 20 個体，dMSXF については k_{max}=5，μ=20 とした．また，dMSXF の近傍個体生成においてノードの変化，削除，挿入が起こる比率を 3:3:4 とした．世代交代モデルとして，交叉を $p_1 \rightarrow p_2$，$p_2 \rightarrow p_3$，$\cdots$，$p_{N_{pop}} \rightarrow p_1$ のように適用する CCM-relay (Characteristic Collection Model)[1] を用いる．母集団サイズを 16 とし，探索終了世代を 20 とした．

図 2 に簡単な同定問題 f_1 において，探索世代に対する目的関数値および成功率の推移を示す．これらは 50 試行の平均である．図の目的関数値については式 (2) のように正規化した値を示している．式中 $fitness$ は実際に用いた目的関数値 (誤差の総和) であり，f_{nomal} は 1 に近いほど最適解に近いことを意味する．

$$f_{nomal} = \frac{num_x}{num_x + fitness} \tag{2}$$

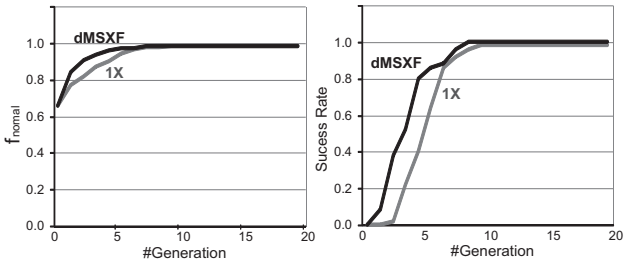

図 2: f_1 における目的関数値と成功率の推移

図 2 より目的関数値でも成功確率でも dMSXF が 1X より優れていることが分かる．また，最適解を得た木の平均サイズは 1X が 25.7，dMSXF が 16.4 であり，dMSXF で得られた木はイントロンが少ないことがわかる．また，dMSXF についてはイントロンが全く含まれない木が得られることも確認している．これは dMSXF が局所探索ベースの交叉であり，急激なプログラム長の変化が起こらないことが要因であると考えられる．

図 3 に複雑な同定問題 f_2 における実験結果を示す．この問題においても dMSXF は 1X よりも目的関数値と成功確率の面で優れていることが分かる．最適解の木のサイズの平均は 1X が 42.9，dMSXF が 25.0 であり，複雑な問題においても dMSXF はコンパクトな木が生成されていることが分かる．なお，図 3 で 1X の正規化した目的関数の値に変化がないのは，目的関数の平均を計算する際に極めて大きな誤差が 1 点でもあると正規化した際に 0 になるためである．

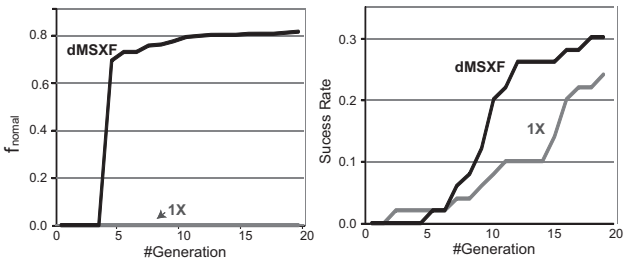

図 3: f_2 における目的関数値と成功率の推移

6 まとめ

本研究では，交叉 dMSXF を GP の交叉に採用し，関数同定問題を用いてその性能の検証を行った．検証を行った結果，1 点交叉と比較して dMSXF は GP において十分な解探索性能を示すことが分かった．また，dMSXF の最適解はイントロンをほとんど含まない，あるいは全く含まないものが得られ，木構造の最適化に有効であることがわかった．しかし，実験で設定した 3 種の近傍生成に関するパラメータが解探索性能に大きく影響を与えるため，問題に適したパラメータの設定が必要である．今後の課題として，近傍生成の適切なパラメータの設定方法の検討，木の成長を解析することにより，dMSXF がブロートを抑制する交叉方法であるか検証することがあげられる．

参考文献

[1] Ikeda, K. and Kobayashi, S.: Deterministic Multistep Crossover Fusion: A Handy Crossover for GAs, Proc. of Parallel Problem Solving fron Nature, PPSN VII, pp. 162–171 (2002)

[2] Jon R. Koza Genetic Programming: On the Programming of Computer by Means of Natural Selection, MIT Press, 1992.

[3] 増田 澄男, 森 一郎, 田中 栄一 : 二つの木のの最大共通部分を求めるアルゴリズム: 電子通信学会論文誌 A. vol. J77-A，No.3，pp. 460–470，1994.

Copyright ©2011 Information Processing Society of Japan.
All Rights Reserved.