木構造断片のランダムサンプリングによるプログラム進化

Program Evolution by Random Tree Sampling

丹治 信　　伊庭 斉志
Makoto Tanji　　Hitoshi Iba

東京大学工学系研究科
Graduate School of Engineering, The University of Tokyo

In this paper, we describe a new method for program evolution named PORTS (Program Optimization by Random Tree Sampling) which is motivated by the idea of preservation and control of tree fragments. We hypothesize that to reconstruct building blocks efficiently, tree fragments of various sizes should be preserved into the next generation, according to their differential fitnesses. Instead of traditional subtree crossover and mutation, PORTS creates new individuals by sampling from the promising trees by traversing and transition between trees. Because the size of a fragment preserved during a generation update follows a geometric distribution, merits of the method are that it is relatively easy to predict the behavior of tree fragments over time and to control sampling size, by changing a single parameter. Our experimental results on the Royal Tree and Symbolic Regression problems show that the performance of PORTS is competitive with SGP (Simple Genetic Programming). We observed that there is a significant difference of fragment distribution between PORTS and SGP.

1. はじめに

進化計算による最適化では，集団中の優良解を選択する”選択”と，得られた優良解の交差や突然変異で解空間の新しい部分を探索する”組み換え”の二つの要素により解空間の探索が進められていく．従来の GP(Genetic Programming) では，”組み換え”オペレータとして GA(Genetic Algorithm) からの自然な拡張として部分木単位の交叉と突然変異が広く使われてきた [Koza 92]．しかし，生物の DNA と似た一次元構造を持つ GA が生物進化のアナロジーから得た交叉と違い，プログラムの最適化や合成のために木構造を扱う GP では部分木の交叉の直接的な意味は明らかではない．また，近年では文法型の GP や進化計算の探索を解の確率分布の推移として捉える考え方が提案されており，様々な探索手法が提案されている．

従来の部分木の交叉では，選ばれた部分木とその残りの木構造が次世代へそのままの形で保存される．本研究では部分木単位ではなく，木構造断片が次世代へどのように保存されるかという観点から，GP の木構造をランダムにサンプリングすることで新しい個体を生成する手法 PORTS(Program Optimization byRandom Tree Sampling) を提案する．手法の詳細については [Tanji 09] を参照されたい．

本稿では GP の探索手法の指針として，効率的な探索のために以下の 2 点が必要であると仮定する．

- 解の多様性維持のため，様々なサイズの木構造断片を評価値に応じて次世代へ保存する．
- 木構造断片の保存される分布を調節可能にする．

2. PORTS

PORTS は GP の世代交代の際に保存される木構造断片の分布に着目し，それを調節可能にすることで解を探索する手法である．ここで，木構造断片とは GP の個体の木構造に含まれる任意のリンクを持った部分構造である．従来の GP との違いは交叉・突然変異の代わりに以下で述べるランダムサンプリング手法を使って解を探索する点にある．

2.1 個体のランダムサンプリング

任意の選択手法で選択された個体群を D とする．ランダムサンプリングは D からランダムに選ばれた個体のルートノード n_r から始まる．n_r からランダムに 1 つずつ子ノードを巡回・追加していく．ここで巡回済みのノードの集まりを断片 F，F の子ノードでまだ巡回されていないノードを候補ノード C とする．次に巡回されるノードは C からランダムに選ばれる．図 1 で，枠に囲まれたノードが F であり，グレーのノードが C である．

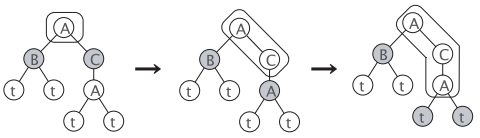

図 1: ノードの巡回

F は，C が無くなる (全て葉ノードに到達する) まで 1 ノードずつ成長するが，PORTS では各ノードのランダムサンプリングを確率 p_t で止め，別な木のノードへと遷移する．遷移が起こると全ての候補ノード C は無視され，下で説明する遷移方法に従い別のノードへ遷移する．この方法によって木構造断片 F は成長を止め，ランダムな形を持つことになる．また，遷移した先のノードからは同様に巡回が進められていく．以上の操作により，複数の木構造断片からなる新しい木構造がサンプリングされ，PORTS ではこれを子個体とする．

確率 p_t で起こる遷移には α 遷移と β 遷移があり，それぞれ確率 p_α，$(1-p_\alpha)$ で選ばれる．ここで，本稿で使用するパラメータを表 1 に示す．α 遷移は現在のノードに依存せず，現在のノードが n_j のとき，以下の確率で遷移先のノード n_i を選択する．

$$p_\alpha(n_i|n_j) = p_\alpha(n_i) = \frac{1}{N_p} \cdot \frac{1}{\#(tree_i)},$$

ここで，$tree_i$ はノード n_i を含む GP の木構造であり，$\#(tree_i)$

Contact: tanji@iba.t.u-tokyo.ac.jp, Makoto Tanji

表 1: 本稿で使うパラメータ

Common parameters	
N_p	population size
D	selected individuals
$depth_{max}$	depth limitation
Parameters of PORTS	
p_t	transition probability
p_α	ratio of the α transition
$p_\beta = (1 - p_\alpha)$	ratio of the β transition

は木 $tree_i$ のノード数を表す．つまり，ランダムな優良個体を一つ選び，そこから一つノードを選択するのが α 遷移である．一方，β 遷移は現在のノード n_j に依存する．β 遷移ではまず，n_j と同じラベルを持つノードを探し，その中からランダムにノード n_p を選択する．遷移先と同じインデックスの位置の n_p の子ノードを遷移先とする．これは，Context Aware Crossover[Majeed 06] と同じように，解が大規模に破壊されることを防ぐことを目的としている．ノード n_j から n_i への β 遷移の確率は以下で与えられる．

$$p_\beta(n_i|n_j) =$$

$$\begin{cases} \frac{1}{N_{label(n_j)}} \cdot \frac{1}{\#\{n_k \in tree_i | label(n_k) = label(n_j)\}}, \\ \text{if } \ label(parent(n_i)) = label(n_j) \ \& \ index(n_i) = index(next) \\ 0, \text{otherwise} \end{cases}$$

ここで，*next* は巡回しようとしていた子ノードを表す．図 2 にランダムサンプリングの様子を示す．この例では，ノード'B' から別の木のノード'D' へと β 遷移が行われている．

PORTS では，EDA(Estimation of Distribution Algorithm) や PMBGP(Program Model Building Genetic Programming) のように明示的なモデルを作らないため，メモリや計算量の観点から利点がある．また，以下のようにサンプリングの際に特別な条件の場合が存在する．

- I を新しくランダムサンプリングされた木構造とする．もし一度も遷移が行われずに I が親個体と同じ場合は，I を棄却し新しくサンプリングした個体で置き換える．
- GP の木構造に深さ制限がある場合，サンプリングの途中で最大深さに到達した場合は，強制的に終端記号に遷移する．

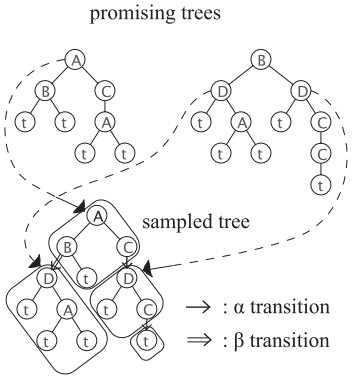

図 2: 優良個体からのランダムサンプリング

2.2 パラメータスケジューリング

ランダムサンプリングにおいて確率 p_t で木構造断片 F の成長が止まるため，サイズがちょうど n の木構造断片が次世代へ保存される確率，及びその期待値は以下の幾何分布で表される．

$$P_g(X = n; p_t) = p_t(1 - p_t)^{(n-1)} \ , \qquad E(X) = \frac{1}{p_t}$$

解の構造が可変長である GP では，通常，小さな部分解を交叉・突然変異によって組み立て，後に部分解を組み合わせてより大きな解を作り出すと考えれられている [Poli 08]．通常の GP では，木構造の平均サイズが増大することで，次世代に保存される木構造断片のサイズも増大する．PORTS では上記のようにパラメータ p_t でサイズが決定されるため，p_t が変化しない場合，進化の序盤では効率が良くても，後半で破壊的な操作になってしまう可能性が大きい．そこで，本稿では"適応的スケジューリング"と名づける調整を行う．$treeSize(P)$ を木構造サイズの集団平均とし，$f\_size(X)$ を，X を構成する木構造が前世代から受け継いでいる木構造断片の平均サイズとする． 世代 g でのパラメータ p_t^g は次世代 $g+1$ で次のように更新される．なお，初期世代は前世代が無いため更新しない．このスケジュールでは，p_t の最小値 p_l を下回らないことを保証しながら，有望な個体の木構造断片のサイズに適応するように更新される．

$$p_t^{g+1} \longleftarrow (p_t^g - p_l) \cdot \frac{p_{selected} - p_l}{p_{population} - p_l} + p_l \qquad (1)$$

$$p_{selected} = \frac{1}{f\_size(D)}, \quad p_{population} = \frac{1}{f\_size(P)}$$

$$p_l = \frac{1}{treeSize(P)}$$

3. 比較実験

SGP(Simple Genetic Programming) と PORTS の性能を比較するため，以下の 2 つの問題で実験を行った．実験条件として，SGP では，交差点を任意のノードから選ぶシンプルな交叉，部分木をランダムに生成し置き換える突然変異を使用する．また，初期世代は Ramped Half-and-Half で生成する．

3.1 Royal Tree 問題

Royal Tree 問題 [Punch 96] は Punch らによって提案された，GA の Royal Road 問題 [Mitchell 92] の拡張であり，GP の性能を測定するためのベンチマーク問題である．終端記号 $\mathcal{T} = \{x\}$，非終端記号 $\mathcal{F} = \{A, B, C \cdots\}$ を持つ．非終端記号 $A, B, C \cdots$ の数=L によって，問題の難易度が決定され，最適解は図 3 で示されるように (L-1) での最適解を組み合わせた構造である．優良部分解の組み合わせでより良い解を得ることができる特徴から，GP で解きやすい問題と言われる．PORTS でも木構造断片を組み合わせるため，解きやすいと予想される．

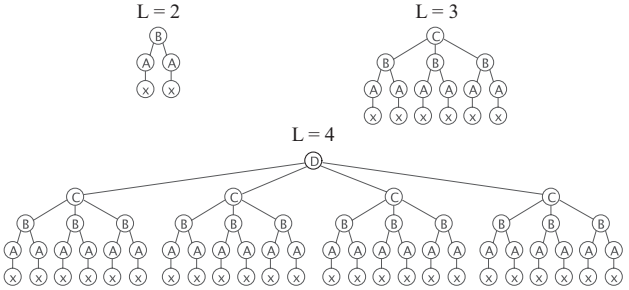

図 3: Royal Tree 問題のそれぞれのレベルに対する最適解

表 2: RoyalTree 問題で使用されるパラメータ

Population size	500 for L=4, 1000 for L=5
Generation	100
Selection	Tournament Selection
Tournament size	6
Maximum Depth	L+1
Transition probability p_t	0.4 for L=4, 0.1 for L=5
α transition ratio p_α	0.0 and 0.1
Crossover Ratio (SGP)	0.9
Mutation Ratio (SGP)	0.1
Success predicate	The perfect tree for each problem size

実験では L=4, 及び 5 の問題に対して 20 回実験を行なった．L=4,5 での最適値はそれぞれ 6144，122880 であり，ノード数は 65，326 である．実験に使用したパラメータを表 2 に示す．

3.1.1 Royal Tree 問題の実験結果

最適解の発見を”成功”とし，独立した 20 回の累積成功確率を図 4 に示す．L=4,5 に対して，PORTS では SGP の半分以下の評価回数で最適解が得られていることがわかる．また，SGP では局所解に陥り，抜け出せない場合があったが，PORTS では．100%解くことができている．しかし図 4 で見ることができるように，$p_\alpha = 0.0$ と $p_\alpha = 0.1$ で性能が違い，PORTS では p_α の値が性能に大きく影響することもわかった．この実験では p_α は経験的に決められている．

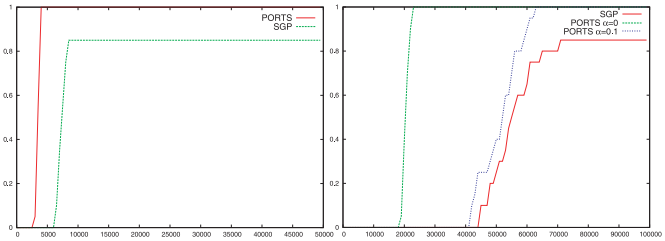

図 4: L=4(左)，5(右) の Royal Tree 問題に対する累積成功確率

また，図 5 は，世代交代の際にどのようなサイズの木構造断片が保存されているかを描画したものであり，図 6 は，SGP において，交叉・突然変異によって，次世代へ保存された部分構造のサイズの分布である．図より，PORTS ではなめらかに木構造断片が保存されていることがわかる．また，図 5 中の直線はそれぞれの世代において木構造が無限だとした場合の p_t^g による幾何分布である (縦軸が対数であることに注意)．一方，SGP では局所的に集中している場合が多いため世代を分けて描画している．図 6 から，SGP ではまったく保存されない部分構造が多く存在していることがわかる．冒頭であげた，解の多様性のために様々なサイズの部分解を保存するという観点からは PORTS では取りこぼしが少ないように見える．

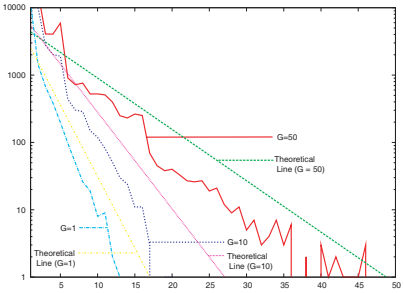

図 5: 世代交代で保存された部分構造のサイズ分布 (PORTS)

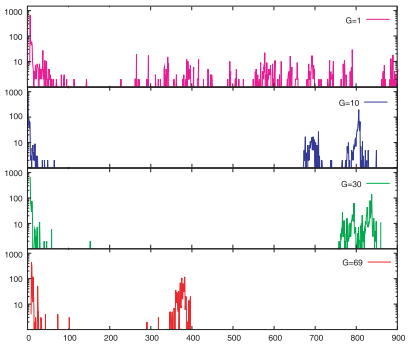

図 6: 世代交代で保存された部分構造のサイズ分布 (SGP)

表 3: Symbolic Regression 問題で使用されるパラメータ

Population size	500
Generation	100
Selection	Tournament Selection
Tournament size	4
Maximum Depth	15
Transition probability p_t	0.5
α transition ratio p_α	0.9 and 1.0
Crossover Ratio (SGP)	0.9
Mutation Ratio (SGP)	0.1
Target Function: F1	$x^6 + x^5 + x^4 + x^3 + x^2 + x$, $(-1 \leq x \leq 1)$
The number sample	20
成功条件	F(t)=0 の場合

3.2 Symbolic Regression 問題

Royal Tree 問題は，部分解の組み合わせによる最適解が一つのみであり，実問題とはかけ離れている．より現実問題に近い問題として，GP の性能を計るために使われる Symbolic Regression 問題 [Koza 92] で実験を行った．この問題の目標は，(x, y) の点の集合のデータが与えられたときに最もデータを良く表す数学的な式を探す回帰問題として表現できる．関数ノード $\mathcal{F} = \{+, -, *, /, SIN, COS, EXP, RLOG\}$，終端記号 $\mathcal{T} = \{x\}$ を持ち，評価関数は以下で与えられるように GP の式 $t(x)$ とデータ y 誤差の和として計算される．

$$F(t) = \sum_{i=1}^{N} abs(t(x_i) - y_i).$$

実験では以下の式から生成したデータを使用した．また，実験条件は表 3 にある通りである．

$$F_1 = x^6 + x^5 + x^4 + x^3 + x^2 + x, \ (-1 \leq x \leq 1)$$

3.2.1 Symbolic Regression 問題の実験結果

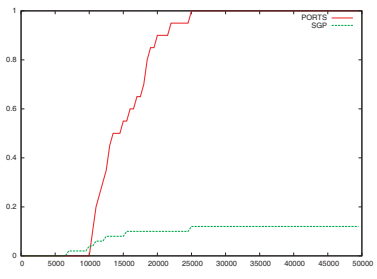

図 7: Symbolic Regression 問題に対する累積成功確率

図 7 は，SGP と PORTS の累積成功確率を表す．図より，PORTS では 25000 回以内の評価で 100%対象関数を見つけていることがわかる．SGP では始めて成功するまでの評価回数

が PORTS より少ない場合も数回あったが，多くの試行が局所解に陥っている．

Royal Tree 問題と同様に，PORTS と SGP の木構造断片及び部分構造が保存される分布を図 8 と 9 に示す．Royal Tree 問題と同様の傾向が見られた．PORTS では幾何分布に近くなめらかに木構造断片が保存されており，SGP ではサイズの大きな部分構造が保存されており，その他のサイズは少ない領域が多かった．

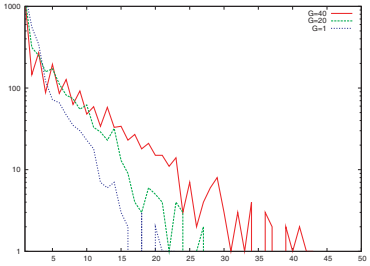

図 8: 世代交代で保存された部分構造のサイズ分布 (PORTS)

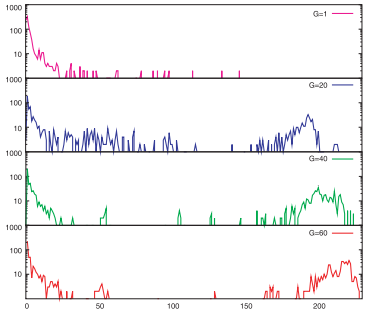

図 9: 世代交代で保存された部分構造のサイズ分布 (SGP)

4. 考察

4.1 部分構造の分布

もし，集団の木構造サイズが無限の場合は (サンプリングが終了しないが)，幾何分布に従ったサイズの木構造断片 F が保存される．しかし，実際には個体数，木構造のサイズ共に有限であるため，Royal Tree 問題での図 5 のように，理論的な分布からは外れる．しかし，おおむね実験でも予測可能な分布に従った形で木構造断片が保存されることが示され，その結果，二つの問題において SGP より少ない評価回数で問題を解くことが示された．

幾何分布の妥当性については，今後研究の余地が残されているが，直感的には以下のように考えることができるであろう．GP の集団中で可能な全ての木構造断片を数える場合，あるサイズの断片は確実により小さな断片を含んでいる．これは木の形状に依存するが，含まれる断片の数はサイズが小さいほど組み合わせ論的に数が増えるであろう．そのため，幾何分布のように指数関数的に減少する幾何分布は，現時点ではある程度の正当性をもつと考えられる．

4.2 遷移確率のスケジューリング

本研究では，遷移確率 p_t に対して，次第に大きな部分解を構築するために適応的スケジューリングと呼ぶ方法で調節している．実験では木構造の平均サイズは世代と共に大きくなり，p_t が下がることで，分布の形状が横に伸び，小さな断片の組み合わせよりも，大きな木構造断片を継承した個体が現れ，解空間を効率よく探索が進められた．適応的スケジューリングが効率的に働くのは仮説の段階であるので，今後理論的に調べていく必要がある．

4.3 パラメータ p_α

実験で使われた PORTS のパラメータ p_α は，経験的に決定されたものであった．Royal Tree 問題において，性能が p_α に大きく依存していたことを考えると p_α に対する何らかの設計方針が必要である．β 遷移は，親ノードのラベルに依存して遷移先が決定するため，親子関係を破壊しない遷移といえる．そのため Royal Tree 問題のような親子のノードに強い依存関係が存在する問題には p_α を小さくすることで効率的に問題を解くことができた．一般には，確実に強い親子関係がある場合を除いて，現時点では p_α は比較的大きな値に設定するのが望ましい．将来的には，遷移確率 p_t のように，問題に対して自動的に適応するような機構が必要である．

5. 結論

本研究では，様々なサイズの部分構造が進化の材料として必要であるという観点から，GP の新しい探索手法 PORTS を提案した．PORTS では様々なサイズの親の木構造断片を幾何分布に従ってサンプリング・結合することで新しい個体を作る．また，適応的スケジューリングによって，この分布の形状を変えながら探索を進める．実験より，従来の SGP と比べて Royal Tree 問題，Symbolic Regression 問題において少ない回数で確実に最適解を得ることができた．PORTS のメリットとして，世代交代の際に保存される木構造断片の分布を予測・コントロールできる点，があげられる．

今後の課題として，遷移確率 p_t のスケジューリングに関する理論的な保証や，α 遷移，β 遷移のパラメータの自動決定などに関する詳しい研究が必要である

参考文献

[Koza 92] Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992)

[Majeed 06] Majeed, H. and Ryan, C.: Using context-aware crossover to improve the performance of GP, in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation* (2006)

[Mitchell 92] Mitchell, M., Forrest, S., and Holland, J.: The royal road for genetic algorithms: Fitness landscapes and GA performance, in *Proceedings of the First European Conference on Artificial Life* (1992)

[Poli 08] Poli, R., McPhee, N. F., and McPhee, N. F.: The Impact of Population Size on Code Growth in GP: Analysis and Empirical Validation, in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation* (2008)

[Punch 96] Punch, B., Zongker, D., and Goodman, E.: *Advances in genetic programming: volume 2*, chapter The royal tree problem, a benchmark for single and multiple population genetic programming, pp. 299–316, MIT Press (1996)

[Tanji 09] Tanji, M. and Iba, H.: Program Optimization by Random Tree Sampling (In Press), in *GECCO '09: Proceedings of the 10th annual conference on Genetic and evolutionary computation, 2009* (2009)