二分岐化プロトタイプ木を用いた確率モデルGP

Probabilistic Model Building GP with Binary Encoded PPT

柳瀬 利彦*1　伊庭 斉志*2
Toshihiko Yanase　Hitoshi Iba

*1東京大学新領域創成科学研究科
Graduate School of Frontier Science, The University of Tokyo

*2東京大学工学系研究科
Graduate School of Engineering, The University of Tokyo

Recently, program evolution algorithms based on the probabilistic prototype tree (PPT) based method have been proposed to improve the search ability of genetic programming (GP) and to overcome GP-hard problems. The PPT based method explores the optimal tree structure by using the full tree whose number of child nodes is maximum among possible trees. This algorithm, however, suffers from problems arising from function nodes having different number of child nodes. These function nodes cause intron nodes, which increase the search space. In order to solve this problem, we propose binary encoding for PPT. Here, we convert each function node to a subtree of binary nodes where the converted tree is correct in grammar. Our method reduces ineffectual search space, and the binary encoded tree is able to express the same tree structures as the original method. The effectiveness of the proposed method is demonstrated through the use of two experiments.

1. はじめに

本研究では，確率モデルを用いたプログラム進化アルゴリズム (Probabilistic Model Building Genetic Programming, PMBGP) で用いられる木構造の二分岐化法の提案を行う．この変換により，適合度に影響を与えない intron の数を減らし，探索効率を向上させることができうる．

GP はプログラムや関数を進化的に獲得する手法である．近年，PMBGP の研究が進展し，様々な問題に対して従来の GP より優れた成績を示している．PMBGP にはプロトタイプ木 (Probabilistic Prototype Tree, PPT) を用いる手法 [Sałustowicz 97, Hasegawa 08] と確立文脈自由文法を用いる手法 [Shan 04] の二つがある．本研究では，PPT を用いる手法を取り上げる．PPT は Sałustowicz らによって提案された確率モデルを用いてプログラム推定を行うためのデータ構造である [Sałustowicz 97]．PPT に基づく確率モデル GP では，最も引数の多い関数ノードにあわせた完全木を作成し，探索を行う．異なる引数の数を持つ関数ノードが混在する問題では，intron の数が多くなり探索効率が悪化する．この問題に対し，PPT を二分木に変換する手法を提案する．本手法により，木構造の表現能力を損なうことなく探索空間を縮小できる．ベンチマーク問題を用いて既存の手法との比較を行い，探索性能が改善することを確認した．

2. Probabilistic Prototype Tree

本研究では，PPT に基づくプログラム探索手法を採用する．PPT を用いる手法では，木構造を配列に変換し，確率モデルの推定を行う．確率モデルの推定には，分布推定アルゴリズム (Estimation of Distribution Algorithm) の手法が用いられ，独立モデル [Sałustowicz 97] や親子関係モデル [Yanai 03]，ベイジアンネットワークを用いるモデル [Hasegawa 08] などが提案されている．本研究では，Hasegawa らが提案したベイジアンネットワークを用いる手法である Program Optimization with Linkage Estimation (POLE)[Hasegawa 08] を採用する．POLE の依存関係にベイジアンネットワークを用いるため，図 1 の (c) のようにノード間依存関係を柔軟に表現できる．POLE では図 1 の (a)(b) のようにネットワーク構造を定めることで独立モデルや親子関係モデルも表現可能である．

連絡先: 柳瀬利彦，東京都文京区 7-3-1 工学部電気系 伊庭研究室，(03)5841-6751，(03)5841-6751，yanase@iba.t.u-tokyo.ac.jp

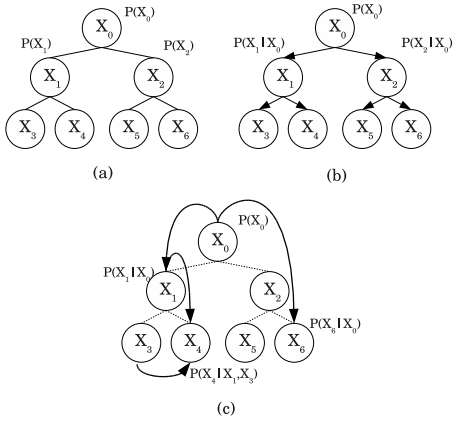

Figure 1: Network structures of (a) Univariate model, (b) Adjacent model and (c) Bayesian network model

2.1 従来の PPT の問題点

PPT は分布推定アルゴリズムの手法を用いて木構造を推定するためのデータ構造である．従来の PPT は関数ノードの最大の引数の数に準ずる分岐数を持つ完全木である．引数の数を b ，深さを D_p とする木のノード数は以下の式で表される．

$$\text{Tree Size} = \frac{1 - b^{D_P}}{1 - b}, \tag{1}$$

同じ深さの木でも引数の増加に伴い，PPT のサイズは急激に増加する．たとえば，$b = 2, D_p = 10$ の場合，木のノード数は 1023 であるが，$b = 4, D_p = 10$ の場合はノード数は 349525 となる．この 4 分木のようにノード数が非常に多い場合，従来の PPT では現実的な計算時間で探索を行うことは難しい．

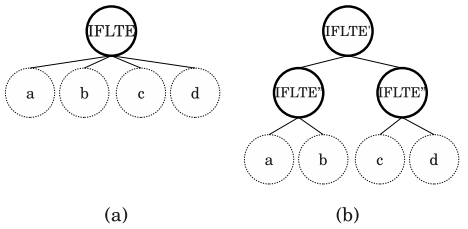

Figure 2: Convert 4-arity node into three binary nodes. (a) original IFLTE node. (b) the subtree for binary encoding of IFLTE which is composed of one IFLTE′ and two IFLTE″.

一つの木に異なる引数の数を持つ関数ノードが存在した場合を考える．この場合，2 引数の関数の 3 番目以降の子ノードおよびサブツリーは評価されない．この評価されないノードを intron と呼ぶ．本研究はこの intron を削減することで，探索空間を縮小することを目指す．

3. 二分岐化 PPT

3.1 関数ノードの変換

本研究では関数ノードを 2 引数のサブツリーで置き換えることで PPT を二分岐化する方法を提案する．引数の数 b の関数ノードは深さ $[\log_2(b-1)]+1$ の完全二分木で置き換えられる．元の関数の引数はサブツリーの葉ノードに左から順に追加される．この二分岐化した PPT と BPPT(Binary encoded PPT) と呼ぶ．図 2 に 4 引数関数である **IFLTE**(a,b,c,d) の二分岐化の例を示す．**IFLTE** は一つの **IFLTE**′ と二つの **IFLTE**″ で表される．関数ノードリスト $\{F\}$ 中の **IFLTE** は **IFLTE**′ に置き換えられる．文法的に正しい木を作り出すため，**IFLTE**″ は $\{F\}$ に追加されず，必ず **IFLTE**′ の生成に伴って木に現れるよう制約を定める．本研究では，PPT を用いた手法のひとつである POLE に BPPT の適用を行った．以下，POLE の場合の制約について説明する．

3.2 二分岐化の制約

POLE はベイジアンネットワークによってノード間の依存関係と，ノードのシンボル S(関数ノードと終端ノード) を推定する．そのため，シンボル決定が根ノードから順に行われるわけではない．POLE-BPPT でのシンボル決定のアルゴリズムを Algorithm1 に示す．本手法では 3 つの制約を設けることで文法的に正しい木を生成する．1. 二分岐化関数ノードはその根ノード以外のシンボルは独立には生成されない．2. 二分岐化関数ノードが生成される場合にはその子ノードにサブツリーを展開する．3. 根ノードに向かって木をさかのぼり，二分岐化関数ノードの深さに応じて禁止シンボルを設定する．禁止シンボル S' とは X_i に存在することができないシンボルである．ノード図 3 に 4 引数関数ノードである **IFLTE** の例を示す．ノード X_0 に二分岐化関数ノードである **IFLTE**′ が生成される場合，X_3, X_4 には **IFLTE**″ が展開される．そして，X_0 には **IFLTE** を展開できないため，禁止シンボルのリスト $\{S'\}$ に **IFLTE** を追加する．

木の評価を行う際には，二分岐化関数ノードを元の関数ノードで置き換える操作が必要になる．この場合，木は完全二分木ではなくなるが，評価以外には使われないため，推定には影響を及ぼさない．PPT の二分岐化の詳細については文献 [Yanase 09] を参照のこと．

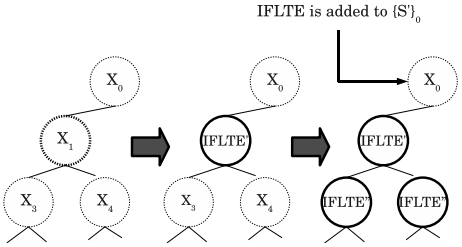

Figure 3: Example of symbol determination from a Bayesian network at node X_1

Algorithm 1 Symbol determination from a Bayesian network at node X_i

1. Refer to the list of forbidden symbols $\{S'\}_i$ at X_i
2. Set the probability of $\{S'\}_i$ zero at X_i in CPT
3. Generate Symbol from CPT
4. Traverse the ancestor nodes X_a of X_i until the root node in order to update $\{S'\}_a$
   $D_a \leftarrow$distance between X_a and X_i
   for each$S_j \in \{S\}$
   (a) $D_{subtree} \leftarrow$subtree depth for binary encoding of S_j
   (b) if $D_a < D_{subtree}$ then add S_j to $\{S'\}_a$

4. 比較実験

探索能力の比較を行うため，提案手法の BPPT を二つのベンチマーク問題に適用した．intron を含む問題として Royal Tree 問題を，intron を含まない問題として DMAX 問題を採用した．従来の PPT と GP でも実験を行い比較を行った．各問題とも最適解の深さを大きくすると問題が難しくなる．GP で表現した最適木の深さを問題のレベル Lv とした．全ての手法について，個体数 M を $M = 100$ から始め，最適解が見つからない場合 $\sqrt[5]{10}$ 倍していった．つまり，$M = 100, 160, 250, 400, 630, 1000, 1600, \ldots$ と増加させ 20 試行の平均適合度評価回数を求めた．平均適合度評価回数は大きな分散を持つため，実験は 10 回繰り返した．

4.1 Royal Tree 問題

Royal Tree 問題で用いた関数ノードは $F = \{A, B, C, \ldots\}$ 終端ノードは $T = \{x, y\}$ である．ここで，$x = 1.0, y = 0.95$ と定める．関数ノードの引数はアルファベット順に $1, 2, 3, \ldots$ と増加する．Royal Tree 問題はノードの親子間に強い依存関係を持つ問題である．本実験では終端ノードに y を導入することで葉ノードの兄弟間にも依存関係を持たせた．

表 1 にレベル 4,5 の Royal Tree 問題を解くのに要した平均適合度評価回数とその標準偏差を示す．レベル 5 の問題について提案手法の POLE-BPPT が最適解を探索するのに要した平均適合度評価回数は 19747.0 であり，従来手法の POLE では 32022.5 であった．POLE-BPPT は POLE の 0.62 倍の適合度評価回数で最適解を発見している．図 4 に POLE-BPPT と POLE による Royal Tree 問題の成功率を示す．横軸は適合度評価回数，縦軸は 20 試行での成功率を示す．図には中央

Table 1: Result: Royal Tree Problem

		$Lv = 4$	$Lv = 5$
POLE (BPPT)	Average	792.9	19747.0
	Std.	(99.5)	(2434.0)
POLE	Average	749.5	32022.5
	Std.	(100.6)	(4906.6)
Univariate (BPPT)	Average	1744.4	196220.0
	Std.	(298.9)	(46067.8)
Univariate	Average	2288.6	329245.0
	Std.	(564.6)	(72216.3)
SGP	Average	2953.8	382957.6
	Std.	(1108.1)	(179342.6)

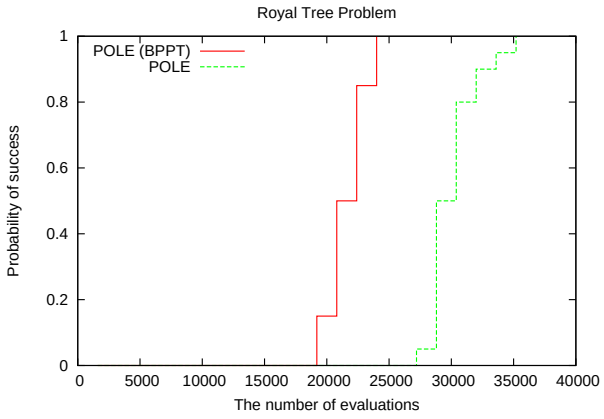

Figure 4: The probability of success using POLE with BPPT and conventional POLE for Royal Tree problem.

値に当たる例を示した．POLE-BPPT が POLE よりも早く収束していることが観察される．

図 6 に探索中に構築されたベイジアンネットワークの例を示す．図 6 の (a) は POLE-BPPT($M = 160, Lv = 4, D_p = 5$) のネットワークであり，図 6 の (b) は POLE($M = 160, Lv = 4, D_p = 4$) のネットワークである．これらは，最終世代からランダムに抽出した．

依存関係推定を行わない場合，つまりノードごとに独立した確率分布を用いた場合 (Univariate Model, UM) でも実験をおこなった．レベル 5 の問題の UM-BPPT の平均適合度評価回数は 196220.0, 従来手法の UM では 329245.0 であった．UM においても BPPT を導入することで性能が改善される．また，依存関係の推定を行った場合と行わなかった場合では，依存関係推定を行った方が著しく性能が向上することが見られた．

4.2 DMAX 問題

DMAX 問題は，MAX 問題を拡張した GP に対する騙し問題である [Hasegawa 08]．DMAX 問題で用いられる関数ノードは $F = \{\times_5, +_5\}$ であり，終端ノードは $T = \{\lambda, 0.95\}$ である．ここで $\times_5, +_5$ はそれぞれ 5 引数の乗算，加算を表し λ は，$\lambda^5 = 1$ となる複素数である．評価値は式の実部で与えられる．用いられる関数ノードが全て 5 引数であるため，PPT には intron が存在しない．この問題は BPPT の intron の影響を調べるためのベンチマーク問題である．

表 2 に平均適合度評価回数と標準偏差を示す．レベル 4 の問題に対する成功率の推移を図 7 に示す．レベル 3,4 の DMAX 問題に対し，POLE-BPPT の平均適合度評価回数は，従来の

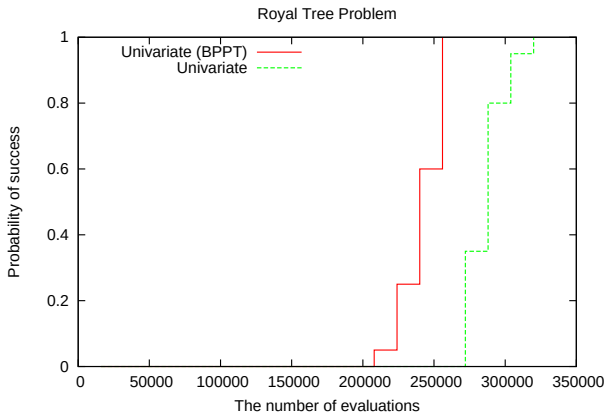

Figure 5: The probability of success using Univariate model with BPPT and conventional Univariate model for Royal Tree problem.

Table 2: Result: DMAX Problem

		$Lv = 3$	$Lv = 4$
POLE (BPPT)	Average	1464.4	103978.5
	Std.	(104.1)	(19192.0)
POLE	Average	1498.9	117616.0
	Std.	(83.0)	(11562.0)
SGP	Average	11105	1632159
	Std.	(6958)	(645696)

POLE と比べて明らかな違いは見られなかった．

5. 考察

Royal Tree 問題の関数ノードを E, F と増やした場合について，プロトタイプ木中の intron の割合がどのように変化するかを図 8 に示す．E ノードを用いる場合，PPT の intron の割合は 0.9 を超す．それに対して，BPPT では F を用いる場合でも 0.7 程度にとどまっており，BPPT が intron を抑制していることがわかる．また，木のサイズについても E ノードを用いる場合，PPT は 3906 と BPPT の場合の 1023 と比べて 3.8 倍になっており，探索空間が著しく増加していることが見られる．BPPT は intron の割合を減少させることで，無駄

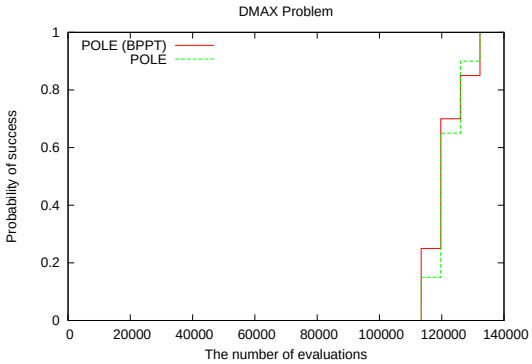

Figure 7: The probability of success using POLE with BPPT and conventional POLE for DMAX problem.

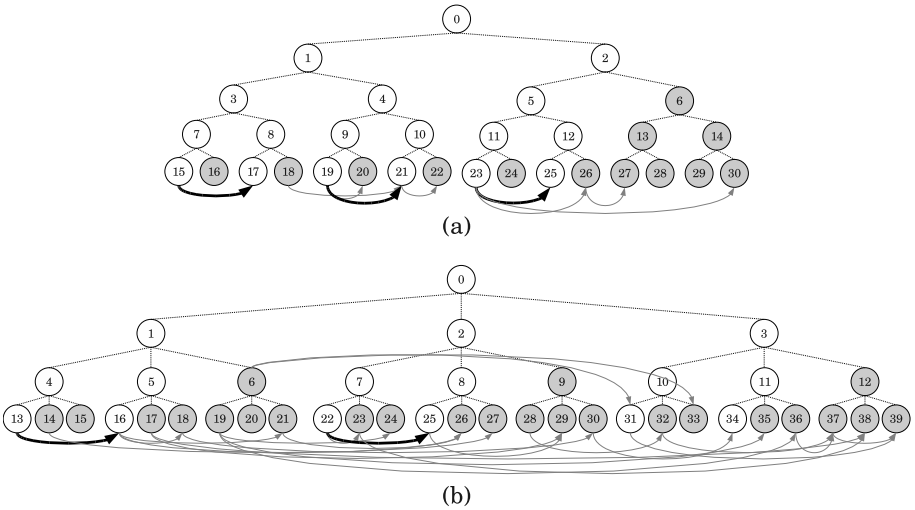

Figure 6: An example of a network for the royal tree problem using (a) POLE with BPPT ($M = 160, Lv = 4, D_P = 5$) (b)POLE ($M = 160, Lv = 4, D_P = 4$). The grey nodes represent introns. Bold arrows means interaction between nodes estimated correctly.

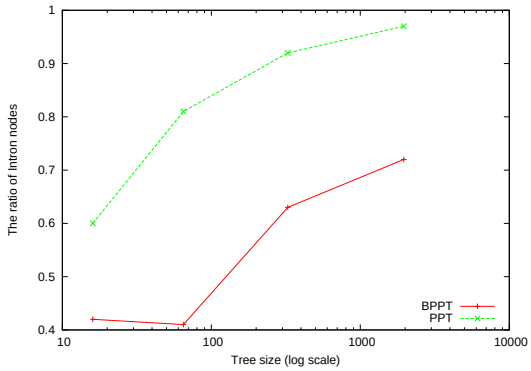

Figure 8: The ratio of intron nodes in Royal Tree Problem.

な探索空間を削減することができ，これが探索性能の向上に寄与していると考えられる．

また，図 6 に示すように BPPT は葉ノードの intron を特に減少させている．レベル 4 の問題に対して，BPPT の葉ノードの intron 率は 0.63 であるのに対し，PPT では 0.78 である．このように，BPPT は葉ノードの intron を削減し兄弟ノード間の依存関係推定に必要な探索量を減らしていると考えられる．

DMAX 問題では PPT は intron を含まないが，BPPT は intron を含む．それにもかかわらず，平均適合度評価回数には大きな差は見られなかった．これは，BPPT は関数ノードがサブツリーで表現され，深さを持っていることによると考えられる．BPPT の方が PPT と比べて関数ノードが存在できる位置が限られる．このため，BPPT の方が PPT よりも intron の影響が小さくなると考えられる．

この実験では最大でも 1000 程度のノード数だったため，確率モデルを用いた木構造の推定に現実的な計算時間で成功した．しかし，より応用的な問題に適用するにあたって，さらに大きな木構造の推定を行う必要が現れると考えられる．その場合，ひとつの解決策として計算時間を削減するために並列化や分散化などを行う必要があると考えられる [Očenášek 01]．また，問題の特性に合わせて依存関係の範囲を限定することも有効であると考えられる．

6. おわりに

本研究では，PPT を二分岐化した木構造 BPPT を提案した．異なる数の引数を持つ関数ノードが混在する問題では，従来の PPT では intron が増加し性能が悪化する．そのような問題に対して BPPT を用いることで intron を削減することができる．ベンチマーク問題を用いて POLE と POLE-BPPT の比較実験を行い，BPPT によって intron の問題を改善できることを示した．今後は，ロボットプログラム学習や，時系列予測，システム同定などのより応用的な問題に適用していきたい．

参考文献

[Hasegawa 08] Y. Hasegawa and H. Iba. A Bayesian network approach to program generation. *IEEE Transactions on Evolutionary Computation*, Vol. 12, NO. 6:750–764, 2008.

[Očenášek 01] J. Očenášek and J.Schwarz. The distributed Bayesian optimization algorithm. *in EUROGEN 2001*, 115-120, 2001.

[Sałustowicz 97] R. Sałustowicz and J. Schmidhuber. Probabilistic incremental program evolution. *Evolutionary Computation*, 5:123–141, 1997.

[Shan 04] Y. Shan, R. I. Mackay, and R. Baxter. Grammar model-based program evolution. *in Proc. of the 2004 Congress on Evolutionary Computation*, 478–485, 2004.

[Yanai 03] K. Yanai and H. Iba. Estimation of distribution programming based on Bayesian network. *in Proc. of Congress on Evolutionary Computation*, 1639-1646, 2003.

[Yanase 09] T. Yanase, Y. Hasegawa, and H. Iba. Binary encoding for prototype tree of probabilistic model building gp. *in Proc. of Genetic and Evolutionary Computation Conference (in Press)*, 2009.