1E2-1

頻出部分木発見に基づく遺伝的プログラミング手法のベンチマーク評価

Benchmark Evaluation of Genetic Programming with Tree Mining

倉田 芳明[*1] 亀谷 由隆[*2] 佐藤 泰介[*2]
Kurata Yoshiaki Kameya Yoshitaka Sato Taisuke

[*1]東京工業大学 大学院理工学研究科 集積システム専攻
Dept. of Communications and Integrated Systems, Graduate School of Science and Engineering, Tokyo Institute of Technology

[*2]東京工業大学 大学院情報理工学研究科 計算工学専攻
Dept. of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

In genetic programming (GP), one of the most crucial issues is how to acquire good building blocks efficiently. Recently, Kumagai et al. proposed a GP method (called GPTM, GP with Tree Mining, in this paper) which encapsulates the subtrees repeatedly appearing in the individuals with higher fitness. To find such subtrees, GPTM utilizes FREQT, a data mining method that efficiently enumerates frequent subtrees. Since GPTM has been applied to only one particular problem so far, in this paper, we further apply GPTM to a couple of well-known benchmark problems. The results show that GPTM is superior to GP and EDP (Estimation-of-Distribution Programming), and is competitive with POLE (Program Optimization with Linkage Estimation), one of the state-of-the-art probabilistic model building GP methods.

1. はじめに

進化論的手法として知られる最適化手法は，生物進化のメカニズムを模倣しており，ビット列や木構造などの染色体表現を変形，合成，選択することにより効率な解探索を行う．中でも，木構造のプログラムを染色体表現とする進化論的手法として遺伝的プログラミング (Genetic Programing, GP) が知られている [Koza 92, Iba 93]．一方，大量のデータから必要な知識を取り出す技術としてデータマイニングが知られており，近年この分野では，半構造データに頻出する木構造パターンを効率良く列挙する頻出部分木発見手法が提案されている．

熊谷ら [熊谷 07] は，頻出部分木発見手法の一つである FREQT [浅井 02] を GP に適用する手法 (以下では GP with tree mining, GPTM と呼ぶ) を提案し，最適化問題の一つである交通信号制御問題に適用した．その結果，GPTM が獲得した信号制御プログラムは GP などの既存手法で得られるものに比べ円滑に交通を制御することを示した．しかし，熊谷らの実験では交通信号通制御問題に対する有用性のみしか示されていなかったため，本論文では GPTM をよく知られる既存のベンチマークに適用し，従来の GP および確率モデルに基づく GP と比較することにより，その有用性を確認する．

2. GPTM

本節では GPTM について簡単に記述する．GPTM の基本的な方針は，高い適合度をもつ個体群に繰り返し出現するパターン（部分木）を優良な部分解と見なし，それらのパターンが遺伝的操作（交叉）により破壊される確率を小さくすることである．まず，GPTM の処理の流れを示す (図 1)．GPTM の処理は基本的に GP と同じであるが，適合度の高い個体から頻出部分木を発見する手続き（具体的には FREQT）が加えられている．FREQT のように任意の頻出部分木を抽出する手法を用いることにより，完全な部分木 (図 2 上) のみならず，葉から遠い不完全な部分木 (図 2 下) も保護されることが期待できる．

FREQT アルゴリズムは，木のデータベース $\mathcal{D}$ に σ 頻出するパターン木 $\mathcal{T}$ すべてを発見するものである．ここで，σ 頻出とは，データベース $\mathcal{D}$ のノード数に対するパターン木 $\mathcal{T}$ の出現数の割合が σ 以上であることを意味する ($0 < \sigma \leq 1$)．FREQT アルゴリズムは最右拡張と呼ばれる効率の良い手法を用いて順序木の枚挙を行う．なお，本論文では [熊谷 07] に記述された方法に従い，σ の値を自動的に調節した．

頻出部分木を獲得した後は，この頻出部分木が破壊されにくくなるように交叉確率を修正する．例えば，図 3 の色付きの部分が頻出部分木であり，全体の木が交叉における父方（交叉点以下が子に受け継がれる方の親）であったとする．このとき，GPTM では図 3 の各内部ノードの交叉確率を割引き，その分の確率を根ノードの交叉確率に加算する．この交叉確率の修正によって，頻出部分木，すなわち優良と期待される部分木が子に受け継がれる確率が高まる．一方，現在の GPTM では突然変異確率は操作していない．

3. 確率モデルに基づく遺伝的プログラミング

近年，遺伝的プログラミング分野において確率モデルに基づく手法 (Probabilistic Model Building Genetic Programming, PMBGP) が注目されている．これらの手法は優良個体における記号の出現分布を確率モデルによって近似し，優良な部分解を残すことを目的としている．PMBGP の中でも，確率プロトタイプ木 (Probabilistic Prototype Tree, PPT) と呼ばれる完全木（図 4 の破線）を用意し，その上に Bayesian network（以下 BN, 図 4 の実線）に基づき確率分布を定義する手法として EDP (Estimation-of-Distribution Programming) [Yanai 03] や POLE (Program Optimization with Linkage Estimation) [Hasegawa 06, 長谷川 07] が知られる．EDP では図 4 (a), (b) のように PPT 上の親子や兄弟の間に依存関係を固定するが，POLE では依存関係のリンクを固

連絡先: 亀谷由隆, 東京工業大学 大学院情報理工学研究科 計算工学専攻, 東京都目黒区大岡山 2-12-1, Tel/Fax 03-5734-2186, kameya@mi.cs.titech.ac.jp

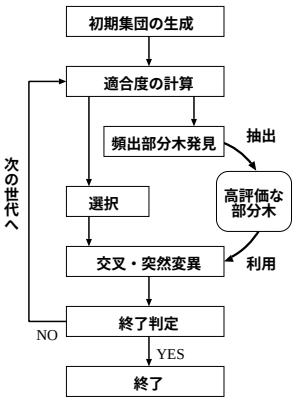

図 1: GPTM の流れ.

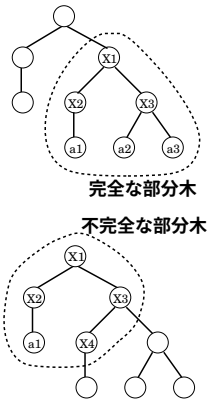

図 2: 完全な部分木と不完全な部分木.

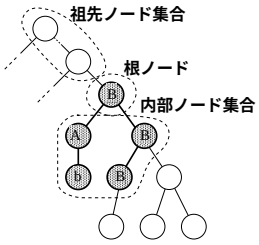

図 3: GPTM における頻出パターンの保護.

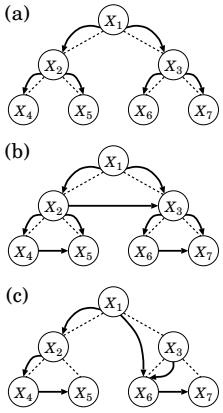

図 4: PPT 上の BN.

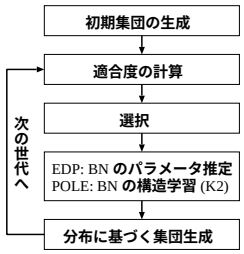

図 5: PMBGP の流れ.

定せず，図 4 (c) のような形の BN 構造を K2 アルゴリズム [Cooper 92] により学習する．また，POLE では BN におけるパラメータ数の増加を拡張構文木と呼ばれる表現方法を用いて解決している．

PMBGP における処理の流れを図 5 に示す．PMBGP では選出した優良個体の集団をデータ集合と見なし，EDP では BN のパラメータ推定（最尤推定），POLE では BN の構造学習が行われる．そして次世代の集団は学習された BN をサンプラーとして生成される．本論文で用いる EDP は，最尤推定後の BN の分布を以下の式 1 のように補正している．

$$\theta' = (1-\alpha)\hat{\theta} + \alpha\theta_{\text{default}} \tag{1}$$

ここで $\hat{\theta}$ は最尤推定されたパラメータベクトル，θ_{default} は対応するデフォルトの確率分布（本論文では一様とする）のパラメータベクトルであり，α をデフォルト比率と呼ぶ．また，本論文では，EDP の提案論文 [Yanai 03] で行われた実験設定に準じ，図 4 (a) に示す親子間の依存関係のみを考える．

一方，本論文では POLE で実行する K2 アルゴリズムにおいてパラメータを MAP (maximimum a posteriori) 推定によって求める．

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + c}{\sum_k N_{ijk} + r_i c} \tag{2}$$

ここで θ_{ijk} は i 番目の変数 X_i が親集合における j 番目の変数割り当てにおいて k 番目の値をとる確率（パラメータ）である ($1 \le k \le r_i$)．同様に N_{ijk} は，優良個体集団で i 番目の変数 X_i が親集合における j 番目の変数割り当てにおいて k 番目の値をとった回数である．c は擬似的なカウントとして扱われる．また，K2 アルゴリズムでは多くの計算時間を要するため，各変数における依存関係の親の探索範囲（例えば図 4 (c) で X_6 の依存関係の親は X_1, X_3）を d 世代前までとする．BN の構造は下の BIC (Bayesian information criterion) で評価する．N は優良個体数，K は BN のパラメータ数である．

$$\text{BIC} = \textstyle\sum_{ijk} N_{ijk} \log \hat{\theta}_{ijk} - \frac{1}{2} K \log N \tag{3}$$

4. ベンチマーク評価

GPTM の有用性を示すために，記号当てはめ (symbolic regression) および Royal Tree という二つの問題に適用した．実装は GPsys-2b (`ftp://cs.ucl.ac.uk/genetic/gp-code/`) を拡張したものを用いた．比較に用いた手法は GP, GPTM, POLE, EDP の四つである．本論文のベンチマークで使用した問題共通のパラメータ設定を表 1 に示す．

表 1: 問題共通のパラメータ設定.

	意味	値
全手法共通	初期集団生成法	Grow
	Grow での関数記号選択確率	0.9
	エリート個体数	1
	試行回数	50
GP, GPTM 共通	選択方式	トーナメント選択
	トーナメントサイズ	7
	突然変異確率	0.05
GPTM のみ	頻出パターンサイズ	3
	マイニング用優良個体数	50
EDP, POLE 共通	選択方式	Truncate 選択
	Truncate 選択確率	0.2
POLE のみ	モデルスコア	BIC

4.1 記号当てはめ問題

記号当てはめ問題は未知の関数 $f(x)$ に対して与えられた関数記号と定数記号を組み合わせて近似関数 $f'(x)$ を求める問題である．記号当てはめ問題では部分解をもつとは限らないが，各手法の一般的な性能を見るために適用した．関数記号および定数記号は以下のものを与える．

関数記号: $\{ADD, SUB, MUL, DIV, SIN, COS\}$
定数記号: $\{x, 0.05, 0.10, 0.15, \ldots, 1.00\}$

ADD, *SUB*, *MUL*, *DIV* はそれぞれ二引数関数で第一引数と第二引数を加算，減算，乗算，除算した値を返す．*SIN*, *COS* はそれぞれ一引数関数である．定数記号は関数の引数を表す x, および 0.05 から 1.00 まで 0.05 刻みで得られる 20 個の定数を使用する．適合度として以下を用いた（最適値は 1000）．

$$\text{適合度} = 1000 - 50 \sum_{j=1}^{30} |f(x_j) - f'(x_j)|, \quad x_j = 0.2(j-1)$$

今回のベンチマークにおいて使用した関数は [Yanai 04] で用いられている下の三つである．f_c は与えられた記号では実

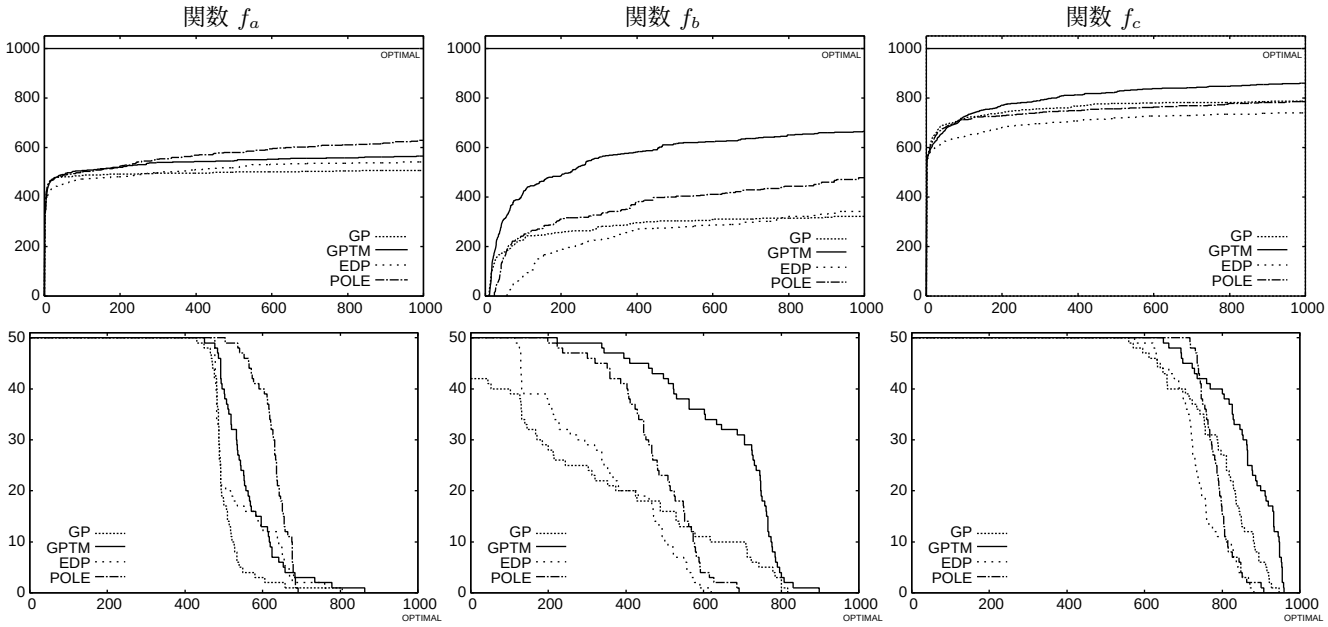

図 6: 記号当てはめ問題における比較．上段：最良個体の適合度の推移，下段：第 1000 世代における試行頻度分布．

現不可能な関数である．実験で用いた各関数で用いたパラメータ設定[*1] は表 2 の通りとする．

$$\begin{cases} f_a(x) = (2 - 0.3x)\sin(2x)\cos(3x) + 0.01x^2 \\ f_b(x) = x\cos(x)\sin(x)(\sin^2(x)\cos(x) - 1) \\ f_c(x) = x^3\cos(x)\sin(x)e^{-x}(\sin^2(x)\cos(x) - 1) \end{cases}$$

図 6 は関数 f_a, f_b, f_c それぞれにおける最良個体の適合度（試行 50 回の平均）の推移と第 1000 世代における試行頻度分布を表したグラフである．適合度の推移グラフにおいて X 軸は経過世代数，Y 軸は適合度である．また，試行頻度分布のグラフでは X 軸の値 x_0 に対し，$x_0 \leq x$ であるような適合度 x を持つ最良個体が得られた試行の回数を Y 軸にプロットしている．図 6 のグラフにおいて最良個体の適合度の試行平均を見たとき，関数 f_a においては POLE が優れており，他の二つでは GPTM が良い成績となっている．

4.2 Royal Tree 問題

Royal Tree 問題 [Punch 96] は部分解を組み上げて全体の解を構築する問題であり，GPTM や PMBGP のように優良な部分解を保持する仕組みを備えた手法の効果を確かめるのに適

表 2: 記号当てはめ問題のパラメータ設定．

	意味	f_a	f_b	f_c
全手法 共通	集団数	200	200	200
	世代数	1000	1000	1000
	木の最大深さ	6	6	6
EDP	デフォルト比率 α	0.2	0.1	0.05
POLE	擬似カウント c	1.0	1.0	1.0
	依存関係の親の範囲 d	4	4	4

*1 表 2 において設定したデフォルト比率 α は 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 の中で最良のもの（最良個体の適合度の試行平均が最も良かったもの）である．同様に表 2 の擬似カウント c は 0.1, 0.5, 1, 5 の中で最良のものである．Truncate 選択確率が 0.2 であるとき，Truncate 選択では集団の 20% を優良個体として選択する．

表 3: Royal Tree 問題のパラメータ設定．

	意味	Royal Tree 1	Royal Tree 2
全手法 共通	集団数	5000	5000
	世代数	50	200
	木の最大深さ	6	7
EDP	デフォルト比率 α	0.01	0.005
POLE	擬似カウント c	0.5	0.5
	依存関係の親の範囲 d	2	4

している．今回は [Punch 96] で記述された Royal Tree 問題 (以下では Royal Tree 1 と呼ぶ) に加え，[長谷川 07] で検証に使われた設定による Royal Tree 問題 (以下では Royal Tree 2 と呼ぶ) でも比較実験を行った．Royal Tree 問題におけるプログラムは関数ノード A, B, C, ... と定数ノード x, y, ... で構成される．Royal Tree 1 では A が 1 引数，B が 2 引数というように関数ノードの引数長はアルファベット順に 1 ずつ増やす．一方，Royal Tree 2 では引数長は全て 2 とする．このとき perfect tree と呼ばれる状態が定義される．perfect tree とは，自分の子ノードが常に自分のアルファベットの辞書順で一つ前のアルファベットの関数ノードを持つ（関数 A に関しては定数ノード x を子ノードとする）状態である．

Royal Tree 問題では，以下のように計算されるスコアをプログラムの適合度とする．木全体のスコアは根のスコアを意味し，それぞれの関数ノードでは自分の子ノードのスコアに重みを乗じ，それらの和をとる．子ノードを根とする部分木が perfect tree であるならば，子ノードのスコアに full bonus (=2) を乗じた値を加える．子ノードはアルファベットの辞書順を守っているが，それを根とする部分木が perfect tree でない場合は partial bonus (=1) を乗じた値を加える．子ノードが辞書順に違反した場合は penalty (=1/3) を乗じた値を加える．さらに自分を根とする部分木が perfect tree である場合はスコア全体に complete bonus (=2) を乗ずる．Royal Tree 1 では定数記号が x の場合のみスコア 1 を与え，その他のスコ

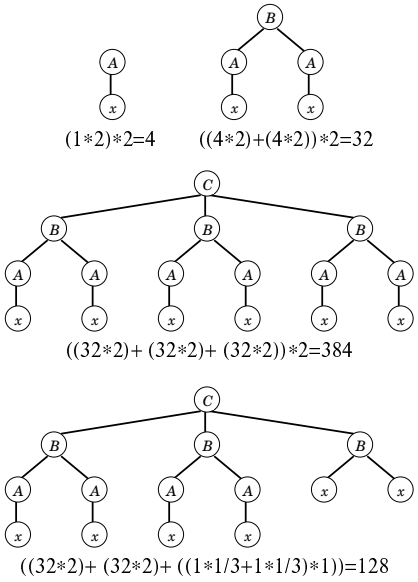

図 7: Royal Tree 1 における perfect tree のスコア例.

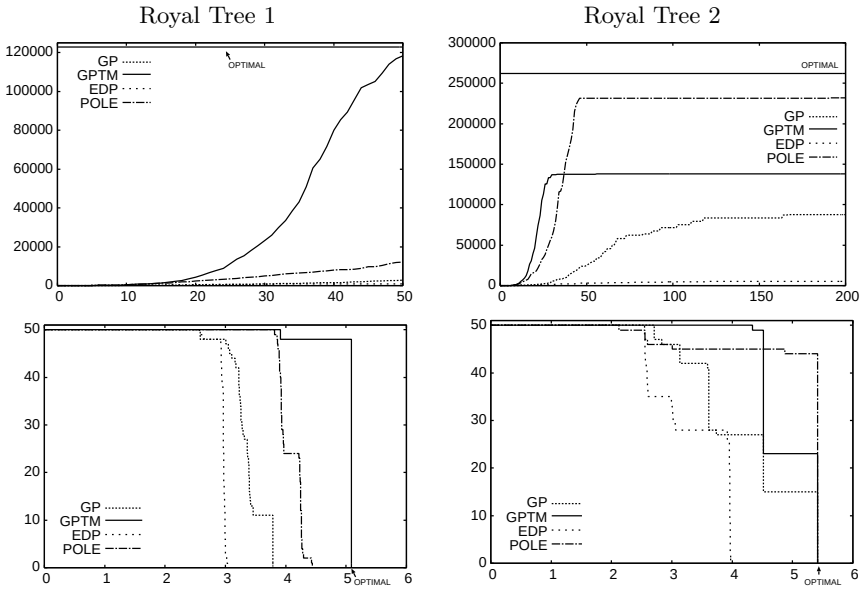

図 8: Royal Tree 問題における比較．上段：最良個体の適合度の推移，下段：第 50 世代 (Royal Tree 1)，第 200 世代 (Royal Tree 2) における試行頻度分布.

アは 0 とする．一方 Royal Tree 2 では定数記号が y のときにもスコア 0.95 を与える．図 7 は Royal Tree 1 のいくつかのプログラムに対してスコアを与えた例である．

関数記号集合として Royal Tree 1 では $\{A, B, C, D, E\}$, Royal Tree 2 では $\{A, B, C, D, E, F\}$ を与え，定数記号集合として Royal Tree 1, 2 ともに $\{x, y, z\}$ を与える[*2]．Royal Tree 1 における最適値は 122880 ($\approx 10^{5.09}$), Royal Tree 2 における最適値は 262144 ($\approx 10^{5.49}$) となる．実験で用いたパラメータ設定[*3] は表 3 の通りである．

図 8 の左側は Royal Tree 1 の最良個体の適合度の推移と第 50 世代における試行頻度分布のグラフである．ただし，試行頻度分布のグラフにおいて X 軸は適合度の対数値（底 = 10）としている．図 8 左上の推移をみると，GPTM が 50 世代あたりで最適値に概ね飽和したことが分かる．GPTM は世代を通じて最大の適合度を獲得し，ほとんどの試行において最適な個体を抽出することに成功している．また，図 8 の右側は Royal Tree 2 の最良個体の適合度の推移と第 200 世代における試行頻度分布のグラフである．Royal Tree 2 では，POLE が最も多くの試行で最適な個体を獲得していることが分かる．

5. まとめ

記号当てはめ問題と Royal Tree 問題に関し，四つの遺伝的プログラミング手法 GP, GPTM, EDP, POLE について性能の比較を行った．その結果，POLE と GPTM は他の手法に比べて良い性能を示した．また，問題設定により優劣があるが，POLE と GPTM は互角の性能を示したと言える．引き続き他のベンチマーク問題に適用すること，GPTM の改良（例えば，引数を区別しない関数記号のために無順序木発見手法を適用するなど）が今後の課題として挙げられる．

∗2 Royal Tree 2 の設定の多くは [長谷川 07] の設定に準じている．ただし，Royal Tree 1 では Royal Tree 2 に比べ分岐数が非常に大きくなるため，実験時間の都合上，関数記号数を一つ減らしている．

∗3 表 3 で設定したデフォルト比率 α は，0.001, 0.005, 0.01, 0.05, 0.1, 0.2 の中で最良のものであり，擬似カウント c は 0.1, 0.5, 1, 5 の中で最良のものである．また，先述したように，Royal Tree 1 では分岐数が非常に大きいため，実験時間の都合上，Royal Tree 1 の世代数を 50 とし，POLE の依存関係の親の探索範囲 d を 2 としている．更に，[長谷川 07] では POLE の初期集団生成における Grow の関数記号選択確率で 0.8 としていたが，Royal Tree 2 の予備実験では表 1 の 0.9 の方が全体的に良い結果が得られたため，0.9 を採用している．

参考文献

[Cooper 92] Cooper, G. and Herskovits, E.: A Bayesian method for the induction of probabilistic networks from data, *Machine Learning*, Vol. 9, pp. 309–347 (1992)

[Hasegawa 06] Hasegawa, Y. and Iba, H.: Optimizing Programs with Estimation of Bayesian Network, in *Proc. of the 2006 Congress on Evolutionary Computation (CEC-2006)*, pp. 1378–1385 (2006)

[Iba 93] Iba, H., Kurita, T., deGaris, H., and Sato, T.: System Identification using Structured Genetic Algorithms, in *Proc. of the 5th Intl. Conf. on Genetic Algorithms (ICGA-93)*, pp. 279–286 (1993)

[長谷川 07] 長谷川 禎彦, 伊庭 斉志：ベイジアンネットワーク推定による確率モデル遺伝的プログラミング, 人工知能学会論文誌, Vol. 22, No. 1, pp. 37–47 (2007)

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

[熊谷 07] 熊谷 潤一, 小島 康夫, 高重 聡一, 亀谷 由隆, 佐藤 泰介：頻出部分木発見手法を用いた遺伝的プログラミングの交通信号制御問題への適用, 人工知能学会論文誌, Vol. 22, No. 2, pp. 127–139 (2007)

[Punch 96] Punch, B., Zongker, D., and Goodman, E.: The Royal Tree Problem — a Benchmark for Single and Multi-population genetic Prgramming, in Angeline, P. and Kinnear, K. eds., *Advances in Genetic Programming II*, MIT Press (1996)

[Yanai 03] Yanai, K. and Iba, H.: Estimation of Distribution Programming based on Bayesian Network, in *Proc. of the 2003 Congress on Evolutionary Computation (CEC-2003)*, pp. 1618–1625 (2003)

[Yanai 04] Yanai, K. and Iba, H.: Program Evolution by Integrating EDP and GP, in *Proc. of the Genetic and Evolutionary Computation (GECCO-2004)*, pp. 774–785 (2004)

[浅井 02] 浅井 達哉, 安部 賢治, 川副 信治, 坂本 比呂志, 有村 博紀, 有川 節夫：半構造データからの頻出パターン発見アルゴリズム, 第 13 回データ工学ワークショップ（DEWS-2002）(2002)