

識別パターンを利用した遺伝的アルゴリズムの部分解保護に向けて

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[†]

1 はじめに

積み木仮説¹¹⁾に述べられているように、遺伝的アルゴリズム(以下GA)の進化過程においては部分解が生成され、またそれらの部分解が適切に組み合わせられることが期待される。そしてリンケージ学習(linkage learning)^{2, 12)}ではそのような部分解を直接的/間接的に抽出し、GAの探索性能の向上を狙う。これまで開発されてきたリンケージ学習手法は主に(1)リンケージ適応手法、(2)摂動に基づくリンケージ同定手法、(3)モデル構築手法の3種類に分類される¹³⁾。これらのリンケージ学習手法の多くは遺伝子の意味が座位に依存する表現(多くの場合はビット表現であるので、以下では単にビット表現と書くことにする)を対象としており、部分解は座位の単位で同定/保護される。また、我々の知る範囲において順列表現を扱える手法は多くない。

一方、新たなリンケージ学習法としてGAP(GA with patterns)と呼ばれる枠組みが著者らによって最近提案されている^{6, 7)}。GAPはGeroとKazakovの遺伝子工学(genetic engineering)アプローチ⁵⁾の拡張と見なすことができ、優れた染色体から特徴的なパターンを複数抽出し、これらのパターンの出現箇所が交叉によって破壊されることを防ぐ。GAPはパターン発見手法を利用してビット表現や順列表現を統一的に扱う点がGeroとKazakovの方法と異なる。更にGAPでは抽出されたパターンに基づき、細粒度(遺伝子単位)で各交叉における親染色体に依存して部分解の保護を行う。従って座位レベルで重複する部分解もGAPでは自然に扱うことができる。更に、摂動に基づく方法と異なり、GAPでは適合度関数の評価を追加的に行う必要がなく、むしろ逆に進化を加速させることで、適合度関数

の評価回数を減らすことを目指す。

また、上述のリンケージ学習手法では識別器の帰納的学習(inductive learning)により部分解(の座位)を同定する試みも行われている^{3, 15)}。例えばILI³⁾では摂動に基づいて決定木を構築し、そこに出現する座位を部分解を含む座位であると考え、一方、近年のデータマイニング分野では二つのデータ集合の違いを特徴付けるパターンを発見する識別(discriminative)パターン発見手法が subgroup discovery¹⁸⁾、顕在(emerging)パターン発見⁴⁾、contrast set mining¹⁾等の名で知られている。従来提案されたGAPでは頻出パターンの情報に基づき部分解を保護していたが、本論文では頻出パターンの代わりに識別パターンを利用してより正確な部分解の同定を試みる。

以下では次の構成をとる。2節では識別パターンを用いるGAPの詳細を記述する。3節で実験結果を示し、4節で結論を述べる。

2 提案手法

2.1 概略

図1に識別パターンを利用するGAPの概要を示す。ステップ3以外は既述のもの^{6, 7)}と同様である。GAPは基本的に通常のGAに従うが、ステップ2b, 3において良好な染色体に頻出するパターン Π を抽出し、ステップ4において Π に基づき交叉を行う点が異なる。 Δ_{mine} を非常に高い適合度を持つ集団、 Δ_{sel} を通常の選択操作で選ばれる集団($\Delta_{\text{mine}} \subseteq \Delta_{\text{sel}}$)としたとき、ステップ3では $\Delta^+ = \Delta_{\text{mine}}$ によく現れ、通常の選択操作でも選ばれない低い適合度を持つ集団 Δ^- にはあまり現れないパターンの集合を Π として得る。そして Π 中の各パターンを推定部分解(induced building block)と見なす。 Δ^+ と Δ^- の対比により、以前提案されたGAP^{6, 7)}に比べてより正確な部分解の推定が可能になると期待される。現在は、切捨て選択(truncation selection)を用いた完全な世代交代を考えており、突然変異操作は従来通り行う。また、ステップ3を抽出ステップと呼

Towards more precise protection of building blocks using discriminative patterns

[†] Yoshitaka Kameya (kameya@mi.cs.titech.ac.jp)

Dept. of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology (†)

-
1. 集団を初期化し, $\Delta^{(0)}$ とする. $t := 0$ とおく.
 2. $\Delta^{(t)}$ から切捨て選択により, 2種類の染色体集合を求める:
 - (a) Δ_{sel} (切捨て率 r_{sel}),
 - (b) Δ_{mine} (抽出切捨て率 r_{mine} , $r_{\text{mine}} \leq r_{\text{sel}}$)
 3. $\Delta^+ := \Delta_{\text{mine}}$, $\Delta^- := \Delta^{(t)} \setminus \Delta_{\text{mine}}$ とし, Δ^+ と Δ^- から識別パターン集合 Π を抽出する.
 4. Δ_{sel} に対し, Π に基づいて交叉を適用する. 突然変異を適用し, 新たな集団 $\Delta^{(t+1)}$ を得る.
 5. $t := t + 1$ と更新する.
 6. 停止条件を満たしたら終了する; そうでなければステップ2に進む.
-

Fig. 1 GAPの概要.

び, 一方ステップ4中で推定部分解を保護する操作を保護ステップと呼ぶ. 保護ステップでは既存のもの^{6, 7)}と同様の手続きをとる. 次の2.2節で抽出ステップ, 2.3節で保護ステップを説明する.

2.2 識別パターン抽出

抽出ステップでは高適合な集団 Δ^+ と低適合な集団 Δ^- をデータベースと見なし, 両者の違いを際立たせる識別パターンを抽出する. まず準備として, 順列表現では染色体を単純に遺伝子の系列データとして表し, ビット表現ではmessy GAと同様に, 染色体を座位-遺伝子(ビット)の対の集合と見なす. 例えば, 順列表現の染色体 $\langle a, c, d, e, b \rangle$ はそのまま系列データとして扱い, ビット表現の染色体 $\langle 1, 0, 1, 1, 0 \rangle$ は座位-遺伝子対の集合 $\{(1, 1), (2, 0), (3, 1), (4, 1), (5, 0)\}$ に翻訳して考える. そのとき, パターン s は順序表現において染色体の部分系列であり, ビット表現においては染色体の部分集合である. 例えば, ビット表現においてパターン $\{(2, 0), (4, 1)\}$ はスキーマ記法では $\langle *, 0, *, 1, * \rangle$ と書かれる. また, 集団 Δ 中でパターン s を含む染色体の数を $\sigma(s, \Delta)$ と記す.

あるパターン s が Δ^+ と Δ^- の違いを際立たせていることを表す指標として情報検索分野で知られるF値を用いる. あるパターン s のF値は

$$F(s) = \frac{2 \cdot \text{Recall}(s)\text{Precision}(s)}{\text{Recall}(s) + \text{Precision}(s)}$$

と定義される. ここで

$$\text{Recall}(s) = p(s | \Delta^+) = \frac{\sigma(s, \Delta^+)}{|\Delta^+|}$$

$$\text{Precision}(s) = p(\Delta^+ | s) = \frac{\sigma(s, \Delta^+)}{\sigma(s, \Delta^+) + \sigma(s, \Delta^-)}$$

である. $\text{Recall}(s)$ は再現率と呼ばれ, 高適合な集団 Δ^+ の中でパターン s を満たす染色体の割合である. 一方 $\text{Precision}(s)$ は精度と呼ばれ, パターン s を満たす染色体が高適合な集団 Δ^+ に属する確率である. 一般に再現率と精度はトレードオフの関係にあり, 両者の調和平均をとったものがF値である. 例えば集団全体において殆どの染色体に現れるありふれたパターンは精度が低くなるため, F値も低くなる. 逆にあるパターン s の精度が非常に高くても再現率が低ければノイズである可能性が高い. 従って再現率, 精度共に高いパターンが理想的であり, GAPでは両者の調和平均であるF値の高いパターン上位 K 個を抽出することを考える[†].

F値を用いて識別パターンを得る上での問題点は, F値がパターンの包含関係において反単調性(anti-monotonicity)を満たさないことである. すなわち $s \subset s'$ なる二つのパターン s, s' に対して $F(s) \geq F(s')$ が常に成り立たないため, 頻出パターン発見で行うような強力な枝刈りが利用できない^{††}. そこでGAPでは分岐限定(branch-and-bound)法を利用した枝刈りを行う. また, $s \subset s'$ である (s' が s に比べて余分な要素を持つ)にも関わらず $F(s) \geq F(s')$ である (s' のF値が s に比べて向上していない)とき, 「 s' は s より弱い」と言い, s' を冗長であるとして取り除く[‡]. これらのアイディアの詳細は著者らが識別パターン発見手法 RP-growth を提案した論文⁸⁾に記述されている.

更に, 以前の提案^{6, 7)}と同様, 抽出するパターンは Δ^+ 上で飽和するパターンに限定し, 効率のよい飽和パターン発見アルゴリズムとして知られるLCM¹⁶⁾ (ビット表現の場合), BIDE¹⁷⁾ (順序表現の場合)を用いる. 上位 K (top- K) パターン発見手法, あるいはパターンの長さ(最大長 L_{max} , 最小長 L_{min}) や出現位置のギャップ

[†]それに対し, 以前提案したGAPでは実質的に再現率のみをパターンの好ましさの指標としていたため, ありふれたパターンが抽出され易いという問題があった.

^{††}一方, 従来用いていた再現率(支持度)では反単調性が常に成り立ち, そのような枝刈りが可能であった.

[‡]もちろん $s \subset s' \subset s''$ なる s'' について $F(s) < F(s')$ となる可能性があるため, 直ちに枝刈りはできない.

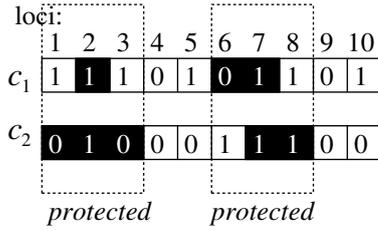


Fig. 2 一様交叉に対する部分解保護 .

幅に関する制約 (最大幅 G_{\max} , 最小幅 G_{\min}) も以前の提案と同様に導入する .

2.3 部分解保護

推定部分解を識別パターンとして得た後, その部分解を次世代に継承し, 組み合わせる必要がある . GAPでは親染色体における推定部分解の出現位置に基づき交叉点の選択確率分布 (以下, 交叉点分布) を修正することで交叉による部分解の破壊を防ぐ . 一点交叉, 二点交叉, オリジナルのエッジ交叉 (edge recombination), 位置に基づく交叉 (position-based crossover) に対する修正手続きは以前提案されたもの^{6, 7)} を用いる . 一方, 本論文では一様交叉に対する交叉点分布修正手続きを述べる .

図 2は長さ10の親染色体 c_1, c_2 に対する一様交叉での部分解保護の様子を示している . 抽出ステップで得られた識別パターンがこの親染色体上の黒く塗りつぶされた座位に出現しているとす . 通常, 一様交叉では各座位において独立に確率 0.5 でビットの入れ替えを行うが, c_1, c_2 どちらかの親に識別パターンが出現している座位において入れ替え確率を 0.5δ とする . ここで δ は割引率 (discount rate) と呼ばれる, ユーザ指定のパラメータである . 図 2においては座位1, 2, 3, 6, 7, 8において c_1, c_2 どちらかに識別パターンが出現しているので, これらの座位は (ソフトに) 保護されている .

3 実験

Royal Road問題, 巡回セールスマン問題の2つについて実験を行った . Royal Road問題ではビット表現の染色体を用い, 巡回セールスマン問題では順列表現の染色体を用いる . Royal Road問題はMitchellらによって導入された¹¹⁾ . GAの巡回セールスマン問題への適用に関しては, Larrañagaらが標準的なベンチマークデータを用いた詳細な比較実験を行っている⁹⁾ . 以前の

Table 1 Royal Road問題に対する制御パラメータ .

手法	制御パラメータ	値
SGA	交叉確率	0.7
GAP	突然変異確率	0.01
	切捨て率 (r_{sel})	0.5
GAP	最大パターン数 (K)	100
	最小パターン長 (L_{\min})	3
	抽出用切捨て率 (r_{\min})	0.05, 0.1, 0.2
	割引率 (δ)	0.1, 0.3, 0.5, 0.7, 0.9, 0.95
	初期最小サポート ($\sigma_{\min}^{(0)}$)	$\max\{5, (0.2 \cdot r_{\min} Pop)\}$
BOA	切捨て率	0.01, 0.05, 0.1, 0.2
	親変数の最大個数	1, 2, 5
	疑似カウント (C)	0.01, 0.05, 0.1, 0.5, 1

GAPの提案論文^{6, 7)} と同様, いずれの問題においても, 基本的に我々は標準的なGA (standard GA, 以下SGA) とGAPを比較し, 推定部分解の保護が探索性能を向上させるかどうかを観察する . そしてRoyal Road問題に対しては, 確率モデル構築に基づくGAとして広く知られるBOA¹⁴⁾ (Bayesian Optimization Algorithm) とも比較する .

3.1 Royal Road問題

Royal Road問題は, 染色体 c に対する適合度関数 $F(c) = \sum_{s \in S} w(s)\sigma(s, c)$ を最適化する問題である . ここで S はユーザ定義のスキーマの集合であり, $w(s)$ はスキーマ $s \in S$ に対する重みである . また $\sigma(s, c)$ はスキーマ s が染色体 c に出現すれば1, そうでなければ0をとる . 交叉に対する符号化の影響を調べるため, Royal Road問題を用いた比較実験では, “tight” な符号化に基づく適合度関数と “loose” な符号化に基づく適合度関数の2種類を用意する . スキーマ集合と重みは図3で与えられており, “tight” な符号化では隣り合うビットからスキーマが構成され, スキーマの破壊が起こりにくくなっている . 一方 “loose” な符号化ではスキーマが染色体内に散在しており, 交叉によるスキーマの破壊が起こりやすい .

比較実験においては, 集団サイズ $|Pop|$ を64から512まで変え, 進化計算手法としてSGA (一点交叉, 二点交叉, 一様交叉), 頻出パターンを用いるGAP (一点交叉, 二点交叉, 一様交叉), 識別パターンを用いるGAP (一点交叉, 二点交叉, 一様交叉; 提案手法), BOAの10手法を比較した . Mitchellらの実験では集団サイズは128に固定されている . また, SGAとGAPでは一様突然変異を用いる . 進化サイクルは最適値256を見つける

Table 2 Royal Road問題における比較結果 .

“Tight”	SGA			BOA
	Pop	1PTX	2PTX	
64	611.94 ± 33.59	518.14 ± 28.04	533.93 ± 28.17	355.57 ± 23.88
128	276.76 ± 19.21	212.95 ± 14.80	235.58 ± 14.20	208.33 ± 10.99
256	96.55 ± 8.06	72.41 ± 7.13	104.93 ± 5.58	154.86 ± 8.54
512	26.96 ± 1.58	25.63 ± 1.09	70.10 ± 2.65	80.44 ± 3.82

“Tight”	GAP (頻出パターン利用)			GAP (識別パターン利用)		
	Pop	1PTX	2PTX	UX	1PTX	2PTX
64	764.36 ± 48.99	494.53 ± 23.49	468.01 ± 23.47	747.27 ± 34.58	491.04 ± 24.04	483.28 ± 21.03
128	340.35 ± 18.53	189.20 ± 13.64	206.11 ± 12.76	344.14 ± 17.30	177.72 ± 12.46	206.14 ± 11.10
256	152.04 ± 12.15	56.16 ± 4.35	92.34 ± 3.90	133.43 ± 10.57	54.44 ± 3.90	98.92 ± 4.68
512	43.32 ± 3.72	24.83 ± 0.89	63.01 ± 1.77	45.81 ± 4.37	23.40 ± 0.53	53.02 ± 1.39

“Loose”	SGA			BOA
	Pop	1PTX	2PTX	
64	607.23 ± 28.36	509.62 ± 21.67	510.69 ± 30.62	384.14 ± 24.45
128	310.16 ± 11.71	273.17 ± 13.02	224.91 ± 10.02	201.55 ± 8.59
256	194.72 ± 7.20	153.96 ± 7.90	106.72 ± 4.40	150.78 ± 8.21
512	121.35 ± 4.28	96.06 ± 3.18	69.23 ± 2.37	72.79 ± 2.92

“Loose”	GAP (頻出パターン利用)			GAP (識別パターン利用)		
	Pop	1PTX	2PTX	UX	1PTX	2PTX
64	677.58 ± 36.13	527.60 ± 24.77	463.36 ± 23.32	670.69 ± 32.33	535.61 ± 22.87	470.08 ± 22.42
128	371.19 ± 15.10	282.77 ± 12.68	213.04 ± 10.93	372.03 ± 14.74	276.34 ± 12.47	219.52 ± 11.21
256	264.08 ± 13.70	155.76 ± 5.96	105.03 ± 4.86	274.04 ± 10.86	153.82 ± 5.58	96.00 ± 5.96
512	167.07 ± 8.10	94.20 ± 3.62	64.22 ± 2.51	161.43 ± 7.31	88.70 ± 2.64	57.21 ± 2.62

供される Grötschels48, Grötschels120 を使って比較する . Grötschels48 には 48 都市が含まれ , 最適解は 5,046 である . 一方 Grötschels120 には 120 都市が含まれ , 最適解は 6,942 である . この実験では , 集団サイズを 200 から 5,000 まで変えて SGA (エッジ交叉 , 位置に基づく交叉) , 頻出パターンを用いる GAP (エッジ交叉 , 位置に基づく交叉) , 識別パターンを用いる GAP (エッジ交叉 , 位置に基づく交叉) の 6 種類を比較する . 実験に使用した制御パラメータを表 3 に示す . この実験でも以前の報告^{6, 7)} と同様の設定を行っている .

Grötschels48 に対する実験結果を表 4 に示す . 表中の “ER” および “PX” はそれぞれエッジ交叉 , 位置に基づく交叉を指す . 表 4 (上) の各エントリは , 100 回の試行における最良個体の適合度の平均である¹ . 一方 , 表 4 (下) の各エントリは , 進化サイクルが停止するまでに要した世代数の平均である (制御パラメータの設定は表 4 (上) と同じ) . これらの表から , 特にエッジ交叉を用

Table 3 巡回セールスマン問題に対する制御パラメータ .

手法	制御パラメータ	値
SGA GAP	交叉確率	0.7
	突然変異確率	0.05
	切捨て率 (r_{sel})	0.5
	位置保存率 (r_{pos})	0.25 (for PX only)
GAP	最大パターン数 (K)	100
	最小パターン長 (L_{min})	2
	最大ギャップ幅 (G_{max})	0, 1, 2
	抽出用切捨て率 (r_{mine})	0.01, 0.05, 0.1
	割引率 (δ)	0.1, 0.3, 0.5, 0.7, 0.9, 0.95
	初期最小サポート ($\sigma_{min}^{(0)}$)	$\max\{5, (0.2 \cdot r_{mine} Pop)\}$

いる場合には² , GAP が最良個体の適合度を向上させ , 停止までに要する世代数を減らしていることが分かる . 更に識別パターンを用いた場合はその向上が顕著になる . 従ってこの実験結果より , エッジ交叉を用いた場合に識別パターンを用いた GAP は従来法に比べて進化を早めることに成功したと言える . また , 停止までの世代数は適合度関数の評価回数に比例するため ,

¹ ただし , 識別パターンを用いる GAP において集団サイズ $|Pop| = 5000$, 位置に基づく交叉という条件では幾つかの場合に 100 試行のうち 1 つで実行が終了しなかったため , そのような場合は進化途中の結果を用いている .

² 位置に基づく交叉については位置保存率 (表 3) の調整が不足している可能性がある .

Table 4 Grötschels48問題における比較結果 .

適合度	SGA		GAP (頻出パターン利用)		GAP (識別パターン利用)	
	ER	PX	ER	PX	ER	PX
200	5441.72 ± 14.27	5665.14 ± 31.62	5430.04 ± 13.34	5635.91 ± 29.41	5440.36 ± 13.84	5641.62 ± 34.01
500	5151.96 ± 5.47	5326.22 ± 21.76	5146.87 ± 5.56	5265.88 ± 14.01	5146.87 ± 5.56	5265.88 ± 14.01
1000	5111.16 ± 4.19	5224.71 ± 12.70	5100.65 ± 3.40	5224.29 ± 11.06	5104.37 ± 3.23	5228.05 ± 10.42
2000	5098.29 ± 2.88	5188.95 ± 9.36	5085.11 ± 3.19	5195.01 ± 11.42	5083.66 ± 2.02	5184.72 ± 8.97
5000	5085.22 ± 2.00	5161.49 ± 4.81	5070.18 ± 1.75	5162.54 ± 6.26	5067.85 ± 1.94	5164.30 ± 53.10

世代数	SGA		GAP (頻出パターン利用)		GAP (識別パターン利用)	
	ER	PX	ER	PX	ER	PX
200	157.78 ± 2.96	318.30 ± 25.64	156.18 ± 3.14	341.73 ± 29.39	159.84 ± 3.30	325.00 ± 26.94
500	165.11 ± 2.27	406.17 ± 45.90	162.56 ± 1.75	296.09 ± 15.84	162.56 ± 1.75	296.09 ± 15.84
1000	164.61 ± 1.53	358.43 ± 39.83	135.88 ± 1.62	413.61 ± 47.50	160.14 ± 1.72	389.72 ± 46.04
2000	166.12 ± 1.85	445.82 ± 54.95	139.30 ± 1.81	427.14 ± 55.98	140.84 ± 1.85	423.15 ± 55.95
5000	163.89 ± 1.89	467.53 ± 58.72	142.17 ± 1.50	514.13 ± 63.99	131.11 ± 1.86	689.79 ± 76.20

Table 5 Grötschels120問題における比較結果 .

適合度	SGA		GAP (頻出パターン利用)		GAP (識別パターン利用)	
	ER	PX	ER	PX	ER	PX
200	9434.51 ± 34.81	9742.30 ± 69.11	9512.32 ± 39.47	9568.72 ± 69.35	9478.09 ± 33.05	9592.61 ± 67.67
500	8410.97 ± 29.11	8075.37 ± 34.20	8394.76 ± 30.16	8054.77 ± 34.76	8390.74 ± 26.67	8052.16 ± 34.99
1000	8258.13 ± 28.88	7733.09 ± 29.17	8213.80 ± 28.55	7768.65 ± 21.79	8185.73 ± 25.44	7674.48 ± 23.11
2000	8148.24 ± 34.78	7579.01 ± 21.06	8038.45 ± 28.61	7560.97 ± 22.14	7858.24 ± 26.98	7551.56 ± 21.18
5000	8012.60 ± 47.68	7489.21 ± 18.18	7733.46 ± 23.35	7486.95 ± 17.48	7508.78 ± 16.66	7474.45 ± 16.89

世代数	SGA		GAP (頻出パターン利用)		GAP (識別パターン利用)	
	ER	PX	ER	PX	ER	PX
200	556.99 ± 8.73	488.16 ± 10.93	539.23 ± 10.17	477.36 ± 10.30	530.41 ± 7.69	485.92 ± 16.14
500	874.70 ± 20.07	528.17 ± 17.80	773.99 ± 15.05	502.17 ± 11.29	854.41 ± 21.93	488.74 ± 10.04
1000	1039.94 ± 20.83	574.18 ± 31.10	678.61 ± 12.09	520.49 ± 15.74	754.46 ± 16.05	521.31 ± 23.09
2000	1275.03 ± 29.39	649.71 ± 39.95	519.92 ± 15.03	519.88 ± 34.46	458.27 ± 10.75	614.96 ± 37.20
5000	1561.83 ± 32.36	826.98 ± 57.37	597.20 ± 21.85	897.67 ± 62.69	359.83 ± 14.96	886.89 ± 60.21

GAPは適合度関数評価にかかるコストを軽減している . 同様の傾向は表 5 に示す Grötschels120 の結果でも観察できる .

4 おわりに

本論文ではパターンに基づき遺伝的アルゴリズムの部分解を保護する枠組みである GAP において識別パターンを用いる手法を提案した . また , Royal Road 問題と巡回セールスマン問題を用いたベンチマーク比較により , 従来手法に比べて提案手法の探索性能の向上を確認した .

今後の課題としては , まずリンクージ学習研究でよく用いられるトラップ問題のような騙し要素を含む問題への適用が挙げられる . また , 現在はある世代で得られた識別パターンの情報を次世代の個体生成でしか活用していないが , 複数世代に跨って部分解の維持・管理を行う仕組み (部分解ライブラリ) の導入も考えられる .

更に , 識別パターンの抽出における省力化のため , 識別パターンの近似探索を検討する . Le らは頻出パターンを用いて遺伝的アルゴリズムの進化過程を解析しており¹⁰⁾ , GAPで同様の解析を行うことも今後の課題である .

参考文献

- 1) S. D. Bay and M. J. Pazzani. Detecting group differences: mining contrast sets. *Data Mining and Knowledge Discovery*, Vol. 5, pp. 213–246, 2001.
- 2) Y.-p. Chen, T.-L. Yu, K. Sastry, and D. E. Goldberg. A survey of linkage learning techniques in genetic and evolutionary algorithms. IlliGAL Report No.2007014, University of Illinois at Urbana-Champaign, 2007.
- 3) C.-Y. Chuang and Y.-p. Chen. Linkage identification by perturbation and decision tree in-

- duction. In *Proc. of the 2007 IEEE Congress on Evolutionary Computation (CEC-07)*, pp. 357–363, 2007.
- 4) G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In *Proc. of the 5th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 43–52, 1999.
 - 5) J. S. Gero and V. Kazakov. A genetic engineering approach to genetic algorithms. *Evolutionary Computation*, Vol. 9, No. 1, pp. 71–92, 2001.
 - 6) Y. Kameya and C. Prayoonsri. Pattern-based preservation of building blocks in genetic algorithms. In *Proc. of the 2011 IEEE Congress on Evolutionary Computation (CEC-11)*, pp. 2578–2585, 2011.
 - 7) 亀谷由隆, C. Prayoonsri. パターンに基づく遺伝的アルゴリズムの部分解保護. 人工知能学会第6回進化計算フロンティア研究会 (SIG-ECF) 予稿集, 2011.
 - 8) 亀谷由隆, 佐藤泰介. 最小サポート上昇法に基づく上位 k 関連パターン発見. データ指向構成マイニングとシミュレーション研究会 (人工知能学会創立25周年記念合同研究会・シンポジウム), 2011. 発表予定.
 - 9) P. Larrañaga, C. Kuijpers, R. Murga, I. Inza, and S. Dizdarevich. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, No. 13, pp. 129–170, 1999.
 - 10) M. N. Le and Y.-S. Ong. A frequent pattern mining algorithm for understanding genetic algorithms. In *Proc. of the 4th Int'l Conf. on Intelligent Computing*, pp. 131–139, 2008.
 - 11) M. Mitchell, S. Forrest, and J. H. Holland. The royal road for genetic algorithms: fitness landscapes and GA performance. In *Proc. of the 1st European Conf. on Artificial Life (ECAL-92)*, pp. 245–254, 1992.
 - 12) 棟朝雅晴. 遺伝的アルゴリズム — その理論と先端的手法. 森北出版, 2008.
 - 13) M. Munetomo and D. E. Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. *Evolutionary Computation*, Vol. 7, No. 4, pp. 377–398, 1999.
 - 14) M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Linkage problem, distribution estimation, and Bayesian networks. *Evolutionary Computation*, Vol. 8, No. 3, pp. 311–340, 2000.
 - 15) M. Sebag and M. Schoenauer. Controlling crossover through inductive learning. In *Proc. of the 3rd Int'l Conf. on Parallel Problem Solving from Nature (PPSN III)*, pp. 209–218, 1994.
 - 16) T. Uno, M. Kiyomi, and H. Arimura. LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets. In *Proc. of the IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI-04)*, 2004.
 - 17) J. Wang and J. Han. BIDE: Efficient mining of frequent closed sequences. In *Proc. of the 20th Int'l Conf. on Data Engineering (ICDE-04)*, pp. 79–90, 2004.
 - 18) S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proc. of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97)*, pp. 78–87, 1997.