

バンド幅問題に対する 遺伝的アルゴリズムの交叉方法の提案

群馬大学・工学部 佐藤 竜也 (Tatsuya Sato) 山崎 浩一 (Koichi Yamazaki)
 Department of Computer Science,
 Gunma University

1 はじめに

バンド幅問題は様々な分野に応用を持つため古くからよく研究されている。バンド幅問題に対する発見的手法として, Cuthill-McKee(CM) 法 [6] や Gibbs Pole Stockmayer(GPS) 法 [9] などが古くからよく知られている。最近においても, volume respecting embedding 法 [8] や半正定値計画法 [1] を用いた近似アルゴリズムや CM 法を基本とした WBRA 法 [7] などが提案されている。

2003 年に Lim 等はバンド幅問題に対する新しいアルゴリズム (GA+HC 法) を提案した [11, 12]。GA+HC 法は, 山登り法 (HC) と遺伝的アルゴリズム (GA) を組み合わせたもので, そこで用いられている遺伝的アルゴリズム (の交叉方法) は極めて単純なものであるにも拘らず, タブサーチ [14] や GPS 法より良い値を出す事が報告されている。

本論文では, Lim 等が提案した GA+HC 法の (山登り法については手を加えず) 遺伝的アルゴリズムのみを改良し実験評価を行う。

2 諸定義及び準備

$G = (V, E)$ をグラフする。 V, E はそれぞれ $V(G), E(G)$ で表す事もある。 $v \in V(G)$ に対し, v と隣接する頂点からなる集合を $N(v)$ で表す。 全単射 $f : V \mapsto \{1, \dots, |V|\}$ を G の (頂点) 配置と呼び, G の配置の集合を $LO(G)$ で表す。 配置 f と頂点 v に対し, 値 $f(v)$ を (v の f による) 配置位置と呼ぶ。 f の並び順を逆にした並び, すなわち $f'(v) = n - f(v) + 1$ を f^R で表す。 頂点集合 $\{v \mid i \leq f(v) \leq j\}$ を $f[i, j]$ で表す。 f を G のある配置とする。 (G の) f に対するバンド幅 $bw_f(G)$ は $bw_f(G) = \max_{\{u,v\} \in E} |f(u) - f(v)|$ で定義され, G のバンド幅 $bw(G)$ は $bw(G) = \min_{f \in LO(G)} bw_f(G)$ で定義される。 各頂点 v に対し, $\max\{|f(u) - f(v)| : u \in N(v)\}$ を $bw_f(v)$ で表す。 M を n 次の対称正方行列とし, $M_i = (m_{i1}, \dots, m_{i|V|})$ を M の i 行目のベクトルとする。 対角成分 m_{ii} から一番離れている非零成分までの距離, $\max_{m_{ij} \neq 0} |i - j|$ を $bw_i(M)$ で表す。 このとき $\max_{1 \leq i \leq |V|} bw_i(M)$ を $bw(M)$ で表し, 行列 M のバンド幅と呼ぶ。 PM_n をサイズ n の置換行列の集合とする。 $b = \min_{P \in PM_n} bw(PMP^T)$ のとき, 行列 M は最小バンド幅 b を持つ行列に変換可能であると言う。 グラフのバンド幅を求める問題や, 行列を最も小さいバンド幅を持つ行列に変換する問題をバンド幅問題と呼ぶ。

2.1 バンド幅問題について

定義から想像できるように, バンド幅問題は様々な分野で, 特にグラフや行列によりモデル化される問題で, 見かけることが出来る。 良く知られた例として, 帯行列に対するガウスの消去法がある。 M を n 次の対称正方行列とする。 もしバンド幅問題を解くことにより, M をバンド幅 b を持つ行列に変換できたならば, (通常 $O(n^3)$ かかる) ガウスの消去法は $O(nb^2)$ で実行できる。 従って, $b \ll n$ の場合計算コストを大きく削減できる。

この例から分かるように, バンド幅問題の効率の良い解法を求めることは, 関連する実際の工学的問題の効率の良い解法を求めることに繋がる。 しかしながら残念なことに, バンド幅問題は計算量的に見て難しい問題であることが知られている。 例えば, 入力グラフを最大次数が 3 である木に制限したとしても, 対応する判定問題が NP 完全であることが知られている¹ [10]。 また, 入力を木に制限しても, 近似率は $4/3$ より良くなる¹ [3] し, 入力に何も制限を加えなければ ($P \neq NP$

¹ 良く知られている問題で, 木に制限しても NP 完全であるものはあまり無い (他には例えば subgraph isomorphism や achromatic number など)

の仮定の下で) 定数近似は不可能であることが知られている [17]. さらに, 与えられた正整数 k に対しバンド幅が k 以下であるかの判定問題は実質 brute force で行う方法しかないことが予想されている [4].²

2.2 レベル構造

バンド幅問題に対して従来からよく用いられている CM 法や GPS 法などの発見的手法は, レベル構造を構成している. グラフ $G = (V, E)$ の頂点の部分集合の列 (L_1, \dots, L_M) がレベル構造を成すとは, 1) $\sum_{i=1}^M L_i = V$, 2) $\{u, v\} \in E, u \in L_i, v \in L_j \Rightarrow |i - j| \leq 1$ を満たすときをいう. $\max_{1 \leq i \leq M} |L_i|$ をレベル構造 (L_1, \dots, L_M) の幅と呼ぶ. グラフ G のレベル構造での最小の幅を $lw(G)$ で表すと, $lw(G) \leq bw(G) \leq 2lw(G) - 1$ という関係が成り立つ. 従って, もし最小の幅を持つレベル構造を見つけることができれば, バンド幅の 2 倍近似が実現できることになる. しかしながら, 最小の幅を持つレベル構造を見つけることは NP 困難であることが知られている [2]. 従って, 最小の幅は保証しないが, 比較的簡単にレベル構造が実現できる, 距離に基づいてレベルを構成する方法がよく用いられる.

R_0 を頂点のある部分集合とし, R_i を R_0 から距離 i にある頂点の集合とする. このとき R_0, \dots, R_h (h は R_0 から最も遠い頂点までの距離) はレベル構造を成す. 通常 R_0 のサイズは 1 である. 距離に基づいてレベルを構成する方法は手軽であるが, (理論的には) それにより得られるレベル構造の幅が最小の幅 $lw(G)$ よりかなり大きくなる場合が起こり得る. また距離に基づいて構成されたレベルの中での最小幅を求める問題は, たとえグラフを木に制限したとしても, $4/3$ 未満に近似率を下げられないことが知られている [18]. 実際距離に基づいて得られたレベル構造の幅が $O(lw(G) \log |V|)$ となる (人工的な) グラフは簡単に作ることが出来る. しかしながら一方では, 距離に基づいたレベル構造を用いた発見的手法がランダムグラフなどに対してはうまく働くことが知られている [16].

2.3 Cuthill-McKee(CM) 法

CM 法は Cuthill と McKee により 1960 年代後半に提案された方法で, レベル構造を用いている [6]. レベル構造に基づいて設計されているアルゴリズムとしては他に, RCM 法, GPS 法や WBRA 法などがある. 以下に CM 法の概略を説明する: 先ず, 各頂点 v に対し, $R_0 = \{v\}$ とし距離に基づきレベル構造を生成し, そのレベル構造の幅を計算する³. 次に, 得られた幅の中で最小の幅を持つレベル構造 R_0, \dots, R_h を選ぶ. そして, $u \in R_i, v \in R_j, i < j \Rightarrow f(u) < f(v)$ となるようにある規則に従って各頂点の並び (すなわち $f: V \rightarrow \{1, \dots, |V|\}$) を計算する⁴.

3 Lim 等の GA+HC アルゴリズム

Lim 等の GA+HC 法の, 山登り法 (HC) と遺伝的アルゴリズム (GA) の概略を説明する.

3.1 Lim 等の山登り法

グラフ G のある配置 f に対し, 頂点 $v \in V(G)$ が $bw_f(v) = bw_f(G)$ を満たす時, v を臨界頂点と呼ぶ. f に対する臨界頂点の集合を $CV_f(G)$ で表す. $CV_f(G)$ に属さない頂点を非臨界頂点と呼ぶ. ある配置 f とある 2 つの頂点 u, v に対し, u と v に関する 2 つの配置位置だけを交換して出来る配置 $f'(v) = f(u), f'(u) = f(v), f'(w)(w \neq u, v) = f(w)$ を $swap(f, u, v)$ で表す. 以下に山登り法の概略を示す.

² 正確には fixed parameterized intractable であることが知られている.

³ 計算時間を節約したい場合, 次数の低い頂点 v のみに対し $R_0 = \{v\}$ の計算を行う

⁴ 規則の詳細な記述は文献 [5] を参照

Procedure HC入力: グラフ $G = (V, E)$ 出力: G の配置

```

1:  $f_w := f$ ;
2: repeat
3:    $flag := true$ ;  $CV := CV_{f_w}(G)$ ;  $width := bw_{f_w}(G)$ ;
4:   foreach  $v \in CV$  do
5:     begin
6:       ある規則に従い非臨界頂点からなるある集合  $N'(v) \subseteq V - CV$  を計算する;
7:       ある規則に従い  $N'(v)$  をソートする (ソート結果を  $(w_1, \dots, w_t)$  で表す);
8:        $i := 0$ ;
9:       repeat
10:         $i := i + 1$ ;  $f' := swap(f_w, v, w_i)$ ;  $m := \max\{bw_{f'}(v), bw_{f'}(w_i)\}$ ;
11:        until  $(m < width)$  or  $(i = t)$  do;
12:        if  $m < width$  then begin  $f_w := f'$ ;  $flag := false$ ; end;
13:      end;
14:   until  $flag$  do;
15:    $f_w$  を出力;

```

3.2 Lim 等の遺伝的アルゴリズム

$G = (V = \{v_1, \dots, v_n\}, E)$ をあるグラフとし, f を G のある配置とする.

染色体 Lim 等の方法では, 配置位置の列 $f(v_1), \dots, f(v_n)$ を個体 f に対する染色体として定義している. 例えば, 図1の配置 f に対する染色体は, 5,2,6,1,3,4 となる.

交叉 Lim 等の方法では, 中央を交叉点とする1点交叉を用いている. 娘(息子)の染色体の前半は父(母)親の染色体の前半をそのまま使用し, 染色体の後半は未使用の配置位置を母(父)親の染色体の順に並べ使用する. 例えば, 図2の例では, (偶然だが) 母の染色体と息子の染色体が一致している.

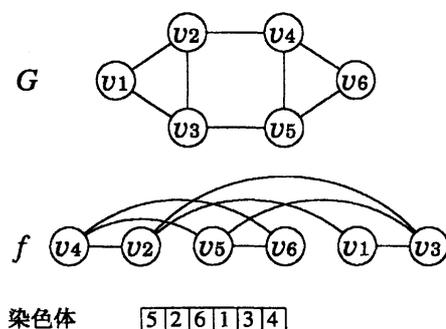


図1: 染色体の例

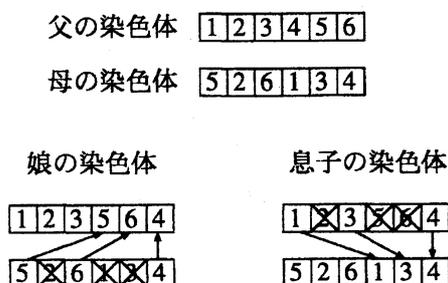


図2: 交叉の例

突然変異 k をパラメータとして, ランダムに選ばれた2点間の(位置の)入れ換えを k 回行う. 例えば, 図3の例では, $k=2$ とし, それぞれ1と6, 2と4の間で位置の入れ換えが行われている.



図3: 突然変異の例

選択 交叉と突然変異を行った後, それぞれの染色体に山登り法を適用する. 次世代を構成するのに必要な個体数を適応度の高い順に(現世代から)選ぶいわゆるエリート戦略を行う.

3.3 GA+HC アルゴリズム

Lim 等の GA+HC アルゴリズムの概略を以下に記す. 以下では, ps (population size) を個体数, mr (mutation rate) を突然変異を起こす割合, cr (crossover rate) を交叉を行う割合, mg (maximum generation) を生成する世代数とする.

Procedure GA+HC

入力: グラフ $G = (V, E)$

出力: 各世代の個体(配置)の集合

- 1: G の頂点を ps 個ランダムに選ぶ; /* 初期集団の生成 */
- 2: **foreach** $v \in U$ **do** /* 選ばれた頂点の集合を U で表す */
- 3: v を根として Cuthill-McKee 法を適用する; /* 得られた配置の集合を \mathcal{F}^w で表す */
- 4: **foreach** $f \in \mathcal{F}^w$ **do**
- 5: f に山登り法を適用する; /* 得られた配置の集合を \mathcal{F}_1 で表す */
- 6: $g := 1$;
- 7: **for** $g = 1$ **to** mg **do** /* 次世代の生成 */
- 8: $\mathcal{F}_g^n := \emptyset$;
- 9: **foreach** $f \in \mathcal{F}_g$ **do**
- 10: **begin**
- 11: **if** $\text{rand}(0, 1) \leq cr$ **then**
- 12: ランダムに選んだ2つの個体を交叉; /* 生成された個体を f' , f'' で表す */
- 13: f' , f'' に対し mr の率で突然変異を行う;
- 14: f' , f'' に対し山登り法を適用する;
- 15: $\mathcal{F}_g^n := \mathcal{F}_g^n \cup \{f', f''\}$;
- 16: **end**;
- 17: $\mathcal{F}_{g+1} := (\mathcal{F}_g; \cup \mathcal{F}_g^n)$ のベスト ps ;
- 18: **end**;
- 19: **for** $g = 1$ **to** mg **do**
- 20: \mathcal{F}_g を出力;

4 Lim 等のアルゴリズムの追試

我々が行った Lim 等の GA+HC アルゴリズムの追試の実験結果について述べる。我々は文献 [11] に従って GA+HC アルゴリズムを実装した。この実装されたプログラムを *BaseTYPE* と呼ぶことにする。文献 [11] で扱われて、かつ一般的に入手可能なテストデータ (行列) に対し追試を行った。結果的には、入手可能なテストデータが bp_1000 と呼ばれる行列だけであったため、我々はこの行列のみに対し追試を行った。行列 bp_1000 は "Harwell-Boeing Sparse Matrix Collection (HBSMC)" のテストセット SMTAPE 内にある非零成分数 4661, サイズ 822×822 の疎行列である。HBSMC は、様々な分野で使われている多種多様の疎行列をテストデータとして集めたものである。

図 4 は文献 [11] からの引用で、Lim 等が GA+HC アルゴリズムを実装したプログラムを bp_1000 に適用した際の世代数とバンド幅の関係を示している。一方図 5 は、我々が実装したプログラム *BaseTYPE* を bp_1000 に 50 回適用し、それにより得られたバンド幅の出現頻度を示している。60 世代までの実験を 50 回繰り返したが、図 4 のような良い結果は一度も得られなかった。

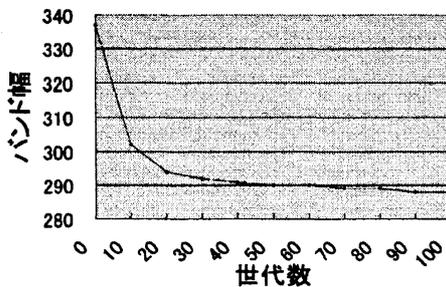


図 4: 世代数とバンド幅の関係

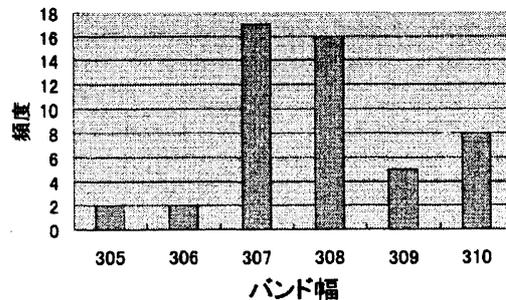


図 5: バンド幅の出現頻度に対する度数分布

5 Lim 等のアルゴリズムの変更

Lim 等の GA+HC アルゴリズムの遺伝的アルゴリズム (GA) に対し、幾つかの点について変更を行う。

5.1 染色体の定義の変更 (*OrderTYPE*)

Lim 等の GA+HC アルゴリズムの遺伝的アルゴリズム (GA) では染色体が "配置位置の列" で定義されていたが、これを "頂点の列" で定義する。すなわち、配置 f に対する染色体を $f^{-1}(1), \dots, f^{-1}(n)$ で定義する。*BaseTYPE* の染色体の定義を "頂点の列" f^{-1} に変更したプログラムを *OrderTYPE* と呼ぶ。

5.2 交叉点の変更 (*XPT*)

我々が新しく提案する交叉方法は Lim 等の方法と同じく 1 点交叉法である。Lim 等の方法では交叉点が中間点に固定されていたが我々の方法では動的となる。 $xnum$ と $minxn$ を以下のように定義する。

$$xnum_P(i) = |(f_f[1, i] \cup f_m[1, i]) \setminus (f_f[1, i] \cap f_m[1, i])|$$

$$minxn(P) = \min_{\frac{n}{4} \leq i \leq \frac{3n}{4}} xnum_P(i)$$

このとき、 $\frac{n}{4} \leq p \leq \frac{3n}{4}$ かつ $xnum_P(p) = \min_{x \in P} x(p)$ かつなるべく中央の $\frac{n}{2}$ に近い p を選ぶ。そのような p は高々2つしか存在しないがそのうちの小さい方を $xpt(P)$ で表す。我々の方法では交叉を行う染色体のペア(両親) $P = (f_f, f_m)$ に対し、 $xpt(P)$ を計算し、 $xpt(P)$ をペア P に対する交叉点(次節で述べる SDP では交叉点の候補)として採用する。このように交叉点を動的にする事で、形質遺伝性(親の形質の継承性)の向上が期待できる。OrderTYPE の中間点交叉法を” xpt による動的な交叉法”に変更したプログラムを XPT と呼ぶ。

5.3 個体の向きの変更 (SDP)

ペア $P = (f_f, f_m)$ に対し、 $\min_{x \in P} x(p) = \min_{\tilde{P} \in \{f_f, f_f^R\} \times \{f_m, f_m^R\}} \min_{x \in \tilde{P}} x(p)$, $f'_f \in \{f_f, f_f^R\}$, $f'_m \in \{f_m, f_m^R\}$ を満たすペア $P' = (f'_f, f'_m)$ を、 P に対する最適な向きの組合せを持つペアと呼び、 $SDP(P)$ で表す。 XPT を以下のように拡張する: 父母の染色体としてペア $P = (f_f, f_m)$ が選ばれた場合、 $SDP(P)$ を計算し、 $SDP(P)$ の向きに従って交叉を行う。このように拡張されたプログラムを $XPT+SDP$ と呼ぶ。

6 実験結果

文献 [11] で扱われているテストデータの中では bp_1000 が唯一の一般入手可能なテストデータであるため我々はこの行列に対しのみ比較実験を行った。実験環境は、CPU, intel pentium 4 / 2.2GHz., メモリ, 1GB., OS, Windows 2000, コンパイラ, Java2SDK 1.4.2 である。 $XPT+SDP$ はプログラム $BaseTYPE$ に、(1) 染色体の変更, (2) 交叉点動的化, (3) 染色体の反転操作, の3つの変更を加えることにより得られるプログラムである。図6は $BaseTYPE$ と $XPT+SDP$ の性能差を表したもので、bp_1000 に50回適用した際のバンド幅とその出現頻度が度数分布により示されている。また、表1に、各プログラムを50回実施したときの、出力値の平均、分散、最良値、最悪値を示す。図6により、上記3つの変更により解の精度の向上が確認できる。

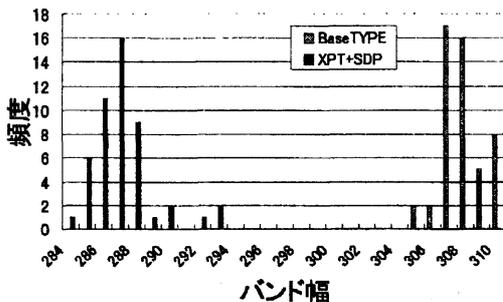


表 1: 各プログラムの性能差

	Ave	Dev	Best	Worst
BaseTYPE	307.88	1.259	305	310
OrderTYPE	289.66	2.5109	285	297
XPT	289.86	1.0772	288	292
XPT+SDP	287.38	1.86	284	293

図 6: BaseTYPE と XPT+SDP の性能差

参考文献

[1] A.Blum, G.Konjevod, R.Ravi and S.Vempala, Semi-Definite Relaxations for Minimum Bandwidth and other Vertex-Ordering problems, Proc. of 30th STOC, (1998), 100-105. (Journal version) Theor. Comput. Sci. 235(1), (2000), 25-42.

[2] H.L.Bodlaender, The complexity of finding uniform emulations on paths and ring networks, Information and Computation 86 (1990) 87-106.

- [3] G.Blache and M.Karpinski, On Approximation Intractability of the Bandwidth Problem, ECCC TR98-014 (1998).
- [4] H.L.Bodlaender, M.R.Fellows, and M.T.Hallett, Beyond NP-completeness for problems of bounded width (extended abstract): hardness for the W hierarchy, Proc. of 26th STOC, (1994), 449-458.
- [5] P.Z.Chinn, J.Chvatalova, A.K.Dewdney, and N.E.Gibbs, The bandwidth problem for graphs and matrices—a survey, J. Graph Theory 6 (1982), 223-254.
- [6] Cuthill, E., McKee, J.: Reducing the bandwidth of sparse symmetric matrices. In: Proc. ACM Nat.Conf. New York (1969), 157-172.
- [7] A.Esposito, M.S.F.Catalano, F.Malucelli, and L.Tarricone, A new matrix bandwidth reduction algorithm, Operations Research Letters 23, (1998), 99-107.
- [8] U.Feige, Approximating the bandwidth via volume respecting embeddings, (Extended abstract) STOC (1998) 90-99: (Journal version) Journal of Computer and System Sciences 60(3), (2000), 510-539.
- [9] N. E. Gibbs, W. G. Poole, Jr., and P. K. Stockmeyer, An algorithm for reducing the bandwidth and profile of a sparse matrix, SIAM J. Numer. Anal. (1976), 13:236-250.
- [10] M.Garey, R.Graham, D.Johnson, and D.Knuth, Complexity Results For Bandwidth Minimization, SIAM J. Appl. Math. 34 (1978), 477-495.
- [11] A.Lim, B.Rodrigues, and F.Xiao, A Genetic Algorithm with Hill Climbing for the bandwidth minimization, preprint, 2003.
- [12] A.Lim, B.Rodrigues, and F.Xiao, Integrated Genetic Algorithm with Hill Climbing for Bandwidth Minimization Problem”, Genetic and Evolutionary Computation Conference 2003, (Chicago, USA).
- [13] Y. Lai and K. Williams, A survey of solved problems and applications on bandwidth, edgsum, and profile of graphs, J. Graph Theory 31 (1999), 75-94.
- [14] R.Martí, M.Laguna, F.Glover, and V.Campos, Reducing the Bandwidth of a Sparse Matrix with Tabu Search, European Journal of Operational Research 135(2), (2001), 211-220.
- [15] E.Piñana, I.Plana, V.Camposy, and R.Martí, European Journal of Operational Research 153, (2004), 200-210.
- [16] J.S.Turner, On the Probable Performance of Heuristics for Bandwidth Minimization, SIAM J. Comput. 15(2) (1986), 561-580.
- [17] W.Unger, The Complexity of the Approximation of the Bandwidth Problem, Proc. 39th FOCS (1998), 82-91.
- [18] K.Yamazaki, On Approximation Intractability of the Path-Distance-Width Problem, Discrete Applied Mathematics Vol.110 (2001), 317-325.