離散最適化に対する高速な厳密アルゴリズム

(Fast exact algorithms for discrete optimization)

岡本 吉央 (Yoshio Okamoto)

豊橋技術科学大学 情報工学系

E-mail: okamotoy@ics.tut.ac.jp

平成 17 年 5 月 30 日

1 はじめに

離散的な最適化問題では，「n 個の要素から成る集合の部分集合で，ある性質を満たすものの中から要素数が最大のものを選ぶ」といった形をしたものによく遭遇する．部分集合全てを列挙して，その中から望みのものを探すという手法があるけれども，これでは列挙しなくてはならない対象が多すぎて (この場合は 2^n 個存在することになり)，有効な方法になりにくい．しかし，いわゆる NP 困難な最適化問題に対しては計算量が指数関数のようになってしまうことは避けられないと考えられている．

近年，アルゴリズムの会議などに足を運ぶとそのような問題に対して 1.4422^n とか 1.3302^n といった計算量を持つアルゴリズムを見受けるようになった．このような計算量もいまだ指数関数であるけれども，2^n に比べると「はるかに」小さい．例えば，仮想的に $2^{n/2}$ という計算量 (これは $\sqrt{2}^n$ と同じである) を持つアルゴリズムを考えて，これと 2^n という計算量を持つアルゴリズムを比較してみよう．計算量の比を計算すると $2^n/2^{n/2} = \sqrt{2}^n$ となり，アルゴリズムの計算量が指数関数的に改善されたことになる．これは，n^3 という計算量を持つアルゴリズムを n という計算量を持つアルゴリズムに改善しても，$n^3/n = n^2$ というだけの改善にしかならないので，この力は絶大である．

では，アルゴリズム設計者としての興味は，1.4422^n というような計算量を持つアルゴリズムをどのようにして組み立てればよいのか，ということである．この分野は大きいけれども，まだまだ手法が確立しておらず，また大きな未解決問題がたくさんあるといった理論情報科学の目から見ると大変興味深い分野になっている．また，以降解説する手法の中にはこれまで発見的アルゴリズムなどと呼ばれて計算量解析があまり行われて来なかったものもあり，実用上役に立っていたアルゴリズムが理論的にどれ程の速さで動くのかということを教えてくれることにもなる．

本稿では次のような手法を紹介することにしたい．

1. 部分集合縦断動的計画法 (dynamic programming across the subsets)．
2. 探索木の縮小 (search-tree pruning)．
3. 局所探索 (local search)．
4. 分離集合定理 (separator theorem) の応用．

この分野は「厳密計算 (exact computation)」と呼ばれることもあるけれども，名前はまだそれ程一般的になっていない．これと関連した分野に「パラメータ化計算 (parameterized computation)」あるいは「パラメータ化計算量理論 (parameterized complexity)」といったものがある．こちらも NP 困難な問題に対する厳密アルゴリズムを考察する分野であるけれども，厳密計算の分野とは少し趣きが違う．

厳密計算の分野に限って解説した書物は今のところ見当たらない．解説論文では Woeginger [87] を推薦したい．本稿はこれに沿っている．最近 Woeginger は別の解説論文 [88] を書いている．パラメータ化計算については Downey & Fellows の本 [29] があり[1]，優れた解説論文もいくつかある．中でも，Niedermeier のハビリテーション論文 [64] はよくまとまっているので，これを推薦したい．

以後，$O^*(2^n)$ のような表記をしていくことがある．これは，$O(2^n \mathrm{poly}(n))$ を意味する．すなわち，計算量の中の多項式因子を無視して，指数関数部分にだけ注目をする，ということである．本稿で扱うアルゴリズムの目標は，その計算量に現われる指数関数の底の部分を出来る限り小さくする (1 に近づける) ことにある．

2 部分集合縦断動的計画法

部分集合縦断動的計画法は厳密計算の分野の中でも古典的な手法であり，ほとんど全列挙法である．ただ，列挙する対象を上手に制御することでよりよい計算量を導くことを主眼にしている．

巡回セールスマン問題 巡回セールスマン問題では，1 から n までの都市と，各都市間のコストが与えられたときに，全ての都市を巡るパターン (巡回路) の中からコストの総和が最小のものを見つける．巡回路は 1 から始まるとしてよい．$c(i,j)$ で i と j の間のコストを表すことにする．自明なアルゴリズムでは，$\Theta(n!)$ 個存在する全ての巡回路を列挙することになる．ここでは，部分集合縦断動的計画法によって，この問題を $O^*(2^n)$ で解く方法を与える．

部分集合 $S \subseteq \{2,\ldots,n\}$ と都市 $i \in S$ に対して，$F[S,i]$ で次の条件を満たすパスの中でコスト和が最小なもののそのコストを表すことにする．

- そのパスは 1 から始まる．
- そのパスは i で終わる．
- そのパスは 1 から i にたどり着くまでに，$S \setminus \{i\}$ の都市全てを一度ずつ通る．

このとき，巡回セールスマン問題の最適値は

$$\min(F[\{2,\ldots,n\},i] + d(i,1) \mid i \in \{2,\ldots,n\})$$

である．よって，全ての可能な S と i に対して $F[S,i]$ の値が計算できれば十分である．そのための再帰式は次のものになる．

$$F[S,i] = \min(F[S \setminus \{i\}, j] + d(j,i) \mid j \in S \setminus \{i\}).$$

この式の左辺の S に比べて，右辺の $S \setminus \{i\}$ は要素数が小さいので，$F[S,i]$ の値を S の要素数が小さい方から順に計算していくことで (すなわち，動的計画法によって) ，最終的に全ての $F[S,i]$ が計算できる．計算するべき値の数は $O(2^n n)$ であり，上の再帰式により，それぞれの値は $O(n)$ 個の他の値を見ることで計算できる．よって，計算量は $O(2^n n^2) = O^*(2^n)$ になる．

グラフの染色数 無向グラフ $G = (V,E)$ が与えられたときに，その各頂点に色を塗って，隣り合う頂点どうしは違う色で塗られるようにしたい．そのような彩色を真の彩色 (proper coloring) と呼ぶ．真の彩色を行なうことの出来る最小の色数のことをグラフの染色数 (chromatic number) と呼ぶ．真の彩色において，同じ色を持つ頂点の集合は独立集合 (independent set，安定集合とも呼ばれる) になっている．自明なア

[1] 著者の Mike Fellows によると，この本は間違いが多いのであまり推薦できないということである．

プローチは頂点集合の分割を全て列挙するものである．要素数 n の集合の分割の数は Bell 数と呼ばれていて，漸近的には $2^{\Theta(n \log n)}$ である．

今から紹介するアルゴリズムで重要な役割を果たしているものが Moon & Moser [63] による次の結果である．

「頂点数が n のグラフには極大独立集合が高々$3^{n/3}$ 個しかない．」

色数最小の彩色においては，その中の同一色頂点集合が極大独立集合になっているようなものが存在していて，グラフの極大独立集合を 1 つあたり多項式時間で出力する列挙アルゴリズムがいくつか知られているので，これを利用するわけである．

頂点部分集合 $S \subseteq V$ に対して，$F[S]$ で S が誘導する G の部分グラフの染色数を表すことにする．S が空のときは $F[S] = 0$ である．S が空でないときは，

$$F[S] = 1 + \min(F[S \setminus T] \mid T \text{ は } G[S] \text{ の極大独立集合})$$

が成り立つ．ただし，$G[S]$ は S の誘導する G の部分グラフである．今，動的計画法を S の要素数の小さい方から順に行なうと，最終的に $F[V]$ を計算することが出来て，これが G の染色数である．

各 S に対して，$G[S]$ の極大安定集合を列挙する必要があるけれども，これは S の要素数が s のとき，$\mathrm{O}^*(3^{s/3})$ で出来る．よって，各 $F[S]$ は $\mathrm{O}^*(3^{s/3})$ で計算できる．動的計画法では可能な全ての S を見ていくので，計算量は総計 $\sum_{s=0}^{n} \binom{n}{s} \mathrm{O}^*(3^{s/3}) = \mathrm{O}^*((1+3^{1/3})^n) \approx \mathrm{O}^*(2.4422^n)$ である．

補足　上の巡回セールスマン問題に対するアルゴリズムは Held & Karp [48] と Bellman [9] が 1962 年の論文で独立に発見した．巡回セールスマン問題に対して $\mathrm{O}^*(2^n)$ よりも速いアルゴリズムは見つかっておらず，これは厳密計算の分野における大きな未解決問題である．Deĭneko, Hoffmann, Okamoto & Woeginger [26] はこの方法を 2 次元ユークリッド平面上の巡回セールスマン問題に適用している．3 正則グラフにおける巡回セールスマン問題に対して Eppstein [33] は $\mathrm{O}^*(1.260^n)$ のアルゴリズムを与えている．これは後で登場する「探索木の縮小」という手法による．

2 つ目に見た染色数に対するアルゴリズムは Lawler [58] による．Eppstein [32] はまず頂点数 n のグラフにおいて要素数が k 以下の極大安定集合の数が高々$3^{4k-n}4^{n-3k}$ になることを証明した．これを利用することで，彼は $\mathrm{O}^*(2.4150^n)$ で染色数を計算するアルゴリズムを与えている．

Lawler は同じ論文 [58] でグラフの 3 彩色可能性を判定するアルゴリズムをまず与えていて，これは上のアルゴリズムよりも単純である．全ての極大独立集合 S に対して $G - S$ が 2 彩色可能であるかどうかを確かめるのである．計算量は $\mathrm{O}^*(3^{n/3}) \approx \mathrm{O}^*(1.4422^n)$ になる．グラフの 3 彩色可能性の判定はその後も研究があり，現在最速のアルゴリズムの計算量は $\mathrm{O}^*(1.3289^n)$ であり，Eppstein [31] による．(ジャーナル版は Beigel & Eppstein [8] である．) これは全く異なるアイディアによる．

Byskov [15] はグラフの 4 彩色可能性，5 彩色可能性，6 彩色可能性を考えて，それぞれ $\mathrm{O}^*(1.7504^n)$，$\mathrm{O}^*(2.1592^n)$，$\mathrm{O}^*(2.3289^n)$ のアルゴリズムを与えている．また，これらのアルゴリズムを用いることで，染色数を $\mathrm{O}^*(2.4023^n)$ 時間で計算することを可能にした．これはこれら 4 つの問題に対する現在最速のアルゴリズムであり，アイディアは Eppstein [32] の流れに従っている．

部分集合縦断動的計画法は部分集合を全部見ていくものなので，素朴な方法では指数関数よりもたくさんの時間がかかってしまう問題に有効だと思われる．例えば，巡回セールスマン問題のように素朴な方法では $n!$ の順列全てを考えるような問題やグラフの染色数計算のように素朴な方法では Bell 数だけの分割全てを考えるような問題である．

Held & Karp [48] では 1 機械スケジューリングに関わる問題も考察されている．これは Woeginger の解説論文 [87] でも紹介されている．1 機械スケジューリングに関わる問題では，n 個のジョブを機械に割り当てる方法が $n!$ だけあるので，そこに部分集合縦断動的計画法を適用するのである．Fomin, Todinca & Kratsch [40] はグラフのパス幅 (pathwidth)，プロファイル (profile) も部分集合縦断動的計画法を適用することで $\mathrm{O}^*(2^n)$ 時間で計算できることを指摘している．

グラフの木幅 (treewidth) の計算とグラフの最小フィルイン問題に対して，Fomin, Todinca & Kratsch [40] は独自の極値グラフ理論的な結果と Bouchitté & Todinca [13] による再帰式を組み合わせることで，部分集合縦断動的計画法による $\mathrm{O}^*(1.9601^n)$ 時間アルゴリズムを開発している．

最小集合被覆問題 (minimum set cover) とは，集合 X とその部分集合族 $\mathcal{F}$ が与えられたときに，$\mathcal{F}$ の部分族で，その合併が X になるものの中で大きさが最小のものを見つけるものである．$n = |X|$，$m = |\mathcal{F}|$ とする．自明なアルゴリズムでは $\mathrm{O}^*(2^m)$ の計算量が必要になる．Fomin, Kratsch & Woeginger [38] は部分集合縦断動的計画法によって $\mathrm{O}^*(2^n)$ で最小集合被覆問題を解くアルゴリズムを与えている．

グラフ $G = (V, E)$ と頂点部分集合 $K \subseteq V$ が与えられたとき，G の K に対する Steiner 木 (Steiner tree) とは，G の部分木で K の頂点を全て含むもののことである．各辺に非負の重みが与えられている

ときに，辺の重み和が最小になる Steiner 木を求める問題は最小 Steiner 木問題と呼ばれる．この問題に対して Dreyfus & Wagner [30] は $O(3^k n + 2^k n^2 + n^2 \log n + nm)$ 時間のアルゴリズムを考案した．ここで，$k = |K|$ である．これは K の部分集合を縦断する動的計画法に基づいている．(3^k という部分は K の 3 分割全体の集合の個数に対応している．)

3 探索木の縮小

問題解決に対する一般的な方法に「場合分け」がある．それを組織立って行なうと「木の探索」になる．ここでは，分枝規則 (場合分けの方法) の工夫によって，探索木が小さくなるようにすることを考える．そこから，高速なアルゴリズムが得られるのである．

最大独立集合　与えられた無向グラフ G の中から要素数が最大の独立集合 (最大独立集合) を見つけたい．最大独立集合は極大独立集合でもあるので，前節で見た Moon & Moser [63] を用いると，$O^*(1.4422^n)$ でこの問題が解けると分かる．(ここで，n は頂点数である．) ここでは探索木の縮小によって，$O^*(1.3803^n)$ で最大独立集合を見つけてみたい．

グラフ G の頂点の次数が全て 2 以下である場合を考えてみる．すると，G の各連結成分はパスかサイクルである．よって，この場合は簡単に (高々多項式時間で) 最大独立集合を発見できる．したがって，G に次数が 3 以上の頂点が存在すると仮定してよい．この頂点を v とする．グラフ G の最大独立集合を 1 つ固定して I とする．いま，I は v を含むか含まないかのいずれかである．I が v を含むとき，I は v の近傍を含むことが出来ない．よって，I のその他の頂点は G から v と v の近傍を取り除いたグラフ G_1 の中にあることになり，このグラフの頂点数は高々 $n-4$ である．(というのは v の次数が 3 以上と仮定しているからである．) I が v を含まないとき，I の頂点は G から v を取り除いたグラフ G_2 の中にあり，このグラフの頂点数は $n-1$ である．この両方の場合を考えて，ここで枝分れ (場合分け) を起こすことになる．すなわち，$\alpha(G)$ でグラフ G の最大独立集合の大きさを表すとすると，

$$\alpha(G) = \max(1 + \alpha(G_1), \alpha(G_2))$$

が成り立つ．右辺の G_1，G_2 に対して，今の手続きを再帰的に繰り返すことで全体の最大独立集合を見つけるのである．

では，$T(n)$ でこのアルゴリズムが頂点数 n のグラフの最大安定集合を求めるために必要な時間の最大値を表すことにする．グラフに次数 3 以上の頂点が存在しないとき，$T(n)$ は n の多項式である．そうでないときは，上記のアルゴリズムに従って 2 つの場合に分かれる．よって，

$$T(n) \leq T(n-4) + T(n-1) + \mathrm{poly}(n).$$

となる．今，多項式因子は無視していて，計算量の上界に興味があるので，上の式を

$$T(n) = T(n-4) + T(n-1)$$

と書き換えよう．ここで，$T(n) = \lambda^n$ と置くと，

$$\lambda^n = \lambda^{n-4} + \lambda^{n-1}$$

が得られて，この方程式の実数解で正のものは $\lambda \approx 1.3803$ である．したがって，アルゴリズムの計算量として $O^*(1.3803^n)$ が得られるのである．

充足可能性問題　与えられた論理積標準形 (conjunctive normal form, CNF) の論理式に対して，それを真とするような論理変数への値の割当が存在するかどうかを判定する問題が充足可能性問題 (satisfiability problem) である．以後，充足可能性問題を SAT と呼んでいく．与えられた CNF の各節 (clause) に含まれるリテラルの数が高々k であるように制限した SAT を kSAT と呼ぶ．この問題は $k \geq 3$ に対して NP 完全である．変数の数が n であるとき，全ての可能な変数割当 2^n 個を総調べする自明なアルゴリズムを考えると，この計算量は $\mathrm{O}^*(2^n)$ になる．

では，3SAT に対する $\mathrm{O}^*(1.8393^n)$ のアルゴリズムを探索木の縮小に基づいて設計しよう．問題で与えられるものは n 個の論理変数を持つ 3CNF 式 F である．この論理式 F の中に大きさ 3 の節がないときには，線形時間で充足可能性が判定できる．そうでないときには，大きさ 3 の節 $C = \ell_1 \vee \ell_2 \vee \ell_3$ を固定する．ここで，ℓ_1, ℓ_2, ℓ_3 はリテラルである．論理式 F が充足可能であるとき，C 自体も充足可能である．節 C が充足可能であるための必要十分条件は次の 3 つのいずれかを成り立たせる変数割当が存在することである．

1. リテラル ℓ_1 が真になる．
2. リテラル ℓ_1 が偽になり，リテラル ℓ_2 が真になる．
3. リテラル ℓ_1, ℓ_2 が偽になり，リテラル ℓ_3 が真になる．

この 3 つの場合に対応させて分枝をする．第 1 の場合では，ℓ_1 を真に固定することで，変数の数が $n-1$ の論理式が得られる．同様に，第 2，第 3 の場合では，変数の数がそれぞれ $n-2$，$n-3$ の論理式が得られる．そして，各場合で再帰的に問題を解くのである．いずれかの場合で論理式が充足可能になれば，元の論理式も充足可能であり，そうでなければ，元の論理式が充足可能でないことが分かる．

では，計算量の解析をしよう．$T(n)$ で変数の数が n の 3CNF 式に対して上のアルゴリズムが要する時間の最大値を表すことにする．そのとき，

$$T(n) \leq T(n-1) + T(n-2) + T(n-3) + \mathrm{poly}(n)$$

が得られる．先と同様に，$\lambda^n = \lambda^{n-1} + \lambda^{n-2} + \lambda^{n-3}$ の解の中で正のものを計算するとおよそ 1.8393 となり，計算量として $\mathrm{O}^*(1.8393^n)$ が得られるのである．

補足　最大独立集合問題に対する厳密アルゴリズムを扱った最初の論文は Tarjan & Trojanowki [84] であり，計算量は $\mathrm{O}^*(1.2599^n)$ であった．アイディアは上で述べたものと同じであるけれども，次数の高い頂点の近傍に関してより詳細な場合分けを施している．Robson [73] は $\mathrm{O}^*(1.2108^n)$ でこの問題を解いている．Robson [74] は $\mathrm{O}^*(1.1888^n)$ のアルゴリズムを作成したと報告している．最大次数が 3 のグラフの最大独立集合を見つける問題に対して，Chen, Kanj & Xia [17] は $\mathrm{O}^*(1.1254^n)$ のアルゴリズムを与えた．

充足可能性問題に対して探索木の縮小のアプローチを最初に用いたのは Monien & Speckenmeyer の論文 [62] である．彼らは，上で紹介したアプローチをより精緻にして，$\mathrm{O}^*(1.6181^n)$ のアルゴリズムを与えた．一般の kSAT に対しては，$\mathrm{O}^*(\beta^n)$ となり，ここで β は $\beta = 2 - 1/\beta^{k-1}$ を満たす最大正実数である．Kullmann [57] はこの手法を押し進めて 3SAT に対する $\mathrm{O}^*(1.4963^n)$ のアルゴリズムを開発した．3SAT の (決定性) アルゴリズムの世界記録は Brueggemann & Kern [14] による $\mathrm{O}^*(1.473^n)$ である．これは次に紹介する Dantsin, Goerdt, Hirsch, Kannan, Kleinberg, Papadimitriou, Raghavan & Schöning [22] の局所探索の枠組と探索木の縮小の組合せに賢い改善を施して得られている．

探索木の縮小は高速厳密アルゴリズムの設計において最も一般的な方法のようであり，適用された問題も多い．先に述べたように，探索木の縮小に基づいて，Eppstein [33] は 3 正則グラフにおける巡回セールスマン問題に対する $\mathrm{O}^*(1.260^n)$ のアルゴリズムを与えている．

最小支配集合問題 (minimum dominating set problem) に対して Fomin, Kratsch & Woeginger [38] は $\mathrm{O}^*(1.93782^n)$ のアルゴリズムを与えている．これは Reed [71] による極値グラフ理論的な結果に探索木の縮小の手法を組み合せたものである．Grandoni [47] は $\mathrm{O}^*(1.8021^n)$ のアルゴリズムを与えている．そのために，彼は集合被覆問題に対する $\mathrm{O}^*(1.3424^{n+m})$ のアルゴリズムを設計した．これは探索木の縮小と動的計画法による記憶化に基づいている．Fomin, Grandoni & Kratsch [37] は測量統治法 (measure and conquer) という技法を導入して，Grandoni による上記のアルゴリズムの計算量解析を $\mathrm{O}^*(1.2303^{n+m})$ に改善した．その帰結として，最小支配集合問題が $\mathrm{O}^*(1.5137^{n+m})$ 時間で解けることが分かる．

最大充足可能性問題 (MaxSAT) に対しても多くの研究があり，そのほとんどは探索木の縮小によっている．Max2SAT に対して，Gramm, Guo, Niedermeier & Rossmanith [46] は $\mathrm{O}^*(2^{m/5}) \approx \mathrm{O}^*(1.1487^m)$

のアルゴリズムを与えた．(ただし，m は 2-節の数である．) 一般の MaxSAT に対しては，Chen & Kanj [16] が $O^*(1.3414^m)$ のアルゴリズムを与えた．(ここで，m は節の数である．)

最大充足可能性問題を一般化したものに，最大制約充足問題 (MaxCSP) というものがある．Scott & Sorkin [82] は Max2CSP に対する $O^*(2^{m/5}) \approx O^*(1.1487^m)$ を与えた．Max2CSP は Max2SAT，最大カット問題，最大有向カット問題を特殊な場合として含んでいるので，これらの問題に対しても同じ上界が得られる[2]．彼らの論文にある小さな間違いを Razen [70] は訂正している．

Feige & Kilian はグラフのバンド幅問題 (bandwidth problem) に対して $O^*(10^n)$ という計算量のアルゴリズムを開発した．しかし，未出版であり，詳細は不明である．Feige [36] はアルゴリズムの概要を紹介している．Woeginger [87] は彼らのアイディアによってどのように $O^*(20^n)$ のアルゴリズムが得られるのか解説している．

探索木に基づく手法では，効果的な分枝規則を見つけることが重要であるけれども，それは容易ではない．そこで，分枝規則，単純化規則の発見を自動化し，また，計算量の算出も自動化する試みが行なわれている [34, 45, 35, 56]．

4 局所探索

局所探索は厳密アルゴリズムの設計手法の中でも新しいものである．適用された問題は今のところ充足可能性問題 (SAT) しか見当たらない．ここでは，3SAT を例にしてそれを簡単に紹介していく．

局所探索 3CNF 論理式 F が与えられて，n をそれに含まれる変数の数とする．n 個の変数に対する真理値の割当は 2^n 通り存在する．今，この 2^n 個の真理値割当を Hamming 立方体と見なすことにする．すなわち，「真」を「1」，「偽」を「0」に対応させて，真理値割当全体の集合を $\{0,1\}^n$ と同一視するのである．例えば，$(0,0,\ldots,0)$ というのは全ての変数を偽に割り当てることに対応する．2 つの割当 $a, a' \in \{0,1\}^n$ が隣接するとは，a と a' の成分の中の 1 つだけが異なり，残り全ては同じになっていることである．この隣接関係は $\{0,1\}^n$ を頂点集合とするグラフを定義し，このグラフを Hamming 立方体ということにする．グラフに対する距離の定義を思い出すと，2 つの割当 $a, a' \in \{0,1\}^n$ が距離 d だけ離れているとは，a と a' の成分の中の d 個だけが異なり，残り全ては同じになっていることになる．

変数割当 $a \in \{0,1\}^n$ と自然数 d に対して，$H(a,d)$ で，a から距離 d 以内にある割当全体の集合を表すことにする．$H(a,d)$ の大きさは $\sum_{i=0}^{d} \binom{n}{i}$ である．探索木を用いることで，$H(a,d)$ に F を充足する割当が存在するかどうかを $O^*(3^d)$ 時間で判定できる．(一般の kSAT に対しては $O^*(k^d)$ になる．) 具体的には次のようになる．初期割当 a が与えられている．この割当 a が既に F を充足していたら，そこで終わりである．そうでなかったら，F には充足されていない節 C が存在する．その C の 3 つのリテラルそれぞれに対して，そのリテラルの割当を変更する．変更した割当をそれぞれ a_1, a_2, a_3 として，$H(a_1, d-1), H(a_2, d-1), H(a_3, d-1)$ を再帰的に探索するのである．

よって，任意に a を選び，$d = n$ とすることで，$O^*(3^n)$ のアルゴリズムが得られる．これは自明な $O^*(2^n)$ のアルゴリズムよりも遅い．

2 点から始める局所探索 では，この局所探索を $0^n := (0,0,\ldots,0)$ と $1^n = (1,1,\ldots,1)$ という 2 点から始めよう．任意の割当は 0^n または 1^n から $n/2$ 以内の距離にある．よって，$H(0^n, n/2)$ と $H(1^n, n/2)$ を探索することによって，充足可能性が判定できる．計算量は $2 \times O^*(3^{n/2}) = O^*(1.7321^n)$ であり，これは $O^*(3^n)$ に比べて遥かによく，また，探索木の縮小のところで与えた $O^*(1.8393^n)$ よりもよい．

多点から始める局所探索 先程は 2 点から局所探索を始めたけれども，もう少し点の数を増やしてみるとどうなるものだろうか．ここで問題になるのは探索を開始する点をどのように選択するかということである．よいアイディアが思いつかないときに行なう方法は「ランダムに選ぶ」ということである．すなわち，ランダムに何個か選択して，そこからある d に対して局所探索をするのである．

[2] この結果は少し注意深く見ないといけない．というのは，このアルゴリズムが自明な $O^*(2^n)$ 時間アルゴリズムよりも速くなるのは $m < 5n$ のときに限るからである．$O^*(c^n)$ という形 $(c < 2)$ のアルゴリズムを Williams [86] が最近提案している．これは「分割法」という違うアイディアによる．詳しくは後の節を参照のこと．

今，変数割当 a をランダムに選んで来て，$d = n/4$ とする．$H(a,d)$ の要素数は $\sum_{i=0}^{d} \binom{n}{i}$ であるけれども，$d = n/4$ のとき，これは $\Theta^*((4/3^{3/4})^n)$ になる[3]．よって，F が充足可能であるとき，$H(a,d)$ の中に F を充足する割当が存在する確率は少なくとも $|H(a,d)|/2^n = \Theta^*((2/3^{3/4})^n)$ である．よって，このランダム選択を $\mathrm{O}^*((3^{3/4}/2)^n)$ 回繰り返すことで，充足割当を発見する回数の期待値が 1 以上になる．1 つのランダム割当に対して，局所探索は $\mathrm{O}^*(3^{n/4})$ 時間かかるので全体の計算量は $\mathrm{O}^*((3^{3/4}/2)^n 3^{n/4}) = \mathrm{O}^*(1.5^n)$ になる．確率的アルゴリズムの解析によくあるように，繰返しを定数 c 回するだけで，失敗確率を $\mathrm{O}^*(\mathrm{e}^{-c})$ 以下にすることが出来る．まとめると，ランダムに初期割当を選ぶことで，失敗確率が非常に小さい $\mathrm{O}^*(1.5^n)$ 時間アルゴリズムが得られた．

多点から決定的に始める局所探索　世の中にはいくら失敗確率が地球滅亡の確率よりも小さいからといって，ちょっとでも失敗する可能性のあるものを好まない人々がいる．そういう人々のために，初期割当をランダムに選ばずになんとかする方法を与えたい．そのためには符号理論の結果を利用する．すなわち，いくつかの初期割当を上手に選択して，小さい d によって Hamming 立方体全体を覆うようにしたいのである．そのようなものは被覆符号 (covering code) という名のもとで研究されている対象であり，そこでの結果を用いると，kSAT に対して $\mathrm{O}^*((2 - \frac{2}{k+1})^n)$ のアルゴリズムが得られる．3SAT に対しては $\mathrm{O}^*(1.5^n)$ となり，先程の確率的アルゴリズムと計算量は変わらないが，こちらは失敗することがない．3SAT に対して，$H(a,d)$ の探索と被覆符号の構成法を改善することによって $\mathrm{O}^*(1.473^n)$ のアルゴリズムを得ることが出来る．詳細はこちらも省略する．

局所探索を確率的にする　世の中にはギャンブルのように確率的なものがとても好きな人々もいる．そういう人々のために，何もかもランダムにするように考えてみよう．先の確率的アルゴリズムでは，初期割当だけをランダムに選んで来た．では，もっとランダム性を取り入れるために，$H(a,d)$ の中の探索もランダムに行なうことにしよう．これは $H(a,d)$ の中でランダム・ウォークを行なうことに対応する．

具体的に，アルゴリズムは次のようになる．まず，初期割当 a をランダムに選ぶ．この割当 a が既に F を充足していたら，そこで終わりである．そうでなかったら，F には充足されていない節 C が存在するので，C から 1 つリテラルをランダムに選んで，そのリテラルの割当を変更する．これをある回数繰り返すのである．ここではその回数として $3n$ を選ぶ．そして，この手続き全体を $\mathrm{O}((4/3)^n)$ 回繰り返す．(すなわち，$\mathrm{O}((4/3)^n)$ 個の初期割当をランダムに選んで，このランダム・ウォークを行なう．) アルゴリズムの正当性の証明はマルコフ連鎖によるけれども，ここではそれを全く解説しないことにする．計算量が $\mathrm{O}^*((4/3)^n) = \mathrm{O}^*(1.3334^n)$ になることは簡単に分かる．これは先程の決定性アルゴリズムの計算量 $\mathrm{O}^*(1.4802^n)$ よりもよい．

> 補足　ランダム・ウォークを用いた 3SAT のアルゴリズムは Schöning [77] による．これは Papadimitriou [66] の 2SAT に対するランダム・ウォークのアルゴリズムを基にしている．その脱確率化 (derandomization) として考案されたものが被覆符号を用いた局所探索アルゴリズムで，これは Dantsin, Goerdt, Hirsch, Kannan, Kleinberg, Papadimitriou, Raghavan & Schöning [22] による．計算量は kSAT に対して $\mathrm{O}^*((2 - \frac{2}{k+1})^n)$ である．彼らは 3SAT に対して 3SAT に対して，$H(a,d)$ の探索と被覆符号の構成法を改善することによって $\mathrm{O}^*(1.4802^n)$ のアルゴリズムを得た．別の探索法により，Yamamoto [89] は同じ計算量のアルゴリズムを得ている．3SAT に対する現在最速の決定性アルゴリズムの計算量は $\mathrm{O}^*(1.473^n)$ であり，これは Brueggemann & Kern [14] による．
>
> Paturi, Pudlák & Zane [68] は違うタイプの確率的アルゴリズムを考えた．これは長さ 1 の節を順次充足していくことで全体の充足を図るけれども，長さ 1 の節が見つからなかった場合に，ランダムに選んだ変数の真理値をランダムに割り当てる，という方法である．この方法によって，彼らは kSAT に対する確率的な $\mathrm{O}^*(2^{(1-\frac{1}{k})n})$ アルゴリズムを得た．これを基にして，Paturi, Pudlák, Saks & Zane [67] は導出原理 (resolution) によって冗長な節を付け加えることで，より高速なアルゴリズムを得た．例えば，3SAT に対して $\mathrm{O}^*(1.362^n)$，4SAT に対して $\mathrm{O}^*(1.476^n)$，5SAT に対して $\mathrm{O}^*(1.569^n)$，6SAT に対して $\mathrm{O}^*(1.637^n)$ といった具合である．$k \geq 4$ に対しては Schöning のアルゴリズム [77] よりも速く，$k \geq 5$ に対しては知られている中で最も速い確率的アルゴリズムである．彼らのアルゴリズムは充足割当の数の少

[3] この計算は簡単ではない．情報理論の本を参照する必要があるかもしれない．

ない論理式に対して速く，例えば unique 3SAT に対して $O^*(1.3071^n)$ という計算量を達成する．このアルゴリズムの unique kSAT に対する場合が最近 Rolf [75] によって脱確率化された．

Iwama & Tamaki [51] は Schöning のアルゴリズム [77] が充足割当の数の多い論理式に対して速く，Paturi, Pudlák, Saks & Zane のアルゴリズム [67] が充足割当の数の少ない論理式に対して速いことに着目して，この 2 つを同時に実行するアルゴリズムを考えた．それにより，3SAT に対して $O^*(1.324^n)$，4SAT に対して $O^*(1.474^n)$ という計算量を得た．これは $k = 3, 4$ に対して知られている中で最速の確率的アルゴリズムである．

これらの SAT アルゴリズムのサーベイが Schöning [78] にある．

一般の SAT に対しては，節の長さの上界 k に依存しない上界を与えることが目的になる．そのようなアルゴリズムとして，Pudlák [69] は計算量が $O^*(2^{n-\sqrt{n}/2})$ の確率的アルゴリズムを与えた．その脱確率化を行なったのは Dantsin, Hirsch & Wolpert [23] であり，計算量は $O^*(2^{n-2\sqrt{n/\log_2 n}})$ である．その後，Schuler [80] は $O^*(2^{n(1-1/\log(2m))})$ という計算量の確率的アルゴリズムを与えた．(ただし，m は節の数．) その脱確率化は Dantsin & Wolpert [24] が行ない，同じ計算量を達成している．最近，Dantsin & Wolpert [25] は $O^*(2^{n(1-1/(\ln(m/n)+O(\ln\ln m)))})$ という計算量の確率的アルゴリズムを考案した．現在のところ，これが一般の SAT に対する最速の (確率的) アルゴリズムである．

Kaufmann [55] は Schöning のアルゴリズムを他の問題へ適用することを考えた．例えば，Not-all-equal-kSAT，1-in-kSAT，k-XOR-SAT のような SAT の変種や，ディスクレパンシーの計算，箱詰め問題，集合被覆問題のようなものも考えられている．

SAT の一般化として QBF があるけれども，これは PSPACE 完全問題である．QBF に対しても $O^*(2^n)$ より小さな計算量のアルゴリズムを開発する試みが Williams [85] によってなされている．これは局所探索の手法を用いていない．というのは，QBF には SAT のような局所性がないからである．

5 分離集合定理の応用

Lipton & Tarjan [59] による平面的グラフに対する分離集合定理 (separator theorem) は平面的グラフに関するアルゴリズムの設計で大きな役割を果たしている．特に，分離集合定理を用いることで，平面的グラフにおける最適化問題に対して $O^*(2^{O(\sqrt{n})})$ という形の計算量を持つアルゴリズムを作成することが出来る．指数関数的な計算量というのは $O^*(2^{O(n)})$ のことを言うので，上のような計算量は準指数関数的 (subexponential) な計算量と呼ばれる．

グラフ $G = (V, E)$，0 と 1 の間にあるパラメータ α，そして関数 $n \mapsto f(n)$ が与えられているとき，集合 $C \subseteq V$ が G の $(f(n), \alpha)$-分離集合であるとは，次の 2 条件を満たすことである (ただし，$n = |V|$ とする)．

1. $|C| \leq f(n)$．
2. G の頂点集合は A, B, C という互いに交わらない部分に分解できる．ただし，A, B は次の性質を満たす．
   (a) $|A| \leq \alpha n$，$|B| \leq \alpha n$．
   (b) A のどの頂点も B の頂点と隣接していない．

Lipton & Tarjan [59] は，平面的グラフには必ず $(2\sqrt{2n}, 2/3)$-分離集合が存在して，それを $O(n)$ 時間で見つけることが出来ることを証明した．

平面的グラフにおける最大独立集合　分離集合定理を用いて，平面的グラフの最大独立集合を $O^*(2^{O(\sqrt{n})})$ 時間で求めるアルゴリズムを設計する．

頂点数が少ないとき (ある定数以下のとき) には全列挙によって最大独立集合を求めることにして，頂点数は十分大きいと仮定する．(具体的には 2 以上でよい．) まず，平面的グラフ G の $(2\sqrt{2n}, 2/3)$-分離集合 C を $O(n)$ 時間で見つける．集合 C の大きさは高々 $2\sqrt{2n}$ である．グラフ G から C を取り除くと，分離集合の定義にある A, B が得られる．今，C の部分集合で G において独立であるもの I_C を考える．そして，A の誘導する部分グラフから S の近傍を除去したものを考えて，その中の最大独立集合 I_A を再帰的に求める．集合 B についても同様のことを行なって，I_B を求める．すると，$I = I_A \cup I_B \cup I_C$ は元のグラフ

G の独立集合である．そして，G の独立集合はそのようなものに限られる．今，このような I_C 全てに対して上の手続きを行ない，その中で得られた I の大きさが最大のものを出力すると，それが G の最大独立集合となる．

計算量はどれ程になるだろうか．$T(n)$ で頂点数 n の平面的グラフに対して上のアルゴリズムが要する時間の最大値を表すことにする．上のアルゴリズムから，

$$T(n) \leq \mathrm{O}(n) + 2^{2\sqrt{2n}} \times 2 \times T(2n/3) \times \mathrm{O}(n+m)$$

が成立する．ただし，m はグラフの辺数である．ここから $T(n) = \mathrm{O}^*(2^{\mathrm{O}(\sqrt{n})})$ が得られるのである．(計算の詳細は省略する．)

補足　分離定理自体に関する文献は数多くある．上に挙げた Lipton–Tarjan の定理 [59, 60] の短い証明を Alon, Seymour & Thomas [3] は与えている．$(f(n), 2/3)$-分離集合に関して，Dijdjev & Venkatesan [28] は $f(n) \approx 1.97\sqrt{n}$ のものも存在することを示していて，これは現在知られている中で最もよい値である．α を変えた場合にはまた状況が変わってきて，$\alpha = 1/2$ に対しては $f(n) \approx 4.90\sqrt{n}$ [1, 12]，$\alpha = 3/4$ に対しては $f(n) \approx 1.65\sqrt{n}$ [83] が世界記録である．$\alpha = 3/4$, $f(n) = 2\sqrt{n}$ の場合の証明が Spielman & Teng [83] によって与えられていて，これは幾何的で面白い．(彼らの証明は，Pach & Agarwal の本 [65] にも紹介されていたと思う．) 下界も調べられているが，ここでは省略する．平面的グラフ以外のグラフに関する分離定理も多く知られている [2, 3, 18, 19, 27]．

平面的グラフの最大独立集合を求めるアルゴリズムとしていままで一番高速のものは Fomin & Thilikos [39] によるもので計算量は $\mathrm{O}^*(2^{3.182\sqrt{n}})$ である．これは，上記の分離集合定理に加えて，平面的グラフの木分解 (tree decomposition) を用いている．彼らは，平面的グラフの木幅が高々$3.182\sqrt{n}$ であることを証明し，これを上手に活用している．この木幅に対する上界も今まで知られているものの中で最良のものである．

分離定理の応用は最大独立集合に留まらない．最新の状況については Fomin & Thilikos の論文 [39] を見て頂きたい．

平面的グラフの分離定理の応用として，平面的グラフの上の問題に対する多項式時間近似スキーム (polynomial-time approximation scheme, PTAS) の設計がある．Grigni, Koutsoupias & Papadimitriou [44] は平面的グラフ上の巡回セールスマン問題に対する PTAS を設計した．彼らはそのために自ら適当な分離定理を定式化して，証明した．同様なアプローチが Czumaj, Grigni, Sissokho & Zhao [21] によって最小 2 連結部分グラフ問題に対して取られている．

分離定理は幾何的な最適化問題に対しても考えられていて，例えば，Hwang, Chang & Lee [50] はユークリッド平面上の巡回セールスマン問題を $\mathrm{O}^*(2^{\mathrm{O}(\sqrt{n})})$ 時間で解くアルゴリズムを開発している．

6　その他の手法

ここでは上で紹介しなかった手法で自分が知っているものを簡単に紹介する．

6.1　分割法

入力を定数個の部分に分割して，各部分を上手に組み合わせることで全体の最適解を見つける．

例えば，部分集合和問題 (subset sum probem) を考える．部分集合和問題では，n 個の自然数の (多重) 集合 $N = \{a_1, \ldots, a_n\}$ ともう 1 つ自然数 b が与えられたときに，N の部分集合 S で $\sum_{a \in S} a = b$ となるものが存在するかどうかを問う．部分集合を全て列挙する自明な方法は $\mathrm{O}^*(2^n)$ 時間必要になる．

では，この問題を $\mathrm{O}^*(\sqrt{2}^n) \approx \mathrm{O}^*(1.414^n)$ で解く方法を考えよう．そのために，まず集合 N を 2 つの部分 $L = \{a_1, \ldots, a_{n/2}\}$ と $R = \{a_{n/2+1}, \ldots, a_n\}$ に分割する．(便宜上，n は偶数であると仮定する．そうでないときには，0 を N に付け加えればよい．) 集合 L の部分集合の数は $2^{n/2} = \sqrt{2}^n$ であり，集合 R に対しても同様である．今，L の部分集合を全て列挙して，各集合の和に従って整列 (ソート) する．この整列は $\mathrm{O}(2^{n/2} \log 2^{n/2}) = \mathrm{O}^*(\sqrt{2}^n)$ で行なうことが出来る．ここで，R の各部分集合 R' に対して，L の部分集合 L' で，$L' \cup R'$ の要素の和が b になるようなものを見つけたい．そのような L', R' が存在するとき，そのときに限り，解 S が存在することになる．(それを確認するためには，$S = L' \cup R'$ とすればよい．) 部分

集合 R' を 1 つ固定したときにそのような L' を見つけるためには，先程整列したところで二分探索を行なえばよい．各二分探索は $\mathrm{O}(\log 2^{n/2}) = \mathrm{O}(n)$ 時間で行なうことが出来て，それを R の各部分集合に対して行なうので，総計算時間は $\mathrm{O}(2^{n/2} \times n) = \mathrm{O}^*(\sqrt{2}^n)$ である．このアルゴリズムは Horowitz & Sahni [49] による．

同様の方法で，ナップサック問題を $\mathrm{O}^*(\sqrt{2}^n)$ 時間 (n は与えられるアイテムの数) で解くことが出来るようになる．Schroeppel & Shamir [79] は与えられた集合を 2 つに分割するのではなくて，4 つに分割することで，上のアルゴリズムの空間計算量を改善した．しかし，その改善があった以外に，厳密アルゴリズムに関しては何も進展がない．つまり，1972 年に与えられた $\mathrm{O}^*(\sqrt{2}^n)$ という上界が部分集合和問題とナップサック問題に対する現在最良の時間計算量である．

最近，Williams [86] は Max2CSP に対して (特に，Max2SAT と最大カット問題に対して) 分割法を適用した．彼のアルゴリズムでは，与えられた論理式を 3 つの部分に分割して，各部分の結合を，アルゴリズムのために構成したグラフの上で長さ 3 の閉路を見つける問題に帰着した．これによって，$\mathrm{O}^*(2^{\omega n/3})$ 時間のアルゴリズムが得られる．ここで ω は高速行列乗法 (fast matrix multiplication) の指数部である．Coppersmith & Winograd [20] による現在最速の行列乗法アルゴリズムでは $\omega \approx 2.376$ になるので，アルゴリズムの計算量は $\mathrm{O}^*(1.732^n)$ となる[4]．

6.2 包除原理

包除原理 (inclusion-exclusion principle) は数え上げの基本であるけれども，これによって数え上げ問題に対する厳密アルゴリズムの高速化が可能になることがある．

最も古典的なものは，正方行列のパーマネント (permanent) の計算に対する Ryser の公式だと思う．$n \times n$ 行列 A のパーマネントとは

$$\mathrm{per}(A) = \sum_{\pi:\{1,\ldots,n\}\text{の置換}} \prod_{i=1}^{n} a_{i\pi(i)}$$

で表される量のことである．(行列式との違いは，置換の符号の有無である．) パーマネントの計算は NP 困難である (より詳細には#P 完全である) ことが知られている．パーマネントの計算は定義通りに行なうと $\mathrm{O}^*(n!)$ 時間かかってしまう．Ryser [72] はパーマネントに対する次の公式を発見した．

$$\mathrm{per}(A) = (-1)^n \sum_{S \subseteq \{1,\ldots,n\}} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j \in S} a_{ij}.$$

この Ryser の公式を用いると $\mathrm{O}^*(2^n)$ 時間で行なえる．

Bax & Franklin [7] は 0-1 行列のパーマネントに対して Ryser の公式にパラメータを導入した一般化を考え，そのパラメータを上手に調整することで，和に出現する非ゼロの項の数を少なくする手法を考案した．これによって，0-1 行列のパーマネントが $\exp(-\Omega(n^{1/3}/(2\ln n)))\mathrm{O}^*(2^n)$ という期待時間で計算できる．

Karp [54] と Bax [4] はハミルトン道の数に対する包除公式を与えて，ハミルトン道が $\mathrm{O}^*(2^n)$ 時間で数え上げられることを示した．同様に，ハミルトン閉路も $\mathrm{O}^*(2^n)$ で数え上げられる．ここで n は与えられたグラフの頂点数である．似た問題に対する変種は Bax [5] および Bax & Franklin [6] が議論している．

6.3 線形計画法の組合せ解法とその抽象化

線形計画問題が強多項式時間で解けるかどうかは数理計画のみならず，アルゴリズム理論の中の大きな未解決問題の 1 つである．現在最速の (確率的な) 強多項式時間アルゴリズムの計算量は準指数関数的

[4] 行列乗法として $\omega = 3$ の定義通りのアルゴリズムを用いると $\mathrm{O}^*(2^{\omega n/3}) = \mathrm{O}^*(2^n)$ という自明なアルゴリズムと同じ計算量が得られてしまうことに注意．

(subexponential) なものであり，これは Kalai [52, 53] と Matoušek, Sharir & Welzl [61] によって独立に与えられた．Matoušek, Sharir & Welzl [61] のアルゴリズムはより一般的な LP 型問題 (LP-type problem) と呼ばれる (一般化線形計画問題，generalized linear program，とも呼ばれる) ものにも適用できる．これに似た抽象的枠組が Gärtner [41] によっても考えられている．解説論文として Gärtner & Welzl [43] のものがある．

パリティ・ゲーム (parity game) と呼ばれる組合せゲームがあるけれども，その勝者を定める問題は多項式時間で解けるかどうか分かっていない．Björklund, Sandberg & Vorobyov [10] はこの問題に対して準指数関数的計算量のアルゴリズムを与えた．このアルゴリズムは線形計画法に対する Kalai のアルゴリズム [52, 53] に似ている．また，彼らは最近，平均利得ゲーム (mean payoff game) に対しても同様のアルゴリズムを開発した [11]．

Szabó & Welzl [76] は n 次元 Hamming 立方体の辺の向き付けで，各部分立方体にシンクがただ 1 つしか存在しないようなものを考え，全体のシンクを探す問題を研究した．これは，立方体上の線形計画問題や係数行列を P 行列とする線形相補性問題を抽象化した枠組である．この問題に対して，彼らは $O^*(1.61^n)$ の決定性アルゴリズムと $O^*(1.47^n)$ の確率的 (Las Vegas) アルゴリズムを与えた．このアルゴリズムは上記の線形相補性問題に対する現在最速のアルゴリズムである．この問題に対する更なる展開については Schurr の博士論文 [81] を参照のこと．Gärtner, Morris & Rüst [42] は立方体から単体の積への拡張を考えている．

7 最後に

厳密計算の分野における異なる手法を見てきた．その中のいくつかについて言えることは「列挙」が要になっていることである．よって，厳密計算という分野の立場からも高速な列挙アルゴリズムの構築は重要な研究対象である．また，グラフに対するアルゴリズムについては，極値グラフ理論に関する結果 (ある対象の個数の上界を与える結果) が重要な役割を果たしている．そのような純粋にグラフ理論的だと思われる結果がアルゴリズムに生かされる部分も興味深く，その観点から極値グラフ理論の結果を見直す必要があると思われる．

始めの節で名前だけ紹介したパラメータ化計算の分野と共に，厳密計算の分野は進展し始めている．2004 年の 9 月にパラメータ化計算と厳密計算に関する最初の国際会議 International Workshop on Parameterized and Exact Computation がノルウェーのベルゲンで開かれた[5]．しかし，この会議での日本人発表者は私だけであった．本稿で紹介した結果のいくつかは極めて古いものであり，また他のいくつかは極めて新しいものである．本稿によって，より多くの日本人研究者がこの古くて新しい分野に魅力を感じて頂ければ幸いである．パラメータ化計算に関する解説はまた別の機会に行ないたい．

参考文献

[1] L.G. Aleksandrov and H.N. Djidjev. Linear algorithms for partitioning embedded graphs of bounded genus. SIAM Journal of Discrete Mathematics **9** (1996) 129–150.

[2] N. Alon, P.D. Seymour and R. Thomas. A separator theorem for graphs with an excluded minor and its applications. Proc. 22nd STOC (1990) 293–299.

[3] N. Alon, P.D. Seymour and R. Thomas. A separator theorem for nonplanar graphs. Journal of the American Mathematical Society **3** (1990) 801–808.

[4] E.T. Bax. Inclusion and exclusion algorithm for the Hamiltonian path problem. Information Processing Letters **47** (1993) 203-207.

[5] E.T. Bax. Algorithms to count paths and cycles. Information Processing Letters **52** (1994) 249–252.

[6] E. Bax and J. Franklin. A finite-difference sieve to count paths and cycles by length. Information Processing Letters **60** (1996) 171–176.

[5] この国際会議の第 2 回が 2006 年にチューリッヒ (スイス) で開かれる予定のようである．

[7] E. Bax and J. Franklin. A permanent algorithm with $\exp[\Omega(n^{1/3}/2\ln n)]$ expected speedup for 0-1 matrices. Algorithmica **32** (2002) 157–162.
[8] R. Beigel and D. Eppstein. 3-Coloring in time $O(1.3289^n)$. Journal of Algorithms **54** (2005) 168–204.
[9] R. Bellman. Dynamic programming treatment of the traveling salesman problem. Journal of the ACM **9** (1962) 61–63.
[10] H. Björklund, S. Sandberg and S. Vorobyov. A discrete subexponential algorithm for parity games. Proc. 20th STACS, Lecture Notes in Computer Science **2607** (2003) 663–674.
[11] H. Björklund, S. Sandberg and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. Proc. 29th MFCS, Lecture Notes in Computer Science **3153** (2004) 673–685.
[12] H.L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. Theoretical Computer Science **209** (1998) 1–45.
[13] V. Bouchitté and I. Todinca. Treewidth and minimum fill-in: grouping the minimal separators. SIAM Journal on Computing **31** (2001) 212–232.
[14] T. Brueggemann and W. Kern. An improved local search algorithm for 3-SAT. Theoretical Computer Science **329** (2004) 303–313.
[15] J.M. Byskov. Enumerating maximal independent sets with applications to graph colouring. Operations Research Letters **32** (2004) 547–556.
[16] J. Chen and I.A. Kanj. Improved exact algorithms for Max-SAT. Discrete Applied Mathematics **142** (2004) 17–27.
[17] J. Chen, I.A. Kanj and G. Xia. Labeled search trees and amortized analysis: improved upper bounds for NP-hard problems. Algorithmica, to appear.
[18] Z.-Z. Chen. Approximation algorithms for independent sets in map graphs. Journal of Algorithms **41** (2001) 20–40.
[19] Z.-Z. Chen, M. Grigni and C. Papadimitriou. Map graphs. Journal of the ACM **49** (2002) 127–138.
[20] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. Journal of Symbolic Computation **9** (1990) 251–280.
[21] A. Czumaj, M. Grigni, P. Sissokho and H. Zhao. Approximation schemes for minium 2-edge-connected and biconnected subgraphs in planar graphs. Proc. 15th SODA (2004) 489–498.
[22] E. Dantsin, A. Goerdt, E.A. Hirsch, R. Kannan, J. Kleinberg, C.H. Papadimitriou, P. Raghavan and U. Schöning. A deterministic $(2-\frac{2}{k+1})^n$ algorithm for k-SAT based on local search. Theoretical Computer Science **289** (2002) 69–83.
[23] E. Dantsin, E.A. Hirsch and A. Wolpert. Algorithms for SAT based on search in Hamming balls. Proc. 21st STACS, Lecture Notes in Computer Science **2996** (2004) 141–151.
[24] E. Dantsin and A. Wolpert. Derandomization of Schuler's algorithm for SAT. Proc. 7th SAT (2004) 69–75.
[25] E. Dantsin and A. Wolpert. An improved upper bound for SAT. Proc. 8th SAT, Lecture Notes in Computer Science **3569** (2005), to appear.
[26] V.G. Deĭneko, M. Hoffmann, Y. Okamoto and G.J. Woeginger. The traveling salesman problem with few inner points. Operations Research Letters, to appear.
[27] H.N. Djidjev. A separator theorem for graphs of fixed genus. Serdica **11** (1985) 319–329.
[28] H.N. Djidjev and S.M. Venkatesan. Reduced constants for simple cycle graph separation. Acta Informatica **34** (1997) 231–243.
[29] R.G. Downey and M.R. Fellows. Parameterized Complexity. Springer, Berlin, 1999.
[30] S. Dreyfus and R. Wagner. The Steiner problem in graphs. Networks **1** (1972) 195–207.
[31] D. Eppstein. Improved algorithms for 3-coloring, 3-edge-coloring, and constraint satisfaction. Proc. 12th SODA (2001) 329–337.[6]
[32] D. Eppstein. Small maximal independent sets and faster exact graph coloring. Journal of Graph Algorithms & Applications **7** (2003) 131–140.[7]
[33] D. Eppstein. The traveling salesman problem for cubic graphs. Proc. 8th WADS, Lecture Notes in Computer Science **2748** (2003) 307–318.
[34] D. Eppstein. Quasiconvex analysis of backtracking algorithms. Proc. 15th SODA (2004) 788–797.
[35] S.S. Fedin and A.S. Kulikov. Automated proofs of upper bounds on the running time of splitting algorithms. Proc. 1st IWPEC, Lecture Notes in Computer Science **3162** (2004) 248–259.
[36] U. Feige. Coping with the NP-hardness of the graph bandwidth problem. Proc. 7th SWAT, Lecture Notes in Computer Science **1851** (2000) 10–19.
[37] F.V. Fomin, F. Grandoni and D. Kratsch. Measure and conquer: Domination – A case study. Proc. 32nd ICALP, to appear.
[38] F.V. Fomin, D. Kratsch and G.J. Woeginger. Exact (exponential) algorithms for the dominating set problem.

[6] http://arxiv.org/cs.DS/0009006 より入手可能．
[7] http://www.cs.brown.edu/publications/jgaa/volume07.html より入手可能．

Proc. 30th WG, Lecture Notes in Computer Science **3353** (2004) 245–246.
[39] F.V. Fomin and D.M. Thilikos. A simple and fast approach for solving problems on planar graphs. Proc. 21st STACS, Lecture Notes in Computer Science **2996** (2004) 56–67.
[40] F.V. Fomin, I. Todinca and D. Kratsch. Exact (exponential) algorithms for treewidth and minimum fill-in. Proc. 31st ICALP, Lecture Notes in Computer Science **3142** (2004) 568–580.
[41] B. Gärtner. A subexponential algorithm for abstract optimization problems. SIAM Journal on Computing **24** (1995) 1018–1035.
[42] B. Gärtner, W. Morris Jr. and L. Rüst. Unique sink orientations of grids. Proc. 11th IPCO, Lecture Notes in Computer Science **3509** (2005) 210–224.
[43] B. Gärtner and E. Welzl. Linear programming — randomization and abstract frameworks. Proc. 13th STACS, Lecture Notes in Computer Science **1064** (1996) 669–687.
[44] M. Grigni, E. Koutsoupias and C. Papadimitriou. An approximation scheme for planar graph TSP. Proc. 28th FOCS (1995) 640–645.
[45] J. Gramm, J. Guo, F. Hüffner and R. Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. Algorithmica **39** (2004) 321–347.
[46] J. Gramm, J. Guo, R. Niedermeier and P. Rossmanith. Worst-case upper bounds for MAX-2-SAT with application to MAX-CUT. Discrete Applied Mathematics **130** (2003) 139–155.
[47] F. Grandoni. A note on the complexity of minimum dominating set. Journal of Discrete Algorithms, to appear.
[48] M. Held and R.M. Karp. A dynamic programming approach to sequencing problems. Journal of SIAM **10** (1962) 196–210.
[49] E. Horowitz and S. Sahni. Computing partitions with applications to the knapsack problem. Journal of the ACM **21** (1974) 277–292.
[50] R.Z. Hwang, R.C. Chang and R.C.T. Lee. The searching over separators strategy to solve some NP-hard problems in subexponential time. Algorithmica **9** (1993) 398–423.
[51] K. Iwama and S. Tamaki. Improved upper bounds for 3-SAT. Proc. 15th SODA (2004) 328–329.
[52] G. Kalai. A subexponential randomized simplex algorithm. Proc. 24th STOC (1992) 475–482.
[53] G. Kalai. Linear programming, the simplex algorithm and simple polytopes. Mathematical Programming **79** (1997) 217–234.
[54] R.M. Karp. Dynamic programming meets the principle of inclusion and exclusion. Operations Research Letters **1** (1982) 49–51.
[55] M. Kaufmann. Schöning's k-SAT-Algorithmus (in German). Diplomarbeit, Institut für Theoretische Informatik, ETH Zürich, 2002.
[56] A.S. Kulikov. Automated generation of simplification rules for SAT and MAXSAT. Proc. 8th SAT, Lecture Notes in Computer Science **3569** (2005), to appear.
[57] O. Kullmann. New methods for 3-SAT decision and worst case analysis. Theoretical Computer Science **223** (1999) 1–72.
[58] E.L. Lawler. A note on the complexity of the chromatic number problem. Information Processing Letters **5** (1976) 66–67.
[59] R.J. Lipton and R.E. Tarjan. A separator theorem for planar graphs. SIAM Journal on Applied Mathematics **36** (1979) 177–189.
[60] R.J. Lipton and R.E. Tarjan. Applications of a planar separator theorem. SIAM Journal on Computing **9** (1980) 615–627.
[61] J. Matoušek, M. Sharir and E. Welzl. A subexponential bound for linear programming. Algorithmica **16** (1996) 498–516.
[62] B. Monien and E. Speckenmeyer. Solving satisfiability in less than 2^n steps. Discrete Applied Mathematics **10** (1985) 287–295.
[63] J.W. Moon and L. Moser. On cliques in graphs. Israel Journal of Mathematics **3** (1965) 23–28.
[64] R. Niedermeier. Invitation to Fixed-Parameter Algorithms. Habilitation Thesis, University of Tübingen, 2002.[8]
[65] J. Pach and P.K. Agarwal. Combinatorial Geometry. John Wiley & Sons, New York, 1995.
[66] C.H. Papadimitriou. On selecting a satisfying truth assignment. Proc. 32nd FOCS (1991) 163–169.
[67] R. Paturi, P. Pudlák, M.E. Saks and F. Zane. An improved exponential-time algorithm for k-SAT. Proc. 39th FOCS (1998) 628–637.
[68] R. Paturi, P. Pudlák and F. Zane. Satisfiability coding lemma. Chicago Journal of Theoretical Computer Science **1999** (1999) Article 11.
[69] P. Pudlák. Satisfiability — algorithms and logic. Proc. 23rd MFCS, Lecture Notes in Computer Science **1450** (1998) 129–141.

[8] http://www-fs.informatik.uni-tuebingen.de/~niedermr/ より入手可能．改訂版が Oxford University Press から (2005 年頃に) 出版されるようである．

[70] A. Razen. Counting satisfying 2-SAT assignments. Diplomarbeit, Institut für Theoretische Informatik, ETH Zürich, 2005.
[71] B. Reed. Paths, stars and the number three. Combinatorics, Probability and Computing **5** (1996) 277–295.
[72] H.J. Ryser. Combinatorial Mathematics. Carus Mathematical Monograph, No. 14, John Wiley & Sons, New York, 1963.
[73] J.M. Robson. Algorithms for maximum independent sets. Journal of Algorithms **7** (1986) 425–440.
[74] J.M. Robson. Finding a maximum independent set in time $O(2^{n/4})$. Manuscript, 2001.[9]
[75] D. Rolf. Derandomization of PPSZ for Unique-k-SAT. Proc. 8th SAT, Lecture Notes in Computer Science **3569** (2005), to appear.
[76] T. Szabó and E. Welzl. Unique sink orientations of cubes. Proc. 42nd FOCS (2001) 547–555.
[77] U. Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. Algorithmica **32** (2002) 615–623.
[78] U. Schöning. Algorithmics in exponential time. Proc. 22nd STACS, Lecture Notes in Computer Science **3404** (2005) 36–43.
[79] R. Schroeppel and A. Shamir. A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. SIAM Journal on Computing **10** (1981) 456–464.
[80] R. Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. Journal of Algorithms **54** (2005) 40–44.
[81] I. Schurr. Unique sink orientations of cubes. PhD Thesis, No. 15747, ETH Zurich, 2004.
[82] A.D. Scott and G.B. Sorkin. Faster algorithms for MAX CUT and MAX CSP, with polynomial expected time for sparse instances. Proc. 6th APPROX + 7th RANDOM, Lecture Notes in Computer Science **2764** (2003) 382–395.
[83] D.A. Spielman and S.-H. Teng. Disk packings and planar separators. Proc. 12th SCG (1996) 349–358.
[84] R.E. Tarjan and A.E. Trojanowski. Finding a maximum independent set. SIAM Journal on Computing **6** (1977) 537–546.
[85] R. Williams. Algorithms for quantified Boolean formulas. Proc. 13th SODA (2002) 299–307.
[86] R. Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. Theoretical Computer Science, to appear.
[87] G.J. Woeginger. Exact algorithms for NP-hard problems: a survey. Combinatorial Optimization — Eureka! You shrink!, Lecture Notes in Computer Science **2570** (2003) 185–207.[10]
[88] G.J. Woeginger. Space and time complexity of exact algorithms: some open problems. Proc. 1st IWPEC, Lecture Notes in Computer Science **3762** (2004) 281–290.
[89] M. Yamamoto. Another proof for an upper bound of a local search algorithm for 3-SAT. Research Reports on Mathematics and Computing Sciences **C-197**, Tokyo Institute of Technology, 2004.[11]

[9] http://dept-info.labri.u-bordeaux.fr/~robson/mis/techrep.ps より入手可能．
[10] http://www.win.tue.nl/~gwoegi/ より入手可能．
[11] http://www.is.titech.ac.jp/research/research-report/C/index.html より入手可能．