頂点彩色問題に対する列生成法アプローチの高速化

玉木 久夫 †　　平川 宗則 ††

† 明治大学理工学部情報科学科　†† 明治大学大学院理工学研究科
〒214-8571　神奈川県川崎市多摩区東三田 1-1-1
{tamaki, hiranasu} @cs.meiji.ac.jp

概要　Mehrotra と Trick はグラフの頂点彩色問題を極大独立点集合による被覆問題として定式化し、その線形緩和問題を厳密に解く列生成法を提案した。しかし全体の目的からは線形緩和問題を厳密に解く必要はなく、最適値を切り上げた整数値が求まれば十分であり、列生成の打ち切り条件を緩和することができる。そのような打ち切り条件の緩和を行うことで、多くの場合に染色数の下界計算が高速化されることを実験により確認した。また、この考え方をさらに進めて、初期列の質が悪く列生成法に時間がかかる時に対処するアプローチを提案し実験した。

キーワード ： 頂点彩色問題 列生成法

Speeding Up a Column Generation Approach for Vertex Coloring Problem

Hisao Tamaki†, Munenori Hirakawa††

†Department of Computer Science, Meiji University
††Graduate School of Science and Technology, Meiji University
1-1-1 Higashi-mita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571

Abstract Mehrotra and Trick formulated the vertex coloring problem as that of covering vertices by maximal independent sets, and proposed a column generation approach that solves the linear relaxation of the problem exactly. For our purposes, however, we only need an integral lower bound, which we can exploit to stop generating columns before we reach the LP optimal. Experiment show that the lower bound calculations are often dramatically sped-up with these early stopping conditions. We also extend this idea to reduce processing time when the initial set of columns is of bad quality.

Key Words : Vertex coloring problem, Column generation

1　はじめに

頂点彩色問題（以下、彩色問題と呼ぶ）とは、グラフ理論における組合せ最適化問題であり、現実世界においては、電波の周波数割り当て問題、レジスタ割り当て問題などに応用されている。また、今回の高速化を行う動機ともなった論理回路の故障検出テストパターンの静的圧縮問題とも深い関わりがある。この問題については 4 章で述べている。

彩色問題は問題の性質上、解の対称性が多く、定式化によっては CPLEX などの数理計画ソルバにとって解きづらい問題である。これを克服するため Mehrotra と Trick は、彩色問題を極大独立点集合による被覆問題として定式化する方法と、その線形緩和問題を厳密に解く列生成法アプローチを提案した [1]。しかし、求めたい最適な染色数が整数であることを考慮すれば、線形緩和問題を厳密に解く必要はなく、線形緩和問題の最適値を切り上げた整数値が求まれば十分である。本稿ではこのことより、列生成を打ち切る条件を緩和できることを示し、多くの場合、高速に染色数の下界を求められるようになったことを実験により確認した。また、目標となる下界を低く設定し（それにより列生成の打ち切りは早くできる）その目標値を徐々に上げていくアプローチを提案し、列生成法の基となる初期列の質が悪い時に効果があることを、実験により確認した。

Mehrotra と Trick が提案した彩色問題の定式化は 2 章で、それに対する列生成法は 3 章で示す。さらに 3 章では、列生成の打ち切り条件を緩和するアイディアも述べる。その効果を 4 章において、計算機実験により確認する。

2 定式化

彩色問題の整数計画問題への定式化はいくつか存在するが、ここでは Mehrotra と Trick[1] による、以下のような定式化を考える。

グラフ $G=(V,E)$ を頂点集合 $V=\{1,\cdots,n\}$ と辺集合 E からなる無向グラフとする。G の頂点彩色（以下、彩色という）とは、辺の両端点は違う色であるという制約の下で、すべての頂点に色をつけることであり、最小な彩色とは、すべての彩色の中で色の種類の数が最も少ないような彩色のことを指す。G の独立点集合とは、その集合内のどの頂点対の間にも辺がないような頂点集合のことである。彩色において同じ色がついている頂点集合は独立点集合である。極大独立点集合とは、ある独立点集合に他のどの頂点を加えても独立点集合でなくなるようなものを指す。

S^* を G の極大独立点集合すべてからなる集合とする。すべての $s\in S^*$ に対して x_s という 0-1 変数を定義する。x_s は、s に属するすべての頂点に 1 つの色を塗るときに 1、そうでないときに 0 を取る。そうすると、G に対する彩色問題は次のような整数線形計画によって定式化される（以下 IS と呼ぶ）。

$$minimize \quad \sum_s x_s \tag{1}$$

$$s.t. \quad \sum_{\{s:i\in s\}} x_s \geq 1 \qquad ({}^\forall i \in V) \tag{2}$$

$$x_s \in \{0,1\} \qquad ({}^\forall s \in S^*) \tag{3}$$

IS の実行可能解において、1 つの頂点に複数の色がつく時は、どの 1 つの色を選んで塗ることにしても彩色の条件を満たす。

IS の x_s に対する制約を $0\leq x_s\leq 1$ に緩和した問題を、以下 $IS-LP$ と呼ぶことにする。

3 列生成法アプローチ

3.1 列生成法

前章の定式化は、極大独立点集合すべてからなる集合 S^* を用いている。しかし、あるグラフ G における極大独立点集合（$IS-LP$ における列）を全て列挙することは、G が大きくなればなるほど困難になる。このように、列の個数が膨大であるような線形計画に対しては、一般に列の小さい集合から出発し、必要に応じて列を生成して加えていく列生成法が用いられる [2]。$IS-LP$ に対して Mehrotra と Trick は次のような列生成法を提案した [1]。

初めに、$IS-LP$ において S^* を一般の $S\subseteq S^*$ で置き換えた問題を考え、その双対問題を次のように定義する（以下 $ISD(S)$ と呼ぶ）。

$$maximize \quad \sum_{i\in V} \pi_i \tag{4}$$

$$s.t. \quad \sum_{i\in s} \pi_i \leq 1 \qquad ({}^\forall s \in S) \tag{5}$$

$$\pi_i \geq 0 \quad ({}^\forall i \in V) \tag{6}$$

まず適当な初期列集合 $S\subseteq S^*$ を用意し $ISD(S)$ を解き、最適解 π を求める。そして π を各頂点の重

みとし、グラフ G の重み付き最大独立点集合問題（以下 $MWIS$ と呼ぶ）を解く。もし $MWIS$ の最適値が 1 を超えていたら、最適値を実現する独立点集合を s とする。この時、π は $ISD(S \cup \{s\})$ の制約に違反しているので、S を $S \cup \{s\}$ で置き直し、$ISD(S)$ を解いて新たな π を計算する。この操作を $MWIS$ の最適値が 1 になるまで繰り返す。繰り返しが終了したときは、任意の独立点集合 $s \in S^*$ に対して π は S に対する制約 (5) を満たすので、π は $ISD(S^*)$ の最適解である。したがって双対定理より、その時の $ISD(S)$ の最適解の値は $IS-LP$ の最適解の値と一致する。

$IS-LP$ に対する列生成法において、列生成を打ち切る条件を次のようにまとめておく。

打ち切り条件 1：
現在の $ISD(S)$ の最適解を π とするとき、π を重みとする $MWIS$ の最適値が 1 であること。

列生成法の良い点は、すべての $s \in S^*$ を列挙しなくても、$IS-LP$ の最適解の値、すなわち IS の下界が求まる点である。難点は、列を生成するごとにこの手順を繰り返さねばならず、また 1 回の $MWIS$ 自体が難しいという問題もある。そこで、$MWIS$ を最後まで解かず、重みの和がある閾値を超えるような独立点集合が見つかり次第、それを列集合に取り入れるようにする。取り入れ条件 1 は、その閾値の値を 1 とする。また Mehrtra と Trick は閾値値を初めは 1.1 に設定し、それを超える重みの和の独立点集合がなくなれば値を 1 にするという提案も行っている。こうすることで列生成ごとの $ISD(S)$ の最適値が降下する速度が速くなることが期待される。この提案を取り入れ条件 2 としておく。

3.2 列生成打ち切り条件の緩和

列生成の打ち切り条件を緩和するために、次の命題を利用する。

命題 3.1. $\pi = \{\pi_1, \cdots, \pi_n\}$ を非負ベクトルとし、w, p をそれぞれ

$$w = \max_{s \in S^*} \sum_{i \in s} \pi_i \tag{7}$$

$$p = \frac{\sum_{i \in V} \pi_i}{w} \tag{8}$$

とおくと、p は、$IS-LP$ の最適解の下界である。

証明. w の定義より、

$$\frac{\sum_{i \in s} \pi_i}{w} \leq 1 \quad ({}^\forall s \in S^*) \tag{9}$$

が成り立つので、$\frac{\pi}{w}$ は $ISD(S^*)$ の実行可能解である。よって双対定理より、p は $IS-LP$ の最適解の下界である。 □

$IS-LP$ の最適値を z^* とおき、列生成法を実行中のある時点における S に対する $ISD(S)$ の最適解を π、最適値を $z = \sum_{i \in V} \pi_i$ とおく。この π を各頂点の重みとする $MWIS$ の最適値は (7) 式により与えられる。命題 3.1 より $\frac{z}{w} \leq z^*$ であり、明らかに $z^* \leq z$ である。もし $\frac{z}{w} > \lfloor z \rfloor$ であれば、$\lceil \frac{z}{w} \rceil = \lceil z^* \rceil = \lceil z \rceil$ であるから、IS に対する $IS-LP$ に基づいた最良の下界 $\lceil z^* \rceil$ は、$\lceil z \rceil$ に等しい。よって、これ以上列生成を続ける必要はない。

上の観察により、$IS-LP$ を解く目的が整数の下界を求めることである場合には、列生成の打ち切り条件を次のように緩めることができる。

打ち切り条件 2：
現在の $ISD(S)$ の最適解を π、最適値を z とするとき、π を重みとする $MWIS$ の最適値が $\frac{z}{\lfloor z \rfloor}$ であること。

打ち切り条件 1 を用いた場合、z が $IS-LP$ の最適値に近づくにつれ、π の微小な変更を繰り返すようになり、ある程度以上の規模の問題ではアルゴリズムの終了までに膨大な時間を費やす。そこで、打ち切り条件 2 を用いることで、アルゴリズム全体にかかる時間の短縮が期待できる。

なお、このアルゴリズムにおいても同様に $MWIS$ を最後まで解かず、重みの和が $\frac{z}{\lfloor z \rfloor}$ を超えるような独立点集合が見つかり次第、それを列集合

に取り入れるようにしている。これを取り入れ条件3としておく。

命題3.1において、wは任意の実数を取ることができるため、下界として結論付けたい値をtと置くと以下のように言うこともできる。

打ち切り条件3：
現在の$ISD(S)$の最適解をπ、最適値をz、下界として結論付けたい値をtとするとき、πを重みとする$MWIS$の最適値が$\frac{z}{t-1}$であること。

上の条件から、tがzより小さい値であればあるほど打ち切り条件が緩和されることがわかる。

一般に列生成法は時間がかかるため、初期列集合の質が良いほど全体の実行時間は短くて済む。しかし、実験結果は4章にて示すが、初期列生成にかけた時間とその質の関係はインスタンスに依存する。そこで、打ち切り条件3を用いて、初期列の質が悪い時に実行時間短縮を期待できるアルゴリズムを考えた。

先に、全ての実験で用いた初期列の生成方法について述べておく。生成には彩色問題に対するヒューリスティックであるDsatur[6]を用いた。手順は、まずDsaturを解き、塗られた色ごとに頂点を集める。そして各頂点集合に対していくつか頂点を加え、極大な独立点集合とした上で列集合に加える。これを繰り返すことで多量の初期列を得られるが、1回の実行だけでも$ISD(S)$が実行可能な列集合が得られる。

打ち切り条件3を用いたアルゴリズムは以下のようにした。まず、Dsaturを1回だけ解き初期列集合を求める。次に、いま下界であるか確かめたい値tを、初期列集合による$IS-LP$の最適解の$\frac{1}{2}$に設定する。打ち切り条件3を用いて列生成法を行いtが下界であると結論できたら、新しいtを、その時の列集合Sによる$ISD(S)$の最適解と$t-1$の平均値に1を足した値に設定し、再び手順を繰り返す。新しいtの値が最適な下界を超えてしまって下界を示せなくなってしまったら、打ち切り条件2に変更して実行する。最適な下界の値は、tが改善しなくなったときの値である。このアルゴリズムでは特に、取り入れ条件の値が$\frac{z}{t-1}$に上がっているため、より早く列生成法が終わることが期待できる。

それぞれの打ち切り条件を用いたときの実験結果は4章にて示す。

4 実験結果

4.1 インスタンスと実験環境

実験には、2種類のインスタンスを用いた。一つは、The Second DIMACS Challenge[3]によるベンチマークインスタンス[4]、もう一つは、回路の故障検出テストパターンから生成されたインスタンスである。

回路の故障検出テストパターンとは、ある論理回路から故障を発見するために行うテストの入力の集合である。入力の集合には、故障をすべて検出するには余分なパターンが多く含まれており、いくつかのパターンをひとつにまとめることができる。この時、必要な最小個数のテストパターン集合を求めるため、入力同士の統合の可否をグラフとして表現し、最小クリーク被覆問題として定式化したアプローチがある[5]。このグラフの補グラフを考えると、同じ問題が頂点彩色問題として定式化できるため、現実問題からくる頂点彩色問題の例として、実験を行った。

計算機環境は以下の通りである。

- CPU：Core2 Duo 2.66GHz
- メモリ：1GB
- コンパイラ：Java 6.0
- 数理計画ソルバ：CPLEX 11.0

4.2 実験の前提

初期列が多ければ多いほど初期列集合の質は良くなる傾向にあるが、どれぐらい初期列生成に時間をかければ十分かはインスタンスに依存する。そこで、この実験では暫定的に、頂点数の1割に相当す

る回数の繰り返しを行う度に、その時点で求まっている列集合による $IS-LP$ の最適解を計算し、それが前回の計算より 1 以上下がらなくなったら繰り返しを終了することにした。

なお、実験では計算誤差を許容できるように計算を行っている。また、全体を通して、1000 秒以内に実行が終了した場合の結果のみを載せている。

4.3 実験 1

まず、打ち切り条件 1 を用いて列生成法を行ったときと、打ち切り条件 2 を用いたときの比較を実験 1 として表 1 に示す。列生成法の下界の質を評価するため、グラフ G の最大クリークを求めるアルゴリズム [7] とも比較を行った。

表 1 からわかるように、時間内に終了したものに関しては、ほぼすべてのインスタンスにおいて時間の短縮が図れていることが確認できる。

4.4 実験 2

次に、打ち切り条件 2 を用いた時と打ち切り条件 3 を用いたときの比較を、実験 2 として表 2 に示す。実験 2 では、初期列の生成量が列生成法に与える影響を観察するために、Dsatur の繰り返し回数を変えながら実験を行った。(1) は Dsatur を 1 回だけ実行した時、(2) は 4.2 章の前提を用いたとき、(3) は Dsatur の開始頂点を変えながら頂点数回繰り返した時である。

表 2 より、初期列は多く生成したほうがその質は良くなり、その方が実行時間全体も速くなる傾向がわかる。しかしその効果もインスタンスに依存していることもわかる。また規模が大きい問題は、初期列生成自体に時間がかかりすぎるものもある。

また表 2 の条件 3 との比較に注目すると、latin_square_10、queen14_14、b14rem、cb15rem、efr_cs13207 など、条件 2 で (1) より (3) が悪い時、つまり初期列の質が悪い時に、条件 3 を用いたアルゴリズムが効果的である傾向が、いくつかのインスタンスから言える。

謝辞

日本大学准教授細川利典博士、明治大学准教授山崎浩二博士にはテストパターンインスタンスのデータを使用させて頂きました。また明治大学八木澤圭氏には最大クリークや Dsatur のプログラムを使用させて頂きました。皆様に深く感謝いたします。

参考文献

[1] Anuj Mehrotra, Michael A. Trick. A column generation approach for graph coloring. INFORMS Journal On Computing, vol. 8, no. 4, pp. 344-354, 1996.

[2] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-Price: Column Generation for Huge Integer Problems. School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, 1994.

[3] The Second DIMACS Challenge. http://mat.gsia.cmu.edu/challenge.html

[4] Michael A. Trick. Network Resources for Coloring a Graph. http://mat.gsia.cmu.edu/COLOR/color.html

[5] 八木澤圭, 山崎浩二, 細川利典, 玉木久夫. テストパターンの静的圧縮における厳密解と貪欲解の比較. 電気情報通信学会 DC2007-67 83,77-82,2007.

[6] D Brélaz. New Methods to color the vertices of a graph. Communications of the ACM, 22:251-256, 1979.

[7] 亀田宗克, 富田悦次. 最大クリーク抽出アルゴリズムの高速化と解析・評価. 情報処理学会 2004-AL-93,33-40,2004.

							列生成法			
					最大クリーク			打ち切り条件 1		打ち切り条件 2
インスタンス名	最適解	頂点数	辺数	密度	下界	時間	下界	*取 1	取 2	
DSJC125.1	5	125	736	0.09	4	0.00	5	>1000	>1000	>1000
DSJC125.5		125	3891	0.50	10	0.00	16	103.67	65.98	37.31
DSJC125.9	44	125	6961	0.90	34	1.22	43	1.09	0.92	0.83
DSJC250.1		250	3218	0.10	4	0.00		>1000	>1000	>1000
DSJC250.5		250	15668	0.50	12	0.14		>1000	>1000	>1000
DSJC250.9		250	27897	0.90	42	>1000	71	11.05	10.20	8.91
DSJC500.1		500	12458	0.10	5	0.02		>1000	>1000	>1000
DSJC500.5		500	62624	0.50	13	8.72		>1000	>1000	>1000
DSJC500.9		500	112437	0.90	50	>1000	123	241.03	239.85	210.59
DSJR500.1	12	109	830	0.14	12	0.00	12	>1000	>1000	0.20
DSJR500.1c	85	500	121275	0.97	76	>1000	85	161.16	120.14	5.16
DSJR500.5		496	58436	0.48	122	12.02	122	184.11	185.92	24.91
DSJC1000.1		1000	49629	0.10	6	0.05		>1000	>1000	>1000
DSJC1000.5		1000	249826	0.50	15	838.92		>1000	>1000	>1000
DSJC1000.9		1000	449449	0.90	56	>1000		>1000	>1000	>1000
flat300_20_0	20	300	21375	0.48	11	0.25		>1000	>1000	>1000
flat300_26_0	26	300	21633	0.48	11	0.28		>1000	>1000	>1000
flat300_28_0	28	300	21695	0.48	12	0.25		>1000	>1000	>1000
flat1000_50_0	50	1000	245000	0.49	15	583.34		>1000	>1000	>1000
flat1000_60_0	60	1000	245830	0.49	15	640.89		>1000	>1000	>1000
flat1000_76_0	76	1000	246708	0.49	15	701.46		>1000	>1000	>1000
fpsol2.i.1	65	86	3341	0.91	65	0.00	65	76.95	78.30	0.20
fpsol2.i.2	30	112	2669	0.43	30	0.00	30	>1000	>1000	0.17
fpsol2.i.3	30	112	2667	0.43	30	0.00	30	>1000	>1000	0.20
inithx.i.1	54	317	12634	0.25	54	0.02	54	>1000	>1000	1.08
inithx.i.2	31	182	4476	0.27	31	0.00	31	>1000	>1000	0.33
inithx.i.3	31	172	4176	0.28	31	0.00	31	>1000	>1000	0.33
latin_square_10		900	307350	0.76	90	>1000	90	193.00	193.13	138.67
le450_5a	5	450	5714	0.06	5	0.02		>1000	>1000	>1000
le450_5b	5	450	5734	0.06	5	0.02		>1000	>1000	>1000
le450_5c	5	450	9803	0.10	5	0.02	5	>1000	>1000	91.61
le450_5d	5	450	9757	0.10	5	0.02	5	>1000	>1000	91.17
le450_15a	15	407	7802	0.09	15	0.02	15	>1000	>1000	>1000
le450_15b	15	410	7824	0.09	15	0.02	15	7.41	7.41	6.48
le450_15c	15	450	16680	0.17	15	0.02		>1000	>1000	>1000
le450_15d	15	450	16750	0.17	15	0.02		>1000	>1000	>1000
le450_25a	25	264	5831	0.17	25	0.00	25	1.80	1.80	1.27
le450_25b	25	294	6240	0.14	25	0.00	25	>1000	>1000	1.52
le450_25c	25	435	17096	0.18	25	0.02	25	>1000	>1000	>1000
le450_25d	25	433	17106	0.18	25	0.02	25	>1000	>1000	>1000

表 1　実験 1 の結果　　　　*取 1・取 2･･･取り入れ条件 1、取り入れ条件 2

							列生成法			
					最大クリーク			打ち切り条件 1		打ち切り条件 2
インスタンス名	最適解	頂点数	辺数	密度	下界	時間	下界	取 1	取 2	
mulsol.i.1	49	123	3462	0.46	49	0.00	49	>1000	>1000	0.25
mulsol.i.2	31	101	2157	0.43	31	0.02	31	>1000	>1000	0.19
mulsol.i.3	31	102	2188	0.42	31	0.00	31	>1000	>1000	0.20
mulsol.i.4	31	103	2218	0.42	31	0.00	31	>1000	>1000	0.22
mulsol.i.5	31	103	2216	0.42	31	0.00	31	>1000	>1000	0.16
school1	14	356	18950	0.30	14	35.23	14	>1000	>1000	35.61
school1_nsh	14	328	14519	0.27	14	2.94	14	>1000	>1000	0.92
homer	13	61	494	0.27	13	0.02	13	>1000	>1000	0.13
jean	10	31	157	0.34	10	0.00	10	71.66	71.44	0.13
games120	9	120	638	0.09	9	0.00	9	>1000	>1000	0.19
miles750	31	36	617	0.98	31	0.00	31	0.14	0.14	0.13
queen5_5	5	25	160	0.53	5	0.00	5	0.27	0.25	0.09
queen6_6	7	36	290	0.46	6	0.00	7	1.19	0.80	0.34
queen7_7	7	49	476	0.40	7	0.00	7	31.63	31.42	0.31
queen8_12	12	96	1368	0.30	12	0.00	12	>1000	>1000	0.20
queen8_8	9	64	728	0.36	8	0.02	9	9.09	6.88	1.75
queen9_9	10	81	1056	0.33	9	0.00	9	89.16	35.88	21.08
queen10_10		100	1470	0.30	10	0.00	10	123.58	110.94	47.89
queen11_11	11	121	1980	0.27	11	0.00	11	110.91	154.11	61.78
queen12_12		144	2596	0.25	12	0.00	12	256.02	248.78	104.08
queen13_13	13	169	3328	0.23	13	0.02	13	448.39	>1000	236.88
queen14_14		196	4186	0.22	14	0.02	14	948.64	>1000	525.28
queen15_15		225	5180	0.21	15	0.02		>1000	>1000	>1000
queen16_16		256	6320	0.19	16	0.02		>1000	>1000	>1000
myciel3	4	11	20	0.36	2	0.00	3	0.13	0.11	0.13
myciel4	5	23	71	0.28	2	0.00	4	0.25	0.17	0.17
myciel5	6	47	236	0.22	2	0.00	4	1.80	0.75	0.28
myciel6	7	95	755	0.17	2	0.02	4	>1000	4.34	8.52
myciel7	8	191	2360	0.13	2	0.00	5	>1000	>1000	25.14
c3540rem	169	552	144933	0.95	165	>1000	168	>1000	>1000	8.50
cs832rem	19	36	580	0.92	18	0.00	19	0.17	0.17	0.14
cs953rem	113	164	12970	0.97	112	0.06	113	4.05	4.06	0.44
cs1488rem	72	135	8775	0.97	71	0.59	72	1.23	1.27	0.30
b14rem	599	702	239773	0.97	598	1.73	599	333.25	323.36	165.55
cb14rem	1249	3535	6057237	0.97	811	>1000	1249	>1000	>1000	>1000
cb15rem	769	2403	2658437	0.92	684	>1000	769	>1000	>1000	>1000
cs5378rem		12431	41460308	0.54	168	0.59		>1000	>1000	205.42
efr_cs9234		12431	41460308	0.54	260	>1000		>1000	>1000	>1000
efr_cs13207		12431	41460308	0.54	309	4.08	309	>1000	>1000	181.57

	打ち切り条件 2				条件 3	
インスタンス名	下界	(1)	(2)	(3)	下界	
DSJC125.1	5	445.79	>1000	549.73	5	648.49
DSJC125.5		47.13	37.31	31.58	16	26.19
DSJC125.9	44	2.17	0.83	0.83	43	1.67
DSJC250.1		>1000	>1000	>1000		>1000
DSJC250.5		>1000	>1000	>1000	(25)	762.35
DSJC250.9		41.83	8.91	11.95	71	21.2
DSJC500.1		>1000	>1000	>1000		>1000
DSJC500.5		>1000	>1000	>1000		>1000
DSJC500.9		994.37	210.59	198.33	123	398.42
DSJR500.1	12	0.63	0.2	0.33	12	0.58
DSJR500.1c	85	5.61	5.16	13.78	85	5.64
DSJR500.5		>1000	24.91	313.09	122	148.36
DSJC1000.1		>1000	>1000	>1000		>1000
DSJC1000.5		>1000	>1000	>1000		>1000
DSJC1000.9		>1000	>1000	>1000		>1000
flat300_20_0	20	>1000	>1000	>1000		>1000
flat300_26_0	26	>1000	>1000	>1000	(22)	157.88
flat300_28_0	28	>1000	>1000	>1000	(22)	117.53
flat1000_50_0	50	>1000	>1000	>1000		>1000
flat1000_60_0	60	>1000	>1000	>1000		>1000
flat1000_76_0	76	>1000	>1000	>1000		>1000
fpsol2.i.1	65	0.19	0.2	0.36	65	0.19
fpsol2.i.2	30	0.33	0.17	0.27	30	0.33
fpsol2.i.3	30	0.25	0.2	0.27	30	0.25
inithx.i.1	54	1.44	1.08	3.19	54	1.31
inithx.i.2	31	0.38	0.33	1.09	31	0.44
inithx.i.3	31	0.39	0.33	1.05	31	0.42
latin_square_10		512.28	138.67	>1000	90	461.66
le450_5a	5	>1000	>1000	>1000		>1000
le450_5b	5	>1000	>1000	>1000		>1000
le450_5c	5	>1000	91.61	146.41		>1000
le450_5d	5	>1000	91.17	92.7		>1000
le450_15a	15	>1000	>1000	17.22		>1000
le450_15b	15	>1000	6.48	16.55		>1000
le450_15c	15	>1000	>1000	>1000		>1000
le450_15d	15	>1000	>1000	>1000		>1000
le450_25a	25	2.06	1.27	10.61	25	1.48
le450_25b	25	1.72	1.52	13.03	25	1.78
le450_25c	25	>1000	>1000	46.63		>1000
le450_25d	25	>1000	>1000	47.17		>1000

	打ち切り条件 2				条件 3	
インスタンス名	下界	(1)	(2)	(3)	下界	
mulsol.i.1	49	0.42	0.25	0.47	49	0.38
mulsol.i.2	31	0.2	0.19	0.33	31	0.22
mulsol.i.3	31	0.2	0.2	0.33	31	0.22
mulsol.i.4	31	0.2	0.22	0.33	31	0.22
mulsol.i.5	31	0.31	0.16	0.33	31	0.33
school1	14	3.08	35.61	5.53	14	591.25
school1_nsh	14	>1000	0.92	5.24	(10)	33.56
homer	13	0.19	0.13	0.14	13	0.19
jean	10	0.14	0.13	0.11	10	0.16
games120	9	0.22	0.19	0.44	9	0.23
miles750	31	0.11	0.13	0.13	31	0.11
queen5_5	5	0.09	0.09	0.09	5	0.09
queen6_6	7	0.3	0.34	0.22	7	0.31
queen7_7	7	0.61	0.31	0.3	7	0.38
queen8_12	12	0.8	0.2	0.59	12	0.95
queen8_8	9	2.52	1.75	1.34	9	2.47
queen9_9	10	12.02	21.08	16.47	9	19.75
queen10_10		55.33	47.89	92.45	10	51.15
queen11_11	11	305.95	61.78	236.11	11	102.93
queen12_12		266.27	104.08	185.25	12	160.67
queen13_13	13	260.52	236.88	378.63	13	673.93
queen14_14		553.14	525.28	750.55	(10)	191.59
queen15_15		>1000	>1000	>1000	(10)	915.53
queen16_16		>1000	>1000	>1000		>1000
myciel3	4	0.11	0.13	0.11	3	0.11
myciel4	5	0.19	0.17	0.13	4	0.2
myciel5	6	0.23	0.28	0.2	4	0.23
myciel6	7	8.83	8.52	1.74	4	1.58
myciel7	8	24.84	25.14	18.58	5	29.81
c3540rem	169	15.02	8.5	21.92	168	11.2
cs832rem	19	0.13	0.14	0.13	19	0.13
cs953rem	113	0.22	0.44	1.42	113	0.23
cs1488rem	72	0.16	0.3	0.72	72	0.2
b14rem	599	2.83	165.55	816.11	599	3.05
cb14rem	1249	287.83	>1000	>1000	1249	292.02
cb15rem	769	217.63	>1000	>1000	769	140.36
cs5378rem		>1000	205.42	>1000	168	409.97
efr_cs9234		>1000	>1000	>1000	(250)	882.59
efr_cs13207		19.922	181.57	529.08	309	19.1

表 2　実験 2 の結果

注:

打ち切り条件 3 において、最適な下界が時間内に求まらなかったインスタンスについては、時間内に示せた最良の下界をカッコつきで載せている。