最小拘束問題の動的計画法アルゴリズム

02103380　防衛大学校情報工学科　*加治屋 政誉司　KAJIYA Masayoshi
01107880　防衛大学校情報工学科　片岡　靖詞　KATAOKA Seiji

1　はじめに

教官 m 人，学生 n 人がおり，各学生には複数の卒論指導教官がいる．卒論発表会において，各教官は，最初の担当学生の発表開始から最後の担当学生の発表終了まで拘束される．このとき，学生の発表順序をスケジュールすることにより，教官の拘束時間の総和を最小化する問題を**最小拘束問題** (Minimum Binding Problem : MBP) と呼ぶことにする．

各教官にとって，担当学生の有無は図1のような0-1行列で表現でき，拘束時間は最後の列に示されている．この行列の列を適当に交換し，図2のようにすれば，教官の総拘束時間は，37から26へと短縮できる．

時限	1	2	3	4	5	6	7	8	9	10	拘束
学生	1	2	3	4	5	6	7	8	9	10	時間
1	1	0	1	0	1	0	0	1	1	0	9
教2	1	0	1	1	0	0	1	0	0	1	10
官3	1	1	0	1	0	1	0	0	1	1	10
4	0	0	1	0	1	1	1	0	1	1	8

図 1: スケジュール前

時限	1	2	3	4	5	6	7	8	9	10	拘束
学生	2	4	7	10	1	3	6	9	5	8	時間
1	0	0	0	0	1	1	0	1	1	1	6
教2	0	1	1	1	1	1	0	0	0	0	5
官3	1	1	0	1	1	0	1	1	0	0	8
4	0	0	1	1	0	1	1	1	1	0	7

図 2: スケジュール後

MBP の適用例は幅広く，例えば教官をレンタル機材，学生を工程とみなした場合，レンタルの固定費が十分高ければ1度借りた機材は，その機材を利用する工程が修了するまで借りておいた方がよく，MBP をレンタル計画に適用できる．また，教官を俳優，学生を撮影の1シーンと捕らえることにより，最適撮影計画にも適用できる．

最適順序付けという意味において，MBP は巡回セールスマン問題 (TSP) と関連が深いが，TSP では点 i の次に点 j を訪問するとき，距離の増加量は事前に与えられる距離行列によって固定されており，点 i 以前に訪問している点集合に依存しない．一方，MBP では，学生 i の次に学生 j を並べるとき，学生 i よりも前にどのような学生が並べられているかに依存して，拘束時間の増加量が異なる．したがって，MBP は目的関数値の増加量が，既に並べた集合に依存する関数として決められ，TSP の距離増加量をさらに一般化した問題と捕らえることができ，このことから MBP の $\mathcal{NP}$-困難性も導くことができる．

MBP の研究としては，1996年に片岡，徳永ら [1] が定式化を行い，分枝限定法によるアルゴリズムを開発しているものの，緩和問題を解いて得られる下界値が非常に低いため，教官数4，学生数10（図1の例題程度）でも10000〜20000秒もの計算時間を要している．したがって，全順列探索よりも効率的な最適解法はまだ知られておらず，全順列探索では学生数13程度以上の問題を解決できる見通しが暗い．

本研究では，動的計画法 (DP) による解法を提案する．この解法では，状態数を記憶するために2の学生数乗程度のメモリを必要とはするが，教官数の影響は少なく，計算機実験では学生数が25程度の問題の最適解を得ることに成功している．

2　動的計画による解法

ここでの DP 解法は，TSP の DP 解法 [2] を元にしているが，先に触れたように，MBP では拘束時間の増加量が，先に並べられた学生に依存して決まることに注意を要する．

学生全体の集合を N とし，N の部分集合を S $(\subseteq N)$ とする．このとき，S に属する学生を並べた直後に，学生 s を割り当てることを考える．次を定義する．

$f(S)$: S に属する学生を1から $|S|$ 時限に割り当て，$N \setminus S$ の学生が後に控えているとき，$|S|$ 時限目までにおける教官の総最小拘束時間．

$d_k(S,s)$: S に属する学生を1から $|S|$ 時限に，学生 s を $|S|+1$ 時限目に割り当て，$N \setminus \{S \cup \{s\}\}$ の学生が後に控えているとき，教官 k が $|S|+1$ 時限

目に拘束されるとき 1，そうでないとき 0 をの値をとる関数．

関数 $d_k(S,s)$ は，教官 k の担当する学生数を p_k とするとき，次のように算出できる．

$$d_k(S,s)=\begin{cases}\text{教官 } k \text{ が，} S \text{ の中の担当する学生数} < p_k \\ \text{かつ } N\setminus\{S\cup\{s\}\} \text{ の中の担当する学生} \\ \text{数} < p_k \text{ のとき } 1, \\ \text{教官 } k \text{ が，} S \text{ の中の担当する学生数} = p_k \\ \text{または } N\setminus\{S\cup\{s\}\} \text{ の中の担当する学} \\ \text{生数} = p_k \text{ のとき } 0.\end{cases}$$

関数 $d_k(S,s)$ は，学生 s が発表しているとき，教官 k が拘束されるかどうかは，s の前または後のいずれか一方に全担当学生が集まっているときは 0，それ以外のときは 1 と解釈すればよい．

このとき，$f(S)$ には次の漸化式が成り立つ．

$$f(S)=\min_{s\in S}\left\{f(S\setminus\{s\})+\sum_k d_k(S\setminus\{s\},s)\right\}$$

境界条件として $f(\emptyset)=0$ であり，求めたい最適値は $f(N)$ である．$f(N)$ を求めるためには，メモリとして $2^{|N|}$ 必要になるが，教官の数には依存しない．教官の数は，拘束時間の増加分 $d_k(S,s)$ を求める際に線形的に増加する程度で，計算効率の大きな妨げにはならない．

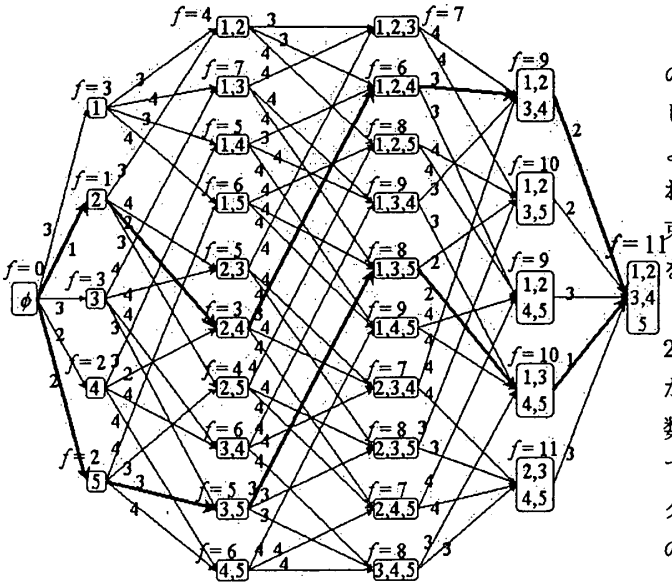

図 3: 図 1 の学生 1,2,3,4,5 のみで問題を考えた場合

図 3 に，例題 (図 1) における学生 1,2,3,4,5 のみで構成される問題として状態と状態間の拘束時間増加を示す．最適解は，この図において，状態 $\emptyset$ から状態 $\{1,2,3,4,5\}$ までの最短距離を求めることに他ならない．また，状態の要素数毎に階層的になっているので，効率よく DP 解法を開発することができる．

3 計算機実験および結果

計算機実験は，学生数 n を 15, 20, 25 の 3 通り，教官数をすべて 10 に固定して行った．また，各教官ごとに担当する学生の割合が 25, 50, 75 %の 3 種類について比較も行った．プログラムはC言語 (Visual C++6.0 コンパイラ使用) で記述し，IBM 300PL, PentiumII(400MHz) 上で実行した．

表 1 にまとめた実験結果では CPU 時間と，初期状態からの拘束時間の改善率をカッコ内に示している．各数値は 10 回の試行の平均であるが，学生数が 25 の場合のみ 1 度だけの試行結果である．

表 1: DP による結果 (CPU 時間 (秒) (初期解からの改善率))

	担当する学生の割合 (= 1 の密度)		
n	0.25	0.50	0.75
15	2.00(0.58)	1.40(0.78)	0.81(0.87)
20	66.71(0.61)	38.78(0.77)	19.68(0.90)
25	2825.86(0.58)	1355.28(0.78)	794.99(0.90)

表 1 より，開発した DP アルゴリズムにより，従来の結果 [1] よりは圧倒的に高速に問題を解くことに成功している．また，担当する学生の割合が多いほど効率よく解けていることもわかる．これは，改善率を見てわかるように，1 の密度が高いと順序の違いによる拘束時間の改善が少なく，各状態 S における最適値 $f(S)$ を求める更新回数も少なくて済むからだと思われる．

しかし，本 DP アルゴリズムの性格上，実行時間が 2 の学生数乗に比例する割合で増加していることもわかる．多少の工夫をほどこすことにより，実行効率を数倍改善することは可能であろうが，指数乗に比例している限り，学生数 25 よりも大きな問題をドラマチックに高速に解くことは困難であり，さらに大きな規模の問題を扱っていくことは，今後の課題である．

参考文献

[1] 片岡, 徳永: 最小滞在時間問題 (1),(2). 1996 年度 OR 学会春季研究発表会, pp.58-61.

[2] 久保, 松井: 組合せ最適化短編集. 朝倉書店 (1999).