同時割当問題の近似解法と厳密解法*

02401640　防衛大学校情報工学科　那須靖司　NASU Yasushi
01700900　防衛大学校情報工学科　山田武夫†　YAMADA Takeo

1　はじめに

I人の学生をJ学科, K要員に同時に配分する問題を考える. ここで学科と要員にはそれぞれ定員b_j, c_kがあり, これらは $\sum_{j=1}^{J} b_j = \sum_{k=1}^{K} c_k = I$ を満たしている. また, 学生 i を学科 j, 要員 kに配分したときの不満足度をp_{ijk}で表す. このとき, 不満足度の総和を最小にする配分を求める問題を**同時割当問題** (**SAP: simultaneous assignment problem**) と呼ぶ. 学生 i を学科 j, 要員 kに割り当てたとき 1, そうでないとき 0 となる**決定変数** x_{ijk}を定義すると, SAP は次のように定式化される.

SAP:

$$\text{Min} \quad \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} p_{ijk}x_{ijk} \qquad (1)$$

$$\text{s. t.} \quad \sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} = 1, \quad \forall i \qquad (2)$$

$$\sum_{i=1}^{I}\sum_{k=1}^{K} x_{ijk} = b_j, \quad \forall j \qquad (3)$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ijk} = c_k, \quad \forall k \qquad (4)$$

$$x_{ijk} \in \{0,1\}, \quad \forall i,j,k \qquad (5)$$

上で要員を考慮に入れずに, 学科への配分のみを考える場合は**割当問題**として知られており, 多項式時間の解法も発表されている [1] が, SAP はこれを拡張したもので, $\mathcal{NP}$困難であることが証明できる.

2　連続緩和による下界値

SAP の 0-1 条件 (5) を非負条件に置き換えた**連続緩和問題を C(SAP)**と記し, その最適値を $\underline{z}$ とすると, これは SAP の下界値を与える. C(SAP) を改訂単体法により解いたところ, **退化**が頻繁に生じてかなり計算時間がかかることがわかった. そこで, 軸演算に部分プライシングや巡回プライシングの技法 [4] を取り入れ, また制約式が極めて疎な 0-1 行列である事実を積極的に利用するようプログラムを工夫したところ, 例えば $I = 100, J = 4, K = 3$ の問題の計算時間が 237.8 秒から 8.3 秒にまで減少した (HP9000 使用).

このようにして得られる C(SAP) の最適解は, 一般に 0-1 条件 (5) を満たさないが, ほとんどの例題で 90 数% の変数が 0 または 1 となり, それ以外の分数値をとるものは数%にすぎないことがわかった.

3　局所探索法による近似解法

局所探索法により, SAP の近似解を求めることを考える. ある実行可能解 x を少し変形して得られる解の集合を x の近傍と呼び, $N(x)$ と記す. 局所探索法ではまず初期解 x を求め, その解の近傍の中でより良い解 y があれば y を x と改めて上の操作を反復する. そこで, 3.1 項では初期解の求め方を述べ, 3.2 項と 3.3 項では 2-opt 近傍と負サイクル近傍の 2 種類の近傍概念に基づく局所探索法を与え, 3.4 項以降ではそれらの改良を検討する.

3.1　初期解

最初に SAP の実行可能解を次のようにして求める. p_{ijk}の小さい順に $\{ijk\}$ を調べ, 学生 i が未配分で学科 j, 要員 kの定員が満杯でなければ学生 i を学科 j, 要員 kに割り当てる. これをすべての学生が割り当てられるまで反復することにより, 初期解 x が得られる.

3.2　2-opt 法

実行可能解 x において, ある学生の対について, それら 2 人の (i) 学科の入れ換え, (ii) 要員の入れ換え, または (iii) 学科と要員の入れ換え, を施して得られる解全体の集合を x の **2-opt 近傍**と呼び, $N_{2\text{-opt}}(x)$ と記す. この近傍概念に基づいて構成した局所探索法を **2-opt 法**と呼ぶ.

3.3　負サイクル法

実行可能解 x において, 何人かの学生の集合について, その学科, 要員の割当を 1 人ずつずらして得られるような解全体の集合を x の**負サイクル近傍**と呼び, $N_{\text{neg-cycle}}(x)$ と記す. この近傍概念に基づいて構成した局所探索法を**負サイクル法**と呼ぶ.

3.4　局所探索法の高速化

2-opt 法では, 実行可能解 x に対して, より良い解 $y \in N(x)$ を求めるのに, すべての学生の対について, 学科と (または) 要員の入れ換えを調べていた. また負サイクル法においても I人の学生を節点とする完全グラフ上で負サイクルを検出していた.

これに対して, 学科×要員に対応した節点集合を持つ補助グラフを導入し, 近傍の探索をこの補助グラフの節点について行うと, 計算時間を大幅に減少させることができる.

3.5　複合的局所探索法

2-opt 法と負サイクル法を複合的に使用することを考える. このとき最初に $N_{2\text{-opt}}(x)$ を探し, そこで x より

*OR 学会春季研究発表会 (1998.5.27-28, 仙台市青年文化センター)
†E-mail: yamada@cs.nda.ac.jp

良い解 y が見つからなければ $N_{\text{neg-cycle}}(x)$ を探すという方式と、逆に $N_{\text{neg-cycle}}(x) \to N_{\text{2-opt}}(x)$ の順に探す方式が考えられる. 予備実験を行ったところ, 計算時間と目的関数値について前者の方がわずかに優っていることがわかったので、これを **LS 法**と呼び、以下では SAP に対する標準的な局所探索法として用いる.

3.6 連続緩和解の利用

第 2 節で見たように、SAP の連続緩和解は少数の分数成分を除きほとんどが 0-1 値であった. そこで連続緩和解の中で値が 1 であるものについては一応配分が確定したものとして、それらの学生とその分の定員を原問題から除くと問題が大幅に縮小される. このようにして得られた縮小問題について初期解を求め、さらに LS 法で改善する. 線形計画問題を解く際には, まず LS 法により初期実行可能解を求めるので, この方法を **LS-LP-LS** 法と呼ぶ. これは SAP の上界値 $\bar{z}$ を与える.

4 厳密解法

以上で得られた上下界値 $\bar{z}, \underline{z}$ を利用すると, 釘付けテスト [3] を適用して SAP の変数 $\{x_{ijk}\}$ のうちのいくつかを 0 または 1 に固定し、問題を縮小することができる. 原問題のサイズは $n := I \times J \times K$ 変数, $m := I + J + K$ 制約式であるが、これらのうち n_1 変数が 1 に, n_0 変数が 0 に固定されたとする. また、これら 0 または 1 に固定された変数のみで制約式 (2)-(4) のいくつかが満たされている場合、このような制約式は除去できる. このようにして, n' 変数, m' 制約式が残ったとすると, あとはこの縮小された 0-1 計画問題を解けば良い. この部分については本研究では分枝限定法に基づいて構成された FORTRAN プログラム mip.f [2] を用いている.

5 数値実験

改訂単体法, 局所探索法, および釘付けテストを行うプログラムを C 言語により作成し, mip.f とあわせて DEC3000 上で数値実験を行った. 実験は学科数, 要員数が $J = 10, K = 4$ の場合と $J = 20, K = 3$ の場合について, 学生数 I が 200 ~ 800 の範囲で計算時間と近似精度, 問題縮小度を計測した. p_{ijk} は [1, 1000] の範囲の一様乱数により定め, 確率 q で無限大に置き換えた. q としては $q = 0.0, 0.8$ の場合を計算した. $J = 10, K = 4$ の場合の実験結果を表 1, 表 2 に示すが, すべての数値は 20 回の独立試行の平均値である. これにより, 以下の知見を得た.

- LS-LP-LS 法は LP の部分に時間がかかるが, 精度の良い解が得られる. 問題のサイズが大きいほど, 近似解の精度は良くなる.
- 厳密解を得るにはまず近似解を求め, そこからさらに同程度の計算時間を要する.
- 計算時間と精度は確率 q によってはあまり変化しない.

表 1: 近似解法の結果

q	I	計算時間 (秒) LS_1	LP	LS_2	誤差 (%)
0.0	200	0.11	5.26	0.02	0.29
	400	0.26	33.60	0.06	0.11
	600	0.45	96.96	0.07	0.03
	800	0.66	215.92	0.11	0.03
0.8	200	0.10	4.93	0.02	0.49
	400	0.24	36.38	0.05	0.05
	600	0.43	94.28	0.05	0.03
	800	0.74	216.16	0.13	0.03

表 2: 厳密解法の結果

q	I	n'	m'	子問題数	計算時間 (秒)
0.0	200	348.3	129.8	7.7	15.05
	400	377.3	165.4	11.6	21.97
	600	318.1	151.1	9.2	30.52
	800	499.9	228.4	11.0	164.20
0.8	200	313.2	104.1	7.9	17.87
	400	263.0	132.2	15.4	13.45
	600	394.2	186.7	12.0	33.00
	800	424.3	198.9	11.6	49.48

6 まとめ

同時割当問題を定式化し, その近似解法と厳密解法を構築するとともに, 数値実験によりそれらの性能と特性を評価した.

参考文献

[1] R.K. Ahuja *et al.*, *NETWORK FLOWS: Theory, Algorithms, and Applications*, Prentice Hall(1993).

[2] 茨木俊秀, 福島雅夫, 「FORTRAN77 最適化プログラミング」, 岩波書店 (1991).

[3] 今野浩, 鈴木久敏, 「整数計画法と組合せ最適化」, 日科技連 (1982).

[4] 反町洋一, 「線形計画法の実際」, 産業図書 (1992).