

研究室配属のための一方式の提案とその数理的考察

片岡 達
日立製作所

茨木 俊秀
関西学院大学

(受理 2008年3月28日 ; 再受理 2008年7月4日)

和文概要 大学の卒業研究などで、学生をどの研究室に配属させるかを決定する問題が生じる。学生や研究室にはそれぞれ配属関係を構築したいと考える相手がいるが、様々な理由により研究室の配属人数は限られるため、全員の第1希望が実現するとは限らない。本論文では、学生と研究室双方の希望を考慮し、合理的に配属先を決定する方法について論じる。本方式では、まず学生側の希望を反映させた研究室の定員を定めた上で、学生と研究室の双方の希望を考慮した合理的な配属を実現させる。具体的には、安定結婚問題の概念を一般化させ、本問題に適した配属の安定性を定義し、明示された半順序と暗黙の全順序という2つの概念を定めた上で、合理的な配属を得る手法を提案する。さらに、この手法の計算量の解析および計算実験による確認を行う。

キーワード: 組合せ最適化, アルゴリズム, 研究室配属問題, 議員定数配分問題, 安定結婚問題

1. はじめに

世の中には、様々な形で配属関係というものが存在している。本研究では、特に具体例として、大学生が卒業研究のために研究室に配属される状況について考える。通常、大学生は4回生になると卒業研究を行うために、所属する学部・学科のいずれかの研究室に配属される。その際、学生は所望する研究室に配属されたいと考えるが、研究室側の希望も考慮されるべきである。また、研究室には収容人数という物理的な制約が存在するため、全ての学生を希望する研究室に配属させることが可能とは限らない。しかし、最終的に全ての学生はどこかの研究室に必ず所属しなければならない。そのため、例えば第1希望の研究室への配属が不可能としても、全体の不満がなるべく小さくなるような配属を考えることが求められる。本論文では、このような配属を求める問題を、研究室配属問題 (laboratory assignment problem, LAP) と呼ぶ。

現実の一例を挙げると、関西学院大学工学部情報科学科における研究室配属では、Webシステムを用い、各学生に第1希望から第5希望まで1つずつ希望研究室を選択させる形式を取っている。他にも、クラス編成問題 [5] のモデルを基に、希望調査を Web アプリケーションで行うという、研究室配属システムの開発に関する研究 [4] なども行われている。しかし、こうしたシステムでは学生側の希望のみが考慮され、基本的に研究室側の要望は無視されている。そこで本研究では、学生側だけでなく研究室側の希望も配属結果に反映されるような方法を考える。すなわち、どのような配属が望ましいかを数理的に定義した上で、学生と研究室両方の希望を考慮した合理的な配属を実現させることを目指す。

配属に当たって、研究室の物理的な制約や、学生に対する教育効果などを考慮すると、収容可能な学生の人数は自ずと限られてくるため、研究室の定員に上限値を設定するのは自

然である。一方、研究室が学生を担当する負担の均等化を考慮すると、どの研究室にもある程度の人数分の学生を配属させるべきであり、定員に下限値を設定することも妥当である。このように、学生数と研究室数、さらに学生の希望を考慮した上で、適切な研究室の定員を決定する必要がある。この点に関する話題としては、クラス編成問題 [5] があり、大学などで学生を予め用意されたクラス (講義) に配分するという問題についての提案がなされている。その解法として、各学生にそれぞれある一定の得点を与え、それを希望するクラスに配分させ、クラスが得た総得点に応じて定員を決めるなどのアイデアが採用されており、得点の配分方法や問題の定式化などについて様々な検討がなされている。また、議員定数配分問題も本研究と関連が強く、文献 [2][3] などにおいて公平な議員定数の決め方についての議論がなされている。本論文の提案方式では、第 1 ステップとして、議員定数配分問題のアイデアを基に、適切な研究室の定員を決定する (研究室の定員の上下限値ではなく、定員そのものを定める)。

研究室の定員が決まると、残る問題はどの学生をどの研究室に配属させるかである。異なる二者がそれぞれの思惑に基づいて利得を得ようとする問題は、ゲーム理論 [6][7] で詳しく研究されていて、とりわけ安定結婚問題 [1] は本研究と関連が強い。この問題は、同数の男女がいて、それぞれが異性に対して選好順序 (好ましさの全順序) を持っているとき、誰からも不満が生じないようなマッチング (男女対の集合) を求める問題である。なお、安定結婚問題はアメリカの研修医配属に端を発しており、実際の応用例 [8][9][10] も知られている。

提案方式では、次の 3 段階を経て配属を決定する。

- 第 1 段階: 学生からの希望調査
- 第 2 段階: 研究室定員の決定 (議員定数配分問題の利用)
- 第 3 段階: 配属学生の決定 (安定結婚問題の利用)

第 1 段階では、学生からの希望調査の負担を軽減するために、希望研究室の全順序を入力するのではなく、同点や省略を許すことで、意識されている部分のみの入力 (明示された半順序) を想定する。同様に、第 3 段階で用いられる各研究室による学生の順序付けについても、決定に必要な最小限の提示だけを要求する。その結果、安定結婚問題で論じられている安定性の概念 (選好の全順序に基づいている) をそのまま適用することはできないので、学生側と研究室側のどちらから見ても明示された半順序に関して不満がないことを示す強安定という概念を採用し、第 3 段階のアルゴリズムによって得られた解が強安定であることを証明する。さらに、明示された半順序の背後に存在する暗黙の全順序を想定することによって、第 3 段階 (現実には、配属会議で実行される) で得られる配属が、会議の中で学生を処理する順番に依らず一意的であることを保証している。以上が、本論文の提案方式の特徴であり、新しい点である。

本論文の構成は次の通りである。2 節では、学生側に課す希望調査の方式について述べる。3 節では、2 節の希望調査の結果を反映させた研究室の定員を決める方法について述べる。4 節では、2 節の希望調査の結果と 3 節で定める研究室定員に基づいて配属学生を合理的に決定する方法について述べる。5 節では、実際に著者の大学で行われた研究室配属のデータを参考にした設定で、本手法を適用した場合の結果を示す。最後に 6 節で結論を述べる。さらに、数理的性質の一部の証明を付録として与える。

2. 学生からの希望調査

学生 s が研究室 l よりも l' を希望するとき、 $l' \succ_s l$ (あるいは $l \prec_s l'$) と記し、また、 s が l あるいは l' を同程度に希望するとき、 $l =_s l'$ と記す。 l についても、学生に対する同様の関係 $\succ_l, \prec_l, =_l$ を定義する。

学生は、希望調査において各研究室の好ましさを順位を申し出る。その結果、各学生 s は研究室の相対的な希望順位を半順序として明示する (換言すると、任意の2研究室 l, l' に対し、 $l \succ_s l', l \prec_s l', l =_s l'$ のいずれかが成り立つ)。これを学生により明示された半順序と呼ぶ。

希望調査の実施方法は、学生にとって順位付けの自由度が高く、順位付け作業の負担が小さいシステムが望ましい。本研究では、Web 上で希望調査を実施するとし、直感的に理解しやすいシステムの構築を行った。すなわち、学生は、Web ページ上に用意されたラジオボタン形式により構成された $m \times m$ の表に (m は研究室数)、以下の3つの Rule に従い、各研究室に対して配属希望順位を入力する。

Rule 1: 学生は、希望順位に矛盾が生じない限り、第 k 希望 ($k = 1, 2, \dots, m-1$) を複数選択できる (例えば、第1希望を2つ選んだとすると、その次点は第3希望となる)。

Rule 2: 学生は、上位3つの研究室は必ず選択しなければならない。最上位の研究室は第1希望となるが、Rule 1 に従うため第2、第3希望が存在しない場合もある (例えば、第1希望を3つ選んだとすると、第2、第3希望は選べなくなる)。

Rule 3: 学生は、Rule 1, 2 に基づき、3~ m 箇所の研究室に対して順位付けを自由に行う。その際、順位付けされなかった全ての研究室については、未定順位のうち最上位を割当てる (以下で例示)。

図1に、研究室数 $m = 6$ の場合にある学生が入力を行った例を示す。表中の‘●’はその列の研究室を選択している状態を、‘○’は選択していない状態を意味する。この例では、Rule 3の適用により、入力されていない l_2, l_6 は共に未定最上位である第4希望と見なされる。

希望	l_1	l_2	l_3	l_4	l_5	l_6
第1希望	●	○	●	○	○	○
第2希望	○	○	○	○	○	○
第3希望	○	○	○	○	●	○
第4希望	○	○	○	○	○	○
第5希望	○	○	○	○	○	○
第6希望	○	○	○	●	○	○

図1: 希望調査の例

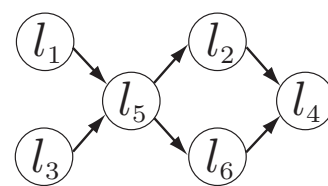


図2: 学生により明示された半順序の例

この入力に対して、学生により明示された半順序は図2のようになる。すなわち、研究室 l, l' について、図中に $l \rightarrow l'$ という関係が (推移的なものも含めて) 存在すれば $l \succ_s l'$ と見なし、 $l \rightarrow l'$ でも $l' \rightarrow l$ でもなければ $l =_s l'$ であると見なす。例えば、互いに有向辺が張られていない l_1 と l_3 に関しては $l_1 =_s l_3$ と書いて、共に s にとって同順位 (かつ最高位) と見なせる。また、推移的關係にある l_1 と l_4 に関しては $l_1 \succ_s l_5 \succ_s l_2 \succ_s l_4$ と書いて、 s にとって l_4 よりも l_1 の方が上位にある。

3. 研究室定員の決定

研究室の定員を決定する方法として、議員定数配分問題 (apportionment problem, AP)[2][3] において利用されている方式を採用する。

3.1. 議員定数配分問題

3.1.1. 基礎概念

AP とは、国政選挙などにおいて、異なる有権者数を持つ複数の選挙区に対して、有権者数にできるだけ比例するように議員定数を配分する問題である。

- 選挙区数: m
- 議員総数: n
- 選挙区 i の有権者数: $p_i, i = 1, 2, \dots, m$
- 選挙区 i の議員定数の理想配分値: $q_i = np_i / \sum_{i=1}^m p_i, i = 1, 2, \dots, m$
- 選挙区 i の定員下限値: $c_i^{LB}, i = 1, 2, \dots, m; c_i^{LB}$ は $1 \leq c_i^{LB} \leq [q_i], \sum_{i=1}^m c_i^{LB} < n$ を満たす定数
- 選挙区 i の議員定数: $c_i, i = 1, 2, \dots, m$

なお、定員下限値 c_i^{LB} とは選挙区 i が最低限確保すべき議員定数を意味し、理想配分値 q_i (実数値) とは選挙区 i の有権者数 p_i に正確に比例した議員定数を意味する。以後、 $n \geq m$ を仮定する。

決定される議員定数 c_i (整数値) は、

$$c_i \geq c_i^{LB}, i = 1, 2, \dots, m, \quad (1)$$

$$n = \sum_{i=1}^m c_i \quad (2)$$

を満たすことが要求される。さらに、少なくとも逆転現象が生じないこと、すなわち

$$c_i > c_{i'} \Rightarrow p_i \geq p_{i'}, \forall i, i' (i \neq i') \quad (3)$$

の成立が望ましい。

c_i を各選挙区に公平に配分するには、 $c_i = q_i$ の成立が望ましい。しかし、 c_i は整数値であり、 q_i は実数値であるため、一般にこれは不可能である。従って、 c_i と q_i の“ずれ”をどのように扱うかがこの問題のポイントである。

3.1.2. 議員定数配分法

議員定数の配分法として最初に用いられたのはハミルトン法である。これは、まず各選挙区 i に $c_i := [q_i]$ の定数を与え、そのあと小数部分 $q_i - [q_i]$ の大きい選挙区から順に式 (2) が成立するまで $c_i := c_i + 1$ と増加させるものである。しかし、ハミルトン法を用いる場合、アラバマ・パラドックスや人口パラドックスと呼ばれる、配分に関する明らかな矛盾が生じることが知られている [2][3]。その1つのアラバマ・パラドックスとは、 n を増加させた後ハミルトン法によって新しい配分を決定すると、定数がかえって減少する選挙区が生じる場合があるというものである。

これらの問題点を解決する方法の1つとして、次に示すハンティングトン法が提案された。この方式によれば、式 (1)~(3) は必ず成立し、上記2つのパラドックスも生じない。

[ハンティングトン法]

入力: 選挙区数 m , 議員総数 n , 有権者数 p_i , 定員下限値 $c_i^{LB} (i = 1, 2, \dots, m)$

出力: 議員定数 $c_i (i = 1, 2, \dots, m)$

Step 1: $c_i := c_i^{LB}$ ($i = 1, 2, \dots, m$) とする.

Step 2: $\sum_{i=1}^m c_i = n$ が成立すれば終了, さもないれば Step 3 へ.

Step 3: 順位関数 $rank(p_i, c_i)$ を最大にする $i = i^*$ を選び, $c_{i^*} := c_{i^*} + 1$ とする. Step 2 へ戻る.

順位関数 $rank(p_i, c_i)$ については様々な提案がなされているが ([2]などを参照), その1つに

$$rank(p_i, c_i) = \frac{p_i}{c_i + 0.5}, \quad i = 1, 2, \dots, m \quad (4)$$

があり, 後述 3.2.2 の定員決定アルゴリズムにおいてもこれを用いる.

3.2. 研究室定員決定への応用

APでのハンティングトン法のアイデアを, LAPに適用することを考える. すなわち, 議員を学生, 選挙区を研究室, 議員定数を研究室定員と捉える.

3.2.1. 研究室の人気度

学生からの希望調査では, 学生は第1希望だけを提出するわけではない. そのため, ハンティングトン法を適用する場合, 有権者数(すなわち, 研究室 l_i を希望する学生の人数) p_i を適切に設定する必要がある. ここでは, 学生の第1, 2, 3希望までを重要と見なし, 第3希望までに選ばれた研究室には, 以下のように傾斜得点を与えることにする.

まず, 全ての学生に合計 $A (> 0)$ 点ずつ持ち点を与え, それを研究室に配分する. すなわち,

- a_{jk} : 学生 s_j が第 k 希望とした研究室の数¹
- d_j : 学生 s_j が第1希望とした研究室全体に与える得点(変数)

を用いて, 学生 s_j が第 k 希望とした研究室1箇所を与える得点 b_{jk} を

$$b_{jk} = \begin{cases} \frac{d_j}{2^{k-1}} \cdot \frac{1}{a_{jk}}, & a_{jk} > 0 \text{ and } 1 \leq k \leq 3, \\ 0, & a_{jk} = 0 \text{ or } k \geq 4 \end{cases} \quad (5)$$

と定める². また, s_j の配分合計点が A であることから,

$$\sum_{k=1}^3 a_{jk} b_{jk} = A, \quad j = 1, 2, \dots, n \quad (6)$$

が成り立つように d_j を定める. 例えば, $a_{j1} = 2, a_{j2} = 0, a_{j3} = 1$ の場合は

$$\begin{aligned} \sum_{k=1}^3 a_{jk} b_{jk} &= a_{j1} b_{j1} + a_{j2} b_{j2} + a_{j3} b_{j3} \\ &= a_{j1} \cdot \left(\frac{d_j}{2^0} \cdot \frac{1}{a_{j1}} \right) + a_{j2} \cdot 0 + a_{j3} \cdot \left(\frac{d_j}{2^2} \cdot \frac{1}{a_{j3}} \right) \\ &= \frac{5}{4} d_j \\ &= A \end{aligned}$$

となり, $d_j = \frac{4}{5}A$ を用いて各 b_{jk} が決定される. これを含め, 第1~第3希望の全ての状況を網羅すると, ($A = 100$ と設定するとき) 表1のようになる.

¹例えば, $a_{j1} = 2, a_{j2} = 0, a_{j3} = 1, a_{jk} = 0 (k \geq 4)$ など.

² b_{jk} の決定要因である 2^{k-1} に関しては, 明確な根拠がある訳ではなく, 常識的な配分法の1つと判断し採用しているが, 他の係数も考えられる.

表 1: $A = 100$ の場合の a_{jk} と b_{jk} の関係

a_{j1}	a_{j2}	a_{j3}	d_j	b_{j1}	b_{j2}	b_{j3}
1	1	≥ 1	57	57	29	$\frac{14}{a_{j3}}$
1	≥ 2	0	67	67	$\frac{33}{a_{j2}}$	0
2	0	≥ 1	80	40	0	$\frac{20}{a_{j3}}$
≥ 3	0	0	100	$\frac{100}{a_{j1}}$	0	0

以上のように b_{jk} を定めた後, 学生 s_j が第 k 希望とした全ての研究室 l_i に対して得点 $e_{ij} = b_{jk}$ を与える. そして, ハンティングトン法での順位関数の引数として, 全ての学生からの得点と

$$p_i = \sum_{j=1}^n e_{ij}, \quad i = 1, 2, \dots, m \quad (7)$$

を採用する. この p_i を研究室 l_i の人気度と呼ぶ.

3.2.2. 定員決定アルゴリズム

ハンティングトン法を少し改変し, 研究室 l_i の定員の下限值 c_i^{LB} と上限値 c_i^{UB} を設定したのち, 定員を決定する. ただし,

$$\sum_{i=1}^m c_i^{LB} \leq n \leq \sum_{i=1}^m c_i^{UB} \quad (8)$$

を仮定する.

[定員決定アルゴリズム]

入力: 学生数 n , 研究室数 m , 人気度 p_i , 定員下限値 c_i^{LB} , 定員上限値 c_i^{UB} ($i = 1, 2, \dots, m$)

出力: 定員 c_i ($i = 1, 2, \dots, m$)

Step 1: $c_i := c_i^{LB}$ ($i = 1, 2, \dots, m$) とする.

Step 2: $\sum_{i=1}^m c_i = n$ が成立すれば終了, さもないと Step 3 へ.

Step 3: $c_i < c_i^{UB}$ を満たし, かつ順位関数 $rank(p_i, c_i)$ を最大にする $i = i^*$ を選び, $c_{i^*} := c_{i^*} + 1$ とする. Step 2 へ戻る.

上記のアルゴリズムを用いることにより, 次式が保証される.

$$c_i^{LB} \leq c_i \leq c_i^{UB}, \quad i = 1, 2, \dots, m, \quad (9)$$

$$n = \sum_{i=1}^m c_i. \quad (10)$$

4. 配属学生の決定

学生と研究室の配属関係を決定するために, 関連研究である安定結婚問題 (stable marriage problem, SMP)[1] についてまず説明し, その後 LAP への応用について述べる.

4.1. 安定結婚問題

4.1.1. 基礎概念

SMP とは, 直感的には, 男性と女性がそれぞれの異性に好みの順位を付けるときに, 誰からも不平が出ないような男女の組合せを求める問題といえる. ここで, 以後の議論に必要な基本的な用語や表記などをまとめる.

- 男性集合: M (男性数 $|M| = n$)
- 男性要素: $m_j \in M, j = 1, 2, \dots, n$ (単に m などと記す場合もある)
- 女性集合: W (女性数³ $|W| = n$)
- 女性要素: $w_i \in W, i = 1, 2, \dots, n$ (単に w などと記す場合もある)
- 選好順序: 全ての男性は全ての女性に対して, また全ての女性は全ての男性に対して, それぞれの好みの順番を全順序として持っているものとし, これを選好順序と呼ぶ. m の選好順序において w が w' よりも上位にある (つまり, m が w' より w を好む) とき, $w \succ_m w'$ あるいは $w' \prec_m w$ と記す. w についても同様の関係 \succ_w, \prec_w が定義される.
- ペア: (m, w) と表される組. ただし, $m \in M, w \in W$.
- マッチング: n 組のペアを要素として持つ集合で, \mathcal{M} と記す. 任意のマッチングにおいて, 同じ男性や女性が重複して出現することはない. 換言すると, 各男性および各女性は, マッチングにちょうど1回ずつ現れる.
- パートナー: $|\mathcal{M}'| \leq n$ を満たす部分マッチング \mathcal{M}' が得られているとき, $(m, w) \in \mathcal{M}'$ となる m と w は \mathcal{M}' において互いにパートナーであるとし, 次のように表記する.
 - \mathcal{M}' における m のパートナーが $w \Leftrightarrow w = p_{\mathcal{M}'}(m)$
 - \mathcal{M}' における w のパートナーが $m \Leftrightarrow m = p_{\mathcal{M}'}(w)$
 このとき, m と w は婚約していると思なし, 婚約していない要素は独身であるとする.
- 求婚: 与えられた男女集合と選好順序に対し, マッチングを構成するための操作. 求婚はどちらか片方の要素のみが行うものとし, 相手が独身であれば即座に受理され, 相手が婚約していれば相手の選好順序次第で受理されるか拒否されるかが決まる. 以後, 特に断りのない限り, 求婚者は男性と仮定する.
- 不安定ペア: $(m, w) \notin \mathcal{M}, w \succ_m p_{\mathcal{M}}(m), m \succ_w p_{\mathcal{M}}(w)$ が成り立つとき, (m, w) を \mathcal{M} に関する不安定ペアと呼ぶ.
- 安定性: \mathcal{M} に関する不安定ペアが1つも存在しないとき, \mathcal{M} を安定マッチングと呼ぶ. 逆に, 不安定ペアが1つでも存在するときの \mathcal{M} は不安定マッチングと呼ぶ.

SMP とは, 与えられた男女集合と選好順序に対して安定マッチングを求める問題である. ただし, 1つの問題例に対して存在する安定マッチングの数は1つとは限らない(後述).

ここで, 適当な問題例を用い, SMP に対する理解を深める. 以後, SMP (および LAP) に関する全ての図において, 円とその中に書かれた文字で集合の要素を, 実線でマッチングが含むペアを, 破線で不安定ペアを, 丸括弧 (...) で選好順序を表現する.

図3には, 男女数がそれぞれ4である場合の問題例を示してある. $m_1, \dots, m_4 \in M$ は男性を, $w_1, \dots, w_4 \in W$ は女性を表している. 例えば男性 m_1 は選好順序 (w_2, w_4, w_1, w_3) を所持しているが, これは m_1 にとっては w_2 が最も好みの女性であり, 残りも w_4, w_1, w_3 の順に好みであるということの意味しており, $w_2 \succ_{m_1} w_4 \succ_{m_1} w_1 \succ_{m_1} w_3$ と表せる. また, 図3にはマッチング

$$\mathcal{M} = \{(m_1, w_1), (m_2, w_3), (m_3, w_2), (m_4, w_4)\}$$

が示されていて, 例えば m_1 のパートナーは w_1 であり $w_1 = p_{\mathcal{M}}(m_1)$ と書け, w_2 のパートナーは m_3 であり $m_3 = p_{\mathcal{M}}(w_2)$ と書ける. ここで, 破線のペア $(m_1, w_4) \notin \mathcal{M}$ を考えると,

³一般には $|M| \neq |W|$ の場合も考えられるが, 本論文では $|M| = |W| = n$ とする.

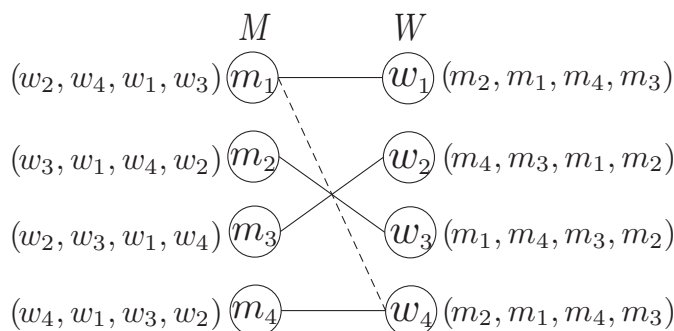


図 3: SMP の問題例と不安定マッチング

m_1 は現在のパートナー w_1 よりも w_4 を好み、 w_4 はパートナー m_4 よりも m_1 を好んでいる。従って、仮にこの M が解として得られたとすると、それを見て m_1, w_4 はそれぞれ自らのパートナーとの婚約を解消し、新たなペア (m_1, w_4) を形成したいと考えるのが自然である。この意味で、破線で描かれた (m_1, w_4) は M に関する不安定ペアとなり、 M は不安定マッチングであると判断される。

4.1.2. Gale-Shapley アルゴリズム

次に示す Gale-Shapley (GS) アルゴリズムは、与えられた男女集合と選好順序を入力として、安定マッチングを出力する。

[GS アルゴリズム]

入力: 男性集合 M , 女性集合 W , 各選好順序

出力: 安定マッチング M

Step 1: 全ての男性および女性を独身とし、任意の男性 m^* を選んで $m := m^*$ とする。

Step 2: w を、 m がまだ求婚を拒否されていない女性の中で、最大の選好順序を持つ女性とする。 w が独身なら m と w を婚約させて Step 4 へ、さもなければ Step 3 へ。

Step 3: w の現在のパートナー m' に対して $m \succ_w m'$ が成り立つならば m' を独身に戻し、 m と w を婚約させて Step 4 へ、 $m \prec_w m'$ ならばこの求婚を拒否させて Step 2 へ。

Step 4: 全ての男性が婚約しているならば婚約している全てのペアを要素として持つ安定マッチング M を出力して終了、さもなければ婚約していない男性 m^* を任意に選んで $m := m^*$ と更新して Step 2 へ。

以上が一般的な GS アルゴリズムの枠組みを示したものであるが、上記のように求婚者が男性である GS アルゴリズムを、男性指向 GS (man-oriented GS, MGS) アルゴリズムと呼ぶ。求婚者が女性である女性指向 GS (woman-oriented GS, WGS) アルゴリズムも可能である。一般に、同じ問題例に対して、MGS アルゴリズムにより得られる安定マッチングと WGS アルゴリズムにより得られる安定マッチングは異なる。

上記の MGS アルゴリズムについて、以下の各定理が知られている [1]。なお、定理 4.2 の求婚順序とは、MGS アルゴリズムの Step 1 と Step 4 における m^* の選び方を意味する。また、MGS アルゴリズムにより得た安定マッチングにおける各要素のパートナーを MGS パートナーと呼ぶ。

定理 4.1 MGS アルゴリズムの計算量は $O(n^2)$ である。また、 $\Omega(n^2)$ の計算量を要する問題例が存在する。

ここでの計算量とは、MGS アルゴリズムでの求婚回数を意味する。MGS アルゴリズムでは1回の求婚に必要となる演算(四則演算など)の回数は $O(1)$ 時間で済むため、コンピュータ上の計算で必要となるステップ数と見なすこともできる。上界については、男性は同じ女性に二度求婚を行うことはないことから、男性 n 人が高々 n 人の女性に求婚を行えば MGS アルゴリズムは必ず終了するため、 $O(n^2)$ と評価できる。下界については、そのような問題例の構成方法が存在する [1]。

定理 4.2 MGS アルゴリズムが生成する安定マッチングは、任意の求婚順序に対して一意的であり、あらゆる安定マッチングの中で男性最適である(すなわち、どの男性にとっても、そのパートナーより良いパートナーを持つ安定マッチングは存在しない)。

4.2. 配属学生決定への応用

4.2.1. 基礎概念

前項で示した SMP の概念を用い、学生集合 S と研究室集合 L に対する安定マッチングを生成することを目指す。ここで、以後の議論に必要な表記を次のようにまとめる。

- 学生集合: S (学生数 $|S| = n$)
- 学生要素: $s_j \in S, j = 1, 2, \dots, n$ (単に s などと記す場合もある)
- 研究室集合: L (研究室数 $|L| = m$)
- 研究室要素: $l_i \in L, i = 1, 2, \dots, m$ (単に l などと記す場合もある)
- 研究室 l_i の人気度: $p_i, i = 1, 2, \dots, m$ (3.2 項で定義した概念)
- 研究室 l_i の定員: $c_i, i = 1, 2, \dots, m$ (3.2 項の方法で決定)

LAP の問題例およびマッチングの例は、図 4 に示したような形で表される。各研究室 $l_i (i = 1, 2, 3)$ の右肩の値は定員 c_i を表す。また、各要素の横にある (\dots) は選好順序(半順序)を表すが、詳しくは後述する。

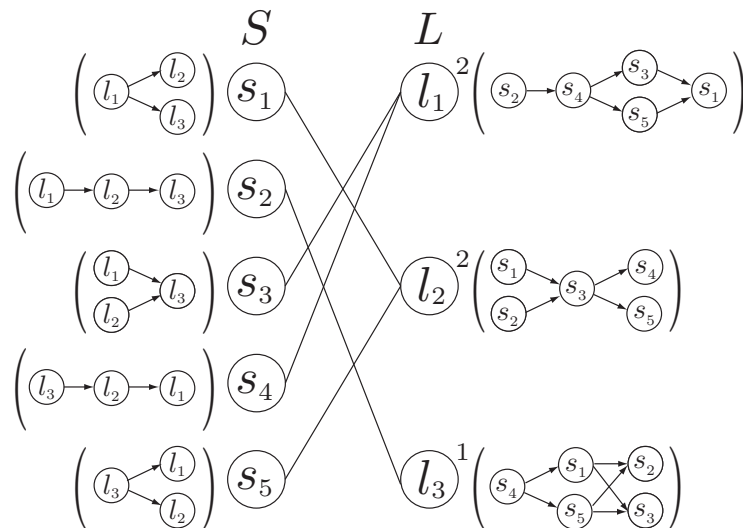


図 4: LAP の問題例 (1) とマッチング

LAP の特徴としては、

- $|S| = n \geq |L| = m$

- 研究室には定員分の (つまり複数の) 学生が配属されるため, マッチングは多対1構造
- 各要素が持つ選好順序は半順序

などが挙げられ, これらの点でLAPとSMPは大きく異なる. そこで, SMPでの安定性などの概念をLAPの構造に適したものに拡張し, GSアルゴリズムをうまく適用させることを考える.

4.2.2. SMPにおける安定性

SMPは, 選好順序が全順序であることを前提としているが, LAPでは半順序である. 文献[1]には選好順序が半順序の(すなわち相手への順位付けに同点を認める)場合について, 超安定性, 強安定性, 弱安定性という3種類の安定性が提唱されている. これらをLAPの設定(男性を学生, 女性を研究室と見なし, 学生数 $n =$ 研究室数 m を想定. 本来 $n \geq m$ を考えるべき点については後述)に合わせて書くと, 表2~4のようにまとめられる. すなわち, $(s, l) \notin \mathcal{M}$ のペアで, 各安定性の定義の下で不安定と判断されるものが存在しないことが要求される. 表2~4より, 超安定性, 強安定性, 弱安定性の順に, 不安定ペアが存在するための条件が狭まっていく. 従って, 超安定ならば強安定であり, 強安定ならば弱安定である. 弱安定性は4.1項で示した安定性と同じ概念である.

表 2: 超安定性

	$l \succeq_s p_{\mathcal{M}}(s)$
$s \succeq_l p_{\mathcal{M}}(l)$	(s, l) は不安定ペア

表 3: 強安定性

	$l \succ_s p_{\mathcal{M}}(s)$	$l =_s p_{\mathcal{M}}(s)$
$s \succ_l p_{\mathcal{M}}(l)$	(s, l) は不安定ペア	(s, l) は不安定ペア
$s =_l p_{\mathcal{M}}(l)$	(s, l) は不安定ペア	

表 4: 弱安定性

	$l \succ_s p_{\mathcal{M}}(s)$
$s \succ_l p_{\mathcal{M}}(l)$	(s, l) は不安定ペア

超安定性や強安定性を基準として用いる場合, 安定解が存在しないことがある. 例えば, 図5には同じ問題例(選好順序は半順序)に対する異なるマッチングを示してあり, この例にはこの2つ以外にマッチングは存在しない. これらを強安定性の観点から見ると, どちらのマッチングにも破線で示された不安定ペアが存在する. すなわち, この問題例には強安定な解は存在しない. より制約の強い超安定性を採用すると, 実行可能性はさらに低くなる.

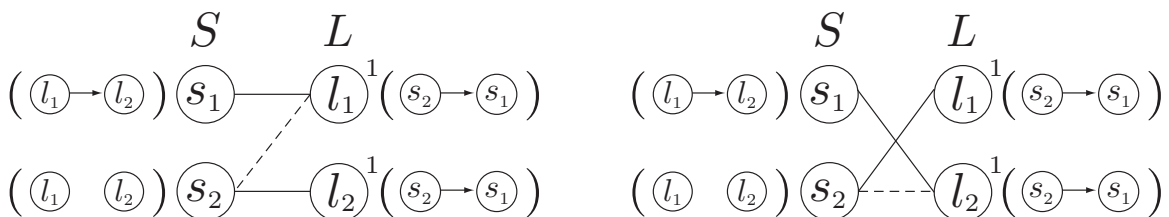


図 5: 強安定マッチングが存在しない問題例

4.2.3. LAPにおける安定性

$n \geq m$ を考慮して、マッチング、パートナー、不安定ペア、安定性などの定義をLAP仕様で再定義する。また、SMPでの求婚をLAPでは申出と呼ぶ。

- マッチング: n 組のペアを要素として持つ集合で、 \mathcal{M} と記す。任意のマッチングにおいて、学生はそれぞれ1回ずつ、研究室はそれぞれ c_i ($i = 1, 2, \dots, m$) 回ずつ出現する。
- パートナー: $|\mathcal{M}'| \leq n$ を満たす部分マッチング \mathcal{M}' が得られているとき、 $(s, l) \in \mathcal{M}'$ である s と l は \mathcal{M}' において互いにパートナーであるとし、次のように表記する。

– \mathcal{M}' における s のパートナーが $l \Leftrightarrow l = p_{\mathcal{M}'}(s)$

– \mathcal{M}' における l のパートナーが $s \Leftrightarrow s \in p_{\mathcal{M}'}(l)$

このとき、 s と l は仮配属関係にあるとし、仮配属状態でない要素は未配属状態であるとする。なお、研究室のパートナーは1人とは限らないため、 $p_{\mathcal{M}'}(l)$ は集合として扱う。

- 不安定ペア: 次の表5に示すように、マッチング \mathcal{M} と $(s, l) \notin \mathcal{M}$ に対し、4種類の不安定性を定義する。SMPの場合と比べると、厳密不安定ペアは表4の弱安定性に、同点不安定ペアは表2の超安定性に、学生不安定ペアと研究室不安定ペアを併せると表3の強安定性に対応している。なお、 $s \succ_l \exists s' \in p_{\mathcal{M}}(l)$ とは、複数存在する l のパートナー $p_{\mathcal{M}}(l)$ のうち、少なくとも1人の学生 (s' とする) について、 $s \succ_l s'$ が成り立つことを意味する。 $s \succeq_l \exists s' \in p_{\mathcal{M}}(l)$ についても同様に考える。

表 5: 不安定ペア (s, l) の分類

	$l \succ_s p_{\mathcal{M}}(s)$	$l \succeq_s p_{\mathcal{M}}(s)$
$s \succ_l \exists s' \in p_{\mathcal{M}}(l)$	(s, l) は厳密不安定ペア	(s, l) は研究室不安定ペア
$s \succeq_l \exists s' \in p_{\mathcal{M}}(l)$	(s, l) は学生不安定ペア	(s, l) は同点不安定ペア

これらの各不安定性の定義に基づき、LAPにおける5つの安定性を定義する。

- 超安定性: 同点不安定ペアを持たない。
- 強安定性: 学生および研究室不安定ペアを持たない。
- 学生安定性: 学生不安定ペアを持たない。
- 研究室安定性: 研究室不安定ペアを持たない。
- 弱安定性: 厳密不安定ペアを持たない。

上記の定義より、超安定ならば強安定であり、強安定ならば学生、研究室安定であり、学生あるいは研究室安定ならば弱安定である。

4.2.4. 学生により明示された半順序と暗黙の全順序

2節において学生により明示された半順序が与えられたが、その半順序における同順位の部分がある基準(例えば研究室 l_i の人気度 p_i)に基づいて並べることにより、元の半順序に矛盾しない全順序を構成する。これを、その学生が持つ暗黙の全順序と考える。なお、暗黙の全順序は学生が提示するわけではなく、明示された半順序に対してトポロジカルソート⁴を行うことにより得られる。同順位に対して適用される基準としては、研究室側の事情を考慮したもの(人気度以外にも、例えば担当教員の年齢順、学生獲得要求度順など)、あるいはラ

⁴一般には、非巡回有向グラフにおいて、辺の両端点の番号が辺の向きの順に出てくる(すなわち、点番号の大小関係と辺の向きが一致する)並べ方を見つけ出す方法。半順序に矛盾しない全順序を生成する。

ンダムであってもよい。以下に示すのは、各学生 s について明示された半順序から、研究室の人気度に基づいて暗黙の全順序を生成するアルゴリズムである。

[暗黙の全順序を生成するアルゴリズム]

入力: 学生 s により明示された半順序, 研究室 l_i の人気度 p_i ($i = 1, 2, \dots, m$)

出力: 学生 s が持つ暗黙の全順序 P

Step 1: $P := ()$ とする。

Step 2: 学生 s の半順序において, P 以外の研究室のうち最上位に位置するものの集合を L^* と定める。このとき, 全ての $l_i \in L^*$ を p_i の降順にソートした順列を P^* とし, $P := (P, P^*)$ と更新して Step 3 へ。

Step 3: P が全研究室を含むならば終了, さもなければ Step 2 へ。

図 6 に, 明示された半順序から暗黙の全順序を生成する簡単な例を示す。その際, 人気度について $p_2 > p_3$ を仮定する。

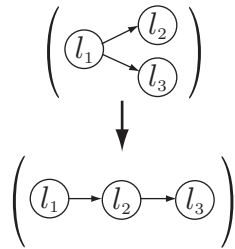


図 6: 学生が持つ暗黙の全順序の生成

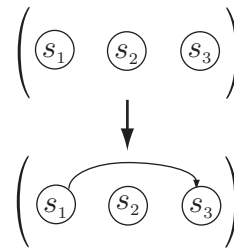


図 7: 研究室により明示された半順序の更新

4.2.5. 研究室により明示された半順序と暗黙の全順序

各研究室は, それぞれの思惑に基づき各学生を一行に並べた全順序 (例えば成績順) を所持していて, これを研究室が持つ暗黙の全順序と呼ぶ。しかし, あらかじめそれを開示する訳ではなく, 後述するアルゴリズムの中で必要に応じてその一部を明らかにする。すなわち, 1つの研究室に対して, 複数の学生について指定された人数だけ配属させたい学生を選ぶことが要求されると, 暗黙の全順序に従って回答を与える。その結果, 全順序の一部が提示され, 明示された半順序が得られる。図 7 に例を示す。今, ある研究室が s_1 と s_3 のどちらを配属させるかを聞かれて, s_1 を配属させたいとする。図の上側が初期段階の半順序 (そもそも研究室には希望調査を課さないため, 初期段階では全ての研究室は全ての学生を同順位と見なすとする), 下側が全順序の一部を明らかにさせた半順序である。

ここで, 学生と研究室それぞれについて明示された半順序と暗黙の全順序を区別するのは, どちらも最低限の情報提示 (半順序) によって暗黙の全順序に沿った解を得ることができるからである。すなわち, 学生も研究室も全順序についての全ての情報提示を行う必要はないことから, 希望順位の提出作業の負担を軽減することができる。

4.2.6. 配属決定アルゴリズム

明示された半順序と暗黙の全順序に基づき, 配属決定アルゴリズムを以下のように構成する。

[配属決定アルゴリズム]

入力: 学生集合 S , 研究室集合 L , 研究室定員 c_i ($i = 1, 2, \dots, m$), 全学生と全研究室の暗黙の全順序

出力: マッチング M (この M が持つ性質については 4.2.7 で示す)

Step 1: 全ての学生を未配属状態とし, $t := 1$ とする.

Step 2: 未配属状態の 全ての学生 はそれぞれ, まだ拒否されていない研究室の中で, 暗黙の全順序において最上位の研究室へ申出を行う. このとき, (仮配属数 + 申出数) が定員を超過する研究室が存在するならば, その 全ての研究室 について, 暗黙の全順序に基づいて (仮配属および申出学生の中から) 定員分の学生を選んで仮配属させ, それ以外の学生からの申出を全て拒否する. 拒否された学生は未配属状態に戻る.

Step 3: 全ての研究室が定員条件を満たせば仮配属関係を M として出力して終了, さもなければ $t := t + 1$ として Step 2 へ.

上記アルゴリズムにおける Step 2 と Step 3 の 1 反復を配属調整処理, t を配属調整回数と呼び, アルゴリズムが終了したときの t を総配属調整回数 T と定める. T の増大は配属会議の長期化を意味するため, 実用的には T をできる限り小さく抑えることが望まれる.

アルゴリズム中に示した 2 つの下線部, すなわち処理順序に関しては, 次の 3 つが考えられる (上記アルゴリズムは処理順序 3).

[処理順序]

- 処理順序 1: 1 学生を処理した後, 1 研究室を処理
- 処理順序 2: 全学生を処理した後, 1 研究室を処理
- 処理順序 3: 全学生を処理した後, 全研究室を処理

以後の議論は, 特に断りのない限り, 上記のいずれの処理順序にも共通する.

4.2.7. 配属解の性質

配属決定アルゴリズムが生成する解について, 一意性と安定性に関する以下の 2 つの定理が成り立つ. なお, 解が一意的であるとは, 配属会議におけるアルゴリズムの進め方 (つまり, 学生の申出の順番) に依存しない解が出力されることを意味する.

定理 4.3 配属決定アルゴリズムは, 学生および研究室が持つ暗黙の全順序によって定まる一意解 (学生主導の GS アルゴリズムが生成する安定マッチング) を生成する.

証明 まず, 学生集合 S と研究室集合 L については一般に $n \geq m$ が成り立つが, 各研究室を定員数に多重化した研究室集合を L' とすると, 式 (10) によって $n = |L'|$ が成り立つ. 次に, 学生と研究室は, それぞれ暗黙の全順序を持つ. 以上から, LAP は SMP と見なすことができる. 配属決定アルゴリズムはこれらの全順序に従って進められるので, 学生側を 4.1 項の男性側と考えると, MGS アルゴリズムを適用した場合と同一であることが判る. すなわち, MGS アルゴリズムを適用することにより男性主導の一意的な解を出力できるという, 定理 4.2 を適用できる. □

定理 4.4 配属決定アルゴリズムは, 学生および研究室により明示された半順序に関して強安定な解を生成する.

証明 アルゴリズムを適用して得られた解を M とする. 以下, M が強安定であること, すなわち任意の $(s, l) \notin M$ が厳密, 学生, 研究室不安定ペアでないことを示す.

- (i) $l \succ_s p_M(s)$ が成り立つ場合: s の暗黙の全順序は半順序 \succ_s に矛盾しないので, s は $p_M(s)$ よりも先に l に申出を行う. s の最終配属先が $p_M(s) (\neq l)$ であるということは, s から l

への申出が拒否されたことを示している(そのとき, l の仮配属および申出学生の中に, \succ_l に関して s より上位の者が l の定員数以上存在した). このこととアルゴリズムのその後の動作を考慮すると, l の持つ半順序において $s \prec_l \forall s' \in p_M(l)$ が成立する. 従って, (s, l) は厳密, 学生, 研究室, 同点不安定ペアにならない.

- (ii) $l =_s p_M(s)$ が成り立つ場合: s が $p_M(s)$ よりも先に l に申出を行う場合は, (i)と同様の議論により $s \prec_l \forall s' \in p_M(l)$ の成立が言えて, (s, l) は厳密, 学生, 研究室, 同点不安定ペアにならない. 一方, s が l よりも先に $p_M(s)$ に申出を行う場合については, 最終的に s は l に一度も申出を行わないことになる. すなわち, l が暗黙の全順序における s の順位を明らかにすることはないので, l において s は他の全ての学生 s' と同順位のまま残り, $s =_l \forall s' \in p_M(l)$ である. 従って, (s, l) は厳密, 学生, 研究室不安定ペアにならない.
- (iii) $l \prec_s p_M(s)$ が成り立つ場合: 各不安定ペアの定義より, (s, l) は厳密, 学生, 研究室, 同点不安定ペアにならない.

以上 (i), (ii), (iii) から, 定理は示された. \square

4.2.6の配属決定アルゴリズムは, 定理4.3の“学生の申出の順番に依存せずに安定マッチングを一意に生成する”という性質を利用して, 実用的な意味での計算時間の短縮を図っている. すなわち, 配属決定アルゴリズムは研究室による配属会議において実行されるが, この会議がいつ終了するかは T に依存する. 配属決定アルゴリズムでは, 学生側の申出を全て終えてから全研究室の判断を求めるという処理順序3を採用することにより, T を小さく抑え, 配属会議をできる限り早期に終了させることを狙っている.

4.2.8. 配属会議による配属学生の決定

LAPでは, 配属決定アルゴリズムを適用するために, 全ての研究室の教員を1箇所に集めて配属会議を開く必要がある. そこに配属決定アルゴリズムを実装したコンピュータを1台用意し, コンピュータによる申出処理と, 教員による受入れ判断処理の反復を行わせることになる.

配属会議において, 配属が決定するまでの流れを以下に示す. 内容は上記の配属決定アルゴリズムとほぼ同様であるが, やや教員の視点に立った記述となっている. なお, ここでも4.2.6で示した処理順序3を用いている.

[配属会議の手順]

Step 1: 学生集合, 研究室集合, 研究室定員, 各選好順序などのデータをコンピュータに入力しておき, コンピュータに申出処理を行わせる.

Step 2: 出力された結果を教員が確認. このとき仮配属数と申出数の和が定員を超過している全ての研究室は, 仮配属および申出学生の中から, 受入れても良いと思う学生を定員分仮配属させる(コンピュータにデータを入力する. この入力を行った時点で, 次の申出処理が即座に実行される). このStep 2を, 全ての研究室が定員条件を満足するまで反復する.

この会議において教員の負担となるのは, 定員を超過した場合の受入れ判断のみである. 一連のシステムをコンピュータに実装しておくことで, 一度の会議で配属を確定させることができる.

4.2.9. 配属決定アルゴリズムの計算量

配属会議における配属決定アルゴリズムの実行には、コンピュータによる処理と、教員による処理が混在しているため、これらを区別して計算量を評価する。ここで、 t 回目の配属調整処理において、未配属状態であるため申出を行う学生の集合を $S^{(t)}$ 、仮配属数と申出数の和が定員を超過した研究室の集合を $L^{(t)}$ と記す。

[コンピュータの計算量] 配属決定アルゴリズムにおいて、コンピュータを必要とする処理には次のようなものが存在する。なお、前述の処理順序 1, 2, 3 のいずれについても、以下の議論は成り立つ。

- 学生が申出を行う研究室の決定: 1 学生につき、暗黙の全順序の記憶に $O(m)$ 、Step 2 の 1 回の実行において申出相手の決定に $O(1)$ 時間を要するため、アルゴリズム終端までの領域量は $O(mn)$ 、時間量は $O(\sum_{t=1}^T |S^{(t)}|)$ 。
- $S^{(t)}$ の決定: $S^{(1)} = S$ 。 $S^{(t)}$ ($t = 2, 3, \dots, T$) は、“ $t - 1$ 回目の配属調整処理において、仮配属数と申出数の和が定員を超過した研究室から申出を拒否された学生集合”とも見なせる。よって、この処理の領域量は $O(n)$ 、アルゴリズム終端までの全時間量は $O(\sum_{t=1}^T |S^{(t)}|)$ 。
- 研究室に仮配属された学生集合の記憶: 各研究室にサイズ n の配列を用意し、学生を表す j 番目の要素が 1 (0) ならば、その学生は仮配属 (未配属) 状態であるとする。この処理の領域量は全研究室を合わせて $O(mn)$ 、時間量は各 t で配属状態が変化する学生数であり、1 申出で高々 2 人の配属状態が変化するので、最悪でも全時間量は $O(\sum_{t=1}^T |S^{(t)}|)$ で抑えられる。

以上から、領域量の上界は $O(mn)$ と評価できる。下界についても、暗黙の全順序の記憶には少なくとも mn の領域量を要することから (仮に他の 2 つの処理の領域量が mn より小さいとしても)、 $\Omega(mn)$ と表せる。

一方、時間量の上界は上記の議論から $O(\sum_{t=1}^T |S^{(t)}|)$ と表せるが、学生は同じ研究室には二度と申出を行わないため、学生 n 人がそれぞれ最大 m 研究室に申出を行うとしても、

$$\sum_{t=1}^T |S^{(t)}| \leq mn \quad (11)$$

と抑えられるため、 $O(mn)$ と評価できる。また、下界については、ある仮定の下で実際に

$$\sum_{t=1}^T |S^{(t)}| = \Omega(mn) \quad (12)$$

となる例が存在する (付録で説明する)。

以上の議論から、次の定理を得る。なお、研究室の最大定員を c_{\max} 、最小定員を c_{\min} と記す。

定理 4.5 配属決定アルゴリズムのコンピュータを用いた計算に関する領域量と時間量について、上界は共に $O(mn)$ である。また、これらの下界は $\Omega(mn)$ と評価できるが、時間量の下界についてのみ $m \geq 3$, $c_{\max} - c_{\min} = O(1)$ を仮定する必要がある。

[教員側の計算量] 配属決定アルゴリズムの実行中、 $l \in L^{(t)}$ なる研究室 (すなわち、仮配属数と申出数の和が定員を超過した研究室) について、暗黙の全順序とその時点までに明示した半順序に基づき仮配属させる学生を決定する必要がある。この計算量の評価基準として、研究室側の負担を表す

- T : 総配属調整回数
- TOTALOFF: 総申出数 $\sum_{t=1}^T |S^{(t)}|$
- TOTALLAB: 処理される総研究室数 $\sum_{t=1}^{T-1} |L^{(t)}|$

について考察する. なお, 配属決定アルゴリズムに処理順序3を採用して議論を進める.

まず, 各評価基準の計算量の上界について述べる. T の上界については, そもそも配属決定アルゴリズムは学生が申出を終えると終了することから, T は申出数の合計を超えない. 従って, 処理順序3を採用していることを考慮しても, 式(11)より $O(mn)$ と評価できる. TOTALOFFの上界についても, 式(11)より $O(mn)$ となる. TOTALLABの上界については, アルゴリズムの動作と $S^{(t)}$ および $L^{(t)}$ の定義より, $|S^{(t)}| \geq |L^{(t)}| (t = 1, 2, \dots, T)$ が自明に成り立つため, やはり $O(mn)$ と表せる.

下界については, ある仮定の下で, 任意の学生集合, 研究室集合, 研究室定員に対してある方法で選好順序(全順序)を定めると, 各評価基準の下界値がいずれも $\Omega(mn)$ となる例を示すことができる(付録). その結果として, 次の定理を得る.

定理 4.6 配属決定アルゴリズム(処理順序3を採用)の教員側の計算量は, T , TOTALOFF, TOTALLABのいずれの評価基準についても, 上界は $O(mn)$ と評価できる. また, さらに $m \geq 3$, $c_{\max} - c_{\min} = O(1)$ を仮定するとき, 下界は $\Omega(mn)$ と評価できる.

5. 計算実験

LAPの実際の問題例について, 提案手法を用いて計算実験を行った. 設定は, 筆者らの研究室配属の状況⁵を参考に, 学生数 $n = 146$, 研究室数 $m = 17$, 定員下限値 $c_i^{LB} = 7$, 定員上限値 $c_i^{UB} = 10$ とした($i = 1, 2, \dots, m$)⁶. また, 計算実験に用いたPCの情報は次の通り. OS: Windows XP, CPU: Intel Pentium4 2.80 GHz, Memory: 0.76 GB RAM.

5.1. 学生からの希望調査

学生からの希望調査を行うために, Web上に配属希望順位を入力できるシステムを実装した. それを用いて, 実際の希望傾向を基に, $n = 146$ 人分の学生の仮想入力を行い, 学生の選好順序を生成した. 詳細なデータは割愛.

5.2. 研究室定員の決定

希望調査の結果に基づき研究室の人気度を計算した上で, 定員決定アルゴリズムを適用し, 各研究室の定員を計算したところ,

定員 7: 7 研究室, 定員 8: 1 研究室, 定員 9: 1 研究室, 定員 10: 8 研究室
という結果を得た.

5.3. 配属学生の決定

上記の各設定に加え, 学生の選好順序, 研究室の人気度および定員などの情報を入力として, 配属決定アルゴリズムを適用した. アルゴリズム内の処理順序には4.2.6の3種を全て試みた. また, 研究室側の暗黙の全順序の影響を見るため, 次の2つの基準を考えた.

⁵関西学院大学理工学部情報科学科, 2005年度卒研配属. もちろん, ここでの計算結果と現実の配属とは無関係である.

⁶ここでは c_i^{LB} および c_i^{UB} を全ての研究室について等しく設定しているが, もちろん諸々の事情を考慮して, それぞれが異なる値をとっても良い.

- 希望順: 自身の研究室を上位に希望している学生ほど優先して配属させるように暗黙の全順序を構成する. すなわち, 学生 $s(s')$ がその研究室を第 $k(k')$ 希望とし, $k < k'$ ならば, 学生 s を学生 s' より上位に置く.
- 成績順: 学生の希望とは無関係に, 学生の成績順で暗黙の全順序を構成する.

以上, $3 \times 2 = 6$ パターンの計算実験を行った場合の T の値を表 6 に, コンピュータの計算に要した時間を表 7 に示す.

表 6: 総配属調整回数 T [回]

	処理順序 1	処理順序 2	処理順序 3
希望順	61	12	6
成績順	147	62	20

表 7: 計算時間 [s]

	処理順序 1	処理順序 2	処理順序 3
希望順	0.016	0.021	0.021
成績順	0.016	0.015	0.016

表 6 を列単位で比較すると, 処理順序 3 を用いた場合に T の値は小さく, 配属会議の早期終了が望める. また, 表 6 の行単位の比較により, 研究室側が学生側の希望を考慮すると, 配属会議を早期に終了させることが期待できる. このことは, 学生の希望を考慮した上で受入れる学生を決めることを研究室に促すための説得材料になる. ただし, 念のため断っておくと, 実際には全ての研究室が希望順あるいは成績順のどちらかのみを基準を採用する必要はなく, 2つの基準を持つ研究室が混在したり, これら以外の独自の基準を用いることも許される.

配属決定アルゴリズムのコンピュータの計算量については, 理論的には領域量も時間量も共に $O(mn)$, $\Omega(mn)$ であることを示したが, 表 7 より, この程度の規模の問題なら現実的に問題にならないことが判る. T についても, 定理にある $O(mn)$ の理論値が必要となることは稀で, 現に処理順序 3 と希望順を用いた場合は $T = 6$ 程度で計算が終了している.

6. おわりに

本研究では, LAP に対する解法の提案, 理論的考察, および一連のシステムの実装による計算実験を行い, 提案手法の妥当性を検証した. 本方式によれば, 学生に課す希望調査の内容は柔軟性と自由度が高いため, それぞれの希望をできる限り考慮することができ, 学生の満足度を高めるという結果につながる. 研究室(教員)側についても, 必要最小限の情報提示で, 暗黙の全順序にかなう結果を得ることができる. これらの特徴から, 本方式は, 現実の手法として有効であると期待される.

本方式は, 研究室定員の上下限が与えられたとき, まずその範囲内で定員を決定(固定)した後に配属を決めるという 2 段階法である. しかし, 研究室定員の上下限をそのままにしておき, その範囲内で満足度の高い配属を求めるといった設定の方が現実性があるとも考えられる. これは将来の課題であるが, 本論文の視点から判断する限り, 得られる解の安定性を何らかの意味で保証することが困難であると考えられ, それを克服する新しいアイデアが必要だと思われる.

参考文献

- [1] D. Gusfield and R.W. Irving: *The Stable Marriage Problem: Structure and Algorithms* (The MIT Press, 1989).
- [2] 茨木俊秀: 議員定数の最適配分法. 数理科学, **209** (1980), 52–66.
- [3] T. Ibaraki and N. Katoh: *Resource Allocation Problems: Algorithmic Approaches* (The MIT Press, 1988).
- [4] 川村和誉, 久保幹雄: 研究室割り当てシステムの開発. 日本 OR 学会 2005 年春季研究発表会アブストラクト集, (2005), 98–99.
- [5] 今野浩: 数理決定法入門 キャンパスの OR (朝倉書店, 1992).
- [6] 岡田章: ゲーム理論 (有斐閣, 1996).
- [7] G. Owen: *Game Theory* (Academic Press, 1995).
- [8] Canadian Resident Matching Service, <http://www.carms.ca/>.
- [9] 医師臨床研修マッチング協議会, <http://www.jrmp.jp/>.
- [10] National Resident Matching Program, <http://www.nrmp.org/>.

付録. 定理 4.5 と定理 4.6 の下界の証明

評価基準 T , TOTALOFF, TOTALLAB の下界を明らかにするために, $T = \Omega(mn)$ となるような問題例を構成する. その内容は, 文献 [1] に述べられている GS アルゴリズムの下界例を研究室定員が複数であるという本研究の設定に拡張したものである.

すなわち, 任意の学生集合, 研究室集合, 研究室定員が与えられたとして, 学生 s_j が持つ全順序 P_j ($j = 1, 2, \dots, n$), 研究室 l_i が持つ全順序 Q_i ($i = 1, 2, \dots, m$) を以下のように構成し, その結果 $T = \Omega(mn)$ が成立することを示す. なお, $c_1 \geq c_2 \geq \dots \geq c_m$ を仮定する (一般性を失わない).

まず, P_j ($j = 1, 2, \dots, n$) を定めるアルゴリズムの概略を説明する. 配属決定アルゴリズムの $t = 1$ において, 全ての学生はまとめて申出を行うが, 最大の定員を持つ l_1 についてはちょうど 1 人定員超過が生じ, 最小の定員を持つ l_m については 1 人定員から不足となるように P_j の一部を構成する. 次に, 反復 $t = 2, 3, \dots, T - 1$ の各時点では, それぞれ 1 反復前の $t - 1$ の時点で定員超過により申出を拒否された学生 1 人に申出を行わせるとし, そのときの申出先は l_m を除いた研究室について巡回的に行わせる ($t = 2$ では l_2 への申出, $t = 3$ では l_3 への申出, ..., というように, t が 1 増加するごとに申出先の研究室の添字は 1 増加し, l_{m-1} の次は l_1 に戻る). その結果, 各反復ではちょうど 1 つの研究室において定員超過が生じる. そして, 最後の $t = T$ では $t = 1$ からずっと 1 人不足状態であった l_m への申出を行わせ, 配属決定アルゴリズムを終了させる.

アルゴリズムの中で必要となる表記を次のように定義する.

- P_j において最大の選好順序を持つ研究室が l_i であるような全ての s_j を, 添字 j の昇順に並べた集合を S_i ($i = 1, 2, \dots, m$) とする. $|S_1| = c_1 + 1$, $|S_i| = c_i$ ($i = 2, 3, \dots, m - 1$), $|S_m| = c_m - 1$ を仮定. 定義より, $S_i \cap S_{i'} = \emptyset$ ($i \neq i'$) かつ $\sum_{i=1}^m |S_i| = n$ である.
- 研究室 l_i の仮の定員を cap_i ($i = 1, 2, \dots, m$) とする.
- $g(t) = (t \bmod (m - 1)) + 1$; $t = 1, 2, 3, \dots$ を与えると, $g(t) = 2, 3, 4, \dots, m - 1, 1, 2, \dots, m - 1, 1, 2, \dots$ と巡回する.

[全順序 P_j を定めるアルゴリズム]

Step 1: $i := 1, j := 1, cap_1 := c_1 + 1, cap_k := c_k (k = 2, 3, \dots, m - 1), cap_m := c_m - 1, S_k := \emptyset (k = 1, 2, \dots, m)$ とする.

Step 2: $i = m$ ならば Step 3 へ, さもなければ次のように場合分け.

Step 2-1: $cap_i > 0$ ならば, $P_j := (l_{g(i-1)}, l_{g(i)}, \dots, l_{g(i+m-3)}, l_m)$ とし, $S_i := S_i \cup \{s_j\}, cap_i := cap_i - 1$ と更新後, $i := (i \bmod m) + 1, j := j + 1$ と更新して Step 4 へ.

Step 2-2: $cap_i = 0$ ならば, $i := (i \bmod m) + 1$ と更新して Step 2 へ.

Step 3: ($i = m$ が成立.) cap_i の値によって, 次のように場合分け.

Step 3-1: $cap_i > 0$ ならば, $P_j := (l_m, *, *, \dots, *)$ とし ($*$ は他と重複しない任意の研究室内で, $m - 1$ 個存在), $S_i := S_i \cup \{s_j\}, cap_i := cap_i - 1$ と更新後, $i := 1, j := j + 1$ と更新して Step 4 へ.

Step 3-2: $cap_i = 0$ ならば, $i := 1$ と更新して Step 2 へ.

Step 4: $j = n + 1$ ならば終了, さもなければ Step 2 へ.

ここで, 次の例について構成例を示す.

$$n = 9, m = 3, c_1 = 4, c_2 = 3, c_3 = 2. \tag{13}$$

結果を図 8 の左側に示す (図の右側については後述). なお, 図中の * 印は, その全順序において他の要素と重複しないような任意の研究室を表す.

S	L
$(l_1, l_2, l_3) \textcircled{S_1}$	$\textcircled{l_1}^4 (S_3, S_2, S_1)$ $= (\{s_3\}, \{s_2, s_5, s_7\}, \{s_1, s_4, s_6, s_8, s_9\})$
$(l_2, l_1, l_3) \textcircled{S_2}$	$= (s_3, s_2, s_5, s_7, s_1, s_4, s_6, s_8, s_9)$
$(l_3, *, *) \textcircled{S_3}$	
$(l_1, l_2, l_3) \textcircled{S_4}$	
$(l_2, l_1, l_3) \textcircled{S_5}$	$\textcircled{l_2}^3 (S_3, S_1, S_2)$ $= (\{s_3\}, \{s_1, s_4, s_6, s_8, s_9\}, \{s_2, s_5, s_7\})$
$(l_1, l_2, l_3) \textcircled{S_6}$	$= (s_3, s_1, s_4, s_6, s_8, s_9, s_2, s_5, s_7)$
$(l_2, l_1, l_3) \textcircled{S_7}$	
$(l_1, l_2, l_3) \textcircled{S_8}$	
$(l_1, l_2, l_3) \textcircled{S_9}$	$\textcircled{l_3}^2 (*, *, *, *, *, *, *, *, *)$

図 8: LAP の問題例 (3)

次に, 研究室 l_i の全順序 Q_i の構成について述べる. P_j の構成の仕方より, 配属決定アルゴリズムを適用すると, (一部の例外を除き) 各 t で 1 人ずつ申出が行われ, その直後に

1人拒否されることになる。このことを踏まえ、 l_1 が定員超過の際に拒否を行う場合には、 $S_1, S_{m-1}, S_{m-2}, \dots, S_3, S_2, S_m$ の順とする(S_i は集合なので、 S_1 の要素が全て拒否されてから S_{m-1} の要素の拒否に移る)、すなわち

$$Q_1 = (S_m, S_2, S_3, \dots, S_{m-2}, S_{m-1}, S_1)$$

とする。 l_2 については、 $S_2, S_1, S_{m-1}, S_{m-2}, \dots, S_4, S_3, S_m$ の順に拒否させる、すなわち

$$Q_2 = (S_m, S_3, S_4, \dots, S_{m-2}, S_{m-1}, S_1, S_2)$$

とする。このように、 Q_1 も Q_2 も S_m 以外の部分は巡回的に並べさせる。これらと同様に考えて、 Q_i ($i = 1, 2, \dots, m$)を次の式(14)のように定めると、各 t で1人分の申出が行われる度に、申出を受けた研究室の全順序において最小の選好順序を持つ学生が拒否される。

$$Q_i = \begin{cases} (S_m, S_{g(i)}, S_{g(i+1)}, \dots, S_{g(i+m-2)}), & i = 1, 2, \dots, m-1, \\ \text{任意の全順序 } (*, *, \dots, *), \text{ 例えば } (s_1, s_2, \dots, s_n), & i = m. \end{cases} \quad (14)$$

例えば、式(13)のように各パラメータを設定すると、図8の右側に示したように Q_i が構成される。なお、図中の研究室側の全順序上の*印は、その全順序において他の要素と重複しないような任意の学生を表す。

また、 Q_i は

$$Q_i = (s_i^n, s_i^{n-1}, \dots, s_i^2, s_i^1), \quad i = 1, 2, \dots, m \quad (15)$$

のようにも表す。すなわち、 Q_i において最下位から u 番目の選好順序を持つ学生を s_i^u ($u = 1, 2, \dots, n$)と記す。

以上のように P_j ($j = 1, 2, \dots, n$)および Q_i ($i = 1, 2, \dots, m$)を定めると、 $t = 1, 2, \dots, T$ の順に、次に示すような申出が行われる。なお、以下に示す u は、 $(m-1)(u-1)+1 \leq t-1 \leq (m-1)u$ を満たす自然数とする(唯一つ定まる)。処理順序3を採用していても、各 t (≥ 2)では学生も研究室もそれぞれ1人ずつ処理されることに注意。また、 $s \rightarrow l$ は学生 s から研究室 l へ申出がなされることを示す。

$$\begin{cases} \forall s \in S_i \rightarrow l_i, & i = 1, 2, \dots, m, \quad t = 1, \\ s_{g(t-2)}^u \rightarrow l_{g(t-1)}, & t = 2, 3, \dots, T-1, \\ s_{g(t-2)}^u \rightarrow l_m, & t = T. \end{cases}$$

図8の例に対して配属決定アルゴリズムを適用すると、反復に従って図9のように拒否が行われ(図中の矢印が拒否される学生の遷移を、矢印の横に付した値が各 t を表している)、各 t における状況は表8のように表される。 $t = T = 9$ の時点での各 Q_i ($i = 1, 2, 3$)を図10に示す。この図より、 T の値は最初にまとめて行う1回に、各研究室が拒否した学生数の和を加えたもの(この例では $T = 1 + 4 \cdot 2 = 9$)となる。

一般的な議論に話を戻す。仮定より l_1 の定員が最大となるため、 l_m 以外の研究室のそれぞれの拒否学生数が等しくある一定の値(後述)をとるとき、その次の申出が l_m に対して行われ、 $t = T$ となり終了する。このとき、 $c_i = c_1$ である全ての i ($\neq m$)については

$$Q_i = (S_m, \text{配属させる学生の集合}, \text{拒否する学生の集合})$$

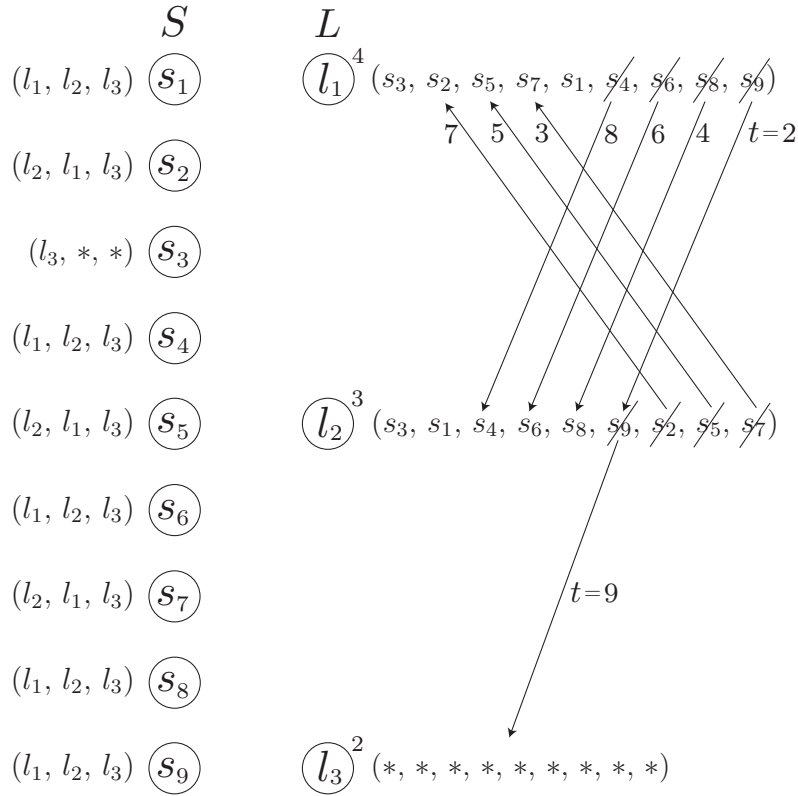


図 9: 拒否される学生の推移

表 8: 各 t での状況

t	u	申出	拒否される学生
1	—	$S_1 \ni s_1 \rightarrow l_1$	—
	—	$S_2 \ni s_2 \rightarrow l_2$	—
	—	$S_3 \ni s_3 \rightarrow l_3$	—
	—	$S_1 \ni s_4 \rightarrow l_1$	—
	—	$S_2 \ni s_5 \rightarrow l_2$	—
	—	$S_1 \ni s_6 \rightarrow l_1$	—
	—	$S_2 \ni s_7 \rightarrow l_2$	—
	—	$S_1 \ni s_8 \rightarrow l_1$	—
	—	$S_1 \ni s_9 \rightarrow l_1$	s_9
2	1	$s_1^1 = s_9 \rightarrow l_2$	s_7
3	1	$s_2^1 = s_7 \rightarrow l_1$	s_8
4	2	$s_1^2 = s_8 \rightarrow l_2$	s_5
5	2	$s_2^2 = s_5 \rightarrow l_1$	s_6
6	3	$s_1^3 = s_6 \rightarrow l_2$	s_2
7	3	$s_2^3 = s_2 \rightarrow l_1$	s_4
8	4	$s_1^4 = s_4 \rightarrow l_2$	s_9
9	4	$s_2^4 = s_9 \rightarrow l_3$	— (終端)

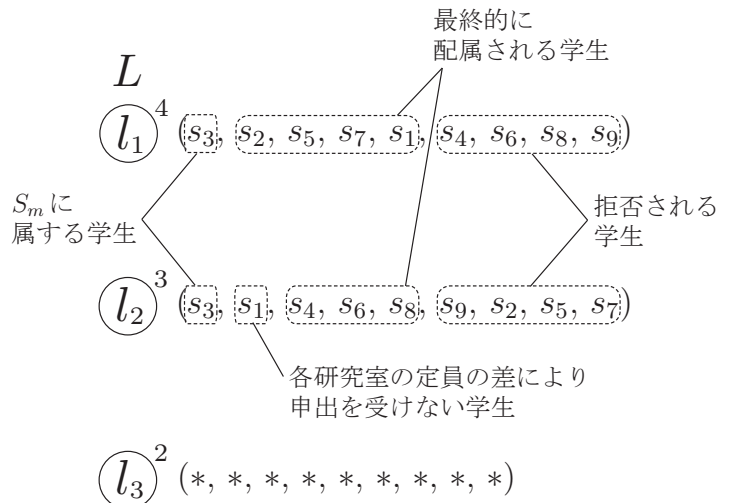


図 10: 研究室の選好順序について

となり, $c_i < c_1$ である全ての $i (\neq m)$ については

$Q_i = (S_m, \text{ 申出されない学生の集合, 配属させる学生の集合, 拒否する学生の集合})$

となっている(それぞれ図 10 を参照のこと). また, S_m は空集合でもよいことと, S_m に属する学生は必ず l_m に配属させられる ($t = 1$ で l_m に仮配属となり, $t \geq 2$ 以降は申出を行わない) ことに注意. 以上を踏まえ, l_m 以外の各研究室 l_i が拒否する学生数(すなわち, $t \geq 2$ での l_i の処理に関する配属調整回数)については, $|Q_i| = n$ より,

$$\begin{aligned} (\text{各研究室が拒否する学生数}) &= (l_1 \text{ が拒否する学生数}) \\ &= |Q_1| - |S_m| - (l_1 \text{ が配属させる学生数}) \\ &= n - (c_m - 1) - c_1 \end{aligned} \quad (16)$$

と書いて, i に依存しない値をとる.

以上より, $m = 1$ のときは明らかに $T = 1$. $1 < m \leq n$ については, 先の議論と式 (16) から,

$$\begin{aligned} T &= 1 + (l_m \text{ 以外の各研究室が拒否する学生数}) \cdot (l_m \text{ 以外の研究室の数}) \\ &= 1 + \{n - (c_m - 1) - c_1\}(m - 1) \\ &= 1 + \{(n + 1) - (c_1 + c_m)\}(m - 1) \end{aligned} \quad (17)$$

がいえる. このとき, $m > 1$ と式 (10) より $n = \sum_{i=1}^m c_i \geq c_1 + c_m$ が成り立つため,

$$(n + 1) - (c_1 + c_m) = \sum_{i=2}^{m-1} c_i + 1 > 0 \quad (18)$$

が導かれる. $m = 2$ の場合や, 一般の m で $c_1 = n - m + 1, c_2 = \dots = c_m = 1$ などとなる場合には, 式 (17) の最大項は mn とならない. 逆に言うと, $|c_1 - c_m|$ が定数 (m や n に依存しない値) であり $m \geq 3$ であれば, 式 (17) の最大項は mn となり, $T = \Omega(mn)$ と表せる. 以上が評価基準 T の下界についての議論である.

TOTALOFF の下界については, 上記の T の下界の議論において, $t = 1$ では n 人の学生が申出を行い, $t = 2, 3, \dots, T$ ではそれぞれ 1 人の学生のみが申出を行うため, 必要な処理数は $n + (T - 1)$ と表せる. よって, 下界は $\Omega(mn)$ と評価でき, これより式 (12) が導かれる.

TOTALLAB の下界については, $t = 1, 2, \dots, T - 1$ のそれぞれの時点で 1 研究室が 1 学生を拒否するため, 必要な処理数は $T - 1$ と等しい. よって, 下界は $\Omega(mn)$ と評価できる.

茨木俊秀

関西学院大学理工学部情報科学科

〒 669-1337 三田市学園 2-1

ABSTRACT

A PROPOSAL FOR THE LABORATORY ASSIGNMENT PROBLEM OF STUDENTS AND ITS MATHEMATICAL PROPERTIES

Toru Kataoka Toshihide Ibaraki
Hitachi, Ltd. *Kwansei Gakuin University*

In many universities, students are assigned to laboratories (i.e., professors) to conduct their graduation study. Since both sides of students and laboratories have their preferences, the situation resembles the stable marriage problem. But it is different in that more than one student is usually assigned to a laboratory, and only partial information about preferences is made explicit (from the consideration that the number of students is large). In this paper, we propose a scheme that contains a rational method to determine capacities of laboratories, and a modified Gale-Shapley algorithm to compute an assignment of students with a desirable stability property. Computational experiment indicates that this can be a realistic method for use in universities and other organizations.