

選好の有無が混在する場合のマッチング手法

武田 雅俊[†] 鈴木 伸崇^{††}

[†] 筑波大学情報学群知識情報・図書館学類 〒 305-8550 茨城県つくば市春日 1-2

^{††} 筑波大学図書館情報メディア系 〒 305-8550 茨城県つくば市春日 1-2

E-mail: [†]s1911502@s.tsukuba.ac.jp, ^{††}nsuzuki@slis.tsukuba.ac.jp

あらまし マッチング理論では、大まかに「二部マッチング問題」と「配分マッチング問題」の二つのタイプの問題について扱う。この二つのマッチング問題において、DA アルゴリズムや TTC アルゴリズムを用いることで安定的なマッチングを求められることがわかっている。しかし、クラウドソーシングのタスク割り当てなどの状況においては、選好を持つのは必ずしもワーカー側だけでなく、タスク側も求めるワーカーのスキルなどの点から選好を持つことがある。現在のマッチング理論では、選好の有無が混在するケースにおいて従来のアルゴリズムを単体で適用することは不可能である。そこで本研究では、選好の有無が混在する多対多マッチング問題を「混在マッチング問題」と新たに定義し、従来のアルゴリズムを複数組み合わせることにより混在マッチング問題に対する手法を提案する。

キーワード マッチング, 二部マッチング問題, 配分マッチング問題, タスク割り当て

1 はじめに

1.1 研究背景

マッチング理論では、数理的アプローチで人と人、人とモノ・サービスをどのようにマッチさせたらよいかを研究する。それぞれの需要と供給をできるだけ満たしたマッチングを実現するため、様々なアルゴリズムが社会で幅広く利用されている。マッチング理論で扱うマッチングは、いくつかのタイプに分類されるが、代表的なものとして以下の二つがある。一つが「二部マッチング問題」と呼ばれるもので、人々が二手に分かれ、一方の側の人が他方の側の人とマッチする状況を考える。二手に分かれた人々はどちらの側もどの相手とマッチしたいかについて選好を持つ。もう一つが「配分マッチング問題」と呼ばれるもので、一方の側の人が他方の側の非分割財とマッチする状況を考える。非分割財とは、自動車や臓器、パソコンなどこれ以上分割できないモノ・サービスのことを指す。二部マッチング市場と異なる点は、人側は希望順位について選好を持つが、財はそれを持たないという点である。

この二つのマッチング問題において、前者は DA アルゴリズム、後者は TTC アルゴリズムが中心的な役割を持ち、これらのアルゴリズムを用いることで安定的なマッチングや効率的なマッチングを求められることがわかっている。この二つのメカニズムは非常に重要であり、実際の市場でマッチングをデザインする場合には、どちらか一方を現実に合わせて改良することが多い。しかし、実際のマッチングにおいて二つのマッチング問題の両方の要素を併せ持つような問題に対し、従来のアルゴリズムがそのまま適用できるとは限らない。

例えば、近年、新たな目的遂行の手段としてクラウドソーシングが注目を集めている。マイクロタスク型のクラウドソーシングでは、単純な作業にタスクを分割しそれを複数のワーカーで分担して行うことで目的を遂行する。そのため、ワーカーの能力

やキャパシティに考慮したタスクの割り当てを行う必要がある。こういったクラウドソーシングのタスク割り当てなどの状況においては、選好を持つのは必ずしもワーカー側だけでなく、タスク側も求めるワーカーのスキルなどの点から選好を持つことがある。例えば、

- 簡単なデータ入力
- アンケート
- ゲームやアプリなどのテスター

などは、どのワーカーに割り当てられても結果のデータ品質に影響がないタスクである。一方で、

- プログラミング
- 意識が必要な翻訳
- ホームページやインテリアなどのデザイン案の考案

などは、ワーカーのスキルによって結果のデータ品質が異なるため、よりスキルの高いワーカーに割り当てられることが好ましいタスクとなる。つまり、選好を持つタスクと選好を持たないタスクが混在するマッチングとなる。また、学生と授業科目のマッチングでは、特に履修条件がなく定員一杯まで希望する学生を受け入れる授業科目と、その授業科目の分野に関連する学部・学科の学生やその授業科目に関連する別の科目を履修済みである学生を優先的に受け入れたいなど、受け入れに関する選好を持つ履修条件がある授業科目が混在するマッチングとなる。

これらの選好の有無が混在するマッチング問題に対して、二部マッチング問題や配分マッチング問題で用いられるマッチングアルゴリズムを単体で適用することは不可能であるため、二部マッチング問題と配分マッチング問題のどちらかに分類することは極めて困難である。つまり、従来のマッチング問題と異なる新たなマッチング問題に対して適用可能な新たなマッチング手法を考える必要がある。こういった応募・提案を受ける側の集合の中で選好の所有の有無が混在しているマッチング問題を本研究では「混在マッチング問題」として取り扱う。また、本論文では想定する状況として、ボランティアベースなど参画

するワーカがあらかじめ把握されている、マイクロタスク型のクラウドソーシングのタスク割り当てを例に提案手法の説明や評価実験を行う。混在マッチング問題では複数のワーカに対し複数のタスクを割り当てる多対多のマッチング手法を必要とする。各ワーカはそれぞれ選好を持っており、各タスクには選好を持つものと持たないものがある。またワーカには許容可能なタスク数、タスクには受け入れ可能ワーカ数がある。本研究の目的は、DA アルゴリズムを含む複数のアルゴリズムを適切に組み合わせることにより、混在マッチング問題に対して適用可能な新たな多対多マッチングの手法を考案することである。

評価実験では、複数通りのテストデータを作成し、提案手法を実行した。マッチした相手の選好順位の合計を評価指標（合計不満度）とし、その観点からベースライン手法との比較を行った。加えて、マッチングパターンごとの比較によって評価を行った。得られた結果では、提案手法を用いることでマッチング全体の合計不満度がベースライン手法と比べて小さいマッチングを実現した。

本論文の構成は以下の通りである。第2章では、二部マッチング問題と配分マッチング問題、混在マッチング問題の定義や主要なアルゴリズムについて述べる。第3章では、提案アルゴリズムについて述べる。第4章では、評価実験について述べる。第5章では、本研究のまとめを述べる。

2 諸定義

2.1 マッチング

離散数学や計算機科学の分野においてマッチングとは、グラフにおける頂点ペアの集合、すなわち頂点と頂点の対応を求める（結び付ける）ことを意味する。就活や入試、結婚など社会における様々な問題において、グラフの頂点を人や企業とみなし、最適なペアづくりや資源配分を実現するメカニズムについて取り扱うのがマッチング理論である。

本研究においても、マッチングはすべての個人またはモノからなる集合からすべての個人またはモノからなる集合への写像を指す。また、各頂点は選好と定員に関する情報を持つ。本研究において、選好は全順序制約を満たす完全リストとする。また、選好 \succsim は完備性と推移性という二つの性質を満たすと仮定する。完備性とは、任意の二つの選択肢 x と y が比較できることを意味し、必ず $x \succsim y$ または $y \succsim x$ となる。次に、推移性とは、任意の三つの選択肢 x, y, z があったとき、選好 \succsim が連結して推移していくこと、つまり、 $x \succsim y$ と $y \succsim z$ であるときに $x \succsim z$ となることを意味する。定員は多対一マッチングまたは多対多マッチングの場合に導入され、各頂点は定員を超えてマッチすることはできない。

2.2 二部マッチング問題

従来のマッチング理論の研究では、マッチングは大きく二部マッチング問題と配分マッチング問題の二つに分類される。二部マッチング問題では、人々が二手に分かれ、一方の側の人が他方の側の人とマッチする二部グラフの状況を考える。二手に

分かれた人々はどちらの側もどの相手とマッチしたいかについて選好を持つ。例として、図 2.1 に簡単な二部マッチング M を示す。各頂点はどちらの側に属しているかに関わらず選好リ

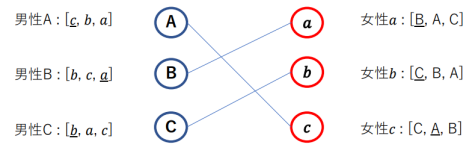


図 1 二部マッチング M

ストを所有している。すなわち、二部マッチング問題では自分が相手を選ぶと同時に、自分も相手に選ばれることに特徴がある。具体例として、一対一マッチングの安定結婚問題や一対多マッチングの研修医配属問題などが二部マッチング問題として挙げられる。

ここで、マッチングの重要な評価指標である安定性について、安定結婚問題を例に挙げて説明をする。安定結婚問題は以下の3つの要素からなる。

(1) p 人の男性を m_1, \dots, m_p , 男性の集合 $M = \{m_1, \dots, m_p\}$ と表す。

(2) q 人の女性を w_1, \dots, w_q , 女性の集合 $W = \{w_1, \dots, w_q\}$ と表す。

(3) 各男性 $m \in M$ は、女性陣 w_1, \dots, w_q のうち誰とマッチしたいかについて選好 \succsim_m を持つ。同様に各女性 $w \in W$ は男性陣 m_1, \dots, m_p のうち誰とマッチしたいかについて選好 \succsim_w を持つ。

この問題の出力は一対一のマッチングとなる。一対一マッチングにおいて、「マッチング」は男女のペアからなる集合で、誰も2人以上とはペアになっていないものと定義される。 m_i と w_j がペアになっているとき、 $M(m_i) = w_j$ および $M(w_j) = m_i$ と書く。また m_i と w_j はマッチしているという。ここで以下のようなマッチングを考える。

$$\begin{array}{ll}
 m_1 : w_3 \ w_5 \textcircled{w_1} \ w_4 \ w_2 & w_1 : m_4 \textcircled{m_1} \ m_3 \ m_2 \ m_5 \\
 m_2 : w_1 \textcircled{w_2} \ w_3 \ w_4 \ w_5 & w_2 : m_5 \ m_1 \textcircled{m_2} \ m_3 \ m_4 \\
 m_3 : w_1 \ w_4 \ \textcircled{w_3} \ w_5 \ w_2 & w_3 : m_2 \ \textcircled{m_3} \ m_1 \ m_5 \ m_4 \\
 m_4 : w_2 \ w_5 \ \textcircled{w_4} \ w_1 \ w_3 & w_4 : m_3 \ \textcircled{m_4} \ m_2 \ m_5 \ m_1 \\
 m_5 : w_4 \ w_1 \ \textcircled{w_5} \ w_2 \ w_3 & w_5 : m_3 \ m_1 \ \textcircled{m_5} \ m_2 \ m_4
 \end{array}$$

この状況のマッチングにおいて m_1 と w_5 はペアになっておらず、男性 m_1 は現在のパートナー w_1 よりも w_5 を好み、女性 w_5 は現在のパートナー m_5 よりも m_1 を好むとき、 (m_1, w_5) をマッチング M に対するブロッキングペアという。つまりブロッキングペアは「マッチングを壊す危険性をはらんだペア」である。ブロッキングペアを含まないマッチングを安定マッチングという。ゲールとシャプレイは、1962年に発表した論文で、結婚市場を導入し、安定マッチングを定義し、安定マッチングの存在を示した。彼らはその存在を示すために用いたのが

DA アルゴリズムである。DA アルゴリズムについては 3 章で詳しく説明する。DA アルゴリズムより本質的に速いアルゴリズムがないことや、多対一マッチングに対しても一対一マッチングから容易に拡張可能であることから、二部マッチング問題においては、DA アルゴリズムを用いてマッチングを求めることが最適であるとされている。

2.3 配分マッチング問題

配分マッチング問題では、一方の側の人々が他方の側の非分割財とマッチする状況を考える。非分割財は意思を持った人ではなく、自動車やパソコンなどこれ以上分割不可能なモノを指す。例として、図 2.2 に簡単な配分マッチング M を示す。基本モ

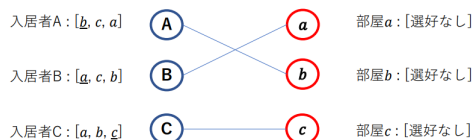


図 2 配分マッチング M

デルとしては、まず個人の集合があり、非分割財の集合がある。二部マッチング問題と異なる点としては、人々ほどの財が好きかという選好を持つが、財は選好を持たないというところにある。財が最初に誰のものであるかによって、配分マッチング問題の中でもさらに細かく分類が可能ではあるが、具体例としてルームマッチング問題や臓器移植問題が挙げられる。

ここで、多対一や多対多の配分マッチング問題において、新たに登場する性質である非浪費性について公立学校選択問題を例に説明する。ここでは、3人の生徒 (i_1, i_2, i_3) と2つの学校 (s_1, s_2) を考える。また、各学校の定員を2とする。学校は選好を持たないが、生徒は以下のように共通の選好をもつ。以下で示されるようなマッチングは浪費的なマッチングである。

$$\begin{aligned} i_1 &: \textcircled{s_1} s_2 \\ i_2 &: s_1 \textcircled{s_2} \\ i_3 &: s_1 \textcircled{s_2} \end{aligned}$$

一番人気の学校 s_1 は定員割れで、二番人気の学校 s_2 は定員一杯となっている。一番人気の学校にマッチしていない生徒は i_1 以外の2人で、この2人はどちらも自分のマッチしている学校よりも定員割れを起こしている学校 s_1 とマッチしたいと思っている。よって、このマッチングは浪費的であるといえる。すなわち、非浪費的なマッチング M では、各生徒が最終的にマッチした学校より上位に希望している学校があった場合に、その学校は必ず定員一杯まで生徒とマッチしている。ここでは、生徒 i_2 と学校 s_1 がマッチしても、このマッチが他の生徒を犠牲にすることはないため、この概念はブロッキングペアとは異なる。非浪費性はマッチングの品質を評価する指標の一つに数えられる。

2.4 混在マッチング問題

従来のマッチング理論の研究では、主に上記の二つのマッチング問題について取り扱っている。しかし、例えば、先ほどの学校選択問題の例で、入学させる生徒について選好を持つ学校と選考を持たない学校が混在しているマッチング問題であった場合、従来の二つのマッチング問題のどちらかに分類することは不可能である。そこで本論文では、こういった選好の有無が混在する場合のマッチング問題を「混在マッチング問題」として扱う。例として、図 2.3 に簡単な混在マッチング M を示す。

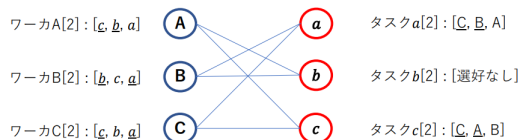


図 3 混在マッチング M

混在マッチング問題では、選好を持つ財、持たない財に対してそれぞれ異なる処理が必要となるため、従来のマッチングアルゴリズムを単体では適用できない。また、財の選好の有無に関わらずワーカーはその区別を行わない。つまり、タスクの分類によって二部マッチングと配分マッチングに分割し、段階的に割り当てを行うことは公平性の観点から望ましくない。本研究では、一回のマッチングメカニズムで混在マッチング問題に対してより安定的かつ合計不満度の小さいマッチングを求めることを目的としている。

3 提案手法

3.1 本研究で用いる記法

説明の都合上、本研究では以下の記法を用いる。

- W : ワーカー集合
- T : タスク集合
- $p(w)$: タスク t に対してワーカー w が持つ選好順位
- $p(t)$: ワーカー w に対してタスク t が持つ選好順位
- $c(w)$: ワーカー w の許容可能なタスク数 (定員)
- $c(t)$: タスク t が受け入れ可能なワーカー数 (定員)

3.2 評価指標について

本研究では、非浪費性の検証と合計不満度を評価指標として用いる。非浪費性の検証について、出力されたマッチングが最大マッチングでなかった場合に定員割れのタスクを希望するワーカーが存在しないことを確認し、非浪費性を満たしているかを評価する。不満度について、第 i 希望の相手とペアになった人の不満度を i とし、そのマッチングの全員の不満度の総和を合計不満度と定義する。そして、各ワーカーおよびタスクがより上位に希望する相手とマッチングできているほど不満度の小さい、より高品質なマッチングであると評価する。同条件のマッチングのシミュレーションデータを用いてベースライン手法と比較し、それぞれの合計不満度の比較を行う。

3.3 アルゴリズム

本節では、本研究で提案するアルゴリズムについて述べる。このアルゴリズムは、非浪費性と一定条件下での安定性を考慮している。二部マッチング問題において、DA アルゴリズムを適用することにより安定マッチングを得ることわかっている。しかし、混在マッチング問題においては選好を持たない財に対して仮マッチを決定することができないため、DA アルゴリズムを単体で適用することができない。そこで、主に配分マッチングで採用される、仮マッチを用いないアルゴリズムであるポストンアルゴリズムを組み合わせて、この問題を解決する。

提案アルゴリズムでは、応募を受ける対象であるタスクが選好を持つか否かで異なる処理を行う。応募を受けるタスクが選好を持つ場合には二部マッチング問題のようにDA アルゴリズムの仮マッチを、選好を持たない場合には配分マッチング問題のようにポストンアルゴリズムの確定マッチを用いる。提案手法の根幹となる、DA アルゴリズムとポストンアルゴリズムについて説明する。

3.3.1 DA アルゴリズム

DA アルゴリズム（受け入れ保留アルゴリズム：Deferred Acceptance algorithm）は安定結婚問題において、安定マッチングを求める方法として、1962年にデービット・ゲール（David Gale）とロイド・シャプレイ（Lloyd Shapley）によって考案されたアルゴリズムである。特徴として、DA アルゴリズムにおいて提案する側は虚偽申告のインセンティブを持たず、かつ、安定マッチングを求めることができる、などが挙げられる。DA アルゴリズムの進行は以下の通りである。

- ステップ1

各ワーカー w は自分の選好リストの1番目に希望しているタスクに応募する。各タスク t は選好リストに従って、応募したワーカーと定員までマッチする。受け入れの決定は、タスク t が定員 $c(t)$ 人のワーカーを受け入れるか、あるいは応募したワーカーをすべて受け入れるまで行う。残りのワーカーはすべて拒否される。

- ステップ $k, k \geq 2$

定員に達していないワーカー w はそれぞれ、許容可能かつまだ応募していないタスクの中で最も希望しているタスクに応募する。定員を超えて応募を受けたタスクは以下の中から最も好きなワーカーを一時的に受け入れる。ほかのワーカーを断る。

(1) 新しく応募をしてきたワーカー

(2) 前のステップで一時的に受け入れたワーカー（複数の可能性あり）

- 終了

どのワーカーも断られなくなったときにアルゴリズムは終了する。各ワーカーはアルゴリズムが終了したときに一時的に割り当てられたタスクとマッチする。

3.3.2 ポストンアルゴリズム

ポストンアルゴリズム（受け入れ即決アルゴリズム：Immediate Acceptance algorithm, IA アルゴリズム）はDAアルゴ

リズムと似ているが、ステップ2以降で別の方法を使う。決定的な違いは、ポストンアルゴリズムではそれぞれのステップで実現するマッチは一時的ではなく、最終決定されるということである。言い換えれば、ここでの最終的な受け入れは即時に決定され、アルゴリズムが終了する最後のステップまで先送りにされない。また、このアルゴリズムはタスク側の選好の有無に関わらず適用が可能であるため、評価実験の際にはポストンアルゴリズム単体の適用をベースライン手法として比較を行う。ポストンアルゴリズムの進行は以下の通りである。

- ステップ1

各ワーカー w は自分の選好リストの1番目に希望しているワーカーに応募する。各タスクは選好リストに従って、応募したワーカーと定員までマッチする。受け入れの決定は、タスク t が定員 $c(t)$ 人のワーカーを受け入れるか、あるいは応募したワーカーをすべて受け入れるまで行う。残りのワーカーはすべて拒否される。

- ステップ $k, k \geq 2$

定員に達していないワーカー w はそれぞれ、許容可能かつまだ応募していないタスクの中で最も希望しているタスクに応募する。

- 終了

どのワーカーも拒否されないステップ、あるいはすべてのタスクの定員が一杯になったステップでアルゴリズムは終了する。タスクが選好リストを持たない場合、ポストンアルゴリズムの確定マッチでは優先順序メカニズムに従って応募が早い順に受け入れる。実際に学校選択問題などにおいて学校側が選好を持たないと設定した場合、メカニズムの主催者が何らかの方法（ランダムに決定など）で応募者の順番を線形的に決定する必要がある。

3.4 提案アルゴリズム

本研究で提案するアルゴリズムは、ポストンアルゴリズムのステップ毎に割り当てを確定する性質を利用し、DA アルゴリズムに組み込むことで仮マッチを決定できない財に対して安定性や非浪費性の性質を満たし、かつ公平な処理を行う。提案アルゴリズムが安定性を持つことは、DA アルゴリズムが安定性を持つ証明と同様に背理法を用いて簡単に示すことができる。まず、提案アルゴリズムにて出力されたマッチングが安定マッチングでない、すなわち、ブロッキングペアが存在すると仮定する。その場合、ブロッキングペア (w_i, t_j) が存在するはずである。しかし、アルゴリズムの進行を考えるとマッチングは応募するワーカーの選好が上位の順に行われるため、アルゴリズムの終了より前のステップにてペア (m_i, w_j) が仮マッチされており、その後、マッチの解消が発生していなければならない。しかし、アルゴリズムの進行上選好を持つ財はステップ数が進むほど希望順位の高い相手とマッチングする。また、選好を持たない財はそもそも選好を持たないためブロッキングペアを生成することができない。つまり、ブロッキングペアに矛盾し、提案アルゴリズムは安定マッチングを出力する。具体的なアルゴリズムの進行は以下の通りである。

- ステップ1

各ワーカー w は自分の選好リストの 1 番目に希望しているタスクに応募する。

(1) 応募を受けたタスクが選好を持つ場合

各タスク t は選好リストに従って、応募したワーカーと定員までマッチする。受け入れの決定は、タスク t が定員 $c(t)$ 人のワーカーを受け入れるか、あるいは応募したワーカーをすべて受け入れるまで行う。残りのワーカーはすべて拒否される。

(2) 応募を受けたタスクが選好リストを持たない場合

各タスク t は定員に達するまで応募したワーカーと確定的なマッチをする。タスク t が定員 $c(t)$ 人のワーカーを受け入れるか、あるいは応募したワーカーをすべて受け入れるまで行う。残りのワーカーはすべて拒否される。また、タスクの定員に達した時点でワーカーの選好リストからそのタスクを削除する。

- ステップ $k, k \geq 2$

定員に達していないワーカー w はそれぞれ、許容可能かつまだ応募していないタスクの中で最も希望しているタスクに応募する。

(1) 応募を受けたタスクが選好リストを持つ場合

定員を超えて応募を受けたタスクは以下の中から最も好きなワーカーを一時的に受け入れる。ほかのワーカーを断る。

(a) 新しく応募を受けたワーカー

(b) 前のステップで一時的に受け入れたワーカー (複数の可能性あり)

(2) 応募を受けたタスクが選好リストを持たない場合

ステップ 1 と同様に各タスク t は定員に達するまで応募したワーカーと確定的なマッチをする。タスク t が定員 $c(t)$ 人のワーカーを受け入れるか、あるいは応募したワーカーをすべて受け入れるまで行う。残りのワーカーはすべて拒否される。また、タスクの定員に達した時点でワーカーの選好リストからそのタスクを削除する。

- 終了

どのワーカーも断られなくなったときにアルゴリズムは終了する。各ワーカーはアルゴリズムが終了したときに一時的に割り当てられたタスクとマッチする。

Algorithm 3.3.1 に疑似コードを示す。アルゴリズムの入力は、ワーカー集合 W 、タスク集合 T 、各ワーカーは w と各タスク t の選好 $p(w)$ および $p(t)$ 、各ワーカーの許容可能なタスク数 (定員) $c(w)$ 、各タスク t の受け入れ可能ワーカー数 (定員) $c(t)$ である。アルゴリズムの出力は、マッチしているワーカーとタスクの組の集合 *resultlist* である。

4 行目はアルゴリズムが終了する条件について示している。 $cw_i > 0$ かつ $len(pw_i) > 0$ 、すなわち定員が 1 以上ありそのワーカーが希望している全員から断られていないようなワーカーが一人でも存在する限りアルゴリズムは終了しない。6 行目から 9 行目は選好を持つタスクに対する処理を示している。 $pt_j > 0$ である t 、つまり選好を持つタスクに対して DA アルゴリズムの仮マッチを用いるため、仮マッチリストである *temlist* に格納する。10 行目から 16 行目は選好を持たないタスクに対する処理を示している。 $pt_j = 0$ である t 、つまり選好を持たない

Algorithm 1 Mixed Acceptance algorithm

Require: ワーカー集合 $WL = (w_1, w_2, \dots, w_n)$, $w_i = (c(w_i), p(w_i))$ $1 \leq i \leq n$, タスク集合 $TL = (w_1, w_2, \dots, w_m), t_j = (ct_j, pt_j)$ $1 \leq j \leq m$

Ensure: マッチしているワーカーとタスクの組の集合 *resultlist*

```

1: temlist = []
2: conlist = []
3: resultlist = []
4: while  $cw_i > 0$  かつ  $len(pw_i) > 0$  の条件を満たすワーカー  $w_i \in WL$  が存在する do
5:   if  $ct_j > 0$  then
6:     if  $pt_j > 0$  then
7:       temlist.append( $w_i, t_j$ )
8:        $cw_i = cw_{i-1}$ 
9:        $ct_j = ct_{j-1}$ 
10:    else
11:      conlist.append( $w_i, t_j$ )
12:       $cw_i = cw_{i-1}$ 
13:       $ct_j = ct_{j-1}$ 
14:      if  $ct_j == 0$  then
15:        WL 内のワーカーのうち、 $t_j$  とマッチしていない各  $w_i$  に
        対して
16:         $pw_i.pop(t_j)$ 
17:      end if
18:    end if
19:  else
20:    if  $pt.index(w_{i+1}) > pt.index(temlist[w_i])$  then
21:       $pw_{i+1}.pop(t)$ 
22:    else
23:      temlist.pop( $w_i, t_j$ )
24:      temlist.append( $w_{i+1}, t_j$ )
25:    end if
26:  end if
27: end while
28: resultlist = temlist + conlist
29: return resultlist

```

タスクに対してはポストンアルゴリズムの確定マッチを用いるため、確定マッチリストである *conlist* に格納する。19 行目から 25 行目はステップ k 以降に応募を受けたタスクが定員を超えた場合の処理を示している。選好を持たないタスクは定員に達した時点でワーカーの選好リストから除外しているためここでは仮マッチを扱う処理のみが行われる。タスクの選好リストを参照し、定員を超えて応募したワーカーの順位と前ステップで仮マッチリストに格納したペアのワーカーの順位を比較し、もし前者のワーカーがより上位の希望であれば現在の仮マッチのペアを削除し、新たにペアを仮マッチリストに格納する。もし仮マッチ中のワーカーがより上位の希望であればステップ k で応募をしたワーカーの選好リストからそのタスクを削除し、次のワーカーの応募に移る。28 行目はアルゴリズムが終了条件を満たしたとき、仮マッチリスト *temlist* に格納していたペアを確定マッチリスト *conlist* に追加し、最終的な割り当てとして出力していることを示している。

4 評価実験

本章では、提案アルゴリズムに関する評価実験について述べる。

4.1 概要

まず、前章で述べた提案アルゴリズムを Python を用いて実装した。次に、多対多マッチングに必要なワーカ集合とタスク集合を用意した。各ワーカは完全リストである選好リストと1以上の定員を持つ。各タスクは、選好を持つタスクと選好を持たないタスクに分けられ、前者は完全リストである選好リストと1以上の定員を持ち、後者は選好リストを持たず1以上の定員のみを持つ。各ワーカ、タスクが所有する選好リストはすべてランダムに順序が線形的に決定されるものを用いる。そして、提案手法とベースライン手法でそれぞれマッチングを行い、不満度を計測した。ベースライン手法はポストンアルゴリズムを選好の有無にかかわらず全タスクに対し適用させたものとする。実験に用いたデータはすべて、(ワーカ数、タスク数(内選好を持たないタスク数)、ワーカ側の合計定員、タスク側の合計定員) = (8, 8 (2), 24, 24) である。また、合計不満度を計測する際に、選好を持たないタスクの不満度の期待値を((ワーカ集合に含まれるワーカの人数×そのタスクの定員)/2)とする。これは、提案手法およびベースライン手法がどちらの側の不満度により影響を及ぼしているかについて、比較を容易にするためワーカ、タスク間で頂点数の条件を同じにする処理を行う。また、混在マッチング問題における多対多マッチングを以下の4つの場合に分けて評価実験を行う。

(1) タスク集合、ワーカ集合がすべて同じ定員数を持つ(同×同)

各ワーカの希望タスク数と各タスクの定員はすべて3とする。

(2) タスク集合は同じ定員数を持つが、各ワーカがランダムな定員数を持つ(異×同)

各ワーカの希望タスク数は総定員が24となるように1~5のうちランダムで決定され、各タスクの定員はすべて3とする。

(3) ワーカ集合は同じ定員数を持つが、各タスクがランダムな定員数を持つ(同×異)

各ワーカの希望タスク数はすべて3であり、各タスクの定員は総定員が24となるように1~5のうちランダムで決定される。

(4) 各タスク、ワーカともにランダムな定員数を持つ(異×異)

各ワーカの希望タスク数と各タスクの定員は総定員が24となるように1~5のうちランダムで決定される。

多対多マッチングにおいては、マッチングパターンによってマッチング不成立の発生条件や確率が異なるため上記のような分類を行う。これには、提案手法がどのマッチングパターンにおいてより効果を発揮するのかを検証するという意図もある。

4.2 評価実験の結果

前節で述べたシミュレーションデータを用い、提案手法とベースライン手法をそれぞれ実行し不満度を計測した。「ワーカ

集合とタスク集合の人数と選好を持たないタスクの数、各側の総定員数」の入力を行うと自動生成される「各ノードの選好リスト、定員」をシミュレーションのデータとして用いる。また、第*i*希望の相手とペアになった人の不満度を*i*と定義しているため、選好を持たないタスクに対して不満度を計測することはできない。そこで評価実験では、各側の頂点数および合計定員を同数に設定し、両側で合計不満度の期待値が同じになるよう設定を行った。

以下に評価実験の結果を示す。表 4.1 は提案手法とベースライン手法をそれぞれ適用させた場合の合計不満度とマッチング不成立数を各マッチングパターンに対して20通り、計80通りのマッチングを実行し出力されたものを示している。

表 1 提案手法

不満度	同×同	異×同	同×異	異×異	計
ワーカ和	1297	1403	1462	1484	5646
タスク和	1584	1606	1542	1589	6321
全体和	2881	3009	3004	3073	11967
不成立数	6	6	6	12	30

表 2 ベースライン手法

不満度	同×同	異×同	同×異	異×異	計
ワーカ和	1183	1286	1265	1354	5088
タスク和	1793	1880	1826	1869	7368
全体和	2976	3166	3091	3223	12456
不成立数	6	20	14	24	64

ベースライン手法と提案手法で同じデータセットを用いたため、DA アルゴリズムにおける仮マッチを導入しているか否かによって生じた合計不満度の差であることを示している。提案手法のワーカ和とタスク和、ベースライン手法のワーカ和の3項目において、不満度に大きく差が生じることはなかった。一方で、ベースライン手法のタスク和はマッチングパターンによって、1793~1880と他の項目と比べて不満度が大きくなることが示されている。そのため、合計不満度では80通りの8対8のマッチングで489の不満度の差が生じた。また、マッチング不成立数についても、ベースライン手法と比べ提案手法を用いることでその数が小さくなっていることが示されている。また、マッチング不成立になった場合、ベースライン手法と提案手法のどちらにおいても、定員に余りを持つタスクを希望するワーカが存在しなかった。すなわち、全マッチングで浪費的なマッチングは一度も発生しなかった。

4.3 考察

前節で示した結果を見ると、多対多マッチングの全パターンにおいて提案アルゴリズムを用いることで不満度の小さいマッチングを出力できたことが分かる。詳細に述べるなら、ワーカ側の合計不満度は提案手法よりもベースライン手法の方がわずかに小さい。しかし、ベースライン手法ではタスク側の合計不満度が非常に大きく、提案手法の合計不満度に大きく差が生まれるため、両側を含むマッチング全体では、提案手法の合計不満度の方がより小さくなった。ベースライン手法ではワーカ側と比べ、タスク側の合計不満度が大きくなる傾向がみられる。

これは、ポストンアルゴリズムにおいて、応募を受けるタスク側の選好リストを参照するのが、同ステップ内で定員を超える応募があった場合のみとなるのが主な理由である。一方で、DA アルゴリズムでは、ステップに関係なく定員を超えるたびに応募したワーカと仮マッチ中のワーカとを選好リストを参照し比較する。そのため、DA アルゴリズムを組み合わせた提案手法では、タスク側の選好リストを参照する回数が多く、タスク集合にとってより希望するワーカが割り当てられやすくなったと考えられる。クラウドソーシングのタスク割り当てのような、選好の有無が混在する場合のマッチング手法は現時点で考案されていなかったが、選考を持つ財と持たない財について、それぞれ別のアルゴリズムを適用することによって、不満度の小さいマッチングを求めることが可能となった。また、マッチング不成立数については、選好リストを持つタスクに対して仮マッチを導入することにより、ベースライン手法と比較してワーカが応募可能なタスク数が大きく減少しないため、全タスクから拒否を受けてマッチが不成立となる最終ステップへ移行する前に定員までマッチしやすくなったと考えられる。また、提案手法はポストンアルゴリズムと DA アルゴリズムのマッチング手法を組み合わせたものであるが、ポストンアルゴリズム、DA アルゴリズムともに非浪費性の性質を満たしているため、提案手法も同様に非浪費性を満たしたマッチングとなると考えられる。

5 むすび

インターネットの普及に伴って、これまで不可能だった様々な出会いが可能となり、人やモノ、サービスを結びつけるマッチングサービスが社会に浸透してきた。社会を取り巻く複雑なマッチングは従来のマッチング理論の研究では扱うことが困難なものも存在する。本論文では、複雑なマッチングの一つである、選好の有無が混在するマッチングを「混在するマッチング問題」として取り上げた。従来のマッチング問題の特徴から混在マッチングを定義した上で、非浪費性や一定条下の安定性を考慮した、多対多マッチングアルゴリズムを提案した。

評価実験では、主に配分マッチング問題において適用されているポストンアルゴリズムをベースライン手法に設定し、シミュレーションを通して得られたマッチング結果の合計不満度を比較することにより提案アルゴリズムの有用性の検証を行った。ベースライン手法では応募を受ける側の希望を満たす性質を持たないことが分かっていたが、提案手法を用いることにより、選好を持たない財が混在するマッチングに対しても仮マッチを導入することに成功した。仮マッチを導入することで、応募を受ける側の希望をある程度満たし、合計不満度の小さいマッチングの出力が可能であることを確認した。

今後の課題として、本研究ではクラウドソーシングのタスク割り当てを想定しているが、個々のタスクを独立に扱っている。しかし、実際のクラウドソーシングにおけるタスクには互いに依存関係がある。依存関係とはタスク同士が関連または連続している関係を指し、同一ワーカが連結した複数のタスクを行う

必要がある場合がある。ワーカとタスクを一对一に結びつけるだけでなく、依存関係を考慮した複雑なマッチング問題に対して機能するアルゴリズムを設計・実装することによってより現実的な問題に対して効率的なタスクの割り当てを実現できると考えられる。

また、本研究では、ワーカと選好を持つタスクがお互いにフルサイズの選好リストを所有している場合には完全マッチングを出力できることを示しているが、部分的な選好リストが含まれる場合については対応できていない。実際のクラウドソーシングのタスク割り当てにおいては、ワーカが対象のタスクに対し厳密な選好リストを所有しているとは限らないため、選好リストのサイズに制限がある状況で機能するアルゴリズムを設計・実装することが望ましい。部分的選好下における学校選択問題を考察している関連研究があり、学生最適性を満たす割り当てを出力するアルゴリズムの提案が行われている。これらを混在マッチング問題へ応用した新たな手法を考案予定である。

文 献

- [1] 宮崎修一. 安定マッチングの数理とアルゴリズム トラブルのない配属を求めて. 現代数学社, 2018, p. 4, 6, 12-13
- [2] 山口大河, 鈴木伸崇. “マイクロタスク型クラウドソーシングにおけるワーカの希望タスク数を考慮したタスク割当て手法”. DEIM Forum, 2021-01-24, 2021.
- [3] 栗野盛光. ゲーム理論とマッチング. 日経文庫, 2019, p. 94-123, 150-193.
- [4] ギオーム・ハーリンジャー. マーケットデザイン オークションとマッチングの理論・実践. 栗野盛光訳. 中央経済社, 2020, p. 199-214, 234-236, 255, 309-320.
- [5] 秋葉俊祐, 鈴木伸崇. “安定性とインクルージョン性を考慮したマッチング手法”. DEIM Forum, 2022-3, 2022.
- [6] 和田凌司, 八尋健太郎, 山口知晃, 東藤大樹, 横尾真. 部分的選好下における学校選択メカニズム. 人工知能学会全国大会論文集, 2019.