

# 空間クラウドソーシングのための 多様性を考慮したタスク割り当て手法

趙 セイ<sup>1,a)</sup> 石川 佳治<sup>1,b)</sup> 肖 川<sup>2,c)</sup> 董 テイテイ<sup>1,d)</sup> 佐々木 勇和<sup>3,e)</sup>

概要：多種のセンサ機能を搭載するモバイルデバイスの普及により，ユーザのモバイルデバイスをセンシング機器として用いる空間クラウドソーシングが注目されている．空間クラウドソーシングでは，システムサーバが位置に関連するタスク（例：ある地点の写真を撮影してほしい）をワーカに割り当て，ワーカが指定される位置に訪ねて，タスクを完成する．ワーカグループに収集されたデータ（時空間データ，写真，テキストなど）を用いて，役立つ知識を発見し，もしくはより良いサービスを提供することができる．しかし，タスクの地点への空間距離はワーカの意欲とタスクの完了時間に影響しているため，移動コストを最小化することを目標として考える．また，ワーカの好みや背景知識などの属性が，収集されたデータの質に影響する可能性もある．そこで，本研究では，データの質と移動コストの両方を考慮したタスク割り当て手法について述べる．

## 1. はじめに

「群衆の英知」に始まり，AMT (Amazon Mechanical Turk) [1] をはじめとするクラウドソーシングプラットフォームが注目を集めている．特に，多種のセンサ機能を搭載するモバイルデバイスの普及により，空間特徴を加えて，一種のクラウドソーシングである空間クラウドソーシングも急激に成長している．既存の空間クラウドソーシングプラットフォームとしては，GeoCrowd [9]，GeoTrueCrowd [10]，gMission [3] などが提案されている．これらの空間クラウドソーシングプラットフォームでは，システムサーバが位置に関するタスク（例，ある場所の写真や口コミを収集してほしい）をワーカに割り当て，ワーカが物理的にタスクに付けられた場所に訪問し，データを収集する必要がある．そして，収集されたデータをシステムサーバに送信する．

空間クラウドソーシングには，タスクを割り当てる際の重要な要素としてデータの質と空間コストがある．既存研

究としては，ワーカの履歴情報に基づき，信頼度を評価する研究が多い [2] [10]．ただし，ワーカの専門知識や好みなど本質的な属性もデータの質に影響すると考えられる．例としてレストランの評価を考える．ワーカの好みに偏りがあると，レビュー結果にもそれが影響し，結果が十分信頼できないものとなる．信頼性を上げるには，複数の多様なワーカに評価してもらうことが有効であると考えられる．一方，空間コストはワーカがタスクを完成するための移動距離（典型的にはユークリッド距離であるが，道路ネットワーク上の距離を考えることもできる）である．空間コストが高いほど，タスクの完了時間は長くなり，ワーカの意欲も低くなる．そのため，タスクの割り当てにおいては，多様性を持ち，空間コストも小さいワーカグループを選ぶ必要があると考える．

本研究においては，タスクは「多様性」と「空間コスト」という二つの観点でとらえ， $k$ 近傍割り当て問題を定義する．各タスクに似通っていない複数のワーカを割り当てることで，タスクの多様性を保証する．一方，空間コストが高いほど，処理時間やワーカの負担の増大につながるため，空間コストを最小化することを目指す．本研究では， $k$ 近傍割り当て問題を解決するために，空間クラウドソーシングにおける効率的な割り当てフレームワークを提案する．割り当て手法としては，深さ優先探索に基づく有効性を考慮した厳密アルゴリズムを提案する一方，効率性を考慮した局所最適化割り当て手法と交換に基づく割り当て手法も併せて提案する．

<sup>1</sup> 名古屋大学大学院情報科学研究科  
Graduate School of Information Science, Nagoya University

<sup>2</sup> 名古屋大学高等研究院  
Institute for Advanced Research, Nagoya University

<sup>3</sup> 名古屋大学未来社会創造機構  
Institute of Innovation for Future Society, Nagoya University

a) zhao@db.ss.is.nagoya-u.ac.jp

b) ishikawa@is.nagoya-u.ac.jp

c) chuanx@nagoya-u.jp

d) dongtt@db.ss.is.nagoya-u.ac.jp

e) yuya@db.ss.is.nagoya-u.ac.jp

## 2. 割り当て問題の定義

本章では、多様な  $k$  近傍割り当て問題を形式的に定義する。まずは、ワーカのタスクへの割り当てを以下のように定義する。なお、本稿で使われた記号とその意味を表 1 に表す。

**定義 1**  $n$  個のタスクとワーカ集合  $W$  が与えられ、各ワーカは  $d$  次元のバイナリベクトルで示される。割り当て (assignment) は、 $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$  で与えられる。ただし、 $1 \leq i \leq n$  について  $A_i \subset W$  かつ  $|A_i| = k$  であり、 $i \neq j$  なる  $1 \leq i, j \leq n$  について  $A_i \cap A_j = \emptyset$  が成り立つ。なお、 $k$  は要求者によって決められる。□

そこで、各タスクに割り当てしたワーカが互いに似ていないという条件を導入するための制約条件を定義する。

**定義 2** 与えられた割り当て  $A_i$  ( $1 \leq i \leq n$ ) において、任意の 2 名のワーカの非類似度が与えられた閾値  $\tau$  より大きいという制約

$$\forall w, w' \in A_i \text{ such that } w \neq w', dsim(w, w') \geq \tau \quad (1)$$

を多様性制約 (diversity constraint) と呼ぶ。ただし、 $0 \leq \tau \leq 1$  である。□

ここでは、類似度をもとに非類似度を  $dsim(w, w') = 1 - sim(w, w')$  と定義する。様々な類似度の定義があるが、本論文では、類似度の一例として Jaccard Similarity を考える。さらに、ほかの類似度 (例、コサイン類似度) あるいは非類似度尺度 (例、ハミング距離) に拡張することができる。Jaccard Similarity は以下のとおり定義される。

$$JS(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

ここで例としてレストランのレビューを考慮する。各ワーカのプロフィールは、五つのカテゴリ { イタリアン, フレンチ, 和食, 中華, タイ } に基づく好みの集合で構成される。例えば、与えられたワーカ  $w_a$  と  $w_b$  のプロフィールは次のようになる:  $prof(w_a) = \{ \text{イタリアン, フレンチ, 和食} \}$ ,  $prof(w_b) = \{ \text{和食, 中華, タイ} \}$ 。つまり、 $w_a$  と  $w_b$  はそれぞれ  $(1, 1, 1, 0, 0)$  と  $(0, 0, 1, 1, 1)$  で表される。そこで、対応する非類似度を  $dsim(w_a, w_b) = 1 - sim(w_a, w_b) = 1 - \frac{1}{5} = 0.8$  と計算することができる。

次は、本論文が対象とする多様な  $k$  近傍割り当て問題を、最適化問題として以下のように定義する。

**定義 3** タスク集合  $T = \{t_1, t_2, \dots, t_n\}$  とワーカ集合  $W = \{w_1, w_2, \dots, w_m\}$  が与えられたとき、多様な  $k$  近傍割り当て問題 (Diverse  $k$  Neighbor Assignment Problem) を、

$$\begin{aligned} & \text{Minimize} && \max_{i=1}^n \max_{w \in A_i} dist(t_i, w) \\ & \text{subject to} && \text{diversity constraint} \end{aligned} \quad (3)$$

と定義する。すなわち、多様性制約を満たすという前提の

もとで、タスクと割り当てられたワーカの距離の最大値を最小化する  $\mathcal{A} \in U_{\mathcal{A}}$  を求める問題とする。□

距離の最大値に着目する理由は、これがタスクの完了時間に影響するためである。最大距離が大きい場合、割り当てられた他のワーカのタスクへの距離が小さくても、タスクの完了時間は遅くなってしまふ。上記の定義はこのように着目している。一方で、ワーカの労力に着目すれば、平均距離を小さくする割り当ても候補として考えられる。他の目的関数については今後の研究対象としたい。

表 1 記号とその意味

Symbol	Definition
$T = \{t_1, \dots, t_n\}$	タスク集合
$W = \{w_1, \dots, w_m\}$	ワーカ集合
$\mathcal{A} = (A_1, \dots, A_n)$	割り当て割り当て
$k$	各タスクに割り当てる人数
$dsim()$	非類似度関数
$\tau$	多様性の閾値

## 3. 関連する問題

多様な  $k$  近傍割り当て問題は、アルゴリズムの分野で知られる独立集合問題 [4] と関連している。独立集合 (independent set) とは、与えられたグラフ  $G = (V, E)$  における頂点の集合  $V^* \subseteq V$  であり、 $V^*$  内の任意の 2 つの頂点をつなぐ辺が存在しない場合をいう。各ワーカをグラフの頂点とみなし、ワーカ  $w, w'$  の間の非類似度  $dsim(w, w')$  の値が  $\tau$  以上であるときにのみ  $w, w'$  間に辺が存在すると考えれば、式 (1) の多様性制約により定義される各  $A_i$  はサイズ  $k$  の独立集合となっている。指定されたサイズ  $k$  の独立集合が存在するかという問題は、NP 完全問題ということが知られており、さらに本研究では  $A_i \cap A_j = \emptyset$  という制約も加わっているため、より難しい問題となっている。全ての組み合わせをチェックして最適解を見つけると、計算量が  $k C_m^n$  になる ( $n$ : タスク数,  $m$ : ワーカの総人数,  $k$ : 各タスクに割り当てるワーカ数)。そのため、最適解ではなく近似解を導くヒューリスティクスが重要となってくる。

一方で、式 (3) に示すように、最大距離を最小化するという最適化が入っている点は独立集合問題と異なっている。そのため、タスクの周辺のワーカを優先的に考慮するようなヒューリスティクスが有効であると考えられる。割り当て問題の処理時間は、アルゴリズムだけでなく、依頼者の要求、対象となるタスクやワーカの分布やパラメータ ( $n, m, k, \tau$ ) にも依存すると考えられるため、どのような状況のもとでどの程度の処理時間が得られるかの解析が重要となる。

## 4. ベースライン手法

ベースライン手法として、深さ優先探索 (DFS) に基づ

くアプローチを考える。基本的なアイデアは、すべての可能な割り当てを列挙することである。次の候補のタスク-ワーカペアを考えるときに、多様性制約と枝刈り条件をチェックし、可能性のない割り当てを枝刈りする。多様性制約は定義 2 のように定義されている。多様性の条件を満たしていなければ、次の候補の処理にただちに移動する。また、枝刈り条件を以下の基準で与える。

- 枝刈り基準：現在の最良の割り当ての最大距離を下限値として考えて、候補タスク-ワーカペアの距離が下限値より大きい場合、枝刈りを行い、次の候補をチェックする。

このベースライン手法では、多少の効率化を図ることができたが、最悪ケースの場合も全部の割り当てを探索する必要がある。そこで、本論文では、厳密アルゴリズムと近似アルゴリズムを併せて提案する。

## 5. 提案手法

### 5.1 空間クラウドソーシングにおける割り当てフレームワーク

本研究では、図 1 のような空間クラウドソーシングにおける割り当てフレームワークを提案する。ただし、実際のアプリを開発することではなく、割り当てモジュールの実現することに注目している。

ユーザインターフェースは依頼者からのタスクを含める問合せとワーカからの位置情報を受け取り、また依頼者と相互に作用する。特に、タスクを割り当てるとき、効率性と有効性のトレードオフを考慮する。異なる依頼者が異なる要求があるので、依頼者ごとに適切な割り当て方針を行う必要があると考えられる。提案されているフレームワークでは、依頼者は、ユーザインターフェースより、現在の割り当て結果を確認できる。結果に満足した場合、終了コマンドを送信し、割り当てモジュールが対応する割り当てを返す。一方、バックエンドサーバがワーカの好みなどの属性や位置情報を格納し、タスク割り当てモジュールにアクセスできる。

割り当てモジュールが初期割り当てとローカル再割り当て二つの部分で構成される。初期割り当てを用いて、短い処理時間で暫定解を見つけることができる。ローカル再割り当て段階では、暫定解に基づき、厳密アルゴリズム、もしくは近似アルゴリズムを使って、割り当てを改善する。ローカル割り当て段階が終了条件を満たすまで、繰り返し行う。特に、効率性と有効性のトレードオフを考慮するので、依頼者のフィードバックによって、ローカル割り当て段階を終了するかどうかを決めることができる。

### 5.2 初期割り当て

本論文では、初期割り当てとしては、二つの割り当て手法が提案されている。貪欲的な割り当て手法、もしくは多

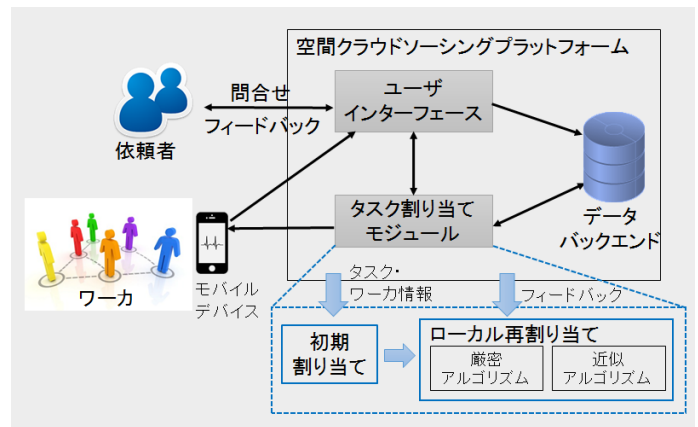


図 1 空間クラウドソーシングにおける割り当てフレームワーク

様性を考慮した割り当て手法を用いて、一つの暫定解を見つける。

#### 5.2.1 貪欲な割り当て手法

本節では、貪欲的な手法に基づく初期割り当てについて述べる。最大距離を最小化することを目指すので、距離が小さいタスク-ワーカペアの優先度が高いと考えられる。つまり、タスク-ワーカペアの距離の小さい順で割り当てを行う。まずは、距離の小さい順でタスク-ワーカペアをソートする。そして、多様性のチェックを行い、多様性の条件を満たすワーカを割り当て結果に入れる。次に、残りのタスク集合とワーカ集合を更新する。各タスクに  $k$  ワーカを割り当てると、アルゴリズムを終了し、暫定解を返す。図 2 で示した実行例では、貪欲な割り当て手法を用いて、暫定解  $A = \{(w_1, w_3), (w_4, w_6), (w_7, w_8)\}$  が見つけられる。

#### 5.2.2 多様性を考慮した割り当て手法

貪欲な割り当て手法では、距離によって優先度を決めているが、多様性の制約があるため、悪い暫定解を見つける可能性がある。本節では、貪欲な割り当て手法を改善し、多様性を考慮した割り当て手法を提案する。

アルゴリズム 1 に示したように、各タスクに  $k$  人を割り当てるまでに、漸進的に候補ワーカ集合から最近傍の  $k'$  ワーカをチェックする (8 行目)。 $k'$  は現在処理している割り当て  $A_l$  に残りの必要なワーカ数 - 候補ワーカ集合に残っているワーカ数である。即ち、 $k' = k - |A_l| - |CW|$  である。各往復において、最適なワーカを抽出し、暫定解

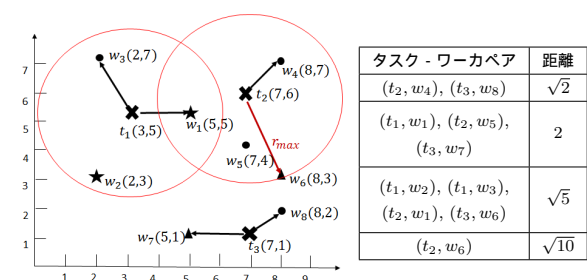


図 2 実行例

---

**Algorithm 1** DIVERSE-AWARE ASSIGN (DA)

---

```

1: function D-ASSIGN( $W, T, k$ )
2:    $\mathcal{A} = \{A_1, \dots, A_n\} \leftarrow \{\emptyset, \dots, \emptyset\}$ ;
3:   foreach  $t \in T$  do
4:     Sort  $W$  according to  $t$ ;
5:      $CW \leftarrow \emptyset$ ;           ▷ 候補ワーカーの集合を初期化
6:     while  $|A_t| < k$  do
7:        $BW \leftarrow NULL$ ;       ▷ 最適なワーカーを初期化
8:        $CW \leftarrow \text{select } k' \text{ workers diverse to } A_t \text{ in } W$ ;
9:        $BW \leftarrow \arg \min_{w \in CW} (lc_w)$ ;
10:      if  $BW \neq NULL$  then
11:         $A_t \leftarrow BW$ ;
12:        update  $W, CW$ ;
13:        ▷ 残りのワーカー集合と候補ワーカー集合を更新
14:      end if
15:    end while
16:  end for
17: end function

```

---

に入れる (9-11 行目). 最適なワーカーはコストの上限値は最も少ないワーカーである. コストの上限値は以下の式で計算される.

$$lc(w_i, t) = \text{dist}(w_{div}^r, t) \quad (4)$$

ただし,  $w_{div}^r$  は次に  $r$  番目の多様なワーカーである ( $r = k - |RA_t| - 1$ ). 多様性条件を考慮してコストの上限値を計算することは, 距離だけを考慮する貪欲な割り当て手法を緩和することができると考えられる.

### 5.3 厳密アルゴリズム

前節で記述した初期割り当て手法で得られた暫定解に基づき, 再割り当てを行い, 最適解を見つけることができる. 基本的なアイデアは, 最大距離であるタスクが処理したタスクであるまで, ローカル再割り当てを繰り返す. 各往復において, 暫定解に基づいて, フィルタ手法で再割り当てが必要があるタスクを抽出し, 改善した深さ優先探索を行い, 最適解を見つける.

#### 5.3.1 フィルタ手法

ローカル再割り当て段階では, 全体のタスクを再割り当てすると, ベースライン手法と同様に, 複雑度が非常に高い. そこで, 再割り当てする必要がないタスクをフィルタリングして, 最適解を探す. ただし, フィルタ手法では, 以下の用語を用いる.

**定義 4** 極大点は割り当て  $\mathcal{A}$  において, ワーカーとの距離が最も大きいタスクと定義する. ここでは,  $t_{\max}$  で表す. □

**定義 5** タスク  $t_i, t_j$  が与えられたとき,  $t_j$  の半径が  $r_{\max}$  とする範囲内に,  $t_i$  に割り当てされたワーカーが存在すると,  $t_i$  は  $t_j$  の関連点と呼ぶ. □

**定義 6** 極大点  $t_{\max}$  を関連チェーンに入れて, 関連チェーンに存在するタスクの関連点も関連チェーンに含まれる. □

図 2 に示した例では, 極大点が  $t_2$  であり,  $t_2$  の関連点が  $t_1$  なので,  $t_1$  を関連チェーンに入れる. ただし,  $t_1$  は関連点を持っていないので, 関連チェーンは  $\{t_2, t_1\}$  になる.

最大距離を最小化することを目指すため, 極大点と関連するタスク以外のタスクを再割り当てする必要はない. そこで, ローカル割り当て段階では, 関連チェーンしか再割り当てしない. 関連チェーンを抽出してから, アルゴリズム 2 を用いて, 最適解を見つける.

#### 5.3.2 改善した深さ優先探索

改善した深さ優先探索をアルゴリズム 2 に示す. 基本的には, 距離の小さい順でタスクとワーカーペアを並べて処理する (12,13 行目). INITCANDQUERY() では問合せを初期化して, 距離の小さい順でタスクとワーカーペアをソートする. NEXTCAND() は, 呼ばれるたびに次の候補を返す. 特に, 4 節で紹介した枝刈り基準を用いて, チェックする必要がない候補を削除する. つまり, 距離が  $r_{\max}$  以上となる候補は生成しない.

そして, 多様性のチェックを行い, 多様性の条件を満たすワーカーを割り当て結果に入れて, 再帰的に次の候補をチェックする (14,15 行目). また, 7 行目の関数 ISFULL() は,

$$\text{ISFULL}(\mathcal{A}) = \begin{cases} true & \text{if } \forall A_i \in \mathcal{A}, |A_i| = k \\ false & \text{otherwise} \end{cases} \quad (5)$$

という定義である. 全てのタスクに  $k$  ワーカーを割り当てると, 対応する割り当てを返す (9 行目). ISDIVERSE() は, 多様性をチェックする関数である.

ベースライン手法とは異なり, 改善した深さ優先探索では全ての割り当てを列挙することはない. 距離による枝刈りが行われるので, 現在の最良の割り当てより悪い割り当て

---

**Algorithm 2** IMPROVED-DFS

---

```

1: function FINDBESTASSIGN( $T, W$ )
2:    $\mathcal{A}_T \leftarrow \{\emptyset, \dots, \emptyset\}$ ,  $r_{\max} \leftarrow \infty$ ;
3:   DFS_SEARCH( $T, W, \mathcal{A}_T$ );
4: end function
5: function DFS_SEARCH( $T, W, \mathcal{A}_T$ )
6:   Mark  $\mathcal{A}_T$  as "checked";
7:   if IsFull( $\mathcal{A}_T$ ) then
8:     Recompute  $r_{\max}$  for  $\mathcal{A}$ ;           ▷ 全体の最大距離を更新
9:     output  $\mathcal{A}_T$ ;
10:    return;
11:   end if
12:   INITCANDQUERY( $T, W, \mathcal{A}_T, r_{\max}$ );
13:   for  $(w_*, A_*) \leftarrow \text{NEXTCAND}()$  do
14:     if ISDIVERSE( $A_* \cup \{w_*\}$ ) and  $\mathcal{A}_T$  is not "checked"
15:       then
16:         DFS_SEARCH( $T, W \setminus \{w_*\}, \mathcal{A}_T|_{A_* \leftarrow A_* \cup \{w_*\}$ );
17:       end if
18:     end for
19:   end function

```

---

ては選択肢から除かれる。

## 5.4 近似アルゴリズム

### 5.4.1 局所最適化割り当て手法

基本的なアイデアは初期割り当てによって得られた割り当て結果に基づき、局所最適化を行う割り当て手法である。対応する疑似コードをアルゴリズム 3 に示す。極大点  $t_{\max}$  が変わらなくなるまで極大点に対する最適化を行う (2-6 行目)。即ち、極大点の最大距離が小さくなくなるまで、再割り当てを行う。ただし、極大点以外のタスクの割り当ては変えずに、残りのワーカと現在極大点に割り当てられているワーカの集合を用いて、極大点の最適解を見つける (4 行目)。各往復において、最適化を行った後に、候補ワーカの集合と現在の極大点、また割り当て結果を更新する必要がある (5 行目)。

---

#### Algorithm 3 LO-ASSIGN

---

```
1: function LO-ASSIGN( $\mathcal{A}$ ,  $W'$ ,  $t_{\max}$ )
2:   repeat
3:      $t'_{\max} \leftarrow t_{\max}$ ;
4:     FINDBESTASSIGN( $t_{\max}$ ,  $W'$ );    ▷ 極大点を最適化
5:     update  $W'$ ,  $t_{\max}$ ,  $\mathcal{A}$ ;
        ▷ 候補ワーカの集合、極大点と割り当て結果を更新
6:   until  $t_{\max} = t'_{\max}$ 
7:   return  $\mathcal{A}$ ;
8: end function
```

---

### 5.4.2 交換に基づく割り当て手法

局所最適化割り当て手法の欠点としては、タスク間の影響を考慮しない点がある。例えば、図 2 に示した実行例に、初期割り当て段階で得られた暫定解では、 $w_1$  は  $t_1$  に割り当てられているが、極大点  $t_2$  に割り当てると、 $r_{\max}$  が小さくなる。そこで、適切な交換を行う必要があると考えられる。

初期割り当て段階で得られた暫定解に対して、以下の二つの交換方針を用いて、再割り当てを行う。

- 交換方針 1:  $r_{\max}$  を半径とする範囲内で、残りのワーカ (まだ割り当てされていないワーカ) から多様性制約を満たす最も近い一人を選び、最大距離である参加者  $w_{\max}$  を交換する。
- 交換方針 2:  $r_{\max}$  を半径とする範囲内で、ほかのタスクに割り当てされたワーカから一人を選び (例えば  $w_j$ )、ワーカ  $w_{\max}$  を置換える一方、 $w_j$  に割り当てされたタスクを交換方針 1 によって再割り当てし、コストが  $r_{\max}$  より小さい割り当てが存在するかどうかをチェックする。存在すると、交換を行う。

交換方針 1 と交換方針 2 を用いて得られる結果を比較し、より良いの割り当てを返す。

### • 補題 1

多様性閾値  $\tau$ 、割り当て  $A_i$  が与えられたとき、 $w_i \in A_i$  は交換できる場合、必ず次の条件を満たすワーカ  $w_j \in W$  が存在する:  $\text{sim}(w_i, w_j) \geq 2 \times \tau - 1$ 。

証明:

- (1)  $\text{sim}(w_i, w_j) < 2 \times \tau - 1$  を仮定する。
- (2) そこで、 $w_j$  と  $w_i$  間に異なる属性数は  $(2 - 2 \times \tau)|A|$  より多い ( $|A|$  は属性の総数)。
- (3) なお、 $\text{sim}(w_j, w_u) \geq 1 - \tau$ 、すなわち、 $w_j$  と  $w_u$  間に異なる属性数は  $\tau|A|$  より多い。
- (4) そこで、 $w_i$  と  $w_u$  間に同じである属性数は  $(2 - 2 \times \tau)|A| + \tau|A| - |A| = (1 - \tau) \times |A|$ 。すなわち、 $\text{sim}(w_i, w_u) > 1 - \tau$  であり、条件と矛盾する。
- (5) これにより、 $\text{sim}(w_i, w_j) \geq 2 \times \tau - 1$  が証明される。

多様性に基づくフィルタリング手法: 交換を行うとき、交換されるワーカとの類似度が  $2 \times \tau - 1$  より小さいワーカをチェックしない。そこで、割り当てにおける全てのワーカとの類似度を計算する必要はなく、計算量を下げることができる。

## 6. 関連研究

クラウドソーシングにおいて、人間の知識をうまく集めて、様々な問題を解決する研究は数多く存在する [2][7]。しかし、一種のクラウドソーシングである、空間情報に着目する参加型センシングにおけるタスク割り当て手法の研究は多くない。Leyla らはワーカの履歴情報に基づき、評価スコアを計算し、結果の有効性を満たす一方で、割り当てできるタスク数を最大化することに着目した割り当て手法を提案した [9][10]。[9] では、Maximum Task Assignment (MTA) 問題を定義している。それは、与えられた時間帯において、割り当てられたタスク数を最大化する問題である。そして、[10] は MTA 問題を拡張して、Maximum Correct Task Assignment (MCTA) を定義している。各タスクに対して、信頼度閾値と割り当て範囲が与えられ、タスクを割り当てる際に、割り当てされるワーカが割り当て範囲内であり、かつワーカ集合の正確率は信頼度閾値より高いという制約が定義されている。一方、各ワーカに割り当てできる最大タスク数の制約がある。これらの条件を満たすと、一部のタスクは割り当てされない場合もある。そこで、結果の正確性を保証するために、割り当てできるタスク数を最大化する問題になる。しかし、正確率のみでワカを評価しているので、単一なデータを収集することは避けられない。

空間データベースにおいて、既存の空間マッチング問題は [11][12] で研究されている。本研究と似ているは [11] で提案された (SPM-MM) 問題である。与えられたサービス

提供者集合  $P$  とカスタマー集合  $O$  に対して、サービス提供者の容量制約かつ顧客 (customer) の要求を満たすことに加え、最大マッチング距離を最小化する割り当て問題である。本研究で提案した問題と違い、SPM-MM 問題は多様性の制約を考慮していない。つまり、割り当てされるオブジェクト (顧客) 間の関係を考慮しない、この手法では、最大マッチング距離を最小化することに対して、正確度を保証する一方、効率性が高いスワップに基づく割り当て手法を適用する。

結果の多様性に関する研究は情報検索分野においてよく研究されている。類似するオブジェクトのリストを返すよりも、様々なオブジェクトからなるリストの方が有益な場合があると考えられている。多様性の定義は主に類似度、新規性、カバー率 (coverage) に基づく多様性の三つである [5]。一般性を考慮し、本研究では類似度に基づく多様性に着目する。一方、空間データベースにおける多様性と近接性を考慮した研究も存在する [8][6]。本研究で取り扱われる問題と近いのは KNDN (K Nearest Diverse Neighbors) 問題 [8] である。KNDN 問題の目的は、ユーザに十分な異なる結果を返すことを目指して、問合せ点に対する空間的に最も近接するサイズは  $k$  である完全多様な集合を見つけることである。ここでは、完全多様 (full-diverse) が次のように定義されている。結果集合  $A$  における任意の二つの点  $P_i, P_j$  が与えられ、多様性距離が閾値  $MinDiv$  以上であれば、点  $P_i$  と  $P_j$  が異なり、結果集合  $A$  は完全多様である。一方、結果集合とクエリ点の距離の平均値を用いて、空間的な近さが定義されている。ただし、問合せ点間の関係は考慮されておらず、単一の問合せ点に対して得られる結果の多様性を最大化する研究である。本研究では、複数の問合せ点に対する多様性を考慮した割り当て問題に着目し、最大マッチング距離を最小化することを目指して、有効性と効率性の両者を考慮したアプローチを提案している。

## 7. まとめと今後の方向

本論文では、空間クラウドソーシングにおいて、タスクを割り当てる際の重要な要素としてデータの質と空間コストを考慮した。多様性を考慮する上に、空間コストを最小化する割り当て問題を定義する。さらに、空間クラウドソーシングにおける効率的な割り当てフレームワークを提案した。割り当て手法としては、深さ優先探索に基づく厳密アルゴリズムを提案する一方、効率性を考慮した局所最適化割り当て手法と交換に基づく割り当て手法も併せて提案する。

今後の方向として、評価実験で主に次の2つを予定している。1) 提案手法の効率性と有効性について比較する。2) パラメータ  $k$  と  $\tau$  による影響について考察する。また、ワーカの時空間情報とプロフィール情報について、索引データ構造の改善、また多様性の定義の拡張などに取り

組みたいと考えている。今回は空間距離を第一順位として、アルゴリズムの構築に着目し、割り当てを行っている。今後は、プロフィール情報に対して、索引構造を改善することに着目し、より有効性と効率性とも高い割り当て手法を提案したいと考える。

謝辞 本研究の一部は、科研費 (25280039, 26540043) による。

## 参考文献

- [1] AMT: <https://www.mturk.com/mturk/welcome>.
- [2] C. C. Cao, J. She, Y. Tong and L. Chen. Whom to Ask? Jury Selection for Decision Making Tasks on Micro-Blog Services. *PVLDB*, 5(11), pp.1495-1506, 2012.
- [3] Chen, Z., Fu, R., Zhao, Z., Liu, Z., Xia, L., Chen, L., Cheng, P., Cao, C.C, Tong, Y. and Zhang, C.J. gMission: A General Spatial Crowdsourcing Platform, *PVLDB*, Vol.7, No.13, pp.1629-1632, 2014.
- [4] Konrad Dabrowski, Vadimzin, Haiko Muller and Dieter Raorithms for thependent Set Problem in Some Hereditary Graph Classes. *LNCS*, Vol.6460, pp.1-9, 2011.
- [5] Drosou, M. and Pitoura, E. Search result diversification, *SRecord*, Vol.39, No.1, pp. 41-47, 2010.
- [6] Ference, G., Lee, W., Jung, H. and Yang, D. Spatial search for K diverse-near neighbors, *Aiona Conference on Information and Knowledge Management, CIKM'13*, pp. 19-28, 2013.
- [7] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering Queries with Crowdsourcing. In *SIGMOD*, pp.61-72, 2011.
- [8] Konrad Dabrowski, Vadim Lozin, Haiko Muller and Dieter Rautenbach. Parameterized Algorithms for the Independent Set Problem in Some Hereditary Graph Classes. *LNCS*, Vol.6460, pp.1-9, 2011.
- [9] L. Kazemi and C. Shahabi. GeoCrowd: Enabling Query Answering with Spatial Crowdsourcing. In *ACM SIGSPATIAL GIS*, pp.189-198, 2012.
- [10] L. Kazemi, C. Shahabi and L. Chen. Geotrucrowd: Trustworthy Query Answering with Spatial Crowdsourcing. In *ACM SIGSPATIAL GIS*, pp.304-313, 2013.
- [11] C. Long, R. C. W. Wong, P. S. Yu, and M. Jiang. On Optimal Worst-Case Matching. In *ACM SIGMOD*, pp.845-856, 2013.
- [12] H. U. Leong, M. L. Yiu, K. Mouratidis and N. Mamoulis. Capacity Constrained Assignment in Spatial Databases. In *ACM SIGMOD*, pp.15-28, 2008.