参加型センシングのためのタスク割当て手法

趙　菁[†]　姜　仁河[†]　董テイテイ[†]　佐々木勇和[††]　肖　川[†††]
石川　佳治[†,††††]

† 名古屋大学大学院情報科学研究科
†† 名古屋大学未来社会創造機構
††† 名古屋大学高等研究院
†††† 国立情報学研究所
E-mail: †{zhao,jiang,dongtt,yuya}@db.ss.is.nagoya-u.ac.jp, ††chuanx@nagoya-u.jp,
†††ishikawa@is.nagoya-u.ac.jp

あらまし　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一種のクラウドソーシングである参加型センシング（Participatory Sensing）が注目されている．参加型センシングにおけるデータ収集では，観測地点への空間距離は参加者の意欲とタスクの完了時間に影響する一方，参加者の属性や背景知識が，収集されたデータの質や価値に影響する可能性もある．本研究では，データの質と空間コストの両方を考慮したタスク割当て手法について述べる．

キーワード　参加型センシング，空間データベース，多様性，アルゴリズム

1. はじめに

近年，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一種のクラウドソーシングである参加型センシング（Participatory Sensing）が注目されている [1]．参加型センシングでは，複数の参加者から得られるデータに基づき，個人あるいはコミュニティに役立つ知識を発見することを目指している．物理センサではなくて，参加者自身もセンサとなり，人間の知覚能力を用いて，様々なデータを収集することができる．例えば特定の場所におけるレビューや，天気に関する体感などの情報を収集することが挙げられる．

一般的なシステムフレームワークは次のようなものである．参加型センシング（PS）サーバが位置に関するタスクを近接する参加者に割り当て，参加者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．そして，観測データ・収集データなどを PS サーバに送信する．一方，PS サーバが活動管理者に求められたデータを送信する．

参加型センシングには，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．データの質は割り当てられた参加者の専門知識や興味などによって決まる．例としてレストランの評価を考える．参加者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，複数の多様な評価者に評価してもらうことが有効であると考えられる．一方，空間コストは参加者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，参加者の意欲も低くなる．そのため，タスクの割当てにおいては，多様性を持ち，空間コストも小さい参加者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る．各観測地点に似通っていない複数の参加者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，処理時間や参加者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．

2. 問題の定義

［定義 1］ **観測地点**（observation point）は空間上に分布する点として表され，参加型センシングでデータ収集の対象となる場所を表す．ここでは，n 個の観測地点からなる観測地点集合を $O = \{o_1, o_2, \ldots, o_n\}$ で表す．なお，本稿で使われた記号とその意味は表 1 に表されている．　□

［定義 2］ 参加型センシングに参加する意思のある人を **参加者**（participant）と呼ぶ．m 人からなる参加者集合を $P = \{p_1, p_2, \ldots, p_m\}$ で表す．参加者も空間上に位置する点と考える．　□

各観測地点に割り当てる参加者の人数を k とする．ユーザどうしの**非類似度**（dissimilarity）を求める $dsim()$ が与えられており，0 以上 1 以下の値を返すとする．ここで，参加者の観測地点集合への割当てを以下のように定義する．

［定義 3］ **割当て**（assignment）は，$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ で与えられる．ただし，$1 \leqq i \leqq n$ について $A_i \subset P$ かつ $|A_i| = k$ であり，$i \neq j$ なる $1 \leqq i, j \leqq n$ について $A_i \cap A_j = \emptyset$ が成り立つ．　□

割当てにはさまざまなものが考えられる．その全体集合を $U_{\mathcal{A}}$ で表す．

そこで，各観測地点に割当てした参加者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 4］ 各割当て A_i $(1 \leqq i \leqq n)$ において，任意の 2 名の参加者の非類似度が与えられた閾値 τ より大きいという制約

$$\forall p, p' \in A_i \text{ such that } p \neq p',\ dsim(p, p') > \tau \qquad (1)$$

を**多様性制約**（diversity constraint）と呼ぶ．ただし，$0 \leqq \tau \leqq 1$ である． □

様々な類似度の定義があるが，本研究では，類似度の一例として Jaccard Similarity を考える．ただし，この類似度を用いることは本質的ではなく，他の類似度にも対応できる Jaccard Similarity を考慮する．Jaccard Similarity は以下のとおりに定義される．

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

本研究では，各参加者に対してプロファイルが存在することを想定する．プロファイルは属性の集合であるとする．属性の全体集合から，参加者はプロファイルの属性を選択する．ここで例としてレストランの調査を行うための参加型センシングを考える．各参加者は，料理の集合（例：{ 和食, 中華料理, イタリア料理, フランス料理 }）から，自身が好みとする料理を選択してプロファイルを作成する．たとえば，参加者 p_a, p_b のプロファイルがそれぞれ $prof(p_a) =$ { 和食, 中華 }, $prof(p_b) =$ { 和食 , イタリア料理, フランス料理 } であるとする．このとき，参加者 p_a, p_b の類似度は $sim(p_a, p_b) = \frac{1}{4} = 0.25$ となる．

本研究が対象とする多様な k 近接割当て問題を，最適化問題として以下のように定義する．

［定義 5］ **多様な k 近接割当て問題**（diverse k-proximity assignment problem）を，

$$\begin{array}{ll} \text{Minimize} & \max_{i=1}^{n} \max_{p \in A_i} dist(o_i, p) \\ \text{subject to} & \text{diversity constraint} \end{array} \qquad (3)$$

と定義する．すなわち，多様性の制約を満たすという前提のもとで，観測地点と割当てられた参加者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，これがタスクの終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参加者の観測地点への距離が小さくても，全体の終了時間は遅くなってしまう．上記の定義はこのような点に着目している．一方で，参加者の労力に着目すれば，平均距離を小さくする割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．

多様な k 最近接割当て問題は，アルゴリズムの分野で知られる独立集合問題 [12] と関連している．**独立集合**（indepdent set）とは，与えられたグラフ $G = (V, E)$ における頂点の集合 $V^* \subseteqq V$ 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．各参加者をグラフの頂点とみなし，参加者 p, p' の間の非類似度 $dsim(p, p')$ の値が τ 以下であるときにのみ p, p' 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．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．

表 1　記号とその意味

記号	意味
n	観測地点の総数
m	参加者の総数
k	各観測地点に割当てする人数
O	観測地点集合：$O = \{o_1, o_2, \ldots, o_n\}$
P	参加者集合：$P = \{p_1, p_2, \ldots, p_m\}$
$\mathcal{A}$	割当て：$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$
$U_{\mathcal{A}}$	割当ての全体集合
$dsim()$	非類似度関数
τ	非類似度の閾値

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，観測地点の周辺の参加者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．割当て問題の処理時間は，アルゴリズムだけでなく，対象となる観測地点や参加者の分布やパラメータ（n, m, k, τ）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．

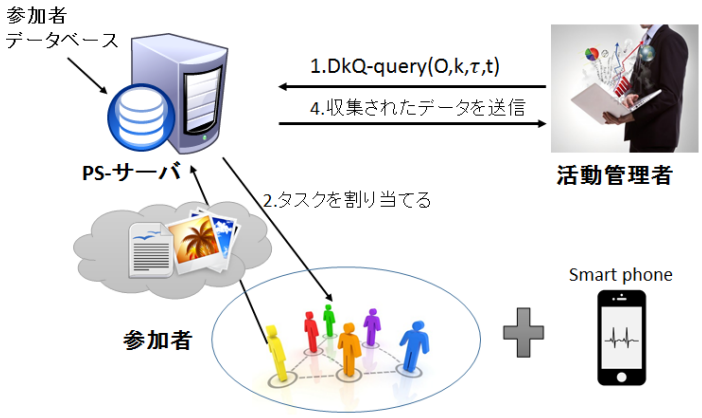

図 1　参加型センシングにおける問合せ処理フレームワーク

3. 割当て手法

本研究では図 1 のような参加型センシングにおける問合せ処理フレームワークを提案する．活動管理者がタスクの要求を含める多様な k 近接問合せ (DkPQ) を PS サーバに送信し，求めるデータを問合せする一方，最適な割当ても目指す．DkPQ において，観測地点集合 O，割り当てる参加者人数 k，多様性閾値 τ やタスクの内容 t が含められる．そして，PS サーバが参加者のプロファイルと位置情報に基づき，タスクを最適な参加者グループに割り当てる．参加者がタスクを受け取ってから，タスクの内容の通りに，物理的にその場所に訪問し，求めるデータを収集し，PS サーバに送信する．しかし，観測地点数・参加者の人数が多い場合最適な参加者グループを見つける計算量が

大きくなるため，本研究では効率的な問合せ処理アルゴリズムも併せて提案する．

3.1 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，深さ優先探索を用いて全ての可能な解を見つけることである．特に，枝刈り方針を用いて，効率を改善する．参加者を処理するとき，まず観測地点との距離を計算し，今までの最適解の最大距離より大きい場合，多様性などをチェックする必要はなく，この参加者を無視し，次の参加者をチェックする．

- **枝刈り基準１**: 観測地点との距離が今までの最適解の最大距離より大きい参加者を処理しない．
- **計算複雑度**: 最悪ケースは $O((n!)^m)$ である（n:参加者の総人数，m:観測地点数）．

3.2 改善手法（バックトラックに基づくアルゴリズム）

フィルタリング手法と枝刈り手法を用いて，ベースライン手法を改善したバックトラックに基づくアプローチを提案する．対応する疑似コードをアルゴリズム 1(BackTrack-basedAssign) に示す．主に以下の三つのステップがある．

- ステップ１（初期割当て）: 各観測地点に対して初期割当てを行い，一つの暫定解を見つける (7-10 行目).
- ステップ 2（再割当て）: フィルタリング手法 1 (15 行目) を用いて，再割当てする必要がある観測地点を抽出し，バックトラックに基づくアプローチ (アルゴリズム 2:BackTrack-Algorithm) により (16 行目)，再割当てを行い，これらの観測地点に対する最適解を見つける．
- ステップ 3（繰り返し）: 最大距離である観測地点 o_{max} が処理された観測地点になるまで，繰り返す．すなわち，最大距離が小さくならなくなるまで，再割当てを行う (11-17 行目).

ただし，アルゴリズムで用いている用語は以下のとおりである．

［定義 6］ **極大点 (Extreme Point)** は割当て集合 $\mathcal{A}$ において，参加者との距離が最も大きい観測地点と定義する．ここでは，o_{max} で表す． □

与えられた観測地点 o_i, o_j に対して，o_i と o_j の距離が全体の最大距離 min_{cost} と o_j の最大距離 o_{jmax} の和より小さいとき，観測地点 o_j は観測地点 o_i に関連すると呼ぶ．さらに，観測地点 o_i に関連する観測地点に関連する観測地点も観測地点 o_i に関連すると呼ぶ．これに基づき，次は関連観測地点を定義する．

［定義 7］ 極大点 o_{max} に関連する観測地点を**関連観測地点（Related Observation Point）**と呼ぶ． □

- フィルタリング手法 1

最適化段階において，極大点と関連観測地点集合しか処理しない．最大距離を最小化することを目指すため，極大点の割当てに関連しうる観測地点を抽出し，極大点と再割当てを行い，それ以外の観測地点を再割当てする必要はない．

- 枝刈り手法 2

割当て集合 $\mathcal{A}$ において，極大点 o の子孫ノードのほかの可能な解を見つける必要はない．

アルゴリズム 1 BackTrack-basedAssign

```
1: function BackTrack-basedAssign(U_o,U_p)
2:     min_cost ← ∞,RA ← {∅,...,∅};
         ▷ 大域変数：最小コスト (min_cost), 最終的な割当て (RA)
3:     o_max ← NULL,                          ▷ 大域変数：極大点
4:     dist_max ← {0,...0};▷ 大域変数：各観測地点に対する最大距離
5:     FO ← ∅, FP ← ∅;
         ▷ 大域変数：フィルタリングされた観測地点（参加者）集合
6:     P ← U_p;                          ▷ P を参加者の全集合になる
7:     foreach o ∈ U_o do
8:         InitialAssign(P, o, ∅);              ▷ 初期割当てを行う
9:         update P, o_max, min_cost, RA;
                          ▷ 候補参加者集合, 極大点と解集合を更新
10:    end for
11:    repeat
12:        if min_cost = FindBestAssign(o_max, U_p) then
13:            return RA;                        ▷ 枝刈り手法 2
14:        end if
15:        Filtering1(P);              ▷ 極大点と関連観測地点を抽出
16:        BaskTrack-Algorithm(FP, FP, FO, ∅);
                                              ▷ 最適化を行う
17:    until o_max ∈ FO
18:    return RA;
19: end function
```

極大点 o の子孫ノードの割当てでは，最大距離が極大点の最大距離より小さいので，どんなに改善しても，全体の最大距離を改善することはできない．そこで，これらの子孫ノードの他の可能な割当てを見つける必要はない．

対応する疑似コードはアルゴリズム 2 (BackTrack-Algorithm) において，割当て集合を見つけるとき，対応する極大点を更新し (15,16 行目)，再割当てするとき，極大点の子孫ノードのほかの解を見つけない（3-7 行目）．すなわち，観測地点を処理する前に，再割り当てる必要があるかどうかをチェックし，枝刈り手法 3 の条件を満たす場合，枝刈りを行う (3,4 行目).

先に述べたとおり，多様な k 近接割当て問題計算コストの高い問題である．本研究では，効率性を考慮し，二つの近似アルゴリズムを提案する．

3.3 近似アルゴリズム 1 (Greedy-based Local-Optimal Assignment)

基本的なアイデアは貪欲（greedy）な局所最適化割当て手法である．対応する疑似コードをアルゴリズム 3 (Greedy-based-LO-Assign) に示す．主に以下の三つのステップがある．

- ステップ１（初期割当て）: 各観測地点に対して初期割当てを行い，一つの暫定解を見つける (7-10 行目).
- ステップ 2（局所最適化）: 極大点に対する最適化である．極大点以外の観測地点の割当ては変えずに，極大点に対して，残りの参加者と現在割当てされている参加者集合から，最適解を見つける．
- ステップ 3（繰り返し）: 極大点 o_{max} が変わらなくなるまで繰り返す．すなわち，最大距離が小さくならなくなるまで，

アルゴリズム 2 BackTrack-Algorithm

```
1: function BackTrack-Algorithm(P, restp, O, R)
2:     foreach p ∈ P do P' = P \ {p};
3:         if o.Id ≠ o_max and o_max ≠ NULL then
4:             break;                                  ▷ 枝刈り手法 2
5:         else
6:             o_max ← NULL;
7:         end if
8:         if dist(o, p) ≧ min_cost then
9:             break;
       ▷ 枝刈り手法 1：全体の最大距離より大きい参加者は処理されない
10:        end if
11:        if |R| = ∅ or sim(p, R) ≦ 1 − τ then
12:            R' ← R ∪ {p};
13:            if |R'| = k then
14:                O' ← Q \ {O[0]}, update restp;
                          ▷ O': 残りの観測地点と参加者集合を更新
15:                if |O'| = 0 then
16:                    update RA, min_cost, o_max;
17:                else
18:                    sort P' according to distance to O'[0];
19:                    BackTrack-Algorithmn(restp, restp, O', ∅);
                                                ▷ 次の観測地点を処理
20:                end if
21:            else
22:                BackTrack-Algorithm(P', restp, O, R');
                                              ▷ 次の参加者をチェック
23:            end if
24:        end if
25:    end for
26: end function
```

再割当てを行う (11-15 行目).

3.4 近似アルゴリズム 2 (Swap-based Local-Optimal Assignment)

この局所最適割当て手法は交換方針を用いて，再割当てを行い，結果を改善する．基本的なアイデアは近似アルゴリズム 1 と同じく，三つのステップがあるが，ステップ 2 の局所最適化段階において，以下の二つの交換方針を用いて，再割当てを行う．対応するアルゴリズムをアルゴリズム 4 (Swap-based-ReAssign) に示す.

- 交換方針 1: 最大距離の範囲内で，残りの参加者（まだ割当てされていない参加者）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 p_{max} を交換する．
- 交換方針 2: 最大距離の範囲内で，関連観測地点に割当てされた参加者から一人を選び (例えば p_j)，参加者 p_{max} を交換する一方，p_j に割当てされている観測地点を交換方針 1 によって再割当てし，最適解を見つける．

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，最適解を返す.

3.4.1 フィルタリング手法 2

多様性制約をチェックする計算量を減らすために，補題 1 に基づくフィルタリング手法 2 を提案する．フィルタリング手法 2 を用いて，全ての割当て集合における参加者との類似度を計算する必要はない.

アルゴリズム 3 Greedy-based-LO-Assign

```
1: function Greedy-based-LO-Assign(U_o, U_p)
2:     min_cost ← ∞, RA ← {∅, ..., ∅};   ▷ 大域変数：最小コスト (min_cost), 最終的な割当て (RA)
3:     o_max ← NULL,                              ▷ 大域変数：極大点
4:     dist_max ← {0, ..., 0};
                          ▷ 大域変数：各観測地点に対する最大距離
5:     RP ← ∅;                          ▷ 大域変数：残りの参加者の集合
6:     P ← U_p;                          ▷ P を参加者の全集合とする
7:     foreach o ∈ U_o do
8:         InitialAssign(P, o, ∅);                ▷ 初期割当てを行う
9:         update RP, o_max, min_cost, RA;   ▷ 候補参加者集合, 極大点と解集合を更新
10:    end for
11:    repeat
12:        o'_max ← o_max;
13:        FindBestAssign(o_max, RP);          ▷ 極大点を再割当てす
14:        update RP, o_max, min_cost, RA;
                          ▷ 候補参加者集合, 極大点と解集合を更新
15:    until o_max = o'_max
16:    return RA;
17: end function
```

- **補題 1**

多様性閾値 τ が与えられ，集合 $P_I = \{p_u, ..., p_v\}$ において，任意の二つのオブジェクト間の非類似度が τ より大きい場合，集合 P_I を独立グループと呼ぶ．そして，$p_i \in P_I$ を p_j と交換したとき，グループ P_I はまた独立グループであった場合，オブジェクト p_j は必ず次の条件を満たす：$sim(p_i, p_j) \geqq 2 \times \tau - 1$.

証明：

（1） $sim(p_i, p_j) < 2 \times \tau - 1$ を仮定する．

（2） そこで，オブジェクト p_j と p_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， $sim(p_j, p_u) \geqq 1 - \tau$， すなわち， p_j と p_u 間に異なる属性数は $\tau|A|$ より多い.

（4） そこで， p_i と p_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，$sim(p_i, p_u) > 1 - \tau$ であり, 条件と矛盾する.

（5） これにより，$sim(p_i, p_j) \geqq 2 \times \tau - 1$ が証明される．

- 補題 1 の変換

独立グループ $P_I = \{p_u, ..., p_v\}$，閾値 τ が与えられ，参加者 p_j に対して，$sim(p_i, p_j) \leqq 2 \times \tau - 1$ の場合, p_i を交換することはできない.

そこで，多様性制約をチェックする計算量を減らすために，補題 1 に基づくフィルタリング手法 2 を提案する．フィルタリング手法 2 では，再割当て段階において，参加者 p_{max} との類似度が $2 \times \tau - 1$ より小さい参加者はフィルタリングされる．対応する疑似コードは，アルゴリズム 4（Swap-based-ReAssign）の 9 行目の類似度の制約条件である．フィルタリング手法 2 を

用いて，全ての割当て集合における参加者との類似度を計算する必要はないので，計算量を下げることができる．

アルゴリズム 4 SWAP-BASED-REASSIGN

```
1: function SWAP-BASED-REASSIGN(RP)
2:     R' ← R \ {p_max};                          ▷ R: o_max の結果集合
3:     AP ← RA \ R_{o_max};
          ▷ AP: 既にほかの問合せ点に割り当てられた参加者の集合
4:     p_min ← NULL
                        ▷ 大域変数：p_max に交換される最適な参加者
5:     REASIGN1(RP, o_max);                       ▷ 再割当て方針 1
6:     Sort AP according to o_max;                ▷ 再割当て方針 2
7:     foreach p ∈ AP do
8:         sim ← sim(p_max, o)        ▷ 候補参加者と p_max の類似度
9:         if dist(p, o_max) < min_cost and sim ≧ 2×τ−1 then
   ▷ フィルタリング手法 2
10:            if sim(p, R') ≦ 1 − τ or sim = 1 then
11:                REASIGN1(RP ∪ {p_max}, p.assignedo);
12:                if dist(p, o_max)≧ cost_{p.assignedo} then
13:                    update min_dist, p_min;
                   ▷ 再割当て案の最大距離と交換できる参加者を更新
14:                    break;
15:                else
16:                    if cost_{p.assignedo} < min_dist then
17:                        update min_dist, p_min;
18:                    end if
19:                end if
20:            end if
21:        end if
22:    end for
23:    if p_min ≠ NULL then
24:        R_{o_max}.erase(p_max);
25:        R_{o_max}.push_back(p_min); ▷ p_min を p_max に置き換える
26:    end if
27:    update RA, min_cost, o_max;                ▷ 結果を更新
28: end function
```

4. 評価実験

4.1 比較手法

比較実験は有効性と効率性について，三つの提案手法とベースライン手法の四つで行った．ベースライン手法を Baseline で表し，提案した正確な結果を保証するバックトラックに基づくアルゴリズムと二つの近似アルゴリズムをそれぞれ Exact method, Greedy-based と Swap-based で表す．

各提案手法に対して，ソート初期割当てを用いて比較を行う．参加者を観測地点との距離の小さい順で処理して，暫定解を見つける手法はソート初期割当てである．

4.2 パラメータとデータサイズ

パラメータとデータサイズを変化させ，三つの評価実験を設計した．パラメータとデータサイズの設定を表 2 に示す，実験 1 では，小規模なデータを用いて三つの提案手法と Baseline を比較する．データサイズが大きい場合，Baseline は実行できないため，小規模なデータに対し四つの手法の有効性と効率性を比較し，大規模データについては Greedy-based と Swap-based を比較する．

表 2　パラメータとデータサイズの設定

	実験 1		実験 2
$\|P\|$	100,500	2k,20k	1k, 10k
$\|O\|$	1-100	10-5k	10,100
k	3	3	2-12
τ	0.5	0.5	0.1-0.5

4.3 実験データ

実験データとして，人工データおよびソーシャルネットワーク Gowalla からの実データを用いる [2]．実データは時空間情報を含めるユーザのチェックインデータである．様々な調査 [14] で示されているように，モバイルデバイスの普及に従って，参加型センシングの参加者の人数も急速に増えているので，本実験では，タスク数よりも参加者の人数が非常に多いことを仮定する．つまり，割当てできないタスクは存在しないことを仮定する．ここでは，参加者として 196,585 人のユーザのデータを用いて，タスクに 10,000 個のデータを用いる．参加者のプロファイルは人工データを用いる．各参加者は 20 個の属性を持ち，各属性の値は 0 あるいは 1 であるとし，ランダムに生成する．

4.4 実験 1: データサイズを変化させた場合

まず，小規模データに対し，Baseline と三つの提案手法の処理時間を比較する．図 2 は，観測地点数の変化による効率性を示したものである．図 2 では，表 5.2 のパラメータの値のもとで，参加者数は 100 としている．観測地点数が増えるに従って，ベースライン手法の処理時間が急速に増加する．観測地点が 2 以上である場合，ベースライン手法は有効時間内に結果が出てこないが，三つの提案手法の処理時間とも短い（5 秒以内）．

一方，有効性については，図 3 に示すとおり，参加者の人数が 500 である場合，観測地点数が増えるにつれ，対応する最大距離の値も若干増え，Greedy-based と Swap-based の結果と最適解はほとんど同じであり，有効性が高い（誤差は 0.1 未満である）．

次に，大規模データに対し，Greedy-based と Swap-based の効率性を比較する．参加者数が 20,000 である場合の Greedy-based と Swap-based の処理時間を図 4 に示す．観測地点数が増えるに従って，二つの近似アルゴリズムとも効率性が高い．

図 5 と図 6 は，参加者数の変化による効率性と有効性を示したものである．参加者観測地点数が 10 の場合は，参加者の人数が増えるにつれ，Exact の処理時間も増えるが，Greedy-based と Swap-based の処理時間はあまり変わらない．また，Greedy-based と Swap-based の結果と最適解はほとんど同じであり，有効性が高い．

4.5 実験 2：パラメータ k と τ の影響

この部分では，パラメータ k（各観測地点に割当てされる参加者の人数）と多様性閾値 τ の変化に対する，効率性と有効性について，提案手法を比較する．

データサイズが大きい場合，二つの近似アルゴリズムに対し

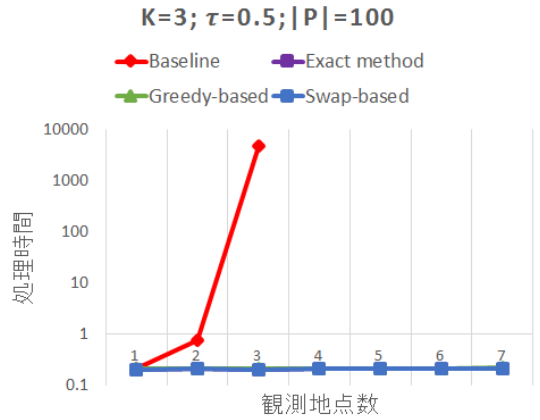

図 2　観測地点数に対する効率性

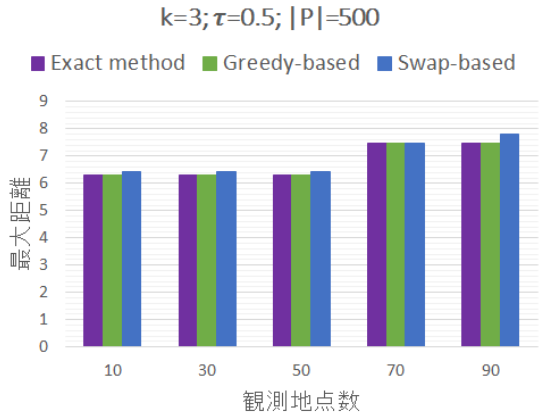

図 3　観測地点数に対する有効性

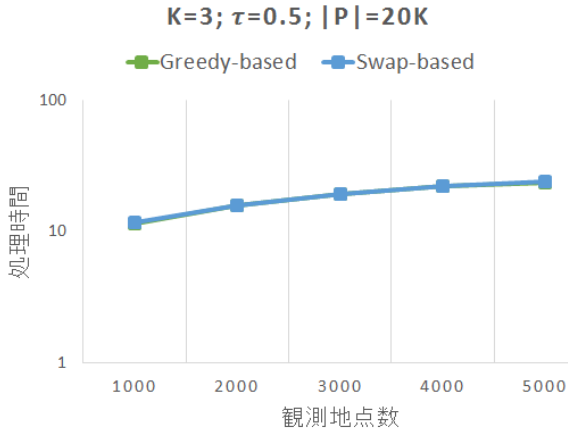

図 4　観測地点数に対する効率性

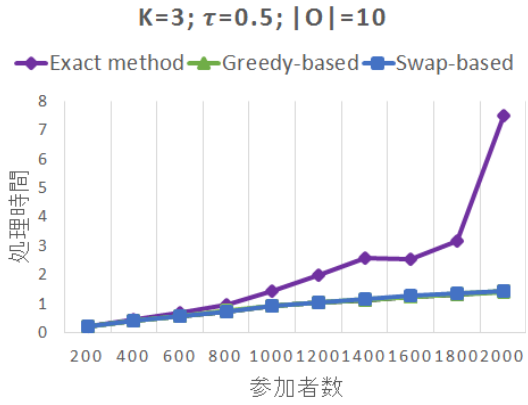

図 5　参加者数に対する効率性

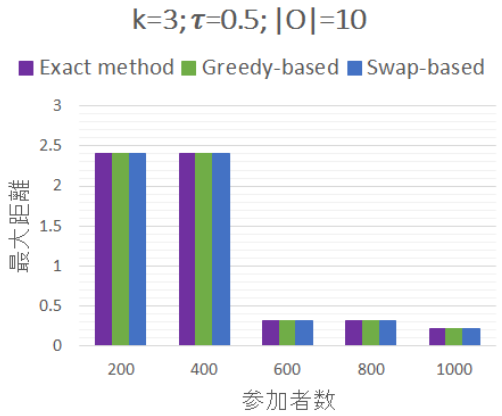

図 6　参加者数に対する有効性

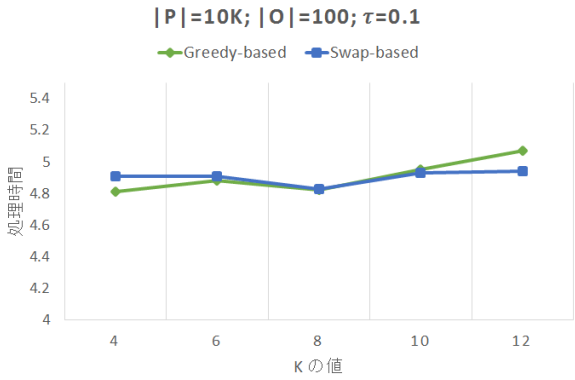

図 7　k に対する効率性

て，k と τ の値に対する処理時間の比較を図 7 と図 8 に示す．Greedy-based と Swap-based とも効率性が高い．

一方，有効性については，図 9 で示したとおり，Greedy-based の有効性は高く，Swap-based の有効性は高くない．Swap-based においては，k が大きいほど，交換条件がきつくなるので，交換できる参加者が少なくなる．そこで，結果の有効性に影響する．

これまで，多様性閾値 τ は固定値 0.5 であった．次に，異なる閾値 τ に対する k の影響を図 10 に示す．k を増やすにつれ閾値 τ を減らすので，多様性制約は緩くなり，交換条件も緩くなるため，Swap-based の有効性も高くなる．

観測地点の密度が高いとき，閾値 τ の変化に対応する有効性については，図 11 に示す．多様性閾値 τ が 0.4 以下の場合には，観測地点の密度が高いほど，お互いに影響している観測地点が多くなる．Greedy-based では，割当てされていない参加者を候補参加者として最適化を行うので，お互いに影響している観測地点の状況を取り扱わないが，Swap-based における再割当て方針 2 はこのような状況を処理できる．そこで，観測地点の密度が高いほど，Greedy-based より，Swap-bsed の有効性も高くなる．

4.6　実験のまとめと考察

今回の実験で，効率性，有効性とパラメータの影響について，三つの提案手法と Baseline の比較を行った．実験結果に基づき，次のような二つの結論をまとめる．

（1）　データサイズの変化

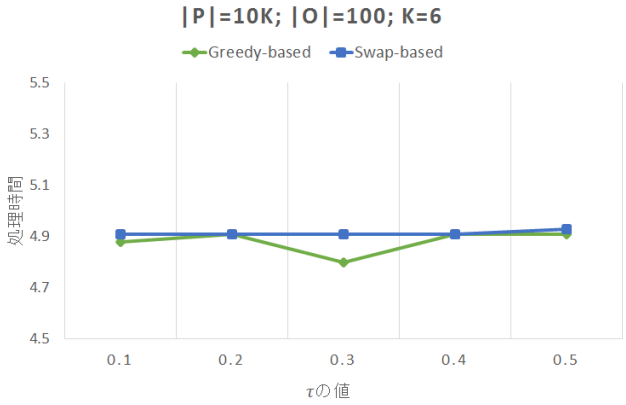

図 8　τ に対する効率性

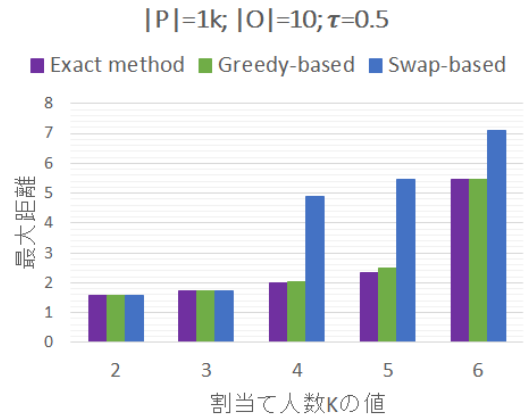

図 9　k に対する有効性

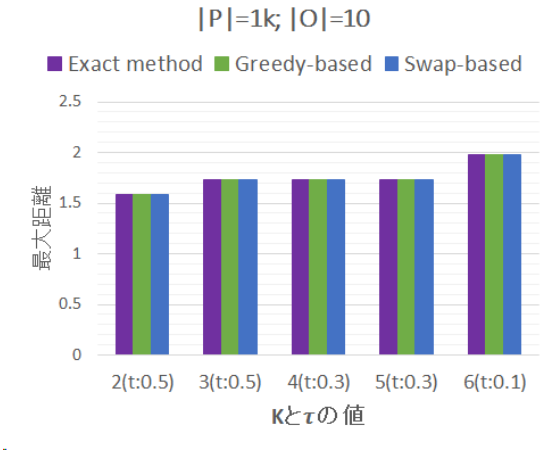

図 10　異なる τ についての k に対する有効性

観測地点数と参加者の人数が増えるに従って，Baseline より，三つの提案手法の効率性は高くなる．そして，データサイズが大きくなると，Exact method よりも二つの近似アルゴリズムとも効率がよく，有効性も正確な解とほぼ同じである．

（2）　Greedy-bsed と Swap-based の比較

Greedy-bsed の有効性は観測地点と参加者の分布に影響される．一方，Swap-based の有効性は交換条件に影響される，つまり，多様性閾値 τ と各観測地点に割当てされる参加者数 k に影響される．なお，多様性制約に対して，k の値よりも，τ の値の影響がより明らかである．

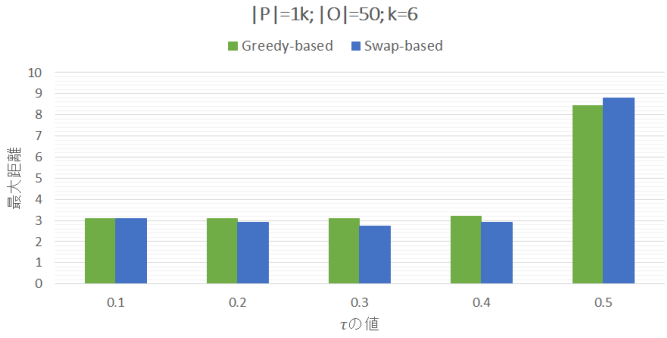

図 11　τ による有効性

そのため，観測地点の密度が高く，多様性の条件が緩い（閾値 τ, k の値が小さい）場合には，Swap-based の方が有効性が高い．一方，観測地点の密度が低く，多様性の条件がきつい場合には，Greedy-bsed の方が有効性が高い．

5.　関 連 研 究

5.1　参加型センシングにおけるタスク割当て手法

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 [8] [9]．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当て手法を提案した [3] [4]. [3] 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, [4] 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，信頼度閾値と割当て範囲が与えられ，タスクを割り当てる際に，割当てされるワーカが割当て範囲以内であり，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，正確率のみでワー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．

5.2　空間マッチング問題

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 [6] [7] で研究されている. [6] では，Capacity Constrained Assignment (CCA) 問題を提案している．容量（capacity）の制約を考慮して，全体として最適な割当てを見つける問題である. [7] で提案された問題は Spatial Matching for Minimizing Maximum matching distance (SPM-MM) 問題と呼ばれる．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当て手法を適用する．

5.3 結果の多様性

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．

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 [10] [11]. [11] では，空間データを閲覧する問合せにおいて，多様性を持つ結果を入手するために，Angular 類似度を定義している．与えられた問合せ点に対して，結果の関連性と多様性を最大化する k 個のオブジェクトを見つけることを目指す．本研究で取り扱われる問題と近いのは KNDN (K Nearest Diverse Neighbors) 問題 [10] 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た．さらに，IG (Immediate Greedy) と BG (Buffered Greedy) 二つの貪欲的な手法が提案された．

これらの研究においては，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ことを目標とするアプローチがほとんどである．しかし，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する．

6. 議論と今後の課題

本論文では，参加型センシングにおいて，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当て問題に着目し，バックトラック探索に基づく有効性を考慮した割当て手法を提案する一方，効率性を考慮した近似アルゴリズム（局所最適化に基づく割当て手法）も併せて提案した．

本研究では，式 (3) のような最適化として問題を与えたが，現実的な制約として「参加者の移動距離は ρ 未満でなければならない」という制約をさらに追加することも考えられる．現実的な応用を考えた場合，移動距離に上限を設けることには妥当性がある．探索空間が狭まることになるので，計算時間の向上にもつながる．

今後の課題として，参加者の時空間情報とプロファイル情報について，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，割当てを行っている．今後は，プロファイル情報に対して，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）による．

文　　献

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy and M. B. Srivastava. Participatory Sensing. In First Workshop on World-Sensor-Web (WSW'06) (ACM Sensys Workshop), pp.117-134, 2006.

[2] Gowalla.www.wikipedia.org/wiki/Gowalla.

[3] L. Kazemi and C.Shahabi.GeoCrowd: Enabling Query Answering with Spatial Crowdsourcing. In ACM SIGSPATIAL GIS, pp.189-198, 2012.

[4] L. Kazemi, C. Shahabi and L. Chen. Geotrucrowd: Trustworthy Query Answering with Spatial Crowdsourcing. In ACM SIGSPATIAL GIS, pp.304-313, 2013.

[5] Drosou Marina and Pitoura Evaggelia. Search Result Diversification. SIGMOD Rec., Vol.39, No.1, pp.41-47, 2010.

[6] H. U. Leong, M. L. Yiu, K. Mouratidis and N. Mamoulis. Capacity Constrained Assignment in Spatial Databases. In *ACM SIGMOD*, pp.15-28, 2008.

[7] C. Long, R. C. W. Wong, P. S. Yu, and M. Jiang. On Optimal Worst-Case Matching. In *ACM SIGMOD*, pp.845-856, 2013.

[8] 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.

[9] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering Queries with Crowdsourcing. In *SIGMOD*, pp.61—72, 2011.

[10] Jayant R. Haritsa. The KNDN Problem: A Quest for Unity in Diversity. In IEEE Data Eng. Bull, Vol.32, No.4, pp.15-22, 2009.

[11] O.Kucuktunc and H.Ferhatosmanoglu. λ-Diverse Nearest Neighbors Browsing for Multi-dimensional Data. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol.25, No.3, pp.481-493, 2013.

[12] Konrad Dabrowski, Vadim Lozin, Haiko Mller and Dieter Rautenbach. Parameterized Algorithms for the Independent Set Problem in Some Hereditary Graph Classes. LNCS, Vol.6460, pp.1-9, 2011.

[13] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In ACM SIGMOD, pp.47-57, 1984.

[14] S. Tilak. Real-World Deployments of Participatory Sensing Applications: Current Trends and Future Directions. In *Int. Scholarly Research Notices for Sensor Networks*, 2013.