クラウドソーシングを用いた Skyline ポイントの収集に関する研究

筑波大学
図書館情報メディア研究科
2016 年 3 月
平木 理恵

目次

第 1 章

はじめに

データポイント集合が与えられたとき，この集合中のデータポイントのうち，他のデータポイントに支配されないもの (Skyline ポイントと呼ぶ) の集合を検索する問合せは Skyline 問合せ [1] と呼ばれ，これまで数多くの研究が行われてきている．データポイントの集合 $D = \{d_1, d_2, d_3\}$ が図 1.1 のリレーションの各タプルで表現されているとする．このとき，D において，属性の集合 $A = \{$ 価格, 雰囲気 $\}$ に関してある $d_i \in D$ が支配されないとは，d_i の**全ての** A 中の属性の値が優れているようなデータポイントが D 中に存在しない (すなわち，$\nexists d_j \in D$ s.t. $\forall a_s \in A(d_j \succeq_{a_s} d_i) \wedge \exists a_t \in A(d_j \succ_{a_t} d_i)$) ことである．価格に関しては小さい方が，雰囲気に関しては大きい方が優れているとすると，ホテル B はホテル A に支配されることになる．また，ホテル A，C は他のどのホテルにも支配されない．したがって，この Skyline 問合せの解はホテル A と C である．

本論文では，既存の Skyline 問合せの研究と異なり，既知のデータポイントが存在しない状態から，マイクロタスク型クラウドソーシングを用いて Skyline ポイントの収集を行う問題を扱う．マクロタスク型クラウドソーシングとは，比較的短時間で処理可能なタスク (マイクロタスク) をネットワークを通じて不特定多数の人々 (ワーカ) に処理してもらうことである．この問題は多くのアプリケーションを持つ．なぜなら，現実に Skyline ポイントを求めたい状況において，必ずしもデータポイント集合が全て既知であり，かつ必要な属性の情報が全て揃っているとは限らないからである．例えば，つくばの洞峰公園に近くパスタが美味しい店を探したいときに，$A = \{$ 洞峰公園からの近さ, パスタのおいしさ $\}$ という属性を持つデータポイント集合は我々が知る限り存在しない．

この問題を解くための単純な手法は，まず全てのデータポイントをクラウドソーシングで収集し，次に既存の Skyline 問合せ処理手法を適用することである．しかし，それでは膨大なデータポイントの収集タスクが一般に必要になる．

本論文では，より少ないタスク数で Skyline ポイントを収集するための手法について提案する．基本的なアイデアは，全てのデータポイントを収集するためのマイクロタスクを用意するのではなく，中間結果にあるデータポイントから，より Skyline に近いポイントを収集するためのタスクを利用することである．

	ホテル名	価格	雰囲気	駅からの距離
d_1	A	7,000	4	100m
d_2	B	8,000	3	150m
d_3	C	9,000	5	50m

図 1.1　リレーション Hotel

本論文の構成は次の通りである．

2 章では関連研究について説明する．3 章では Skyline ポイント収集について定義する．4 章では提案手法について説明する．5 章では提案手法を用いた実験について説明する．6 章では本論文全体のまとめと今後の課題について述べる．

第 2 章

関連研究

Skyline 問合せに関して，クラウドソーシングを利用する研究がこれまでも行われてきた [9]．具体的には，データポイントは DB に格納されているが，Skyline のための属性が必ずしも格納されていない場合の処理の研究が行われている．例えば，図 1.1 の各タプルで表されるホテルのデータポイントの集合 D が存在するとする．このとき，D において，属性の集合 $A = \{$ 価格, 部屋の大きさ $\}$ に関して支配されないある $d_i \in D$ を検索することを考える．このとき，データポイントの集合 D には属性「部屋の大きさ」が存在せず，このデータポイント集合から Skyline 問合せを行うことはできないため，クラウドソーシングを用いてその値を収集するというアプローチが考えられる．論文 [9] はその際のタスク数をできるだけ少なくしようとするものである．一方，我々が提案する手法は，既知のデータポイントが存在しないもしくは既知のデータポイントに Skyline ポイントが含まれない状態からマイクロタスク型クラウドソーシングを用いて Skyline ポイントの収集を行う．

Skyline 問合せ以外に関してもクラウドソーシングによる問合せ処理の研究は数多くある．例えば，選択 (filterling)，結合 (join)，ソート (sort) などのデータベース演算の処理が知られており，これらの演算についての効率化の議論が行われてきた [5][6][7][8]．その中に，機械では検索困難な情報を人手で検索する人力検索 (Human-powered Search) がある．Human-powered Search の特徴は，データを絞る際に元となる対象のデータが，計算機に格納されているデータ等に限定せず，データを収集することである．Human-powered Search の処理は，多くのシステムで利用されている [2][3][4]．本論文で扱う Skyline ポイントの収集は，一種の Human-powered Search であると言える．

第 3 章

定義

本章では，Skyline ポイント収集について定義する．

定義 1 (支配)**.** d_k，d_l をそれぞれデータポイントとし，これらが持つ属性の *(*部分*)* 集合を $A = \{a_1, a_2, ..., a_m\}$ とする．このとき，d_k が A に関して d_l を**支配する**ことを $Dominates_A(d_k, d_l)$ と表記し，次のように定義する．

$$Dominates_A(d_k, d_l) \equiv \forall a_s \in A\ (d_k \succeq_{a_s} d_l) \wedge \exists a_t \in A\ (d_k \succ_{a_t} d_l)$$

定義 2 (Skyline ポイント)**.** データポイントの集合を $D = \{d_1, d_2, ..., d_n\}$ とし，これらのデータポイントが持つ属性の *(*部分*)* 集合を $A = \{a_1, a_2, ..., a_m\}$ とする．このとき，あるデータポイント d_k が D の A に関する *Skyline* ポイントであることを $Skyline_{D[A]}(d_k)$ と表記し，次のように定義する．

$$Skyline_{D[A]}(d_k) \equiv \nexists d_l \in D\ (Dominates_A(d_l, d_k))$$

定義 3 (Skyline ポイント収集)**.** *(*暗黙の*)* データポイントの集合を $D = \{d_1, d_2, ..., d_n\}$ とし，これらのデータポイントが持つ属性の *(*部分集合*)* を $A = \{a_1, a_2, ..., a_m\}$ とする．このとき，*Skyline* ポイント収集とは，D の部分集合 $S = \{d_k | d_k \in D, Skyline_{D[A]}(d_k)\}$ を求めることである．

第 4 章

提案手法

本章ではまず，提案手法の概要について説明する．次に，タスクデザインについて説明し，このタスクデザインによって Skyline ポイントの収集が行えることを証明する．最後に，効率的な Skyline ポイントの収集方法について説明する．

4.1 提案手法概要

本節では，マイクロタスク型クラウドソーシングを用いた Skyline ポイントの収集手法について説明する．提案手法の入力は，次の通りである．

- 求めるデータポイントの質問文
- Skyline のための属性の集合

これらが与えられると，本手法はシードポイント収集タスク (以下，シードタスク) と，Skyline ポイント収集タスク (以下 Skyline タスク) をもちいて Skyline ポイントの収集を行う．図 4.1 に，提案手法の概要を示す．具体的には，シードタスクを行い，Skyline ポイント収集の元となるデータポイントを収集する．次に，収集したデータポイントを元に，Skyline タスクを用いてより Skyline に近いデータポイントを収集する．Skyline タスクを繰り返し行うことで，より Skyline に近いデータポイントを収集する．

本論文では，Skyline タスクのための 2 つのタスクデザインを検討する．

4.2 タスクデザイン

本節では，提案手法で用いたシードタスクと Skyline タスクについて説明する．特に，Skyline タスクでは，2 つのタスクデザインを検討する．以下では，次の入力が行われた例を利用して説明する．

- 質問文：“つくばのレストランで洞峰公園から近くカルボナーラが安い店を探したい”
- 属性集合：｛「洞峰公園からの距離」,「カルボナーラの値段」｝

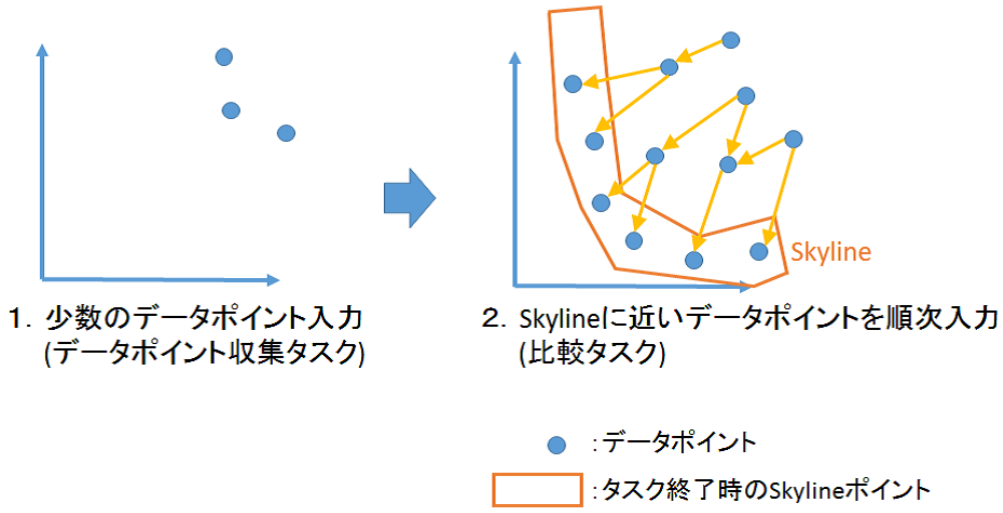

図 4.1　提案手法の概要

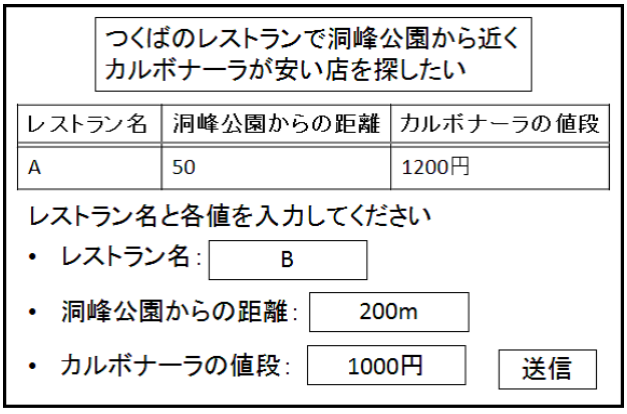

図 4.2　シードタスクの画面

4.3　シードポイント収集タスク

シードタスクでは，リクエスタが指定した質問文に従い，データポイントを入力する．図 4.2 はシードタスクの画面例である．図にあるように，シードタスクには，入力として与えられた質問文が表示され，ワーカは，Skyline のための属性を入力する．図 4.2 では，ワーカはレストラン名として「B」，洞峰公園からの距離として「$200m$」，カルボナーラの値段として「1000 円」と入力している．

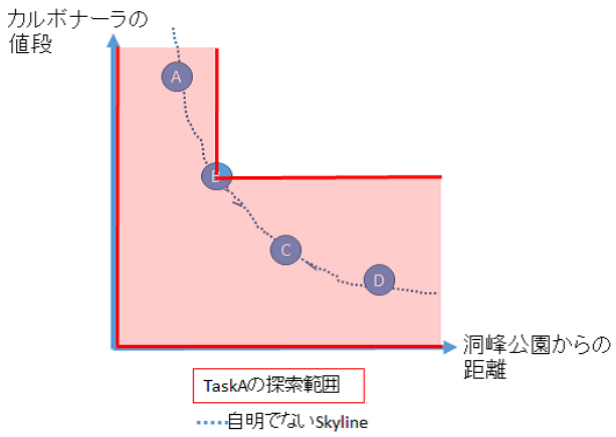

図 4.3　タスクデザイン A の探索範囲

4.4　Skyline ポイント収集タスク

本論文では，Skyline タスクに関して，2 つのタスクデザインを検討する．これらのタスクデザインの違いは，Skyline ポイントの探索範囲の違いである．

タスクデザイン A. タスクデザイン A(図 4.4) は，提示されたデータポイント (**基準データポイント**と呼ぶ) に対して，少なくとも一つの属性において優位なデータポイントを入力するものである．図 4.4 のタスク画面で表形式で表示されているのが基準データポイントである．このタスクを TaskA と呼ぶ．ワーカは，基準データポイントに対して，少なくとも一つの属性において優位なデータポイントを入力する．

図 4.4 のタスクを説明する．まず，質問文として，最初に入力された “つくばのレストランで洞峰公園から近くカルボナーラが安い店を探したい”が表示されている．基準データポイントは B であり，ワーカは，「洞峰公園からの距離」または「カルボナーラの値段」の少なくとも一方において優位なデータポイントの入力を行う．ここでは，ワーカは，カルボナーラの値段が優位なレストラン C をデータポイントとして入力している．図 4.3 は，ワーカが探索するデータポイントの範囲を示す．

タスクデザイン A の十分性を示す根拠は次の二つの定理である．

定理 1 (データポイント間の関係 1)**.** ある D，A に対して，$\exists d_i \in D(Skyline_{D[A]}(d_i))$ とする．このとき，$\exists a_k \in A(d_j \succ_{a_k} d_i)$ となるような ***Skyline*** ポイント d_j が存在可能である．

証明： 定義 1 により，$\exists a_k \in A(d_j \succ_{a_k} d_i)$ である時，$Dominates_A(d_i, d_j)$ が成立しない．したがって，$Skyline_{A[D]}(d_i)$ と $Skyline_{A[D]}(d_j)$ は両立しうる．　□

定理 2 (データポイント間の関係 2)**.** ある D，A に対して，$\exists d_j \in D(Skyline_{D[A]}(d_j))$ とする．こ

図 4.4　TaskA: 提示されたデータポイントに対して，少なくとも一つの属性において優位なデータポイントを入力

のとき，$\nexists a_k \in A(d_j \succ_{a_k} d_i)$ であるようなデータポイント d_i は存在しない．

証明： 上記の条件を満たす場合，定義 1,2 により，$d_i = d_j$ もしくは $Dominates_A(d_i, d_j)$ が成立する． □

タスクデザイン B. タスクデザイン B では 2 つのタスク Task B1, Task B2 を用いて Skyline ポイントの収集を行う．これは，タスクデザイン A に比べて，探索範囲を狭めたものである．

Skyline に関する属性集合 $A = \{a_1, a_2\}$ とする．Task B1 は，現時点の Skyline 上で隣り合う 2 つのデータポイント d_i, d_j (ただし $d_i \succ_{a_1} d_j$) を基準ポイントとし，d_j より a_1 に関して優位で，かつ，d_i より a_2 に関して優位なデータポイントを収集する．Task B2 は，現時点の Skyline 上の端にあるデータポイント d_i を基準ポイントとして生成し，そのデータポイントが他と比べて最も優位な属性 (図 4.7 の A の場合には洞峰公園からの距離) に関して，d_i より優位なデータポイントを収集する．

Task B1　図 4.5 では, 基準データポイントとしてレストラン B とレストラン C (B$\succ_{\text{洞峰公園からの距離}}$C) が提示されており，レストラン C よりも洞峰公園からの距離が短く，かつ，レストラン B よりもカルボナーラの値段が安いデータポイントの入力を求めている．この例では，ワーカは，レストラン E をデータポイントとして入力している．

Task B2　図 4.6 では，基準データポイントとして，レストラン A が表示し，レストラン A より洞峰公園からの距離が優位なデータポイントの入力を求めている．A は，現時点の Skyline 上の端のに位置しており，属性「洞峰公園からの距離」が他のデータポイントより優位なデータポイントである．この例では，ワーカは，レストラン F をデータポイントとして入力している．

4.4.1　暗黙の属性値の扱い

Skyline ポイントの収集のための属性に関して，具体的な値を入力するのが難しい場合がある．例えば，「店の雰囲気」や「パスタのおいしさ」に関しては，その味が 3 なのか 5 なのか，難しい．ま

図 4.5　TaskB1 : 2 つの基準データポイントより優位なデータポイントを入力するタスク

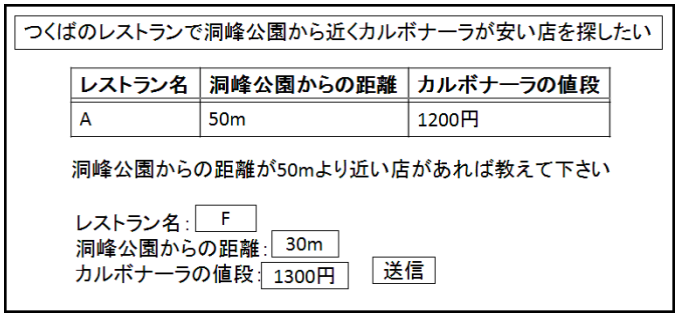

図 4.6　TaskB2 : 基準データポイントより指定された属性において優位なデータポイントを入力するタスク

た，自然文の自由入力にした場合，「良い」なのか「やや良い」なのか，表現に関しても統一は困難である．本提案手法では，Skyline タスクにおいて，そのような属性値に関して基準データポイントより優位か優位でないかを選択式で選ぶことにより，このような暗黙の属性値を扱う事が出来る．

具体例を図 4.8 に示す．これまでと同じく，質問文は “つくばのレストランで洞峰公園から近くパスタがおいしい店を探したい”である．暗黙の属性値を入力させる場合，タスクの基準データポイントの提示部分に明示的な属性値は表示しない．また，新たなデータポイントの属性の入力フォームは，「より良い」「より悪い」「どちらでもない」の 3 択ラジオボタンを用意する．この例では，新たなレストラン C のパスタのおいしさとして「より良い」が選ばれている．

Skyline 収集の処理の際には，パスタのおいしさを数値化する．具体的には，比較結果を推移的に適用し，各ポイントよりもおいしさが劣ると判断されたデータポイントの数を，パスタのおいしさとする．

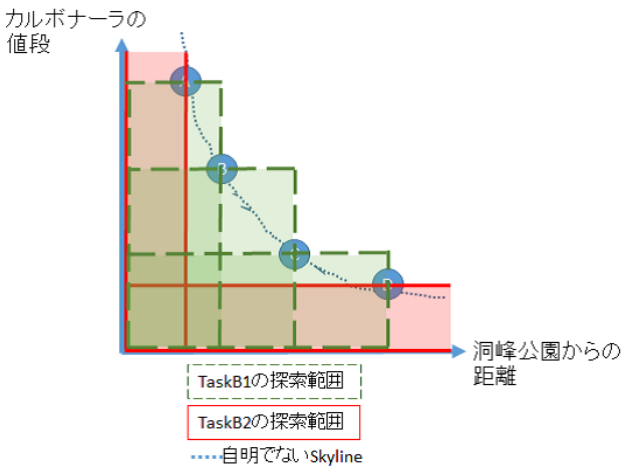

図 4.7　タスクデザイン B の探索範囲

つくばのレストランで洞峰公園から近くパスタがおいしい店を探したい

レストラン名	洞峰公園からの距離
B	200m

次の条件を少なくとも1つ以上満たすレストランを教えて下さい
・洞峰公園からの距離が200mより近い
・Bよりパスタがおいしい

レストラン名: C
洞峰公園からの距離: 400m
パスタのおいしさ: より良い ☑ より悪い ☐ どちらでもない ☐

送信

図 4.8　感性的な属性を用いる場合のタスク例

4.4.2　Skyline ポイントの収集の効率化

本節では，Skyline タスクを利用した Skyline ポイントの収集の効率化について議論する．具体的には，考えられる Skyline タスクの中から適切なタスクを選んでワーカに提示することにより，効率化をはかる．簡単化のため，2 次元のケースを例に議論するが，より高次元でも同じ議論が可能である．

4.4.2.1　タスクデザイン A の効率化

4.2 節のタスクデザイン A の TaskA では，これまで収集したデータポイントのいずれを基準ポイントとしても良い．しかし，どのデータポイントを基準ポイントとして選ぶかは効率化に影響をもたらすと考えられる．例えば，効率良く Skyline ポイントにたどり着くためには，各時点で収集済みのデータポイント中の Skyline ポイントを基準点とすることが必要と考えられる．そこで，本節で

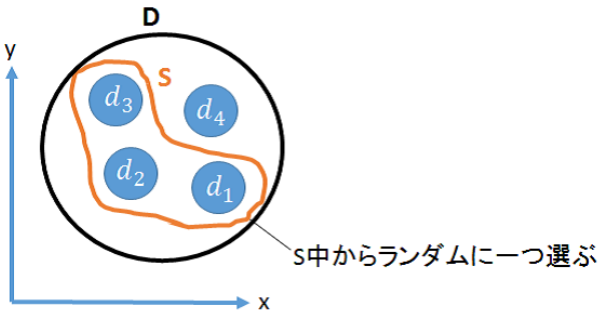

図 4.9　ランダム戦略

は，各時点での Skyline ポイントを基準ポイントとして選択する際の戦略について議論する．

まず，最も単純な戦略は**ランダム戦略**である．これは，既に収集したデータポイント中の，その時点での Skyline ポイントからランダムに一つ選ぶ戦略である．例えば，図 4.9 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ と D 中の Skyline ポイントの集合 $S = \{d_1, d_2, d_3\}$ があった場合，S 中からランダムに一つ選択することである．

しかし，ランダム戦略は必ずしも効率が良いとは限らない．より少ないタスク数で Skyline ポイントの収集を行うために，タスク選択順序に関して次のような戦略が考えられる．

面積最小タスク選択戦略　この戦略は，「より Skyline に近いデータポイントを基準にタスクを行った方が，より早く Skyline ポイントにたどり着ける可能性が高い」というヒューリスティクスに基づいている．既に収集したデータポイントから探索範囲が最小となるものを選ぶ戦略である．正確な探索範囲の把握は難しいため，Skyline は反比例の式 $xy = \alpha$ で表せると仮定する． この戦略では，α が小さいデータポイントを選択する．例えば，図 4.10 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ があった場合，D 中から原点に近いデータポイントを一つ選択する．図 4.10 では d_2 となる．

支配数期待値最大タスク選択戦略　この戦略は，「多くのデータポイントを支配する期待値が高いタスクを行った方が，より早く Skyline ポイントにたどり着ける可能性が高い」というヒューリスティクスに基づいている．各タスクで支配できるデータポイント数の期待値を計算し，期待値が高いタスクを優先的にワーカに割り当てる．例えば，図 4.11 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ があった場合，d_2 を基準データポイントとするタスクの期待値を考える．まず，実線で囲まれた支配数計算領域 (X_i, Y_j) に新たにデータポイントが入力された時に支配できるデータポイント数を数える．支配数計算領域 (X_4, Y_4) に新たにデータポイントが入力された場合，4 つのデータポイント (d_1, d_2, d_3, d_4) を支配することができる．支配数計算領域のアルゴリズムの詳細は，4.4.3 節で説明する．

これにより，各タスクの結果に関して，支配データポイント数の期待値 (支配期待値) を計算

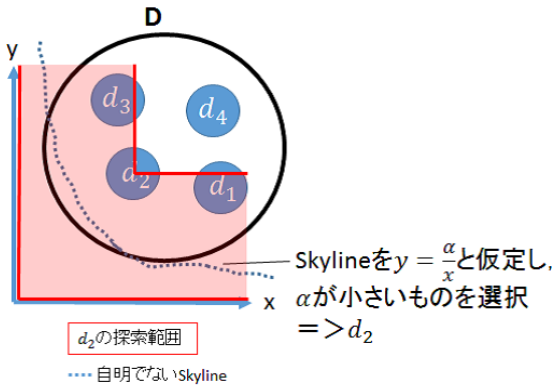

図 4.10　面積最小タスク選択戦略

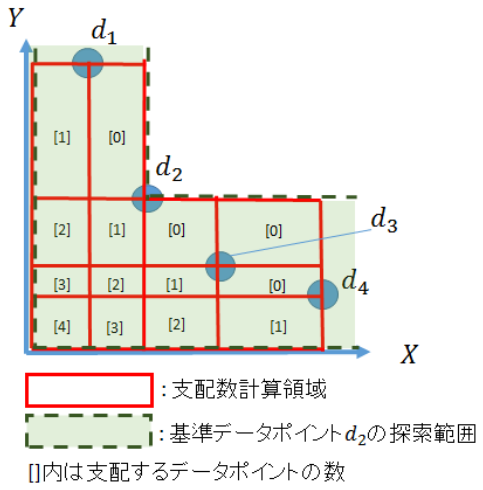

図 4.11　支配数期待値最大タスク選択戦略

する事が出来る．支配期待値は，各タスク (の基準データポイント) が規定する探索範囲に含まれる支配数計算領域の支配数の合計を，探索範囲の面積で割った値である．

定理 3 (戦略の一致)**.** タスクデザイン A において，面積最小タスク選択戦略は支配数期待値最大タスク選択戦略と一致する．

証明： タスクでデザイン A において，どの Skyline ポイントを基準データポイントとして選択しても全ての支配数計算領域が含まれる．したがって，面積最小タスク選択戦略が支配期待値を最大化する． □

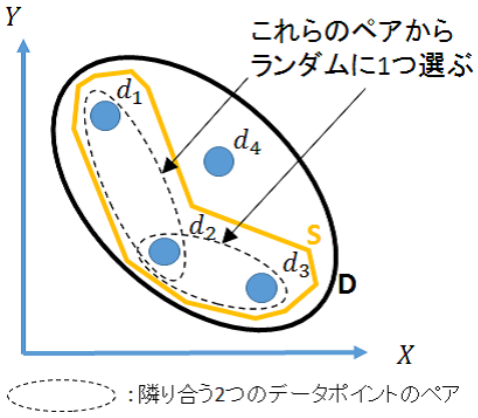

図 4.12　ランダム戦略

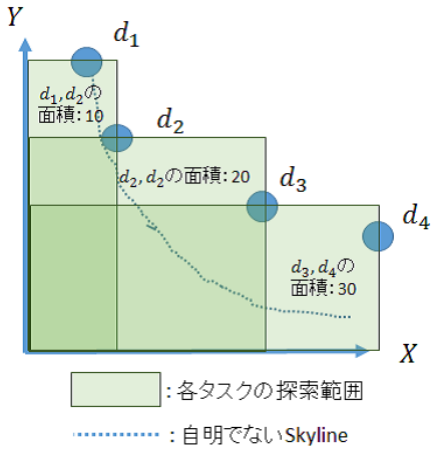

図 4.13　面積最小タスク選択戦略

4.4.2.2　タスクデザイン B の効率化

本節ではタスクデザイン B の TaskB1 のタスク順序について議論する．タスクデザイン A の場合と同様に，3 つの戦略を考えることが出来る．

ランダム戦略　既に収集したデータポイント中の Skyline ポイントから隣り合う 2 つのデータポイントのペアをランダムに一つ選ぶ戦略である．例えば，図 4.12 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ と D 中の Skyline ポイントの集合 $S = \{d_1, d_2, d_3\}$ があった場合，S 中の隣り合うデータポイントのペアからランダムに一つ選択する．

面積最小タスク選択戦略　タスクデザイン A と異なり，タスクデザイン B では探索範囲の面積を

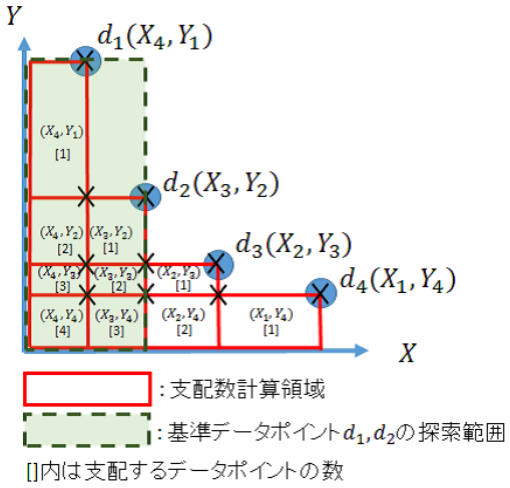

図 4.14　支配数期待値最大タスク選択戦略

簡単に計算できる．例えば，図 4.13 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ があった場合，データポイント d_1, d_2 を基準データポイントとするタスクが面積最小となるので，このタスクを選択する D 中から原点に近いデータポイントを一つ選択する．

支配数期待値最大タスク選択戦略　タスクデザイン A の場合と同様に，各タスクで支配できるデータポイント数の期待値を計算し，期待値が高いタスクを優先的にワーカに割り当てる．例えば，図 4.14 のようなデータポイントの集合 $D = \{d_1, d_2, d_3, d_4\}$ があった場合，各タスクの探索範囲に含まれる支配数計算領域の数の和を求め，タスクの面積で割ると支配期待値が最も大きいのは d_1, d_2 を基準データポイントとするタスクであるので，これらを基準データポイントとするタスクを選択する．

4.4.3　支配期待値計算アルゴリズム

図 4.15 を用いてアルゴリズムの説明を行う．データポイントの集合を $D = \{d_1, d_2, ..., d_{n-1}, d_n\}$，属性の集合を $A = \{X, Y\}$ とする．まず，各データポイントに各属性を優位な順に並べた順序に従ってラベル付けを行う．各属性のラベルは，属性名に順序に従った添え字をつけて表す．例えば，d_1 ならば $d_1(X_n, Y_1)$ というようにラベル付けを行う．属性名の添え字はその属性において順序が最後尾のものを 1 とし，順序が優位になるに従って添え字を 1 ずつ大きくしていく．

次に，各データポイントの属性値から支配数計算領域 (以降，領域と呼ぶ) を列挙する．この領域は各データポイントから各属性の軸に対して垂直に直線を引いた時にできる各格子である．領域の右上部の × 印の点を識別子とする．この点は，各データポイントのラベルのペアで表現できる．

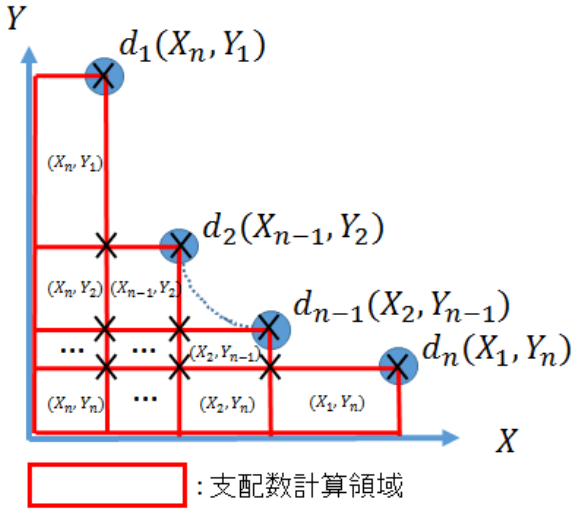

図 4.15　支配数期待値最大タスク選択戦略

領域を列挙したら，各領域にデータポイントが新たに入力された場合に支配できるデータポイント数を算出する．各領域 (X_i, Y_j) の支配できるデータポイントの集合を $DominatedPoints(X_i, Y_j)$ と書く．これは，D 中のデータポイント (X_k, Y_l) のうち，$k \leq i \wedge l \leq j$ であるもの全てとして計算できる．例えば，$DominatedPoints(X_4, Y_2) = \{(X_4, Y_1), (X_3, Y_2)\}$ となる．したがって，領域 (X_4, Y_2) の支配値は 2 である．

第 5 章

実験

本章では，4 章で説明したタスくデザイン A，タスくデザイン B を用いて行ったシミュレーション実験について説明する．提案手法において次のことを評価する．(A) より少ないタスク数で Skyline ポイントを収集できるか．(B) データの性質の違いによって，適した手法が異なるのか．この 2 つを確認するため，2 種類の実験を行った．各実験とその結果の概要は次の通りである．詳細については次節より説明する．

(1) 実データを用いたシミュレーションによる実験．これまで我々が行ってきたの研究の中で，少数のデータポイントを持つデータによる予備実験を行ってきた．本論文では，多くの Skyline 問合せの研究の実験で使われる NBA 選手データを用いて，多数のデータポイントを持つデータについての実験を行った．実験は，ワーカが本来行うタスクの処理をシミュレーションし，比較した．その結果，タスクデザイン B に支配数期待値最大タスク選択戦略を適用した場合が，もっとも少ないタスク数で Skyline ポイントを収集できた。

(2) 人工データを用いたシミュレーションによる実験．正の相関、負の相関、無相関の人工データを用いて，(1) と同様にシミュレーションを行い比較した．その結果，どのデータにおいても，タスクデザイン B に支配数期待値最大タスク選択戦略を適用した場合がもっとも少ないタスク数で Skyline ポイントを収集できた。

5.1　実データを用いたシミュレーションによる実験

本節では，提案手法を評価するために，実データを用いてシミュレーションを行い，比較した．

5.1.1　実験概要

実験では，タスクデザイン A,B に効率化のための各戦略を適用した場合に，より少ないタスク数で Skyline ポイントを収集できたかを比較するために，提案手法によるシミュレーションを行った．提案手法の入力と使用したデータは次の通りである．

- (入力 1) 質問文：“NBA 選手の中で出場試合数が少なくかつ得点数が多い選手が知りたい”
- (入力 2) 属性の集合：「出場試合数」,「得点」
- データ：NBA 選手のデータ．属性として出場試合数と得点を持つ．データ数は 21959 個，この内 Skyline ポイントの数は 27 個である．

作業手順まず，NBA 選手データのからどのデータポイントも支配しないデータポイントをシードポイント収集タスクで得たデータポイントとする．このデータポイントを基準データポイントとし，各タスクデザインに従ってシミュレーションを行う．今回，4.4.2.1 節において述べたように，タスクデザイン A における面積最小タスク選択戦略と支配数期待値最大タスク選択戦略は一致するため，タスクデザイン A は支配数期待値最大タスク選択戦略のシミュレーションは行わない．
評価方法より少ないタスク数で Skyline ポイントを収集することができるかを評価するために，実験結果から再現率を求める．ここで，A を NBA 選手データ中の Skyline ポイント集合とし，この集合 A が正解集合となる．B_i をタスクが行われた時系列順で，i 番目のタスク終了時の Skyline ポイントの集合とする．この時の再現率を以下の様に求める．

$$\textbf{再現率}：Recall(A,\ B_i) = \frac{A \cap B_i}{A}$$

5.1.2　実験結果

本節では，シミュレーション結果より得られた再現率について説明する．

5.1.2.1　タスクデザイン A

タスクデザイン A にランダム戦略，面積最小タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.1 に示す．ランダム戦略では 40091 タスク目，面積最小タスク選択戦略では 32478 タスク目に再現率 1 となった．

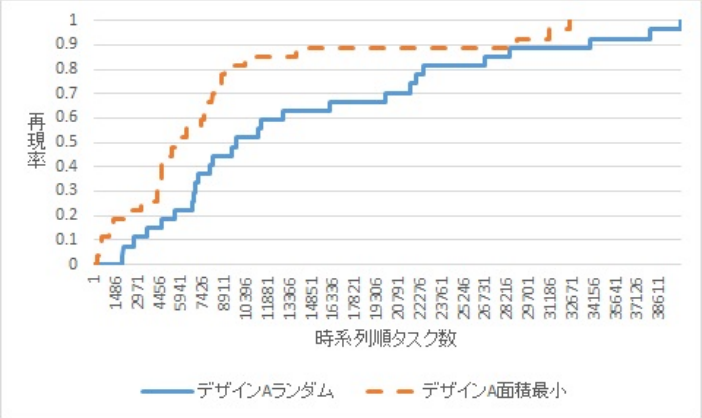

図 5.1　実データ実験：タスクデザイン A の再現率

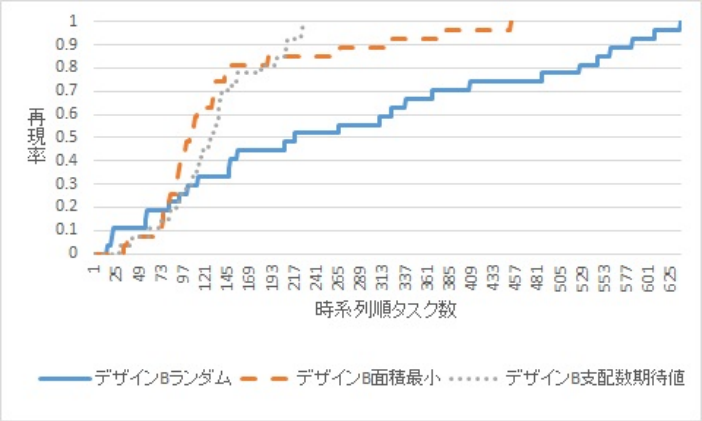

図 5.2　実データ実験：タスクデザイン B の再現率

5.1.2.2　タスクデザイン B

タスクデザイン B にランダム戦略，面積最小タスク選択戦略，支配数期待値最大タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.2 に示す．ランダム戦略では 635 タスク目，面積最小タスク選択戦略では 452 タスク目，支配数期待値最大タスク選択戦略では 227 タスク目に再現率 1 となった．

5.1.3　考察

5.1.2 節で示した結果より，考察を行う．タスクデザイン B の支配数期待値最大タスク選択戦略が最も少ないタスク数で再現率 1 の結果を得た．また，タスクデザイン A とタスクデザイン B の結果を比較すると，タスクデザイン A よりタスクデザイン B の方がより少ないタスク数で再現率 1 と

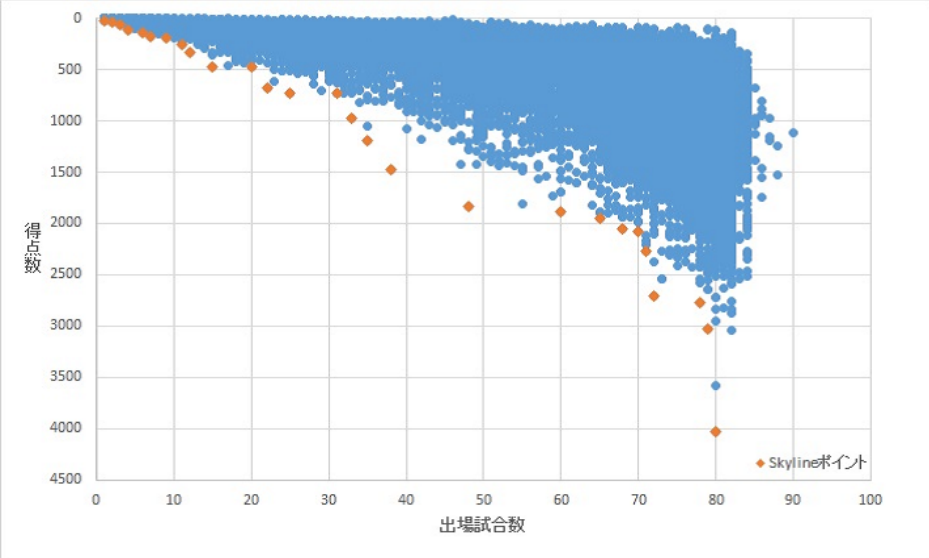

図 5.3　実データ実験に用いたデータの散布図

なった．これは，タスクデザイン A に比べてタスクデザイン B は探索範囲が狭いため，既に入力されているデータポイントを重複して入力する回数がタスクデザイン A に比べて少なかったためである．

全てのデータポイントを収集し，Skyline 問合せを行う単純手法でこの問題を扱う場合，データポイントの数だけデータポイントを収集するタスクが必要となるため，単純手法では 21959 タスク必要である．タスクデザイン A は単純手法に比べて，ランダム戦略，面積最小タスク選択戦略に比べ，より多くのタスクを必要とした．タスクデザイン A のタスク数が膨大となった原因はデータの性質によるものと考えられる．今回実験に使用したデータの散布図を 5.3 に示す．左下に近いほど優れているデータポイントである．データポイントは右上に点在しており，タスクデザイン A の多くのタスクで，既に入力されているデータポイントに支配されるデータポイントが入力されていた．その結果，タスク数が増えてしまったと考えられる．

タスクデザイン B の各タスク戦略を比較すると，面積最小タスク選択戦略より支配数期待値最大タスク選択戦略の方がより少ないタスク数で再現率 1 となった．これは，支配数期待値が大きいタスクを優先して行うことで，各タスク処理時点でより多くのデータポイントを支配し，探索範囲面積を絞ることができたためである．その結果，より Skyline に近いデータポイントを入力することができた．

5.2　人工データを用いたシミュレーションによる実験

本節では，データの性質によって適した手法が異なるのかを確認するために，人工データを用いてシミュレーションを行い，比較した．

5.2.1　実験概要

実験では，タスクデザイン A,B に効率化のための各戦略を適用した場合に，データの性質によって適した手法が異なるのかを確認するために，人工データを用いてシミュレーションを行い，比較する．人工データは 3 種類用いた．これらのデータは 2 次元のデータであり，各データは 1000 個のデータポイントからなる．使用したデータと Skyline ポイントの数は次のとおりである．

- 正の相関を持つデータ．この内 Skyline ポイントの数は 2 個である．
- 負の相関を持つデータ．この内 Skyline ポイントの数は 24 個である．
- 無相関のデータ．この内 Skyline ポイントの数は 9 個である．

作業手順まず，データ内のどのデータポイントも支配しないデータポイントをシードポイント収集タスクで得たデータポイントとする．このデータポイントを基準データポイントとし，各タスクデザインに従ってシミュレーションを行う．5.1 節と同様に，タスクデザイン A における面積最小タスク選択戦略と支配数期待値最大タスク選択戦略は一致するため，タスクデザイン A は支配数期待値最大タスク選択戦略のシミュレーションは行わない．
評価方法より少ないタスク数で Skyline ポイントを収集することができるかを評価するために，実験結果から再現率を求める．ここで，A をデータ中の Skyline ポイントとし，この集合 A が正解集合となる．B_i をタスクが行われた時系列順で，i 番目のタスク終了時の Skyline ポイントの集合とする．この時の再現率を以下の様に求める．

$$\textbf{再現率}：Recall(A, B_i) = \frac{A \cap B_i}{A}$$

5.2.2　実験結果

本節では，シミュレーション結果より得られた再現率について説明する．

5.2.2.1　正の相関を持つデータ

タスクデザイン A. タスクデザイン A にランダム戦略，面積最小タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.4 に示す．ランダム戦略では 25 タスク目，面積最

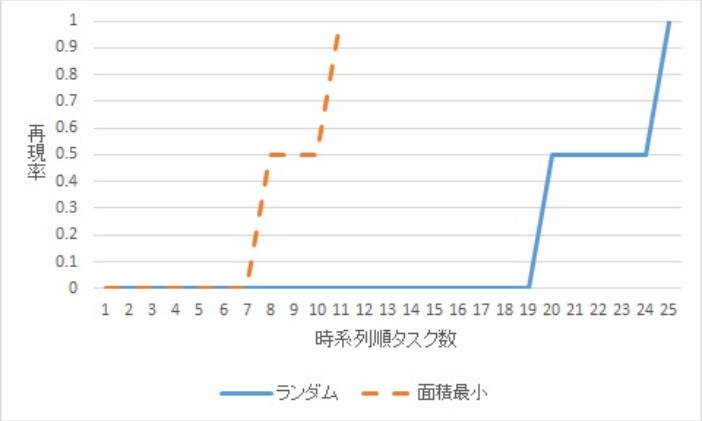

図 5.4　正の相関：タスクデザイン A

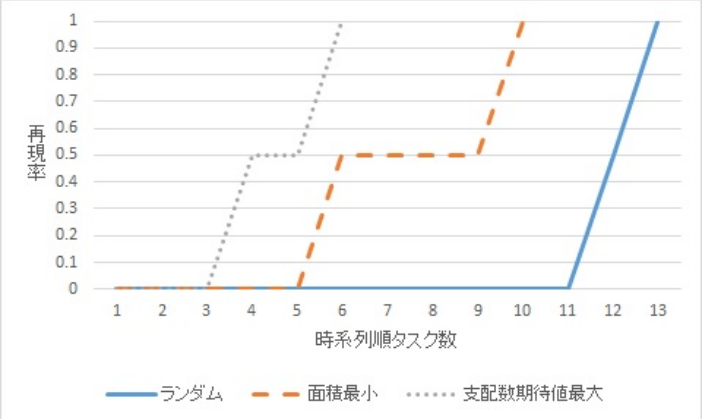

図 5.5　正の相関：タスクデザイン B

小タスク選択戦略では 11 タスク目に再現率 1 となった．

タスクデザイン B. タスクデザイン B にランダム戦略，面積最小タスク選択戦略，支配数期待値最大タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.5 に示す．ランダム戦略では 13 タスク目，面積最小タスク選択戦略では 10 タスク目，支配数期待値最大タスク選択戦略では 6 タスク目に再現率 1 となった．

5.2.2.2　負の相関を持つデータ

タスクデザイン A. タスクデザイン A にランダム戦略，面積最小タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.6 に示す．ランダム戦略では 1790 タスク目，面積最小タスク選択戦略では 1175 タスク目に再現率 1 となった．

タスクデザイン B. タスクデザイン B にランダム戦略，面積最小タスク選択戦略，支配数期待値最

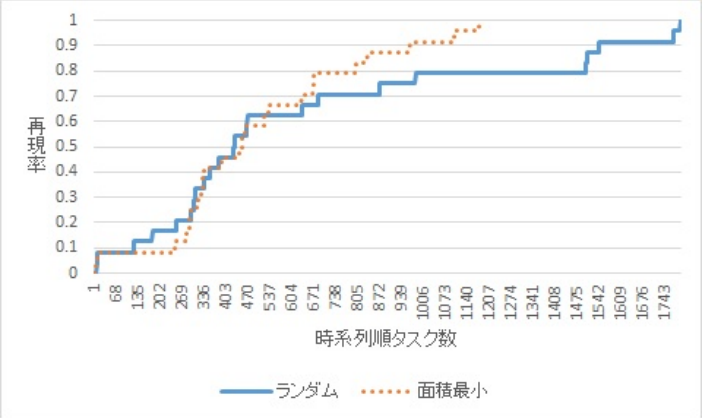

図 5.6　負の相関：タスクデザイン A

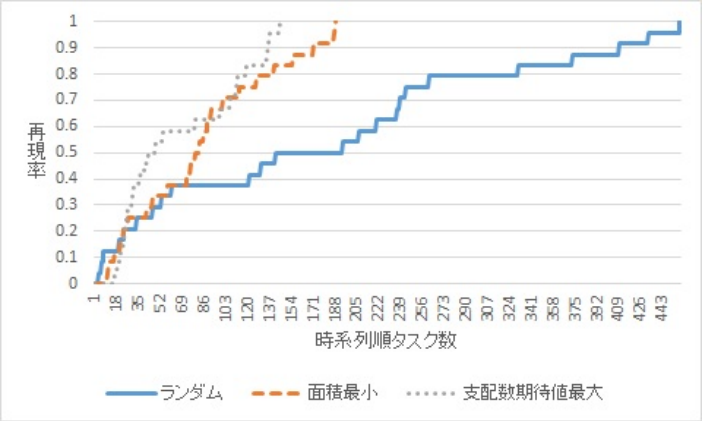

図 5.7　負の相関：タスクデザイン B

大タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.7 に示す．ランダム戦略では 457 タスク目，面積最小タスク選択戦略では 189 タスク目，支配数期待値最大タスク選択戦略では 145 タスク目に再現率 1 となった．

5.2.2.3　無相関のデータ

タスクデザイン A. タスクデザイン A にランダム戦略，面積最小タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.8 に示す．ランダム戦略では 432 タスク目，面積最小タスク選択戦略では 408 タスク目に再現率 1 となった．

タスクデザイン B. タスクデザイン B にランダム戦略，面積最小タスク選択戦略，支配数期待値最大タスク選択戦略を適用した場合のシミュレーションの結果得られた各再現率を図 5.9 に示す．ランダム戦略では 66 タスク目，面積最小タスク選択戦略では 42 タスク目，支配数期待値最大タスク

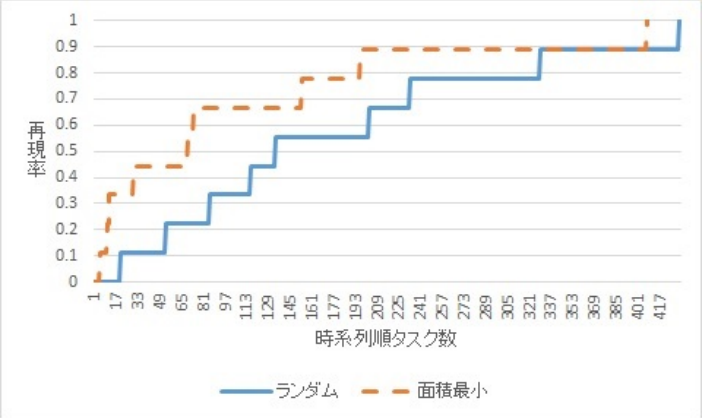

図 5.8　無相関：タスクデザイン A

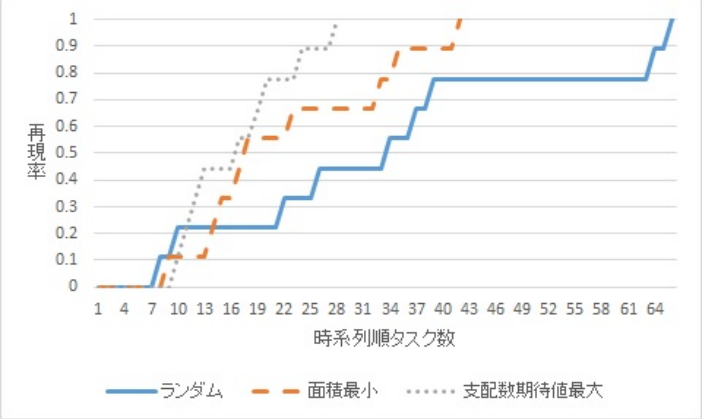

図 5.9　無相関：タスクデザイン B

選択戦略では 28 タスク目に再現率 1 となった．

5.2.3　考察

5.2.2 節で示した結果より，考察を行う．どのデータにおいても，タスクデザイン B の支配数期待値最大タスク選択戦略が最も少ないタスク数で再現率 1 の結果を得た．また，タスクデザイン A よりタスクデザイン B の方がより少ないタスク数で再現率 1 となった．

全てのデータポイントを収集し Skyline 問合せを行う単純手法でこの問題を扱う場合，データポイントの数だけデータポイントを収集するタスクが必要となるため，単純手法では 1000 タスク必要である．実験結果より，負の相関を持つデータに関するタスクデザイン A のランダム戦略，面積最小タスク選択戦略のタスク数は，単純手法よりも多いことが分かる．原因として，負の相関を持つデータの中で基準ポイントを 1 つ設定すると，他のデータポイントに支配される可能性のあるデータポ

イントが多数含まれためと考えられる. . 実際，タスクデザイン A で入力されたデータポイントは，各タスク終了時点で入力されているデータポイントに支配されるデータポイントの入力が多かった．このことから，負の相関を持つデータに対しては，タスクデザイン A は有用ではないことが分かる．

タスクデザイン B は，どのデータにおいても単純手法より少ないタスク数で Skyline ポイントの収集を行えている．また，支配数期待値最大タスク選択戦略が最も少ないタスク数で再現率を得たことから，タスクデザイン B の支配数期待値最大タスク選択戦略が有用であることが言える．

第 6 章

おわりに

本論文では，既知のデータポイントが存在しない状態から，マイクロタスク型クラウドソーシングを用いて Skyline ポイントの収集を行う手法を提案した．また，Skyline ポイントを収集するために 2 つのタスクデザイン (タスクデザイン A, タスクデザイン B) を設計し，その効率化について議論した．本論文では，手法の説明に加え，本手法の有効性を確認するために次の 2 つの評価結果を示した．(1) 実データを用いたシミュレーション実験．提案手法を用いて NBA 選手の中で出場試合数が少なく得点数が多い選手を探す問題を解くシミュレーションを行い，手法を比較した．その結果，タスクデザイン B に効率化手法の 1 つである支配数期待値最大タスク選択戦略を適用した場合，最も少ないタスク数 227 で再現率 1 となった．(2) 人口データを用いたシミュレーション実験．異なる性質のデータを用いてシミュレーションを行い，手法を比較した．その結果，どのデータにおいても，タスクデザイン B に効率化手法の 1 つである支配数期待値最大タスク選択戦略を適用した場合が，最も少ないタスク数で再現率 1 となった．

今後の課題は次の通りである．(1) 暗黙の属性値を用いる実験やシミュレーションによる評価．実験では，暗黙の属性値 (例えば「メンタルの強さ」など) を扱っていないため，暗黙の属性値を用いた実験を今後行う必要がある．(2) ワーカのドメイン知識を考慮した適切なタスク割り当て．実験では，データを用いたシミュレーションを行ったため，ワーカはつくば市のレストランを知っていることが前提としている．しかし，実用上ではつくば市のレストランのことを文京区に住むワーカに尋ねても回答されない可能性が高い．そのため，実用上ではワーカのドメイン知識を考慮して適切なタスクを割り当てる必要がある．実際にはワーカに自分にできるタスクを選択してもらう方法が考えられる．(3) データポイントに関する属性の部分的な入力への対応．実験では，データポイントの各属性値をワーカは全て入力できることを前提としている．しかし，実用上ではワーカが各属性値を全て入力できるということは保証できない．そのため，ワーカが分かる属性値のみ部分的な入力をしてもらい，入力されなかった属性値は別のワーカに新たなタスクとして割り当てるといった仕組みを考える必要がある．

参考文献

[1] Stephan Borzsony, Donald Kossmann, Konrad Stocker: The Skyline Operator. ICDE 2001, pp. 421-430 (2001)

[2] Yahoo!ANSWERS. http://answers.yahoo.com/ , (参照 2016-01-13)

[3] OK Wave. http://okwave.jp/ , (参照 2016-01-13)

[4] 人力検索はてな. http://q.hatena.ne.jp/ , (参照 (2016-01-13)

[5] Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Human-powered Sorts and Joins. PVLDB 5(1), 13-24 (2011).

[6] Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Crowdsourced Databases: Query Processing with People. CIDR 2011, 211-214 (2011).

[7] Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin: CrowdDB: answering queries with crowdsourcing. SIGMOD 2011, 61-72 (2011).

[8] Aditya G. Parameswaran, Neoklis Polyzotis: Answering Queries using Humans, Algorithms and Databases. CIDR 2011, 160-166 (2011).

[9] Christoph Lofi, Kinda El Marry, Wolf-Tilo Balke: Skyline Queries in Crowd-Enabled Databases. EDBT 2013, 465-476 (2013).

本研究に関連する発表論文

国内研究会論文

- 平木理恵，森嶋厚行 “クラウドソーシングを用いた Skyline ポイントの収集” DEIM2015, 8pages, 2015-3.

その他発表論文

- 米良俊輝，平木理恵，若宮翔子，森嶋厚行，荒牧英治．“クラウドソーシングを用いた仮説入手・検証の自動化” DEIM2016，5pages，2016-3.(投稿中)

謝辞

本研究を進めるにあたり，指導教員である森嶋厚行教授に深く感謝致します．大学学群から 3 年間，熱心に御指導して下さいました．ここまで研究を進めることができたのは森嶋先生のおかげです．ありがとうございました．

また，森嶋研究室に所属する方々にも大変お世話になりました．既に卒業されている方も含みますが，権守さん，丹治さんの先輩方には研究に困った時やゼミなどで日ごろからアドバイスをいただきました．先輩方から頂く適切なアドバイスは，私の研究生活においてなくてはならないものでした．さらに，リサーチアシスタントの池田さん，同期である櫻井さん，後輩である太田さん，根本君，林君，熊井君，佐々木さん，鈴木君，中村君，米良君にも大変お世話になりました．論文の添削から研究に関する鋭い指摘，日々の雑談まで，充実した研究生活が送れたのは皆さんのおかげです．学会参加時の書類作成時等にお世話になりました秘書の篠崎さんにも感謝致します．

また，副研究指導員を引き受けて下さいました上保准教授に感謝致します．

最後になりましたが，合同ゼミなどでご指導いただきました杉本重雄教授，阪口哲夫准教授，永森光晴講師に，深く感謝いたします．