

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-107385
(P2017-107385A)

(43) 公開日 平成29年6月15日(2017.6.15)

(51) Int.Cl.		F I		テーマコード (参考)
G06Q 10/04 (2012.01)		G06Q	10/04	5 L049
G06Q 10/06 (2012.01)		G06Q	10/06	1 30
G06F 9/50 (2006.01)		G06F	9/46	4 65 C

審査請求 未請求 請求項の数 7 O L (全 18 頁)

(21) 出願番号	特願2015-240391 (P2015-240391)	(71) 出願人	000004226 日本電信電話株式会社 東京都千代田区大手町一丁目5番1号
(22) 出願日	平成27年12月9日 (2015.12.9)	(74) 代理人	110001519 特許業務法人太陽国際特許事務所
		(72) 発明者	羽田野 真由美 東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内
		(72) 発明者	西田 京介 東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内
		(72) 発明者	戸田 浩之 東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内

最終頁に続く

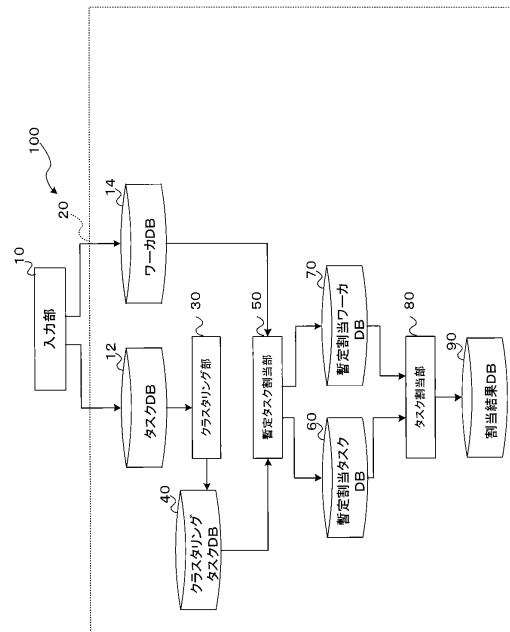
(54) 【発明の名称】 タスク割当装置、方法、及びプログラム

(57) 【要約】

【課題】タスクの密度に応じて、ワーカに最適なタスクを割り当てることができる。

【解決手段】クラスタリング部30が、複数のタスクの各々の位置情報に基づいて、複数のタスクを複数のクラスタに分類する。暫定タスク割当部50が、複数のクラスタの各々について、クラスタに属する全てのタスクを割り当て可能なワーカのうち、必要時間が最小となるワーカに、クラスタに属する全てのタスクを割り当てる。タスク割当部80が、ワーカに割り当てられていないタスクの各々について、タスクを割り当て可能なワーカのうち、必要時間が最小となるワーカに、タスクを割り当てる。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

複数のタスクを、移動する複数のワーカに割り当てるタスク割当装置であって、
前記複数のタスクの各々の位置情報に基づいて、前記複数のタスクを複数のクラスタに分類するクラスタリング部と、

前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間に基づいて、前記複数のクラスタの各々について、前記クラスタに属する全てのタスクを割り当て可能なワーカのうち、前記クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、前記クラスタに属する全てのタスクを割り当てる暫定タスク割当部と、

10

前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間と、前記クラスタに属する全てのタスクが割り当てられたワーカの前記必要時間とに基づいて、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるタスク割当部と、
を含むタスク割当装置。

【請求項 2】

前記タスク割当部は、特定の人から多く訪問されているエリアである度合いを表すロケーションエントロピーが低いエリアに含まれる位置情報を有するタスクを優先して、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てる請求項 1 記載のタスク割当装置。

20

【請求項 3】

前記クラスタに属する全てのタスクを処理するのに必要な必要時間は、前記クラスタに属する全てのタスクの各々についての、前記ワーカの移動系列に含まれる位置のうち、前記タスクの位置情報に最も近い位置から計算される、前記タスクの位置との距離が最も短い位置、及び前記タスクの位置の間の移動時間と、前記クラスタに属する全てのタスクを処理するための移動時間との和である請求項 1 又は 2 記載のタスク割当装置。

【請求項 4】

複数のタスクを、移動する複数のワーカに割り当てるタスク割当装置におけるタスク割当方法であって、

30

クラスタリング部が、前記複数のタスクの各々の位置情報に基づいて、前記複数のタスクを複数のクラスタに分類するステップと、

暫定タスク割当部が、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間に基づいて、前記複数のクラスタの各々について、前記クラスタに属する全てのタスクを割り当て可能なワーカのうち、前記クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、前記クラスタに属する全てのタスクを割り当てるステップと、

タスク割当部が、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間と、前記クラスタに属する全てのタスクが割り当てられたワーカの前記必要時間とに基づいて、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるステップと、

40

を含むタスク割当方法。

【請求項 5】

前記タスク割当部が割り当てるステップは、特定の人から多く訪問されているエリアである度合いを表すロケーションエントロピーが低いエリアに含まれる位置情報を有するタスクを優先して、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てる請求項 4 記載のタスク割当方法。

50

【請求項 6】

前記クラスタに属する全てのタスクを処理するのに必要な必要時間は、前記クラスタに属する全てのタスクの各々についての、前記ワーカの移動系列に含まれる位置のうち、前記タスクの位置情報に最も近い位置から計算される、前記タスクの位置との距離が最も短い位置、及び前記タスクの位置の間の移動時間と、前記クラスタに属する全てのタスクを処理するための移動時間との和である請求項 4 又は 5 記載のタスク割当方法。

【請求項 7】

コンピュータを、請求項 1～請求項 3 のいずれか 1 項に記載のタスク割当装置の各部として機能させるためのプログラム。

【発明の詳細な説明】

10

【技術分野】

【0001】

本発明は、タスク割当装置、方法、及びプログラムに係り、特に、ワーカに最適なタスクを割り当てるためのタスク割当装置、方法、及びプログラムに関する。

【背景技術】

【0002】

複数の仕事（以下、タスクという）を複数の人（以下、ワーカという）に割り振る際、1 人あたりのタスク数の上下限や、1 タスクあたりのワーカ数の上下限などの制約条件を満たしながらワーカ対タスクの割り当て数を最大にする最適化計算を行うことで、効率的な割り当てを実現する取組みが行われている。さらに、あるワーカ i があるタスク j を受け持つときにコスト c_{ij} がかかるとき、総コストが最小となる最適化計算を行うことで効率的な割り当てを実現する取組みが行われている。これらのワーカ対タスクの最適な割り当て方法を見つける問題を「割り当て問題」という。

20

【0003】

本技術では、クラウドソーシングサービスにおけるワーカ対タスクの割り当て問題を解くことを目的とする。クラウドソーシングとは、不特定多数のワーカにタスクを委託する枠組みのことであり、すでに商用サービス（例えば、ランサーズ（R）、URL: <http://www.lancers.jp>）として成立しているものもある。その中でも特に、以下の 2 つの条件を満たす状況を考える。1 つめはタスクに時空間条件が付与されており、ワーカが指定された場所に物理的に移動しなければタスクを完了することができない場合である。2 つめはワーカの将来行動の移動系列（時刻と緯度経度の組）が既知である場合である。ワーカの移動系列は GPS センサなどの測位技術を用いることで記録でき、またそのログを利用して日常生活の移動系列を予測することは可能である（非特許文献 1 参照）。

30

【0004】

以上のような場合、従来技術ではワーカの日常行動のための移動にタスクのための移動を追加することによって、タスク割り当てを行っていた（非特許文献 2、及び非特許文献 3 参照）。非特許文献 3 の手法を具体的に説明する。この手法はタスクノード集合 N_t 、ワーカ集合 M 、全ノード間の移動時間 c_{ij} 、ワーカの移動系列ノード集合 R^m 、ワーカの最大寄り道時間集合 T^m を入力として、ワーカの通常移動時間から最大寄り道時間の分だけ余分に移動ができるという制約条件を設け、その中で割り当てるタスクの数を最大化する最適化計算を行っている。この研究のタスク割り当て例を図 8 に示す。ワーカの移動系列（白丸と実線矢印）の途中にタスク（黒四角）への移動を挿入することによってタスクを割り当てた結果の移動系列（点線矢印）を出力する。具体的なタスク割り当て手法としては、整数計画問題に定式化して厳密解を求める手法があるものの、この手法は計算コストが高いため、本発明では近似アルゴリズムによって近似解を求める手法に焦点をしばる。既存の近似アルゴリズムには、貪欲法によって逐次的に最小コストのワーカにタスクを割り当てるといったものが存在する（非特許文献 3 参照）。

40

【先行技術文献】

【非特許文献】

【0005】

50

【非特許文献 1】Monreale, A., Pinelli, F., Trasarti, R., & Giannotti, F. (2009, June). Wherenext: a location predictor on trajectory pattern mining. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 637-646). ACM.

【非特許文献 2】羽田野真由美, 中辻真, 戸田浩之, 小池義昌. (2015). クラウドソーシングの時空間拡張におけるワーカの時間制約を考慮したタスク割当て手法. 第7回データ工学と情報マネジメントに関するフォーラム.

【非特許文献 3】Chen, C., Cheng, S. F., Gunawan, A., Misra, A., Dasgupta, K., & Chander, D. (2014). TRACCS: Trajectory-Aware Coordinated Urban Crowd-Sourcing. Second AAAI Conference on Human Computation and Crowdsourcing (HCOMP-14).

10

【非特許文献 4】Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Second International conference on Knowledge Discovery and Data Mining (Vol. 96, No. 34, pp. 226-231).

【非特許文献 5】Cranshaw, J., Toch, E., Hong, J., Kittur, A., & Sadeh, N. (2010). Bridging the gap between physical location and online social networks. In Proceedings of the 12th ACM international conference on Ubiquitous computing (pp. 119-128).

【発明の概要】

【発明が解決しようとする課題】

20

【0006】

従来技術を用い、ワーカの移動系列中にタスク割り当てを行う際、割り当てるタスクの順序によって結果が大きく異なるという問題がある。その結果、割り当てるタスクの順序によっては、あるワーカでなければできないような低密度タスクを割り当て損ねたり、低コストでたくさんのタスクを一度にこなせるような高密度タスクの一部を取りこぼしたりするおそれがある。なお、非特許文献 3 では、このような貪欲法の改善として、一度貪欲法で解を求めた後に局所探索法によって得られた解を改善していく手法が提案されているが、この手法はランダムに改善候補解を選択するため、対象エリアが広い場合にほとんど効果が得られない。

【0007】

30

その結果、割り当てられなかったタスクを次の割り当てのタイミングに繰り越す必要があるため、時間的、又は金銭的なコストが余計にかかるという課題があった。

【0008】

本発明は、上記問題点を解決するために成されたものであり、タスクの密度に応じて、ワーカに最適なタスクを割り当てることができるタスク割当装置、方法、及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【0009】

上記目的を達成するために、第 1 の発明に係るタスク割当装置は、複数のタスクを、移動する複数のワーカに割り当てるタスク割当装置であって、前記複数のタスクの各々の位置情報に基づいて、前記複数のタスクを複数のクラスタに分類するクラスタリング部と、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間に基づいて、前記複数のクラスタの各々について、前記クラスタに属する全てのタスクを割り当て可能なワーカのうち、前記クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、前記クラスタに属する全てのタスクを割り当てる暫定タスク割当部と、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間と、前記クラスタに属する全てのタスクが割り当てられたワーカの前記必要時間とに基づいて、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるタスク

40

50

割当部と、を含んで構成されている。

【0010】

また、第1の発明に係るタスク割当装置において、前記タスク割当部は、特定の人から多く訪問されているエリアである度合いを表すロケーションエントローピーが低いエリアに含まれる位置情報を有するタスクを優先して、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるようにしてもよい。

【0011】

また、第1の発明に係るタスク割当装置において、前記クラスタに属する全てのタスクを処理するのに必要な必要時間は、前記クラスタに属する全てのタスクの各々についての、前記ワーカの移動系列に含まれる位置のうち、前記タスクの位置情報に最も近い位置から計算される、前記タスクの位置との距離が最も短い位置、及び前記タスクの位置の間の移動時間と、前記クラスタに属する全てのタスクを処理するための移動時間との和であるようにしてもよい。

10

【0012】

第2の発明に係るタスク割当方法は、複数のタスクを、移動する複数のワーカに割り当てるタスク割当装置におけるタスク割当方法であって、クラスタリング部が、前記複数のタスクの各々の位置情報に基づいて、前記複数のタスクを複数のクラスタに分類するステップと、暫定タスク割当部が、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間に基づいて、前記複数のクラスタの各々について、前記クラスタに属する全てのタスクを割り当て可能なワーカのうち、前記クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、前記クラスタに属する全てのタスクを割り当てるステップと、タスク割当部が、前記複数のワーカの各々について予め求められた前記ワーカが移動する位置の移動系列、及び最大寄り道時間と、前記クラスタに属する全てのタスクが割り当てられたワーカの前記必要時間とに基づいて、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるステップと、を含んで実行することを特徴とする。

20

【0013】

また、第2の発明に係るタスク割当方法において、前記タスク割当部が割り当てるステップは、特定の人から多く訪問されているエリアである度合いを表すロケーションエントローピーが低いエリアに含まれる位置情報を有するタスクを優先して、前記ワーカに割り当てられていないタスクの各々について、前記タスクを割り当て可能なワーカのうち、前記タスクを処理するのに必要な必要時間が最小となるワーカに、前記タスクを割り当てるようにしてもよい。

30

【0014】

また、第2の発明に係るタスク割当方法において、前記クラスタに属する全てのタスクを処理するのに必要な必要時間は、前記クラスタに属する全てのタスクの各々についての、前記ワーカの移動系列に含まれる位置のうち、前記タスクの位置情報に最も近い位置から計算される、前記タスクの位置との距離が最も短い位置、及び前記タスクの位置の間の移動時間と、前記クラスタに属する全てのタスクを処理するための移動時間との和であるようにしてもよい。

40

【0015】

第3の発明に係るプログラムは、コンピュータを、上記第1の発明に係るタスク割当装置の各部として機能させるためのプログラムである。

【発明の効果】

【0016】

本発明のタスク割当装置、方法、及びプログラムによれば、複数のタスクの各々の位置情報に基づいて、複数のタスクを複数のクラスタに分類し、複数のクラスタの各々について、クラスタに属する全てのタスクを割り当て可能なワーカのうち、必要時間が最小とな

50

るワーカに、クラスタに属する全てのタスクを割り当て、ワーカに割り当てられていないタスクの各々について、タスクを割り当て可能なワーカのうち、必要時間が最小となるワーカに、タスクを割り当てることにより、タスクの密度に応じて、ワーカに最適なタスクを割り当てることのできる、という効果が得られる。

【図面の簡単な説明】

【0017】

【図1】本発明の実施の形態に係るタスク割当装置の構成を示すブロック図である。

【図2】本発明の実施の形態に係るタスク割当装置におけるタスク割当処理ルーチンを示すフローチャートである。

【図3】本発明の実施の形態に係るタスク割当装置におけるクラスタリング処理ルーチンを示すフローチャートである。

【図4】本発明の実施の形態に係るタスク割当装置における暫定タスク割当処理ルーチンを示すフローチャートである。

【図5】タスク集合を構成するタスクをすべて行う際の移動系列の一例を示す図である。

【図6】本発明の実施の形態に係るタスク割当装置における低密度タスク割当処理ルーチンを示すフローチャートである。

【図7】タスクを行う際の移動系列の一例を示す図である。

【図8】タスク割当の一例を示す図である。

【発明を実施するための形態】

【0018】

以下、図面を参照して本発明の実施の形態を詳細に説明する。

【0019】

<本発明の実施の形態に係るタスク割当装置の構成>

【0020】

まず、本発明の実施の形態に係るタスク割当装置の構成について説明する。図1に示すように、本発明の実施の形態に係るタスク割当装置100は、CPUと、RAMと、後述するタスク割当処理ルーチンを実行するためのプログラムや各種データを記憶したROMと、を含むコンピュータで構成することが出来る。このタスク割当装置100は、機能的には図1に示すように入力部10と、演算部20と、を備えている。

【0021】

入力部10は、緯度経度で表された位置情報を含む複数のタスクを受け付け、タスクDB12に格納する。また、入力部10は、移動する複数のワーカの日常的な移動系列情報を受け付け、ワーカDB14に格納する。

【0022】

演算部20は、タスクDB12と、ワーカDB14と、クラスタリング部30と、クラスタリングタスクDB40と、暫定タスク割当部50と、暫定割当タスクDB60と、暫定割当ワーカDB70と、タスク割当部80と、割当結果DB90とを含んで構成されている。

【0023】

タスクDB12のデータ構造の例を表1に示す。タスクDB12には、複数のタスクの緯度経度を表す位置情報が格納される。外部からレコードを指定するために、各レコードにはタスクIDが付与されている。

【0024】

10

20

30

40

【表 1】

タスク ID	タスク 緯度	タスク 経度
1	35.67661	139.76119
2	35.57641	139.76120
3	35.87661	139.76121
4	35.87661	139.76121
...

10

【0025】

クラスタリング部 30 は、ワーカ DB 14 に格納された複数のタスクの各々の位置情報に基づいて、タスク DB 12 に格納された複数のタスクを、タスク間の距離に応じて複数のクラスタに分類し、クラスタリングタスク DB 40 に格納する。

【0026】

クラスタリングタスク DB 40 には、クラスタリング部 30 によって分類された複数のクラスタが格納される。クラスタリングタスク DB 40 のデータ構造の例を表 2 に示す。クラスタリングタスク DB 40 の各レコードには、タスク ID、タスクの緯度経度情報の他、タスクをタスク間の距離に応じてクラスタリングした結果のクラス ID が格納される。

20

【0027】

【表 2】

タスク ID	タスク 緯度	タスク 経度	クラスID
1	35.67661	139.76119	0
2	35.57641	139.76120	1
3	35.87661	139.76121	2
4	35.87661	139.76121	0
...	

30

【0028】

ワーカ DB 14 のデータ構造の例を表 3 に示す。ワーカ DB 14 には、複数のワーカの各々について、一定時間分の当該ワーカの日常的な移動系列情報が格納されており、具体的には各時刻の緯度経度と、最大寄り道時間の情報が格納されている。表 3 の例ではワーカの 1 日分の情報を格納している。なお、ワーカの日常的な移動軌跡は、GPS ログなどから日常生活の移動を抽出する既存研究によって抽出することができる（非特許文献 1 参照）。また、最大寄り道時間とは、ワーカがタスクのために余分に移動してもよいと考える時間のことを示している。例えば、表 3 のワーカ ID 1 の最大寄り道時間は 60 分であり、タスク遂行のために 1 日 60 分までであれば余分な移動を許容することを示している。なお、外部からレコードを指定するために、各レコードにはワーカ ID が付与されている。

40

【0029】

【表 3】

ワーカ ID	時刻	ワーカ 緯度	ワーカ 経度	最大 寄り道 時間
1	2014-02-02 06:00	33.67661	139.76119	60
1	2014-02-02 06:30	35.67661	139.76120	60
1	2014-02-02 07:00	35.67661	139.76121	60
...
2	2014-02-02 06:00	35.67661	139.76121	30
2	2014-02-02 06:30	35.69805	139.77001	30
...

10

【0030】

暫定タスク割当部 50 は、ワーカ DB 14 に格納された複数のワーカの各々が移動する位置の移動系列、及び最大寄り道時間と、タスク DB 12 に格納された複数のタスクの位置情報とに基づいて、クラスタリング部 30 で分類された複数のクラスタの各々について、当該クラスタに属する全てのタスクを割り当て可能なワーカのうち、当該クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、当該クラスタに属する全てのタスクを割り当てる。そして、全てのクラスタについての割り当て結果を、暫定的なタスクの割り当て結果として、暫定割当タスク DB 60 に格納し、ワーカに対してタスクを割り当てた結果を暫定割当ワーカ DB 70 に格納する。

20

【0031】

ここで、クラスタに属する全てのタスクを処理するのに必要な必要時間は、クラスタに属する全てのタスクの各々についての、ワーカの移動系列に含まれる位置のうち、タスクの位置情報に最も近い位置から計算される、タスクの位置との距離が最も短い位置、及びタスクの位置の間の移動時間と、クラスタに属する全てのタスクを処理するための移動時間との和である。

30

【0032】

暫定割当タスク DB 60 のデータ構造の例を表 4 に示す。暫定割当タスク DB 60 には、タスク ID、タスクの緯度経度、タスクが属しているクラス ID の他、ワーカに割り当てられたかどうかを示す割り当てフラグの情報が格納される。割り当てフラグ情報には、割り当てられたタスクの場合は 1 が、未割り当てのタスクの場合は 0 が格納されている。

【0033】

【表 4】

タスク ID	タスク 緯度	タスク 経度	クラス	割当て フラグ
1	35.6766 1	139.7611 9	0	1
2	35.5764 1	139.7612 0	1	0
3	35.8766 1	139.7612 1	2	0
4	35.8766 1	139.7612 1	0	1
...

10

【 0 0 3 4 】

暫定割当ワーカ DB 7 0 のデータ構造の例を表 5 に示す。暫定割当ワーカ DB 7 0 には、ワーカ DB 1 4 と同じ情報に加えて、ワーカに割り当てたタスクの割り当てタスク ID を追加したレコードが格納される。

【 0 0 3 5 】

20

【表 5】

ワーカ ID	時刻	ワーカ 緯度	ワーカ 経度	最大 寄り道 時間	割当て タスクID
1	2014-02-02 06:00	33.67661	139.76119	60	1
1	2014-02-02 06:30	35.67661	139.76120	60	0
1	2014-02-02 07:00	35.67661	139.76121	60	0
...
2	2014-02-02 06:00	35.67661	139.76121	30	0
2	2014-02-02 06:30	35.87661	139.76121	30	4
...

30

【 0 0 3 6 】

タスク割当部 8 0 は、暫定割当ワーカ DB 7 0 に格納されたワーカが移動する位置の移動系列、及び最大寄り道時間と、タスク DB 1 2 に格納された複数のタスクの位置情報と、クラスに属する全てのタスクが割り当てられたワーカの必要時間とに基づいて、暫定割当タスク DB 6 0 から得られる、ワーカに割り当てられていないタスクの各々について、当該タスクを割り当て可能なワーカのうち、当該タスクを処理するのに必要な必要時間が最小となるワーカに、当該タスクを割り当てる。ここで、タスク割り当ては、特定の人から多く訪問されているエリアである度合いを表すロケーションエントロピーが低いエリアに含まれる位置情報を有するタスクを優先して、ワーカに割り当てられていないタスクの各々について、当該タスクを割り当て可能なワーカのうち、当該タスクを処理するのに必要な必要時間が最小となるワーカに、当該タスクを割り当てる。そして、すべてのタスクの割り当て結果を割当結果 DB 9 0 に格納する。

40

50

【 0 0 3 7 】

割当結果 DB 9 0 のデータ構造の例を表 6 に示す。割当結果 DB 9 0 は、暫定割当ワーカ DB 7 0 の情報に追加でタスクを割り当てた結果が格納される。そのため、データ構造も暫定割当ワーカ DB 7 0 と同様であり、ワーカ ID、時刻、ワーカ緯度経度情報、タスク ID、及び最大寄り道時間の情報が格納される。

【 0 0 3 8 】

【表 6】

ワーカ ID	時刻	ワーカ 緯度	ワーカ 経度	タスク ID	最大寄り道 時間
1	2014-02-02 06:00	33.67661	139.76119	0	30
1	2014-02-02 06:30	35.67661	139.76120	0	30
1	2014-02-02 06:45	35.67662	139.76119	1	30
1	2014-02-02 07:15	35.66231	139.87561	4	30
1	2014-02-02 07:30	35.67661	139.76121	0	30
...		
2	2014-02-02 06:00	35.67661	139.76121	0	15
2	2014-02-02 06:15	35.87661	139.76121	3	15
2	2014-02-02 06:30	35.87672	139.88129	6	15
2	2014-02-02 06:45	35.69805	139.77001	0	15
...		

10

20

【 0 0 3 9 】

< 本発明の実施の形態に係るタスク割当装置の作用 >

30

【 0 0 4 0 】

次に、本発明の実施の形態に係るタスク割当装置 1 0 0 の作用について説明する。入力部 1 0 において複数のタスク、及び複数のワーカの移動系列情報を受け付け、それぞれをタスク DB 1 2、及びワーカ DB 1 4 に格納すると、タスク割当装置 1 0 0 は、図 2 に示すタスク割当処理ルーチンを実行する。

【 0 0 4 1 】

まず、ステップ S 1 0 0 では、ワーカ DB 1 4 に格納された複数のタスクの各々の位置情報に基づいて、タスク DB 1 2 に格納された複数のタスクを複数のクラスタに分類し、クラスタリングタスク DB 4 0 に格納する。

【 0 0 4 2 】

40

次に、ステップ S 1 0 2 では、ワーカ DB 1 4 に格納された複数のワーカの各々が移動する位置の移動系列、及び最大寄り道時間と、タスク DB 1 2 に格納された複数のタスクの位置情報とに基づいて、クラスタリングタスク DB 4 0 に格納された複数のクラスタの各々について、当該クラスタに属する全てのタスクを割り当て可能なワーカのうち、当該クラスタに属する全てのタスクを処理するのに必要な必要時間が最小となるワーカに、当該クラスタに属する全てのタスクを割り当てる。

【 0 0 4 3 】

ステップ S 1 0 4 では、暫定割当ワーカ DB 7 0 に格納されたワーカが移動する位置の移動系列、及び最大寄り道時間と、タスク DB 1 2 に格納された複数のタスクの位置情報と、クラスタに属する全てのタスクが割り当てられたワーカの必要時間とに基づいて、暫

50

定割当タスク DB 60 から得られる、ワークに割り当てられていないタスクの各々について、当該タスクを割り当て可能なワークのうち、当該タスクを処理するのに必要な必要時間が最小となるワークに、当該タスクを割り当てる。

【0044】

上記ステップ S 100 は、図 3 に示すクラスタリング処理ルーチンによって実現される。

【0045】

まず、ステップ S 200 では、集合 S を用意し、空集合とする。

【0046】

ステップ S 202 では、タスク DB 12 よりタスクのレコード集合を検索し、タスクの集合 T とする。 10

【0047】

ステップ S 204 では、タスクの集合 T から緯度経度情報を利用してタスクのクラスタリングを行い、クラス ID を集合 T に追加した集合を集合 S とする。クラスタリングの際は、緯度経度情報を二次元平面として距離を計算するほか、地球の丸みを考慮した大圏距離を利用して距離を求めることも出来る。クラスタリングのアルゴリズムは、例えば DB SCAN 法（非特許文献 4 参照）などを用いることができるが、他のクラスタリングアルゴリズムを用いても構わない。

【0048】

ステップ S 206 では、ステップ S 204 で得た集合 S の情報をクラスタリングタスク DB 40 に出力し、処理を終了する。 20

【0049】

上記ステップ S 102 は、図 4 に示す暫定タスク割当処理ルーチンによって実現される。

【0050】

まず、ステップ S 300 では、集合 A を用意し、空集合とする。

【0051】

ステップ S 302 では、クラスタリングタスク DB 40、及びワーク DB 14 よりレコード集合を検索し、それぞれタスクの集合 T、ワークの集合 W とする。

【0052】

ステップ S 304 では、集合 T から未割り当てのクラスを選択し、クラス c とする。また、クラス c を構成するタスク集合を T_c とする。また、集合 R を用意し、空集合とする。 30

【0053】

ステップ S 306 では、集合 W から未選択のワークを選択し、ワーク w とする。

【0054】

ステップ S 308 では、ワーク w の最大寄り道時間に余裕がある場合ステップ S 310 に進み、ワーク w の最大寄り道時間に余裕がない場合はステップ S 306 に戻る。

【0055】

ステップ S 310 では、ワーク w がタスク集合 T_c を構成するタスクすべてを行う際の必要時間 $time(w, T_c)$ を計算し、ワーク w、必要時間 $time(w, T_c)$ 、及び移動系列の要素のインデックスの三組み $(w, time(w, T_c), k^*)$ を集合 R に追加する。ここで、移動系列の要素のインデックスの計算方法を示す。まず、タスク集合 T_c の i 番目のタスクの緯度経度を (lat_i, lon_i) 、ワーク w の移動系列の k 番目の緯度経度を (lat_k, lon_k) とする。また、ある 2 点 (lat_i, lon_i) 、 (lat_j, lon_j) 間の距離を $dist(lat_i, lon_i, lat_j, lon_j)$ とする。ワーク w の移動系列の中で i 番目のタスクに一番近い要素 k_i は以下のように計算できる。 40

【0056】

【数 1】

$$k_i = \operatorname{argmin}_k \operatorname{dist}(lat_k, lon_k, lat_i', lon_i') \dots (1)$$

【0057】

また、2点間の距離の計算はユークリッド距離や道路ネットワーク距離などが利用できる。

【0058】

上記(1)式の k_i の計算を、タスク集合 T_c の各々のタスクの各々について行う。

【0059】

次に、タスク集合 T_c の各々のタスクについて計算された k_i の中で、タスク集合 T_c のタスクとの最短距離が最も短いものを以下(2)式のように計算する。

【0060】

【数 2】

$$k^* = \operatorname{argmin}_{k_i} \operatorname{dist}(lat_{k_i}, lon_{k_i}, lat_i', lon_i') \dots (2)$$

【0061】

次に、 $\operatorname{time}(w, T_c)$ の計算方法の一例を示す。図5に示すように、ワーカ w の移動速度を v とし、ワーカ w がタスク集合 T_c を構成するすべてのタスクを行った後に元の場所に戻るような移動を考える。その際の必要時間は、ワーカ w の移動系列からタスク集合 T_c までの移動時間とタスク集合 T_c 内の移動時間の和で表すことが出来、以下(3)式のように計算できる。

【0062】

【数 3】

$$\operatorname{time}(w, T_c) = \frac{2\operatorname{dist}(lat_{k_i'}, lon_{k_i'}, lat_i', lon_i')}{v(w)} + \frac{\operatorname{dist}(T_c, i)}{v(w)} \dots (3)$$

【0063】

ここで、 $\operatorname{dist}(T_c, i)$ はタスク集合 T_c 内の i 番目のタスクを起終点にし、タスク集合 T_c 内のすべてのタスクを巡回する際の最短経路の距離を示す。なお、ワーカ w がタスク集合 T_c を構成するタスクをすべて行った後に元の位置に戻らず、次に行く予定であった移動系列に移動をする場合でもこのような計算で最小必要時間を計算することができる。

【0064】

ステップS312では、集合 W に未処理のワーカ w が存在する場合ステップS306に戻り、集合 W に未処理のワーカ w が存在しない場合ステップS314に進む。

【0065】

ステップS314では、集合 R より、必要時間が最小となる三組みのワーカ w を選択し、そのワーカ w にタスク集合 T_c を構成するタスクすべてを割り当てた結果を集合 A に追加する。ここで、集合 A は、ワーカID、時刻、ワーカ緯度、ワーカ経度、最大寄り道時間、及び割り当てタスクIDの要素を持ち、タスクのための移動をワーカDB14に追加した結果が格納される。ここで、必要時間が最小なワーカを w_m 、そのときの必要時間を time_m 、元の移動系列の最大寄り道時間を $\operatorname{time}_{max}$ とすると、ワーカ w_m をタスク t に割り当てた結果は以下のように計算できる。まず、ワーカ w_m がタスクを行う際の移動系列は、ワーカのもとの移動系列の $k+1$ 番目の要素に挿入される。また $k+1$ 以降の時刻はそれぞれ time_m だけ後ろにずれる。さらに、最大寄り道時間は $\operatorname{time}_{max} - \operatorname{time}_m$ となる。

10

20

30

40

50

【 0 0 6 6 】

ステップ S 3 1 6 では、集合 T に未処理のクラス c がある場合ステップ S 3 0 4 に戻り、集合 T に未処理のクラス c がない場合ステップ S 3 1 8 に進む。

【 0 0 6 7 】

ステップ S 3 1 8 では、ワーカ w_m にタスク集合 T_c の各々を割り当てることにより得た集合 A を利用して、集合 T に割当てフラグ、集合 W に割当てタスク ID をそれぞれ付与した結果を暫定割当タスク DB 6 0 と、暫定割当ワーカ DB 7 0 とに出力し、処理を終了する。

【 0 0 6 8 】

上記ステップ S 1 0 4 は、図 6 に示す低密度タスク割当処理ルーチンによって実現される。

10

【 0 0 6 9 】

まず、ステップ S 4 0 0 では、集合 B を用意し、空集合とする。

【 0 0 7 0 】

ステップ S 4 0 2 では、暫定割当タスク DB 6 0 と暫定割当ワーカ DB 7 0 よりレコード集合を検索し、それぞれ集合 T' 、集合 W' とする。集合 T' はワーカに割り当てられていないタスクの集合である。

【 0 0 7 1 】

ステップ S 4 0 4 では、ワーカ DB 1 4 に格納された複数のワーカの各々が移動する位置の移動系列に基づいて、各エリアのロケーションエントロピー値を計算する。ロケーションエントロピー値の計算例を示す。ロケーションエントロピー値とは、人の訪問ログ情報から計算するエリア特性指標として、非特許文献 5 で提案されたものである。一般に、多くの人々が等しく訪問するような繁華街などでは高い値を、特定の人しか訪問しないような住宅街では低い値を示すことが知られている。以下に、ロケーションエントロピー値の計算方法を示す。まず、対象地域をグリッドなどの単位に分割し、分割した各エリアをそれぞれロケーション l とする。全ワーカがロケーション l を訪問した回数を $|O_{l1}|$ 、ワーカ u がロケーション l を訪問した回数を $|O_{u,l}|$ とすると、ワーカ u がロケーション l を訪問する確率 $P_l(u)$ は以下 (4) 式のように計算できる。

20

【 0 0 7 2 】

【数 4】

30

$$P_l(u) = \frac{|O_{u,l}|}{|O_l|} \dots (4)$$

【 0 0 7 3 】

ここで、 l を訪問したワーカ集合を U_l とすると、ロケーション l のロケーションエントロピー値は、以下 (5) 式のように計算される。

【 0 0 7 4 】

【数 5】

40

$$\text{LocEnt}(l) = - \sum_{u \in U_l} P_l(u) \log P_l(u) \dots (5)$$

【 0 0 7 5 】

ステップ S 4 0 6 では、集合 T' から未割り当てタスクのうち、タスクの位置を含むエリアのロケーションエントロピー値が最小となるものを 1 つ選択し、 t' とする。また、集合 D を用意し、空集合とする。

【 0 0 7 6 】

50

ステップS 4 0 8では、集合 W' から未処理の要素を取り出し、ワーカ w' とする。

【0077】

ステップS 4 1 0では、ワーカ w' の最大寄り道時間に余裕がない場合ステップS 4 0 8に戻り、ワーカ w' の最大寄り道時間に余裕がある場合はステップS 4 1 2に進む。

【0078】

ステップS 4 1 2では、ワーカ w' がタスク t' を行う際の必要時間 $time(w', t')$ を計算し、ワーカ w' 、必要時間 $time(w', t')$ 、及び移動系列の要素のインデックスの三組み (w', t', k') を集合Dに追加する。ここで、移動系列の要素のインデックス k' の計算方法を示す。まず、タスク t' の緯度経度を (lat', lon') 、ワーカ w' の移動系列の k 番目の緯度経度を (lat_k, lon_k) とする。また、2点 (lat_i, lon_i) 、 (lat_j, lon_j) 間の距離を $dist(lat_i, lon_i, lat_j, lon_j)$ とする。ワーカ w' の移動系列の中でタスク t' に一番近い要素 k' は以下(6)式のように計算できる。

【0079】

【数6】

$$k' = \operatorname{argmin}_k dist(lat_k, lon_k, lat', lon') \dots (6)$$

【0080】

なお、2点間の距離の計算はユークリッド距離や道路ネットワーク距離などが利用できる。次に、 $time(w, t)$ の計算方法の一例を示す。図7に示すように、ワーカ w の移動速度を v とし、ワーカ w がタスク t を行った後に元の場所に戻るような移動を考える。その際の必要時間は以下(7)式のように計算できる。

【0081】

【数7】

$$time(w, t) = \frac{2dist(lat_{k'}, lon_{k'}, lat', lon')}{v(w)} \dots (7)$$

【0082】

また、ワーカ w がタスクを行った後に元の位置に戻らず、次に行く予定であった移動系列に移動をする場合でもこのような計算で必要時間を計算することができる。

【0083】

ステップS 4 1 4では、集合 W' に未処理のワーカ w' が存在する場合、ステップS 4 0 8に戻り、集合 W' に未処理のワーカ w' が存在しない場合はステップS 4 1 6に進む。

【0084】

ステップS 4 1 6では、集合Dより必要時間が最小なワーカ w' を選択し、そのワーカ w' に t' を割り当てた結果を集合Bに追加する。

【0085】

ステップS 4 1 8では、集合 T' に未処理のタスクがある場合、ステップS 4 0 6に戻り、 T' に未処理のタスクがない場合はステップS 4 2 0に進む。

【0086】

ステップS 4 2 0では、ワーカ w' に割り当てられていないタスクの各々について割り当てることにより得た集合Bを利用して、集合 W' のレコードに割り当て結果を付与した結果を割当結果DB 9 0に出力し処理を終了する。

【0087】

以上説明したように、本発明の実施の形態に係るタスク割当装置によれば、複数のタスクの各々の位置情報に基づいて、複数のタスクを複数のクラスタに分類し、複数のクラスタ

10

20

30

40

50

タの各々について、クラスタに属する全てのタスクを割り当て可能なワーカのうち、必要時間が最小となるワーカに、クラスタに属する全てのタスクを割り当て、ワーカに割り当てられていないタスクの各々について、タスクを割り当て可能なワーカのうち、必要時間が最小となるワーカに、タスクを割り当てることにより、タスクの密度に応じて、ワーカに最適なタスクを割り当てることができる。

【0088】

また、本発明の実施の形態に係る手法によって、まず高密度なタスクをクラスタリングによってまとめ、優先的にワーカに割り当てたのちに、ロケーションエントロピーが低い順にタスクを貪欲法で割り当てることで、ワーカへ割り当てることができるタスク数を出来るだけ多くすることが可能となる。その結果、従来手法では割り当てることが出来なかった追加タスクも割り当てることが可能になるため、時間的、金銭的なコストを削減することが可能になり、従来の課題が解決される。

10

【0089】

また、上記の効果により、全体で割り当てることができるタスク数を増やすことが可能になる。その結果、従来手法では割り当てることが出来なかったタスクを追加で割り当てる時間的、金銭的なコストを削減することが可能となる。また、再割り当てが必要なくなるため、タスク完了までの納期を削減することにつながり、少しでも早く仕事を終わらせたいと考えているタスク依頼者の満足度向上につながる。

【0090】

なお、本発明は、上述した実施の形態に限定されるものではなく、この発明の要旨を逸脱しない範囲内で様々な変形や応用が可能である。

20

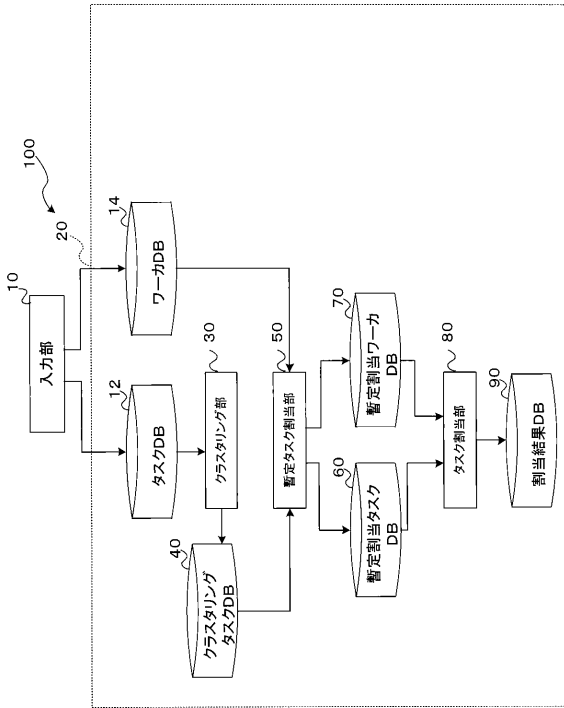
【符号の説明】

【0091】

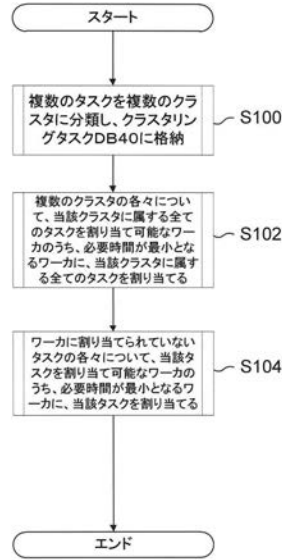
- 10 入力部
- 12 タスクDB
- 14 ワーカDB
- 20 演算部
- 30 クラスタリング部
- 40 クラスタリングタスクDB
- 50 暫定タスク割当部
- 60 暫定割当タスクDB
- 70 暫定割当ワーカDB
- 80 タスク割当部
- 90 割当結果DB
- 100 タスク割当装置

30

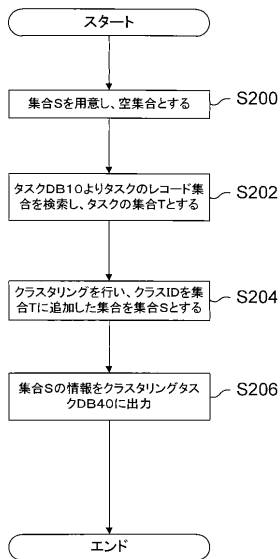
【図 1】



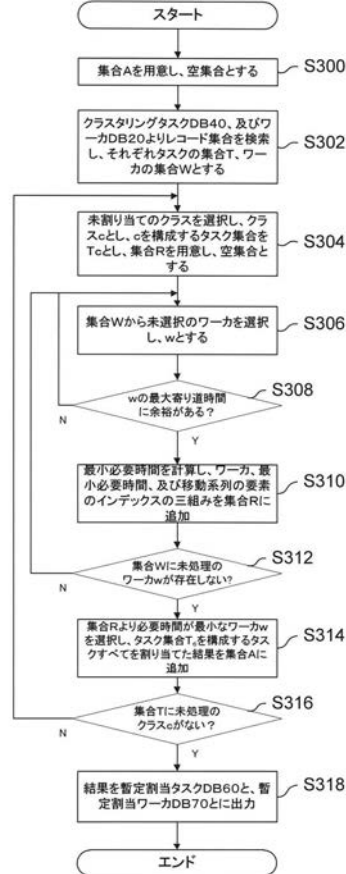
【図 2】



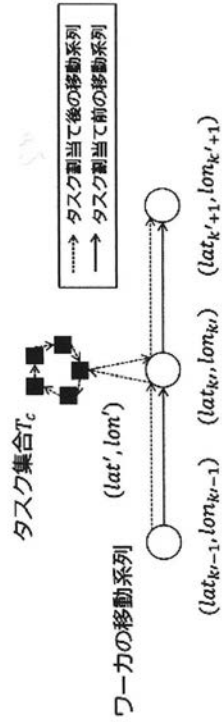
【図 3】



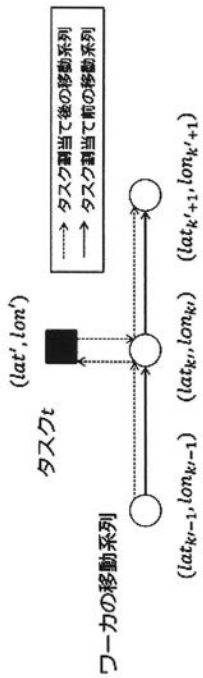
【図 4】



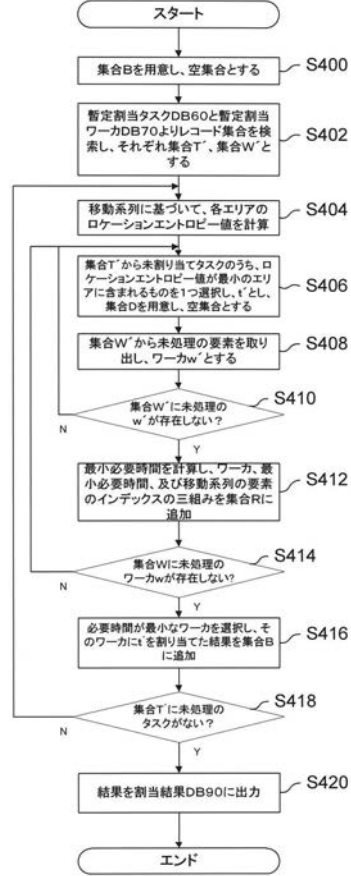
【 図 5 】



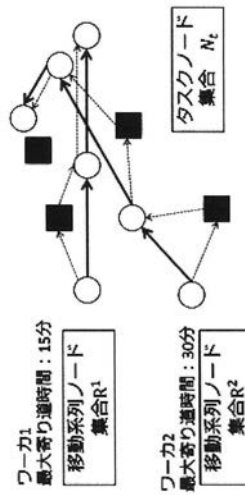
【 図 7 】



【 図 6 】



【 図 8 】



フロントページの続き

(72)発明者 川野辺 彰久

東京都千代田区大手町一丁目5番1号 日本電信電話株式会社内

Fターム(参考) 5L049 AA04 AA10