マルチモバイルエージェントによる情報探索行動における移動制御方式

堀 口 晃 佑[†1]　川 越 恭 二[†2]

複数のモバイルエージェントによる情報探索を行う場合，各々のモバイルエージェントが探索結果や作業情報を共有する事でさらにその探索効率をあげる事ができる．しかし，モバイルエージェントはネットワーク上を逐次移動しながら処理を行うために，モバイルエージェント同士が互いの位置情報を正確に把握するのは困難である．互いに正確な位置が分からなければ，情報共有が行えず情報探索効率を落としてしまう．そこで，本稿ではこれらの問題を解決するために，モバイルエージェントの移動時における動作をロック方式，スタンプ方式，切替え方式の３つの方法で制御する事によって状況に応じた適切な協調方式を提案する．この３方式のモデルにおける情報探索効率を比較するためにシミュレーションによる実験を行った．その結果，方式特性が明らかになった．さらに，各方式を用いたモバイルエージェントの移動制御方式の特性を考察し，効率的な情報探索が可能となることを明らかにした．

Control Method in Information Searching with Multi-Mobile Agent

KOSUKE HORIGUCHI[†1]　KYOJI KAWAGOE[†2]

In order to realize efficient information searching in multiple mobile agent environment it is important to coordinate each other and to share information on their searching activities among the mobile agents. However, mobile agents often move their locations independently, which cause that the mobile agents might lose others' location addresses. In such situations, it is very difficult for the mobile agents to communicate with each other. In this paper, we propose three control methods for mobile agents to coordinate each other: Stamp method, Lock method, and Combination method. With these methods, each mobile agents can freely move one to another without loss of othes' location addresses, which enables mobile agents to share searching related information. At first, we explain these three control methods, and then by some simulation experiments, we show the proposed methods to be effective.

1. はじめに

ネットワーク環境の急速な発展により人々は多くの情報を公開し取得できるようになった．ネットワークの広帯域化につれ，扱える情報はテキストデータから画像・音楽・動画データと多種多様となり，その情報量も増加している．しかし，この情報量の増加によりネットワークトラフィックの増加やネットワークの複雑化の問題が発生している．この問題を避けるために，Gnutellaなどに代表されるような非集中型の情報探索手法が注目されている．

本稿では，ネットワークノードを複数のモバイルエージェント（以降，マルチモバイルエージェントと呼ぶ）を用いて探索させる情報検索システムを対象とし，モバイルエージェント間で情報交換しながら移動を可能とする移動制御方法を提案する．

†1 立命館大学大学院 理工学研究科
Graduate School of Science and Engineering, Ritsumeikan Univ.
†2 立命館大学 理工学部
School of Science and Engineering, Ritsumeikan Univ.

2. モバイルエージェントを用いた情報検索システム

モバイルエージェントを用いた情報検索システムの仕組みについて以下に述べる．

2．1ネットワーク環境

従来の整備された分散データベースに比べ，Gnutella のような情報検索の全てのフェーズをPeer-to-Peerで行うタイプはＰｕｒｅPeer-to-Peer型情報検索システムと呼ばれている[1]．この検索システムでは，中央サーバのように情報を一元管理できる仕組みを持たない．ネットワーク内における全てのデータ構造・ネットワーク環境を把握して検索処理を行うのは不可能である．

2．2 マルチモバイルエージェントシステム

2. 1 で示したネットワーク環境に対し，モバイルエージェントを利用した検索システムについて説明する．Gnutellaは中央サーバによる管理なしに情報検索が行えるシステムである．繋がれているホストに対し，利用者が波状バケツリレー式に検索要求を送信する事によりネットワークの状態や検索対象の有無を調べる．モバイルエージェントを用いた情報検索システムでは，マルチキャストされる検索要求ではなく，より自律的検索処理が行えるモバイルエージェントに検索処理を代行させるシステムである[2]．マルチモバイルエージェントシステムにおけるエージェントの持つ特徴を次に示す．

（1）移動

各ネットワークノードにおいて必要なデータ処理を行い作業完了後検索結果や学習内容といったエージェント内部情報を保持したまま他のノードへと移動する．

（2）複数動作

探索処理を分担させるために複数のエージェントによって作業を行う．

（3）協調動作

複数のモバイルエージェントが各自作業を行っているネットワークノード先において，有効な情報を全てのモバイルエージェント間で共有する事により探索効率を上げる．

また，一般的にモバイルエージェントシステムによって従来の Gnutella プロトコルに代表されるファイル検索システムと大きく違う点を以下に示す．

・検索手法

現在多くの検索システムは，マルチメディアデータを検索対象にしているが，検索方法としてはテキスト一致検索のみと脆弱である．モバイルエージェントはユーザ要求に合致した検索エンジンを持って移動する事により移動先でのより詳細なデータ処理が可能となる．

・検索効率

検索要求をマルチキャストするよりも各探索結果から最適なルーティングの決定や検索対象の絞り込みを行う事ができる． これによって無駄な相手先ノードへの計算処理を軽減する事ができる．

・検索にかかるネットワークコスト

Gnutellaが検索時において，マルチキャストされる検索要求に対して，モバイルエージェントはそれ自体が何らかの検索アルゴリズムを備えて，相手先ホストに移動するため，ある程度のデータサイズをもっているが，これにより高度なデータ処理が行える．

また，検索条件に一致するデータを全てユーザ側に提供するのではなく，モバイルエージェントがある程度の取捨選択作業を行い情報をフィルタリングする事で，精度が高い情報だけを検索結果として提供する事ができる．これによってネットワークトラフィックを軽減する事ができる．

3． モバイルエージェントの動作

一般に，モバイルエージェントの動作は移動処理とデータ処理の２つに大別される．まず，モバイルエージェントはネットワークノード内を移動し，次に，移動先に存在する資源に対してデータ加工や検索といった処理を行う．さらに，このエージェントは一つの場所で処理を終えると次の移動先を決定して，探索が完了するまで，この動作を繰り返す．

本稿で扱うマルチモバイルエージェントによる情報探索システムでは，WWW検索に代表されるHTMLデータからマルチメディアデータに至るまで，あらゆる資源を検索対象とし，様々な検索方法を想定したシステムである．そのため状況によって，モバイルエージェントの持つ検索エンジンやフィルタリング方式といったデータ処理アルゴリズムは状況によって異なると仮定する．

また，あらかじめモバイルエージェントが最適な移動経路を決定する方法に関する研究や，動的に経路を決定するためにネットワークコストなどの数値を基にモバイルエージェントをメタ制御するといった研究がこれまでに行われている．[3][4][5][8]

そこで本稿では，モバイルエージェントの移動処理におけるモバイルエージェント間の情報共有のための協力作業（以下，協調動作）に焦点をあてて，より効率的な探索方法を提案する．

3．1 協調動作

一般的には，複数のモバイルエージェントが動作する場合，各々のモバイルエージェントが独立して移動処理を行うよりも移動先で得た情報をやり取りした方が効率が上がる．これは，探索結果から学習した内容を全てのモバイルエージェントの動作に逐次反映させる事で，後の情報探索に改善を加える事が出来るためである[6]．

そこで，シミュレーション実験によって，モバイルエージェントが協調動作することによって得られる有用性を示すために，次の２つの事を明確に定義しなければならない．１つ目は，モバイルエージェントの協調動作のアルゴリズムであり，２つ目はモバイルエージェントが協調動作で得られるメリットである．

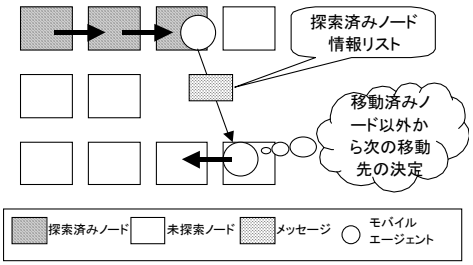

図１　協調動作による移動先ノードの決定

本稿では，図１に示すように，協調動作とは「モバイルエージェントが探索したノードの場所情報と取り扱ったデータ情報を互いに通信させて互いのモバイルエージェントが“探索済みノード情報リスト“として保持する」事であると定義する．次に，協調動作によって得る事のできるメリットは，「この“探索済みノード情報リスト“を参照する事により，異なるモバイルエージェントが重複したノードにデータ処理する事や重複した移動先へ移動する事を防ぐ」という内容に限定して，モバイルエージェントの動作を考えこととする．モバイルエージェントシステムでは移動先ノードでデータ処理が行われるため，モバイルエージェントの重複移動処理は大変効率が悪い．そこで，この問題を解決するため，メッセージをやり取りして情報を共有する必要がある．しかしモバイルエージェントのメッセージ通信において，複数のモバイルエージェントは非同期に独立して移動するという特徴をもつため，モバイルエージェントの位置情報を互いに把握していないといけない．

また，協調動作のために考えられる手法として中央管理サーバのようなものを利用して位置情報を認識する間接通信方式が考えられる[7]．しかし，この手法を用いた協調動作では，Gnutella の方法を基本とするサーバレスな Peer-to-Peer 情報検索システムとしてのコンセプトからはずれてしまう．また他に固定スペースのメッセージのやり取りによる負荷の集中といった問題やダウンする事で全体の作業が中断されるといったリスク集中の問題や全てのモバイルエージェントが情報を更新するためにわざわざ固定スペースに問合せをかけないといけないため，情報のリアルタイム性が失われるといった問題が生じる．このため本稿で提案する方法では直接通信による協調動作を考える．しかし直接通信方式では，間接通信のように，固定共有空間を介して，情報をやり取りするわけではないので，全てのモバイルエージェントが他の全てのモバイルエージェントの移動ノード先を把握する必要がある．

３．２　モバイルエージェントの動作

Ｎ 個のネットワーク内ノードにおいて，Ｍ個のモバイルエージェントが，動作する場合を考える．モバイルエージェントの持つ内部変数は以下の通りである．ここでMessageは既に探索したネットワークノード情報をやり取りするための通信用オブジェクトである．

モバイルエージェントの存在場所（以下 Address とする）を CurrentNode，相手方モバイルエージェントの AgentID x の Address を RemoteNode(x)と記述する．またモバイルエージェントの次の移動先ノードを NextNode とする．

[Agent]

<u>Agent field</u>

-AgentID:

Agent は０～Ｍ－１　のＩＤを持つとする．

-AgentNode[M]:

Ｍ個の配列．各 Agent が把握している他の Agent Address リスト {RemoteNode(0), RemoteNode(1), ・・・ RemoteNode(M-1)}

-Node[N]:

Ｎ個の配列．探索状況を保持するリスト．

既探索の場合に１を，未探索の場合は０をセットする．

<u>Agent method</u>

-Lock:

他のエージェントに対する移動を禁じる．

-Release:

Lock を開放する．

-Migrate:

Node[0～Ｎ－1]==0 である集合から，ランダムで１つ次の移動先ノードを決定し，移動する．（下記 Message 参照）

-CreateMessage (CurrentNode, RemoteNode(x), NextNode):

AgentID x 宛の Message の生成．

-CreateStamp (NextNode):

Stamp を currentNode 内に生成する．

-LocalTransaciton:

ローカルデータ処理. 本稿では研究の焦点は移動フェーズであるために，詳細な記述は省略する．

[Message]

Message field

-CurrentNode:

送信元 Agent の Address.

-RemoteNode(x):

送信先 AgentID x の Address.

-NextNode:

送信元 Agent の次の移動先 Address.

Message method

-Cast:

Message. RemoteNode(x)へ Message 送信.

-Rewrite:

AgentID x の持つ Agent の Node[NextNode]に 1 をセット.

[Stamp]

Stamp field

-CurrentNode:

スタンプ発行 Agent の CurrentNode.

-NextNode:

スタンプ発行 Agent の NextNode.

3．3 提案するモバイルエージェント探索方式

探索作業の際に協調を全く行わない独立方式と提案する 3 方式による 4 つの方法のいずれかを用いて移動処理行う．3．2 で定義したメソッドを使って，以下に各方式のアルゴリズムの説明をする．

3．3．1独立方式

協調動作を行わないで探索する方式である．このため，全体としての統制をとる制御がいらず，個々の Agent は以下に示すように独立して探索処理を行っている．

（動作）

$$While(\sum_{i=0}^{N-1} Node[i] < N){$$

:Local Transaction ----探索処理を行う
:Agent. Migrate -----移動する
}

探索終了後結果を返す.

3．3．2スタンプ方式

スタンプ方式は、スタンプを発行してから移動処理を行う移動方式である．ここで、スタンプとは，モバイルエージェントの移動先Addressを記述したもので，旧作業ノードに配置されることで，CurrentNodeはNextNodeとリンクした状態となる． メッセージCast先が相手方モバイルエージェントであればRewriteを実行し，StampであればStampのNextNodeを参照して，再びCasttされる．

（動作）

$$While(\sum_{i=0}^{N-1} Node[i] < N) \ \{$$

:Local Transaction -----探索処理を行う
:CreateStamp ---スタンプを作成
:Migrate ---移動する
:Message. Cast(RemoteAgentID)
---メッセージの送信
:While(Stamp in Node?){
:Message.Cast (RemoteAgentID)
---メッセージの送信
}
:Message. Rewrite(RemoteAgentID)
---リストの更新

}:

探索終了後結果を返す.

3．3．3ロック方式

ロック方式において，移動しようとするAgentは，他の全てのAgentに対してロックを掛けてから移動処理を行う．ロックを掛けられる事により，移動処理を禁じられる．ロックの解除により予約されたAgentから順に移動処理を実行する．

（動作）

$$While(\sum_{i=0}^{N-1} Node[i] < N){$$

:Local Transaction ----探索処理を行う
:Lock ----ロックを設定する
:Message. Cast(RemoteAgentID)
---メッセージの送信
:Message. Rewrite(RemoteAgentID)
---リストの更新
:Migrate ----移動する
:Release ----ロックを解除する
}

探索終了後結果を返す．ここで，LockおよびReleaseは各々，3．2 で記述したロックを設定する処理である．

3．3．4切替え方式

Agentの処理状況に応じて，先に述べた独立方式，スタンプ方式，ロック方式の 3 種類の方式を切替えて探索を行う方式である．Agentは探索済みであるノード数を基準として切替えるポイントを決める．独立方式からスタンプ方式への切替えポイントとスタンプ方式からロック方式への切替えポイントなる $\sum_{i=0}^{N-1} Node[i]$ の値をそれぞれn1，n2 とする．この(n1，n2)の値の決定に関する詳細は４．３節で説明する．

4 シミュレーション

ローカル内に仮想的に用意されたネットワークノードを複数のモバイルエージェントに探索させる．モバイルエージェントの協調動作の有用性を検証するため協調動作を行わない独立方式と提案方式の 3 つの方式の比較実験を行った．またシミュレーション環境は以下のような条件で行った．

4．1シミュレーション環境

シミュレーション環境を以下に示す．

- 本実験では，1000 個の仮想ネットワークノードを用意し，5 個のモバイルエージェントが独立方式，スタンプ方式，ロック方式の各 3 タイプの処理アルゴリズムで探索する．その結果を元に切替え方式における探索コストを求める．
- シミュレーションの終了条件は 1000 個のノードを全て探索したときとする．
- 移動するエージェントは移動先ノード上でのデータ処理を行い，処理が終了すれば次のノードへの移動を繰り返す．
- 移動コストやデータ処理コストにはあらかじめかかる時間として決められた以下に示すコストをセットする．
- 探索終了後にその探索にかかった時間と移動ノード数を算出し，これを評価項目とする．また，ロック方式以外では重複移動する場合もあるため重複移動ベース数などを動作の正確性として評価基準に加える

4．2コスト

モバイルエージェントの処理における最小時間単位を t=1 とおき，この単位を基準に評価する．各処理フェーズに以下に示す移動コストとデータ処理コストを設定し，その処理に掛かる時間とする．

・移動コスト

実際には個々のノード間の距離は様々であるが，シミュレーションでは全てのモバイルエージェントが次のノードに移動するコストは一定とする．本実験において t=10 と設定する．

・データ処理コスト

移動先において処理対象となる資源の容量は移動するまでは，ある程度の予測しかできない．例えば検索対象がテキストであるのか，音楽ファイルなのであるかという検索対象が指定されれば容量のラフな予測は可能であるが，移動先にどの程度のデータが存在するのか，計算機処理の性能はどの程度か等の推定は移動するまで不明である．このため，今回の実験では，平均レート t=10 のランダム値を使用した．

4．3 実験結果

探索結果を以下に示す．

- 独立方式：t=10000 の時，node=985
- スタンプ方式：探索終了時，t＝3632
- ロック方式：探索終了時，　t＝3956

4．3．1探索ノード数

独立方式とスタンプ方式とロック方式において，時間 t を横軸，探索ノード数を縦軸として図 2 に結果を示す．

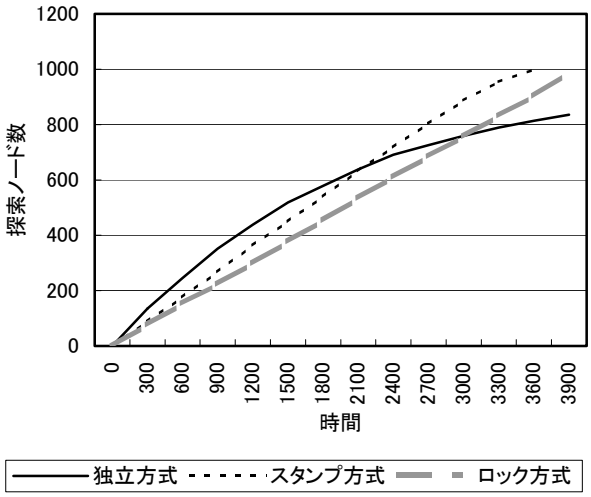

図 2　探索済みノード数の比較

4．3．2探索ノード数

時間 t を横軸，異なるモバイルエージェントが同じノードを重複して探索した数を縦軸としてとして，図 3 に示す．

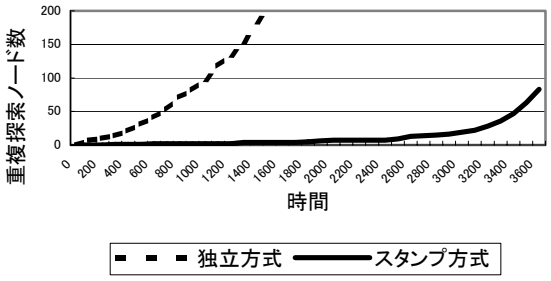

図 3　重複探索ノード数の比較

4．4切替えポイント

切替え方式では，上記に示した方式毎の結果を状況に応じて使い分ける事によって，最適な探索を行う．以下の２つの基点によって，切替えることを考える．

（1）　探索ペース優先の切替え

探索ペースが高いものを選択し，探索する．

（2）重複探索ノード数を基点とする切替え

超えてはいけない重複ノード数　N_{MAX}　を設定し，N_{MAX}　を超えないように探索する．N_{MAX}=0 の場合はロック方式のみの探索する．

ここで実験結果各方式における探索ノード数を（1）の切替えポイントで行った場合にどの程度改善されるのかを以下に示す．切替えコストを０と仮定した場合の結果を図 4 に示す．図 4 では n1=380，n=2940 の場合の探索ノードと時間の遷移を示している．

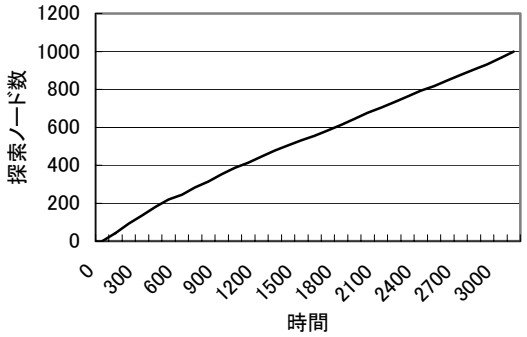

図 4 探索ペース数による切替え探索

4．5 考察

本実験において，各方式には以下のような特性があることが分かった．

-独立方式

探索初期に探索ペースが高いが，探索ノード数が増えるにつれ極端に探索効率が下がる．

-スタンプ方式

スタンプ方式では,移動のために，Agent 同士の同期を必要としない．また複数のモバイルエージェントが同時に移動できるという利点がある．これらの理由により，ロック方式よりも探索時間は早い．しかし t=3300 あたりから徐々に，探索ペースが下がってくる．これは Node リスト更新に多少のタイムラグがあるためであると考えられる．

-ロック方式

ロック方式では,リスト情報の更新が同時に行われるために,探索ペースは安定している．また一度に一つの Agent しか移動できないために探索ペースは遅いが，未探索ノードが残り70件(t=3200)あたりでは，一番探索ペースが高い．

-切替え方式

切替え方式では,探索時間コストが最短（t=3120）である．しかし切替えのための通信オーバヘッドが発生するため，現実にはさらに時間コストが増加するものと予想される．

最初に実験を行った３方式において探索効率は残り探索ノード数に依存する事がわかった．したがって，あらかじめシミュレーションを通して得られた切替えポイント（n1，n2）を使用する事で，実際の情報探索においても最適な探索を行う事ができると考える．

5.　おわりに

本稿では，マルチモバイルエージェントによる情報検索システムにおいて，モバイルエージェント同士が互いの位置情報を正確に把握できないことから情報共有ができず情報探索ペースが小さくなるという問題を解決するために，モバイルエージェントの移動時における動作をロック方式，スタンプ方式，切替え方式という3つの協調方式を提案した．

本方式の有効性を示すためにシミュレーション実験を行った結果，モバイルエージェントにより最適な探索方法を選択させて情報探索を行わす事によって検索効率が上がる事を示した．

今後，切替え方式についてはなお課題が残っている．今回シミュレーション結果から出た値を参考に切替えポイントを決定したが，現実に各々のモバイルエージェントが自律的な判断を行うには，全体の見えない局所的な情報において決定行動が行われる仕組みを検討しなければならないと考える．

参考文献

[1]http://wiki. cs. uiuc. edu/cs427/Gnutella，P2P Architecture Pure P2P -Gnutella

[2] 本位田真一，飯島正，大須賀昭彦　著:「オブジェクト指向トラック　エージェント技術」

[3]何斌達，データマイニング手法を用いたモバイルエージェント分散データ検索システム，学会誌「マルチメディア通信と分散処理」pp. 49-54，(2001)

[4]菅原真司，複数の動的エージェントによるネットワーク内情報探索に関する検討，信学技報，SSE2000-238，pp. 17－24 (2000)

[5]国分俊介：時間制約下での WWW 検索のためのモバイルエージェントメタ制御操作方法の提案，情報処理学会マルチメディア，分散，協調とモバイル(DICOMO)ワークショップ論文集，Vol. 2001，No. 7，pp. 271-276，June，(2001)

[6]佐藤尚，マルチエージェント・システムによるリーダーおよび群れの創発，学会誌「知能と複雑系」pp. 25-30(2001)

[7] Giacomo Cabri，Letizia Leonard，Franco Zambonelli: Mobile-Agent Coordination Models for Internet Applications. IEEE Computer，Volume 33 (2): 82-89 (2000)

[8]國分俊介，灰原清太郎，川越恭二：時間制約下での WWW 検索のためのモバイルエージェント制御方法，情報処理学会論文誌，pp. 2945-2948，Vol. 41，No. 19，2000