2. OWL の機能を利用した推論

只見町インターネット・エコミュージアムの「キーワード」検索を改善するにあたって、まず、民具を Ontology 化するプログラムを作成する。これにより民具に新たな関係性とキーワードの増加を試みる。次に、キーワードにシソーラスを用いるプログラムの作成をする。これによりキーワードの柔軟性を試みる。そして、この二つのプログラムを用いた「キーワード」検索のプログラムを作ることを提案する。これにより必要な情報が検索可能になると考えられる。また、民具の Ontology に Wikipedia Ontology を用いることによって民具に不十分な情報を補うことができると考えられる。[12]

本研究では、法造を用いて民具カードに記載されているデータを Ontolgy で表す。それを用いて推論に使用する OWL を作成する。これにより民具に新たな関係性とキーワードの増加を試みる。

図 21 は「SHIGOTOGI SHIGOTOSHI」および「WORK CLOTHES」が is-a 関係であり、「HADAKKO」および「WORK CLOTHES_2」も is-a 関係であることを示している。

図 22 は「WORK CLOTHES」および「CLOTHES」が is-a 関係で結ばれていることを示す。

図 23 は、推論に使用する OWL データの一つである。図 23 において、2 行目の rdf: RDF 要素で名前空間の定義を行う。また、10 行目の owl: Ontology 要素で、Ontology 文章内で情報を管理する「ヘッダ」としての役割を果たす。25 行目の owl: Class はある概念を抽象化して表現し、33 行目の rdfs: subClassOf はサブクラスを定義している。このことより、25 行目から 27 行目はトップのクラス「SHIGOTOGI_SHIGOTOSHI（シゴトギ、シゴトシ）」を定義している。28 行目から 30 行目はトップのクラス「HADAKKO（ハダッコ）」を定義してい

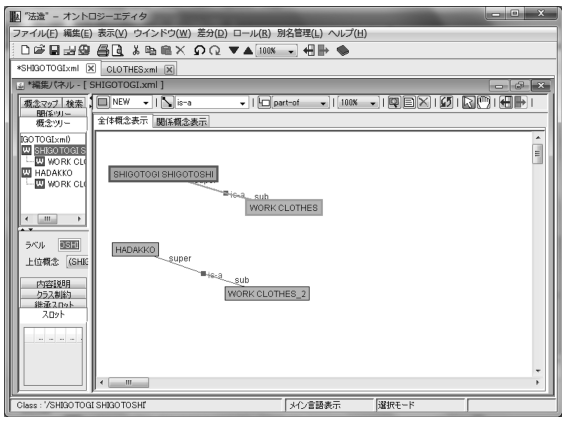


図 21　推論に使用する法造 1

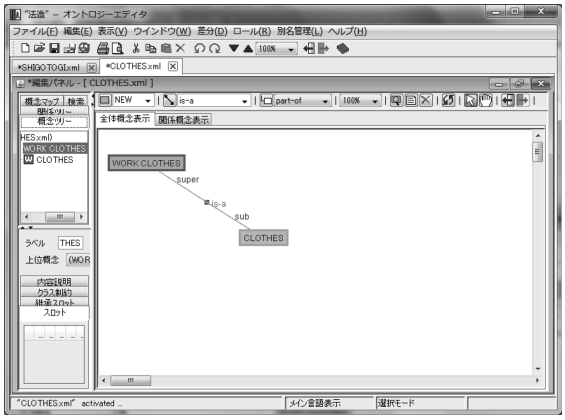


図 22　推論に使用する法造 2

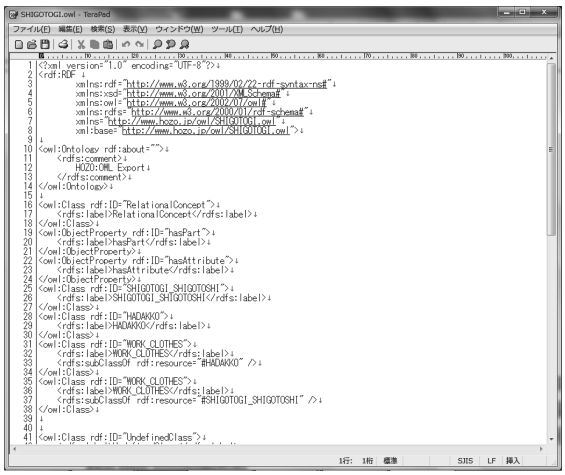


図 23　推論に使用する OWL1

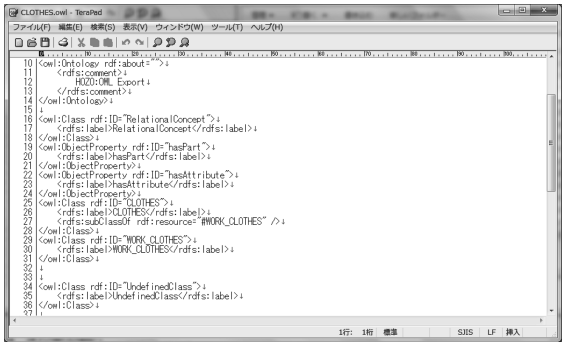


図 24　推論に使用する OWL2

る。31 行目から 34 行目は「WORK_CLOTHES（仕事着）」が「HADAKKO（ハダッコ）」のサブクラスであることを定義している。35 行目から 38 行目は「WORK_CLOTHES（仕事着）」が「SHIGOTOGI_SHIGOTOSHI（シゴトギ、シゴトシ）」のサブクラスであることを定義している。

図 24 は推論に使用する OWL データの 1 つである。29 行目から 31 行目は、トップのクラス「WORK_CLOTHES（仕事着）」を定義している。また、25 行目から 28 行目は、「CLOTHES（服）」が「WORK_CLOTHES（仕事着）」のサブクラスであることを定義している。

図 25 は、推論を行うためのプログラムである。13、14 行目の ReasonerRegistry. getRDFSSimpleReasoner（　）メソッドにより必要な Reasoner を取得する。15、16 行目で Resoner にスキーマのデータをバインドする。18、19 行目の ModelFactory. createInfModel（reasoner, model）メソッドにより InfModel オブジェクトを生成する。

図 26 に示すように、Jena の lib フォルダに入っている 13 個の jar ファイルをクラスパスに含むようにしてコンパイルする。

図 27 より、「WORK_CLOTHES（仕事着）」が「CLOTHES（服）」をサブクラスに持つことを表示する。

図 28 より、「WORK_CLOTHES（仕事着）」が「SHIGOTOGI_SHIGOTOSHI（シゴトギ、シゴトシ）」と「HADAKKO（ハダッコ）」をサブクラスに持つことを表示する。このタグは元の図 24 および図 25 の OWL に直接含まれていなかった情報である。このことから推論を行ったことでタグが自動的に追加されたことがわかる。

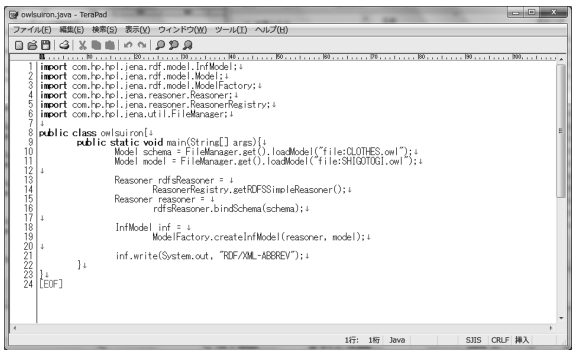


図 25　推論を行うためのプログラム

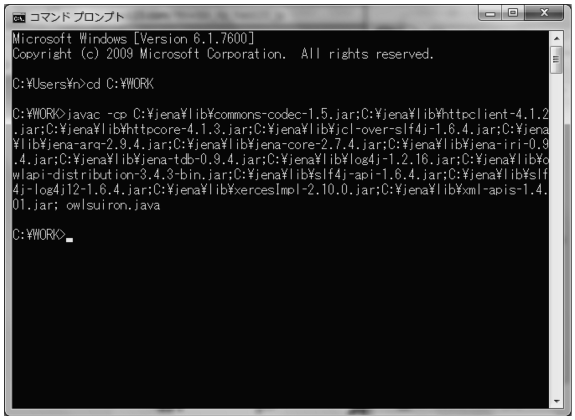


図 26　推論プログラムのコンパイル

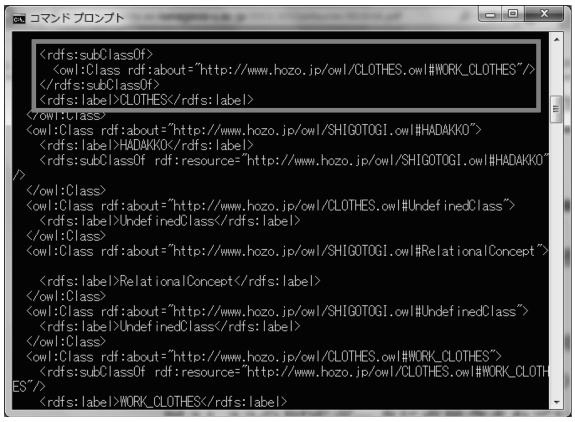


図 27　推論実行結果 1

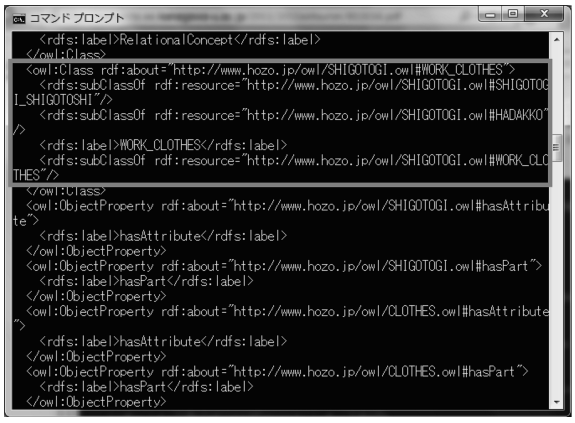


図 28　推論実行結果 2

4. むすび

本研究では、只見町インターネット・エコミュージアムの「キーワード」検索の改善のために「キーワード」の増加を行った。OWL を使用した推論システムの結果、コンピュータが民具カードに記載されていない情報を推論した。また、新たな関係性も示された。本研究により、キーワードの増加による「キーワード」検索の改善に近づいた。本研究において、法造を用いて OWL を自動的に作成可能であるが、細かな部分は手動で修正する必要があった。また、記述のミスがあった場合の修正に手間が掛かるなどの問題点があった。

今後の課題としては、Excel などに記述されている情報を利用できる Ontology 構築ツールの開発やキーワードにシソーラスを用いるシステムと、構築された Ontology を使用した検索システムの開発が挙げられる。

3　自己組織化可能な群知能を用いた情報リソースの管理

提案管理手法は、データ管理の過程に応じて自己組織化する手法にも適用可能である。

1. まえがき

近年、スマートフォン、タブレット端末、PC など個人の持つ電子端末の量が増えている。それに準じ、管理する情報やファイルの数も増してきている。[37] フラッシュメモリや SD カードといったメモリーカードによる端末間でのファイルの移動、クラウド[39]による複数端末からのアクセスも増えている。これらによる複製、移動による作業で個々の端末に在るファイル数は増加の一途を辿る。ファイルの数が増えると管理の手間も増加し、ユーザが自らファイルを管理することは困難になると考えられる。また、現在、増加し続けているファイルは、木構造（階層構造）のディレクトリで管理されている。木構造の中のファイルは静的であり自ら移動することはない。すなわち、木構造のディレクトリのみで管理するという一つのルールのみに縛られ、多様な方向でのファイル管理をすることはできない。

他の問題として、複数端末で多くの情報を扱うため、管理が分散・複雑・煩雑化していることが挙げられる。例えば、複数端末間でファイル管理を行う際、端末間でデータが分散してしまうのに加えて、個々の端末内では、依然として、従来から用いられている木構造のファイルシステムを使用するため、特定のファイルを探すことが困難になると考えられる。[38],[20] これは一つのファイルが、ディレクトリという一つの属性しか持たないために引き起こされる、いわゆる“こうもり問題”が存在しているためである。

例えば、一つの写真データ（画像データ）があったとする。昨今ではこの写真データには基本情報の他に Exif 情報と呼ばれるメタデータが付属されている。撮影日、タイトル、撮影場所、撮影者、画像サイズ、ファイルサイズ、拡張子、カメラの種類など、他にも数十種類の属性が付いている。このような写真データが大量にあった場合、任意の属性でファイルにまとめるのは一つまでであり、図 29 で示すように複数の属性を同時に管理するのは非常に難しい。

これらのデータはソートすることで検索すること自体は安易であるが、検索も一つの属性のみであ

り、問題の解決には至らない。また、視覚的な面で欲しいデータがどこにあるのか直感的に分かりにくく、検索をするにしても時間が掛かるため、個人や組織の活動を少なからず阻害している。

最近では、情報リソースの関係性を表すために、UI（User Interface）など視覚的な面の研究も盛んに行われている[58],[59],[21],[22]。Microsoft が Windows Vista へ搭載するために、WinFS[23]と呼ばれる統合ファイルシステムの開発を行った。また、Semantic ファイルシステム[38]と呼ばれるタグの概念を用いたファイルシステムの研究も行われている。Semantic ファイルシステムでは複数の要素を一括で管理できるタグの概念が取り入れられており、関連性のファイルを探し出すことは容易になるが、目に見える形で整理はされておらず、瞬時に目的のファイルを見つけ出すのは困難である。

プロパティ	値
撮影日	2013/10/10 12:34
撮影場所(GPS情報)	35.360496, 138.727284
製造元	Canon
機器名	Canon IXY 600F
解像度	3000 × 4000
撮影方向	
シャッタースピード	1/160
F値	2.6
サムネイル	ー
⋮	

図 29　写真データの Exif 情報

本研究では、上記に示したファイルシステムにおける問題と、新しいファイルシステムの問題点に対処すべく木構造の静的ファイル管理を脱した動的なファイル管理を提案する。大量のファイルや情報リソースを人間が扱いやすいシステムを作ることを目的とする。

2. 関連研究

1. ファイルシステム

ファイルシステムは、コンピュータにおける情報リソースを管理するための、オペレーションシステムが持つ機能の一つである[24]。前述のとおり、現在ほとんどのコンピュータではディレクトリ、つまり階層構造を持つファイルシステムが採用されている。階層構造は、ディレクトリの中にファイルや他のディレクトリがあり、その中にもディレクトリが存在している。このような概念は、ファイルの整理をするという観点で優れている。一般的に、ファイルシステムが持つ機能として、ファイルの移動、ファイルの削除、ファイルのコピーがある。また、OS ごとに異なるファイルシステムが用いられている。Windows では、NTFS が、Mac OS では、HFS が用いられている。

近年では、増加するファイルに対し、スムーズにファイルを見つけるためのデスクトップファイル検索も多く研究がなされている。Google が研究開発を行った Google Desktop[25]があり、これは HTML、文章ファイル、チャットの履歴などをあらかじめ残しておき、履歴内でファイル間の関係性を抽出し、検索を行っている。また、似たようなファイル検索として、Windows Desktop[26]、Yahoo! Desktop[27]がある。

2. 群知能

群知能（Swarm Intelligence）とは、昆虫や動物が採餌行動や防衛などのために行う生物の集団行動に着目し、最適化問題を解くために作られた人工知能である。魚の群れから着想を得た PSO（Particle Swarm Optimization）や、蟻の採餌行動から着想を得た ACO（Ant Colony Optimization）、ミツバチの群れから着想を得た ABC（Artificial Bee Colony Algorithm[28]）など、様々な群知能が開発されている。群知能では一つ一つの個体（虫、魚、鳥）を particle（エージェント）として扱い、群れを形成させている。最近ではこれらの particle がさまざまな法則に従って群れをなすための研究がなされているが、この群れをなすその“ふるまい”に着想を得て、ファイルシステムと関連付けをさせている。ファイルシステムと群知能を組み合わせることで、より見やすく使いやすいファイルシステムの開発につながる。

3. boid

boid（bird android）とは 80 年代、アメリカで開発された鳥の群れを再現するためのアルゴリズムである。boid は Separation, Alignment, Cohesion で構成されている。Separation は particle 間での斥力、Alignment は並列的な動きをするための力、Cohesion は particle 間での引力である。これら三つの力の組み合わせで、boid は“鳥らしさ”を再現する。

本研究では、各ファイルの持つパラメータの値を照らし合わせ、似ているということを表現して、図 30 で示すように boid の三つの要素と組み合わせ、群れを形成させていく。

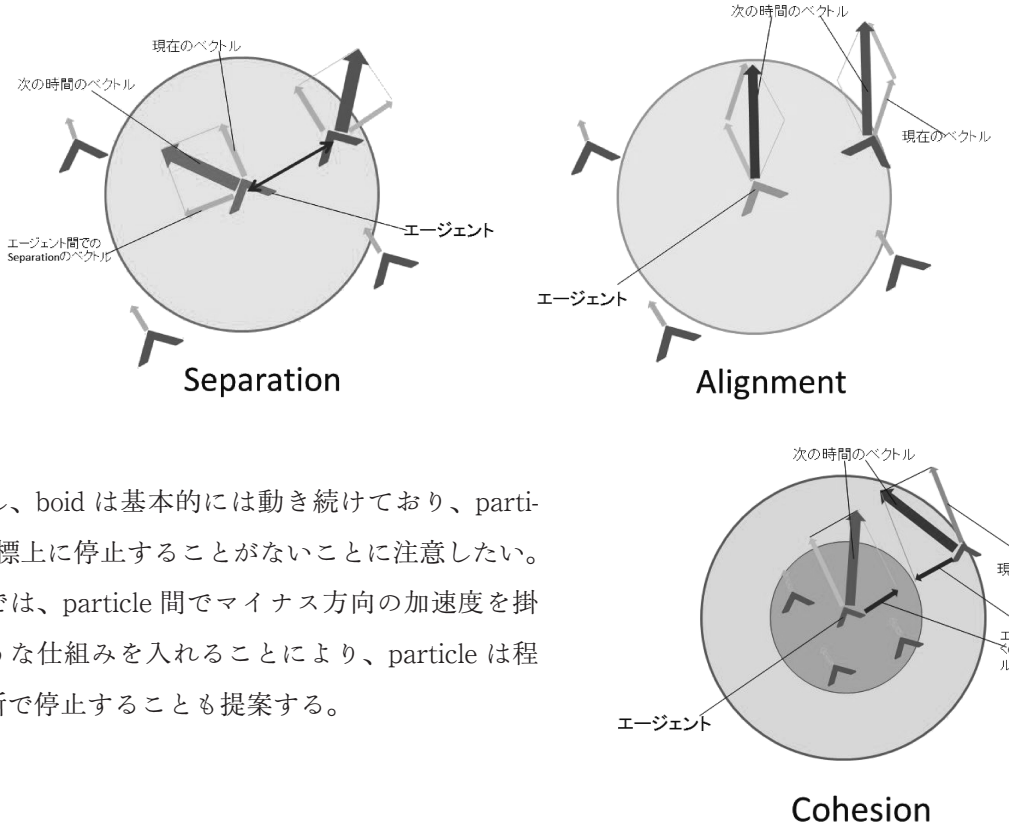
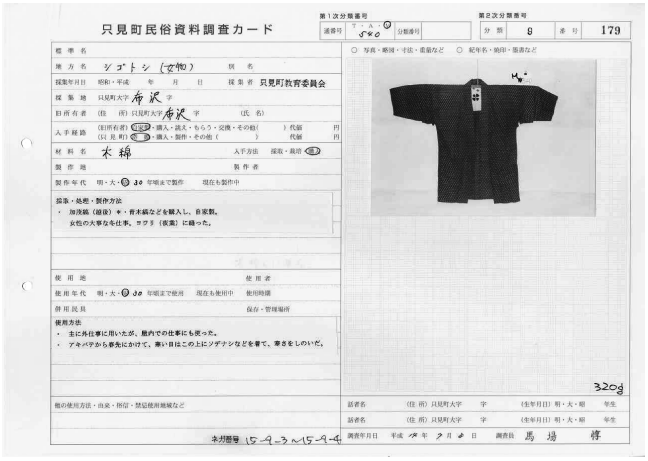


図 30　boid

ただし、boid は基本的には動き続けており、particle が座標上に停止することがないことに注意したい。本提案では、particle 間でマイナス方向の加速度を掛けるような仕組みを入れることにより、particle は程よい場所で停止することも提案する。

4. 非文字資料の検索

そもそも非文字資料とは、文字やデータ化、資料化されていない古い民俗道具のことである。ここでは「民具」と呼ぶ。民具の多くは、地方の農村で使われてきたものであり、古くは親から子へと使い方を伝えられてきたが、近年は機械化や少子高齢化などが進み完全に廃れてしまった。


只見町民俗資料調査カード

第1次分類番号 第2次分類番号

通番号 分類番号 分類 8 番号 179

標準名

地方名 ジゴトシ（女物） 別名

採集年月日 昭和・平成 年 月 日 採集者 只見町教育委員会

採集地 只見町大字 布沢 字

旧所有者 （住所）只見町大字 布沢 字 （氏名）

入手経路 （旧所有者）購入・譲入・読入・もらう・交換・その他（ ）代価 円
（只見町）購入・製作・その他（ ）代価 円

材料名 木綿 入手方法 採取・栽培

製作地 製作者

製作年代 明・大・昭 30 年頃まで製作 現在も製作中

採集・処理・製作方法
・ 加茂縞（銘仙）や、着古縞などを購入し、自家製。
女性の大事な仕事。ヨクヲ（夜業）に織った。

使用地 使用者

使用年代 明・大・昭 30 年頃まで使用 現在も使用中 使用時期

併用民具 保存・管理場所

使用方法
・ 主に外仕事に用いたが、屋内での仕事にも戻った。
・ アキバデから春先にかけて、寒い日はこの上にジゲテシなどを着て、寒さをしのいだ。

320g

他の使用方法・由来・所有・禁忌使用地域など

話者名 （住所）只見町大字 字 （生年月日）明・大・昭 年生

話者名 （住所）只見町大字 字 （生年月日）明・大・昭 年生

調査年月日 平成 19 年 7 月 8 日 調査員 馬場 悠

図 31　民具カード

民具の多くは近年では使われておらず、使い方、使用目的、使う時期などの保存が必要とされた。実際に民具を使っていた先人は、使い方などを記録した民具カードを残し、鬼籍に入った。しかし、民具カードの非文字資料は大量であり、各民具には様々なパラメータ（目的、地域、使用者など）があるため検索や整理が困難である。

これらの民具カードをデジタル化し、インターネット上で検索できるようなシステムの開発が行われている（只見町インターネット・エコミュージアム）[29], [30]。しかし、デジタル化してもインターネット・エコミュージアムにある民具のデータの要素の種類は多量にあり、整理が非常に困難である。そこでデジタル化されたデータをファイルと同じように扱うことで、本研究の提案と同様に群れとして形成させることが可能となり、道具として似ているものは近くに集まるような仕組みが形成され、民具（非文字資料）の検索に応用ができると考えている。

3. 群れを形成するための仕組み

1. 提案手法の概略

提案手法では、様々な属性を反映させたファイル管理を行うことが可能な、多次元管理の手法を導入する。多次元管理には、大きく二つの機能が備わっている。一つは、ファイル及びそれを示すアイコンが群れる仕組みになっているということ。もう一つは、一視点ではなく複数の視点から切り替え閲覧することが可能となることである。これら二つの機能により、一つの視点では、離れていて一見無関係なファイル同士でも、視点を変えることにより同じ群の中におり、中身、振る舞い、ジャンルなどの似たファイルやデータが集まり群れを形成するため、関係性のあるファイルであることが瞬時に分かるようになる。様々な属性でファイルを同時に管理することができる多次元管理として提案している。この多次元管理は、関連性の強い物が中心になるため、自分だけにチューニングされたものを作ることが可能である[31], [58]。そのため、個人にチューニングされた、あるいは特定の企業・コミュニティなどにチューニングすることが可能でもある。

本提案では、この多次元管理を実現する方法として、群知能を使用している。群れを形成しそれらを多次元から管理する仕組みを boid をベースに提案する。群知能は集まるということが目に見えるので、視覚的な観点から提案のベースとして取り入れている。また、群の階層化 particle は群をなすが、その群も大きな群をなす。階層のように、小さな群は他の小さな群れで集まり、より多くの群と

なる。各情報リソース（particle）が独自で自分の周りの検索をかけることで、自ら群れを組織することが可能である。小さな群れをとるため、現在一般的に使われているディレクトリの階層構造とは、隣り合う群が一目で分かる点で明確な差異がある。

2. particle の動きの定式化

particle が動き回る座標は、particle 間での相対的な距離のみに意味を持つアフィン空間であり、ユークリッド空間ではない。ここで i は particle の番号であり、j は対象となる particle の番号である。n は particle の数である。

$\vec{a_i}$ は、i 番目の particle に掛かる加速度となる。S は j の particle に対する近似度である。

$\overrightarrow{A_i}$ は合成ベクトルである。t は単位時間、$\overrightarrow{V_i}$ は i 番目の particle の速度、$\overrightarrow{Po_i}$ は i 番目の particle の今の座標となる。ゆくゆくは particle 自ら対象の particle との距離を計算することになるが、現段階ではシステム全体が位置を把握するため直交座標を用いている。$f_{Similarity}(P_i, P_j, e_k)$ を particle P_i, P_j 間の要素 e_k の類似度を表す関数とする。

w_k は各要素の重み付けとする。

$$\overrightarrow{a_{ij}} = S_{ij} = \sum_{k=1}^{m} w_k \cdot f_{Similarity}(P_i, P_j, e_k) \tag{1}$$

$$\overrightarrow{A_i} = \sum_{k=1}^{n} a_{ik} \tag{2}$$

$$\overrightarrow{V_i}(t) = \overrightarrow{V_i}(t-1) + \overrightarrow{A_i} \tag{3}$$

$$\overrightarrow{Po_i}(t) = \overrightarrow{Po_i}(t-1) + \overrightarrow{V_i} \tag{4}$$

群を作る際、i 番目の任意の particle における t 時の、j 番目の particle に対する動き V と位置 Po の計算式である。近似度はお互いの属性がどれだけ一致したかによって変化する。近似度が高ければ高いほど、お互いは強く引かれ合い、結果、より早くそしてより近くへと向かう。つまり、似ているものほど、近くへ向かう。Po は particle の座標のことである。一つ前の単位時間の座標位置から V を足し合わせる。

3. particle 同士が引き合う引力アルゴリズム

今回提案する一つの particle が引き合うためのアルゴリズムを以下の図 32 に示す。これは boid における Cohesion の部分に相当している。複数の particle が群を作る際、i 番目の任意の particle（P）における次の単位時間後の描画処理のための計算式である。このアルゴリズムにおいて、j は対象の particle の番号であり、k は particle が持つ属性の番号である（m はファイルが持つ属性の数である）。i 番目の particle と j 番目の particle の持つ属性が一致していればしているほど近似度 S が上がる。近似度 S の値が大きいほど、i 番目と j 番目の particle は互いに引かれ合い、加速度 a に加算されていく。速度 v は、計算された加速度 a を常に加算し、速度を変化させていく。全ての計算の後、加速度を足し合わせ、合成ベクトル $\overrightarrow{A_i}$ とする。図は particle 同士の単位時間での 1 回の動きを示したものである。

4. particle 同士が離れる斥力アルゴリズム

図 32 では、particle は近づく一方で、少しでも似ていれば最終的にはそれらは全て重なってしまう。そこで各 particle にテリトリー（*Territory*）を持たせる。テリトリーは、個々の particle が自分を中心に円状に持つ。テリトリーの半径は particle も変わらないが、対象の particle によってその大きさが変化する。対象の particle に対して斥力を働かせる。お互いの近似値 S が大きいほど、テリトリーは狭くなる。結果として、似ているもの同士ほど、より近づくことが可能であるといえる。particle i と particle j のテリトリーはデフォルトのテリトリー Te を、近似値 S で割ったものである。図 33 の例では、particle j は particle i の持つテリトリー範囲内には入っていないため、お互いはまだ引かれ合っている。

また、図 34 の例では二つの particle はお互いのテリトリー内に入っている。テリトリー内に入った場合、引き合うアルゴリズムから離れるアルゴリズムに切り替わる。離れるアルゴリズムは、テリトリー内に入ってしまえば、斥力の加速度はどれも変わらないでいる。しかし、前述の通りテリトリーの範囲やテリトリー内に入ったときの速度及び方向が個々で違うため、結果として動きは particle

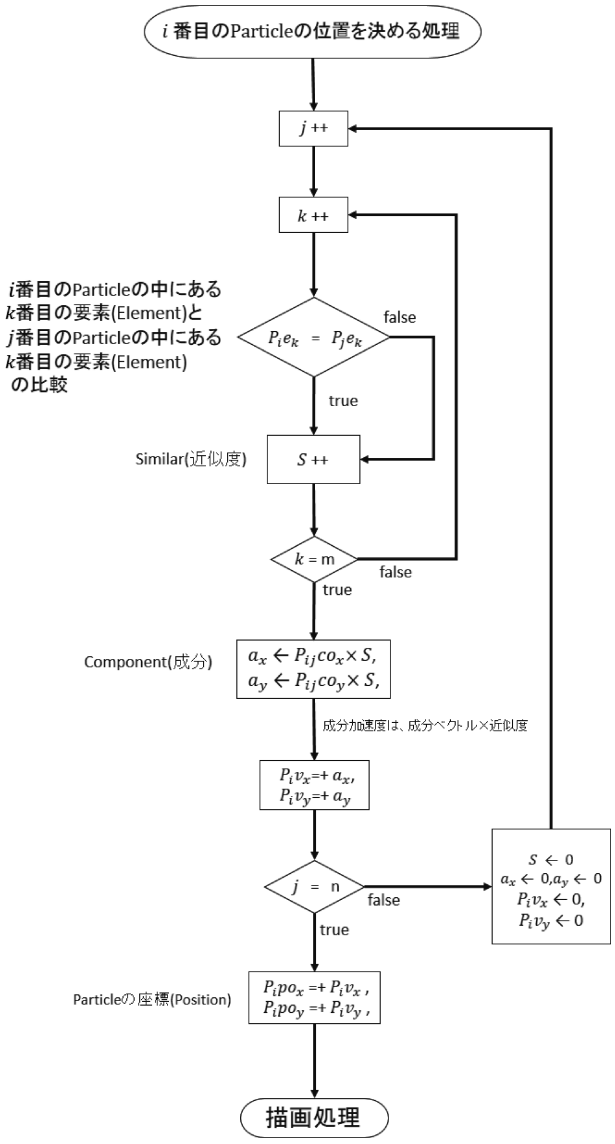


図 32　引力のアルゴリズム

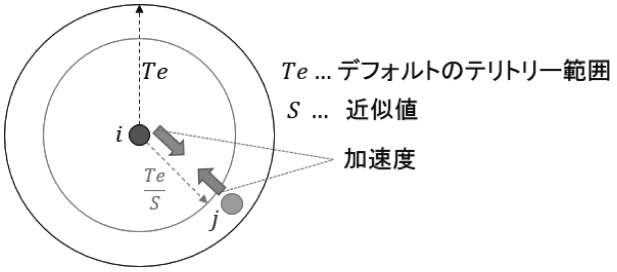


図 33　particle とテリトリー

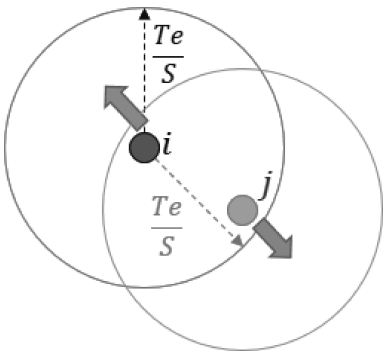


図 34　particle の斥力

によってまちまちである。また、加速度は互いに逆方向を向いていても実際に動いている方向は、互いに向き合っていることもあるので注意されたい。

引力のアルゴリズムに斥力のアルゴリズムを加えると以下に示した図 35 のようになる。

太線で示した部分が、斥力の計算式である。D_{ij} は particle 間の距離である。テリトリー内では、逆方向の加速度を加える。この図では、マイナス加速度を近似値 S としているが、この加速度が近似値 S である必要はない。定数でも特に問題はない。

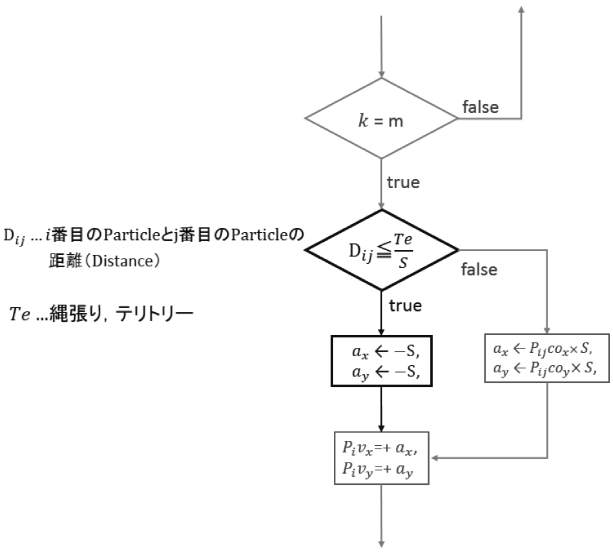


図 35　斥力のアルゴリズム

4. 非文字資料のデータへの適用

1. 引力システムの仕組み

実際に、非文字資料（民具）の検索で開発を行っている。民具の csv を表にまとめたものを、各種別にアイコンをつけ座標上に出力している。ここでは、服、道具、祭事道具等に番号をつけ分類をしている。プログラムの実行時は、各民具アイコンは座標上に間隔が空いた状態でランダムな位置に置かれる。各民具には分類番号の他に、地方名、統一名、作業名、材料、採掘地、寄贈者が書かれている（ここではそれらを要素と呼ぶ)。各民具は自分の要素を、任意の一つの民具の要素と照らし合わせ、一致しているものがないか探す。一致している要素があれば、民具はお互いに座標として 1 単位分近づく。

2. 実装

今回は android SDK（android シミュレータ）を使用し、実装を行った。android のバージョン及び API レベルは、android 4.0.3 API level 15 である。

実行直後　図 36 は、アプリケーションの実行直後のスクリーンショット画像である。民具アイコンが均等に並んでいるのがわかる。しかし、アイコンが集まっておらず、ランダムに並べてあるだけの状態である。各民具の各要素と一致がないか、すべての民具と総当たりで行い、ゆっくりと群れを形成させていく。

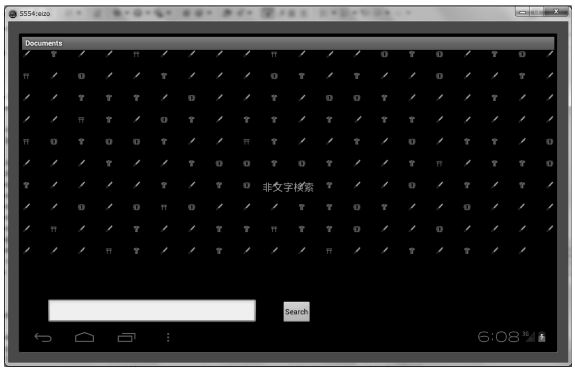


図 36　非文字に資料の引力システム 1

実行の数十分後　図 37 が、アプリケーション実行数分後のスクリーンショット画像である。民具アイコン同士が少しずつ集まってきているのが分かる。尚、このアイコンの位置はドラッグすることで、任意の位置に移動させることも可能である。これにより、システムによっ

て、集まった任意の民具アイコンを人間の意志によって変更させることができる。

民具アイコンの集合　図 38 は、図 37 の状態からさらに数十分経ったアプリケーションのスクリーンショット画像である。図 39 で示したように、アイコンの群れは衣類、生活道具・用具、家具、信仰・宗教用具の 4 種類で構成される。民具アイコンが画面のやや左に集合しているのがわかる。左に集まっている物ほど、互いに関連性が強い物である。逆に、右側にあるアイコンは要素のどこかで一致している物はあるものの、その種類が少ない民具である。そのため、左側に比べ、右側はややアイコンが疎らである。

また、各民具は必ず一つ以上の民具と一致する要素が存在している。このアプリケーションでは、他の民具と要素が一つでも一致しているものがあれば、その一致する要素の数に関わらず、決まった 1 単位分近づく。さらに、このアプリケーションでは、斥力のアルゴリズム及びテリトリーの概念は存在していない。そのため、時間が無限に近づくにつれ、一点に収束をしてしまう。

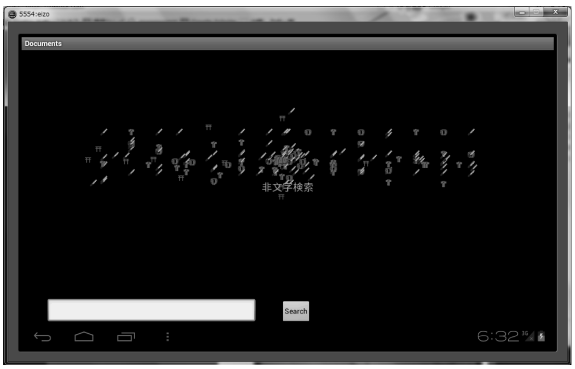


図 37　非文字資料における引力システム 2

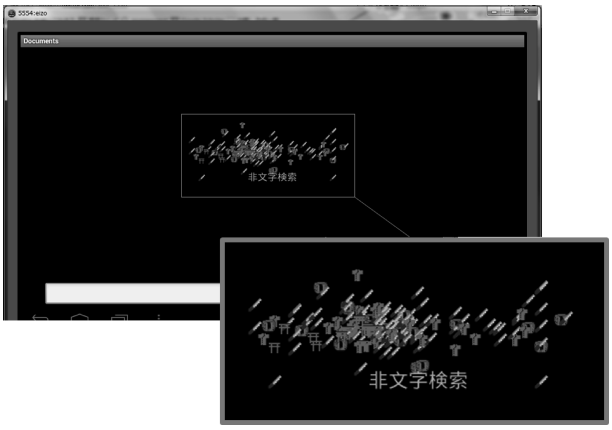


図 38　非文字資料における引力システム 3

図 39　非文字資料のアイコンの種類

5. 教育への応用

日本の地方で古くより伝わっている民具の多くは、現代の機械化や少子高齢化により後世に伝えられていない。日本の歴史の中で先人がどう生活をしてきたのか、どう食べ物を作ってきたかを後世に残していく教育は非常に重要なことであると考えている。古いものをただそのまま伝えるというだけでなく、その時代に適合した教育方法に合わせながら古い資料を伝承していくということが未来へ長く残していくために必要であると考える。非文字資料をデジタル化し残すことにより、現代人や教育を受ける子供たちは古い資料を扱いやすくそして触れやすくなる。

本研究では、座標上でデータを管理しているので、パソコンやデジタルが苦手な人でも視覚的かつ直感的に資料を取り出せ扱えるというのも、多くの人に受け入れられやすい仕組みとなっている。古いアナログの資料と新しいデジタルの仕組みを組み合わせることで、日本の教育のためのツールの一つとして貢献できる。

6. むすび

本研究では、群知能を利用した新しいファイルシステムが群れるためのアルゴリズムを提案した。この提案の特徴としては、単純に集まるだけでなく、テリトリーという概念を持ち、斥力が働くという部分である。また、非文字資料における引力システムにおいては、開発段階ではあるが群れを形成させることができた。今後の取り組みとしては、今回提案した引き合う引力アルゴリズム及び離れる斥力アルゴリズムと、非文字資料検索の引力システムを組み合わせ、より実用的なシステムを作りあげていく。特に、近似度 S は今回の場合、1 か 0 のみであるため、少しでも似ている particle 同士は引かれ合ってしまい、最終的には収束してしまう。そのため、釣り合う前の遷移段階では、群れを形成しているが、釣り合ってしまうと、群れは形成されない。また、テリトリーや斥力の概念が存在しないことも収束してしまう原因の一つである。斥力がうまく働いたとしても振動してしまう可能性も否定できない。任意の particle 間での近似度の差別化及び、斥力を組み込むことが今後の課題といえる。

4　ACO を用いた検索過程を重視した検索手法

1. まえがき

近年、インターネットを利用してさまざまな情報を調べることができる。また、現代の企業活動において企業間や顧客とのクラウドを介した情報のやり取りが増えてきている。調べたいものを検索すればすぐに欲しい情報は手に入るであろう。しかし、欲しい情報の一般的なものだけがわかり、その情報についての詳しい内容まではわからない場合が多い。そこで、調べたいものに対して有益となる情報を多く取得しながら検索を行うことで、より理解を深めることができるのではないかと考えた。

例えば、数学の問題を調べるとした場合、その問題を検索したら答えは簡単に求めることができる。しかし、答えだけがわかり、その答えがどのように導かれて解かれたかわからない場合が多い。だが、問題を調べるとき、その問題に使われている公式、定理の存在を知り、公式の使い方を知り解き方を理解して問題の答えがわかると、その問題に対して有益な情報を知ることができ、より理解が深まる。また、検索するにあたって他者が行った検索を参考にすることで、その人が問題に対してどのような情報を得て理解したかを自分の検索に影響させることによって、その人の得た有益な情報を自分にも取り入れることができるのではないかと考えられる。

検索の手助けをしてくれるシステムとしては「推薦システム（Recommender System）」が提案されている。[34], [35] 推薦システムというのは、インターネットソフトウェアツールでオンラインショップなどに対してユーザが調べたいことを見つける手助けをするものである。個人の好みのアイテムや製品を推奨したり、提案したりする。これらのシステムは、多くの場合顧客が購入する可能性がある製品を提示することによって収入を増やすためのマーケティングツールとして、電子商取引のウェブサイトで使用されている。ユーザの個人的な好みを学習し、ユーザに合った推薦を提供するためのシステムである。

「他者の行った検索を参考にする」という機能を実現するために、推薦システムを拡張し、群知能の一種のアントコロニー最適化アルゴリズムを組み込むことで検索過程を推薦する機能を実現する。

多くの工学設計問題では、準最適解で十分な場合が多いので、群知能では厳密解（大域的最適解）ではなく、解に近いものを探索するのに用いられるメタヒューリスティック手法の一つである。群知能にはさまざまな種類があり、「巡回セールスマン問題」など、多くの最適解を求めるものではアントコロニー最適化（ACO）や粒子群最適化（PSO）が用いられている[32]。また、一つの大域的最適解ではなく、許容できる複数解を効率的に求めることを目的とした改良法が提案されており[1]、とくに検索手法を求める用途には適している。また、一般に群知能は最適解を求めるために用いられるが[32],[33]、本研究では最適解に至るまでの過程を重要視する。おおよその経路を、有益になる情報を得るような経路、評価の良いものを探り、理解を深められるような検索方法を提案する。

推薦システムは群知能の一つである PSO を組み込み、推薦システムに改良を加えた。ユーザの好みを選択する際に PSO のアルゴリズムを適応させることにより、ユーザの個人的な好みをより正確に予測して、ユーザに対する推薦を改善している[35]。

この推薦システムでは PSO のみを使用している。また、個人の好みといった「個人」のデータを利用している。ここにさらに「他者」のデータを加えることで、自分のデータと他者のデータを組み合わせ、より良いものを探すことができるのではないかと考えた。この他者のデータを利用する際に ACO を用いることで表現できるか試みる。

たとえば、CiNii のようなさまざまな論文を検索するツールの場合、その検索に関係する論文が新しいものから順に出てくる。ここに他者がその検索でどのような論文を見たかを自分の検索に取り入れることで、他者が参考になった論文を自分も見ることができ、理解がさらに深まる。

また、メタ検索手順に Pheromone を付加させて、他社の検索の意図を自分の検索に取り入れられるか検討する。

たとえば、学生が研究を進めていく上で、学生は研究についてさまざまなものを調べることになる。調べるものがある程度わかっていたとしても、検索するときに数多くの情報を一つ一つ見ていくことになる。こうなってしまうと調べものにたくさんの時間を費やしてしまう。その時に、教授がどんな論文を読んだか、どのようなやり方で検索を行っているかといった、どのように情報を集めているかという教授の意図を自分の検索に取り入れる。こうすることによって、学生が検索を行う際に、教授の考えを取り入れた検索となり、教授が参考になったものや教授が優先するもの（情報が新しいものなど）を出してくれる検索となり、学生の検索の手助けになるのではないかと考えた。このような教授の知識をメタ検索手順と定義し、この手順に Pheromone を付加させ、学生もこの手順で検索を行うようにする考えである。

一般に群知能は最適解を求めるために用いられるが[32],[33]、本研究では最適解に至るまでの過程を重要視する。おおよその経路を、有益になる情報を得るような経路、評価の良いものを探り、理解を深められるような検索方法を提案する。

2. 群知能

1. 一般的な群知能

群知能は、メタヒューリスティック手法の一つで、厳密解（大域的最適解）ではなく解に近いものを探索するのに用いられる。それらは個々の行動が単純で、なおかつ群全体の行動を導くような管理

者がいないにもかかわらず、群全体の高度に統制されたかのようにふるまう。群知能はこのふるまいを応用し現実社会の困難な問題を解決するために用いられる。

困難な問題の一つに組み合わせ最適化問題というものがあり、有限個の解集合の中から各々の解に対応した評価値が最も良い解を求めるものである。この問題に対して現実的な計算時間内に最適解ではなくても十分許容可能な精度の解を求める手法としてメタヒューリスティックが使われる。群知能にはさまざまな種類がある。

2. ACO

アントコロニー最適化（Ant Colony Optimization）とは、群知能の一つで、現実のアリの採餌活動からヒントを得た組み合わせ最適化問題に対するメタヒューリスティック手法の一つである。[33], [36] アリは採餌活動の際に、他のアリとの情報交換の手段として Pheromone と呼ばれる芳香性の物質を地表に分泌することが観察されている。各個体は、餌を探索する際や巣に持ち帰る際に、他の個体が過去に分泌した Pheromone の分布にしたがって確率的な経路選択を行い、その意思決定の結果を再び Pheromone として地表に分泌する。その結果、多くのアリが選択した経路には多くの Pheromone が分泌され、さらに多くのアリをその経路に誘引するという正のフィードバック効果がもたらされる。このアリの群行動を組み合わせて最適化問題の解の探索に応用した最適化手法 ACO が提案されている。図 40 は ACO の動き方を示している。アリが餌を見つけると Pheromone を残しながら巣に戻る。しかし時間とともに Pheromone の痕跡は蒸発しはじめ、その吸引力がなくなっていく。その経路が長いほど Pheromone は蒸発しやすい。それに対し、経路が短ければ行進にも時間がかからず、Pheromone が蒸発するよりも早く補強されるため、Pheromone 濃度は高いまま保たれる。

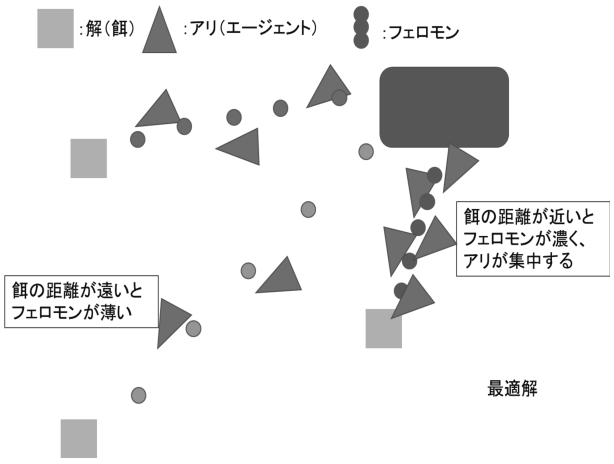


図 40　ACO の動作

ACO が適用される問題は「巡回セールスマン問題」であることが多い。複数の都市を各々一度だけ訪問して出発都市に戻ってくる際に移動距離が最短となる訪問順序を求める問題である。

この問題に適用させた AntSystem というものがある。[36] AntSystem とは、アリが餌を巣に運ぶ行動とその際に分泌される Pheromone をモデル化したもので、探索エージェントはヒューリスティックな情報である都市間の距離情報と Pheromone 情報をもとに探索を行う。AntSystem の特徴はエージェントの探索にヒューリスティックス値と呼ばれる探索領域への静的な評価値と、Pheromone と呼ばれる探索領域への動的な評価値を組み合わせたところにある。エージェント k が時点 t において都市 i から次に訪問できる都市集合 N_k の未訪問都市 l の中で、都市 j への移動確率 $P_{ij}^k(t)$ は次式で与えられる。

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{l \in N^{k}}[\tau_{il}(t)]^{\alpha}[\eta_{il}(t)]^{\beta}} \tag{5}$$

ここで $\tau_{ij}(t)$ は都市 i から都市 j の間に蓄積されたフェロモン量 $\eta_{ij}(t)$ はヒューリスティックな情報で、都市間の距離の逆数として与えられ、α と β は Pheromone 情報と距離情報の重みを定義するパラメータである。[33]

図 41 は巡回セールスマン問題における ACO の動き方である。アリは距離が短いと Pheromone を濃く残すので、距離が短いルートを選ぶ。

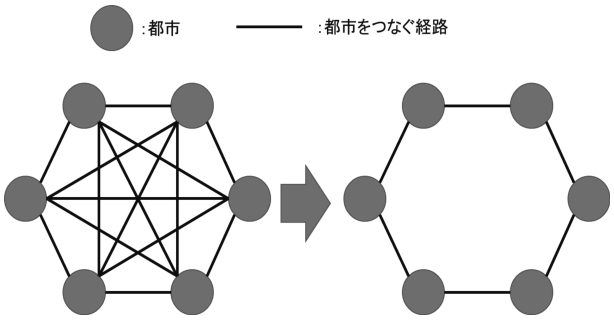


図 41　巡回セールスマン問題における ACO を用いた手法

Pheromone 情報は蓄積と蒸発を繰り返し、Pheromone が多いほどエージェントに対する誘因性が高まる。それにより、さらなるエージェントが都市間を移動して Pheromone を分泌することで、他のエージェントがその都市間を移動する確率が高まるのが基本的な原理である。Pheromone はいわば過去の探索情報の蓄積であり、この Pheromone のコントロールが AntSystem の探索性能を決定する。[36]

また、ACO にはさまざまな拡張手法がある。階層的クラスタリングのためのアリの自己組織化モデルというものがある。[40] アリはあるサポートと呼ばれるスタート地点から出発する。アリはこのサポートへつながり始め、つながったアリにさらにつながっていく。あるアリがつながったとき、それは組織の一部となり、他のアリがこのアリを超えて移動する。もしくはそのアリとさらにつながる。この組織は、アリによって行われる局所的な行動に従い、時間経過とともに成長していく。移動中のアリは局所的な組織や視覚的に惹きつけられる（たとえば目的地の影響を受ける）。

このモデルはアリがその他のアリを探す際、似ているものを探す。その時、データベースでアルゴリズムを検証する際、このような類似度計算を行っている。

$$Sim(i,j) = 1 - \sqrt{\frac{1}{M}\sum_{k=1}^{M}(v_{ik} - v_{jk})^{2}} \tag{6}$$

M は属性の数。v_{ik} はデータ v_i における第 k 番目の属性値である。[51]

3. PSO

粒子群最適化（Particle Swarm Optimization）は粒子群により解を探索するもので、連続問題に優れた解探索性能を持ち、メタ戦略の中の GA（遺伝的アルゴリズム）などと異なる手法として注目されている。

これは多次元空間において位置と速度を持つ粒子群でモデル化される。これらの粒子は空間を飛びまわり、最善な位置を探す。位置の評価は適応度関数で行う。群れの粒子は良い位置について情報交換し、それに基づいて自身の位置と速度を調整する。

PSO は群れ上の粒子（エージェント）が情報を共有しながら多次元空間内を探索する群知能アルゴリズムである。各粒子は位置ベクトル（x_i）と速度ベクトル（v_i）で特徴づけられる。ここで下付

$i(i=1,\cdots,N)$は粒子番号を表す。各粒子は自身が探索の過程で発見した最良解［$pbest_i$］、および群れ全体で共有する（全粒子中の）最良解［$gbest$］を用いて、探索の終了条件が満たされるまで、群を構築する全粒子でより良い解を探索し続ける。具体的には一つ前の速度ベクトル（v_i^k）、自身が探索の過程で発見した最良解［$pbest_i^k$］、群れ全体で共用している最良解［$gbest^k$］の線形結合として新たな速度ベクトル（v_i^{k+l}）を生成し、新たな探索点（x_i^{k+l}）まで移動する。ここで上付 k は探索回数を表し、$k+l$ 回目を探索における粒子 i の速度ベクトル（v_i^{k+l}）と位置ベクトル（x_i^{k+l}）は次式によって生成および更新される。

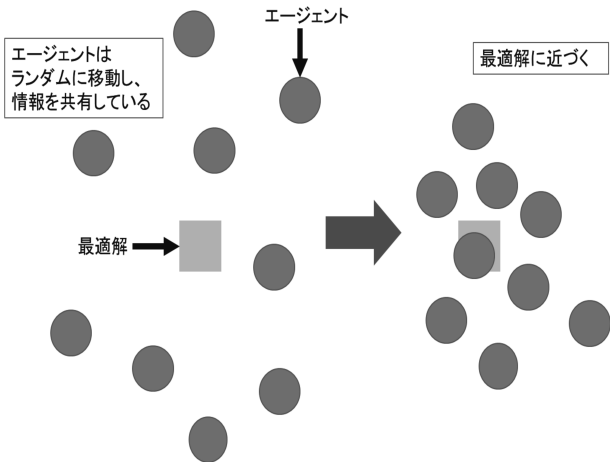


図 42　PSO の動作

$$v_i^{k+l} = \omega \cdot v_i^k + C_1 \cdot r_1 \cdot (pbest_i^k - x_i^k) + C_2 \cdot r_2 \cdot (pbest_i^k - x_i^k) \tag{7}$$

$$x_i^{k+l} = x_i^k + v_i^{k+L} \tag{8}$$

ここで、r_1, r_2 は［0, 1］の一様乱数、ω は粒子の慣性力係数、C_1, C_2 は学習係数であり、それぞれの項に対する重みの係数として機能する。[32]

図 42 は PSO の動き方である。各エージェントがランダムに動き、最終的に最適解の位置に動く。

3. 推薦システム

推薦システムとは、その人の好みに基づいて顧客に商品や製品を推奨したり提案するシステムを指す。これらのシステムは、多くの場合顧客が購入する可能性があるものを提示するマーケティングツールとして、電子商取引のウェブサイトで使用される。このようなウェブサイト中で、推薦システムを用いることで、顧客ごとのニーズを取得することができる。

参考文献［41］では、Movie-Lens というウェブサイトを用いて推薦システムの提案を行っている。この推薦システムにおいては、あらかじめ各ユーザごとに年齢や映画の好みなどのプロフィールを作成し、各ユーザの映画の履歴情報を蓄積することで、同様のプロフィールを持つユーザに対して、適切な映画を推薦することを可能としている。プロフィールが似ているユーザは、類似度を用いて判断される。

さらに、この推薦システムでは、映画を particle としてとらえ、PSO を用いて、最も良い映画を提供することができる。

4. 提案推薦システム

1. 教育現場において必要な検索とは

例えば、大学の研究室において、学生が研究を始める際のことを考えてみる。研究に配属になった学生が研究を始める際、まずは自分の研究の動向を調べるために、情報収集を行うだろう。しかし、ほとんどの学生がその研究についての知識は浅いものである。たとえば、論文の検索を行うときには

一つのキーワードのみで検索を行って情報収集を行うだろう。そのキーワードの関連した情報は出てくるが、どの情報から手を付ければいいのかはそのキーワードにおいての知識がある程度ないと分からないものであり、手当たり次第に探すことになる。さらに情報を見ていくうちにそのキーワードと関連のあるキーワードの出現により、本来の研究とは方向性が違ったキーワードも出てくる可能性がある。そうなってしまうとさらに情報収集に時間を費やしてしまい、情報の整理もままならず、研究がうまくいかなくなる恐れがある。

そこで、本研究では、検索をする際に、他者の検索履歴を参考にする検索方法を考える。たとえば、「他者」をその研究室の教授とし、学生が研究の参考文献を調べる際に、教授が文献を調べた際の検索履歴を参考にすることを考える。一般に、教授は知識が豊富で、数多くの学会を知っている。教授は、信頼性があり本研究室の分野に近いであろう学会や国際会議から、論文や国際会議原稿などを集めることができるだろう。一方で、学生は、学会や国際会議の情報を知らないまま、キーワードからヒットした多数の論文の中から、関連する文献を探すこととなる。この場合、仮に原稿の種類（論文誌、国外会議原稿、国内研究会原稿）や、学会や国際会議の種類をタグ付けし、そのタグの推移で検索を行うことができれば、学生が参考文献を調べた際にも関連のある論文を効率よく探すことができる可能性がある。

2. 提案手法

群知能を用いた検索および探索において、最短距離や解への速さなどを求めているのではなく、解に至るまでのおおよその経路を探るために群知能を用いる。おおよその経路について、有益となる情報が得られるような経路を探るような検索方法を、群知能を使って提案する。「有益となる情報を得るような経路」を探る上で、このような経路を「他者が通った経路」と考えた。他者が理解した検索を自分の検索に影響を与えることによって、自分も理解が深まると考える。

「他者が通った」という表現に ACO の Pheromone を用いる。他者が行った検索に Pheromone を残し、そして、自分が検索を行う際に、自分の検索と他者の検索の類似度を計算し、似ているものを提供する。その提供するものの中で、他者の Pheromone が強いものを提供させるようにする。

そうすることによって、他者が有益になった情報を自分に取り入れることができると考えた。

また、他者が行った検索に対し他者がどのような「意図」で検索を行ったかについてを視野に入れる。たとえば、他者が検索を行う際、まず新しい情報から調べ、そこから順に古いものを調べていくという探索を行ったとする。そこに検索したものと更新日時の新しいデータに Pheromone をつける。そして自分が検索を行うときに、同じ検索なら他者の検索したものが順に出てくるが、その他の検索に対しては他者の「新しいものから順に探索する」という考え方を自分にも反映した検索を行う。

このような他者の意図を自分の検索に取り入れようと考えている。対象物（URI）に Pheromone をつけることによって他者の検索で優先にしているものを自分の検索に影響を与えることができる。

本研究では推薦システムを用いた検索と ACO の検索の二つの要素を用いた方法を提案する。

3. 推薦システムを用いた検索ベース

今回の検索手法に推薦システムを用いたものを提案する。推薦システムは各ユーザごとに年齢や映画の好みなどのプロフィールを作成し、各ユーザの映画の履歴情報を蓄積することで、同様のプロフィールを持つユーザに対して、適切な映画を推薦することを可能としている。

まず、利用者の属性、対象物の属性、利用者の対象物に対する評価のデータベース化を行う。ここで、教授などの信頼できる人が集合に多く含まれるように設定する。候補決定のための利用者間の類似度に基づく評価式の解析を PSO の評価関数で行う。

次に利用者の過去の評価のデータ集合の利用者と対象物の属性および評価から、利用者の価値観の予測を行う。PSO で価値観を反映するための各要素の最適な重みの組み合わせを決定する。

そして、着目した利用者以外の利用者のデータ集合に予測した価値観を適用して推薦結果を提示する。推薦する対象物の候補に評価式を適用し推薦対象物を決定する。

プロフィールについて、利用者の属性、対象物（URI）の属性、利用者の対象物に対する評価（検索時の選択順序、検索の過程での適合度など)、ACO で求めた対象物（URI）探索履歴などの属性を決める。この属性をプロフィールとして行う。

そして、自分のプロフィールと似ているものを探す際に類似度計算を行う。ユーグリッド距離が近い利用者を評価の対象とする。

そして、同じような研究をしている研究者の検索履歴を使ってメタ検索の方法（メタ検索の戦略）を推薦する。この推薦された対象物（URI）の戦略を、自分の検索に取り入れることでより効率のよい検索を行うことができる。

あるプロフィールを持ったユーザが、検索に対してある戦略をとった回数に応じてレーティングが自動的につくようにすることで、どの戦略が良いかがわかり、良い戦略が選ばれる。

4. ACO を用いた検索ベース

今回の検索手法にもう一つ、ACO を用いたものを提案する。ACO の Pheromone を用いて、検索経路を探す。ACO の座標系に URL を使用する。こうすることにより、細かい位置を決めることができる。Pheromone は他者の通った道筋を自分もその道筋をたどっていくために使用している。Pheromone は他者の見たデータの URL に付加される。そして他者の検索で通った URL の推移を自分の検索においてその URL の推移で検索を行うことができるのではないかと考えた。

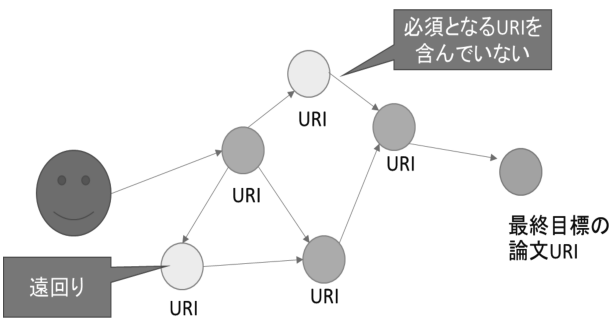


図 43　ACO による探索経路

検索した URI に Pheromone を分泌させ、利用者ごとの最適な URI の検索履歴情報を取得する。

推薦する対象物を URI とし、必須となる関連研究や基礎知識の URI を通る検索が必要になるので、この必須となる URI を通るような URI の探索経路を ACO を用いて最適化する。

ACO で見つけた解から戦略が生成される。

5. むすび

検索に不慣れな人でも、インターネットに偏在する情報リソースを有効活用し、検索目的以外の有用な情報リソースを参照可能な方式について考察した。ポイントは、「他者が参照した情報リソースの軌跡」を有効に利用すること、そして「情報リソースの軌跡を残した『他者のプロフィール』」を利用することである。

Ant Colony Optimization は、個々の情報リソースまで至る軌跡を追跡可能である。しかし、他者のプロフィールを反映させる概念は Pheromone に想定されていない。

一方、推薦システムはユーザのプロフィールを利用し、個々の情報リソースが推薦される群知能の仕組みを持つ。しかし、推薦される対象は個々の情報リソースであって、参照された情報リソースの軌跡を得る仕組みはない。

今回は ACO と推薦システムという二つの群知能システムを融合する「検索推薦システム」の概念を提案した。

今後、「検索推薦システム」概念をさらに研究し、その効用を明らかにする。

5　価値交換システムにおけるゲーム理論的解析

多様な価値を表現可能にする価値の交換システムを提案する。ここでいう価値とは、単なる金銭的な価値ではなく、地域通貨的に多様な価値を提案し、サービスに対して地域通貨的価値を付与した上で価値の交換システムを作ることを目的としている。これまでに、異なる価値観を持つ二者間の価値交換システムについて検討がなされてきた。しかし、これまで提案されてきた異なる価値観を持つ二者間の価値交換システムでは、各ユーザが満足する効用が得られるかどうかは未解決であり、複数人の場合にそのまま適用することができない。そこで、本稿ではゲーム理論を用いて特定の条件下で、n 人の各ユーザにおける効用が満足するようなモデルを提案する。

1. まえがき

今日、ほとんどのものが法定通貨で取引が行われている。しかし、法定通貨だけだと特定のコミュニティの価値が反映されずうまく流通しない[41]。例えば、大型商店街と小型商店街があった場合、大型商店街にお金が集中してしまい、小型商店街では情報やサービスが流通しない可能性がある。そこで、近年地域通貨が導入され、小型商店街だけで使える地域通貨により、コミュニティで共通の価値観を反映させることで、流通しやすくなってきた[42],[43],[44]。

また、別の例を挙げると、例えば「高齢者を大切にしよう」というコミュニティがあり、そのコミュニティ内で高齢者の話し相手になるとポイントが付く制度があったとする。中にはポイントは要らず、純粋に高齢者の話し相手になってあげたいという人もいれば、ポイントが付くとしても高齢者の人とは話しにくいという人もいるかもしれない。この様に同じコミュニティでも価値間が異なると流通しない可能性がある。そこで、本稿では、コミュニティ内での異なる価値を考慮し、情報を流通させたときの変化について考察する。ここでいう価値とは単なる金銭的な価値ではなく、地域通貨的に多様な価値を意味している。これまでに、異なる価値観を持つ二者間の価値交換システムについて検