

蟻コロニー最適化を用いた渋滞時の巡回経路探索

2003MT095 関原 啓太

2005MT041 伊藤 祐基

2005MT123 和久田 崇

指導教員 河野 浩之

1 はじめに

現在のカーナビには、VICS(Vehicle Information and Communication System)などの普及により、渋滞や交通規制などの道路交通情報をリアルタイムに受信し、動的環境下で新たな推奨経路を生成するカーナビも市販されている。しかし、複数の目的地を設定し、最後に出発地点に戻るような経路を案内する機能を持ったカーナビはない。このような巡回路を求めることを巡回セールスマン問題(TSP)という[4]。

そこで本研究では、TSPを解くための代表的な解法(最近傍法、貪欲法、最遠挿入法、2-opt法、山登り法、焼きなまし法、蟻コロニー最適化)を比較し、より最適解に近づくことができるアルゴリズムを選択する。そして、道路交通情報をリアルタイムに計算し、動的にTSPを解く機能を提案する。まず、現在のカーナビにTSPを解く機能が用いられていないことを示し、各アルゴリズムの中から、最近傍法、2-opt法、蟻コロニー最適化を挙げ、性能評価を行う。そして、ACOは、フェロモンというパラメータを用いて経路探索を行っているため、前回の解で用いたフェロモン量を利用することで、より効率の良い経路を探索すると思われる。よって我々はACOに着目し、CA法によるシミュレーションを行った。そして、動的環境下で効率の良い巡回経路を求めることができているかを確かめるために、渋滞状況を考慮しない場合との比較を行った。その結果、動的環境下では経路を迂回することで距離は長くなるが、コストはより小さくなることが分かった。次に我々が採用したACOについて、ACOが本研究に適しているかを確かめるために、精度と計算時間について2-opt法と比較する。

2 TSPに関する研究

2.1 カーナビの機能

現在市販されているカーナビの主な機能として、

1. 静的環境における渋滞データバンク探索
2. 動的環境におけるスイテルート案内
3. 目的地に着くまでの経由地を最大5ヵ所まで設定可能

が挙げられる。現在のカーナビには、TSPに似た機能として何ヵ所か経由地点を設定することができるが、この場合各地点を回る順番は利用者自身があらかじめ決定するため、その経路が最短ではない。

2.2 TSPの解法

構築法の中には、ランダムサーチ、最近傍法、貪欲法、最遠挿入法があり、改善法の中には、2-opt法、山登り法、焼

きなまし法、タブー探索法、遺伝的アルゴリズムがある。本研究では、代表的な最近傍法、2-opt法、蟻コロニー最適化を挙げる。

1. 最近傍法

貪欲的に次々と近い点を訪れる方法。最近傍法のアルゴリズムを図1に示す。

```
任意に一点を選び、出発点とする；
while(すべての点を選ぶまで){
    選んだ点以外から、選んだ点に最も近い点を選ぶ；
    選んだ点を新たな出発点とする；
}
最後に選んだ点と最初に選んだ点を結び、巡回路を作る；
```

図1 最近傍法のアルゴリズム

(特徴)

- プログラミングが容易である
- 出発地点の取り方によって、構成される巡回路が異なることが多い
- 求めた巡回路との間にどのくらいの差があるのかを評価する効果的な方法がない

2. 2-opt法

解の一部について改善を行い、これを繰り返して全体の改善を行う方法。2-opt法のアルゴリズムを図2に示す。

```
巡回路を1つ作る；
while( となる枝がなくなるまで ) {
    枝(a, d)と枝(b, c)を枝(a, b)と枝(c, d)で
    つなぎ直す；
}
```

図2 2-opt法のアルゴリズム

(特徴)

- 近傍を大きくすれば、得られる解の精度は向上するが、近傍の探索に要する時間が長くなり、計算効率が下がる

3. 蟻コロニー最適化(ACO)

ACOは、都市間の距離の短さと都市間のフェロモン量の多さから次に訪れる都市を確率的に選択する手法で

ある[1]。まず、蟻はえさを探するためにランダムに動き回るが、えさを見つけて巣に戻るときはフェロモンを出す。フェロモンは限りがあり、経路が長くなるほどフェロモンが薄くなってしまふので、濃い経路を他の蟻も辿ることで、最適経路となる。

(特徴)

- グラフが動的に変化する場合に遺伝的アルゴリズムより有効
- 継続的に実行されるので、リアルタイムでの変化に適應することができる
- 大域探索に弱い

2.3 動的環境におけるTSPの解法

環境の変化があった場合、その新しい環境が前回の環境に近い場合、新しい解も前回の環境の解に近いことが予想できる。よって、新しい解は前回の解の近傍にあることになり、前回の解をもとにして探索を行うことで、効率の良い探索ができると考えられる。動的環境用プログラムでは前回の解を改善することで新しい解を求める。そのため、解法は改善法しか利用することができない。代表的な解法では、2-opt法、タブー探索法、焼きなまし法が改善法であるが、最近傍法、貪欲法、最遠挿入法は構築法であるため、利用できない。

環境の変化が小さい場合は局所改善型である2-opt法は動的環境用において探索時間が短縮する。ACOは、フェロモンというパラメータを用いて経路探索を行っているため、前回の解で用いたフェロモン量を利用することで、より効率の良い経路を探索すると思われる。よって本研究ではACOに着目する。

3 動的環境におけるTSPの導入

3.1 カーナビの新機能の提案

現在のカーナビには、動的環境における最短経路を探索する機能は備えているが、TSPのように複数の目的地について最適経路を計算する機能を備えていない。この場合、どの順番で目的地を巡回するかは利用者の直感によって判断されるため、遠回りをしてしまう場合や、混んでいる場所に先に向かってしまう場合などが考えられる。

本研究では、道路交通情報をリアルタイムに計算し、動的にTSPを解く機能を提案する。

カーナビにTSPを解く機能を導入することができれば、車で自社を出発し、営業で各支店を回る場合や時間指定のない荷物の配送で各地点を回る場合に、最適経路を導くことができ、時間やガソリンなどのコストを削減することが可能である。

3.2 蟻コロニー最適化(ACO)のフローチャート

ACOのフローチャートを図3に示す。ACO、2-opt法、最近傍法によって求められた距離と計算時間を表1に示す。ここで都市は、x座標、y座標共に0から10

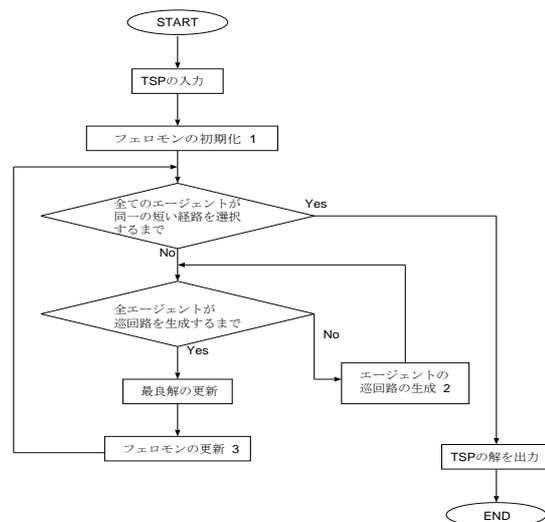


図3 ACOのフローチャート

までランダムに出力した値であり、時間の単位は秒である。

表1 各アルゴリズムの実行結果

	5都市	10都市	30都市	50都市
ACO (距離)	22.1	26.9	53.7	57.1
(時間)	0.05	0.21	1.88	5.32
2-opt (距離)	22.1	26.9	56.3	60.5
(時間)	0.00	0.01	0.00	0.00
最近傍法 (距離)	22.1	36.6	69.2	68.1
(時間)	0.00	0.00	0.00	0.00

表1より、解の精度はACO、2-opt法、最近傍法の順に悪くなり、計算時間は圧倒的にACOが長い。我々は、解の精度を重要視し、また動的環境に適應しているACOを用いてシミュレーションを行っていく。

4 セルオートマトン(CA)法によるTSPの実装

4.1 CA法を用いた交通流シミュレーション

動的環境下でTSPを適應させるために、交通流を再現する必要がある。本研究では、CA法とNSモデルを用いたシミュレーションを行う。CA法とは道路を多数のセルをつないだ帯として表し、道路の状態を各セルに車が存在するか否かで表現するものである。最も基本的なモデルとして、先のセルが空いていれば車を移動させ、空いていなければそのセルに留まるという、ウルフラムのルール184がある。図4にルール184の車の進行の様子を示す。

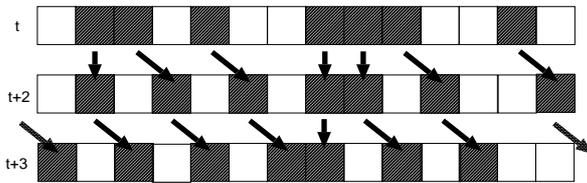


図4 ルール184の車の進行の様子

本研究ではそれに車の加減速を取り入れたNSモデルを使用する.NSモデルに対する各セルの更新は以下の4つの規則によって行われる.ただし車の速度は0から v_{max} の整数とする.なお本研究では, v_{max} を5とする.

加速:もし車両の速度 v が最大速度 v_{max} より小さかった場合,かつ今の速度 v で進む距離に1セル分を加えた距離に他の車両がいなかったら,速度を $(v+1)$ とする

減速:車両がセル i にいてその前の車両がセル $(i+j)$ にいる場合,もし $j \leq v$ ならセル i にいる車両の速度を $(j-1)$ とする

ランダムブレーキ:確率 p で車両の速度が0でないとき,速度を1小さくする

車両の運動:速度 v の車は,1ステップでその位置から進行方向に v 進める

4.2 シミュレーションの詳細

本研究では,メッシュ型の道路を用いる.そしてできるだけ実際の道路に近づくために,東西,南北にそれぞれ主要道を設け,一方通行を2箇所を作り,一部の道路を削除した.また,主要道とは車の流入率を他の道路に比べて多くし,交差点での右左折の割合を減らし,直進する割合を増やす.信号も主要道が青の時間を増やす.また,各交差点間の経路に距離を表す基本コスト1を設定し,車の渋滞状況に応じてコストを追加する.コストとは,各交差点間内の車が5以下の場合には0を,10以下の場合には1を,15以下の場合には2を,20以下の場合には3を,20より多ければ5をコストとして追加する.

図5に設定の詳細を示す.

4.3 ACOと2-opt法を用いた最適経路の探索

CA法により再現したネットワーク内に,複数の経由地点を設定する.それらから任意の2地点を選択し,その間の移動コストと経路をダイクストラ法により求める.全ての2地点の組合せについてこれを行い,データを配列に格納する.このコストの配列を用いて,TSPを解く.CAのステップが進むごとに,渋滞状況は変化し,コストが更新されるため,巡回経路は動的に求められる.効率の良い探索ができていないかを確かめるために,渋滞状況を考慮しない静的環境での経路探索も同時に行い,それぞれの距離とコストを比較する.

各交差点間:24セル

車の流入率(主):1/3

車の流入率(-):1/10

右左折率

(主→主):直進60%,左折20%,右折20%

(主→-):直進80%,左折10%,右折10%

(-→主):直進20%,左折40%,右折40%

(-→-):直進70%,左折15%,右折15%

信号制御(主と-):青信号の時間を一般道の2倍

信号制御(上記以外):青信号の時間を一般道と同じ

主:主要道 -:一般道

図5 設定の詳細

5 シミュレーション評価

5.1 ACOによる静的環境と動的環境の比較

3都市から20都市まで,渋滞を考慮しない静的環境における場合と,渋滞を考慮する動的環境における場合をそれぞれACOを用いて解き,結果を表2に示す.ここで集計した数値とは,セルオートマトンの開始から交通状況の安定する300ステップ目のものである.

表2 ACOによるシミュレーション結果

	静的環境		動的環境	
	距離	コスト	距離	コスト
3都市	15	19	16	16
4都市	15	20	17	17
5都市	17	26	17	19
6都市	19	28	19	20
7都市	20	27	21	22
8都市	22	32	23	25
9都市	23	34	25	26
10都市	23	36	26	30
11都市	25	34	27	29
12都市	26	36	29	31
13都市	25	38	29	32
14都市	28	41	31	37
15都市	29	41	32	39
16都市	31	43	33	39
17都市	32	44	34	40
18都市	35	46	36	43
19都市	37	49	39	47
20都市	38	48	41	46

表2より,距離は静的環境が短い,コストは高い.

これを動的に計算することで、渋滞を迂回するなど、より低コストの経路を導くことができる。本研究の場合、多くの経由地点を入力すると渋滞が発生しても迂回路を生成することが少なくなっていくが、それは本研究で用いたネットワークが経由地点の数に対して狭いためである。経由地点を設定する際、ある地点から次の地点へ訪れる経路として複数の候補が存在する程度に各地点の間隔を大きくとることができ、それを受容するだけの広さを持つネットワークならば、効率の良い巡回経路探索を行うことができると思われる。

5.2 ACOと2-opt法の比較

表2から10都市までが妥当な数値を出力すると考えられ、3都市から10都市まで、それぞれ10回ずつシミュレーションを行い、どちらがより低コストな経路を導くことができるか集計を行う。その結果を表3に示す。表3は、10回のコストの平均値とコストの最大値、最小値を示し、計算時間の単位は、 μ 秒である。

表3 ACOと2-opt法のシミュレーション結果

	ACO			
	平均値	計算時間	最大値	最小値
3都市	12.8	11000	17	10
4都市	13.4	19000	16	10
5都市	16.8	29000	20	15
6都市	20.1	41000	24	18
7都市	22.2	56000	25	19
8都市	23.2	72000	26	21
9都市	25.4	90000	30	23
10都市	28.9	110000	40	24
	2-opt法			
	平均値	計算時間	最大値	最小値
3都市	12.7	33.5	17	10
4都市	13.4	31.5	16	10
5都市	18.9	33.8	20	18
6都市	21.5	36.5	24	17
7都市	25.6	38.6	28	24
8都市	28.8	43.3	34	23
9都市	32.4	44.6	34	28
10都市	36.7	39.2	43	34

図6は、5都市におけるACOと2-optの比較を行った際のシミュレーション結果を示す。

この結果より、計算時間では圧倒的に2-opt法が短いですが、精度は2-opt法よりACOがより短いコストで巡回することがわかる。また、都市数を多くする程、精度の差は大きくなる。我々は、計算時間を短縮させることよりも、解の精度を向上させることで、より効率の良い経路を導き、その結果、移動時間を短縮させることができ、より大きなメリットがあると判断した。

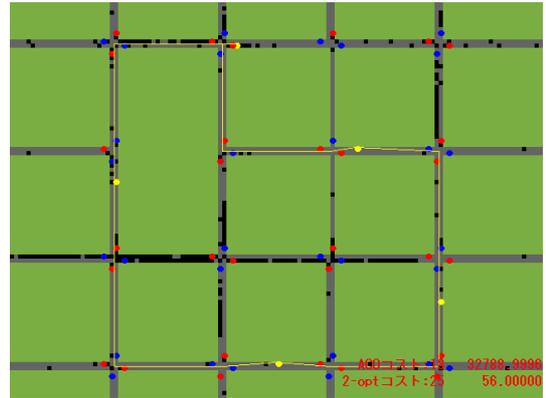


図6 ACOと2-opt法の比較

6 まとめ

本研究では、複数の経由地点を巡回し、動的環境下での最適経路を求めため、TSPに対する様々なアルゴリズムを研究し、それらの性能や特徴について比較した。我々はACOに着目し、CA法によるシミュレーションを行った。そして、動的環境下で効率の良い巡回経路を求められているかを確認するために、渋滞状況を考慮しない場合との比較を行った。その結果、動的環境下では経路を迂回することで距離は長くなるが、コストはより小さくなるのが分かった。次に我々が採用したACOについて、ACOが本研究に適しているかを確認するために、精度と計算時間について2-opt法と比較した。その結果、2-opt法に比べ計算時間では圧倒的に長いものの、解の精度はより良い巡回路を導くことができる。また、表1と表3から、動的環境で蟻コロニー最適化を用いることにより、より効率の良い巡回路を生成すると判断できる。

我々は、今後の課題として、地図を読み込み、地図上でTSPのシミュレーションが可能になれば、もっと身近に利用できるだろう。

参考文献

- [1] 亀田陽介, “局所最適解をフェロモンの初期配置に利用したアントコロニー最適化法によるTSPの解法”, 修士論文, 筑波大学大学院システム情報工学研究科, 2007.
- [2] 狩野均, 小塚英城, “CA法による広域道路交通シミュレータを用いた経路案内方式の評価”, 情報処理学会, 高度交通システム研究会ITS-10-6, pp.37-43, 2002.9.
- [3] 加藤恭義, 光成友孝, 築山洋, “セルオートマトン法-複雑系の自己組織化と超並列処理-”, 森北出版, 1998.
- [4] 久保幹雄, “巡回セールスマン問題への招待”, 朝倉書店, 1997.