Vol.2011-MPS-83 No.10
2011/5/17

粘菌ネットワークを用いた巡回セールスパーソン問題の解法

榎原博之[†1]　田中裕也[†1]　石川琢士[†1]

自然界には粘菌という単細胞生物が存在する．粘菌は森の土の中などに数多く存在しており，採餌行動の際には自分自身の体で栄養を運ぶネットワークを構築する．本研究では，粘菌の採餌行動のシミュレーションプログラムを元に，巡回セールスパーソン問題 (TSP) を解く粘菌アルゴリズムを提案する．まず，粘菌の作るネットワークをシミュレーションし，そこから巡回路を作成し，TSP に対応させる．最終的には 2-*opt* 法による局所探索法を用いて，解の改善を行う．ベンチマーク問題 (TSPLIB) を用いて計算機実験を行い，評価を行う．

A Slime Mold Algorithm for Solving the Traveling Salesperson Problem

Hiroyuki Ebara,[†1] Yuya Tanaka[†1]
and Takuji Ishikawa [†1]

The slime mold is single-celled which exists in the natural world. It exists a lot in the soil of the forest etc. and in the foraging action it constructs the network in which nourishment is carried by own body. In this research, we propose a algorithm which solves the travelling salesperson problem (TSP) based on the simulation program of the foraging action of the slime mold. First,we simulate a network made by the slime mold, and make a TSP tour. Finally, in order to improve the solution, we use 2-*opt* which is the local search method. We experiment on the computer by using the benchmark problems(TSPLIB), and we evaluate the slime mold algorithm.

†1 関西大学大学院 理工学研究科
Graduate School for Science and Engineering, Kansai University

1. はじめに

近年，計算機の能力が向上するにつれ，要求される計算量も増加してきている．このような膨大な計算を計算機を使って行う際，組合せ最適化を用いることがある．組合せ最適化問題とは，与えられた制約条件を基に，目的関数の値を評価し，最小もしくは最大になる組合せを求める問題である．

組合せ最適化問題において，求めたい結果が厳密な最適値でなければならない場合も存在するが，一般的にはある程度の精度を持つ近似解でも許容できる場合が多く，短時間で解を得ることができる．組合せ最適化問題には問題の規模に対して指数関数的に実行時間がかかることが知られている問題が多く，長時間をかけて最適な値を出すよりも短時間で充分最適解に近い近似解を得ることが重要あり，それに対する解法も数多く研究されている[1)]．数ある近似解法の中でも，特に，メタヒューリスティック手法は多くの問題に適応できる汎用的な解法であることから幅広く研究されている[2)]．

自然界には粘菌という単細胞生物が存在する．粘菌は森の土の中などに数多く存在しており，採餌行動の際には自分自身の体で栄養を運ぶネットワークを構築する．この粘菌ネットワークをコンピュータシミュレーションにより，構築する研究がある[3)]．

本研究では，粘菌の採餌行動のシミュレーションプログラムを元に，巡回セールスパーソン問題 (TSP) を解く粘菌アルゴリズムを提案する．まず，粘菌の作るネットワークをシミュレーションし，そこから巡回路を作成することで TSP に適応させる．最終的には局所探索法である 2-*opt* 法を用いて，解の改善を行う．実際にコンピュータ上でベンチマーク問題 (TSPLIB) を用いて計算機実験を行うことで，評価を行う．

2. 粘菌ネットワーク

人やエネルギーを効率的に輸送するネットワークは，インフラの中でも重要な部分を占めている．輸送ネットワークでは，低コストで効率的に輸送を行うことが優先されるが，障害に強いシステムを作るためには，コスト効率が悪い経路を追加する場合がある．そのため，近年のネットワークでは高い耐障害性が求められるようになってきている．

粘菌は，採餌行動の際，ネットワークを形成する．形成されたネットワークは，厳しい環境に適応するために，耐障害性とコストとのバランスをとる必要がある．そのため，コスト，効率性，耐障害性が適切なバランスをとったネットワークを形成すると考えられている[3)]．

2.1 粘菌のシミュレーション

文献[3] では，粘菌のネットワークを都市間における適正なネットワークの理論に応用できると考え，シミュレーションを行っている．シミュレーションでは，粘菌ネットワークを電気回路網に置き換えている．そのアルゴリズムは以下のようになる．

(1) 粘菌の初期状態としてメッシュを作成する
作成したメッシュが回路網と対応する．メッシュの交点間をつなぐのが粘菌，交点のうちいくつかの点に粘菌の餌を設置する．電気回路に置き換える際には，粘菌が抵抗 (アドミッタンス)，餌を設置した点が電源となる．

(2) 各点に流れ込む電流量を設定する
各点に設置する餌の量を決定する．このとき設置する餌の量が多ければ多くの電流が流れる．流入する電流量 ($\sum_{i\in A}\sum_j Q_{ij}$) と流出する電流量 ($\sum_{i\in B}\sum_j Q_{ij}$) の総和 ($\sum_{i\in A\cup B}\sum_j Q_{ij}$) は 0 となるようにする．ここで，A は流入点の集合，B は流出点の集合である．

(3) キルヒホッフの法則により各点の電圧値を求める
回路網の各抵抗および，電源からの電流量から各点の電圧値が計算できる．
キルヒホッフの第一法則により，メッシュの任意の交点に流れ込む電流の総量は 0 になる (式 (1))．

$$\sum_{i=1}^{N} I_i = 0 \tag{1}$$

各点に対してこの式をたて，これらの n 元連立一次方程式を解くことにより，各点の電圧値を得ることができる．

(4) 各点間の電流値を求める
各点の電圧値から各点間の電流値を計算する．ij 間の電流値 Q_{ij} は粘菌管内の流量がハーゲン・ポアズイユ流れに近似することから，ハーゲン・ポアズイユの式で求めることができる：

$$Q_{ij} = \frac{\pi r^4 (p_i - p_j)}{8\eta L_{ij}} = \frac{D_{ij}(p_i - p_j)}{L_{ij}} \tag{2}$$

求めた各点間の電流値 Q_{ij} は実際の粘菌での餌の流量に対応する．
D_{ij}/L_{ij} がアドミッタンス，p_i が各点の電圧値である．D_{ij} が各点間の粘菌の量，L_{ij} が各点間の距離を表している．

(5) 抵抗値の更新
各点間の電流値に応じて各点間の抵抗値を更新する．次の抵抗値は以下の式によって決定される：

$$\frac{dD_{ij}}{dt} = f(|Q_{ij}|) - D_{ij} \tag{3}$$

$$f(Q_{ij}) = \frac{|Q_{ij}|^\gamma}{1 + |Q_{ij}|^\gamma} \tag{4}$$

γ は非線形フィードバックの制御パラメータで 0 よりも大きい値をとる．

3. 巡回セールスパーソン問題

巡回セールスパーソン問題 (TSP) とは，訪問対象となる n 個の都市を一度ずつ訪問して出発した都市に戻る巡回路の中で距離が最小のものを求める問題である[4]．

n 個の都市の集合 $V = \{1, ..., n\}$ と都市 i と都市 j の間の距離を C_{ij} とすると，目的関数 $f(x)$ を最小化する式は以下のように数式化できる：

$$f(x) = \sum_{k=1}^{n-1} C_{x(k)\ x(k+1)} + C_{x(n)\ x(1)} \tag{5}$$

ここで，$x(k) = i$ は，k 番目に訪れる都市が i であることを表す．

4. 粘菌アルゴリズム

4.1 アルゴリズムの概略

本実験では，粘菌のアルゴリズムを TSP に適用するために，アルゴリズムに以下の変更を加える．

(1) メッシュの作成
2.1 節の (1) が対応する．メッシュの作成には三角形分割を用いる．

(2) 抵抗値の初期化
それまでに得た解を元に抵抗値の初期化を行う．

(3) ネットワークの構築
2.1 節の (2)~(5) が対応する．粘菌のシミュレーションではネットワークの構築は一度しか行わないが，本アルゴリズムではネットワークの構築を繰り返し行う．
構築に使う式や各パラメータの値などは，一部を除いて文献[3] のシミュレーションと

Vol.2011-MPS-83 No.10
2011/5/17

同じものを使っている．

(4) 巡回路作成

ネットワークを元に巡回路を作成する．

(5) (2)〜(4) を指定回数繰り返す

(6) 局所探索法による解の改善

指定回数 2-*opt* 法による局所探索を行い，解を改善させる．

本アルゴリズムは，繰り返し回数によって異なる処理を行う．各項目で行う処理については以下の各節で詳しく述べる．

4.2 メッシュの作成

粘菌ネットワークの初期状態として三角形分割を用いる．ドロネー三角形分割は TSP の巡回路にふさわしい枝を含む場合が多いことを実験的に示されている[5)6)]．このことを根拠とし，三角形分割の手法にはドロネー三角形分割を用いる．

4.3 抵抗値の初期化

粘菌ネットワークを構築するために各枝の太さ D_{ij} を設定する．D_{ij} の値が大きいほど抵抗値が低く，電流が流れやすい枝であることを表している．

D_{ij} の初期化は以下のように行う．

(1) 奇数回目の初期化

最初に初期化を行う際は基準とする解がないため，全ての枝に対して太さとして 1 を与える．

以降の初期化では，その時点での最良の解である暫定解 T_1(図 1) と，前回の探索で見つかった解 T_2(図 2) に共通する枝 (i,j) は，解構築において有効な枝であると考え，通常よりも大きな値 $d(d>1)$ を与える．

$$D_{ij} = \begin{cases} d & \text{if}(i,j) \in T_1 \cap T_2 \\ 1 & \text{otherwise} \end{cases}$$

図 3 の細い線は通常の太さの枝，太い線は通常よりも太い枝を表している．

(2) 偶数回目の初期化

偶数回目の構築では，前回の探索で見つかった解を元に初期化を行う．前回の探索で見つかった解 T_2 に含まれる枝 (i,j) には，通常よりも大きな値 $d(d>1)$ を与える．

図 5 の細い線は通常の太さの枝，太い線は通常よりも太い枝を表している．

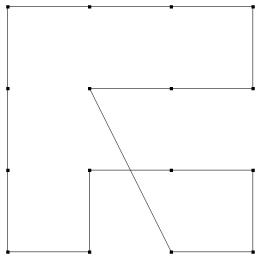

図 1 暫定解 T_1

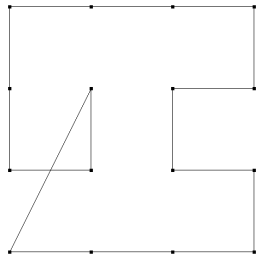

図 2 前回の探索で見つかった解 T_2

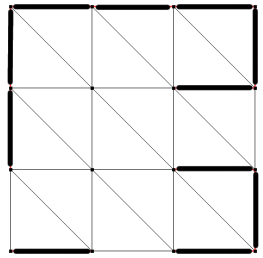

図 3 初期化後

$$D_{ij} = \begin{cases} d & \text{if}(i,j) \in T_2 \\ 1 & \text{otherwise} \end{cases}$$

4.4 ネットワークの構築

三角形分割を行ったグラフ G および，初期化によって決定された各枝の抵抗値 D_{ij} を元に，粘菌のシミュレーションと同様に粘菌ネットワークを構築する．この際，構築回数に応じて 2 種類の方法で電源接続ノードを決定する．

(1) 奇数回目の構築

奇数回目の構築では，ノードをランダムに選択する．ノード間距離が一定以上あいて

Vol.2011-MPS-83 No.10
2011/5/17

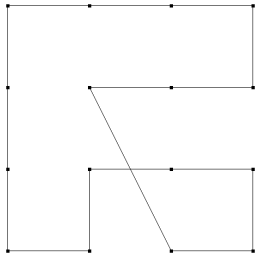

図 4 前回の探索で見つかった解 T_2

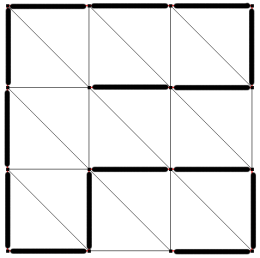

図 5 初期化後

おり，枝により直接接続されていないノードを 4 点選択する．流入点と流出点はともに 2 点ずつとし，選択した 4 点に対してランダムに割り振る．

(2) 偶数回目の構築

偶数回目の構築では，前回得た解の改善を行う．解に含まれる枝から，グラフ G に含まれていない枝を取り除いたグラフ G' を作成する．グラフ G' において，接続されている枝が 1 本以下のノードの集合 V' を求める．ネットワーク構築の際には，この集合 V' に含まれるノードにのみ電源を接続することにより，前回接続されていなかったノード同士が強固に接続されたネットワークを構築する．

集合 V' に含まれるノード数に応じて以下のように電流を流す．

(a) $|V'| \geq 5$

i および i と枝で接続された集合 V' に含まれる点 i' を流入点とする．i および i' と枝により接続されていない j および j' を流出点として抵抗値の更新を行う．

(b) $|V'| = 3$ *or* 4

互いが枝により接続されておらず，集合 V' に含まれる点 i, j を選択する．i を流入点，j を流出点として抵抗値の更新を行う．

(c) $|V'| \leq 2$

奇数回目の構築と同様の基準でノードを選択する．

4.5 巡回路作成

構築されたネットワーク N の抵抗値を参考に巡回路を作成する．ネットワーク N に対して以下の 3 つの処理を行うことで巡回路を得る．

(1) D_{ij} の小さな枝を除去

ネットワーク N で使用されている枝 (i, j) を D_{ij} の昇順に並べた集合 E_1 を求める．E_1 を D_{ij} の小さいものから順に枝 (i, j) を除去する．このとき各ノードに接続されている枝が 1 本以下にならないようにする．E_1 に除去すべき枝 (i, j) が存在しなくなれば処理は終了する．

処理終了段階で巡回路ができていれば解として出力し，以降の処理は行わない．

図 6, 図 7 の細い線は D_{ij} の小さい枝，太い線は D_{ij} の大きい枝を表している．

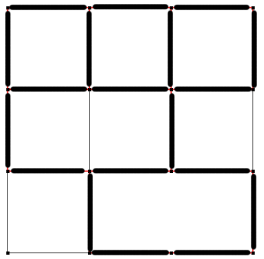

図 6 ネットワーク N

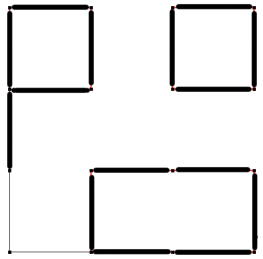

図 7 ネットワーク N'

(2) D_{ij} の大きな枝を接続

枝を除去したネットワーク N' で使用されている枝 (i, j) を D_{ij} の降順に並べた集合 E_2 を求める．E_2 を D_{ij} の大きなものから順に枝 (i, j) を接続していく．このとき枝 (i, j) を接続することで閉路ができた場合，接続は行わない．

(3) 残りのノードを接続

ここまでの処理を行った結果，接続された枝が 1 本以下のノードが存在する分断されたネットワーク N'' ができる．接続された枝が 1 本以下のノード間をつなぐ，三角形分割に含まれない枝を含めたすべての枝を距離の昇順に並べた集合 E_3 を求める．E_3 に含まれる枝 (i, j) を距離の短い順に閉路をつくらないように接続していく．E_3 の全ての枝をチェックし終わると，全てのノードを一度だけ通る経路ができる．最後に経路の両端をつなぐことで巡回路を得ることができる．

図 9 の細い線はネットワーク N''，太い線はグリーディに接続した枝を表している．

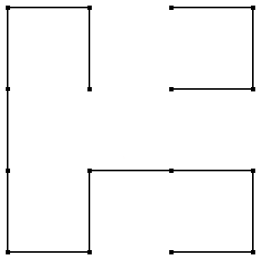

図 8 ネットワーク N''

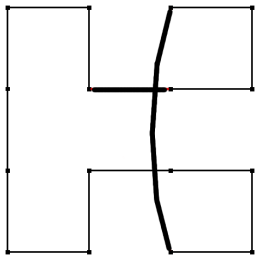

図 9 巡回路

4.6 2-*opt* 法による解の改善

巡回路作成の際，残っているノードをグリーディに接続したことで，交差した非常に長い枝が存在している．この枝による解の悪化を補正するため，一定回数 2-*opt* 法による局所探索を行い，解の改善を行う．

5. 実験結果

5.1 実験環境

本実験で使用した計算機のスペックを表 1 に記す．

5.2 TSPLIB

実験対象の問題は TSPLIB[7] より入手する．
使用した問題とその最適解の一覧を表 2 に示す．

表 1 計算機のスペック

OS	CPU	メモリ
Ubuntu 10.04	Intel Core i7 860 2.80GHz × 8	8GB

5.3 実　　験

提案したアルゴリズムを用いて探索を行う．
TSP に適応するにあたり，距離の重みを増すため式 (2) を以下のように変更した：

表 2 TSPLIB の名前と最適解

問題名	都市数	最適解
eil51	51	426
eil76	76	538
lin105	105	14379

$$Q_{ij} = \frac{\pi r^4 (p_i - p_j)}{8\eta L_{ij}^{10}} = \frac{D_{ij}(p_i - p_j)}{L_{ij}^{10}} \tag{6}$$

予備実験を行った結果，距離を 10 乗の重みを加えた．

また，粘菌のシミュレーション[3] と同様に，式 (4) の γ は 1.8，電源から流れ込む電流の総和は 2 とした．抵抗値の更新を行う回数は一度の解生成につき，都市数 $\times 10$ 回とし，解生成は 30 回行った．2-*opt* 法による解の改善は都市数の 2 乗回繰り返して行った．

図 10, 図 11, 図 12 は改善を行っている途中の解を，図 13 は最終的な解を表している．図 11 では図 10 で断線していた箇所を接続しやすくなるような処理を行っているので，断線箇所同士をつなぐ，より良い形路のいくつかが粘菌により生成されていることがわかる．図 12 では暫定解と前回の解から改善を行っているが，直前の図 11 で暫定解が更新されたため，ほとんどの枝が図 11 で使われている枝に固定されてしまっていることがわかる．図 13 では 2-*opt* による局所探索が行われたことで，交差した太い枝がなくなっていることがわかる．

TSPLIB から「eil51」，「eil76」，「lin105」の 3 例を選択し，各 10 回ずつ探索を行った．

各都市数における探索結果を表 3~6 に記す．表 3~5 は，各々の結果から総距離の平均値と最小値，表 2 にある最適解との誤差を計算したものであり，表 6 は計算時間の平均値と最小値をまとめたものである．表 3~5 より，すべての都市数において，ある程度最適解に近い結果がでていることが分かる．また，表 6 より，都市数の増加に従い計算時間が大きく増加していることがわかる．

表 3 総距離と誤差 (eil51)

eil51	総距離	最適解との誤差 (%)
平均	436.21	2.4
最小	429.93	0.92

表 4 総距離と誤差 (eil76)

eil76	総距離	最適解との誤差 (%)
平均	555.94	3.33
最小	548.71	1.99

表 5 総距離と誤差 (lin105)

lin105	総距離	最適解との誤差 (%)
平均	14777.04	2.77
最小	14456.17	0.54

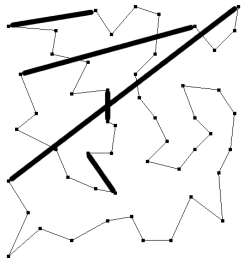

図 10　1 回目の解

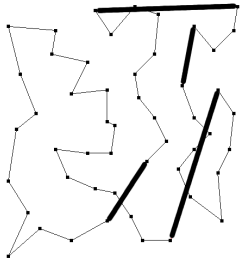

図 11　2 回目の解

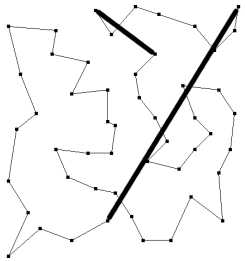

図 12　3 回目の解

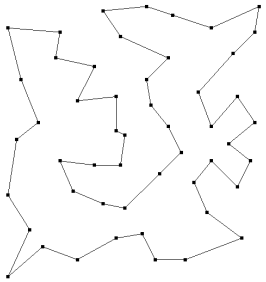

図 13　最終的な解

表 6　各問題の計算時間

問題名	平均時間 (sec)	最小時間 (sec)
eil51	51.42	51.18
eil76	248.4	246.87
lin105	881.69	876.19

5.4 考　　察

結果より，粘菌ネットワークのシミュレーションによって得られたネットワークから，巡回路を作成することで，巡回セールスパーソン問題が解くことができ，ある程度良い結果が得られることがわかった．しかし，これには 2-*opt* 法による局所探索の恩恵も大きいと考えられる．また，都市数の増加とともに計算時間が大幅に増加するため，計算時間を減らす工夫を行い，問題数の大きな問題に対応することが必要である．

6. おわりに

本研究では，粘菌シミュレーションに変更を加え，TSP の解に適応可能なネットワークを構築させることによって TSP の解を求めた．枝を固定してネットワークを再構築させることで，より良い TSP の解を求めるためには有効であるが，枝が交差せざるを得ない状況に陥る可能性があることがわかった．そこで，交差した枝を 2-*opt* 法によって繋ぎ直すことで，ある程度良い結果を出すことができた．

今後の課題として，解精度の向上と計算時間の短縮が挙げられる．本研究では分断されたネットワークを接続する際に短い枝からグリーディに接続を行っている．そのため，最初の枝を接続する時点では最良であったが，全体的に見るとあまり良い解ではないといったことがある．この接続法を変更することでより良い解を得やすくなると考えられる．また，計算時間の大半はネットワーク構築時の各ノード電圧の計算なので，計算方法をより高速な方法に変えることで計算時間を大幅に短縮することができると考えられる．

参　考　文　献

1) S.Sait and H.Youssef: 組合せ最適化アルゴリズムの最新手法，丸善 (2002).
2) C.R.Reeves: モダンヒューリスティックス 組合せ最適化の先端手法，日刊工業新聞社 (1997).
3) Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D., Fricker, M., Yumiki, K., Kobayashi, R. and Nakagaki, T.: Rules for biologically inspired adaptive network design, *Science*, Vol.327, No.5964, p.439 (2010).
4) 山本芳嗣，久保幹雄：巡回セールスマン問題への招待，朝倉書店 (1997).
5) Letchford, A. and Pearson, N.: Good triangulations yield good tours, *Computers & Operations Research*, Vol.35, No.2, pp.638–647 (2008).
6) 玉木久夫，土屋裕希：平面ユークリッド TSP の分割統治法ヒューリスティツクス，情報処理学会研究報告. AL, アルゴリズム研究会報告， Vol.2007, No.23, pp.121–128 (2007-03-09).
7) TSPLIB: . http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.