適応システム特論 前半レポート課題 2119012 鎌田幸希

1　実装したモデル

講義で扱ったモデルとして，今回 Ant Colony Optimization を実装した．講義の動画以外にも，https://qiita.com/ganariya/items/25824f1502478a673005 を参考に今回実装を行った．

2　実験

実験には，各都道府県の県庁所在都市の緯度経度を x 座標 y 座標としたときに，すべてのノードを回る距離を最短化するための最適化を行った．パラメータごとの結果への影響をまとめるため，以下の項目を変化させ実験を行った．結果は，サイクルごとのパスの距離の変化をプロットしたものを図にした．

- サイクル数
- アリの数
- フェロモンの蒸発率
- 開始地点・終端
- 移動時の確率を決定するための alpha と beta

基本的なパラメータは，表 1 に示す．ant_prob_random は，アリが過学習をすることを防ぐため，次のノードを決める際に完全ランダムにする確率である．

表 1　シミュレーションパラメータ

パラメータ	設定値
サイクル数	100
アリの数	100
フェロモンの蒸発率	0.9
alpha	5
beta	3
フェロモン決定の分母 Q	100
ant_prob_random	0.1

サイクル数

N 体のアリを始点 S から移動させ，パスを構築し，フェロモンの更新を行うまでを一つのサイクルとして扱い．これを 1，10, 100, 1000 回と変化させて実験を行った．

アリの数

1 度にネットワーク上を移動させるアリの数を 1,10,100,1000 体と変化させて実験を行った．

フェロモンの蒸発率

フェロモンを更新する際の，前のサイクルまでのフェロモンを残す割合を 0.1,0.5,0.9,1.0 と変化させて実験を行った．

開始地点・終端

開始地点・終端となるノードを，札幌，東京，大阪，那覇と変化させて実験を行った．

移動時の確率を決定するための alpha と beta

アリが次のノードを決定する際に，フェロモンとノード間のヒューリスティックどちらを重視するかという値である alpha と beta の値を，(0,1),(1,0),(1,1),(3,5),(5,3) と変化させて実験を行った．

3　結果・考察

3.1　サイクル数

図 1 から図 3 が，総サイクル数を変えたときのサイクルごとのパスの距離の変化を図示したものである．確率的な偏りがある可能性があるが，それぞれの最短距離は，63,62,61 と変化しており，サイクル数を増やしたときのほうが，より短いパスを算出できている．

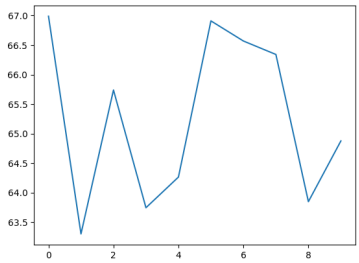

図 1　サイクル 10 回時の最短距離の変化

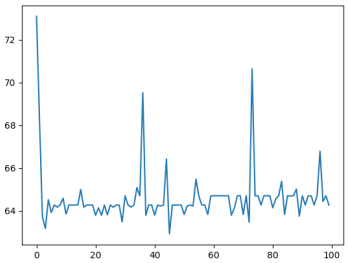

図 2　サイクル 100 回時の最短距離の変化

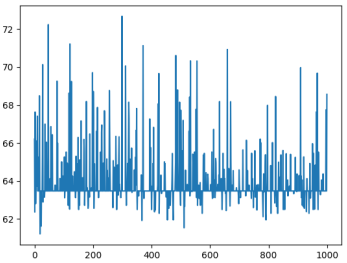

図 3　サイクル 1000 回時の最短距離の変化

3.2　アリの数

図 4 から図 7 がアリの数を 1 から 1000 まで変化させたときの，サイクルごとのパスの距離の変化を図示したものである．それぞれ最短距離が 65,63,62,61 と変化しており，より多いアリの数で

より短いパスが求められることがわかる．また，多いアリで実行したときのほうが，結果が上振れすることが少なくなっていることもわかる．

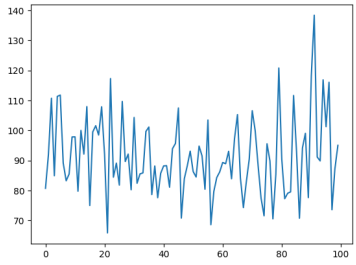

図 4　アリ 1 体の最短距離の変化

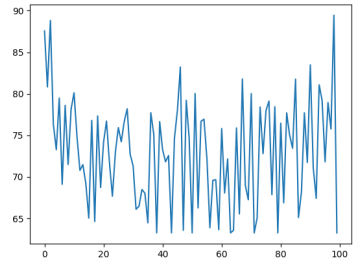

図 5　アリ 10 体の最短距離の変化

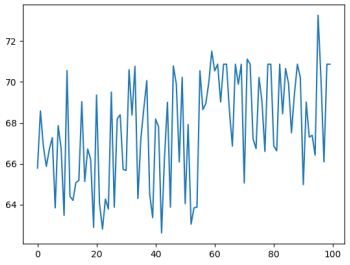

図 6　アリ 100 体の最短距離の変化

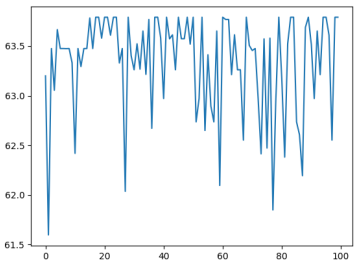

図 7　アリ 1000 体の最短距離の変化

3.3　フェロモンの蒸発率

図 8 から図 11 がフェロモンの蒸発率を 0.1 から 1.0 まで変化させたときのパスの距離の変化を図示したものである．それぞれ最短距離は，62,62,62,61 となっており，かなり誤差ではあるものの，徐々に最適化されていることがわかる．平均的に見ると，0.9,1.0 の場合のときが，0.1,0.5 のときと比べると上に大きくなく，より良いのではないかと考えられる．しかし，1.0 だと，前回までのフェロモン値をすべて残すことになり，大きな問題で実行するときにフェロモン値がオーバーフローすることが予想されるため，0.9 あたりが最適だと考えられる．

3.4　開始地点・終端

図 12 から図 15 は，それぞれ開始・終端を札幌，東京，大阪，那覇と変えて実行した結果である．それぞれ最短距離は，62,61,61,62 となっている．グラフを見ると札幌，那覇は，サイクルごとの出力パスの距離の変動が大きいことや，最短距離の僅かな違いを見ると，よりグラフの中央となるような位置から始めると最適解が導きだされやすいと考えることもできる．これに関しては，より細かな検証が必要である．

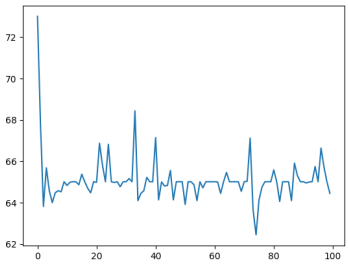

図 8　蒸発率 0.1 の最短距離の変化

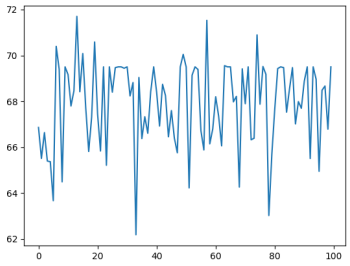

図 9　蒸発率 0.5 の最短距離の変化

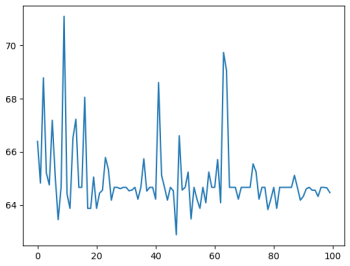

図 10　蒸発率 0.9 の最短距離の変化

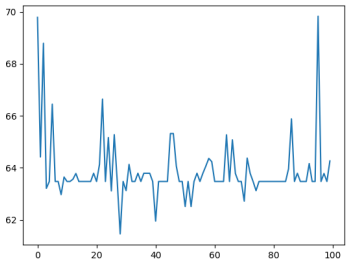

図 11　蒸発率 1.0 の最短距離の変化

3.5　移動時の確率を決定するための alpha と beta

図 16 から図 20 までは，alpha と beta の値を (0,1),(1,0),(1,1),(3,5),(5,3) と変化させたときの，出力パスのサイクルごとの変化の様子をプロットしたものである．それぞれ，最短距離は，122,125,68,62,61 となっている．この結果からどちらかの値を 0 にしてしまうと，全く最適化ができないことがわかる．

4　付録 (プログラムなど)

実験結果の図．出力されたパスを図にしたもの，プログラムを zip の中に同封しました．実行環境 Python 3.8.3

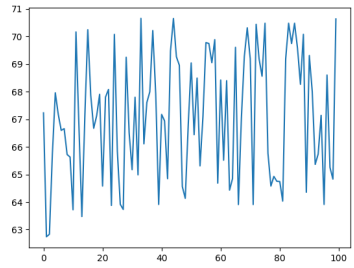

図 12　札幌からの最短距離の変化

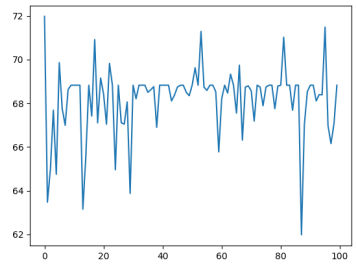

図 13　東京からの最短距離の変化

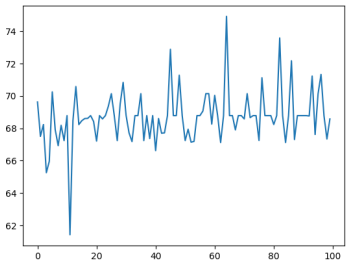

図 14　大阪からの最短距離の変化

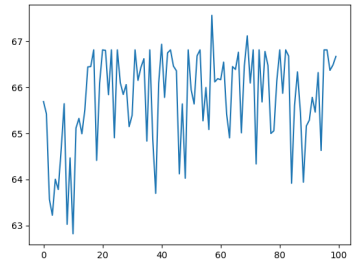

図 15　沖縄からの最短距離の変化

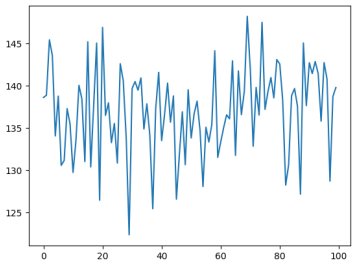

図 16　(0,1) の最短距離の変化

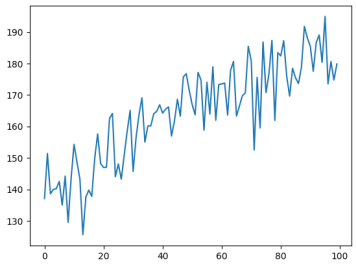

図 17　(1,0) の最短距離の変化

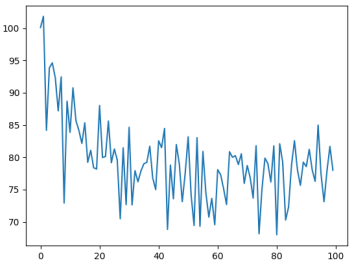

図 18　(1,1) の最短距離の変化

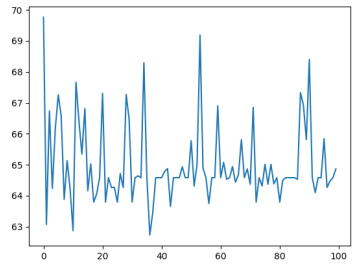

図 19　(3,5) の最短距離の変化

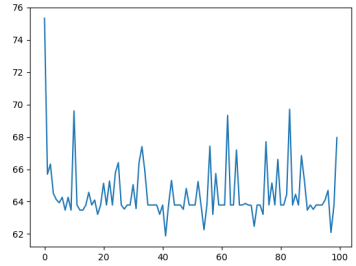

図 20　(5,3) の最短距離の変化