

## 1. プログラムの概要

- 生体シミュレーションを行うプログラム
- 蟻コロニー最適化の様子を見ることができる。

### ※蟻コロニー最適化

実世界では、アリは始めランダムにうろつき、食物を見つけるとフェロモンの跡を付けながらコロニーへ戻る。他のアリがその経路を見つけると、アリはランダムな彷徨を止めてその跡を辿り始め、食物を見つけると経路を補強しながら戻る。

しかし、時間とともにフェロモン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フェロモンは蒸発しやすい。それに対して、経路が短ければ行進にも時間がかからず、フェロモンが蒸発するよりも早く補強されるため、フェロモン濃度は高いまま保たれる。

従って、あるアリがコロニーから食料源までの良い(すなわち短い)経路を見つけると、他のアリもその経路を辿る可能性が高くなり、正のフィードバック効果によって結局すべてのアリが1つの経路を辿ることになる。蟻コロニー最適化アルゴリズムの考え方は、解決すべき問題を表しているグラフを歩き回る「シミュレーションされたアリ」によってこの行動を真似ることである。

蟻コロニー最適化アルゴリズムは、巡回セールスマン問題に近似最適解を生み出すために用いられた。この手法はグラフが動的に変化する場合に焼きなまし法や遺伝的アルゴリズムよりも有効である。蟻コロニー最適化アルゴリズムは継続的に実行されるので、リアルタイムで変化に適応することができる。このことから、ネットワークのルーティングや都市交通システムでの応用が考えられる。

(Wikipedia より)

### <登場するもの>

- ・ 蟻
- ・ 巣
- ・ エサ
- ・ 障害物

### <機能>

- ・ 蟻は障害物をよけながらエサを探し、巣まで運ぶ動作を繰り返す。
- ・ 蟻は自分がたどってきた道にフェロモンを残す。
- ・ 蟻は他の蟻（自分も含む）が残したフェロモンに引き寄せられ、同じ道を辿ろうとする。
- ・ 残されたフェロモンは時間が経つと蒸発していく。
- ・ 蟻の出すフェロモンはエサや巣から離れて時間が経つほど弱くなっていく。

## 2. クラス図



### 3. 各クラスの概要

- **AntsApplication**

アプリケーションとして Ants を表示する。

- **AntsApplet**

アプレットとして Ants を表示する。

- **Ants (プログラムの本体)**

蟻のコロニー（グリッド）全体を制御・表示し、シミュレーションを更新する。  
各セル・蟻の状態を管理を管理して表示に反映させる。

- **AntsControlPanel**

各パラメータを制御する。

- **AdvancedControlPanel**

高度なパラメータを制御する。

- **Ant**

各蟻の動きを制御する。

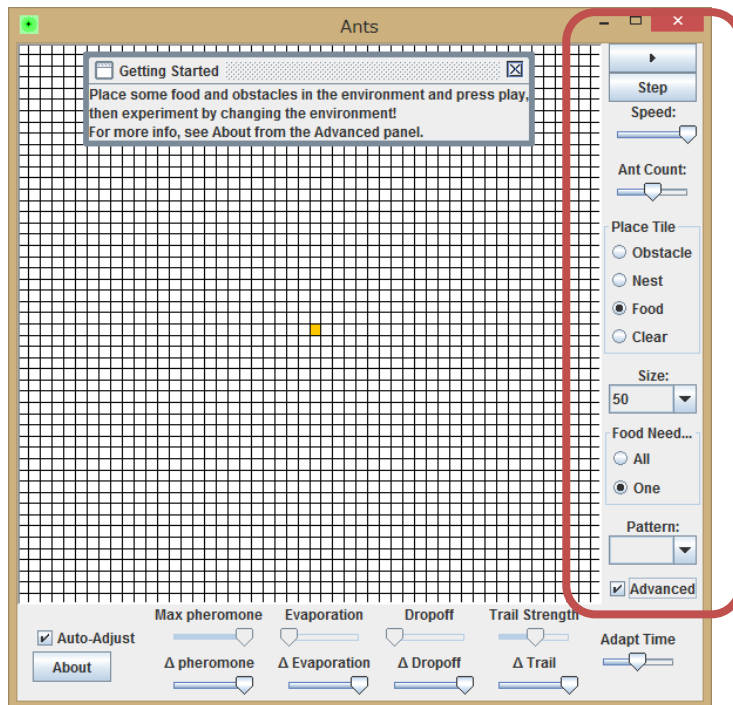
- **Cell**

各セルの状態を制御する。

#### 4. クラスごとの解説

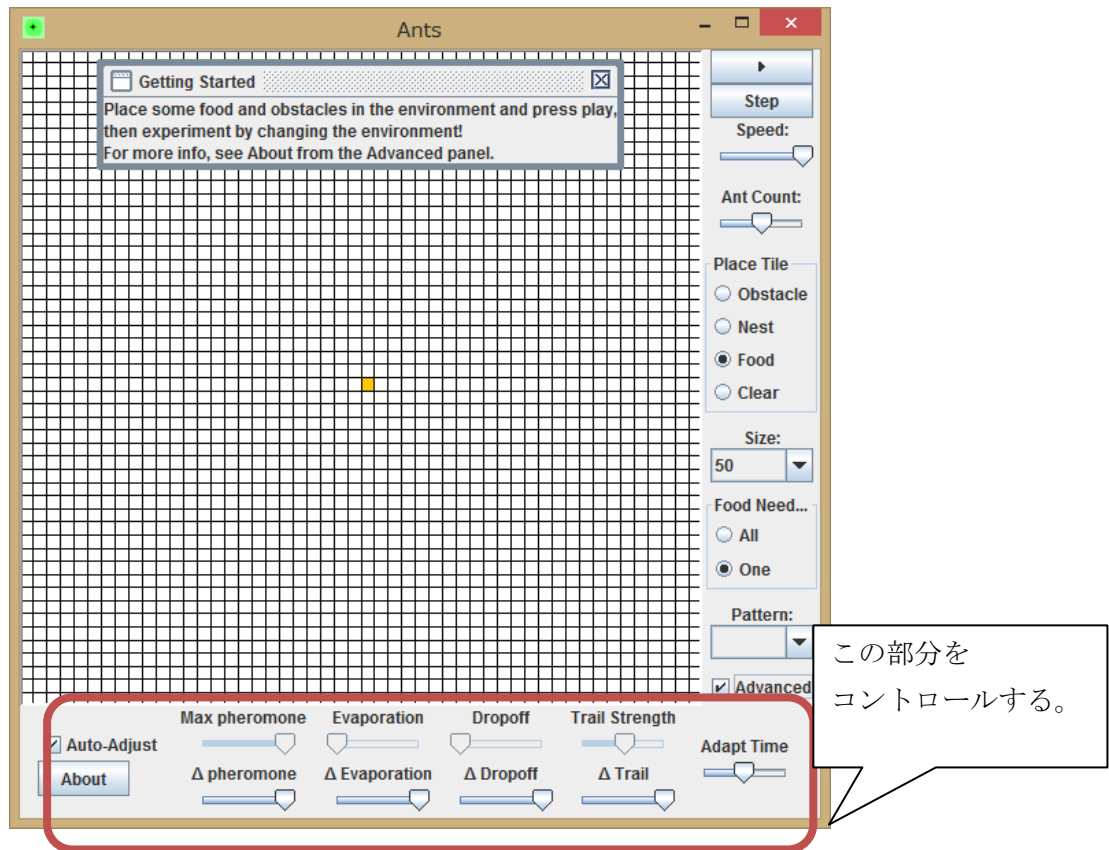
- AntsControlPanel

この部分を  
コントロールする。



- Pause / ▷ : シミュレーションを再開・停止する。
- Step : シミュレーションをワンステップずつ表示させる。
- Speed : シミュレーションのスピードを制御。
- AntCount : 蟻の数を制御。
- PlaceTile : マウスで指定した場所のセルを以下の性質に変える。
  - Obstacle : 障害物
  - Nest : 巣
  - Food : エサ
  - Clear : 何もない状態に戻す
- Size : グリッドの大きさ（セルの数）を変える。
- FoodNeeded : 蟻が巣に帰るまでに辿らなければいけないエサの数を指定。
  - All : すべて
  - One : どれかひとつでよい
- Pattern
  - Clear : すべてのエサ・障害物・巣をクリアする。
  - Filled : すべてのセルを障害物にする。
  - Random : 障害物をランダムに設置する。

- AdvancedControlPanel



- Auto-Adjust : 時間の経過とともに自動的にパラメータを調節する。
- Δ ... : どれくらいの速さでパラメータを調節するか。
- Max pheromone : フェロモンの最大値
- Evaporation : どれくらいの速さでフェロモンが蒸発するか。
- Dropoff : エサや巣から遠くなるにつれ、  
どのくらいの割合でフェロモンが弱くなるか。
- Trail Strength : どのくらいの確率で  
蟻がフェロモンが強い方へ向かうか。
- Adapt Time : どのくらいの速度でパラメータを調節するか。

- **Ant** (各蟻の動きを制御する)

蟻の状態は二つ：エサを見付ける or 巣に戻る

＜蟻の状態が「エサを見付ける」の場合＞

- ① 自分のまわりの 8 つのセルを順に見て、エサのフェロモン（エサから離れた蟻が出すフェロモン）の量を調べていく。
- ② 指定された確率（デフォルトは 50%）で、エサのフェロモンが一番強いセルに、または、ランダムに動く。
- ③ 以上を、指定された数(One or All)のエサを見付けるまで、1 ステップごとに繰り返す。
- ④ 指定された数のエサを見付けた蟻は巣に戻る。

＜蟻の状態が「巣に戻る」の場合＞

- ① 自分のまわりの 8 つのセルを順に見て、巣のフェロモン（巣から出た蟻が出すフェロモン）の量を調べていく。
- ② 指定された確率（デフォルトは 50%）で、巣のフェロモンが一番強いセルに、または、ランダムに動く。
- ③ 以上を、巣に辿り着くまで繰り返す。
- ④ 巣にたどり着いた蟻はエサを見付ける。

- **Cell** (各セルの状態を制御する)

- ・蟻が通るとフェロモンが残される。
- ・時間が経つとフェロモンが蒸発していく。