

個体ベース空間動態モデル

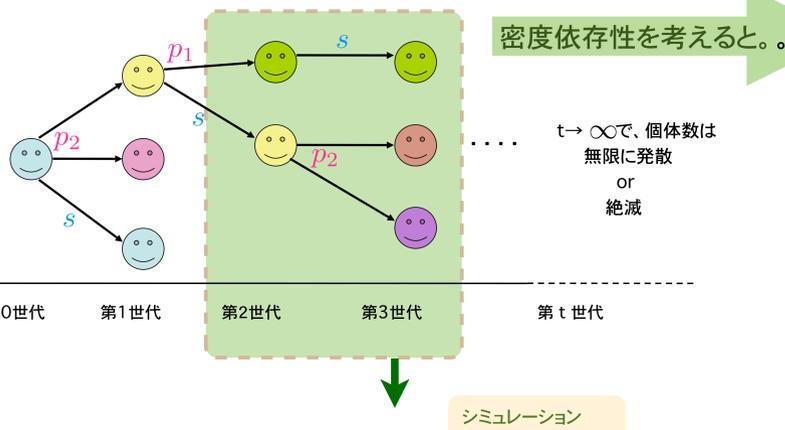
Individual-Based Spatio Temporal Dynamics

●皆藤千穂 (奈良女大院 人間文化), 高須夫悟 (奈良女 理) Email: chiho01k@ics.nara-wu.ac.jp

背景と目的

個体ベースモデルとは < Individual-Based Model >

- 個体を単位としたモデル
- 出生・死亡のルールをアルゴリズムとして表現して、各個体の生涯を計算機シミュレーションで再現するもの



集団の振る舞いや特性がどのようなものかがわかる!

個体数Nの時、 N^2 の計算量!

並列処理

複数のCPUを用いて効率的なシミュレーションを実現したい

総個体密度に依存する平均子供数:
実際の子供の数はポアソン分布で確率的に生まれるようにする

混み合っている!

空いている

$n_{t+1} = e^{r(1-\frac{n_t}{K})} n_t$

総個体密度に依存する子供の数/密度依存パラメータ

時刻 $t+1$ における個体密度

i の平均子供数 $e^{r(1-\frac{n_{i,local}}{K})}$

少ない i が感じる局所密度

j の平均子供数 $e^{r(1-\frac{n_{j,local}}{K})}$

多い j が感じる局所密度

●: 個体 i の個体の感じる局所密度

$n_{i,local} = \sum_{j \neq i} w(d(i,j))$

$w(d) = e^{-\frac{d^2}{2\sigma^2}}$ 個体 i と j の距離

σ : 相互作用が働く距離

$\sigma \gg \infty \rightarrow n_{i,local} = n_t$

$\sigma \ll 1 \rightarrow n_{i,local} < n_t$

個体の感じ $W(d)$

$\sigma = 10$

$\sigma = 0.5$

$\sigma = 0.1$

効率化するための手段

MPI (Message-Passing-Interface):

分散メモリ型の並列処理に用いられる。CPU間の通信を行うライブラリ

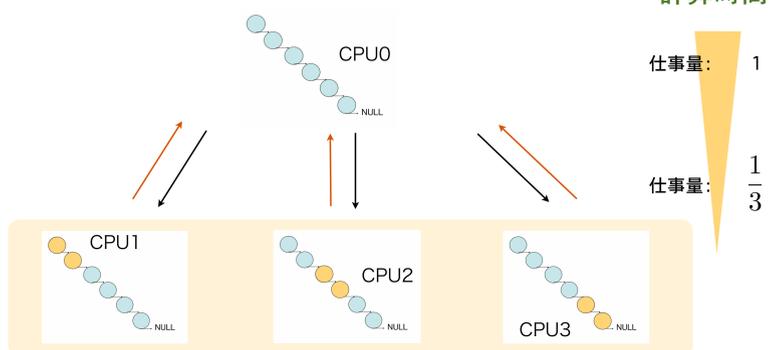
(例)



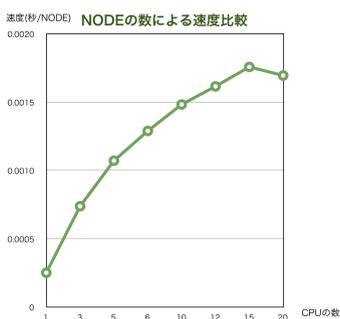
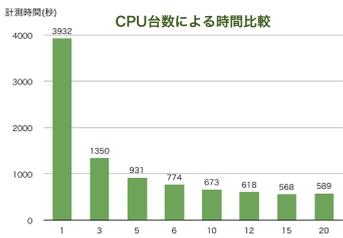
CPU 0	CPU 1
処理してほしいデータをCPU1に送信	CPU0からデータを受信
	受信データに対して計算
CPU1からデータを受信	CPU0に対して得られたデータを送信
受信したデータを表示する	

(例): CPU4台で計算させた場合

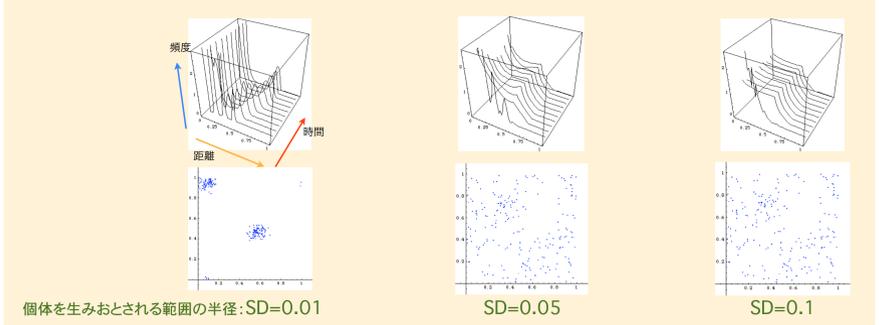
計算時間



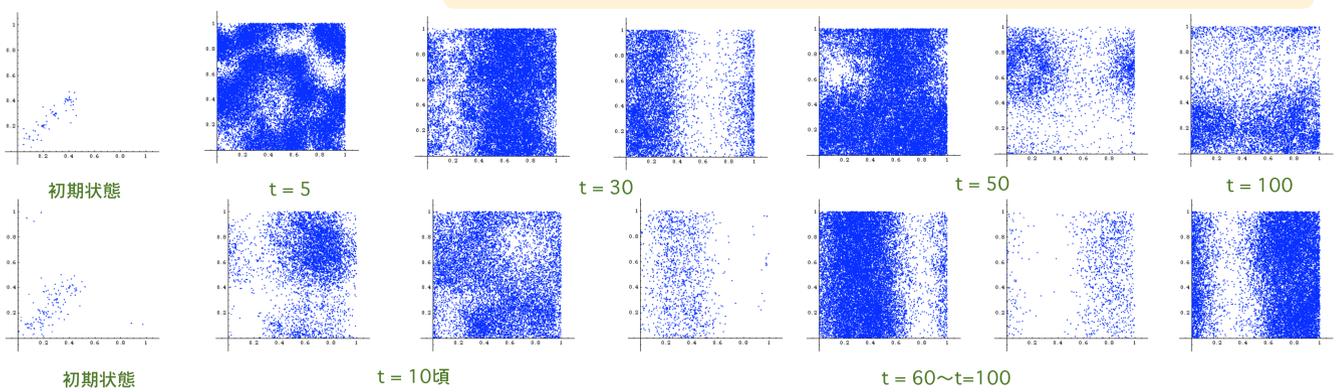
シミュレーション結果



2個体間の距離分布



(A) パラメータ値の設定:
密度依存がないときの子供の数 $b: 3.0$
密度依存パラメータ $b1: 0.01$
相互作用が働く距離 $\sigma: 0.1$
個体を生み落とされる範囲の半径SD: 0.05

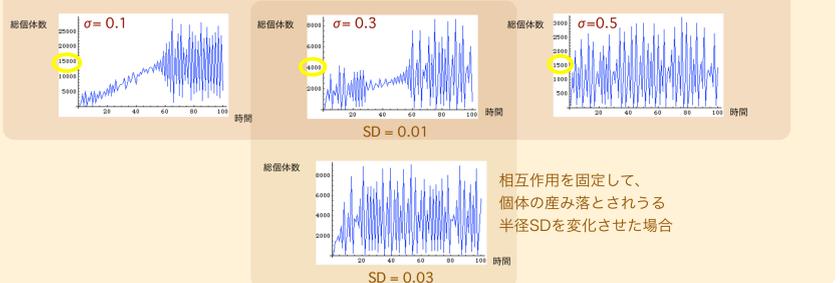


(B) パラメータ値の設定:
密度依存がないときの子供の数 $b: 3.5$
密度依存パラメータ $b1: 0.01$
相互作用が働く距離 $\sigma: 0.15$
個体を生み落とされる範囲の半径SD: 0.01



総個体数分布

個体の産み落とされる半径を固定して、相互作用 σ を変化させた場合



まとめと今後の展望

- 個体ベースモデルをMPIを使って複数のCPUで並列処理することで、処理時間の効率化を測った
- 2次元空間における局所密度依存性を考えたモデルに拡張することで、一步現実的なモデルへと近づけた
- 具体的な生物にあてはめてより現実的なモデルを作ること
- 1種系ではなく、寄宿主、宿主系のような2種系の解析
- これ迄は2個体間の距離分布のみに注目したが、3個体間の位置関係に注目して空間パターンの性質を解析