

## 可視化における独立固有時間刻み法による計算の高速化

松林達史 山田武士

日本電信電話株式会社 NTT コミュニケーション科学基礎研究所

**概要:** 本研究では、大規模なネットワークデータのための高速かつ効率的な可視化座標計算の手法を提案する。従来のネットワーク可視化手法のひとつとして Fruchterman 等による力学モデルを用いる手法が良く知られている。彼らの手法はノード間やエッジに対し力学関数を与えることにより、系全体のエネルギーを定義し、加速度方向に各ノードの座標を更新することによって、系のエネルギーの極小状態を求める。この手法では座標の更新頻度は一様で、全てのノードを毎回更新していたが、提案手法では階層的独立固有時間刻み法を用いて個々のノードに独立な更新時間を設定し、局所的に更新頻度を変えることにより計算の高速化を可能にした。この手法は、天体力学において用いられている局所的に密集した領域を精度よく計算する手法を、グラフ可視化手法に拡張したものである。また、提案手法は並列性に適しており、粒子間相互作用専用並列計算機 MDGRAPE-3 PCI-X に実装することによって、計算速度の数百倍高速化が可能であることを示した。

### Hierarchical Individual time algorithm for large scale graph-drawing

Tatsushi Matsubayashi and Takeshi Yamada

NTT Communication Science Laboratories, NTT Corporation

**Abstract:** In this paper, we propose a fast and efficient method for the large-scale graph layout problem. A conventional force-directed method proposed by Fruchterman and Rheingold and well-known in the literature, defines attractive and repulsive forces between nodes and obtains an optimal layout by minimizing the system potential energy. In their method, the positions of all the nodes are updated for every time step at the same time. In the proposed method, each node has its own individual time and time step, and they are updated independently with different frequencies depending on the local situation. The proposed method is inspired by the hierarchical individual timestep method used for the high accuracy calculations for dense particle field such as star cluster in the astrophysical dynamics. The experimental results show that the proposed method outperforms the conventional method in both speed and accuracy. We have implemented the proposed method on the MDGRAPE-3 PCI-X special purpose parallel computer, and succeeded in several hundred times speedup.

## 1 はじめに

複雑な関係データの表現方法として、ネットワークもしくはグラフを用い、これ等を可視化するという手法は様々な研究分野で広く用いられている。

ネットワーク表現を用いる利点は、グラフ理論など数学的手法による構造解析が可能という点である。また、ネットワークを低次元空間に配置することにより、データの潜在的な構造を浮き彫りすることができるという大きな利点がある。また、可視化することにより、大量なデータでも直感的理解が得やすいという点も挙げられる。グラフ可視化においてはネットワークの特性を利用して、効果的な可視化手法を考える必要がある。

本研究では、無向かつ重みのないグラフデータを対象に、数万ノードを越えるような大規模なネットワークの計算に耐えうる可視化の演算手法の開発を目指し、新しい可視化の計算手法を提案する。

## 2 従来研究

グラフ可視化において最もよく知られているのは“force-directed method”と呼ばれる手法であり、その原型は Eades (1984)[1] により提案された。この手法は、ノード間のグラフ距離や接続関係などに基づき、系にポテンシャルを与え、加速度方向に座標を更新してゆくことによって系の最小エネルギー状態を求める手法である。

Kamada and Kawai (1991)[2] は Eades の手法を基に、新たな手法を提案した。全てのノード間に仮想的なバネを仮定し、エネルギー準位が高いものから逐次的に1ノードずつ座標更新を行う。しかしながら、この手法ではノード数  $N$  に対して  $O(N^2)$  のメモリ領域が必要になる。それ故、この手法は大規模な可視化計算には適していない。

Fruchterman and Reingold (1991)[3] もまた Eades の手法を基に、独自の手法を提案した(以下 FR 法

と略す)。この手法では冷却関数を用いて、座標更新の最大幅を制限した。FR法ではノードにかかる力は、エッジ間に働く力を引力 $f_a$ 、ノード間に働く力を斥力 $f_r$ として、二つの力の和で与えられる。 $f_a$ 、 $f_r$ はそれぞれの式は以下で与えられる

$$\begin{aligned} f_a(x) &= x^2/k \\ f_r(x) &= -k^2/x, \end{aligned} \quad (1)$$

ここで $x$ は二つのノード間の距離であり、 $k$ は無次元量の距離でおおよそバネの自然長に相当する。FR法では(1)式で定義される加速度の方向に、全ノードを同時に座標を更新してゆき、この際最大更新幅を $t_{\text{cool}}$ で制限する。この時更新式は以下で与えられる

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{a}_i \times \min(1, t_{\text{cool}}(t)/|\mathbf{a}_i|). \quad (2)$$

ここで $a_i$ は(1)式から計算される加速度、 $t_{\text{cool}}(t)$ は次式で与えられる冷却関数である

$$t_{\text{cool}}(t) = C \times (1 - t/T_{\text{END}}). \quad (3)$$

$C$ は更新幅の初期値でフリーパラメータであり、FRの提案手法では $C = 0.1$ としている。しかしながら、式(3)による冷却関数では、最適な $T_{\text{END}}$ をあらかじめ設定する必要があり、大規模なネットワークの計算では適切な $T_{\text{END}}$ の設定が問題である。

### 3 提案手法

ノード間の加速度計算に関しては、これまでも様々な効率化手法が提案されてきたが、座標の更新手法は未だ改良の余地が残っている。従来のForce-directed法はいずれも“Shared timestep”と呼ばれる手法であり、座標更新は全て同時に行なわれる。ここで本研究では独立固有時間刻み法(Individual timestep method)を用いて、各ノードに固有の更新時間を持たせる手法を提案する。この提案手法は天体力学の高精度かつ高速な計算のために開発され[4]-[6]、特に局所的に密集した領域を精度良く計算を行なう手法として用いられている。本研究ではこの手法をグラフ可視化に拡張した手法を提案する。ここで“Shared timestep”と“Individual timestep”の更新手法の概念図を図1に示した。

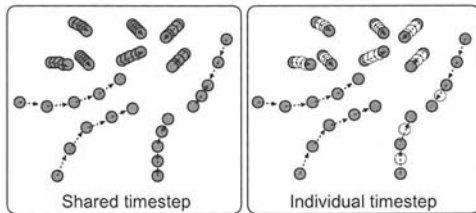


図1: 更新概念図。○は予測子を表す

Shared timestepでは全ノードを毎回更新を行なうために、更新座標がほとんど変化しない(加速度

がゼロに近い)場合でも毎回ノードを更新し続ける。一方 Individual timestepは、全体を支配する大局的な時間とは別に、個々のノードに独立な更新時間を設定し、加速度の小さなノードは更新のタイミングを長くする事により、大局的な時間には予測子を用いることによって効率良く計算を高速化することができる。

また固有時間を階層化することにより、複数のノードを時間同期させ、予測子の計算量の軽減のみならず、並列性を高め、計算のさらなる効率化を図ることができる。以下、階層的独立固有時間刻み法(Hierarchical Individual timestep)の詳細を述べる。

#### 3.1 階層的独立固有時間刻み法

座標更新はFR法と同様に force-directed法に基づき、加速度方向に座標を更新してゆく。この時 $i$ 番目のノードは $\Delta t_i$ 倍だけ更新してゆき、更新式は以下で与えられる

$$\mathbf{x}_i(t_i + \Delta t_i) = \mathbf{x}_i(t_i) + \Delta t_i \times \mathbf{a}_i(t_i). \quad (4)$$

ここでは $\Delta t_i$ はポテンシャル勾配に反比例させ、以下の式で与える

$$\Delta t_{\eta,i} = \frac{\eta}{|\mathbf{a}(t)_i|}. \quad (5)$$

ここでパラメータ $\eta$ は、本研究においては1.0を使用する。さらに階層的に固有時間を与えることによって粒子の同期性を高めて高速化を行なう。そこでを以下のように $\Delta t_i$ を2のべき乗で与える

$$\Delta t_i = 2^k \text{ if } 2^k \leq \Delta t_{\eta,i} < 2^{k+1} \quad (6)$$

ここで、注意すべき点は更新間隔の $\Delta t_i$ の変更である。 $i$ 番目のノードは更新毎に(5)、(6)式で評価を行ない、座標の更新条件は以下のようになる

$$\Delta t_{i,\text{new}} = \begin{cases} \Delta t_i/2 & \text{if } \Delta t_{\eta,i} < \Delta t_i \\ 2\Delta t_i & \text{if } \Delta t_{\eta,i} > 2\Delta t_i \\ & \text{and } t_i \bmod 2\Delta t_i = 0 \\ \Delta t_i & \text{otherwise.} \end{cases} \quad (7)$$

また計算の更新条件として時間刻みに上限( $\Delta t_{\text{max}}$ )と下限( $\Delta t_{\text{min}}$ )を設定する。上限を与えることにより、一定の間隔で定期的に時間の同期を行い、また下限を設定することによってFR法の冷却関数と同様に最大の更新幅の上限をある程度定めておく。これはランダム初期配置に対する計算の発散を抑えるためである。なお本研究では $\Delta t_{\text{max}} = 1$ 、 $\Delta t_{\text{min}} = 2^{-10}$ とした。

#### 3.2 更新手法

階層的独立時間刻み法では同じ固有時間 $t_i$ を持つノードは時間同期しているため、同時に更新を行なうが、異なる固有時間を持つノードからの力の計算は予測子を用いて仮想ノード座標からの相互作用

を計算する。予測子は対極時間  $t$  と固有時間の差分だけ座標更新させて、以下のように与える

$$\mathbf{x}_{p,j}(t) = \mathbf{x}_j(t_j) + (t - t_j) \times \mathbf{a}_j(t_j). \quad (8)$$

上記の予測子を用いて再び加速度を求める。ここで、加速度はFR法の加速度公式(1)を用いて以下のように与える

$$\mathbf{a}_i = \frac{1}{k} \sum_{j \neq i}^{E_i} |\mathbf{x}_{ij}| \mathbf{x}_{ij} - k^2 \sum_{j \neq i}^N \frac{\mathbf{x}_{ij}}{(\mathbf{x}_{ij}^2 + \epsilon^2)}. \quad (9)$$

ここで

$$\mathbf{x}_{ij} = \mathbf{x}_{p,j} - \mathbf{x}_{p,i} \quad (10)$$

であり、 $E_i$  は  $i$  番目のノードから出ているエッジの本数である。Softening parameter  $\epsilon$  は、(9)式において、右辺第2項が  $|\mathbf{x}_{ij}| \rightarrow 0$  で発散してしまうために、N体数値計算においてよく用いられる。ここで本研究では  $\epsilon=0.01$ 、また定数  $k=1$  とした。

(9)式において、加速度の計算は全ノードで行なわず、時間が同期している  $n_p$  個のノードだけ更新を行う。その粒子数を  $n_p$  として、この  $n_p$  個のノードに対して

$$\mathbf{x}_i(t_i + \Delta t_i) = \mathbf{x}_{p,i}(t) \quad (11)$$

の更新を行い、再び  $\mathbf{a}$  を求め上記を繰り返し更新してゆく。

### 3.3 評価方法

#### 3.3.1 系のエネルギー $\Psi$

本節では、大規模なネットワークに提案手法を適用し、その結果を定量的に評価する。力学方程式を与えれば、ポテンシャルエネルギー  $\Psi$  は加速度の式(9)とポテンシャル勾配の関係より以下のように決定される、

$$\Psi = \frac{1}{3k} \sum_{i,j}^E |\mathbf{x}_{ij}|^3 - \frac{k^2}{2} \sum_i^N \sum_{j \neq i}^N \log \sqrt{\mathbf{x}_{ij}^2 + \epsilon^2}. \quad (12)$$

以下はこの  $\Psi$  を評価関数として議論を行なう。

#### 3.3.2 平均ステップ数 $n_{ave}$

ここで計算量に対してのエネルギー収束性評価のために、FR法の計算ステップ数  $T$  に対し、提案手法では  $n_p$  の和を全ノード数  $N$  で割った、以下の式で定義される  $n_{ave}$  を用いて評価した、

$$n_{ave} = \frac{1}{N} \sum_{t=0}^T n_p(t). \quad (13)$$

実際にはエッジの数  $E$  も計算負荷に関わるが、 $N^2 \gg E$  よりもノード間の反発力部分にかかる負荷が高い事がわかる。

## 4 計算結果と考察

本研究において、グラフデータは Opte Project[7] の IP ネットワークのデータを用いた(ノード数  $N =$

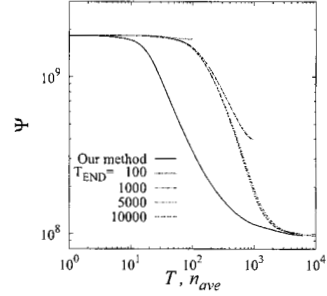


図 2: FR 法と提案手法との比較



図 3: FR 法の可視化結果. 左より  $T_{END} = 100, 1000, 10000$

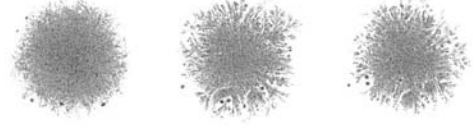


図 4: 提案手法の可視化結果. 左より  $n_{ave} = 100, 1000, 10000$

40027, エッジ数  $E = 47215$ ). このデータは同サイトからデータファイルをダウンロードすることが可能であり、本研究では 2005 年 1 月に収集されたデータを用いた。

### 4.1 可視化結果

我々の手法と FR 法で大きく異なるのは  $T_{END}$  の設定である。FR 法では最大更新幅を抑えるために収束させる時間  $T_{END}$  をあらかじめ設定する必要がある。しかしながら大規模なネットワークの可視化座標計算では、あらかじめ  $T_{END}$  を予測するのは困難であり、十分にエネルギー  $\Psi$  が収束していない場合、計算を再度やり直す必要がある。そこで FR 法に対し、 $T_{END} = 100, 1000, 5000, 10000$  と変えて計算を行なった。結果を図 2 に示す。

この図より  $T_{END} = 100, 1000$  ではエネルギーがまだ十分収束せず、 $T_{END} \geq 5000$  とする必要があることが分かる。対比のために提案手法を実線で示した。提案手法では FR 法よりも早期にエネルギーが収束していることが分かる。

ここで可視化結果を図 3 と図 4 に示す。FR 法では  $T_{END} \geq 5000$  としなければ全く見えてこなかった構造が、提案手法では初期の段階から全体的な構



造が見え始める。これは提案手法が階層的に構造を計算してゆくために、クリーク性の高いサブネットワーク構造に対し、初期段階から優先的に計算が行なわれるからである。

提案手法ではノードの更新を階層的に設定することにより、FR法よりも早い段階でネットワーク全体の構造を与えることが示された。

## 4.2 結論と考察

提案手法とFR法では、大きな違いは二点ある。一点は更新ノードの独立性を持たせ、FRが全て同時に更新し続ける手法に対し、提案手法は階層的に更新時間を持たせている点である。そのため提案手法ではノードの更新粒子を効率化させることにより計算の高速化に成功した。もう一点は $T_{\text{END}}$ の設定の有無である。FR法では図3からも分かるように可視化結果が $T_{\text{END}}$ の値に依存し、例えば $T_{\text{END}} = 100$ の計算の後に $T_{\text{END}} = 1000$ 、 $T_{\text{END}} = 5000$ とパラメータを変えて再計算を行なわなければならない。一方提案手法ではあらかじめ計算の終了時刻を設定する必要はなく、再計算の必要性は無い。本研究ではFR法と対比させるために $n_{\text{ave}}$ を用いて $n_{\text{ave}} = 10000$ ( $\simeq T = 300$ )まで計算を行なったが、経験的には提案手法では、 $T = 100$ まで行なえば十分な結果が得られる。

従来FR法ではこのような大規模なネットワークの可視化は想定されておらず、中規模の $N \simeq 100$ 程度のネットワークの可視化を想定し、 $T_{\text{END}}$ も100程度とされてきた。しかしながら、近年の計算機環境の向上とネットワークの肥大化に従いより大規模なネットワークの計算が必要となっている。今後は提案手法のような、より高速かつ効率的な計算手法が必要となってくる。

## 5 MDGRAPE-3による高速化

可視化の加速度及び、エネルギーの計算においては式(9)、(12)に示されるように、ノード間の計算は通常 $O(N^2)$ の計算量を要する。大規模ネットワークの計算においてはここの部分の計算が全体の99.9%以上を占める。そこで我々は、ノード間の計算に関してはMDGRAPE-3を、その他の計算はホストマシンで行う手法を用いた。MDGRAPE-3は理化学研究所で開発された粒子間相互作用専用計算機[8]であり、本研究ではMDGRAPE-3チップを二つ搭載したPCI-X版を用いて計算を行った。

結果としてMDGRAPE-3を用いた計算では両手法に対して数百倍の高速化を実現し、実際にかかった計算時間はFR法で $T_{\text{END}} = 10000$ で60分、提案手法では $n_{\text{ave}} = 10000$ で70分であった。汎用マシンを利用した場合はそれぞれ一週間以上の計算時間を要することになる。

## 6 おわりに

本研究では更新粒子の選択効率を評価するために全ノードからの加速度を計算した。しかし近年の手法では、FR法では遠方からの影響を完全に無視し、Quigley and Eades(2000)[9]の手法ではBarnes-Hut Tree code[10]を用いて重心近似を行っている。また近年において最も大規模な可視化計算では500万ノードを超える可視化が、Lyon(2003)[7]によって行なわれた。“Opte Project”と呼ばれ、本研究でも同Projectのデータを用いた。しかしながら、この手法は初期にネットワーク構造を破壊してしまうため、可視化結果の品質には問題を残している。

提案手法でもQuigleyらの手法と同様に、粒子の加速度をさらに近似をすることが可能であり、今後はこれらの近似手法との混合手法の検討を行なう。

## 参考文献

- [1] Eades, P.: A heuristic for graph drawing, *Congressus Numerantium*, 42, 149-160 (1984)
- [2] Kamada, T., and Kawai, S.: An algorithm for drawing general undirected graphs, *Information Processing Letters*, 12, 31, 7-15 (1989)
- [3] Fruchterman, T. M. J., and Reingold, E. M.: *Graph Drawing by Force-directed Placement*, Software - Practice and Experience, 11, 21, 1129-1164 (1991)
- [4] Ahmad, A., and Cohen, L.: A numerical integration scheme for the N-body gravitational problem, *Journal of Computational Physics*, 12, 389 (1973)
- [5] McMillan, S. L. W.: *The Vectorization of Small-N Integrators*, Lecture Notes in Physics, 267, 156 (1986)
- [6] Makino, J.: A Modified Aarseth Code for GRAPE and Vector Processors, *Publications of the Astronomical Society of Japan*, 43, 859-876 (1991)
- [7] Lyon, B.: The Opte Project(2005) <http://www.opte.org/>
- [8] T. Narumi et al.: A 55 TFLOPS Simulation of Amyloid-forming Peptides from Yeast Prion Sup35 with the Specialpurpose Computer System MDGRAPE-3, *Proceedings of the SC06*, CDROM, Tampa, USA, Nov. (2006)
- [9] Quigley, A., and Eades, P.: FADE:Graph Drawing, Clustering, and Visual Abstraction, *Proceedings of Graph Drawing 2000*, Lecture Notes in computer Science, 183-196 (2001)
- [10] Barnes, J., and Hut, P.: A hierarchical  $O(N \log N)$  force-calculation algorithm, *Nature*, 04, 324, 446-449 (1986)