

研究紹介

超並列環境での進化型計算

モデリング研究系助手 染谷 博司

確率的最適化手法のひとつである遺伝的アルゴリズムなどの進化型計算は、集団型の探索を特徴とします。そのため、並列処理を得意とするような計算環境で進化型計算を実行できれば、より広範な進化型計算の利用が期待できます。本稿では、このような計算環境における進化型計算の例として、「グリッド環境における進化型計算」と、「分

子計算環境における進化型計算」を紹介いたします。

【グリッド環境での進化型計算】

近年、高速ネットワークを利用しインターネット上の膨大な計算資源を共有利用した超並列計算を可能とするグリッド技術が注目されています。本研究では、グリッド環境上にて実現可能な進化型計算を示しその性質および有効性について調査しました。

グリッドでの計算では、(1)セキュリティ、(2)

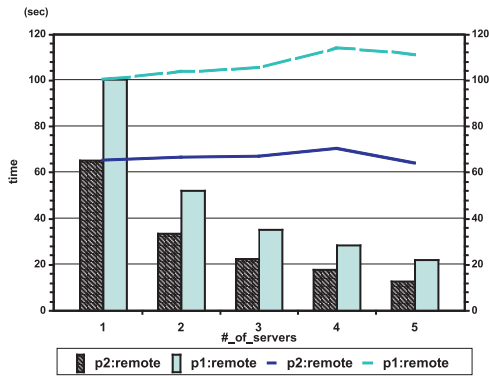


図1 グリッド環境での実験結果

不均質性、(3)不確実性、(4)非同期性、(5)超並列性、を考慮する必要があります。(1)~(3)はミドルウェア等での工夫が可能ですが、(4)(5)に関してはアプリケーションレベルでの工夫が必要であり、グリッド環境において実装される進化型計算はこれらについて考慮されていなければなりません。また、通信量についても考慮する必要があり、グリッドでは通信の遅延時間が大きいため、通信量が小さく通信頻度が少ないことが望ましいと考えられます。非並列な高性能な進化型計算と同程度の性能を維持したままで計算時間を短縮することも望ましい条件として挙げられます。

筆者らは、適応的な探索をする進化型計算の一手法として、*Genetic algorithm with Search area Adaptation*(GSA)を提案しています。GSAは、代表的ないくつかのベンチマーク問題において、その有効性が確認されています。グリッド環境におけるGSAの実現可能性に関する知見を得るために、簡略化したGSAを実装しその計算時間短縮効果を調査しました。その結果、サーバ数が多い場合でも並列数に対してほぼ比例した計算時間の短縮が可能であることを示唆する実験結果が得られました(図1)^{(1)~(3)}。

【分子計算における進化型計算】

分子計算では、DNAをデータ格納媒体とし、分子生物学的手法によって計算アルゴリズムを実行します。電子計算機と比較し、生体分子に特有の利点と制約を有しており、例えば、膨大な解候補を一度に生成し評価することができるという特徴があります。この超並列性を活かした、確率的最適化手法のDNA分子による実装の可能性を検討しました。

本研究では、タンパク質工学への応用を念頭に、特に、「探索が任意のある一点から開始される」という条件のもとでの確率的最適化を興味の対象

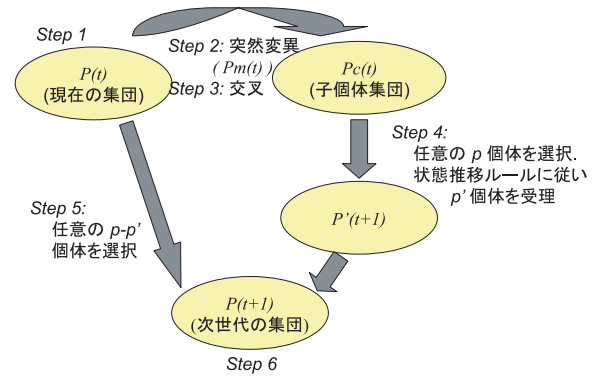


図2 DNA分子による進化型計算

としました。まず、現在広く知られている代表的な確率的最適化手法のDNA分子での実装可能性を検討したところ、これは困難であるとの結論に至りました。そして、進化型計算に基づいた手法を提案し(図2)、その有効性を計算機シミュレーションにより確認しました^{(4)~(6)}。

文献

- (1) 染谷博司：グリッド環境に適した遺伝的アルゴリズムによる最適化，統計数理，Vol.52, No.2, pp.381-391(2004)。
- (2) 染谷博司：進化型計算による適応的探索およびグリッド環境への応用，最適化：モデリングとアルゴリズム17，統計数理研究所(2004)。
- (3) 染谷博司：グリッド環境に適した遺伝的アルゴリズムに関する考察とその実現，電子・情報・システム部門大会2003講演論文集(CD-ROM)，電気学会(2003)。
- (4) 坂本健作，山村雅幸，染谷博司：タンパク質工学のための遺伝的アルゴリズムの設計，電子・情報・システム部門大会2004講演論文集(CD-ROM)，pp.435-438(OS5-10)，電気学会(2004)。
- (5) 染谷博司，山村雅幸，坂本健作：マルコフ連鎖モンテカルロ法の分子計算による実現の一検討，電子・情報・システム部門大会2005講演論文集(CD-ROM)，pp.466-469(OS3-4)，電気学会(2005)。
- (6) Sakamoto, K., Yamamura, M. and Someya, H.: Toward "Wet" Implementation of Genetic Algorithm for Protein Engineering, in Ferretti, C., Mauri, G. and Zandron, C. eds., DNA Computing, Vol. 3384 of *Lecture Notes in Computer Science*, pp.308-318, Springer-Verlag Berlin Heidelberg(2005)。