NetSolve System を用いた PSA/GAc によるタンパク質立体構造予測

Protein Tertiary Structure Prediction by Parallel Simulated Annealing using Genetic Crossover with NetSolve System

青井 桂子
Keiko AOI

Abstract: In this study, a protein tertiary structure prediction system by energy minimization in the Grid environment is proposed. The system uses NetSolve which is one of the GridRPC systems as a tool which can be used in the Grid environment, and it was applied to Parallel Simulated Annealing using Genetic Crossover (PSA/GAc) which is one of the optimization algorithms. On the other hand, in NetSolve, an overhead time occurs in case RPC is performed. When many RPC is performed in parallel, the parallel efficiency of the model synchronizing frequently declines. Therefore, in this study, the asynchronous model which can perform efficient search in shorter total execution time is mounted. Moreover, the asynchronous model and the synchronous model are applied to the energy minimization of Met-enkephalin, and the total execution time and the searching ability of two models are compared. The searching ability of two models was revealed almost more nearly equal from the experiment result, and it became clear that the asynchronous model can obtain faster speed than the synchronous model.

1　はじめに

自然に存在するタンパク質の立体構造は，系の自由エネルギーの最小状態に対応している．このため，タンパク質の立体構造予測の 1 手法として，タンパク質のエネルギー最小状態を最適化アルゴリズムを用いて探索する手法が挙げられる．本研究では，最適化アルゴリズムとして，タンパク質のエネルギー最小化に有効であるとされている遺伝的交叉を用いた並列シミュレーテッドアニーリング (Parallel Simulated Annealing using Genetic Crossover : PSA/GAc) [1] を用いた．

一方で，数百残基からなるタンパク質のエネルギー最小化には，膨大な解空間を探索する必要があり，1 機関が持つ計算資源では十分でなく，Grid の利用が考えられる．本研究では，Grid 環境を構築するためのツールとして NetSolve [3] を用いた．NetSolve は RPC を実行する際にオーバーヘッドが生じる．従来の PSA/GAc は交叉の際にすべての個体間で同期をとるため，NetSolve を用いた場合，個体数が増えると高速化がはかれないと考えられる．このため，本研究ではより短い実行時間で効率の良い探索が行える PSA/GAc の非同期モデルを提案する．また，非同期モデルと同期モデルを，総実行時間と解探索性能から比較し，非同期モデルの有効性を検討する．

2　GridRPC と NetSolve

2.1　GridRPC

現在，Grid ミドルウェアは様々なものが開発されており，いくつかのアプリケーション・プログラミングモデルを提供する．GridRPC [2] はその代表的なモデルの 1 つである．

GridRPC は，遠隔地に存在する計算機上のライブラリ呼び出しを提供する規格であり，Global Grid Forum[4] にて API の標準化が検討されている．

2.2　NetSolve システム

NetSolve は，Tennessee 大学 Innovative Computing Laboratory の Jack Dongarra 等によって開発された GridRPC システムの 1 つである．

NetSolve は Client，Agent，Server の 3 つから構成されている．NetSolve システムの概念図を Fig. 1 に示す．Grid 上のライブラリやハードウェアを使用したいユーザやアプリケーションは NetSolve Client となる．NetSolve Client は API として提供されている NetSolve Client Interface を使用することで，NetSolve Agent に対して，ライブラリの使用を要求する．

NetSolve では，Server での計算時間以外に Agent への Server の問い合わせ，Server へのデータの送受信など，オーバーヘッドが生じる．

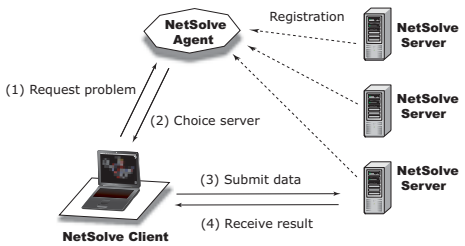

Fig. 1　Use of the NetSolve System

3 遺伝的交叉を用いた並列 SA

3.1 PSA/GAc の概要

遺伝的交叉を用いた並列シミュレーテッドアニーリング (Parallel Simulated Annealing using Genetic Crossover: PSA/GAc) は，並列に実行する各 SA の解の伝達時に GA の交叉を用いたものである [1)]．GA のオペレータを用いた SA であるため，SA の探索点の総数 (並列数) を個体数と呼んでいる．

3.2 GridRPC を用いた PSA/GAc の同期モデル

GridRPC を用いた PSA/GAc を実装するには，PSA/GAc の SA 部分を計算 Server で処理し，交叉を Client で処理するモデルが考えられる．PSA/GAc の同期モデルの概念図を Fig. 2 に示す．

Fig. 2 において，1 つの Server につき 1 個体が SA を行う．NetSolve を用いて実装する際には，Server 側に SA のプログラムを用意する．PSA/GAc の同期モデルでの探索手順を以下に示す．

step1 初期個体を生成する．

step2 Client は Agent に個体数分の RPC 要求を行い，Agent から紹介された SA が実行できる Server に個体のデータを送信する

step3 各 Server で交叉周期 d まで SA を実行する

step4 すべての Server から結果が Client に返されると，ランダムな個体のペアで設計変数間交叉を行う

step5 終了条件を満たすまで step2〜step4 の処理を繰り返す

PSA/GAc の同期モデルは通常の PSA/GAc のアルゴリズムを変更することなく実装できるため，解探索能力は同等である．NetSolve システムを用いた PSA/GAc の同期モデルは，Server での SA の計算時間や通信時間は各々独立して並列化が可能である．しかし，RPC を呼び出す部分は逐次で実行する必要があり，Client で繰り返し RPC を実行する際に，1RPC につき 0.2*sec* ほどの遅延が生じることがわかっている．このため，NetSolve を用いた PSA/GAc の同期モデルは，最低でも (個体数-1) 回の待機時間と 1 回の RPC にかかる通信時間と Server での計算時間がかかる．

また，いずれかの Server の負荷が高く，計算時間が他の Server よりも長い場合も他の個体は待機しなければならない．つまり，同期モデルは，様々なスペックの計算機が混在する Grid 環境において並列化効率が著しく低下すると考えられる．

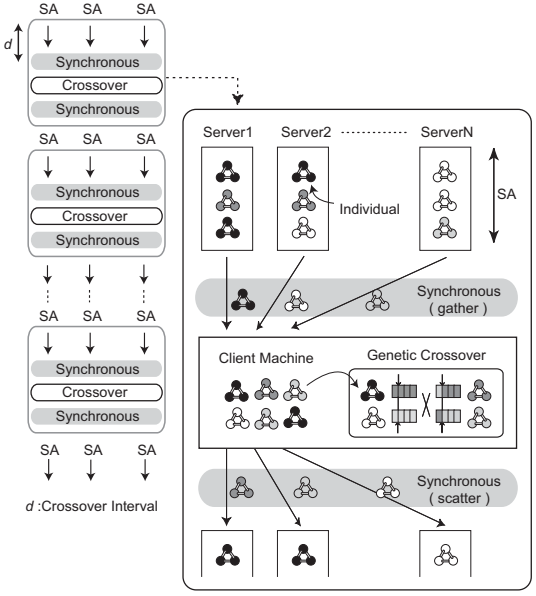

Fig. 2 Synchronous Model of PSA/GAc

3.3 GridRPC を用いた PSA/GAc の非同期モデル

本節では，GridRPC を用いたタンパク質立体構造予測システムのための PSA/GAc の計算モデルを提案する．NetSolve は RPC を実行する際に通信時間などのオーバーヘッドが生じる．また，1 回の RPC が失敗した場合，探索の途中結果が返されることがないため，再度 Client から RPC を実行する必要がある．このため，交叉周期毎にすべての個体で同期をとると，並列化効率が著しく低下する．提案モデルでは，各個体が交叉周期ですべての個体が同期するわけではなく，Server から 2 個体受信した時点で交叉を行い，交叉後に選択された個体を再び Server に送信する非同期モデルである．PSA/GAc の非同期モデルでの探索手順を以下に示す．

step1 初期個体を生成する

step2 Client は全個体に対して Agent に RPC 要求を行い，Agent に紹介された各 Server に個体を送信する

step3 各 Server では交叉周期 d まで SA を実行する

step4 交叉周期 d になると，結果が Client に返され，受信アーカイブに格納される

step5 受信アーカイブに 2 つ以上の個体が存在する場合，アーカイブの先頭 2 個体を用いて設計変数間交叉を行い，子個体を生成する

step6 もとの親と生成した子との 4 個体のうち評価値の高い 2 個体を次の SA の探索開始点とし，送信アーカイブに格納する

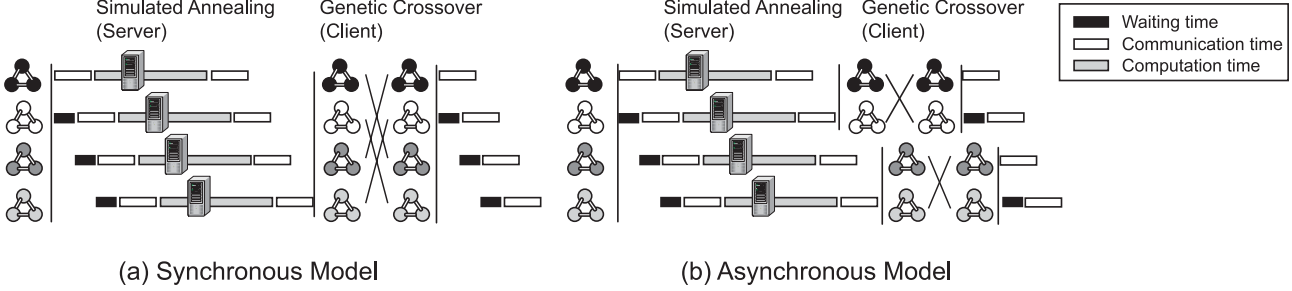

Fig. 3　Synchronous Model and Asynchronous Model of PSA/GAc with NetSolve

step7 送信アーカイブに個体が存在する場合，Client は Agent に RPC 要求を行い，Agent に紹介された Server に個体を送信する

step8 終了条件を満たすまで step3〜step7 の処理を繰り返す

Fig. 3 に NetSolve を用いて実装した PSA/GAc の同期モデルと非同期モデルの処理の違いを示す．

PSA/GAc の非同期モデルは通常の PSA/GAc のアルゴリズムとは異なり，交叉に用いられる個体にランダム性がなくなる．また，ステップ数の異なる個体が交叉する場合が生じると考えられ，解探索性能が異なる可能性がある．しかし，非同期モデルでは個体数に依存しないため，Client での待機時間が短縮できる．

また，いずれかの Server の負荷が高く，計算時間が他の Server よりも長い場合も，2 個体づつ同期をとるため，処理の遅い Server で計算されている個体を待つ必要はない．特に，計算環境が Cluster 環境のような均一な計算資源がそろっているものではなく，Grid 環境を想定した場合，各 Server が異なるスペックを持っているため，すべての個体間で同期するようなモデルではなく，非同期モデルが有効であると考えられる．

4　数値実験

4.1　実験条件

本研究においては，タンパク質の二面角を設計変数とし，エネルギー関数 ECEPP/2 [5] に基づいた気相中のエネルギー最小化を行う．本研究で対象とするタンパク質は，19 個の二面角を持つ Met-enkephalin であり，ECEPP/2 エネルギー関数に基づいた気相中において，-11$kcal/mol$ 以下で最小エネルギー構造を形成することが確認されている．

各二面角において順に SA の生成・受理判定を行ってから 1 回のクーリングを行うこととし，これらの処理を 1Monte Carlo sweep（MCsweep）と呼ぶこととする．

PSA/GAc のパラメータには Table 1 に示したものを用いた．Table 1 のうち，温度スケジュールと近傍には，岡本らが実験で用いたパラメータ [6] を採用している．クーリングには指数型クーリングを用いており，1MCsweep ごとに一定のクーリング率を現在の温度に乗じるものとした．また，交叉周期は予備実験により経験的に得た値を定めた．終了条件は RPC を 個体数×15 回まで繰り返し，1 個体あたりの平均評価計算回数が 1500MCsweep となるように設定した．

4.2　数値実験：同期モデルと非同期モデルの比較

Grid 環境に最適化システムを構築する際，高速化と最適化アルゴリズムの解探索性能が求められる．本節では，NetSolve を用いた同期モデルと非同期モデルに対

Table 1　Parameters of PSA/GAc

Parameter	Value
Initial Temperature	2.0 (1000K)
Population size	2, 4, 8, 16
Crossover Interval	100
Cooling Rate	0.998
Range Size	$180° \rightarrow (180 \times 0.3)°$
Trials	20

Table 2　Spec of Cluster

Number of node	18
Processor	Pentium　800MHz
Memory per processor	256MB
OS	Debian GNU/Linux3.0
Communication Layer	100Mbps Ethernet

Table 3　Spec of Machine

	# Proc	Processor	Memory
UTK(Server)	16	Pentium　550MHz	512MB
IS (Server)	3	Pentium4 2400MHz	512MB
IS (Agent)	1	Pentium　1100MHz	256MB
IS (Client)	1	Pentium4 2400MHz	512MB

Proc = Number of Processor

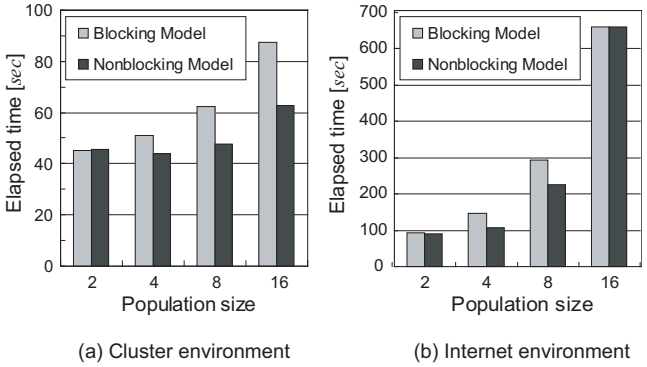

Fig. 4　Average Elapsed Time

して総計算時間と最適解の発見試行数を比較することにより，両モデルの時間効率と解探索性能を検討する．また，実験環境としては，クラスタ環境とインターネット環境の 2 種類で実験を行った．

クラスタ環境とインターネット環境での実験

クラスタ環境での実験で用いた PC クラスタの性能を Table 2 に示す．インターネット環境での実験に用いた計算機の性能を Table 3 に示す．また，同志社大学と Tennessee 大学間の通信スループットを Table 4 に示す．2 つの実験に用いたパラメータは Table 1 である．

Table 5 に 20 試行における最適解の発見試行数を示す．また，Fig. 4 に各モデルでの 20 試行の平均計算時間を示す．縦軸は時間 (*time*)，横軸は PSA/GAc の個体数 (並列数) を示している．

Table 5 の実験結果より，クラスタ環境，インターネット環境共に 2 つのモデルにおける最適解の発見試行数はほぼ同等である．このため，両モデルの解探索性能は同等であると言える．

また，Fig. 4 より，クラスタ環境において 2 つのモデルは PSA/GAc の個体数が増えるに従い，1 試行の実行時間が増加している．しかし，2 つのモデルを比較した場合，非同期モデルは同期モデルに比べてすべての個体間で同期を取らないため，短い時間で探索が終了している．インターネット環境においては，2 個体から 8 個体まではクラスタ環境同様，同期モデルよりも非同期モデルは短い時間で探索を終了しているが，16 個体の場合には両モデルで 8 個体の計算時間の 2 倍以上かかっている．これに関しては原因はわかっていない．

Table 4　Communication Throughput

Message size	Doshisha → UTK	UTK → Doshisha
1KB	1.97Mbps	2.13Mbps
16KB	1.80Mbps	2.13Mbps
256KB	1.83Mbps	2.18Mbps
1204KB	1.84Mbps	2.11Mbps

Table 5　Number of Times that Optimum is Found in 20 Trials

Population size	2	4	8	16
Blocking (Cluster)	0	4	8	9
Nonblocking (Cluster)	0	3	7	11
Blocking (Internet)	0	3	4	10
Nonblocking (Internet)	1	6	11	7

5　まとめ

本研究では，NetSolve システムを用いた PSA/GAc の非同期モデルの有効性を検証した．数値実験より，並列数が増えるにしたがい，非同期モデルは同期モデルよりも短い時間で解探索が行えることを示した．また，2 つのモデルの解探索性能はほぼ同等であることから，NetSolve システムを用いた PSA/GAc の非同期モデルの有効性が示された．

参考文献

1) Tomoyuki Hiroyasu, Mitsunori Miki, Maki Ogura and Yuko Okamoto. 遺伝的交叉を用いた並列シミュレーテッドアニーリングの検討 情報処理学会論文誌:数理モデル化とその応用, Vol. 43, SIG7(TOM6), pp. 70-79, 2002.

2) H.Nakada. GridRPC: A remoto procedure call api for grid computing. *http://www.eece.unm.edu/apm/*, 2002.

3) Henri Casanova, Jack Dongarra. Netsolve: A network server for solving computational science problems. *Proc. of Supercomputing '96 Conference*, 1996.

4) Global grid forum. *http://www.gridforum.com.*

5) M.J. Sippl and G. Nemethy and and H.A. Scheraga. *J. Phys. Chem.* , Vol. 88, pp. 6231-6233, 1984.

6) Yuko Okamoto, Takeshi Kikuchi, and Hikaru Kawai. Prediction of Low-Energy Structures of Met-Enkephalin by Monte Carlo Simulated Annealing. *CHEMISTRY LETTERS*, pp. 1275–1278, 1992.