

目的別テーマ：知能化及び最適化手法の確立

研究テーマ

15-6-19：繊維生産システムのための共進化型設計の確立

## ABSTRACT

*Our focus is on evolutionary systems that help us design intelligent manufacturing systems. In particular we will work on techniques from an area of computing known as Natural Computation, computation inspired by nature, examples of which include artificial neural networks (ANNs) and evolutionary algorithms (EAs). Especially, we will investigate the co-evolution of neural controllers for a team of robots, since co-evolution is a useful model of adaptation in societies of creatures. In the first stage we will evolve the controllers in simulation, and then transfer the evolved neural networks to the real robots.*

## 研究目的

現状の進化型計算を繊維生産システムに単体で適用したのみでは、その最適化能力が不十分な場合が多い。そこで、効率性の観点から共進化型設計手法を構築し、繊維生産システムに適用するための共進化型進化計算手法の設計論を構築することを目的とする。

## 5年間の研究内容と成果

繊維生産システムを想定した複数台の自律要素からなる繊維生産ロボットシミュレータを構築した。このシミュレータの特色は、計算時間短縮を狙った分散コンピューティング環境を利用できるように改良した共進化型進化計算手法のアルゴリズムを利用する点にある。これには、共進化型進化計算手法を並列分散計算可能にする共進化型進化計算のフレームワークを設計し、これに従った繊維生産ロボットシミュレータと共進化型進化計算手法のアルゴリズムを開発している。このフレームワークに従った設計指針とそれに従ったアルゴリズムの設計方法とは、様々な繊維生産システムに適用するための共進化型進化計算手法の設計論となる。また、副次的に整備された共進化型進化計算のフレームワークに従って分散コンピューティングを利用できるデスクトップグリッド計算環境では、信州大学、大阪大学、神戸大学、広島大学、バーミンガム大学の計算機群を利用できるようになった(図1)。さらにそこでは、利用可能な計算リソースがヘテロなグリッド環境の特徴を考慮した計算資源の選択・割り当て方法が重要であるため、動的なグリッドタスクスケジューリングを開発することによって、準最適にグリッド計算を行えるようになった。その結果、実験時間の大幅な短縮とエネルギー効率を考慮した運用が可能となった。

このヘテロなグリッド計算機環境では、クラス構造(図2)を持つWebサービスによるデスクトップグリッド計算環境上でRRを用いて進化型計算のプログラムを実行している。新規開発したデスクトップグリッド計算のプログラムの流れを以下に簡略化して示す。

1. クライアント側でメインプログラムを実行
2. メインプログラム内でジョブ投入用RRThreadとジョブを投げかけるためのESClientを起動
3. ESClientはIP, ジョブ, キュー, タスクリングを管理するTable, List, TaskRingを参照し, 全てのジョブを割り当てるまでWQを実行
4. ジョブはESClientが各マシンのESServiceを呼び出して実行

5. 割り当ての終了後 RR によるスケジューリングを実行
6. 強制終了及び再起動のため KillerThread, Invoker を実行
7. RRThread は強制終了後 ESClient の代わりとして起動し、ジョブを実行

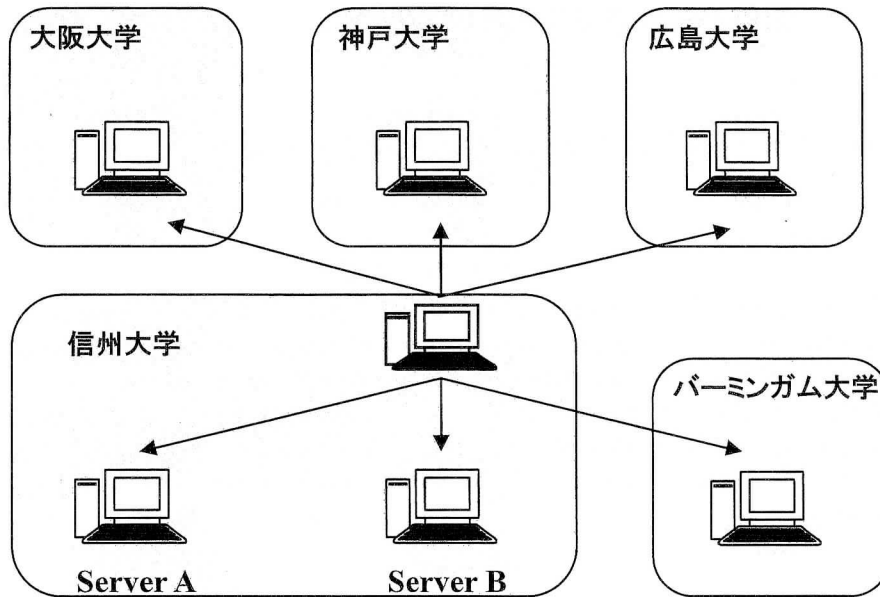


図1. デスクトップグリッド計算環境

この開発したデスクトップグリッド計算環境において基礎的な計算機実験を行った結果、共進化型進化設計をグリッド環境で行うためのいくつかの知見が得られた。代表的な知見を以下に簡単に列挙する。

1. 分散台数に比例して総実行時間は短くなる
2. RR の使用により実行時間が短縮される
3. ジョブの粒度が大きいほど RR に適している
4. ヘテロ環境ほど RR に適している
5. 通信時間が短い場合、実際の損失率は理論値に対して大きく下回る
6. 分散台数が増加すると損失率は極めて緩やかに増加する

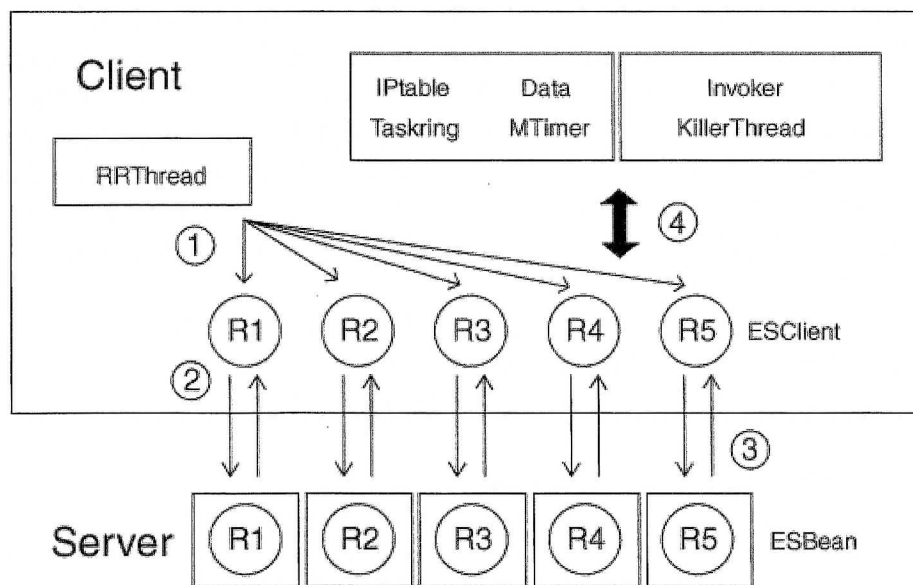


図2. グリッド計算環境におけるクラス構造