

目的別テーマ：知能化及び最適化手法の確立

16年度研究テーマ

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。

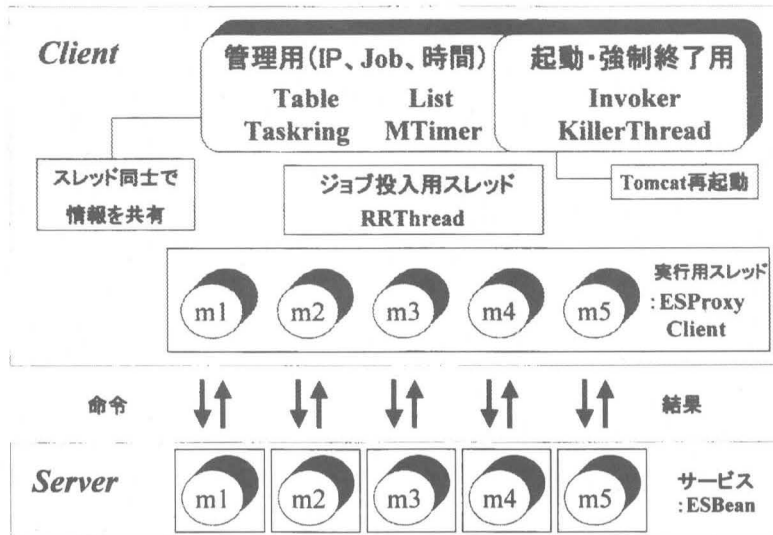
一年間の研究内容と成果

下記のようなクラス構成(表1)とクラス構造(表2)でWebサービスによるグリッド上でRRを用いる進化型計算のプログラムを実装し、ローカルな計算機環境において計算機実験を行った結果、今後のグローバルなグリッド環境で実験を行うためのいくつかの知見が得られた。

表1 クラス構成

ESサービス: Webサービスで、各マシンで実行させる	
ESBean.java	ESメインプログラム
CES.java	ESモジュール
Functions.java	ESモジュール
Cotype.java	ESモジュール
Sort_class.java	ESモジュール
クライアント側プログラム	
ExeRR.java	メインプログラム
ESProxyClient.java	実行スレッド
RRWorkerThread.java	RR用実行スレッド
RRThread.java	ジョブの投入用
Invoker.java	起動・強制終了スレッド呼び出し用
KillerThread.java	強制終了専用スレッド
Table.java	実行ジョブとListの管理
List.java	各マシンの実行ジョブとIPを管理
TaskRing.java	RR実行中のジョブを管理
ESBean.java	soapモジュール
ESBeanService.java	soapモジュール
ESBeanServiceLocator.java	soapモジュール
AxisServletSoapBindingStub.java	soapモジュール

表2 クラス構造



新規開発した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 ExeRR 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 ESProxyClient を起動
3. ESProxyClient 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. 割り当ての終了後 RR によるスケジューリングを実行
5. ジョブは ESProxyClient が各マシンの ESBean を呼び出して実行
6. 強制終了及び再起動のため KillerThread、Invoker を実行
7. RRWorkerThread は強制終了後 ESProxyClient の代わりとして起動し、ジョブを実行

ローカルな環境における計算機実験により得られた知見を以下に簡略化して示す。

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
7. 複製回数、強制終了回数は均質な環境と非均質な環境で大きな差が出ない

## 展望

昨年度に構築したローカルな計算機ネットワーク環境を利用して、Web サービスを利用するグリッド計算が可能となった。本年度の結果から得られた知見を基に、グローバルなグリッド計算環境を構築し、複数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する。これに並行して、グローバルな環境を利用できるフレームワークへの拡張と新規の共進化型進化計算アルゴリズムを開発する。また、実環境を考慮したシミュレータの制作が重要であるため、シミュレータと実環境とのギャップを意識した入出力のキャリブレーションを念入りに共同研究者と再調整する必要がある。