

논문 2004-41CI-4-8

# 서버의 계산능력을 활용한 네트워크기반 병렬유전자알고리즘의 성능향상

(Performance Improvement of Network Based Parallel Genetic  
Algorithm by Exploiting Server's Computing Power)

송 봉 기\*, 김 용 성\*\*, 성 길 영\*\*\*, 우 중 호\*\*\*\*

(Bong Gi Song, Yong Seong Kim, Kil Young Sung, and Chong Ho Woo)

## 요 약

본 논문에서는 네트워크기반의 클라이언트-서버모델에서 병렬유전자알고리즘의 최적해 수렴속도를 향상시키는 방법을 제안한다. 전역 최적해를 지역 엘리트의 평가만으로 구하는 기존의 방법과는 달리 제안한 방법은 서버에서 지역 엘리트의 평가를 통해 전역 최적해를 구하고 유휴시간에 유전자알고리즘을 적용하여 전역 최적해의 적합도를 개선한다. 서버에서 개선된 전역 최적해를 클라이언트의 유전자알고리즘에서 사용하므로 전체 알고리즘의 최적해 수렴속도가 향상된다.  $F_{\max}(g)$ 는  $g$ 번째 세대의 최대 적합도,  $G$ 는 서버에서 개선되는 세대수일 때, 지역 최적해의 이주 시 서버에서 개선되는 적합도는  $\sum_{g=1}^G (F_{\max}(g) - F_{\max}(g-1))$ 이다. 여기서 클라이언트의 수가 증가하면  $G$ 가 작아져서 적합도 개선치는 줄어드나 기존의 방법보다 적합도가 개선됨을 확인할 수 있었다.

## Abstract

This paper proposes a method improving the convergence speed of optimal solution for parallel genetic algorithm in the network based client-server model. Unlike the existing methods of obtaining global elite only by evaluating local elites in server, the proposed method obtains it by evaluating local elites and improving its fitness by applying genetic algorithm during idle time of the server. By using the improved chromosome in server for the client's genetic algorithm processing, the convergence speed of the optimal solution is increased. The improvement of fitness at the server during the interval of chromosome migration is  $\sum_{g=1}^G (F_{\max}(g) - F_{\max}(g-1))$ , where  $F_{\max}(g)$  is a max fitness of the  $g$ -th generation and  $G$  is the number of improved generation by the server. As the number of clients increases and  $G$  decreases, the improvement of fitness goes down. However the improvement of fitness is better than existing methods.

**Keywords:** 유전자알고리즘, 병렬유전자알고리즘, 염색체 이주, 분산 컴퓨팅, 클라이언트-서버모델

\* 정회원, 부경대학교 정보시스템  
(Interdisciplinary Program of Information System.,  
Pukyong National University)

\*\* 정회원, 경남정보대학 컴퓨터정보계열  
(Computer Information System Sub-division,  
Kyungnam College of Information & Technology)

\*\*\* 정회원, 경상대학교 정보통신공학과  
(Department of Information and Communication  
Eng., Gyeongsang National University)

\*\*\*\* 정회원, 부경대학교 전자컴퓨터정보통신공학부  
(Division of Electronics Computer and Communi-  
cation Eng., Pukyong National University)

접수일자: 2004년3월15일, 수정완료일: 2004년6월30일

## I. 서 론

유전자알고리즘(genetic algorithm)은 생물의 유전시스템을 모방하여 개체군에 선택, 교차, 돌연변이 연산들을 반복적으로 적용하여 근사 최적해를 구하는 다점검색기법으로 간단하고 유효시간 내에 근사 최적해를 얻을 수 있기 때문에 함수의 최적화, 조합문제, 제어문제, 기계학습 등의 다양한 분야에 적용된다.<sup>[1-3]</sup>

병렬유전자알고리즘(parallel genetic algorithm)은 유전자알고리즘의 계산속도 향상을 위해 연구되었고,<sup>[4-7]</sup>

최근에는 소규모 분산네트워크환경에서의 병렬유전자 알고리즘이 연구되고 있다.<sup>[8,9]</sup> M. Golub<sup>[5,6]</sup> 등은 마스터가 유전연산자를 수행하고 슬레이브는 염색체를 평가하는 마스터-슬레이브(master-slave) 구조의 병렬유전자 알고리즘을 제안하였다.<sup>[5,6]</sup> 이 연구에서는 공유메모리(shared memory)를 이용하여 통신량에 의한 계산속도의 감소를 줄인다. L. Tan<sup>[7]</sup> 등은 가상공동체모델(virtual community model)을 이용한 병렬유전자알고리즘을 제안하여 이주지연과 멀티캐스트에 따른 통신부하를 줄였다. 그러나 두 연구 모두 특수한 병렬컴퓨팅환경을 이용하기 때문에 기존의 네트워크를 이용하는 분산네트워크환경에는 적합하지 않다. K. Kojima<sup>[8,9]</sup> 등은 클라이언트-서버모델(client-server model)을 이용한 네트워크기반 병렬유전자알고리즘을 제안하였고 이 모델에서 통신량을 감소시키는 방안을 연구하였다. 여기서 서버는 염색체의 평가만을 수행하고 수렴속도 향상 방안에 대한 연구가 이루어지지 않았다.

본 논문에서는 네트워크기반 병렬유전자알고리즘을 기반으로 서버가 유전자알고리즘을 수행하여 최적해로의 수렴속도와 각 세대까지의 평균 최적해를 나타내는 오프라인성능이 향상되고 고정된 이주주기를 갖는 병렬유전자알고리즘을 제안한다. 서버는 버퍼와 엘리트 풀(elite pool)을 이용하여 각 클라이언트의 최적해인 지역 엘리트들을 모집단으로 하여 유전자알고리즘을 수행하고 전역 엘리트들 생성한다. 이 전역 엘리트가 클라이언트들에 전달되어 유전자알고리즘에 사용되기 때문에 최적해로의 수렴속도 및 오프라인성능이 향상된다.

본 논문의 구성은 II장에서 네트워크기반 병렬유전자알고리즘에 대해 설명하고 III장에서 성능향상을 위한 서버의 활용에 대해 설명한다. IV장에서는 제안한 모델을 평가하고 V장에서 결론을 제시한다.

II. 네트워크기반 병렬유전자알고리즘

병렬유전자알고리즘은 대부분 대형 병렬컴퓨터에서 구현되고 그 성능은 병렬컴퓨팅시스템의 연결구성(Topology) 및 성능에 의존적이다. 이 알고리즘들은 기존의 PC나 워크스테이션을 상호 연결한 네트워크를 이용하는 분산네트워크환경에서 구현이 어렵다. 이러한 환경에는 K. Kojima<sup>[8]</sup> 등이 제안한 클라이언트-서버모델을 이용한 네트워크기반 병렬유전자알고리즘이 적합하다.

Kojima의 알고리즘모델은 대리자관리모델(delegates

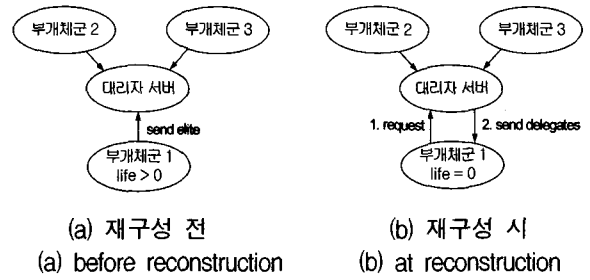


그림 1. 대리자 관리모델  
Fig. 1. Delegate management model.

management model)과 생명파라미터(life parameter)를 이용한 통신제어모델로 구성되고 그림 1과 같이 대리자 서버를 이용하여 염색체 교환을 수행한다. 이 모델은 부개체군을 대상으로 유전자알고리즘을 수행하는 클라이언트들과 각 클라이언트의 엘리트들을 관리하는 대리자 서버로 구성된다. 대리자 서버는 각 클라이언트들의 엘리트들을 평가하여 적합도가 가장 높은 엘리트를 대리자로 지정하고 이 대리자를 각 클라이언트의 요청에 따라 클라이언트로 전달한다.

생명파라미터는 유전자알고리즘을 통해 엘리트가 갱신되지 않으면 1 감소한다. 생명파라미터가 0보다 크면 서버로 지역 엘리트 전송하고 생명파라미터가 0이면 서버에 대리자를 요구해서 부개체군을 재구성한다.

병렬유전자알고리즘의 성능은 초기 집단에서부터 현재 집단까지 각 세대까지의 최적값의 평균을 나타내는 오프라인 성능  $X_e^*(s)$ 로 평가할 수 있다.<sup>[10]</sup>

$$X_e^*(s) = \frac{1}{T} \sum_{t=1}^T f_e^*(t) \tag{1}$$

여기서  $f_e^*(t) = BEST(f_e(1), f_e(2), \dots, f_e(t))$ .  $f_e(t)$ 는 세대  $t$ 에서의 목적함수이고  $f_e^*(t)$ 는 세대  $t$ 까지의 최적 적합도이다.

III. 성능향상을 위한 서버의 활용

네트워크기반 병렬유전자알고리즘에서 지역 엘리트의 이주간격 동안 서버에 유희시간이 발생한다. 이 유희시간에 서버가 유전자알고리즘을 수행하면 지역 엘리트의 적합도를 개선할 수 있을 것이다. 본 장에서는 제안한 알고리즘모델의 구조와 서버의 계산능력을 활용한 성능향상을 설명한다.

1. 병렬유전자알고리즘 모델

병렬유전자알고리즘 모델은 그림 2와 같이 하나의

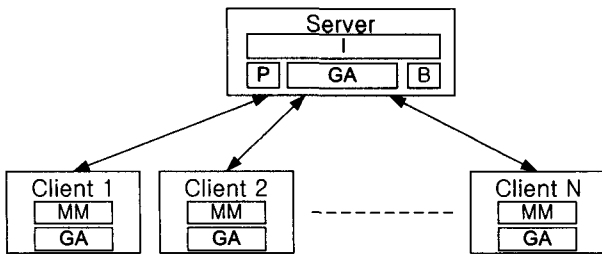


그림 2. 병렬유전자알고리즘 모델  
Fig. 2. Proposed parallel genetic algorithm model.

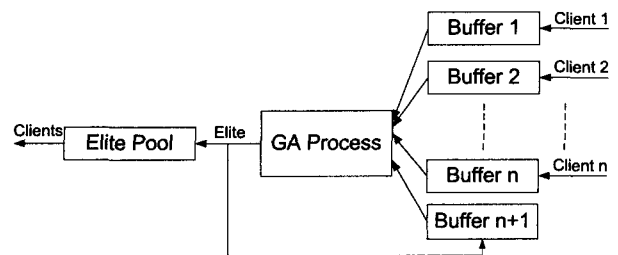


그림 4. 서버의 유전자알고리즘 프로세스  
Fig. 4. Genetic algorithm process in server.

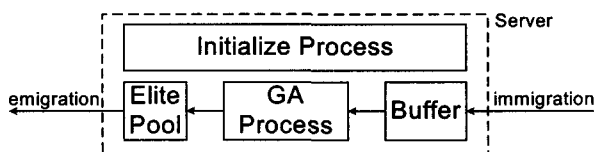


그림 3. 서버의 구성  
Fig. 3. Composition of server.

서버와 다수의 클라이언트를 가지는 클라이언트-서버 모델로 표현된다. 서버에서 I는 부개체군을 클라이언트에 할당하는 초기화 프로세스, B는 이주해 온 지역 엘리트들을 저장하는 버퍼, GAs는 버퍼의 지역 엘리트들을 개체군으로 하는 유전자알고리즘 프로세스, P는 서버에서 구한 전역 엘리트를 저장하는 엘리트 풀이다. 클라이언트에서 MM은 엘리트 이주를 담당하는 이주관리자(migration manager), GAc는 지역 엘리트를 생성하는 유전자알고리즘 프로세스이다.

2. 서버의 구성

서버는 그림 3과 같이 초기화 프로세스, 버퍼, 유전자알고리즘 프로세스, 엘리트 풀로 구성되고 버퍼와 엘리트 풀을 엘리트 이주에 사용한다. 초기화 프로세스는 각 클라이언트 접속 시에만 처리되도록 다른 모듈과 독립되어 구성되고 버퍼와 엘리트 풀은 각 클라이언트들과 연결된다.

서버의 유전자알고리즘 프로세스는 그림 4와 같이 각 클라이언트에 연결된 버퍼로 지역 엘리트들이 이주를 하고 버퍼의 지역 엘리트들과 엘리트보존선택연산에 의해 보존된 전역 엘리트를 개체군으로 유전자알고리즘을 수행한다. 유전자 알고리즘 프로세스의 처리 후에 새로운 전역 엘리트가 생성된다. 유전자알고리즘을 통해 구해진 전역 엘리트는 엘리트 풀을 갱신하고, 엘리트보존 선택연산에 사용된다.

3. 클라이언트의 구성

클라이언트는 그림 5와 같이 이주관리자와 유전자알

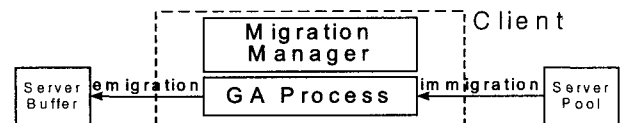


그림 5. 클라이언트의 구성  
Fig. 5. Composition of client.

고리즘 프로세스로 구성된다. 이주관리자는 클라이언트에서 서버로의 지역 엘리트 이주와 서버에서 클라이언트로의 전역 엘리트 이주를 담당하고, 유전자알고리즘 프로세스는 전역 엘리트를 개체군에 포함시켜 유전자알고리즘을 수행한다.

4. 시스템의 동작

제안한 모델의 동작은 기본적으로 클라이언트/서버 모델을 따른다. (1) 서버는 클라이언트의 접속을 대기하고 클라이언트의 접속 시점에서 초기 개체군을 생성하여 클라이언트에 할당한다. (2) 각 클라이언트들은 서버로부터 할당받은 개체군을 모집단으로 하여 유전자 알고리즘을 수행하고 이주주기에 맞춰서 지역 엘리트를 서버에 전송한다. (3) 서버는 현재 자신이 유지하고 있는 전역 엘리트와 각 클라이언트로부터 전송된 지역 엘리트들을 모집단으로 하여 유희시간에 유전자 알고리즘을 수행한다. (4) 새로 생성된 전역 엘리트를 클라이언트의 유전자 알고리즘에 사용할 수 있도록 전송한다. (5) 종료조건에 이를 때까지 (2)~(4)의 과정을 반복 수행한다. (6) 각 클라이언트는 종료조건에 도달하면 서버에 종료를 알리고 유전자 알고리즘 수행을 종료한다. 서버는 모든 후 클라이언트의 종료를 통보받은 후 수행을 종료한다.

5. 성능향상

제안한 모델에서 서버의 영향은 다음 그림 6과 같이 i번째 이주에서 (i+1)번째 이주 사이의 시간간격동안 서버에서 수행되는 세대수까지 개선된 적합도의 함으로 표현될 수 있다.

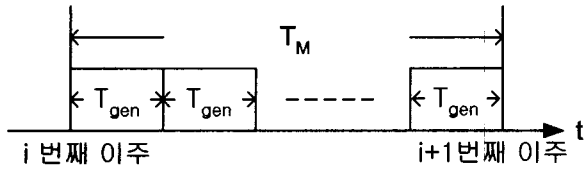


그림 6. 서버에서의 이주간 처리

Fig. 6. Process during the interval of migration in server.

여기서 지역 엘리트의 이주주기를  $T_M$ , 서버의 한 세대 처리시간을  $T_{gen}$ 이라고 하면 서버에서 개선되는 세대수  $G$ 는  $T_M$ 시간동안  $T_{gen}$ 이 몇 회 수행되는 지를 나타낸다. 즉, 다음과 같다.

$$G = \frac{T_M}{T_{gen}} \quad (2)$$

식 (2)에서 한 세대 처리시간  $T_{gen}$ 은 다음과 같다.

$$T_{gen} = (N+1) \times (P_c \times T_c + P_m \times T_m + T_f) + T_s \quad (3)$$

여기서  $N$ 은 클라이언트의 수,  $P_c$ 는 교차율,  $T_c$ 는 교차연산 수행시간,  $P_m$ 은 돌연변이율,  $T_m$ 은 돌연변이연산 수행시간,  $T_f$ 는 평가함수 수행시간,  $T_s$ 는 선택연산 수행시간을 나타낸다. 식 (2), (3)에 의해  $G$ 는 다음과 같이 나타낼 수 있다.

$$G = \left\lfloor \frac{T_M}{(N+1) \times (P_c \times T_c + P_m \times T_m + T_f) + T_s} \right\rfloor \quad (4)$$

세대간의 최대 적합도의 차  $Gengap(g)$ 는 다음과 같다.<sup>[11]</sup>

$$Gengap(g) = F_{max}(g) - F_{max}(g-1) \quad (5)$$

여기서  $F_{max}(g)$ 과  $F_{max}(g-1)$ 는 각각  $g$ 번째 세대의 최대적합도와  $(g-1)$ 번째 세대의 최대적합도를 나타낸다. 그러므로 이주시 서버에서의 적합도 개선  $F_{IS}$ 는 다음 식과 같다.

$$\begin{aligned} F_{IS} &= \sum_{g=1}^G Gengap(g) \\ &= \sum_{g=1}^G \{ F_{max}(g) - F_{max}(g-1) \} \end{aligned} \quad (6)$$

서버의 유전자알고리즘은 개체군의 크기가 (클라이언트의 수 + 1)이므로 클라이언트의 유전자알고리즘에 비해 개체군의 크기가 작고 고속으로 수행될 수 있다.

## IV. 성능평가 및 고찰

본 장에서는 제안한 모델의 성능향상과 파라미터의 영향을 실험을 통해 검증하였다. 네트워크기반 병렬유전자알고리즘 모델을 처음 제시한 Kojima의 방법과 클라이언트의 수에 따른 적합도를 비교하여 서버의 계산능력 활용을 통해 성능이 향상됨을 보인다.

### 1. 평가함수의 계산시간에 따른 성능

평가함수의 계산시간이 다른 블랙박스 문제와 배낭 문제(Knapsack Problem)를 본 논문에서의 방법과 기존의 방법에 각각 적용하여 평가함수의 복잡도에 따른 영향을 확인하였다.

#### 1.1 실험조건

본 논문에서는 평가함수에서 다른 파라미터의 영향을 최소화하기 위해서 가장 단순한 형태인 이진스트링 염색체표현을 사용한 단순교차연산자, 전형적인 돌연변이연산자, 속도가 빠른 엘리트보존 선택연산자를 사용하여 100회 실험하여 분석하였다.

#### 1.2 블랙박스 문제에 적용

블랙박스 문제는 32비트의 비트열에 대한 검색문제로 유전자알고리즘을 적용하는 기본적인 응용이다. 이 문제에서 적합도는 임의의 32비트에 대해 일치하는 비트수이다.

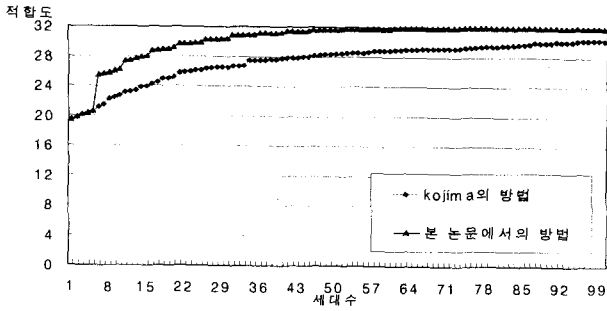
그림 7의 결과에서 (a)의 결과를 통해 전체적으로 본 논문의 방법이 수렴속도가 빠름을 확인할 수 있고 (b)의 결과를 통해 기존의 방법보다 전체적으로 적합도가 높음을 확인할 수 있다.

#### 1.3 배낭 문제에 적용

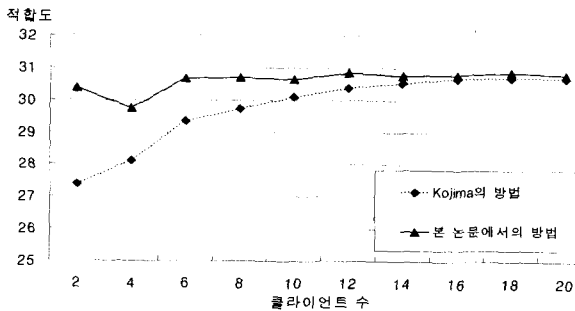
배낭 문제는 각각의 물체가 다른 무게와 가치를 가지는 복수개의 물체가 주어진 경우에, 무게가 어느 범위 이내에서 몇 개의 물체를 선택하고, 그때 가치의 합계가 최대가 되도록 선택방법을 찾는 문제이다. 본 논문에서는 배낭문제에서의 유효성을 보이는 것을 위주로 하므로 배낭문제 중 간단한 형태인 각 물체의 무게와 가치를 각각 30 이하에서 임의로 선택하고 전체 무게 40 이하에서 전체 가치의 합이 최대가 되도록 하는 문제를 대상으로 하였다. 이 때 적합도는 선택된 물체들의 전체 가치의 합이다.

그림 8의 결과에서도 그림 7의 결과와 같이 기존의 방법보다 수렴속도와 적합도가 개선됨을 확인할 수 있다.

두 실험에서 오프라인 성능은 표 1과 같다. 이 때 오프라인 성능은 블랙박스 문제에서는 32비트 중 일치하

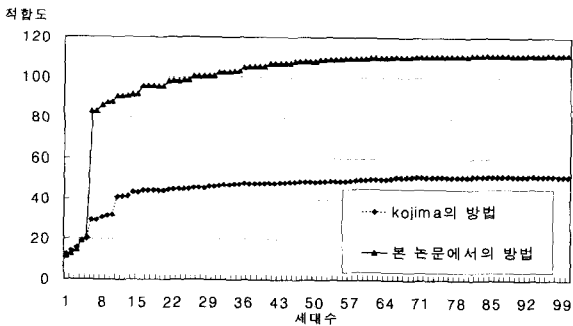


(a) 세대수에 따른 적합도 비교  
(a) The comparison of fitness values by the generation.

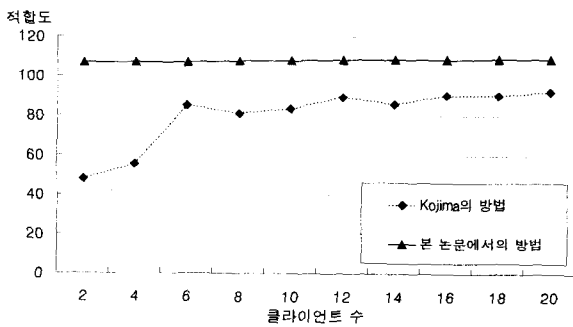


(b) 클라이언트 수에 따른 적합도 비교  
(b) The comparison of fitness value by the number of clients.

그림 7. 블랙박스 문제에서의 결과  
Fig. 7. Results of blackbox problem.



(a) 세대수에 따른 적합도 비교  
(a) The comparison of fitness values by the generation.



(b) 클라이언트 수에 따른 적합도 비교  
(b) The comparison of fitness value by the number of clients.

그림 8. 배낭 문제에서의 결과  
Fig. 8. Results of knapsack problem.

표 1. 각 문제에 대한 오프라인 성능  
(A: 기존의 방법, B: 본 논문에서의 방법)  
Table 1. The offline performance of each problem.  
(A: existing method, B: proposed method)

적용된 클라이언트 수	블랙박스 문제		배낭 문제	
	A	B	A	B
2	27.4	31.1	46.3	102.8
4	28.5	30.7	59.9	109.6
6	30.3	31.0	91.6	109.3
8	30.8	31.2	88.7	109.2
10	31.1	31.1	91.3	110.2
12	31.3	31.3	101.9	110.3
14	31.3	31.2	98.5	110.3
16	31.3	31.2	102.7	110.3
18	31.2	31.2	100.2	110.6
20	31.2	31.2	100.8	108.4

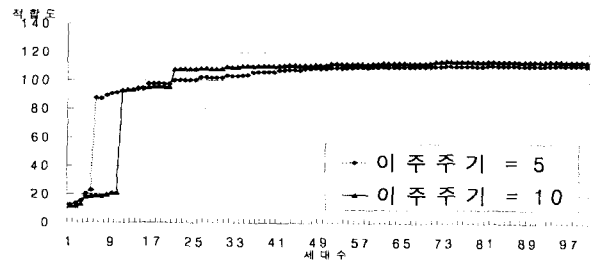


그림 9. 이주주기에 따른 적합도 비교  
Fig. 9. The comparison of fitness value by the interval of chromosome migration

는 비트수 단위이고 배낭문제에서는 선택된 가치의 합으로 표현된다.

## 2. 고찰

위의 실험들을 통해 본 논문의 방법을 적용함으로써 클라이언트의 수가 증가함에 따라 그 영향은 줄어들지만 기존의 방법보다 최적해의 수렴속도가 향상됨을 확인할 수 있었다. 또, 본 논문의 방법을 적용하면 서버에 버퍼와 엘리트 풀이 추가되어 필요 메모리량이 증가한다. 클라이언트의 수  $N$ , 염색체 크기  $C$ , 엘리트 풀의 추가메모리  $M_{AE}$ , 버퍼의 추가메모리  $M_{AB}$ 라고 할 때 추가되는 메모리량  $M_A$ 는 다음과 같다.

$$\begin{aligned}
 M_A &= M_{AE} + M_{AB} \\
 &= C + (N+1) \times C \\
 &= (N+2) \times C
 \end{aligned}
 \tag{7}$$

그러나 메모리량은 클라이언트 수가 커지더라도 추가된 클라이언트 수와 염색체의 크기의 곱만큼만 증가한다. 이는 전체 개체군의 크기에 비해서 매우 작다.

이주주기를 5세대에 한번 이주한 경우와 10세대에 한번 이주한 경우로 나누어 실험한 결과 그림 9와 같이

이주주기가 커지면 이주회수가 줄어들고 지역 엘리트의 갱신회수도 줄어들어 최적해로의 수렴속도는 감소한다.

그러나 서버의 유휴시간이 증가하고 서버에서의 개선세대수가 증가하여 수렴속도 감소폭을 줄임을 확인할 수 있다.

### V. 결 론

네트워크기반 병렬유전자알고리즘의 성능 향상을 위해 서버의 계산능력을 활용하는 방법을 제안하였다. 서버에서 엘리트 평가만을 수행하여 전역 최적해를 구하는 기존의 방법과 달리 서버의 유휴시간에 유전자알고리즘을 수행하여 전역 엘리트의 적합도를 개선하고 이를 클라이언트의 유전자알고리즘 수행에 이용함으로써 전체 알고리즘의 최적해 수렴속도가 향상된다. 또한 버퍼와 엘리트 풀을 이용해서 비동기수행이 가능하여 이주대기시간이 필요치 않고 엘리트 이주주기의 고정에 따른 수렴속도 감소를 서버에서 보완하여 고정이주주기를 사용함으로써 통신량을 줄일 수 있다. 그러나, 클라이언트의 수가 증가하면 성능향상이 감소하므로 소규모 분산네트워크환경에 적합하다.

### 참 고 문 헌

[1] J.H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975

[2] H. Kitano, "Empirical Studies on the Speed of Convergence of the Neural Network Training by Genetic Algorithm", Proc. of AAI-90, 1990.

[3] L. Wang, A.A. Maciejewski, H.J. Siegel and V.P. Roychowdhury, "A Comparative Study of Five

Parallel Genetic Algorithms Using the Traveling Salesman Problem", 1998 IPPS/SPDP. Proc. of the First Merged International. 1998. p.345-349, 1998. 4.

[4] M. Nowostawski and R. Poli, "Parallel Genetic Algorithm Taxonomy", International Conference on Knowledge-Based Intelligent Information Engineering Systems, p.88-92, 1999. 8.

[5] M. Golub and D. Jakobovic, "A New Model Of Global Parallel Genetic Algorithm", Information Technology Interfaces ITI 2000, p.363-368, 2000. 1.

[6] E. Cantú-Paz, "Designing Efficient Master-Slave Parallel Genetic Algorithms", Genetic Programming: Proc. of the Third Annual conference, p.455-460, 1998

[7] L. Tan and K.A. Smith, "A New Parallel Genetic Algorithm", Proc. of the International Symposium on Parallel Architectures, Algorithm and Networks, IEEE, 2002,

[8] K. Kojima, W. Kawamata, H. Matsuo and M. Ishigame, "Network based Parallel Genetic Algorithm using Client-Server Model", Proc. of CEC2000, p.244-249, 2000.

[9] K. Kojima, H. Matsuo and M. Ishigame, "Reduction of Communication Quantity for Network Based Parallel GA", Proc. of the CEC2002 Congress on , Volume: 2, p.1715-1720, 2002. 5.

[10] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, NY, 1989

[11] J. Cheng, W. Chen, L. Chen and Y. Ma, "The Improvement of Genetic Algorithm Searching Performance", Proc. of the First International Conference on Machine Learning and Cybernetics, Beijing, p.947-951, 2002. 11.

### 저 자 소 개

**송 봉 기**(정회원)  
 1997년 부경대학교 전자공학과 학사 졸업.  
 1999년 부경대학교 정보시스템 협동과정 정보공학 석사 졸업  
 2003년 부경대학교 정보시스템 협동과정 박사과정 수료.  
 <주관심분야: 유전자알고리즘, 네트워크, 마이크로프로세서 등>

**성 길 영**(정회원)  
 제39권 SC편 제3호 참조

**김 용 성**(정회원)  
 1987년 경북대학교 전자공학과 학사 졸업  
 1991년 경북대학교 컴퓨터공학과 석사 졸업  
 1998년 부경대학교 정보시스템 협동과정 박사과정 수료  
 현재 경남정보대학 컴퓨터정보계열 부교수  
 <주관심분야: 컴퓨터>

**우 종 호**(정회원)  
 제39권 SC편 제3호 참조