

경쟁 공진화 알고리즘에서 경쟁전략들의 비교 분석

김여근[†] · 김재윤

전남대학교 산업공학과

Comparison and Analysis of Competition Strategies in Competitive Coevolutionary Algorithms

Yeo Keun Kim · Jae Yun Kim

Department of Industrial Engineering, Chonnam National University, Kwangju, 500-757

A competitive coevolutionary algorithm is a probabilistic search method that imitates coevolution process through evolutionary arms race. The algorithm has been used to solve adversarial problems. In the algorithms, the selection of competitors is needed to evaluate the fitness of an individual. The goal of this study is to compare and analyze several competition strategies in terms of solution quality, convergence speed, balance between competitive coevolving species, population diversity, etc. With two types of test-bed problems, game problems and solution-test problems, extensive experiments are carried out. In the game problems, sampling strategies based on fitness have a risk of providing bad solutions due to evolutionary unbalance between species. On the other hand, in the solution-test problems, evolutionary unbalance does not appear in any strategies and the strategies using information about competition results are efficient in solution quality. The experimental results indicate that the tournament competition can progress an evolutionary arms race and then is successful from the viewpoint of evolutionary computation.

Keywords: coevolutionary algorithm, competitive coevolution, competition strategy, evolutionary arms race

1. 서론

경쟁 공진화 알고리즘은 생태계에서 진화적 무기경쟁(evolutionary arms race)을 통해 생물학적 종들이 공진화하는 현상을 모형화하여 개발된 확률적 탐색기법이다(Rosin, 1997; Nolfi, and Floreano, 1998). 이것은 자연계에서 숙주가 이에 기생하는 기생충을 이기기 위하여 진화하고, 기생충 또한 숙주가 진화하면 이에 적응하기 위하여 진화하는, 즉 종간에 경쟁하면서 공진화하는 과정을 모방한 것이다. 경쟁 공진화 알고리즘은 복잡도가 높고 해공간이 넓은 적대문제(adversarial problems)들을 효율적으로 해결하기 위한 방법론으로 사용되고 있다(Rosin, 1997). 적대문제는 해결하려는 문제의 후보해(candidate solutions)와 이에 대한 테스트 문제(test cases)가 경쟁적인 구조로

구성된 문제를 의미한다. 이러한 문제로는 게임문제, 여러 경우에 대해 테스트를 필요로 하는 문제(예로 신경망 문제), 그리고 제약만족 문제 등을 포함한다. 경쟁 공진화 알고리즘의 기본 개념은 이들 문제에 대한 후보해를 숙주로 테스트 문제를 기생충으로 두고, 이들 종들을 상호 경쟁적으로 공진화시켜 결과적으로 좋은 해를 얻는 데 있다.

진화계산(evolutionary computation)의 관점에서, 효율적인 경쟁 공진화 알고리즘을 구축하기 위해서는 경쟁하는 종(모집단)들이 지속적인 진화적 무기 경쟁에 의하여 상호간에 복잡도의 수준을 증가시키도록 유도하여야 한다(Rosin, 1997; Nolfi and Floreano, 1998). 어느 한 종이 일방적으로 우세하게 되면, 상대의 진화를 유도할 수 있는 무기경쟁이 깨져서 결국 상호 진화가 멈추게 된다. 이는 알고리즘에서 조기수렴의 결과를 초래할 수 있다. 따라서 숙주와 기생충 종간에 경쟁을 통해 상

본 연구는 한국과학재단 특정기초연구(과제번호: 98-0200-09-01-3) 지원으로 수행되었음.

[†] 연락저자: 김여근 교수, 500-757 광주광역시 북구 용봉동 300번지 전남대학교 산업공학과, Fax: 062-530-1789, e-mail: kimyk@chonnam.ac.kr
2000년 9월 접수, 1회 수정(3주 소요) 후, 2001년 12월 게재 확정.

호간에 점진적으로 개선되어야 한다(Rosin, 1997). 즉, 이들 중간에 진화의 균형을 유지하여야 한다. 이와 관련된 여러 연구가 행하여 졌다(Paredis, 1995; Rosin, 1997; Nolfi and Floreano, 1998; Pagie and Hogeweg, 1998; Kim et al., 2000).

본 연구에서는 경쟁 공진화 알고리즘에서 경쟁상대의 선택 전략(이를 '경쟁전략'이라 부르기로 함)들을 진화계산의 관점에서 비교 분석한다. 표준 진화 알고리즘에서 적응도(fitness)는 외부의 고정된 환경, 즉 주어진 평가함수에 의해 흔히 평가된다. 그러나 경쟁 공진화 알고리즘에서 개체의 적응도는 다른 모집단(종)에 있는 개체들과 경쟁에 의하여 평가된다. 이때 개체의 적응도는 어떤 경쟁자와 경쟁하느냐에 따라, 즉 경쟁 전략에 따라 변한다. 각 개체의 적응도 변화는 알고리즘에서의 탐색방향과 진화속도를 변화시킬 수 있으며, 생존을 위한 선별(selection)에 영향을 주어 지속적인 무기경쟁을 유도하지 못할 수 있다. 또한 경쟁전략에 따라 알고리즘의 계산소요 시간에 차이가 발생한다. 이러한 이유로 경쟁 공진화 알고리즘의 설계시 개체의 적응도를 결정하는 경쟁전략은 중요한 문제이다.

경쟁전략에 관한 연구로 Koza(1992)는 상대 모집단 전체, Pagie and Hogeweg(1998)는 상대 모집단의 이웃, Kim et al.(2000)은 토너먼트경쟁, 그리고 Rosin(1997)은 샘플링한 일부 개체들과 경쟁하는 방법을 제안하였다. 이와 같이 경쟁 공진화 알고리즘에서 개체의 적응도 평가를 위하여 여러 경쟁 전략들을 채용하고 있으나, 경쟁전략들에 대한 특성과 그 효과에 대한 연구는 아직까지 이루어지지 않고 있다. 본 연구에서는 먼저 경쟁 공진화 알고리즘이 적용될 수 있는 적대문제들을 분류한다. 그리고 문제의 유형에 따라 경쟁전략의 특성을 비교 분석한다. 분석하는 특성으로는 해의 질, 진화속도(수렴 속도), 진화의 균형성, 모집단의 다양성, 계산소요시간 등을 포함한다.

2. 경쟁 공진화 알고리즘과 비교 경쟁전략

먼저, 경쟁전략을 비교 분석하는 데 기본 틀로서 경쟁 공진화 알고리즘이 요구된다. 본 연구에서는 일반적 모형(Kim et al., 2000)으로 다음 알고리즘을 사용한다.

단계 1 (초기화)

숙주 개체와 기생충 개체로 이루어진 두 모집단, $Pop-H$ 와 $Pop-P$ 를 각각 구성한다.

단계 2 (경쟁전략)

경쟁전략에 따라 경쟁자들을 선택한다.

단계 3 (적응도 평가)

경쟁자들과의 경쟁에 의해 개체의 적응도를 평가한다.

단계 4 (모집단 진화)

4.1 $EP = Pop-H$, $F=0$ 으로 둔다.

4.2 EP 에서 단계 3의 결과에 의해 가장 높은 적응도를 보유한 개체를 선택하여 보관한다.

4.3 EP 에서 적응도에 따라 재생산에 참여할 개체를 선별한다.

4.4 단계 4.3에서 선별된 개체(부모개체)에 대하여 유전 연산(교차와 돌연변이)하여 자손개체를 생산한다. EP 에 있는 부모개체는 생산된 자손개체로 대체된다.

4.5 단계 4.2에서 선택된 개체를 EP 에 있는 임의의 한 개체와 대체한다.

4.6 만약 $F=0$ 이면, $EP = Pop-P$, $F=1$ 로 두고 단계 4.2로 가고, 그렇지 않으면 단계 5로 간다.

단계 5 (종료조건 검사)

알고리즘 종료조건이 만족하면 끝내고, 그렇지 않으면 단계 2로 간다.

단계 3은 단계 2의 경쟁전략에 의해 선택된 경쟁자와 경쟁하여 각 개체의 적응도를 평가하는 단계이다. 단계 4에서는 숙주 모집단과 기생충 모집단이 번갈아가며 진화한다. 단계 4.2와 4.5를 통해 이루어지는 가장 높은 적응도를 갖는 개체의 보존은 최적화의 관점에서 좋은 해로의 탐색을 유도하기 위한 것이다. 그러나 이것은 표준 진화 알고리즘에서의 우수개체 보존전략(elitism)과 달리 현 세대의 가장 좋은 해는 다음 세대의 진화에만 영향을 주도록 하였다. 두 모집단의 진화를 위해 사용되는 선별, 유전연산, 유전파라미터, 그리고 실험조건 등은 3.3절에서 제시한다.

단계 2에서 경쟁자를 선택할 때 경쟁전략이 요구된다. 앞서도 언급하였듯이, 경쟁 공진화 알고리즘에서는 경쟁전략에 따라 개체의 적응도가 변화된다. 본 연구에서는 다음 여섯 가지 경쟁전략을 진화계산의 관점에서 비교 분석한다.

(1) TTC(ToTal Competition)

상대 모집단에 있는 모든 개체와 경쟁한다(Koza, 1992).

(2) RSC(Random Sampling Competition)

상대 모집단에서 임의로 추출한 일부 개체와 경쟁한다.

(3) FSC(Fitness based Sampling Competition)

상대 모집단에서 각 개체가 갖는 경쟁에서의 승리횟수를 적응도로 하여, 적응도를 기준으로 추출한 일부 개체와 경쟁한다.

(4) SSC(Shared Sampling Competition)

Rosin(1997)에 의해 제안된 이 방법은 배분 적응도(shared fitness)를 기준으로 추출한 일부 개체와 경쟁한다. 숙주 개체 i 의 배분 적응도는 다음과 같이 계산한다.

$$f_i = \sum_{j \in X} \frac{1}{N_j}$$

여기서 X 는 숙주 개체 i 가 승리한 기생충 개체의 집합을 의미한다. N_j 는 기생충 개체 j 에 대해 승리한 숙주 개

체의 개수이다. 기생충 개체의 배분 적용도도 같은 방법으로 계산된다.

(5) SHC(Shared sampling and Hall of fame Competition)

경쟁자의 일부는 SSC에 의하여 추출하고, 다른 일부는 과거에 발견된 높은 적용도를 갖는 개체들, 즉 우수개체 집합[hall of fame(Rosin, 1997)]에서 추출하여 이들과 경쟁한다. 우수개체집합은 매 세대마다 가장 높은 적용도를 갖는 개체들로 구성된다.

(6) TMC(Tourna Ment Competition)

한 모집단의 개체들은 상대 모집단의 그것들과 게임에서 토너먼트 경기방식과 유사하게 임의로 짝을 지어 경쟁한다(Kim et al., 2000). TMC의 절차는 다음과 같다. 숙주와 기생충 모집단에서 생존해 있는 숙주와 기생충을 임의로 짝지어 경쟁시킨다. 여기서 짝지어지지 않는 숙주 또는 기생충은 경쟁에 참여하지 않으나 생존해 있는 것으로 본다. 경쟁에서 패배한 숙주와 기생충을 제거한다. 숙주와 기생충 모집단에서 어느 한 모집단에 생존한 개체가 없으면 끝내고, 그렇지 않으면 생존한 숙주와 기생충은 임의로 짝지어 경쟁한다. 이 과정을 경쟁이 종료될 때까지 반복한다. 여기서 초기에 두 모집단의 모든 개체는 생존해 있는 것으로 보고, TMC의 사용시에는 위의 경쟁 공진화 알고리즘에서 단계 2와 3이 동시에 행하여진다.

3. 문제유형의 분류와 실험문제

경쟁 공진화 알고리즘이 적용될 수 있는 적대문제는 ① 게임문제(Rosin, 1997; Nolfi and Floreano, 1998)와 ② 해-테스트 문제(Hillis, 1991; Paredis, 1995)로 분류해 볼 수 있다. 게임문제에서는 참가자들이 각기 자신의 전략을 가지고 상대자와 경쟁을 한다. 상대의 어떤 전략에도 이길 수 있는 전략을 찾는 것을 목표로 한다. 참가자가 사용하는 게임전략(숙주)의 적용도는 승리와 패배 횟수에 의해 평가된다. 따라서 적용도는 상대자의 게임전략(기생충)에 종속된다. 해-테스트 문제는 사전(事前)에 주어진 기준을 만족하는 해를 탐색하는 문제이다. 분류를 위한 신경망 문제나 제약만족 문제들이 이에 속한다. 흔히 이들 문제에서 테스트를 요하는 자료의 규모가 아주 크다. 따라서 후보해(숙주)는 테스트문제(기생충) 공간의 아주 작은 부분 테스트집합에 의해 평가된다. 이때 후보해는 이 부분집합과 경쟁을 통해 주어진 기준의 만족도에 의해 평가된다. 문제의 유형에 따라 그 특성이 다름에도 불구하고 경쟁 공진화 알고리즘에 관한 기존 연구들에서는 문제 유형이 분류되지 않아 알고리즘의 적용에 혼동을 주었다.

본 연구에서는 경쟁전략을 분석하는 데 있어서 앞에서 언급한 두 유형의 문제를 분리하여 다룬다. 게임문제로는 2DTTT(Tic-Tac-Toe) 게임문제(Rosin, 1997)와 Nim 게임문제(Rosin,

1997)를, 해-테스트 문제로는 분류를 위한 신경망(neural network) 문제(Paredis, 1995)와 정렬망(sorting network) 문제(Hillis, 1991; Juillé, 1995; Olsson, 1996)를 사용한다.

3.1 게임문제

3.1.1 2DTTT 게임문제와 개체표현

2DTTT 문제는 두 참가자가 3×3의 격자에 ○, × 표시를 하며, 가로, 세로, 대각선으로 3개 연속하여 자신의 표시를 먼저 얻는 사람이 승리하는 게임이다. 2DTTT 게임은 두 참가자의 무승부가 나올 수 있다는 특징을 갖는다.

각 참가자의 전략을 개체로 표현하기 위하여 첫 번째 참가자의 전략을 숙주개체로 표현하고, 두 번째 참가자의 전략을 기생충개체로 표현하였다. 두 참가자의 전략 표현은 동일하며, 단지 평가 순서만 서로 다르다. 개체에서 각 인자는 전략을 취한 후 3×3의 격자에 표시되어 있는 ○, ×의 상태를 의미한다. 격자는 방향성과 순서를 갖지 않으므로, 한 격자의 상태는 7개의 대칭적인 경우를 갖는다(Kim et al., 1997, Page 101). 격자의 모든 가능한 상태는 대칭적인 경우를 제외하면 593가지이므로, 개체는 593비트로 표현된다(Rosin, 1997).

인자의 위치는 현재의 상태에서 ○, × 표시가 되어 있지 않은 위치의 수를 나타낸다. 평가는 다음과 같은 과정을 통해 이루어진다. 먼저, 숙주개체가 현재 격자의 상태를 나타내는 인자위치 값에 따라 자신의 표시를 한다. 그리고 숙주개체의 취한 행동에 맞는 상태를 나타내는 기생충개체의 인자위치를 찾아 그 값에 따라 표시를 해 나감으로써 두 개체의 승패를 결정한다. 2DTTT 게임문제의 정의와 개체표현에 구체적인 내용은 Rosin의 연구(1997)를 참조할 수 있다.

3.1.2 Nim 게임문제와 개체표현

Nim 게임은 하나의 돌더미는 여러 개의 돌로 구성되어 있고, 돌더미가 다수 개 있다고 할 때 두 참가자는 번갈아가면서 돌더미 중 하나에서 1개 이상의 돌을 가져간다. 게임의 승자는 마지막 남은 돌을 가져간 참가자가 된다. 본 연구에서는 각각 3, 4, 5, 4개의 돌로 구성된 4개의 돌더미로부터 게임을 시작한다고 본다. Nim 게임은 매회 참가자가 취할 수 있는 경우의 수가 다양하여 2DTTT 게임보다 문제의 복잡성이 상대적으로 높다.

2DTTT 게임에서와 마찬가지로, 첫 번째 참가자의 전략을 숙주개체로 표현하고, 두 번째 참가자의 전략을 기생충개체로 표현하였다. 개체에서 각 인자의 위치는 참가자가 한번의 행동을 취한 후 4개의 돌더미들의 상태, 즉 각 돌더미에 있는 돌의 남은 개수를 뜻한다. 따라서 모든 가능한 경우의 수는 $599(= 4 \times 5 \times 6 \times 5 - 1)$ 가지이므로 개체는 599비트로 표현된다(Rosin, 1997). 총 경우의 수에서 1을 감한 이유는 행동을 취한 후의 돌의 상태를 나타내므로, 초기 상태, 즉 (3, 4, 5, 4)의 상태는 발생되지 않기 때문이다. 개체의 각 인자는 0 또는 1로 표현된다. 현재 하나 이상의 돌이 남아 있는 돌더미들 중에서

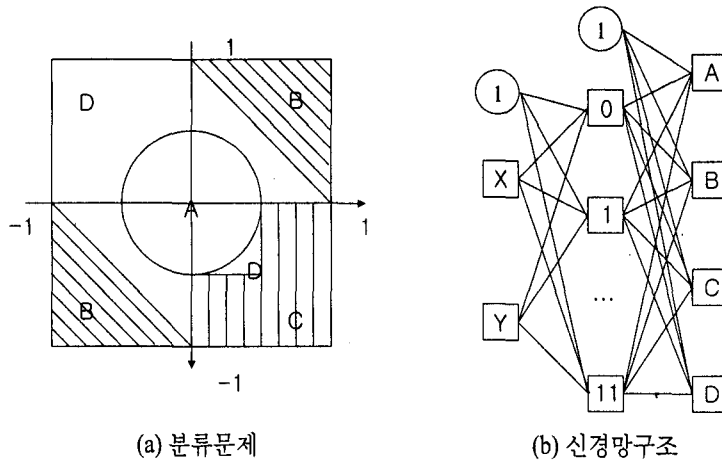


그림 1. 분류를 위한 신경망 문제.

차례로 하나씩 선택하고, 선택된 돌더미에 k 개의 돌이 있다고 하자. 그러면 이 돌더미에서 0부터 $(k-1)$ 개의 돌이 남아있는 상태를 의미하는 인자를 순서대로 조사하여 처음으로 1이 나타나는 인자의 상태를 다음 행동으로 취한다. 만약 조사했던 어떠한 인자에서 1이 발견되지 않으면, 평가했던 것 중에서 첫 번째 행동을 취한다. 이와 같은 과정을 반복하여, 숙주와 기생충 개체들이 사용한 전략에 의해 두 개체가 모든 행위를 취한 후의 상황으로 승패를 결정한다. Nim 게임문제의 정의와 개체 표현에 구체적인 내용은 Rosin의 연구(1997)를 참조할 수 있다.

3.2 해-테스트 문제

3.2.1 신경망 문제와 개체 표현

분류를 위한 신경망 문제는 <그림 1>의 (a)와 같이 두 좌표를 입력으로 하여 4종류 {A, B, C, D}의 결과를 정확히 분류하는 신경망의 적절한 연결 가중치를 찾는 문제이다. 신경망의 구조는 <그림 1>의 (b)와 같이 1개의 은닉층(hidden layer)과 2개의 편의(bias)를 가지며, 은닉층의 은닉노드(hidden node) 수는 12개로 가정한다(Paredis, 1995).

숙주개체는 신경망의 가중치를 표현하고, 기생충개체는 분류해야 할 (x, y) 좌표를 표현한다. 숙주개체와 기생충개체는 모두 이진수로 표현한다. 신경망의 가중치는 $[0, 1]$ 구간 내의 실수값이므로, 10비트로 표현하여 소수점 이하 3자리의 수로 표현될 수 있도록 하였다. 따라서 연결 가중치가 총 88개 있으므로, 숙주개체의 길이는 880비트로 표현하였다. 기생충개체에서 (x, y) 의 좌표는 $[-1, 1]$ 사이의 실수값이며, 소수점 이하 4자리까지 표현하기로 하면 각 좌표마다 15비트가 요구된다. 따라서 기생충개체의 길이는 30비트로 표현하였다.

적응도 평가는 다음의 과정을 따른다. 경쟁상대로 선택된 숙주개체와 기생충개체가 경쟁하면, 기생충개체의 (x, y) 좌표를 입력으로 하여 숙주개체에 의한 신경망의 출력값을 계산한다. 그리고 가장 큰 출력값을 나타내는 출력노드를 분류 결과

로 하였다. 신경망에서 사용한 시그모이드(sigmoid) 함수는 $\sigma(x) = 1/(1 + e^{-x})$ 이다.

3.2.2 정렬망 문제와 개체 표현

정렬망 문제는 임의로 나열된 숫자열에서 두 숫자의 비교-교환을 통해 오름차순(또는 내림차순)으로 정렬하는 망을 찾는 문제이다. <그림 2>는 구성요소 개수, $n = 6$ 인 숫자열을 오름차순으로 정렬하는 망을 나타내고 있다. 각 수평선은 목록에 있는 한 구성요소를 나타내고, 각 수직선은 두 구성요소들의 비교를 나타낸다. 이 정렬망은 왼쪽에서 오른쪽으로 진행하면서 비교된 두 구성요소의 숫자가 오름차순으로 되어있지 않으면 순서를 교환하여 정렬하는 것을 나타낸다. 정렬망 문제의 목적은 최소의 비교-교환으로 올바르게 숫자를 나열할 수 있는 정렬망을 찾는 것이다. 본 연구에서는 $n = 7, 10, 13$ 인 숫자열을 실험문제로 사용한다.

정렬망을 숙주개체로 표현하고, 숙주개체의 정렬 능력을 테스트하기 위한 숫자열을 기생충개체로 표현하였다. 숙주개체의 길이는 지금까지 알려진 최적 비교-교환횟수로 하였다(Juill 6, 1995). 이들 횟수는 7, 10, 13개의 숫자열을 갖는 정렬망에서 16, 29, 46이므로, 이들 문제에서 숙주개체 길이는 각각 이 횟수와 같게 두었다. 기생충개체의 길이는 각 문제의 숫자열 크기와 동일하다. <그림 2>에서 정렬망 위쪽에 표현된 문자열은 하나의 숙주개체를 나타내고, 정렬망 좌측에 표현된 숫자열은

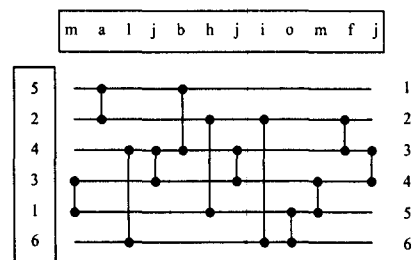


그림 2. 정렬망 문제.

표 1. 유전연산자 및 파라미터

문 제	교차(교차율)	돌연변이(돌연변이율*)	모집단 크기	종료조건
2DTTT 게임	이점교차(0.7)	선택가능한 인자값 범위 내에서 임의선택($r_i = 0.1, r_g = 0.4$)	100 (= 10×10)	5,000
Nim 게임	이점교차(0.7)	대립형질 돌연변이 ($r_i = 0.1, r_g = 0.4$)	625 (= 25×25)	10,000
신 경 망	이점교차(0.7)	대립형질 돌연변이 ($r_i = 0.1, r_g = 0.4$)	100 (= 10×10)	5,000
정렬망	숙 주	이점교차(0.7)	100 (= 10×10)	10,000($n = 7$) 15,000($n = 10$) 20,000($n = 13$)
	기생충	순서교차(0.7)		

* r_i = 개체단위 돌연변이율, r_g = 인자단위 돌연변이율

하나의 기생충개체를 나타낸다. 숙주개체의 첫 번째 인자, m 은 기생충개체의 4번째와 5번째 숫자쌍 (3, 1)을 비교하여 오름차순으로, 즉 (1, 3)으로 위치를 교환한다. 이 과정을 숙주개체의 좌측에서 우측으로 반복한다.

3.3 실험 파라미터

실험 분석을 위한 파라미터들은 예비실험을 통하여 결정하였다. <표 1>은 실험에 사용한 유전연산자와 파라미터들을 나타낸 것이다. 각 문제에서 유전연산자는 개체 표현에 적합한 것을 선택하였다. 예를 들어, 정렬망 문제의 기생충개체는 숫자열을 나타내므로, 순서문제에 사용되는 순서교차(Davis, 1985)와 역순돌연변이(Fox and McMahon, 1991)를 사용하였다. 모집단 크기와 종료조건은 각 문제의 해공간 크기에 따라 달리 두었고, 종료조건은 세대수로 하였다. 알고리즘은 C++ 프로그램 언어로 구현되었으며, 모든 실험은 300MHz Pentium CPU를 장착한 IBM-PC에서 수행되었다.

비교 분석의 대상이 되는 6가지의 경쟁전략 중에서 샘플링 전략(RSC, FSC, SSC, SHC)은 샘플크기가 요구된다. 샘플크기는 모두 모집단 크기의 10%로 하였다. 단, SHC에서는 모집단 크기의 5%는 SSC에 의해 선택하고 나머지 5%는 우수개체집합에서 선택하여, 전체 샘플크기는 다른 샘플링전략과 동일하게 하였다. 마지막으로, 단계 4.3에서 사용되는 선별은 토너먼트 크기가 2인 토너먼트 선별을 사용하였다.

한편, 본 연구에서는 진화하는 기생충 모집단 이외에 기생충 테스트집합을 두어 최적화의 관점에서 숙주의 절대적인 진화 정도를 판별하였다. 공진화하는 숙주와 기생충 두 모집단의 상호경쟁에 의한 개체의 적응도 평가는 상대적 척도라고 볼 수 있다. 테스트 집합은 예비실험을 통해 비교적 강한 기생충들로 구성하였으며, 실험하는 동안 진화하지 않고 고정되어 있다. 따라서 테스트집합과의 경쟁 결과는 절대적 진화 척도로 사용될 수 있다. <표 2>는 각 문제에 대한 테스트집합을 보

표 2. 문제별 테스트집합

문 제	테스트집합	
2DTTT 게임	전체경쟁(TTC)으로 구한 승리하기 어려운 게임전략 500개	
Nim 게임	전체경쟁(TTC)으로 구한 승리하기 어려운 게임전략 1,000개	
신 경 망	분류경계에 근접하여 분류하기 어려운 (x, y) 좌표 200개	
정렬망	$n = 7$	7개의 숫자로 나열가능한 모든 숫자열 5,040(= 7!)개
	$n = 10, 13$	비정렬도가 50% 이상인 임의 숫자열 5,000개

인 것이다. 여기서 정렬망 문제의 비정렬도는 길이 n 의 숫자열에서 총가능한 비교쌍 nC_2 에 대해 오름차순으로 정렬되어 있지 않은 교환쌍의 비율로 정의한다.

실험에서 개체의 적응도는 상대 개체들과의 경쟁에서 승리한 횟수로 한다. 단, SSC와 SHC의 적응도는 숙주와 기생충간의 승패관계에 의한 배분 적응도로 한다. 2DTTT와 Nim 게임 문제에서는 경쟁자들이 사용한 전략에 의해 승리와 패배를 결정한다. 단, 2DTTT 게임문제에서 무승부 상황이 발생하면 숙주와 기생충의 승패횟수에 영향을 주지 않은 것으로 하였다. 신경망 문제에서는 신경망에서 가장 큰 출력값을 갖는 출력노드를 분류 결과로 하였다. 옳게 분류하면 숙주의 승리, 그렇지 않으면 기생충의 승리로 둔다. 마지막으로, 정렬망 문제에서는 기생충개체(숫자열)의 비정렬도에 대한 숙주(정렬망)에 의한 정렬후 비정렬도의 비율이 40% 이하로 낮아지면 숙주의 승리, 그렇지 않으면 기생충의 승리로 보았다. 그러나 숙주개체의 절대적 진화 정도를 판단하는 데 사용되는 테스트집합에 있는 기생충개체와 숙주와의 경쟁에서는 정렬망(숙주)이 숫자열(기생충)을 완벽하게 정렬하는 경우에만 숙주가 승리한 것으로 하였다.

표 3. Nim 게임에서 경쟁전략별 성능 비교

분석요소		TTC	RSC	FSC	SSC	SHC	TMC
승률 (%)	Best	100.00	100.00	100.00	100.00	100.00	100.00
	Worst	98.30	90.70	21.50	20.40	63.90	95.70
	Mean	99.04	98.07	71.24	76.95	92.79	98.81
	Std	0.617	2.770	34.55	37.17	10.86	1.496
완전해발견횟수		3	2	5	6	3	3
평가횟수		1.84E+09	6.82E+08	6.82E+08	6.82E+08	6.82E+08	3.65E+06
계산시간(초)		12,852	2,497	3,051	5,289	6,844	901

4. 게임문제에서의 비교 분석

본 장과 다음 장에서 각각 게임문제와 해-테스트 문제에 대해 경쟁전략을 비교 분석한다. 분석은 경쟁전략에 따른 해의 질, 진화속도, 진화의 균형성, 그리고 모집단의 다양성 유지 측면에 관해 이루어졌다.

4.1 해의 질

<표 3>은 Nim 게임에서 세대수에 따른 경쟁전략별 해의 질을 비교한 것으로, 10회 반복실험의 결과이다. 이 표에서 '승률'이란 알고리즘을 종료한 후 숙주 모집단에 있는 개체들이 테스트집합과 경쟁하여 얻은 승률의 가장 높은 값을 의미한다. Best와 Worst는 각각 반복실험 동안의 최고 승률과 최저 승률이고, Mean과 Std는 승률의 평균과 표준편차이다. '완전해발견횟수'는 반복실험에서 테스트문제를 모두 승리한 개체, 즉 승률이 100%인 개체를 발견한 실험의 횟수이다. 평가횟수는 개체가 경쟁을 통해 적응도를 평가받은 횟수이며, 평가횟수와 계산시간은 반복실험의 평균값이다.

실험 결과, 평균 승률 측면에서 TTC, TMC, RSC가 좋은 결과를 보이지만, 완전해의 발견횟수 측면에서는 정보(경쟁의 승리횟수, 배분적응도)를 이용한 샘플경쟁인 FSC, SSC가 더 나은 결과를 보였다.

TTC는 많은 평가횟수와 계산시간을 요구하지만, 성능의 결과는 만족스럽지 못하였다. TTC는 RSC나 TMC와 비교할 때 성능에 있어 크게 차이가 있는 것으로 보이지는 않으며, 계산시간이 많이 소요된다는 단점을 갖는다. RSC가 TTC와 비슷한 수준의 성능을 보인 것은 경쟁 공진화 알고리즘에서는 상대개체도 진화하며, 매 세대마다 이전 세대에서 갖는 진화정보인 승리횟수에 무관하게 적응도를 평가하기 때문인 것으로 보인다. 이는 또한 통계학에서 샘플링 이론과 그 개념이 일치한 것으로 판단된다. TMC는 다른 전략과 비교하여 만족스러운 결과를 보이면서도 평가횟수와 계산시간 측면에서는 우수하였다.

FSC와 SSC는 완전해를 발견하는 횟수는 많으나, 해의 편차가 심하여 평균 승률 측면에서 좋지 않았다. 이는 개체들이 아

직 높은 경쟁력을 갖지 못한 상태에서 승리하기 어려운 상대개체가 많이 샘플링되어 이로부터 테스트되면 대부분 낮은 적응도를 부여받게 된다. SHC는 SSC전략에 비해 평균 승률이 높고 표준편차가 낮다. 이는 우수개체집합의 사용으로 SSC가 갖는 단점을 어느 정도 극복한 것으로 보인다. 경쟁전략들이 위와 같은 결과를 보인 세부과정은 4.3절의 '진화의 균형성'에서 다시 설명하기로 한다.

4.2 진화속도

진화속도는 절대적 진화 정도에 의해 측정하는 것이 타당할 것이다. <그림 3>은 경쟁 전략별 절대 진화 정도를 보인 것이다. 그림에서 Y축은 테스트집합과 경쟁하여 얻은 숙주의 최고 승률을 의미한다. <그림 3(a)>는 세대별 변화경향을 보인 것이고, <그림 3(b)>는 계산시간에 따른 변화를 보인 것이다. <그림 3>에서 RSC는 TTC, FSC는 SSC와 유사한 경향을 보여 생각하였다. <그림 3(a)>를 통해 SSC와 FSC의 세대에 따른 진화속도가 다른 전략들에 비해 빠름을 알 수 있다. SSC와 FSC는 적응도를 기초로 경쟁자를 선택하므로, 어떤 한 방향으로의 탐색이 상대적으로 빠른 것으로 보인다. 이와 달리, 다른 전략들은 SSC나 FSC보다 경쟁자를 선택하는 데 있어 특별한 제약이 없거나 유연하므로, 다양한 개체들과 경쟁할 가능성이 높아 진화속도가 늦은 것으로 보인다. 그러나 SSC와 FSC는 두 종간의 진화 불균형이 발생하기 전까지는 진화속도가 빠르나, 불균형이 발생하면 급속히 퇴화하는 경향이 있다.

<그림 3(b)>로부터 TMC전략은 절대적 진화가 일정 수준 이상 이루어지는 데 소요되는 시간이 다른 전략에 비해 짧음을 알 수 있다. SSC는 일단 무기경쟁이 깨지면 진화를 연장하더라도 진화의 균형성을 회복하지 못하여 우수하지 않았다. <표 3>에서 알 수 있듯이 TMC는 TTC 계산시간의 10%도 되지 않은 수준에서 승률이 크게 뒤지지 않는다. 이로부터 TMC가 계산소요시간 측면에서 월등히 우수하다는 것을 확인할 수 있다. TTC는 모든 상대 개체와의 경쟁, FSC는 승리횟수를 기준으로 개체들을 정렬하여 샘플링 과정, SSC는 배분 적응도를 구하는 과정, 그리고 SHC는 배분 적응도의 계산과 우수개체집합의 갱신과정에서 TMC보다 많은 계산시간을 요구한다.

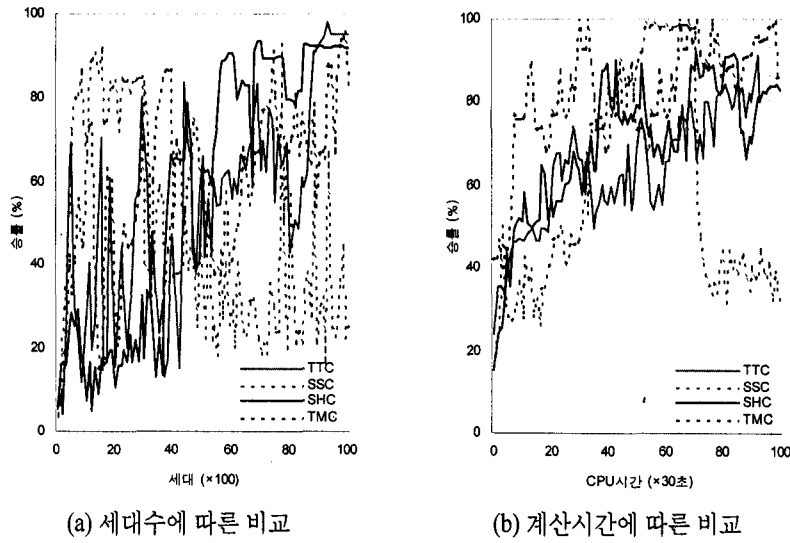


그림 3. Nim 게임에서 경쟁전략별 진화속도 비교.

RSC는 경쟁자들을 임의로 추출하므로, TMC를 제외한 다른 전략에 비해 적은 계산시간을 요구하였다.

4.3 진화의 균형성

숙주 모집단과 기생충 모집단의 평균 승률이 진화가 계속되는 동안 비슷한 수준을 유지하면, 두 모집단은 진화가 균형적으로 이루어진다고 말할 수 있다. 경쟁 공진화 알고리즘에서 경쟁하는 두 모집단간 진화의 균형은 무기경쟁을 지속시킬 수 있는 중요한 조건이며, 이를 통해 알고리즘에서 좋은 해를 탐색할 수 있다(Nolfi and Floreano, 1998). 진화의 균형성을 살펴보기 위하여 경쟁전략들의 진화경향을 분석하였다. 예비실험 결과, TTC, RSC, SHC, TMC의 진화경향은 반복실험 동안 매 회마다 큰 차이를 보이지 않았다. 그러나 FSC와 SSC는 <표 3>

의 표준편차 결과에서 알 수 있듯이, 승률에 큰 차이를 보이는 결과들이 발견되었다. <그림 4(a)>는 TMC의 많은 반복실험 중에서 임의로 선택한 한 회의 결과이고, <그림 4(b)>는 SSC의 실험 결과 중에서 승률이 상당히 낮아 해의 편차를 높게 하는 특징적인 한 결과를 보인 것이다. FSC에서 승률이 상당히 낮은 경우도 <그림 4(b)>와 같은 경향을 보였다. 이 그림에서 HBest(HMean)는 기생충 모집단과 경쟁하여 얻은 숙주의 최고 승률(숙주 모집단의 평균 승률)이다. Best/T(Mean/T)는 테스트 집합과 경쟁하여 얻은 숙주의 최고 승률(숙주 모집단의 평균 승률)이다. HBest와 HMean으로 상대적 진화를, Best/T와 Mean/T로 절대적 진화를 살펴볼 수 있다. 기생충의 최고 승률과 평균 승률은 숙주의 그것들과 반비례하므로, 이들로부터 알 수 있다.

상대적 진화를 살펴볼 수 있는 HMean이 50%의 승률 주위

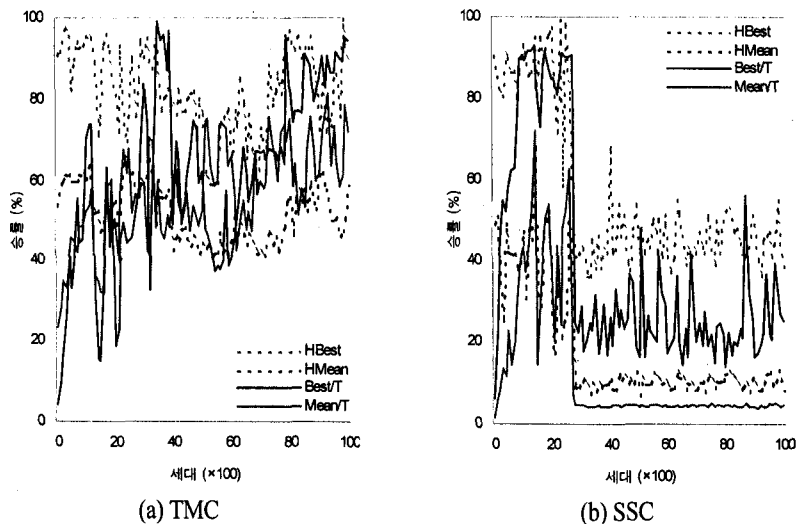
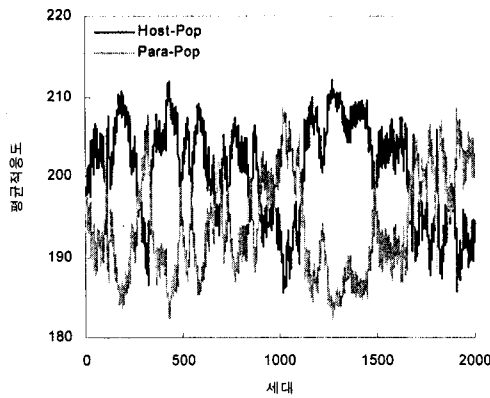
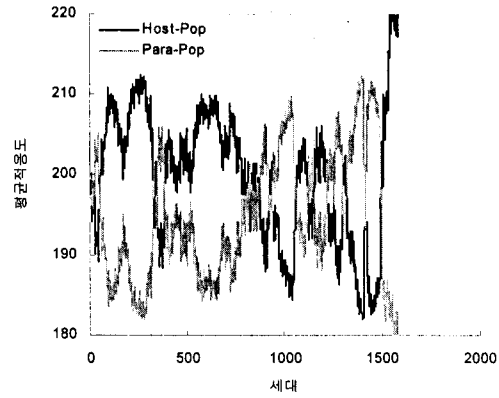


그림 4. Nim 게임에서 경쟁전략별 진화경향.



(a) TMC에서 모집단의 평균 적응도 변화



(b) FSC에서 모집단의 평균 적응도 변화

그림 5. 숙주와 기생충 모집단의 평균 적응도 변화.

에 분포하면 두 모집단의 평균 승률이 비슷하여 균형적으로 진화되고 있다고 볼 수 있다. <그림 4(a)>의 TMC는 HMean이 50% 승률 주위를 크게 벗어나지 않는다. 진화의 균형이 유지되면, 테스트 집합과의 경쟁에서 승률이 상승하고 숙주개체가 좋은 해로 진화함을 알 수 있다. 이는 Best/T와 Mean/T가 상승해 가는 것을 통해 확인할 수 있다. TTC, RSC, SHC도 모두 같은 경향을 보였다. 그러나 <그림 4(b)>에서 보는 바와 같이 SSC는 진화 초기에 HMean의 값이 50% 승률을 보이다가 그 값이 하락하여 두 모집단의 평균 승률에 현저한 차이가 나타나는 경우가 발견된다. 이는 한 모집단이 상대 모집단의 개체를 일방적으로 이기는 경우이다. 상대적 진화가 불균형적으로 이루어지면, Best/T나 Mean/T가 하락하여 절대적 진화 측면에서 좋은 해로의 진화가 멈추거나 오히려 적응력이 낮아지는 경향을 초래한다. FSC에서도 이와 같은 경향을 보였다.

<그림 5>는 TMC와 FSC에서 숙주와 기생충 모집단간의 평균 적응도 변화를 보인 것이다. 이 그림에서 TMC는 진화의 균형이 지속적으로 이루어지나, FSC는 일정 세대 후 한 모집단이 일방적으로 우세하여 두 종간의 불균형이 발생함을 관찰할 수 있다.

4.4 모집단의 다양성 유지

<그림 6>은 Nim 게임문제에서 경쟁전략들이 보이는 숙주 모집단의 다양성을 보였다. 다양성의 척도 D 는 다음 식에 의 해 구하였다.

$$D = \frac{2 \sum_{i=1}^n \sum_{j=i+1}^n H_{i,j}}{n(n-1)l}$$

위의 식에서 n 은 숙주 모집단의 크기, l 은 숙주개체의 길이, 그리고 $H_{i,j}$ 는 개체 i 와 j 의 해밍거리(hamming distance)이다. 해밍거리는 개체의 같은 위치에서 서로 다른 인자값을 갖는 비트수를 더한 값으로 계산하였다. 그림에서 알 수 있듯이,

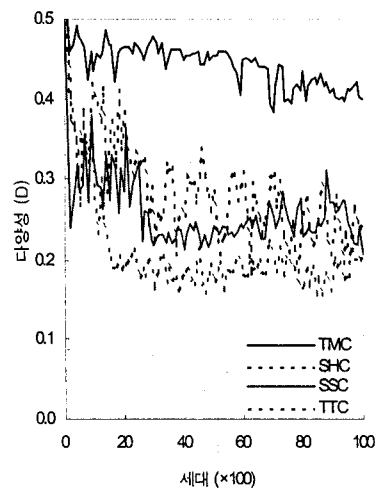


그림 6. 경쟁전략별 다양성 비교.

TMC는 다른 전략에 비해 다양성이 가장 높게 나타났고, 다음으로 SHC와 SSC의 다양성이 높다. 나머지 전략들은 TTC와 뚜렷한 차이를 보이지 않았으므로, TTC의 결과만을 보였다.

TMC는 첫 번째 토너먼트에서 패배한 개체들이 같은 적응도를 부여받기 때문에 이들 개체들은 차별성을 두지 않아 모집단의 다양성 유지에 도움이 되는 것으로 여겨진다. 모집단의 다양성 유지는 다양한 개체들과의 경쟁을 통하여 무기경쟁이 지속될 가능성이 높다는 것으로 해석될 수 있다. 또한 TMC에 의하면 개체의 상대적 순위가 결정되면서도 경쟁상대에 따라 적응도 값에 의한 상대적 순위에 차이가 날 수 있다. 즉, TMC가 <표 3>의 결과에서 좋은 결과를 보인 것은 모집단의 다양성과 선택압력(selective pressure)이 적절히 조화를 이루기 때문인 것으로 판단된다.

SHC와 SSC는 배분 적응도를 기준으로 개체를 선택한다. 배분 적응도는 경쟁하여 승리하기 어려운 상대를 패배시킨 개체에 대하여 상대적으로 높은 적응도를 부여함으로써 개체의 난이도에 따라 부여된 적응도이다. 이 방법은 다양하고 강력한

표 4. 2DTTT 게임에서 경쟁전략별 성능 비교

분석요소		TTC	RSC	FSC	SSC	SHC	TMC
승률 (%)	Best	100.00	100.00	100.00	100.00	100.00	100.00
	Worst	98.60	97.50	38.40	47.40	98.10	98.20
	Mean	99.79	99.49	96.87	97.23	99.73	99.81
	Std	0.400	0.806	13.76	11.74	0.589	0.438
완전해발견횟수		15	13	18	17	15	15
평가 횟 수		2.33E+07	6.46E+06	6.46E+06	6.46E+06	6.46E+06	1.83E+04
계 산 시 간(초)		8,611	1,673	2,044	3,544	4,585	603

개체들을 경쟁자로 선택하는 방법으로 제안되었다(Rosin, 1997). 두 전략의 다양성이 TMC를 제외한 다른 전략들에 비해 높은 것은 이러한 주장을 뒷받침한다. 그리고 SHC의 다양성이 SSC의 그것보다 약간 높은 것은 우수개체집합의 역할로 보인다.

4.5 실험결과에 대한 토의

다음으로, 또 다른 게임문제인 2DTTT 게임문제에 대한 실험결과들을 제시하고, 경쟁전략들이 보이는 특성을 좀더 살펴 보도록 하자.

<표 4>는 2DTTT게임에서 세대수에 따른 경쟁전략별 해의 질을 비교한 것으로, 20회 반복실험의 결과이다. 전체적인 실험 결과의 경향은 <표 3>의 Nim 게임결과와 유사하다. 그러나 이 결과에서 보인 두드러진 특징은 FSC와 SSC전략에서 진화의 불균형 현상이 발견되는 빈도가 훨씬 낮다는 것이다. 실험에서 두 전략은 모두 20회의 반복실험중 유일하게 한번의 실험에서 진화의 불균형 현상이 보였다. 따라서 Nim 게임결과와 비교하여 FSC와 SSC의 표준편차값은 낮게 나타났다. Nim 게임과는 달리 반복실험의 횟수를 증가한 것도 초기 10회의 반복실험 동안은 진화의 불균형 현상이 보이지 않았기 때문이다. 많은 예비실험을 통해 20~30회의 반복실험중 한두 번 정도 이러한 결과가 보임을 확인할 수 있었다. 이로부터 게임문제에서 FSC나 SSC와 같이 적응도 정보를 이용한 샘플경쟁에서 진화의 불균형 현상은 게임 특성의 복잡도에 따라 그 발생 빈도가 결정되는 것으로 보여진다. 2DTTT 게임문제는 한 참가자가 한 회에 취할 수 있는 경우의 수가 Nim 게임에 비해 적고, 흔히 3-4회의 행동만을 취하면 게임의 승패가 결정되는 비교적 간단한 게임이다. 따라서 2DTTT 게임문제는 Nim 게임문제에 비하여 최적전략을 찾는 해 공간이 적다는 특징을 갖는다(Rosin, 1997).

이상의 실험결과로부터 게임문제를 해결하고자 할 때, 경쟁 전략으로 정보를 이용한 샘플경쟁전략은 위험스러운 전략일 수 있다. 이는 샘플링에서 발생하는 치우침 또는 오류에 의해 모집단 또는 문제의 특성중 제한적 요소만을 반영하여 해의 탐색을 좋지 않은 방향으로 유도할 수 있는 것으로 판단된다. 그리고 복잡한 게임일수록 모집단 또는 문제의 특성이 다양하

고 복잡하므로, 샘플경쟁전략의 위험성은 더욱 높아질 것으로 판단된다.

결론적으로, 게임문제에서 TMC는 해의 질, 진화의 균형성, 모집단의 다양성, 그리고 계산소요시간 측면에서 분석하여 볼 때 좋은 경쟁전략으로 사용될 수 있다. TMC전략은 상위의 좋은 개체들이 진화를 유도하고 나머지 개체들의 다양성을 높임으로써 진화과정 동안 해공간을 다양하게 탐색할 수 있는 효과를 갖는다.

5. 해-테스트 문제에서의 비교 분석

5.1 해의 질

<표 5>는 3가지의 정렬망 문제에서 세대수에 따른 경쟁전략별 해의 성능을 보인 것으로, 10회 반복실험의 결과이다. 완전해 발견횟수에서 dash(-)는 어떤 실험에서도 승률이 100%인 해를 발견하지 못했음을 의미한다.

실험결과, 종료조건을 세대수로 하여 단순 비교하면 전반적으로 TTC가 비교적 좋은 결과를 보이며, 다음으로 SHC, FSC, SSC가 비슷한 수준의 성능을 보였고, TMC와 RSC는 낮은 성능을 보였다. 그러나 해공간이 작은 문제에서는 그 차이가 크지 않음을 알 수 있다. 또한 정렬망 문제에서는 Nim 게임과 달리 모든 전략의 승률에 대한 표준편차가 큰 차이를 보이지 않았다. 계산시간 측면에서는 전략들간에 큰 차이를 보이며, 완전해의 발견횟수 측면에서는 숫자열의 길이가 증가할수록 문제의 복잡도가 증가하여 발견횟수가 하락함을 알 수 있다.

<표 5>의 결과로부터, 정렬망 문제에서는 계산소요시간과 평균 승률의 성능을 동시에 고려하면 경쟁 결과에 따른 승리와 패배의 정보를 이용하여 경쟁자를 선택하는 전략들이 그렇지 않은 전략들에 비해 유리한 것으로 보인다. 정보를 이용한 경쟁전략들은 FSC, SSC, SHC, TMC이다. TMC는 토너먼트경쟁시 전단계의 토너먼트에서 승리한 개체들이 다시 짝지어 경쟁하기 때문에 개체들의 승패정보를 이용한다고 볼 수 있다. TTC는 소요되는 계산시간에 비하여 평균 승률이 그다지 월등하지 않는다. RSC는 동일한 평가횟수를 갖는 다른 샘플경쟁

표 5. 정렬망 문제에서 경쟁전략별 성능 비교

문제	분석요소		TTC	RSC	FSC	SSC	SHC	TMC
7개 숫자열 정렬망	승률 (%)	Best	100.00	100.00	100.00	100.00	100.00	100.00
		Worst	95.06	85.50	89.05	88.23	90.48	79.54
		Mean	98.93	95.47	98.10	97.17	98.76	95.79
		Std	1.807	6.321	4.065	4.759	3.047	7.855
	완전해발견횟수		7	5	8	7	8	6
	평가횟수 계산시간(초)		5.35E+08 9,975	2.30E+08 2,886	2.30E+08 3,793	2.30E+08 4,064	2.30E+08 5,509	2.30E+08 1,873
10개 숫자열 정렬망	승률 (%)	Best	100.00	100.00	100.00	100.00	100.00	100.00
		Worst	92.48	83.12	86.52	88.50	89.44	80.18
		Mean	97.05	91.56	93.49	94.60	95.01	91.92
		Std	2.540	5.289	5.197	5.181	4.572	7.277
	완전해발견횟수		3	1	3	4	3	2
	평가횟수 계산시간(초)		6.67E+08 13,579	2.87E+08 3,729	2.87E+08 4,674	2.87E+08 5,784	2.87E+08 7,349	1.83E+06 2,384
13개 숫자열 정렬망	승률 (%)	Best	96.90	92.78	94.56	94.66	96.70	94.16
		Worst	89.44	78.00	84.14	83.98	84.48	79.70
		Mean	94.08	84.05	88.64	89.54	91.74	85.15
		Std	2.136	4.084	3.302	3.089	4.137	4.938
	완전해발견횟수		-	-	-	-	-	-
	평가횟수 계산시간(초)		1.07E+09 21,511	4.60E+08 5,978	4.60E+08 7,386	4.60E+08 9,061	4.60E+08 11,712	2.93E+06 3,822

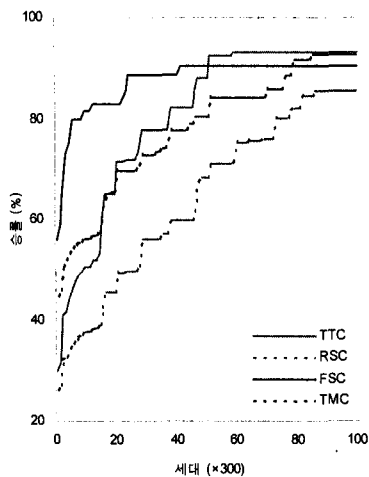
(FSC, SSC, SHC)에 비해 성능이 뒤지고, 훨씬 적은 계산시간과 평가횟수를 갖는 TMC보다도 우수하지 않다.

5.2 진화속도

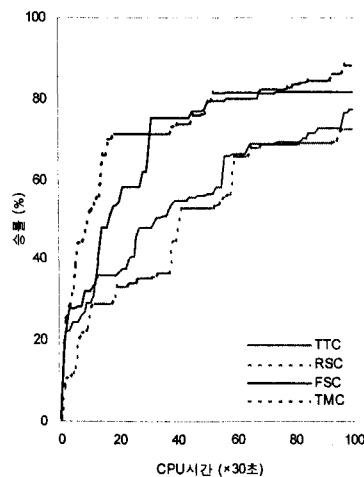
다음으로 경쟁전략간 진화속도를 비교해 보자. <그림 7>에 13개의 숫자열을 갖는 정렬망 문제에서 세대별, 그리고 계산시간별로 절대적 진화 척도인 Best/T를 제시하였다. 그림에서

SSC, SHC는 FSC와 큰 차이를 보이지 않아 생략하였다.

<그림 7(a)>에서 알 수 있듯이 FSC가 다른 전략들에 비해 진화 초기의 진화속도가 빠르다. 나머지 전략들은 FSC보다 상대적으로 진화속도가 느리다. 이것은 Nim 게임에서와 같은 결과이다. TMC는 세대의 진행에 따른 진화속도가 다소 느리지만 진화는 계속적으로 이루어져, 진화가 충분히 이루어지면(약 24,000세대) 다른 전략들에 뒤지지 않은 결과를 보인다. 이로부터 TMC는 같은 세대를 진화하는 데 소요되는 계산시간이



(a) 세대수에 따른 비교



(b) 계산시간에 따른 비교

그림 7. 정렬망 문제에서 경쟁전략별 진화속도 비교.

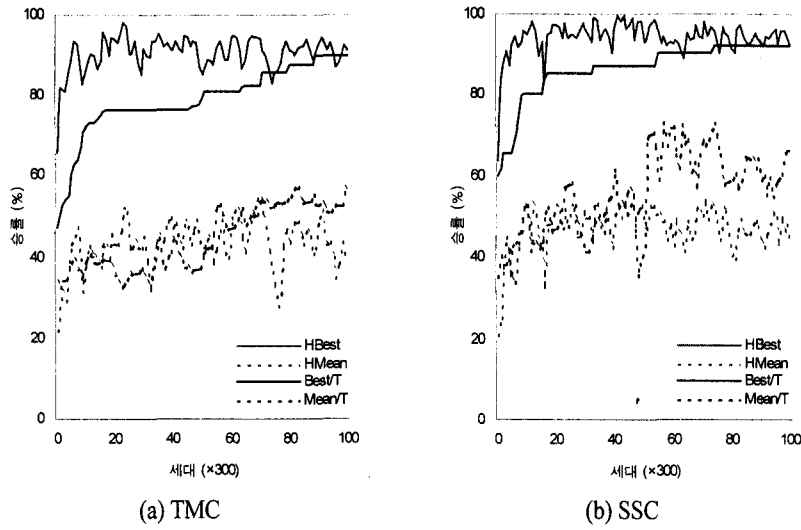


그림 8. 정렬망 문제에서 경쟁전략별 진화경향.

다른 전략에 비해 훨씬 적기 때문에 계산시간이 동일하게 주어지면 정렬망 문제에서도 TMC전략은 효과적인 전략이라고 말할 수 있다. <그림 7(b)>에서 계산시간별 Best/T의 변화를 보면 TMC가 가장 빠른 진화를 보임을 알 수 있다.

5.3 진화의 균형성과 모집단의 다양성 유지

<그림 8>은 13개의 숫자열을 갖는 정렬망 문제에서 진화의 균형성을 판단하기 위하여 보인 경쟁전략별 진화경향이다. <표 5>의 결과에서도 알 수 있듯이, 정렬망 문제에서 모든 경쟁전략은 반복실험 동안 특징적인 결과를 보이지 않았다. 따라서 <그림 8>은 TMC와 SSC의 반복실험 결과 중에서 임의로 선택한 한 회의 결과를 보인 것이다. 그림에서도 TMC와 SSC의 진화경향이 유사함을 확인할 수 있으며, 다른 전략들의 결과도 이와 유사하다.

그림에서 진화의 균형성을 판단할 수 있는 HMean을 살펴보면 어떤 전략에서도 그 값이 50%의 승률 주위에 분포되어 있음을 볼 수 있다. 즉, 정렬망 문제에서는 어떤 경쟁 전략에서도 진화의 불균형이 발생하여 무기경쟁이 깨지는 경우는 발견되지 않았다. 이는 사전에 진화의 기준이 주어지므로, 일관된 방향으로 숙주(해)들이 적응되어 가기 때문인 것으로 판단된다.

그리고 숙주와 기생충 모집단의 진화가 균형적으로 이루어지면서 점차적으로 절대적 진화의 척도인 Best/T와 Mean/T가 높아져 감을 확인할 수 있다.

또한 모집단의 다양성은 Nim 게임의 결과와 유사하여, TMC의 다양성이 가장 높게 나타났다. 이는 <그림 8>에서 Best/T와 Mean/T의 차이가 SSC보다 큰 것으로도 확인할 수 있다.

5.4 실험결과에 대한 토의

여기에서는 분류를 위한 신경망 문제에 대한 실험결과들을 제시한다. 신경망 문제에서 경쟁전략들이 보이는 해의 성능 결과는 <표 6>과 같다.

실험 결과, 신경망 문제에서 경쟁전략들이 보이는 결과들은 정렬망 문제와 유사하였고, 진화경향도 모든 전략들이 <그림 8>과 같은 경향을 보였다. 이로부터 게임문제에서와는 달리, 해-테스트 문제에서는 문제의 특성이나 문제가 갖는 복잡도에 무관하게 모든 경쟁전략들은 진화의 불균형에 대한 위험성이 없었다. 그리고 개체가 갖는 진화정보인 적응도를 이용하여 경쟁자를 선택하는 경쟁전략이 그렇지 않은 전략에 비해 진화 계산 측면에서 우수하였다.

앞에서도 언급하였듯이, TMC는 해-테스트 문제에서도 짧

표 6. 신경망 문제에서 경쟁전략별 성능 비교

분석요소		TTC	RSC	FSC	SSC	SHC	TMC
승률 (%)	Best	100.00	92.00	100.00	100.00	100.00	98.00
	Worst	89.00	77.50	86.00	86.50	87.50	84.50
	Mean	93.85	87.45	91.35	91.90	92.30	90.15
	Std	3.735	5.520	4.123	4.575	4.354	4.750
완전해발견횟수		2	-	1	2	2	-
평가횟수		2.02E+07	8.69E+06	8.69E+06	8.69E+06	8.69E+06	5.53E+04
계산시간(초)		5,238	1,456	1,798	2,206	2,852	931

은 시간 내에 숙주(해)가 일정 수준 이상 진화할 수 있도록 하며, 모집단이 다양한 개체들로 구성되도록 한다. 따라서 계산 시간이 동일하게 주어지면 다른 전략들에 비해 많은 세대를 진행하므로, 진화 기회를 많이 갖고 좋은 개체를 빨리 파악할 수 있다는 특징으로 인하여 해의 질 측면에서 다른 전략에 뒤지지 않았다.

6. 결론

본 연구에서는 경쟁 공진화 알고리즘에서 개체의 적응도를 결정하는 경쟁전략들의 성능 및 특성을 진화계산의 관점에서 비교 분석하였다. 이를 위하여 경쟁 공진화 알고리즘이 적용되는 적대문제를 게임문제와 해-테스트 문제로 나누고, 문제 형태에 따라 경쟁전략들을 분석하였다.

게임문제에서는 해의 질 측면에서 TTC, RSC, TMC가 비슷한 수준을 보였고, FSC, SSC는 해의 편차가 심하여 안정적으로 좋은 해를 탐색하지 못하였다. SHC는 우수개체집합의 역할이 SSC가 갖는 단점을 어느 정도 극복한 것으로 보인다. FSC와 SSC가 상당히 낮은 성능을 보인 경우는 숙주와 기생충 모집단의 진화가 균형적으로 이루어지지 못하는 경우에 해당한다. 또한 FSC와 SSC는 진화 초기의 진화속도는 다른 전략들에 비해 우수하나 무기경쟁이 쉽게 깨질 가능성이 있으며, 무기경쟁이 깨지면 이의 회복이 어렵다. 그리고 문제의 복잡도에 따라 FSC와 SSC의 성능에는 차이를 보인다. 따라서 이들 전략들은 좋은 해를 보장하기 어렵다. TMC는 개체들간의 적은 경쟁으로도 TTC가 보이는 성능과 근접한 결과를 보이면서, 계산시간은 짧게 소요되었다. 이는 모집단의 다양성이 유지되고, 적응도가 높은 개체들을 빨리 찾아낼 수 있다는 TMC의 특성에 기인한 것으로 보인다.

해-테스트 문제에서는 진화 불균형의 위험성은 없는 것으로 나타났다. 이는 진화의 기준이 사전에 주어져 일관된 적응방향을 가질 수 있기 때문인 것으로 판단된다. 알고리즘의 성능은 경쟁자를 선택하는 과정에서는 정보(적응도 또는 경쟁결과)를 기준으로 하는 전략, 즉 FSC, SSC, SHC, TMC가 그렇지 않은 전략들에 비해 좋은 성능을 보였다. TMC 전략은 계산시간이 동일하게 주어지면 해의 질 측면에서 다른 전략들과 비슷하거나 우수한 성능을 보이는 것으로 나타났다.

결론적으로, TMC는 경쟁 공진화 알고리즘이 적용되는 문제의 특성에 무관하게 해의 질 측면에서 다른 전략들의 성능에 크게 뒤지지 않거나 우수하면서, 계산시간 측면에서 유리

한 결과를 보였다. TMC는 개체의 적응도를 효율적으로 평가하여 숙주와 기생충의 진화가 균형적으로 이루어질 수 있도록 하며, 모집단의 다양성을 유지함으로써 경쟁 공진화 알고리즘이 갖추어야 할 요소들을 충족시키는 전략이라고 본다. 본 연구에서 제시한 경쟁전략들의 비교분석 결과는 효율적인 경쟁 공진화 알고리즘의 설계에 적용될 수 있을 것으로 기대된다.

참고문헌

- Davis, L. (1985), Applying adaptive algorithms to epistatic domains, *Proceedings of the International Joint Conference on Artificial Intelligence*, 162-164.
- Fox, B. R. and McMahon, M. B. (1991), Genetic operators for sequencing problems, *Foundations of Genetic Algorithms*, 284-300.
- Hillis, W. D. (1991), Co-evolving parasites improve simulated evolution as an optimization procedure, *Artificial Life II*, 313-324.
- Holmes, J., Routen, T. W. and Czarnecki, C. A. (1995), Heterogeneous co-evolving parasites, *Proceedings of Artificial Neural Nets and Genetic Algorithms '95*, 156-159.
- Juillé, H. (1995), Evolution of non-deterministic incremental algorithms as a new approach for search in state spaces, *Proceedings of the Sixth International Conference on Genetic Algorithms*, 351-358.
- Kim, S. J., Kim, Y. K., Kim, J. Y. and Kwak, J. S. (2000), A competitive coevolutionary algorithm with tournament competitions, *Journal of the Korean Institute of Industrial Engineers*, 26(2), 101-109.
- Kim, Y. K., Yoon, B. S. and Lee, S. B. (1997), *Metaheuristic*, Yeongji Moonhwas, Seoul, Korea.
- Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural selection*, The MIT Press, Cambridge.
- Nolfi, S. and Floreano, D. (1998), Coevolving predator and prey robots: Do 'Arms Races' arise in artificial evolution?, *Artificial Life*, 4, 311-335.
- Olsson, B. (1996), Optimization using a host-parasite model with variable-size distributed populations, *International Conference on Evolutionary Computation '96*, 295-299.
- Pagie, L. and Hogeweg, P. (1998), Evolutionary consequences of coevolving targets, *Evolutionary Computation*, 5(4), 401-418.
- Paredis, J. (1995), Coevolutionary computation, *Artificial Life*, 2, 355-375.
- Rosin, C. D. (1997), *Coevolutionary Search Among Adversaries*, Ph.D. dissertation, University of California, San Diego.