

자율주행로봇을 위한 진화 알고리즘에서 vector화 된 fitness function의 적용

윤석배*, 이종호
인하대학교 전기공학과

Genetic algorithm using the vectored fitness function for autonomous mobile robot

SeokBae Yun*, ChongHo Lee
Dept. of Electrical Eng. Inha university

Abstract - In this paper, we suggest a vector fitness strategy for obstacle avoidance of autonomous mobile robot with genetic algorithm. Ordinary genetic algorithms provide not such a viable solution for autonomous running in a variant environment, because of the difficulty in fitness evaluation in real time. We show that the suggested method is efficient for the problem of autonomous mobile robot. Its control function evolves to adapt the varying environment. The experiment is done using the real mobile robot 'Khepera', with which we use a tournament genetic algorithm model with the Vectored Fitness Genetic Strategy.

1. 서 론

유전 진화 알고리즘은 뛰어난 global search 능력과 해의 최적화 능력으로 인해 여러 응용 분야에서 매우 주목받고 있다.[1] 이러한 진화 알고리즘의 파생분야 중 하나인 진화형 로보틱스(Evolutionary Robotics)는 로봇 스스로가 환경에 대해 적응하며 컨트롤러의 구조나 파라미터에 진화 알고리즘을 적용시킴으로써 기존의 알고리즘 기반의 설계에서 미처 생각하지 못한 예외적인 부분이나, 다양한 환경에 대해 적응적으로 대처할 수 있다.

하지만 실시간 진화의 경우, 진화 알고리즘의 적용도 평가 문제에 있어서 장시간이 소요되는 단점이 있으며, 다양한 환경에 대한 알맞은 적합도 함수(fitness function)의 설계가 어렵기 때문에 진화에 너무 많은 시간이 소요되거나, 국소해(local minima)에 수렴되기도 하며, 혹, 전역해(global minima)로 수렴했다고 하더라도 시간에 따른 환경의 변화로 인해 하나의 유전자 형(chromosome)에 대해서도 적합도 값이 수렴하지 못하는 경우가 생긴다.

이러한 점에 대해서 본 논문에서는 벡터화 된 fitness function을 사용하고 평가하는 방법인 Vectored Fitness Genetic Strategy를 제안하고, 실제 주행 로봇인 Khepera를 통해 제안된 방법과 기존의 GA를 비교함으로써 제안된 방법의 우열여부를 평가하였다.

2. 본 론

2.1 진화형 로보틱스

진화형 로보틱스는 외부의 도움 없이 스스로 주어진 환경에 대해 문제 해결이 가능한 로봇, 즉 다양한 환경에 대한 자율성을 확보할 목적으로 생겨났다. 기본적으로는 Alife에 기반하고 있으며, 스스로에 대한 구조 진화나 파라미터 진화를 통해 로봇이 스스로 환경에 적응 할 수 있도록 한다.

복잡한 구조를 설계할 필요가 없고, 로봇 스스로가 환경에 적응한다는 것은 진화형 로보틱스의 가장 큰 장점인 반면 진화까지의 시간이 너무 많이 소요되는 단점이 있다.[2] 본 논문에서는 진화형 로봇의 구현을 자율 주행 로봇의 장애물 회피 능력 구현으로 선정하였다.

2.1.1 Braitenberg's Vehicle

Valentino Braitenberg에 의해 제안된 vehicle 모델은 단순히 vehicle에 있는 센서들과 구동계 사이를 직접 연결하는 것이다. Braitenberg는 이러한 단순한 구조의 vehicle만으로도 실제 곤충의 행동양식을 모방 할 수 있을 정도의 복잡한 동작을 나타낼 수 있다는 것을 보여주었다. 이러한 Braitenberg의 vehicle은 센서의 개수, 연결부호와 방향에 따라 몇 가지 모델로 나누어 질 수 있는데 이중에서 vehicle 3c는 여러 개의 센서가 좌우에 분산되어 연결되어 센서들의 부호도 positive와 negative가 혼합되어 있다. 이러한 점으로 인해 vehicle 3c는 특정 방향의 광원(light source)에 대하여 추적하거나 회피하는 동작이 가능하며 이를 조합하여 복잡한 행동을 구현할 수 있다.[3]

본 논문에서는 장애물 회피를 구현하기 위한 컨트롤러로써, vehicle 3c 모델을 기반으로 신경회로망 구조의 소프트웨어적인 컨트롤러를 구성하였다. 그림 1)은 위에서 설명한 컨트롤러에 대한 블록도이다.

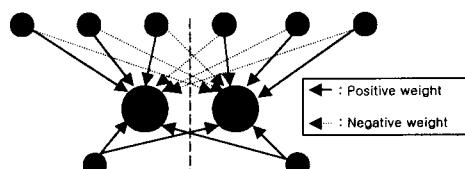


그림 1) Braitenberg 모델의 controller 블록도

Braitenberg의 vehicle은 광원에 대한 추적과 회피 기능을 가지지만, 본 실험의 경우는 물체에 대한 회피 판단이 필요하게 된다. 따라서, Braitenberg의 제어기를 적용하기 위해서 IR(Infra Red) 센서와 IR LED를 사용하여 스스로 방출한 빛에 대한 반사도를 이용하여 물체의 존재여부를 판단하게 한다.

기본적인 Braitenberg 모델은 좌우 센서에 대한 대칭성으로 인해 모든 센서에 대한 광량이 동일한 경우 즉, 그림 2)와 같은 경우가 되면 아무런 움직임도 보이지 않는다.

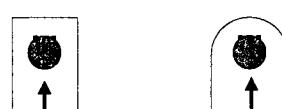


그림 2) Braitenberg vehicle이 동작하지 않는 경우

본 실험에서는 기본적인 Braitenberg 모델의 위와 같은 문제점을 진화 알고리즘을 통해 극복하는 것을 보여준다.

2.1.3 유전 진화 알고리즘

진화 알고리즘은 1960년대에 John Holland에 의해 창안되었다. 자연계의 적자생존 과정을 기본 모델로 한 이 이론은 어떠한 문제에 탐색공간내의 후보해(chromosome)들에 대해서 교배(crossover), 변이(mutation)등의 연산을 거친 자손들을 환경에 대한 적응도 평가를 함으로써 좀더 적합한 해들을 골라내는 과정을 통해 문제의 전역 해에 접근하는 알고리즘이다.[1]

2.1.4 Vectored Fitness Genetic Strategy

진화 알고리즘의 가장 중요한 요소 중의 하나는 환경에 대한 적합도 함수이다. 특히 자율이동 로봇의 진화 같은 경우는 다양한 환경변수 및 시간에 따라 달라지는 환경하에서 적응도 평가를 해야 한다는 어려움이 있다. 이에 대해 기존의 일반GA의 적합도 함수은 1차원, 즉 scalar적인 정보밖에 가질 수 없다. 이러한 점에 대해 우리는 하나의 후보해에 대해 복수 차원에서 평가를 함으로써, 서로 다른 차원에서 독립적인 정보를 많이 갖는 후보해에게 선택될 확률을 많이 주는 방법을 고안했다.

VFGS(Vectored Fitness Genetic Strategy)의 구현 과정은 다음과 같다. 먼저 서로 독립하는 벡터 성분에 대한 적합도 함수들을 정의한다. 그리고 하나의 후보해에 대해 각각의 벡터 성분에 대한 적합도 평가를 한다. 이 과정을 통해 얻은 결과를 hyper plane상에서 표현하고, 이러한 일련의 과정을 모든 같은 세대의 개체들에게 적용시킨다.

각각의 적합도 값들은 각각의 벡터성분이 가지는 적합도 함수에만 의존하며, 각각의 이러한 성분들은 벡터 F 안에서 서로에 대해 동등하지 potential을 가질 수 있으며 환경에 대해 동등한 potential인 경우는 모든 벡터의 크기보다 방향을 중요시하여 선택율을 결정한다.

이러한 과정을 통해 구현되는 VFGS는 일반 GA보다 안정적이고 빠른 수렴 결과를 보여준다는 것을 자율이동 로봇의 장애물 회피 실험을 통해 증명하였다.

2.1.5 자율이동로봇의 장애물 회피를 위한 VFGS

실제로 자율이동로봇의 장애물 회피의 구현을 목적으로 작성된 적합도 함수를 식(1), 식(2), 식(3), 식(4)에 나타내었다. 여기서 사용한 적합도 함수들은 모두 에너지가 감소하는 방향, 즉 오차에 대한 평가로 나타내었기 때문에 0에 가까워질수록 좋은 값을 가지는 것으로 하였다.

$$fit_1 = \sum_{i=1}^N c_i \quad c_1 = \begin{cases} 1 & v_{left} < v_{th} \text{ and } v_{right} < v_{th} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$fit_2 = \sum_{i=1}^N c_2 \quad c_2 = \begin{cases} 1 & sensor_{th} < sensor_{front} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$fit_3 = \sum_{i=1}^N \frac{c_3}{\alpha + v_{right}} \quad c_3 = \begin{cases} 1 & sensor_{left} = 0 \text{ or } sensor_{th} < sensor_{left} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$fit_4 = \sum_{i=1}^N c_4 \quad c_4 = \begin{cases} 1 & sensor_{left} < sensor_{th} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

식(1)(2)(3)(4)에서, N은 평가 시간 내 반복횟수, v_{th} 는 모터의 collision threshold값, $sensor_{th}$ 는 센서의

collision threshold이다. 이 중에서 식(3)의 fit_3 은 로봇의 좌수법(left hand method)주행으로 미로를 탐색하는 능력에 대한 평가를 하는 적합도 함수이다. 각각의 적합도 함수들은 fitness vector F의 성분이 된다. 따라서 4개의 차원을 갖는 F는 식(5)와 같이 정의 될 수 있다.

$$F = [fit_1 \ fit_2 \ fit_3 \ fit_4] \quad (5)$$

selection method에 대한 선택 우선권은 각각의 후보해들이 갖는 벡터 성분들의 순서, 즉 $fit_1 \ fit_2 \ fit_3 \ fit_4$ 와 같은 순서로 선택 우선권을 부여받을 수 있다.

2.2 자율 주행 로봇의 장애물 회피에 관한 VFGS의 적용 실험

2.2.1 Khepera Robot 과 실험용 필드

실험에 사용된 주행 로봇은 Khepera라는 로봇으로써, 직경55mm 높이 30mm의 소형 로봇이다. 스위스 Lausanne 대학의 마이크로 일렉트로닉스 연구소가 자신들의 실험용으로 사용했으나, 1993년, 일본에서 개최된 실험 로봇 심포지엄에서 소개된 이후 로봇진화에 관련된 분야에서 가장 많이 사용되는 로봇이다.[4]

Khepera의 사양은 CPU로 MC68331을 채택하고 있으며, RAM 256KB, ROM 256KB의 메모리를 가지고 있다. 센서는 모두 8개로 환경센서와 근접센서의 두 가지 기능을 가지고 있으며, Motor는 PID기반의 PWM 속도제어의 DC모터로 1mm/s에서 1m/s까지 속도조절이 가능하며 약 45분간 단독 구동 가능하도록 배터리를 탑재하고 있다.[5]

실험용 필드는 55cm×55cm×7.5cm크기의 미로이며 Balsa판을 이용해 제작되었다. 그림 3)은 Khepera 로봇과 실험용 필드의 외형 사진이다.

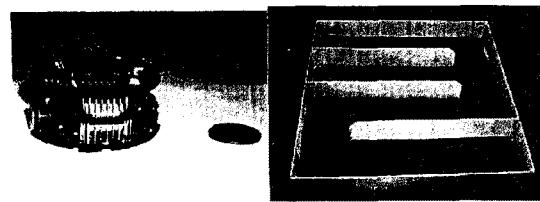


그림 3) Khepera 로봇과 실험용 field

2.2.2 실험을 위한 유전 진화 알고리즘

로봇을 진화시키기 위해 사용된 프로그램은 C 언어로 작성되었으며, Khepera에 직접 탑재시키기 위해 Linux 환경에서 68k cross-compiler를 사용하여 제작되었다.

사용된 GA는 단순 GA로써, 선택방법에는 토너먼트 선택법이 사용되었으며, 이러한 것들은 단일 적합도를 갖는 GA와 VFGS의 두 가지에 대해 동일하게 적용되었다. 단일 적합도 평가 실험에 사용된 적합도 함수는 식(6)에 나타내었다. 역시 VFGS와 동일하게 평가하기 위해서 0에 가까워질수록 좋은 값을 나타내는 것으로 충돌에 대한 센서와 모터라는 측면에서 센서쪽에 비중을 많이 두어 제곱항을 사용해 나타내었다.

$$F = \frac{1000 \times (0.7 + \sum_{i=1}^8 contact_i^2)}{0.5 + |v_{left}| + |v_{right}|} \quad contact_i = \begin{cases} 1 & sensor_i > sensor_{th} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$sensor_{th}$ 는 850으로 정했으며 $F[fit_1]$ 에 사용된 v_{th} 는 마찰을 고려해 10으로 정했다. 사용된 GA의 유전자형

은 4bit × 8개의 32bit로써 각각의 4bit는 센서와 모터 간을 연결하는 연결강도가 된다. 연결강도는 총 16개가 있는데 좌우 대칭으로 positive connection과 negative connection이 존재하므로 8개를 2번 중복해서 사용하였다.

유전자형의 개체수는 30, 진화연산에 필요한 교배율은 50%, 그리고 돌연변이율은 0.7로 일반 GA의 파라미터 결정 범위 안에서 정하였다.

2.2.3 실험 방법

실험은 구동 및 평가용으로 제작된 프로그램을 Khepera에 다운로드 한 후 결과를 RS-232C를 통해 PC에서 기록하였다. 하나의 후보해 평가에 걸리는 시간은 20초였으며, 30개의 후보해들에 대해서 연속적으로 평가되었다. 하나의 세대가 평가받은 후에는 토너먼트 방법에 의해 선택된 유전자형들을 토대로 교배와 변이를 거쳐서 태어난 다음 세대들로 다음 세대를 구성했다.

이와 같은 과정을 40세대동안 반복하였고 이를 다시 vector fitness 모델과 scalar모델에 대해 반복하였다.

2.2.4 실험 결과

40세대동안 로봇의 주행은 VFGS쪽이 매우 안정적이었다. 약 30세대를 지나면서부터는 벽에 충돌하는 chromosome이 거의 사라졌으며 최소값은 항상 0을 유지했다. 그림 5)는 40세대동안 VFGS와 일반 GA의 평균 적합도를 나타낸 것으로 scalar모델의 경우 수렴의 모습을 전혀 보여주지 못하는 한계를 보여주었으나, VFGS의 경우는 fit_1 , fit_2 , fit_4 가 거의 완벽히 수렴하는 결과를 보여주었다. 하지만 좌수법 능력에 대한 평가인 fit_3 은 그다지 만족스러운 결과를 보여주지 못했지만 그래도 감소하는 경향을 보였기 때문에 더 많은 세대에 대한 평가를 통해 수렴할 수 있는 가능성이 있음을 보였다.

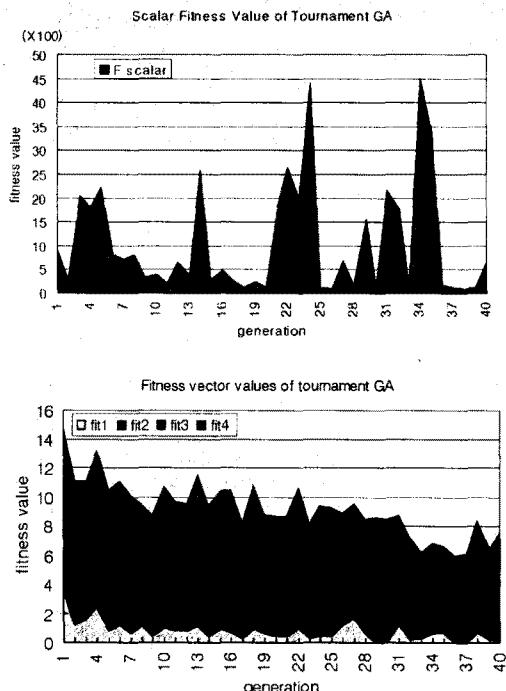


그림 5) 일반 GA와 VFGS의 평균 적합도

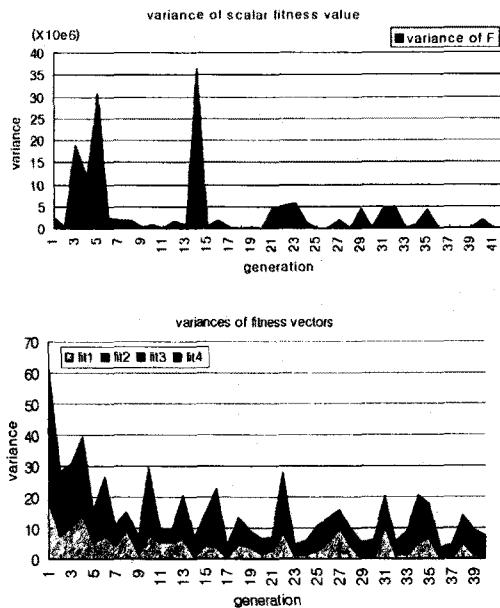


그림 6) 일반 GA와 VFGS의 적합도 분산

그림 6)은 후보해들의 분산 정도를 나타낸 것으로 일반 GA가 매우 큰 편차를 보이는 반면, VFGS쪽은 전체적으로 exponential하게 감소하는 형태를 보여주었다. 최종 평균 적합도 값에 대한 상대 분산값은 일반 GA쪽이 작아서 수렴율이 좋다는 것으로 해석 될 수 있지만, 국소해를 방지하는 개체의 다양성 확보라는 측면을 고려하면 분산율은 조금 크더라도 전체적으로 안정적인 감소율을 보이는 VFGS쪽이 우월하다는 결론을 내릴 수 있었다. 또한 이러한 주장은 평균 적합도의 우월성에 의해 뒷받침 될 수 있다.

3. 결 론

자율 주행 로봇의 진화 같은 시간에 따라 계속 변화하는 환경에서의 적합도 평가는 매우 어렵다. 같은 유전자형이라도 변화된 환경에 따라 전혀 다른 값을 내기 때문이다. 이러한 환경에 대해서 VFGS는 효율적인 평가가 가능하다. 다차원을 통해서 평가하기 때문에 하나의 벡터성분이 매우 높은 값을 가지더라도 우선 순위가 높은 벡터의 성분이 낮으면 우선순위의 차이에 의해 선택도가 낮아지는 경향을 보였다. 이러한 VFGS는 일반 GA가 보이는 적합도 판정의 어려움이나 진화의 안정성 문제에 대해 큰 영향을 받지 않고 수월하게 진화할 수 있음을 소프트웨어 시뮬레이션이 아닌 실험을 통해 증명하였다.

(참 고 문 현)

- [1] Melanie Mitchell, An Introduction to Genetic Algorithm, MIT press, 1996
- [2] Henrik Hautop Lund, "Biologically Inspired Robots", Evolutionary Robotics (ER98), volume2, p105~p131, 1998
- [3] Valentino Braatenberg, Vehicles:Experiments in Synthetic Psychology, MIT press, 1984
- [4] David E. Goldberg, Genetic algorithm in Search, Optimization, and Machine Learning, Addison-Wesley, 1989
- [5] K-team, Khepera User Manual, Lausanne Univ., 1999