

유전 프로그래밍에 의한 자율이동로봇군의 협조행동 및 제어

Cooperative Behavior and Control of Autonomous Mobile Robots Using Genetic Programming

*이 동 옥, 심 귀 보

중앙대학교 제어계측공학과 로보틱스 및 지능제어 연구실

Tel: 820-5319; Fax: 817-0553; E-mail: dwlee@jupiter.cie.cau.ac.kr, kbsim@juno.cie.cau.ac.kr

Abstract In this paper, we propose an algorithm that realizes cooperative behavior by construction of autonomous mobile robot system. Each robot is able to sense other robots and obstacles, and it has the rule of behavior to achieve the goal of the system. In this paper, to improve performance of the whole system, we use Genetic Programming based on Natural Selection. Genetic Programming's chromosome is a program of tree structure and it's major operators are crossover and mutation. we verify the effectiveness of the proposed scheme from the several examples.

Keyword : Autonomous Mobile Robots, Distributed Robot System, Cooperative Behavior, Genetic Programming, Tree structure.

1. 서론

자율이동로봇은 여러 종류의 센서에 의해 주위 환경을 판단하여 장애물이나 목표물을 인식하고 주어진 일을 달성하는 시스템이다. 현재까지는 주로 하나의 로봇이 문제를 해결하기 위한 방법들이 연구되어 왔으며 최근에는 여러 대의 자율이동로봇의 분산적인 상호작용에 의해 어떤 전체적인 목표를 달성하도록 하는 군지능 또는 군행동의 연구에 대한 관심이 높아지고 있다. 이것은 자율분산 로봇, 멀티 로봇/agent 시스템, 자기조직화 로봇, 복수의 로봇의 협조 등에 관한 논문이 많이 등장하고 있는 것을 보아도 알 수 있다^[1].

이와 같이 여러 대의 자율이동로봇을 이용하면 단일 로봇을 가지고 할 수 없는 일을 처리할 수 있으며 로봇간의 상호작용에 의한 새로운 능력이 생겨날 수도 있다. 또한 복수의 로봇으로 구성된 자율 분산적인 시스템은 작업 목적에 따라 행동규칙을 바꾸어 줌으로서 여러 가지 다른 일에도 적용할 수 있으며, 만약 한 두 대의 로봇이 고장나더라도 전체 시스템의 동작에는 영향을 주지 않으며, 시스템이 커지더라도 복잡성이 증가하지 않는 등의 장점이 있다.

앞으로 마이크로 로봇의 발전과 더불어 단순한 기능을 가진 로봇의 군행동 발현이나, 협조 행동 등은 앞으로 중요한 요소가 될 것으로 보인다. 이에 본 논문에서는 주변환경의 감지 능력을 갖춘 여러 대의 자율이동로봇을 가지고 협조행동을 이끌어 내기 위한 알고리즘의 개발을 목표로 한다. 이러한 자율이동로봇은 주변정보를 인지하고 자신의 해야할 일을 결정할 수 있는 규칙이나 프로그램이 필요하게 된다. 그러나 군 행동의 발현을 위하여 각 개체에 복잡한 행동 규칙을 넣어주는 것은 어려우며 넣어 주었다 하더라도 다른 개체와의 상호작용에 의해 군 전체에 대한 원하는 결과가 나오지 않을 수도 있다 그러므로 이보다는 개체들

이 단순한 규칙을 가지고 이에 의해 나타나는 결과를 토대로 개체들의 규칙을 변화 시켜 나가게 되면 군 전체의 행동에 의해서 개체들의 규칙이 발전되어 가는 것이 되므로 더 효과적이다. 본 논문에서는 이와 같이 개체의 행동을 이끌어내기 위하여 자연계의 진화이론을 이용한 진화 연산(Evolutionary Computing)을 이용한다. 진화 연산에는 대표적으로 유전 알고리즘(GA: Genetic Algorithms), 유전 프로그래밍(GP: Genetic Programming), 진화 전략(ES: Evolution Strategies), 진화 프로그래밍(EP: Evolutionary Programming) 등이 있으며^{[2][3]} 각각의 특징을 살리면 어느 것이나 적용 가능하지만, 특히 유전 프로그래밍은 개체의 행동을 나무구조의 프로그램으로 표현할 수 있으며 프로그램의 크기에 제한성이 없어서 표현력에 있어서 융통성이 많으므로 창발적인 행동의 발현에 좋은 결과를 기대할 수 있다. 따라서 본 논문에서도 유전 프로그래밍을 이용하여 복수의 로봇의 협조 제어를 실현한다.

2. 군지능 및 군행동

개미나 벌과 같이 집단을 이루고 사는 곤충들에게서 우리는 집단적인 행동과 전체적인 질서의 유지력을 볼 수가 있다. 이러한 곤충들은 집단을 유지시키기 위한 절대자나 명령자가 없으며 단지 개개의 곤충이 다른 곤충과의 관계에 따른 적절한 행동을 함으로서 침략자를 막고 먹이를 구하여 집단을 유지시켜 나간다. 이와 같이 혼자서는 할 수 없는 일에 개개의 행동이 모여 전체적으로 어떤 목적을 이루어 나가는 것을 군지능(Collective Intelligence, Swam Intelligence)이라고 한다.^{[4][5]}

최근에는 생물학과 로봇공학 분야에서 이와 같은 군의 지능을 인공적으로 구현해 보려는 시도가 많아지고 있으며 대표적인 것

으로 자율분산 로봇 시스템(DRS: Distributed Robot System)^[5]과 세포 로봇 시스템(CEBOT: Cellular Robotic System)^[6]에 관한 연구가 있다. 이러한 시스템은 각 개체가 비교적 단순한 기능만을 가지며 하나의 개체가 커다란 일을 할 수 없는 것과는 관계없이 어느 정도 이상의 집단이 형성되면 집단의 지적인 행동이 가능하게 된다는 특징이 있다.

MIT 대학의 Mataric는 오직 감지(Sensing)에 의해서만 주변의 다른 로봇이나 물체의 존재를 감출하여 다른 개체와의 기본적인 상호작용에 의한 기초적인 군행동을 제시하였다. 제시한 기초적인 군행동은 충돌회피(Collision Avoidance), 따라가기(Following), 흩어지기(Dispersion), 모이기(Aggregation), 목적지로 향하기(Homing), 무리 짓기(Flocking) 등이며 예를 들어 넓은 지역에서 먹이의 찾아 모으는 작업은 먹이의 밀도가 높은 지역은 따라가기 및 무리 짓기, 밀도가 낮은 지역은 흩어지기과 목적지로 향하기 등의 동작으로 군의 목적을 달성 할 수 있다^[7].

또한 최근에는 인공지능의 개념을 도입함으로써 자율이동로봇군의 자기조직화 및 진화에 의하여 개개의 로봇이 스스로 행동 방식을 획득하여 군행동이나 군지능을 형성 할 수 있는 연구에도 관심이 나타나고 있다^{[8][9][10]}. 여러 대의 자율이동로봇이 협조 행동을 통해서 복잡한 문제를 해결하기 위해서는 주위 및 다른 개체와의 상대적인 관계로부터 적절한 행동을 결정해야 하는데 이러한 결정에 의해 나타나는 집합적 행동은 몇 개의 상호작용 규칙을 가진 자기조직 시스템의 창발적 성질로 볼 수 있으며 진화론적인 방법이나 퍼지, 신경망, 카오스이론 등에 의한 융합에 의해 더욱 뛰어난 군지능을 구현할 수 있을 것이다.

3. 유전 프로그래밍

인간 프로그래머에 의해서만 생성되던 컴퓨터의 프로그램을 컴퓨터에 의해 자동생성 되도록 하기 위한 방법으로 다윈의 진화이론을 도입한 것이 유전 프로그래밍^[11]이다. 유전 알고리즘, 진화 전략, 진화 프로그래밍 등이 60~70년대에 개발된 것에 비해 이것은 90년대 초 Stanford 대학의 John Koza에 의해 개발, 소개되어 연구가 진행중이며 아직 개발의 여지가 많이 남아있다.

유전 프로그래밍의 염색체 표현은 LISP언어의 S-식 표현방법인 나무구조로 되어있으며 다른 진화 연산 방법과 마찬가지로 나무구조의 프로그램을 임의로 여러 개를 생성한 후 이것의 수행능력에 따라 경쟁적으로 선택, 도태되어 진화시킴으로서 원하는 프로그램을 얻어낸다. 유전 알고리즘에 비해 염색체의 구조만 확장했을 뿐 기본적인 실행 방법은 거의 유사하다. 그러나 유전 알고리즘에서 스트링으로 표시되던 염색체를 나무구조의 프로그램으로 확장하면서 학습, 추론, 문제해결 등 적용분야도 넓어졌다.

프로그램 진화를 위한 기본적인 조작 방법은 다음의 세 가지가 있다.

(1) 돌연변이(Mutation) - 노드의 임의적인 변경

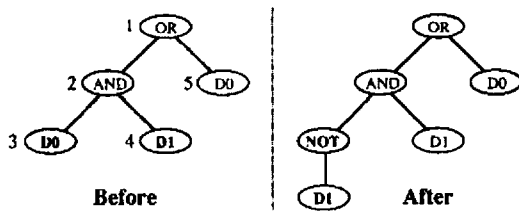


그림 1. 돌연변이 전과 후

그림1의 S-식 표현은 (OR (AND (D0 D1) D0))이고 3번 노드의 D1이 바뀌어 NOT D1으로 되었으므로 (OR (AND (D0 (NOT D1)) D0))으로 바뀌었다.

(2) 교배(Crossover) - 다른 나무와 부분나무의 교환

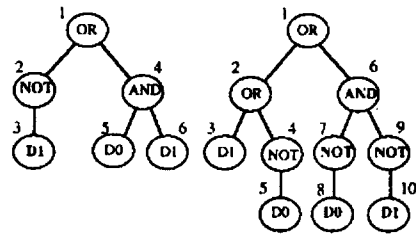


그림 2. 교배 전 구조

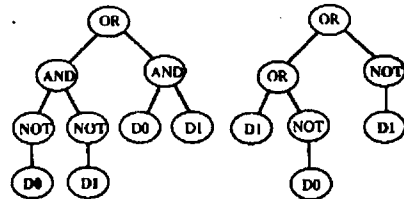


그림 3. 교배 후 구조

그림2의 (OR (NOT D1) (AND D0 D1))과 (OR (OR D1 (NOT D0)) (AND (NOT D0) (NOT D1)))의 2번과 6번 노드가 교배되어 그림3의 (OR (AND (NOT D0) (NOT D1)) (AND D0 D1))과 (OR (OR D1 (NOT D0)) (NOT D1))로 되었다.

(3) 역위(Inversion, Permutation) - 한 나무 내에서 노드간의 위치 바꿈

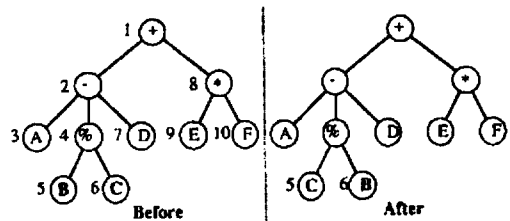


그림 4. 역위 전과 후

그림4의 (+ (- A (% B C) D) (* E F))의 5와 6번 노드가 바뀌어 (+ (- A (% C B) D) (* E F))로 되었다.

이외에도 편집(Editing), 캡슐화(Encapsulation), 소거(Decimation) 등 문제에 따라 여러 가지를 함께 사용할 수 있다.

유전 프로그래밍으로 문제를 해결하기 위해서는 우선 다음의 5단계가 필요하다.

- (1) 말단 기호의 집합 결정
- (2) 함수의 집합 결정
- (3) 적합도 산정방법의 결정
- (4) 알고리즘의 파라메타 결정
- (5) 종결조건 결정

유전 프로그래밍은 로봇의 프로그램 생성, 게임 프로그램, 화상인식, 인공지능의 여러 가지 문제해결이나 학습 등에 응용되고 있으며 알고리즘의 유효성도 이미 증명이 된바 있다. 특히 본 논문에서는 군의 협조행동을 달성하기 위한 로봇의 프로그램을 만들어내고 이에 따른 로봇들의 행동에 의해 군 목적의 수행정도를 측정하여 선택, 도태시킴으로써 해서 목적에 맞는 로봇의 프로그램을 얻어내기 위한 방법으로 유전 프로그래밍을 사용한다.

4. 상자 밀어내기 문제

4.1 문제의 설정

하나의 로봇이 운반하기 힘든 물체의 운반에 여러 로봇을 함께 사용하면 일을 수행할 수가 있을 것이다. 그러나 여러 개를 로봇이 있어도 서로 협조가 되지 않는다면 그 일을 수행하기는 어려워진다. 그러므로 로봇들이 일단 상자 앞으로 모인 후 동시에 상자를 밀어낼 수 있는 규칙을 찾아내는 것이 목적이다.

4.2 문제 해결을 위한 준비

문제를 유전 프로그래밍에 적용하기 위해 위 3절에서 언급한 다섯 단계의 준비가 필요하다.

(1) 말단 기호의 집합(Terminal set) 결정

말단 기호는 나무구조의 프로그램의 맨 아래에 오는 것으로서 시스템의 입력이나 출력이 된다. 따라서 본 문제에서는 로봇의 8방향에서 들어오는 센서의 입력정보(물체와의 거리)와 전진, 좌회전, 우회전 등 3가지의 출력 값을 사용한다.

즉, 8방향의 센서 정보 S0 ~ S7과 MOVE, RIGHT, LEFT이다. 이외에도 충돌 감지 센서가 있지만 프로그램에는 표시하지 않는다.

(2) 함수의 집합(Function set) 결정

함수는 나무의 자식가지(sub tree) 사이의 관계를 나타내주는 것으로서 다음의 6가지를 사용한다.

- IF_LTE (If Less Than Equal) - 자식가지 4개를 가지며 첫 번째 가지의 입력 값과 두 번째 가지의 입력 값을 비교하여 전자가 작거나 같으면 세 번째를 그렇지 않으면 네 번째 가지를 실행한다.
- IF_RBE (If Robot or Box Exist) - 만약 첫 번째 입력 값에서 박스나 로봇이 감지되면 두 번째 가지를 그렇지 않으면 세 번째 가지를 실행한다.
- IF_BMP (If BuMPed) - 현재 박스와 충돌 중이라면 첫 번째 가지를 그렇지 않으면 두 번째 가지를 실행한다.
- IF_STK (If STRuck) - 현재 벽이나 다른 로봇과 충돌 중이라면 첫 번째 가지를 그렇지 않으면 두 번째 가지를 실행한다.
- PROG2, PROG3 - 자식가지를 차례로 실행한다. PROG2는 자식가지를 2개 PROG3는 3개를 가진다.

(3) 적합도 함수(Fitness function)의 결정

우선 로봇이 상자의 앞까지 와야 하므로 상자 앞에서 로봇까지의 거리를 적합도로 한다. 그리고 상자가 움직여 벽에까지 닿아야 하므로 상자와 가장 가까운 벽까지의 거리를 적합도에 더해준다. 적합도의 값이 작을수록 우수한 프로그램이며 이에 비례하여 다음세대의 개체를 선택한다.

(4) 알고리즘 파라메타의 결정

알고리즘의 파라메타로 개체군(Population)의 크기(M), 교차율(cr: Crossover rate), 돌연변이율(mr: Mutation rate), 역위율(ir: Inversion rate)등이 있으며 다음과 같이 정하였다.

$$M = 1000, cr = 1, mr = 0.5, ir = 0.5$$

(5) 종결조건 결정

상자를 벽까지 밀어낸 프로그램이 발견되면 멈춘다. 또는 세대가 100세대가 넘어도 찾지 못하면 다시 개체군을 구성하여 진화를 시작한다.

4.3 시뮬레이션 환경

상자 한 번의 길이는 로봇의 폭의 3배로 하였으며 최소한 2대가 밀어야 상자가 움직일 수 있다. 로봇은 3대로 하였고 3대 모두 기능은 동일하고 사방 8방향의 센서와 충돌 감지센서 및 방향 전환을 할 수 있도록 모터를 조절할 수 있다.

4.4 시뮬레이션 결과

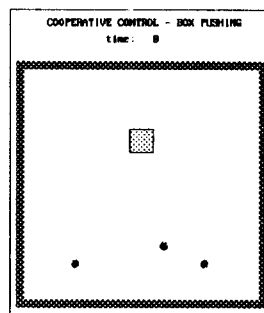


그림5. 초기 상태

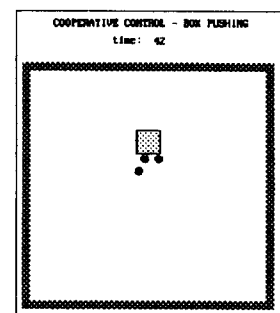


그림6. 중간 상태

5. 물체 운반 문제

5.1 문제의 설정

초기 조건은 길다란 물체를 3대의 로봇이 나란히 들고 있는 경우이며 주어진 경로를 따라 나란히 이동하여 반대쪽까지 가는 것이 목표이다. 이 문제는 여러 로봇들의 장애물 회피 및 병렬 주행의 행동 문제이다.

5.2 문제 해결을 위한 준비

문제를 유전 프로그래밍에 적용하기 위한 다섯 단계의 준비중 4절 내용과 다른 것만 언급한다.

(1) 말단 기호의 집합(Terminal set) 결정

S0, S1, S2(3방향의 센서, 각각 前方, 前友方, 前左方)

MOVE, RIGHT, LEFT, STOP.

SM(= 200) SS(= 50)

(2) 함수의 집합(Function set) 결정

함수는 나무의 자식가지(sub tree) 사이의 관계를 나타내주는 것으로서 다음의 4가지를 사용한다.

- IF_LTE (If Less Than Equal)
- IF_ROB (If Robot Exist) - 센서에 로봇이 감지되면 참
- PROG2
- PROG3

(3) 적합도 함수의 결정

여러 대의 로봇이 물체를 운반하므로 떨어뜨리거나 벽에 부딪히지 않고 운반한 거리를 적합도로 정한다. 여기서는 적합도의 값이 클수록 우수한 프로그램이며 이에 비례하여 다음세대의 개체를 선택한다.

(4) 알고리즘 파라메타의 결정

알고리즘의 파라메타는 다음과 같이 정하였다.

$$M = 1000, cr = 1, mr = 0.5, ir = 0.5$$

(5) 종결조건의 결정

경로를 따라 끝까지 무사히 물체를 운반하면 종료한다. 또는 세대가 100세대가 넘어도 찾지 못하면 다시 개체군을 구성하여 진화를 시작한다.

5.3 시물레이션 결과

그림7과 같은 경로를 3대의 로봇이 물체를 운반한다.

벽을 따라 나란히 진행해가는 개체를 찾는 것을 목적으로 하였다.

100세대 진화 후 얻어진 결과의 프로그램 및 결과는 아래와 같다.

```
(IF_ROB (IF_LTE (S2 S0 IF_LTE (S0 S1 PROG2
(RIGHT LEFT ) IF_LTE (PROGN3 (S0 S0 S1 ))SS IF_LTE
(SM S2 STOP MOVE) STOP)) LEFT) IF_LTE (S0 SM
IF_LTE (S0 S1 RIGHT LEFT )STOP )))
```

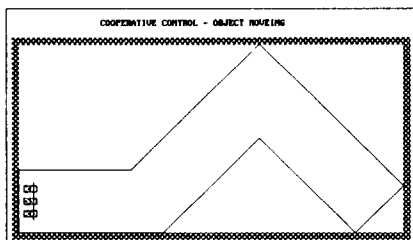


그림7. 초기 상태의 작업환경

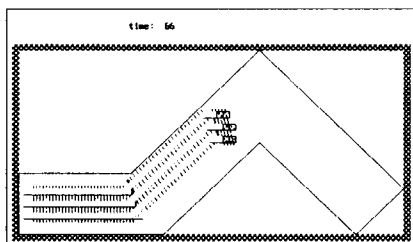


그림8. 중간 상태 및 경로

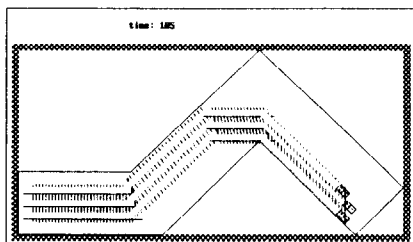


그림9. 종료시의 상태

6. 결론 및 추후 과제

여러 대의 자율이동로봇을 가지고 군지능을 발현시키기 위해서는 우선 로봇들의 군행동이 나타나야 한다. 집단을 이루고 있는 곤충에서 볼 수 있듯이 단순한 행동에서 출발하여도 군행동 그리고 군지능을 발현 할 수 있으며, 본 논문에서도 이러한 군지능이 발현되기 위한 군행동으로서 간단한 예제를 제안한 방식을 통하여 구현해 봄으로서 이러한 연구의 한가지 방법이 됨을 보였다. 본 논문에서 제안한 유전 프로그래밍을 이용한 방법은 다윈의 진화론에 바탕을 두고 점점 더 나은 개체를 찾아낸다는 점에서 다른 방법에 비해 프로그래머의 수고를 덜 수가 있으며 개체 하나하나의 규칙을 생각해내기 힘든 행동도 구현 할 수가 있다. 여기에서도 기본적으로 주변의 센서에 의해 감지한 정보만을 이용해 목적을 이루었으며 중앙 관리자는 두지 않았다. 그러나 이 방법에도 아직 개선의 여지는 많이 남아 있다. 특히 군행동에 대해 적합한 정도를 판단하기가 어려운 경우와 작업이 복잡하고 여러 가지 일을 동시에 해야 할 경우는 이 방법만으로는 부족한 점이 있을 것이다. 이때는 어느 정도의 기본적인 규칙은 프로그래머가 기본으로 작성하고 나머지 부분을 진화시키거나 또는 퍼지 이론이나 카오스 이론 등의 결합도 고려해 볼 만하다. 또한 보다 효율적이고 깔끔한 프로그램을 생성하기 위해서는 유전 프로그래밍에 대한 연구도 더 이루어져야 될 것이다. 여러 가지의 군행동과 군지능의 구현에 대한 시물레이션에 의한 검증과 실제 하드웨어 실험의 병행도 필요하다고 생각된다.

참고문헌

- 1] 浅間一, “複数の移動ロボットによる協調行動と群知能,” 計測と制御, Vol. 31, No. 11, pp. 1155-1161, 1992.
- 2] 上田完次, “人工生命の方法,” 工業調査會, 1995.
- 3] 장병탁, “유전 알고리즘의 이론 및 응용”, 대한전자공학회지, Vol. 22, No. 11, pp. 1290-1300, 1995.
- 4] C. R. Kube and H. Zhang “Collective Robotic Intelligence,” Proc. of 2nd International Conference on Simulation of Adaptive Behavior, pp. 460-468, 1993.
- 5] ジェラルド ベニ “自律分散ロボットシステムと群知能,” 日本ロボット學會誌, vol. 10, No. 4, pp. 457-463, 1992.
- 6] Fukuda, “動的再構成可能ロボットシステムに関する研究,” 日本機械學會論文誌(C編), Vol. 59, No. 568, pp. 3837-3844, 1993.
- 7] M. J. Mataric, “Designing Emergent Behaviors: From Local Interactions to Collective Intelligence,” Proc. of 2nd International Conference on Simulation of Adaptive Behavior, pp. 432-441, 1993.
- 8] 堀内英一, 谷和男, “分散実行可能な遺伝的アルゴリズムを用いた移動ロボット群の行動学習,” 日本ロボット學會誌, vol. 11, No.4, pp. 1212-1219, 1992.
- 9] 浅間一, “群ロボットシステムにおける創発の實現,” 計測と制御, Vol. 35, No. 7, pp. 545-549, 1996.
- 10] R. J. Collins and D. R. Jefferson, “AntFarm: Towards Simulated Evolution,” Artificial Life II, vol. X, Addison-wesley, pp. 579-601, 1991.
- 11] J. R. Koza, “Genetic Programming: On the Programming of Computers by Means of Natural Selection,” MIT PRESS, 1992.