사물인터넷융복합논문지 Vol. 12, No. 4, pp. 171-176, 2026 https://doi.org/10.20465/KIOTS.2026.12.4.171

# 동적 장애물 회피를 위한 유전자 알고리즘 구현 및 성능분석

**이면재***
**백석대학교 컴퓨터공학부 교수**

# Implementation and Performance Analysis of a Genetic Algorithm for Dynamic Obstacle Avoidance

MyounJae Lee*
Professor, Division of Computer Engineering, BaekSeok University

**요 약** 본 연구는 게임 환경 내 비플레이어 캐릭터(NPC)의 자율적인 경로 탐색 및 동적 장애물 회피 성능을 향상시키기 위해 유전 알고리즘(Genetic Algorithm)을 적용하였다. 기존 단일 유전자 구조가 유발하는 형질 간 간섭 문제를 해결하고자 유전자를 배회, 목표, 위험 형질로 분리하여 설계하였다. 본 모델의 성능을 검증하기 위해 유니티(Unity) 2D 환경에서 우성 유전자 선택 비율, 돌연변이율, 총 학습 세대수를 변인으로 설정하여 실험을 수행하였다. 이를 통해 각 파라미터의 변화가 에이전트의 목표 도달 횟수와 소멸 횟수에 미치는 영향을 비교 분석하였다. 본 연구는 복잡한 지형에서 유전 알고리즘의 파라미터 제어가 NPC의 적응력을 극대화할 수 있다는 실증적 근거를 제공한다.

**주제어 :** 유전 알고리즘, 최적화, 세대별 진화, 수렴 성능, 적합도 분석, 경로 탐색

**Abstract** This study applied a Genetic Algorithm (GA) to improve the autonomous pathfinding and dynamic obstacle avoidance performance of Non-Player Characters (NPCs) in a game environment. To address the trait interference problem caused by the conventional single-gene structure, the genes were designed by separating them into wandering, targeting, and risk-avoidance traits. To verify the performance of this model, experiments were conducted in a Unity 2D environment by setting the dominant gene selection rate, mutation rate, and total learning generations as variables. Through this, the impact of each parameter's variation on the agent's target arrival count and extinction count was comparatively analyzed. This study provides empirical evidence that parameter control of genetic algorithms can maximize the adaptability of NPCs in complex terrains.

**Key Words :** Genetic Algorithm (GA), Path Optimization, Generational Evolution, Convergence Analysis Fitness Function

## 1. 서론

게임 콘텐츠가 고도화됨에 따라 비플레이어 캐릭터(NPC)의 지능적 상호작용이 필수적으로 요구된다. 기존에 주로 활용된 유한 상태 기계(FSM)와 사전 정의된 스크립트 기반 제어 방식은 실시간으로 지형과 장애물이 변하는 동적 환경에서 NPC의 자율성을 보장하는 데 한계가 있다[1, 2]. 전통적인 탐색 기법인 A* 알고리즘이나

본 논문은 2026학년도 백석대학교 학술연구비 지원을 받아 작성되었음
*교신저자 : 이면재(davidlee@bu.ac.kr)
접수일 2026년 07월 17일 수정일 2026년 08월 07일 심사완료일 2026년 08월 21일

다익스트라(Dijkstra) 알고리즘 역시 정적 환경에서는 효율적이나, 이동 궤적이 급변하는 상황에서는 연산 지연 및 환경 적응력 저하를 초래한다[3]. 이를 극복하기 위한 대안으로 진화 연산의 일종인 유전 알고리즘(Genetic Algorithm, GA)을 적용하여 NPC의 자가 학습을 유도하는 방식이 연구되고 있다[4, 5]. 유전 알고리즘은 비정형 탐색 공간에서 최적해를 찾는 데 유리하며[6], 적합도 함수를 통해 비선형적 공간 내 NPC의 상황 판단력을 높일 수 있다[7].

그러나 기존의 유전 알고리즘을 활용한 경로 탐색 연구는 주로 정적인 미로 탐색이나 고정된 장애물 회피에 머물러 있다[8]. 센서 데이터 밀도와 진화 성능 간의 상관성 분석이 부족하며[9], 최근 유니티 엔진과 유전 알고리즘을 결합한 퍼즐 게임 등 일부 응용 사례[10]가 등장하였음에도, 상용 엔진의 물리 시스템과 연동한 실시간 학습 연구 또한 미흡한 실정이다[11].

본 연구의 목적은 유니티(Unity) 2D 환경에서 레이캐스트(Raycast) 센서와 유전 알고리즘을 결합하여, 동적 장애물을 회피하고 최적의 궤적으로 동적 타겟에 도달하는 자율 제어 NPC(에이전트) 모델을 구현하는 것이다. 또한 유전 알고리즘의 핵심 파라미터가 최적 경로 수렴 및 탐색 성능에 미치는 영향을 정량적으로 규명한다. 이전의 선행 연구[12]는 유니티 엔진에서 유전 알고리즘을 활용한 경로 탐색 최적화를 입증하였으나, 동적 장애물과 목적지가 혼재된 복잡한 환경에서는 이러한 단순 배열이 진화적 병목을 유발할 위험이 있다. 후속 연구[13]에서는 개체의 유전자를 배회, 목표, 위험 형질로 독립 분리하여 구조적 한계를 개선하고 파라미터 제어에 따른 수렴도를 분석하였다. 본 연구는 이들 선행 연구[12, 13]를 확장하여, 동적 장애물과 동적 목표물이 혼재된 환경으로 모델을 고도화하였다. 특히 이러한 동적 탐색 과정에서 우성 유전자 선택 비율, 돌연변이율, 세대수의 변화가 에이전트의 소멸 횟수와 목표 도달 횟수에 미치는 영향을 정량적으로 규명한다는 점에서 명확한 차별성을 갖는다.

이를 위해 동적 및 정적 장애물이 혼재된 '2D Beginner: Adventure Game'[14]를 실험 환경으로 채택하였다. 탐색 성능에 영향을 미치는 선택률(Selection rate), 돌연변이율(Mutation rate), 세대 수를 독립 변인으로 설정하였다. 목표 지점까지의 거리, 생존 시간, 최종 도달 여부를 평가하는 적합도 함수를 구성한 뒤, 각 시나리오당 30회 반복 수행하여 성능을 분석하였다.

본 연구는 복잡한 지형에서 유전 알고리즘의 자율적 학습 효과를 입증하고, 유전자 구조 재설계 및 파라미터 조절을 통해 NPC의 생존율과 목적 도달률을 동시에 높였다는 데 의의가 있다. 이는 향후 NPC의 환경 적응력을 극대화하는 차세대 인공지능 설계 방향을 제시하는 기초 자료로 활용될 수 있다.

본 논문의 구성은 다음과 같다. 제2장에서는 유전 알고리즘과 경로 탐색에 관한 관련 연구를 고찰한다. 제3장에서는 유니티 엔진 기반의 모델 설계, 유전자 인코딩 방식 및 구현 과정을 설명하고 실험 결과를 비교 분석한다. 마지막으로 제4장에서는 본 연구의 결과를 종합하여 결론을 도출하고 향후 연구 과제를 제시한다.

## 2. 유전 알고리즘

유전 알고리즘은 자연선택과 유전자 변이 과정을 기반으로 최적해를 도출하는 메타 휴리스틱 알고리즘이다[2]. 이 기법은 임의로 구성된 초기 개체군을 대상으로 문제 해결 능력을 평가하여 적합도가 높은 개체를 선별한 뒤, 교차 및 돌연변이 연산을 통해 새로운 세대를 생성한다. 이러한 반복 과정을 거침으로써 탐색 과정에서 지역 최적해에 갇히는 문제를 완화하며, 비선형적이고 복잡한 탐색 공간에서 우수한 성능을 보인다. NPC의 인공지능 제어 영역에서 유전 알고리즘의 활용성은 뚜렷하게 나타난다. 과거 주로 채택되었던 유한 상태 기계와 같은 스크립트 기반 방식은 실시간으로 변화하는 환경에 대한 유연성이 부족하고 행동 패턴이 단조롭다는 명확한 한계가 존재한다[1]. 이를 보완하기 위해 유전 알고리즘을 NPC 설계에 적용할 경우, 지형이나 동적 장애물의 변화에 맞춰 이동 경로와 속도를 스스로 최적화할 수 있다. 또한 플레이어의 행동 양식에 대응하여 다채로운 전투 전술을 자율적으로 학습하는 것이 가능하다[1].

기존 유전 알고리즘을 활용한 연구에는 축구 게임 시뮬레이션 내 캐릭터의 상황별 행동 패턴 생성[15], 환경 요인이 수시로 변하는 공간에서의 동적 경로 탐색[16], 그리고 인공신경망과 결합한 롤플레잉 게임(RPG) 캐릭터의 의사결정 학습[17] 등이 있다. 최근에는 적합도 함수를 설계하여 기능성 게임의 난이도를 사용자의 수준에 맞춰 동적으로 조절하거나[18], 복잡한 퍼즐 게임 환경에서 최적의 해결 경로를 산출하는 연구[19]로 그 영역이 확장되고 있다. 결론적으로 유전 알고리즘은 사전 정의된 규칙에 얽매이지 않고 환경과의 지속적인 상호작용을 통해 행동 양식을 진화시킴으로써, 게임 내 생태계의 사

실성과 플레이어의 몰입감을 향상시키는 핵심 기술로 평가받는다.

## 3. 구현

### 3.1 유전자 구조

유전자 구조는 유전 알고리즘의 교차, 돌연변이, 선택 연산의 효율을 결정하는 핵심 요소이다. 복잡한 환경일수록 개체의 각 행동 특성을 독립적으로 제어하고 우성 형질을 보존할 수 있는 체계적인 유전자 설계가 필수적이다.

선행 연구[12]의 유전자 구조는 이동 행동을 단일 변수(하나의 각도와 속도)로 묶어 제어할 경우, 진화 연산(교차 및 돌연변이) 과정에서 위험 회피를 위해 발생한 형질 변이가 목표 접근에 필요한 우성 형질을 훼손하는 유전자 충돌 현상이 발생한다. 이를 개선하기 위해 연구[13]에서는 세 가지 독립된 형질로 분리하여 각 인지 상태별로 독립된 행동 변수를 호출하게 함으로써, 유전자 간의 논리적 간섭을 원천적으로 차단하고 복잡한 환경에서 에이전트의 생존율과 목적 도달률을 동시에 향상시킬 수 있다.

### 3.2 구현

〈Table 1〉은 연구[14]에서 제안된 것으로 유전 알고리즘의 과정을 나타낸다. 먼저 임의의 속성을 지닌 초기 개체군을 생성한 후(단계 (1)), 매 세대마다 각 개체의 환경 적응 수준을 적합도 함수를 통해 평가한다(단계(2)). 이후 새로운 세대를 구성하기 위해 토너먼트 선택(Tournament Selection) 기법을 적용하여 적합도가 우수한 부모 개체를 선별한다(단계 (6)).

〈Table 1〉 GA Algorithm[13]

```
PROCEDURE ReproductionAndReplacement
01:  WHILE Termination_Condition_is_NOT_Met DO
02:   Next_Population ← Empty List
03:  WHILE size(Next_Population) 〈 size(Current_Population)
04:      Child ← Crossover(Selected_Parents, Crossover_Rate)
05:      Child ← Mutation(Child, Mutation_Rate)
06:      Next_Population.Add(Child)
07:   END WHILE
08   Current_Population ← Next_Population
09:      END WHILE
10: END PROCEDURE
```

선별된 부모 개체는 정해진 비율에 따라 교차 연산을 거쳐 자식 개체를 생성하며(단계(7)), 탐색 과정에서 지역 최적해에 갇히는 문제를 예방하기 위해 무작위 돌연변이 연산을 추가로 적용한다(단계(8)). 이러한 유전 연산을 거쳐 생성된 신규 자식 개체들의 수가 기존 집단의 크기와 동일해지면 기존 개체군을 완전히 대체하는 세대 교체가 이루어지며(단계(9)) 이 전체 주기는 알고리즘의 종료 조건이 충족될 때까지 지속된다.

식 (1)은 적합도 함수이다. 적합도 함수는 개체의 최종 상태(성공, 사망, 단순 생존)에 따라 보상을 차등 부여한다. 목표 도달 개체는 시간 비례 가점을 포함해 최고점을 받으며, 위험 구역 충돌 개체는 최하점을 받아 다음 세대에서 도태된다. 단순 생존 개체는 안전 구역에만 머무르는 지역 최적해 현상을 막기 위해, 목표 접근성(0.7)과 위험 회피성(0.3)을 가중 합산하여 최종 적합도를 산출한다. t는 개체의 생존 시간, G는 개체와 목표 지점 간의 거리를 역산하여 목표에 근접할수록 1에 수렴하도록 정규화한 값이다. H는 에이전트와 위험지역까지 거리를 세션 시작시의 초기 위치로부터 위험지역까지의 거리로 나누어 계산한 값이다. 즉 위험지역으로 거리가 멀어질수록 큰 값을 갖는다.

$$F=\begin{cases}10.0+5.0(1-t), & if\ \ S=SUCCESS\\ 0.001+0.1t, & if\ \ S=DEAD\\ 5.0+0.7G+0.3H, & otherwise\end{cases} \quad (1)$$

〈Table 2〉는 에이전트의 Move() 함수를 보여준다. 에이전트는 주변의 위험 지역과 목표를 탐색한다(단계(1), 단계(2). 위험지역(목표)이 탐지되는 경우 위험지역(목표)에 해당하는 인덱스를 계산한다(단계(3)-(4)). 위험지역과 목표가 모두 인지 범위를 벗어난 경우(단계(6)), 염색체의 중앙 인덱스로 지정한다. 이러한 과정을 거쳐 최종적으로 선택된 유전자의 각도 및 속력 파라미터를 가지고 이동한다. 식 (2)는 단계(4)와 단계(5)에서의 인덱스를 구하기 위한 수식이다. N은 에이전트의 유전자(염색체) 크기를 말하고 $d_{goal}(d_h)$는 센서를 통해 감지된 목표(위험지역)와의 실제 최단 거리이다. $d_{\max}$는 에이전트가 객체를 감지할 수 있는 센서의 최대 탐색 거리이다. 따라서 $1-\frac{d_{goal(h)}}{d_{\max}}$는 0과 1사이로 정규화하는 과정으로 목표(위험지역) 와 가까울수록 1에 수렴하고, 멀어질수록 0에 수렴하도록 정규화된 값을 역산하여 부여한다. 이 값의 결과에 N-1을 곱한 값과 유전자의 개수 중에서 최

소의 값을 유전자 인덱스로 설정한다.

$$i = \max\left(0, \min\left(N-1, \left\lfloor \left(1 - \frac{d_h}{d_{\max}}\right)(N-1) \right\rfloor\right)\right) \quad (2)$$

〈Table 2〉 Move

```
PROCEDURE Move_Agent(){
 01: Dist_Hazard = Scan_Nearest_Distance(Hazard_Zone)
 02: Dist_Goal = Scan_Nearest_Distance(Goal_Zone)
 03: IF Dist_Hazard 〈 Detection_Range THEN
 04:    Gene=Chromosome[Calc_Index(Dist_Hazard)].Hazard
 05:  ELSE IF Dist_Goal 〈 Detection_Range THEN
        Gene = Chromosome[Calc_Index(Dist_Goal)].Goal
 06:  ELSE    Gene = Chromosome[Middle_Index].Default
 07:  END IF
 08:  Apply_Movement(Gene.Angle, Selected_Gene.Speed)
END PROCEDURE
```

### 3.3 결과

본 연구의 실험은 Intel Core i7, 16GB RAM, 유니티6.4 환경에서 진행되었다. 유전 알고리즘의 성능을 객관적으로 평가하기 위해서 우성 선택률, 돌연변이율, 세대수를 다양하게 설정한다.

〈Table 3〉은 실험 파라미터를 보여준다. Base 그룹은 기준 집단으로 연구[20]를 참고하였다. S1과 S2 S3는 우성 유전자 선택 비율, M1과 M2 M3 그룹은 돌연변이율, G1과 G3, G4는 세대수를 각각 다르게 설정하여 수렴 정도를 확인한다. 한 세대의 실험 시간은 시작 지점에서 목표까지 도착하기에 충분한 60초로 설정한다. 우성 유전자 선택 비율은 이전 세대에서 우성 유전자가 다음 세대로 복사되는 비율이다. 0.2의 경우 전체유전자중에서 상위 20%가 복사된다는 의미이다. 돌연변이율은 우성 유전자를 제외한 다른 유전자들의 값들을 무작위로 설정하는 비율로 돌연변이율이 0.05라는 것은 새로운 개체가 생성될 때 염색체를 구성하는 각 유전자가 5%의 확률로 무작위 변형됨을 의미한다.

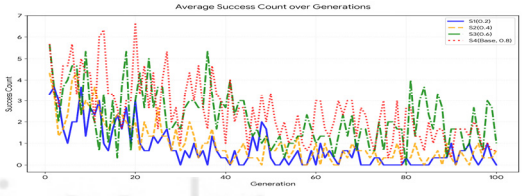
〈Table 3〉 Experimental Parameter

| Group | Selection rate | Mutation rate | Generations |
|---|---|---|---|
| S1 | 0.2 | 0.05 | 100 |
| S2 | 0.4 | 0.05 | 100 |
| S3 | 0.6 | 0.05 | 100 |
| Base | 0.8 | 0.05 | 100 |
| M1 | 0.8 | 0.1 | 100 |
| M2 | 0.8 | 0.2 | 100 |
| M3 | 0.8 | 0.4 | 100 |
| G1 | 0.8 | 0.05 | 50 |
| Base | 0.8 | 0.05 | 100 |
| G3 | 0.8 | 0.05 | 200 |
| G4 | 0.8 | 0.05 | 400 |

각 실험군은 세션 시간 종료 시점에서의 목표 도착 갯수(성공 개수), 데미지존에 도착한 개수(죽은 개수)를 비교 분석한다. 무작위성에 의한 오차를 최소화하기 위해 각 실험군은 30회 독립 수행을 거쳐 평균값으로 계산한다. [Fig. 1][14]은 시작점과 목표 지점 그리고 데미지존, 나무 등의 장애물을 보여준다. 데미지존에 에이전트가 도착하면 에이전트는 죽게 된다. 데미지존을 목표 지점보다 앞에 위치시켜서 세대수가 거듭될수록 에이전트가 데미지 존을 피해 가는 여부를 알 수 있도록 하였다. 데미지존과 목표 지점의 이동 속도는 에이전트들의 평균 이동속도의 1/10로 설정하여 에이전트들이 충분히 데미지존과 목표 지점에 도달할 수 있도록 한다.

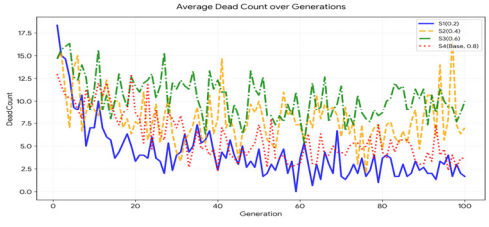

[Fig. 1] Starting Point, Target Point, Damage Zone

[Fig. 2]는 우성 인자 선택 비율에 따른 성공 횟수 비교이다. 이 비율이 커질수록 전반적 성공 빈도가 높게 나타남을 확인하였다. 파라미터가 가장 낮은 S1(0.2) 그룹은 초기 세대 이후 성공 개수가 1 미만으로 급감하며 가장 낮은 성과를 보였다. 반면 파라미터가 0.8로 설정된 S4(Base, 0.8) 그룹은 세대 진행 과정에서 변동성은 존재하나, 다른 실험군(S1, S2, S3) 대비 일관적으로 우수한 성공 개수를 유지한다. 이는 에이전트의 행동 결정 과정에서 해당 파라미터가 목표 지향성에 핵심적인 가중치로 작용하고 있음을 시사한다.



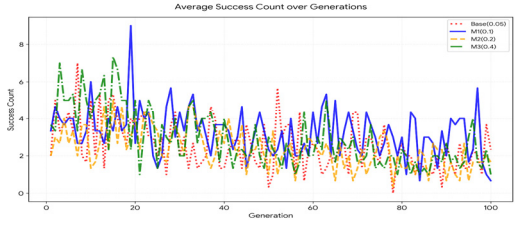

[Fig. 2] Success Count by Selection Rate

[Fig. 3]은 우성 유전자 선택 비율에 따른 죽은 개수 비교이다. 우성 유전자 선택 비율이 가장 낮은 S1(0.2) 그룹이 세대 진행에 따라 소멸 횟수가 가장 급격히 감소하여 5 미만의 낮은 수치에서 안정화됨을 확인하였다. 반면, S3(0.6) 그룹은 전 세대에 걸쳐 가장 높은 소멸 수치를 유지하며 불안정한 변동성을 보였다. 기준 모델인 S4(Base, 0.8) 그룹은 S1과 S3의 중간 수준에서 하향 안정화되는 추세를 나타낸다. 그러나 앞선 평균 성공 개수 지표에서 S1 그룹의 목표 도달률이 가장 낮았음을 고려할 때, 이는 에이전트의 환경 적응력이 높아진 것이 아니라 목표 지점 탐색을 포기하고 위험 회피(단순 생존)에만 머무르는 지역 최적에 빠졌음을 의미한다. 반면 Base(0.8) 모델은 적정 수준의 생존율과 높은 성공률을 동시에 확보하여 가장 균형 잡힌 진화 양상을 보여준다.



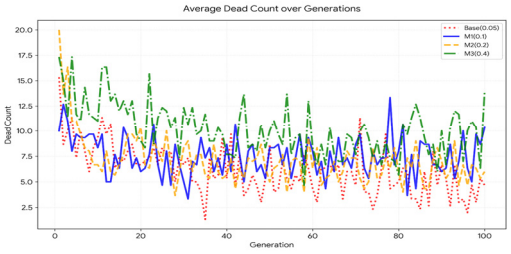

[Fig. 3] Dead Count by Selection Rate

[Fig. 4]는 돌연변이율 변화에 따른 성공 개수 비교이다. 비교 결과 0.1로 설정된 M1 그룹이 가장 안정적인 목표 도달률을 기록하였다. 돌연변이율이 0.4(M3)로 과도하면 우성 유전자가 훼손되어 목표 지향성을 상실하고, 0.05(Base)로 지나치게 낮은 경우 성공 개수가 적었다. 따라서 해당 시뮬레이션 환경에서는 돌연변이율 0.1이 우성 유전자 보존과 환경 적응을 위한 탐색 사이의 최적 균형을 이루는 수치로 확인된다.



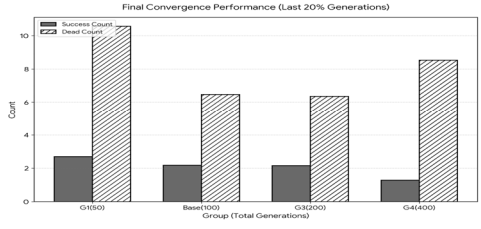

[Fig. 4] Success Count by Mutation Rate

[Fig. 5]는 돌연변이율 변화에 따른 죽은 개수 비교이다. Base(0.05) 모델은 전체 세대 구간에서 가장 낮은 평균 죽은 개수(약 6.16)를 기록하여 생존성 측면에서 우수성을 입증하였다. 반면 M3(0.4) 그룹은 평균 10 이상의 높은 소멸 수치를 기록하였다. 이는 돌연변이 확률이 증가할수록 에이전트가 장애물을 회피하지 못하고 충돌하여 소멸하는 빈도가 높아짐을 직관적으로 보여준다.




[Fig. 5] Dead Count by Mutation Rate

[Fig. 6]은 세대수의 변화에 따른 결과로 제시된 세대수의 마지막 20% 구간을 분석한 결과이다. 세대수가 가장 짧은 G1(50)은 목표 도달 빈도(약 2.7)가 가장 높으나 극심한 에이전트 소멸 비용을 수반한다. 반면, 가장 긴 G4(400)는 장시간 진화로 인한 다양성 감소나 지역 최적화로 인해 성공 개수(약 1.28)가 크게 하락하고 죽은 개수도 상승한다. 성공 개수와 죽은 횟수를 종합할 때, 생존율을 보장하며 유의미한 탐색 성공을 유도하는 최적의 시뮬레이션 종료 지점은 100~200세대(Base, G3)로 판단된다.




[Fig. 6] Success Count, Dead Count by Generation

## 4. 결론 및 추후 연구방향

연구는 유니티 2D 환경에서 유전 알고리즘을 적용하여 동적 장애물을 회피하고 목표 지점에 도달하는 자율제어 NPC 모델을 구현하고, 주요 파라미터 변화가 탐색 성능에 미치는 영향을 분석하였다. 기존 모델의 한계를 극복하기 위해 유전자 구조를 배회, 목표, 위험 형질로 분리 설계하여 유전자 간 간섭 및 충돌 현상을 방지하였다.

실험 결과, 우성 유전자 선택률이 0.8로 설정될 때 가

장 안정적인 목표 도달률을 유지하였으며, 돌연변이율은 0.1 내외일 때 탐색 효율과 생존율 사이의 최적 균형을 형성함을 확인하였다. 또한, 전체 학습 세대 수는 100~200세대 구간이 개체 군집의 생존율과 탐색 성공률을 동시에 극대화하는 최적의 종료 지점임이 규명되었다. 본 연구는 복잡한 동적 환경에서 유전 알고리즘의 자율적 학습 효과를 실증하고, 독립적인 유전자 설계 및 파라미터 최적화를 통해 NPC의 생존율과 목적 도달률을 성공적으로 향상시켰다는 데 의의가 있다.

추후 연구 방향은 시뮬레이션의 진행 상태나 환경의 복잡도 변화에 따라 선택률과 돌연변이율이 실시간으로 조절되는 적응형 동적 파라미터 제어 기법을 연구할 예정이다.

## REFERENCES


[1] I. Millington and J. Funge, Artificial Intelligence for Games, 2nd ed., CRC Press, 2009.

[2] G. N. Yannakakis and J. Togelius, Artificial Intelligence and Games, Springer, 2018.

[3] N. R. Sturtevant, "Benchmarks for Grid-Based Pathfinding," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 2, pp. 144-148, 2012.

[4] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[5] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.

[6] M. Buckland, Programming Game AI by Example, Wordware Publishing, 2004.

[7] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," Evolutionary Computation, vol. 10, no. 2, pp. 99-127, 2002.

[8] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-Based Procedural Content Generation: A Taxonomy and Survey," IEEE Transactions on Computational Intelligence and AI in Games, vol. 3, no. 3, pp. 172-186, 2011.

[9] S. M. Lucas, "Computational Intelligence and Games," 2008 IEEE Symposium on Computational Intelligence and Games, 2008.

[10] Matthew McConnell, Richard Zhao, "From Frustration to Fun: An Adaptive Problem-Solving Puzzle Game Powered by Genetic Algorithm," Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, pp.277-286, 2025.

[11] J. K. Haas, "A History of the Unity Game Engine," M.S. thesis, Worcester Polytechnic Inst., 2014.

[12] MyounJae Lee, "Performance and Evolutionary Characteristics of Genetic Algorithm-based Optimal Pathfinding for Genetic Algorithm NPCs in Unity Engine," Journal of KIOTS, Vol.12, No.3, pp.61-66, 2026.

[13] MyounJae Lee, "Analysis of Autonomous Exploration Performance and Convergence of Game NPCs Based on Genetic Algorithm Parameter Control," Journal of KIOTS, Vol.12, No.4, 2026(Submitted for publication).

[14] Unity Technologies. "2D Beginner: Adventure Game." Unity Learn, 2020.

[15] H. S. Roh and D. W. Lee, "A Study on The Game Character Creation Using Genetic Algorithm in Football Simulation Games," Journal of Korea Game Society, vol. 17, no. 6, pp. 129-138, 2017.

[16] J.W Ko and D.Y Lee, "Heuristic-based Genetic Algorithm," Journal of Korea Game Society, vol. 17, no. 5, pp. 129-138, 2017.

[17] O. K. Kwon and J. G. Park, "Control of RPG Game Characters using Genetic Algorithm and Neural Network," Journal of Korea Game Society, vol. 6, no. 2, pp. 13-22, 2006.

[18] V. Vaghani and K. Oyibo, "A Scoping Review of Genetic Algorithms in Serious Games: Applications, Challenges, and Future Directions," Preprints.org, 2024.

[19] N. H. S. Hasibuan et al., "Using Genetic Algorithm to Solve Puzzle Games: A Review," Journal of Computer Networks Architecture and High Performance Computing, 2024.

[20] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. Thesis, University of Michigan, Ann Arbor, 1975.



이 면 재(MyounJae Lee) [종신회원]

■ 2009년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

〈관심분야〉
사물인터넷, 게임, MPEG