사물인터넷융복합논문지 Vol. 12, No. 3, pp. 61-66, 2026 https://doi.org/10.20465/KIOTS.2026.12.3.061

유니티 엔진 기반 게임 NPC의 유전 알고리즘 최적 경로 수렴 성능 및 진화 특성 분석

이면재*
백석대학교 컴퓨터공학부 교수

Performance and Evolutionary Characteristics of Genetic Algorithm-based Optimal Pathfinding for Game NPCs in Unity Engine

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

요 약 본 논문은 게임 NPC의 경로 탐색 효율을 극대화하기 위해 유전 알고리즘의 최적 경로 수렴 성능과 진화 특성을 분석한다. 이를 위해 유니티 엔진 환경에서 타겟 거리, 생존 시간, 도착 여부를 반영한 적합도 함수를 설계하여 총 90세대에 걸친 진화 연산 시뮬레이션을 수행하였다. 분석 결과, 세대가 반복될수록 최소 및 평균 거리가 하향 안정화되고 성공 개체수가 지속적으로 증가하며 안정적인 최적해로 수렴함을 확인하였다. 본 연구는 복잡한 환경에서 유전 알고리즘의 우수한 자율 학습 능력을 실증하였으며, 향후 적응형 파라미터 제어 모델로의 확장 가능성을 제시하였다.

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

Abstract This paper analyzes the optimal path convergence performance and evolutionary characteristics of genetic algorithms to maximize the pathfinding efficiency of game NPCs. To this end, an evolutionary computation simulation spanning a total of 90 generations was conducted within the Unity engine environment, utilizing a fitness function designed to reflect the distance to the target, survival time, and successful arrival. The analysis results confirmed that as generations progressed, both the minimum and average distances exhibited downward stabilization, while the number of successful individuals continuously increased, ultimately converging to a stable optimal solution. This study demonstrates the excellent autonomous learning capabilities of genetic algorithms in complex environments and presents the potential for future expansion into adaptive parameter control models.

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

본 논문은 2026학년도 백석대학교 학술연구비 지원을 받아 작성되었음
*교신저자 : 이면재(davidlee@bu.ac.kr)
접수일 2026년 04월 09일 수정일 2026년 06월 17일 심사완료일 2026년 06월 22일

1. 서론

최근 게임 산업의 성장에 따라 유저들은 NPC와의 상호작용에서 더욱 지능적이고 몰입감 있는 경험을 기대하고 있다. 기존의 유한 상태 기계(FSM)나 스크립트 기반 제어 방식은 동적이고 복잡한 환경에서 정형화된 행동 패턴만을 반복하는 한계가 있다[1]. 특히 실시간으로 지형이 변하거나 유동적인 장애물이 출현하는 어드벤처 게임에서는 고정형 알고리즘만으로 NPC의 유연한 대응과 생존력을 담보하기 어렵다[2].

경로 탐색에 널리 사용되어 온 A*나 다익스트라(Dijkstra) 알고리즘 역시 정적인 지도에 최적화되어 있어, 실시간으로 궤적이 변하는 동적 장애물에 대처하는 데 한계가 있다[3]. 이에 따라 생물학적 진화 원리를 차용한 유전 알고리즘(Genetic Algorithm,GA)를 게임 AI에 접목해 NPC의 자가 학습을 유도하는 연구가 활발히 진행 중이다[4]. GA는 광범위하고 비정형적인 환경에서 전역 최적해를 도출하는 데 탁월하며[5], 적합도 함수의 설계를 통해 행동 목표를 유연하게 제어할 수 있다[6]. 이러한 진화 메커니즘은 비선형적 게임 환경에서 NPC의 의사결정 능력을 극대화할 핵심 대안으로 평가받는다[7].

그러나 기존의 GA 기반 NPC 연구는 대부분 정지된 미로 탈출이나 고정 장애물 회피 시나리오에 편중되어 있다[8]. 또한 센서 데이터의 밀도가 진화 효율에 미치는 상관관계 분석이 미비하며[9], 유니티 등 상용 엔진의 물리 연산과 결합된 실시간 학습 연구는 아직 초기 단계이다[10]. 이에 본 연구는 유니티 2D 플랫폼을 바탕으로 레이캐스트(Raycast) 센서와 GA를 융합한 자율형 NPC 제어 시스템을 제안한다.

실험의 객관성을 위해 정적·동적 장애물이 공존하는 '2D Beginner: Adventure Game'[11] 프로젝트를 채택하여, NPC가 실시간 센싱으로 함정을 극복하고 목적지에 도달하는 메커니즘을 분석한다. 특히 장애물의 이동 패턴 변화가 세대별 적응도에 미치는 영향과 변이율의 조절이 최적 경로 수렴에 미치는 효과를 실험적으로 규명한다. 이를 통해 NPC의 자율성과 생존력을 극대화할 수 있는 차세대 인공지능 설계 방안을 제시하는 데 본 연구의 의의가 있다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 고찰하고, 제3장에서는 유니티 엔진 기반의 GA 알고리즘 설계 및 구현 과정을 기술한다. 마지막으로 제4장에서는 실험 결과를 바탕으로 결론을 도출하고 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 GA

GA는 복잡한 환경에서 장애물을 회피하고 최적 경로를 도출하는 자율형 NPC 제어에 탁월한 확률적 탐색 기법이다. 선행 연구들은 축구 시뮬레이션의 캐릭터 생성[12], 가변적 환경의 경로 탐색[15], 신경망 결합을 통한 RPG 캐릭터 학습[17] 등 다방면으로 GA를 응용해 왔다. 최근에는 적합도 함수를 활용한 기능성 게임의 적응형 난이도 조절[18]과 퍼즐 게임의 효율적인 최단 경로 탐색[19] 등 동적 환경 적응 중심으로 발전하고 있다.

GA는 크게 4단계로 구성된다. 첫 번째 단계는 인코딩 및 초기 집단 형성(Encoding and Initialization)이다. NPC의 이동 방향과 속도 제어 수치를 유전자로 정의하고, 이들의 시퀀스인 염색체를 무작위로 다수 생성하여 1세대 개체군을 구성한다[6].

두 번째 단계는 적합도 평가(Fitness Evaluation)이다. 목적지와의 거리에 반비례하고 생존 시간에 비례하도록 적합도를 산정한다. 목적지 도달 시 높은 보상을, 장애물 충돌 시 강한 페널티를 부과하여 부적격 개체가 차세대에 전달되지 않고 도태되도록 유도한다.

세 번째 단계는 선택 및 유전 연산(Selection and Genetic Operations)이다. 룰렛 휠(Roulette Wheel) 방식으로 우수 개체에 번식 기회를 제공하고, 교차(Crossover) 연산으로 부모의 전략이 결합된 자식을 생성한다. 이때 적절한 선택 압력 조절로 다양성과 수렴 속도의 균형을 맞춘다[13]. 선택 압력은 우수한 개체가 다음 세대에 살아남아 자손을 남길 확률이 열등한 개체에 비해 얼마나 높은지를 나타내는 척도이다.

마지막 네 번째 단계는 변이 및 반복 진화(Mutation and Iteration)이다. 낮은 확률의 변이 연산을 통해 지역 최적해 문제를 방지한다[3]. 이러한 변이는 동적 게임 환경에서 예측 불가한 변화에 대응할 수 있는 전략적 다양성을 제공하며, 의사결정 고도화에 필수적인 역할을 한다[14][15].

결과적으로 유전 알고리즘은 '평가-선택-교차-변이'의 반복적 진화 과정을 통해, 세대가 거듭될수록 NPC가 복잡한 장애물을 회피하며 최적 경로로 목적지에 도달하는 지능형 제어 모델을 구축하게 한다[16].

2.2 경로 최적화 및 이동 제어

게임 NPC 경로 탐색은 결정론적 방식과 확률적 방식으로 나뉜다. 대표적 결정론적 방식인 A* 알고리즘은 정

적 환경에서 최단 경로를 보장하지만[2], 동적 장애물 환경에서는 경로 재설정으로 연산 부하가 급증하고 선제적 회피가 어렵다는 한계가 있다[12]. 목적지의 인력과 장애물의 척력을 이용하는 잠재적 필드 기법[13]은 연산 속도가 빠르나, 힘의 평형 지점에서 고립되는 지역 최적해 문제와 좁은 통로에서의 궤적 진동 현상을 유발한다.

본 연구가 제안하는 GA 기반 경로 최적화는 이러한 한계를 극복하기 위해 확률적 탐색을 도입한다. 기존 방식과 차별화되는 GA의 강점은 첫째, 변이 연산을 통한 지역 최적점 극복, 둘째, 센서 학습을 통한 동적 장애물의 예측 기반 지능적 회피, 셋째, 맵 구조에 종속되지 않는 범용적 적응성이다[4][10].

결과적으로 본 연구는 기존 기법의 연산 부하 및 고립 문제를 극복하기 위해 Unity 2D 환경에서 센서 기반 GA를 구축하고 자율형 NPC의 최적해 수렴 과정을 검증한다.

3. GA 기반 시스템 구현

3.1 NPC의 유전자 구조 및 인코딩

본 연구의 자율형 NPC는 환경 신호와 이동 제어를 직접 매핑하는 발현 게놈(Expression Genome) 구조를 채택하여 중간 연산 단계를 최소화하였다. 전체 염색체는 가변적 환경에 대응하는 다차원 실수형 배열로 구성되며, 개별 유전자는 특정 거리에서 실행될 [회피 각도, 이동 속도]의 쌍으로 정의된다.

유전자 기반의 실시간 제어 프로세스는 다음 3단계로 진행된다.

첫째, 거리 감지 및 정규화 단계이다. 유니티 물리 엔진의 센서를 통해 탐지한 장애물과의 거리를 0과 1 사이의 정규화된 값(d)으로 변환한다.

둘째, 유전자 매핑 단계이다. 산출된 d값을 수식(Index = d * (n-1))에 적용하여 현재 거리에 가장 적합한 회피 각도 및 속도 유전자를 즉각 추출한다. n은 유전자 개수이다.

셋째, 물리 제어 단계이다. 추출된 유전자 데이터를 NPC의 현재 진행 방향 벡터에 가산하여 유니티 물리 엔진(Rigidbody2D)에 실시간으로 반영한다.

이러한 직접 매핑 방식은 거리가 변하는 즉시 유전자가 교체되어 생물학적 반사 신경에 준하는 신속한 응답 속도를 제공한다. 이를 통해 근접 구간(d 〈 0.2)에서는 급격한 회피를, 원거리 구간(d 〉 0.7)에서는 직선 가속을 수행하는 직관적이고 효율적인 기동이 가능해진다.

3.2 구현

본 연구는 자율형 NPC의 환경 정보를 즉각적인 행동으로 전환하여 최적해 탐색 속도를 가속화하는 핵심 단위로 Gene 구조체를 정의한다(〈Table 1〉). Gene 구조체는 장애물과의 거리에 따라 최적의 궤적을 생성하는 회피 각도(-180도~180도)와 생존율 극대화를 위해 가감속을 조절하는 이동 속도 정보를 포함한다. 센서를 통해 측정된 정규화 거리(d)를 기반으로 해당 구조체가 즉각 호출되며, 추출된 유전자의 각도와 속도 값은 유니티 물리 엔진에 실시간 회전력 및 추진력으로 반영되어 NPC의 직관적인 회피 기동을 제어한다.

〈Table 1〉 Gene

```
public struct Gene
{
    public float angle;
    public float speed;
}
```

〈Table 2〉는 NPC 행동 제어 로직을 보여준다. NPC의 실시간 행동 제어 로직은 다음의 9단계로 수행된다. 단계 (1)에서는 NPC의 활성화 상태를 검사하고, 단계 (2)에서는 NPC의 현재 위치에서 장애물이 존재하는지 실시간으로 탐색한다. 레이저에 무언가 감지되었다면 실제 충돌 지점까지의 거리(hit.distance)를 취하고(단계 3), 수집된 거리 데이터를 0과 1 사이의 값으로 변환하는 데이터 정규화 과정을 수행한다(단계 4). 이는 장애물이 코앞에 있으면 0에 가깝고, 멀리 있으면 1에 가까운 상태를 의미하며, 유전자의 배열 인덱스를 찾기 위한 표준화 과정이다. 단계 (5)는 유전자 인덱스 매핑(Gene Index Mapping)으로, 정규화된 거리 값을 유전자 배열(chromosome)의 크기에 맞게 인덱스로 변환한다. 이 구조체 안에는 진화를 통해 해당 거리에서 생존율이 가장 높았던 회피 각도와 이동 속도 정보가 담겨 있다(단계 6). 단계 (7)에서는 회전 쿼터니언을 생성하여 NPC가 장애물을 피하기 위해 얼마나 꺾어야 하는지를 수학적으로 정의한다. 단계 (8)에서는 NPC의 현재 정면 방향 벡터에 유전자의 회전 값을 곱하여 새로운 이동 방향 벡터를 산출함으로써 최적의 회피 궤적을 생성한다. 마지막으로 단계 (9)에서는 최종 결정된 방향과 유전자의 속도를 곱하여 물리 엔진(Rigidbody2D)의 속도값에 대입하여 실제 이동을 수행한다.

〈Table 2〉 Behavioral Logic

```
void FixedUpdate(){
 (1)  if (isDead) return;
 (2) RaycastHit2D hit = Physics2D.Raycast(transform.position,
     transform.up, sensorRange);
 (3) float distance = (hit.collider != null) ? hit.distance :
     sensorRange;
 (4) float normalizedDist = distance / sensorRange;
 (5) int geneIndex = Mathf.Clamp((int)(normalizedDist *
     (chromosome.Length - 1)), 0, chromosome.Length - 1);
 (6) Gene currentGene = chromosome[geneIndex];
 (7) Quaternion rotation = Quaternion.Euler(0, 0,
     currentGene.angle);
 (8) Vector2 moveDirection = rotation * transform.up;
 (9) rb.velocity = moveDirection * currentGene.speed;
}
```

〈Table 3〉은 GA 기반 NPC 진화 다섯 단계의 흐름으로 보여준다. 첫째, 무작위 회피 각도와 이동 속도를 지닌 1세대 개체군을 생성하여 초기화한다(단계 1). 둘째, 실제로 이동하며 회피 동작을 수행한 후(단계 3-4), 종료 시 NPC 상태에 따라 적합도를 차등 산출한다(단계 5). 이때 최단 경로 탐색 유도를 위해 목표 도달 NPC에는 높은 보상을 부여하고(Fitness = 2.0 + 0.8 × F_time), 조기 사망 NPC에는 부적합 형질 억제를 위해 1/10 삭감 페널티를 적용하며(Fitness = (F_dist + 0.2 × F_time) × 0.1), 미도달 생존 NPC는 제자리 회피 방지를 위해 시간 가중치를 낮춰 평가한다(Fitness = F_dist + 0.2 × F_time). F_time은 생존시간을 F_dist는 타겟까지의 거리를 나타낸다. 셋째, 산출된 적합도를 바탕으로 상위 10%는 엘리트주의로 보존하고(단계 7), 나머지는 룰렛 휠 방식으로 부모를 선정한다(단계 8). 넷째, 부모 간의 교차 연산으로 강점을 결합한 자식을 생성하고(단계 9-10), 낮은 확률의 변이를 통해 회피 전략의 다양성을 확보한다(단계 11-13). 마지막으로 자식 집단으로 세대를 대체하고 갱신하며(단계 15-16), 최대 세대 도달 시까지 이 과정을 반복하여(단계 2) NPC가 동적 환경에 최적화된 실시간 대응 지능을 스스로 확보하도록 한다.

〈Table 3〉 Implementation of GA

```
ALGORITHM GeneticEvolutionForNPC
BEGIN
(1) population with Random Genes (Angle, Speed);
(2) WHILE (Generation 〈 MaxGenerations) DO
(3) FOR each Agent IN Population DO
(4) Simulate Movement and Avoidance;
(5) Fitness=Calculate distance from Agent to Target;
(6) END FOR
(7) Select Elite Individuals (Top 10%);
(8) Select ParentA, ParentB using Roulette Wheel Selection;
(9) FOR each NewChild IN NextGeneration DO
(10) newChild.Genes=Crossover(ParentA.Genes,ParentB.Genes);
(11) IF (RandomValue 〈 MutationRate) THEN
(12) Apply Random Change to NewChild.Genes;
(13) END IF
(14) END FOR
(15) Population = NextGeneration;
(16) Generation = Generation + 1;
(17) END WHILE
END
```

3.3 결과

본 연구의 실험은 Intel Core i7, 16GB RAM, 유니티(Unity) 6.4 버전 환경에서 진행되었다. 검증을 위해 유니티 공식 프로젝트인 '2D Beginner: Adventure Game'의 'Ruby-ExampleScene'을 채택하였으며, 이는 콜라이더가 적용된 장애물들이 사전 구축되어 있어 센서 기반 실시간 회피 알고리즘 검증에 최적화되어 있기 때문이다.

실험 파라미터는 개체군(Population) 크기 15, 세대별 탐색 시간(Session Time)은 30초로 설정하였다. 특히, 데미지 존 진입 시 물리 연산을 즉각 중단하는 즉사(Instant Death) 메커니즘을 도입하여 부적합 개체가 차세대에 확실히 도태되도록 설계하였다. 알고리즘의 유효성은 매 세대 종료 시점마다 최고 적합도(최단 거리)와 평균 적합도(평균 거리)를 기록하여 최적해로의 수렴 여부를 분석함으로써 검증한다.

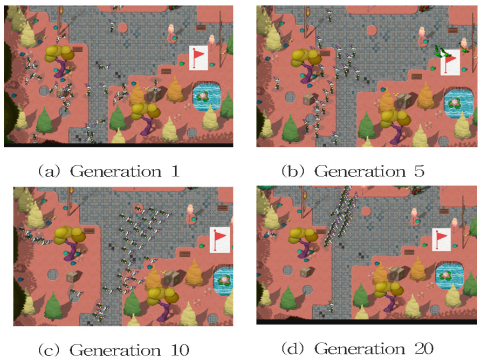
[Fig. 1]은 시작점과 도착점 그리고 데미지 존, 나무 등의 장애물을 보여준다.

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

[Fig. 2]는 제안 알고리즘의 세대별 진화 및 최적해 수렴 과정을 보여준다. 진화 단계별 특성은 크게 세 가지로 요약된다.

첫째, 1세대의 초기 탐색 단계에서는 에이전트들이 사전 정보 없이 맵 전체에 무작위로 분산되어 이동하는 양상을 보였다. 둘째, 5~10세대의 형질 보존 및 경로 집중 단계에서는 진화 연산을 통해 우수한 유전 형질이 보존되며, 목표 지점을 향한 주 경로로 개체들이 밀집하기 시작했다. 셋째, 20세대의 수렴 단계에서는 대다수의 에이전트가 최적화된 경로를 따라 집단적으로 이동하며 가장 효율적인 생존 및 이동 전략을 학습했음을 나타냈다. 결

과적으로 세대 반복에 따른 개체군의 경로 집중과 성공적인 목표 도달은 본 유전 알고리즘이 복잡한 환경에서 매우 효율적으로 작동함을 입증한다.

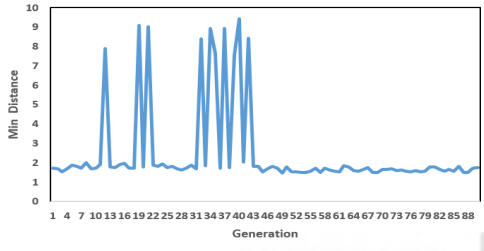


(a) Generation 1 (b) Generation 5

(c) Generation 10 (d) Generation 20

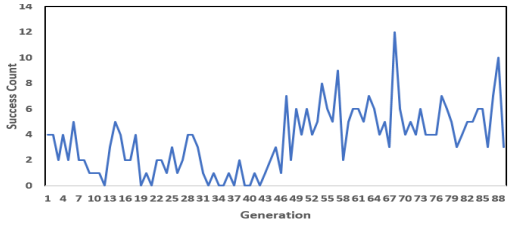
[Fig. 2] Evolutionary Progress

[Fig. 3]은 세대 수에 따른 최소 거리를 나타내며 유전 알고리즘의 전형적인 최적화 수렴 곡선을 보여준다. 분석 결과는 다음과 같다. 1~42세대의 초기 탐색 구간에서는 구간별 최단 거리 수렴이다. 1~42세대의 초기 탐색 구간에서는 전역 최적해를 찾기 위한 높은 변동성(8.0 이상 상승)이 관찰되었으나, 43~90세대 구간부터는 1.5~1.8 수준으로 안정화되었다. 특히 49세대에서 최저치인 1.48을 기록하며 고정밀 학습 단계에 진입했음을 확인하였다. 둘째, 최소 거리의 유의미한 증가 현상이다. 1세대의 최소 거리는 2였으나, 34~42세대 구간에서는 평균 8~9로 약 4배가량 증가하는 현상이 관찰되었다. 이와 같은 최소 거리의 주기적인 등락은 개체군의 유전적 다양성이 적절히 유지되고 있음을 의미한다. 특히 43세대 이후 최소 거리가 2로 다시 안정화되는 양상은, 누적된 우수 형질이 최적으로 조합되어 완전한 수렴 단계로 전이되었음을 시사한다. 이는 유전 알고리즘이 초기 무작위 탐색 과정의 시행착오를 극복하고, 동적 환경에서 NPC의 목적지 지향적 지능을 효과적으로 고도화하였음을 보여준다.



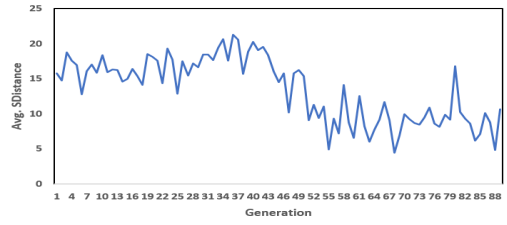
[Fig. 3] Min Distance by Generation

[Fig. 4]는 세대수에 따른 타겟 도착 성공 횟수를 보여준다. 초기 42세대까지는 성공 횟수가 0에 머무는 세대가 8번 정도 발생하였으나, 이는 유전적 다양성을 확보하기 위한 탐색 과정으로 해석된다. 68세대에서 가장 높은 성공수(12개)를 달성하였으며, 후반부 세대에서는 초기 대비 안정적으로 성공 개체가 발생함을 보여주고 있다.



[Fig. 4] Success Count by Generation

[Fig. 5]는 세대 수에 따른 평균 거리를 보여준다. 평균 거리는 집단 전체가 목표 지점에서 얼마나 떨어져 있는지를 보여준다. 이 선이 서서히 낮아지거나 특정 수준을 유지하는 것은 집단 전체의 형질이 안정화되고 있음을 의미한다. 즉 수렴되고 있음을 보여주고 있다.



[Fig. 5] Average Distance by Generation

4. 결론 및 추후 연구방향

본 연구는 2D 게임 환경에서 90세대에 걸친 유전 알고리즘 기반 NPC 경로 탐색 실험을 통해 제안 모델의 효용성을 검증하였다. 분석 결과, 세대가 진행될수록 목표 도달 개체수가 증가하여 진화적 학습 능력을 입증하였고, 49세대에서 최단 거리가 1.48로 하향 안정화되며 정밀한 최적해 수렴을 확인하였다. 또한, 평균 거리와 최단 거리 간의 유동적인 간격 변화를 통해 지역 최적해에 빠지지 않고 유전적 다양성과 탐색의 균형을 성공적으로 유지함을 입증하였다.

향후 연구로는 본 모델의 성능 고도화를 위해 적합도에 따라 연산 파라미터를 실시간으로 조정하는 적응형

유전 알고리즘의 도입을 검토한다. 아울러 동적 목표 이동 및 복잡한 장애물 배치를 통한 알고리즘의 강건성 검증을 수행하고, 최종적으로는 신경망의 가중치를 진화 연산으로 최적화하는 심층 신경진화(Deep Neuroevolution) 기법을 접목하여 더욱 지능적인 자율형 행동 제어 모델로 발전시킬 계획이다.

REFERENCES

[1] I. Millington and J. Funge, AI for Games, 2nd ed., Burlington, MA: CRC Press, 2009.

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

[3] P. Yap, "Grid-based Pathfinding," in Proceedings of the 15th Conference on Canadian Society for Computational Studies of Intelligence, Calgary, Canada, pp. 44-55, 2002.

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

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

[6] M. Buckland, AI Techniques for Game Programming, Cincinnati, OH: Premier Press, 2002.

[7] K. O. Stanley and R. Miikkulainen, "Evolving Neural Networks through Augmenting Topologies," Evolutionary Computation, vol. 10, no. 2, pp. 99-127, Jun. 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, Sep. 2011.

[9] S. M. Lucas, "Computational Intelligence and Games," in Proceedings of the 2008 IEEE Conference on Computational Intelligence and Games (CIG), Perth, Australia, pp. 21-28, 2008.

[10] J. K. Haas, "A History of the Unity Game Engine," M.S. thesis, Dept. Interact. Media Game Develop., Worcester Polytechnic Inst., Worcester, MA, 2014.

[11] Unity Technologies, 2D Beginner: Adventure Game, Unity Asset Store [Online]. Available: https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-learn-2d-adventure-game-robot-repair-271239 (accessed Apr. 7, 2026).

[12] 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, v.17 no.6,, pp.129 – 138, 2017.

[13] J. E. Baker, "Adaptive Selection Methods for Genetic Algorithms," in Proceedings of the 1st International Conference on Genetic Algorithms, Hillsdale, NJ, pp. 101-111, 1987.

[14] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, 2nd ed., Berlin, Germany: Springer, 2015.

[15] J.W. Koh and D.W Lee, "Path-finding Algorithm using Heuristic-based Genetic Algorithm", Journal of Korea Game Society, vol.17, No.5, pp.129-138, 2017.

[16] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, West Sussex, UK: John Wiley & Sons, 2001.

[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.5, pp.13-22, 2017.

[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.

이 면 재(MyounJae Lee) [종신회원]

■ 2009년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

〈관심분야〉
사물인터넷, 게임, MPEG