

genetic algorithm

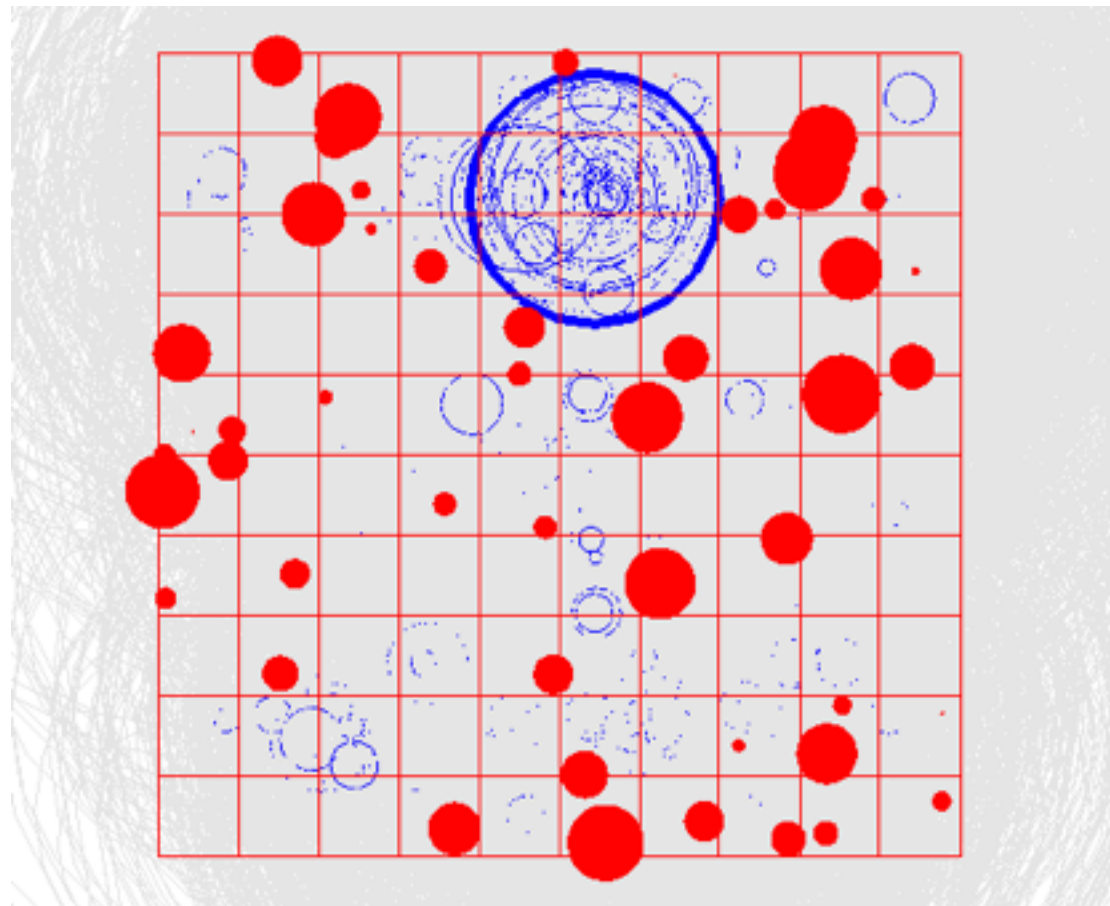
유전자 알고리즘

강영민

동명대학교 / 게임공학과

그게 뭔가요?

- 최적화 알고리즘
- 자연의 진화 과정에서 영감을 얻은 알고리즘
 - 다윈의 자연선택
- John Holland 1975



어디에 쓰이나?

- 어려운 문제
- 풀이법을 찾기 힘들 때
- 일종의 탐색문제
- 공학, 과학, 경제학 등에서 나타나는 복잡한 문제의 풀이에 응용

기본적인 동작

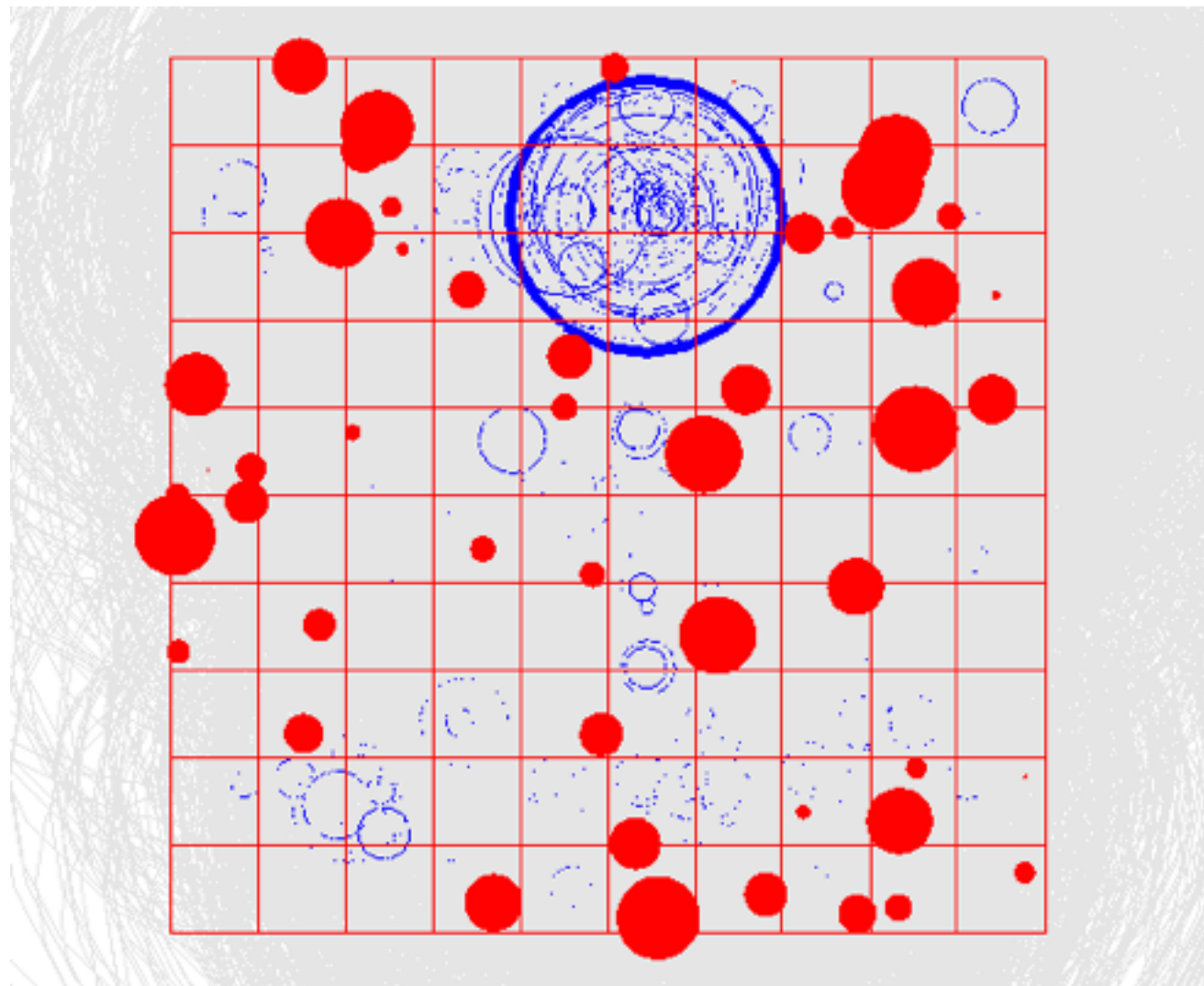
- 선택
 - 뛰어난 유전자를 선택하여 복제하도록 함
- 재조합
 - 두 개의 서로 다른 유전자를 섞어 새로운 유전자를 만듦
- 돌연변이
 - 하나의 유전자에 임의적인 변형을 가함

간단한 유전자 알고리즘

- 문제를 정의한다
- 문제의 해를 표현하는 방식을 결정한다.
- 이 표현방식에 따라 다양한 해를 생성한다.
- 각각의 해가 가지는 적합도(fitness)를 계산한다.
- 종료 조건이 만족될 때까지 다음을 반복
 - 좋은 유전자를 골라 복제한다.
 - 유전자들을 골라 재조합한다. (좋은 유전자일수록 재조합에 참여할 확률이 높다.)
 - 일부 유전자에 돌연변이를 일으킨다

적용 예

- 다음 그림의 붉은 점에 부딪히지 않고, 사각형 구역에 들어가는 최대 크기의 원을 찾으시오.



원의 유전자

```
// phenotype
class Circle {
public:
    float x,y;
    float radius;
};
```

```
// genotype
class Gene {
public:
    //sign(1) + x(20) + sign(1) + y(20) + radius(20);
    bool str[62];
};
```

Genotype-> Phenotype

```
float convertNumber(bool *str, int idx) {  
    float val=0;  
    for(int i=0;i<20;i++) {  
        val*= 2.0;  
        val+= str[i+idx]?1:0;  
    }  
    val /= 1048576.0;  
    return val;  
}
```

```
void convertGeneToPhenotype(Gene g, float &x, float &y, float &r) {  
    float val = convertNumber(g.str, 1);  
    x = g.str[0]?val:-val;  
    val = convertNumber(g.str, 22);  
    y = g.str[21]?val:-val;  
    r = convertNumber(g.str, 42);  
}
```


적합도

```
float fitness(Circle c) {  
    for(int i=0;i<MAX_OBSTACLES;i++) {  
        if( Distance(c,obstacleCircles[i])<0 ) {  
            return 0 ;  
        }  
    }  
    if( OutOfBounds(c) ) return 0;  
    return c.radius ;  
}
```

유전자 알고리즘 적용

```
if(reset) {  
    createRandomCircles();  
    createInitialGeneration();  
    reset = false;  
}  
drawCircles();  
computeFitness();  
drawGeneticCircles();  
  
nextGeneration();
```

탐색 결과

