

유전 알고리즘 프로젝트 1 보고서

April 20, 2018

1 사용한 GA의 구조

이번 과제에서는 Maxcut 문제를 풀기 위하여 가장 기본적인 GA 연산자들만 적용하였다. 기본적으로 N개의 유전자 집합인 해를 Population을 랜덤으로 생성하고, 그것으로 GA를 실행한다. 기본적으로는 K개의 해를 Population으로부터 교체하는 Generation의 교체를 반복하는데, 그 과정은 유전자로부터 Crossover시킬 부모의 형질을 선택하는, Selection, 그리고 선택된 두개의 부모를 교차시키는 Crossover, 그리고 교차된 자식 유전자에 대해 특정 확률을 통해 변이시키는 Mutation. 이 과정을 통해 K개의 자손 해를 생성시키면, Population 집단으로부터 K개의 해와 교체시킨다. 이 과정이 하나의 Generation 교체 과정.

1.1 문제 인코딩

그래프 인코딩은 기본적으로 인풋으로 주어진 그래프의 형태를 그대로 따라, (from, to, weight)를 가진 Edge의 집합으로 구성하였다. Unweighted 그래프의 경우 weight가 일정하게 주어지지만, 그대로 자료구조를 사용할 수 있고, Weighted 그래프의 경우 두말 할 필요없이 그대로 이 인코딩을 적용할 수 있기 때문이다. 해 - 유전자 인코딩, 자료구조 설명
놈놈.....

1.2 GA의 세부 구조

;Pseudo Code;

GA의 기본적인 실행 구조는 주어진 TimeConstraint 동안 여러 Generation을 거쳐서 최적의 해를 탐사하는 과정을 계속해서 거쳐나간다. 구현된 ga() 함수에서는 매 Generation, 즉 Select/Crossover/Mutation/Replace 과정을 TimeConstraint 안에서 계속해서 반복문을 돌면서 수행해나간다. 이러한 수행과정에서 계속해서 해집단인 Population은 진화해나가고,
놈놈.....

1.3 사용한 연산자에 대한 설명

Selection의 경우 처음에는 두개의 부모를 무작위로 선택하는 Random 방식을 사용하다가, Population의 각 해의 fitness를 계산하여 공간 탐색을 하는 Roulette Wheel 알고리즘을 사용하였다.

Crossover의 경우, 1-Point Crossover와 2-Point Crossover를 구현하여 실험해보았다. 1-Point Crossover와 2-Point Crossover 두가지를 사용하여 실험한 이유는, 단지 1-Point Crossover와 Multipoint Crossover의 차이를 비교해보기 위함이었는데, 부모 형질을 반영하지 않는 Random Point Crossover의 경우 둘 다 큰 차이는 없었다.

Mutation은 Uniform 방식을 사용하여 Chromosome내 Random한 지점들에 대해 각각 변이시키는 방식을 선택하였다. 처음에는 Mutation Probability를 1.0%를 주고 실험하다가 Mutation Probability를 1.5%를 주고 실험을 하는 식으로 진행하였다.

Replacement는 구해진 k개의 자손 들을 Population 중 가장 품질이 나쁜 k개의 해와 교체 하는 Genitor-style 방식으로 구현하고 실험하였다.

놈놈.....

2 실험 및 결과

실험 환경에 대한 설명. 컴퓨터 환경 및 OS 환경 싱글 코어 및 싱글 스레드 프로그램에 대한 수행. 실험 수행 시간 및 전체 수행 횟수.

놈놈.....

2.1 실험 1 $n=100, k=20$

가장 기본적으로 수행한 실험 결과. Population size $n = 100$, Replacement size $k = 20$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.

이번 과제에서는 Generation gap(Population size / Replacement size)을 0.2로 유지하면서, 실험을 진행하였다. 다른 여러가지 변수로 맞추어서 실험해본 결과 이 게 가장 품질이 좋았기 때문이다. 처음에는 단순히 Population size가 크면, 초기에 탐색하는 공간에서 넓은 범위를 가질 확률이 높다고 생각해서 n 을 크게 주고 실험해보았다. 다음 실험 결과에서 보듯, n 을 바꿔가면서 실험하였지만, 사실 n 이 큰 것이 좋은 품질을 주지는 않았다. 그것보다 n 의 크기를 적당하게 준 것이 오히려 좋은 결과로 나타날 수 있었다.

Table 1: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted</i> 50	95.9	96	0.2
<i>Unweighted</i> 100	349.5	350	1.3
<i>Unweighted</i> 500	3070.3	3080	18.1
<i>Weighted</i> 500	4340.6	4360	36.3
<i>WeightedChimera</i> 297	8035.4	8512	1084.4

2.2 실험 2 $n=50, k=10$

가장 기본적으로 수행한 실험 결과. Population size $n = 50$, Replacement size $k = 10$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.

이 실험 결과도 실험 1과 비슷한 추이를 보였다. Population size는 유전자 해의 품질에 크게 영향을 미치지 않는 것 같다. 하지만, 크기를 조금 낮추었을 때, 보이는 Generation evolution 횟수의 증가가, 눈에 띄게 비교되어 다만 이것을 좀 더 optimal한 위치로 찾고자 계속해서 낮추어 실험해보기로 하였다.

Table 2: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차	진화횟수
<i>Unweighted</i> 50	95.9	96	0.2	1
<i>Unweighted</i> 100	349.5	350	1.3	1
<i>Unweighted</i> 500	3070.3	3080	18.1	1
<i>Weighted</i> 500	4340.6	4360	36.3	1
<i>WeightedChimera</i> 297	8035.4	8512	1084.4	1

2.3 실험 3 n=20,k=4 Generation gap 조절(최적)

가장 기본적으로 수행한 실험 결과. Population size $n = 20$, Replacement size $k = 4$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.

이 실험 결과에서는 실험 1,2의 진행에 맞추어 Population size를 계속해서 낮춘 것이었다. 사실 이 것 이외에도 중간에 실험으로

Table 3: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted50</i>	95.9	96	0.2
<i>Unweighted100</i>	349.5	350	1.3
<i>Unweighted500</i>	3070.3	3080	18.1
<i>Weighted500</i>	4340.6	4360	36.3
<i>WeightedChimera297</i>	8035.4	8512	1084.4

2.4 실험 4 Mutation prob 조정 - 1% (실패)

가장 기본적으로 수행한 실험 결과. Population size $n = 20$, Replacement size $k = 4$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.
 뚝뚝.....

Table 4: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted50</i>	95.9	96	0.2
<i>Unweighted100</i>	349.5	350	1.3
<i>Unweighted500</i>	3070.3	3080	18.1
<i>Weighted500</i>	4340.6	4360	36.3
<i>WeightedChimera297</i>	8035.4	8512	1084.4

2.5 실험 5 Selection 전략의 변경 - Roulette Wheel(성공)

가장 기본적으로 수행한 실험 결과. Population size $n = 20$, Replacement size $k = 4$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.
 뚝뚝.....

Table 5: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted50</i>	95.9	96	0.2
<i>Unweighted100</i>	349.5	350	1.3
<i>Unweighted500</i>	3070.3	3080	18.1
<i>Weighted500</i>	4340.6	4360	36.3
<i>WeightedChimera297</i>	8035.4	8512	1084.4

2.6 실험 6 Crossover 전략의 변경 - Multipoint Crossover(실패)

가장 기본적으로 수행한 실험 결과. Population size $n = 20$, Replacement size $k = 4$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.
 뚝뚝.....

Table 6: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted50</i>	95.9	96	0.2
<i>Unweighted100</i>	349.5	350	1.3
<i>Unweighted500</i>	3070.3	3080	18.1
<i>Weighted500</i>	4340.6	4360	36.3
<i>WeightedChimera297</i>	8035.4	8512	1084.4

2.7 실험 7 g++ 최적화 옵션 사용 -O3

가장 기본적으로 수행한 실험 결과. Population size $n = 20$, Replacement size $k = 4$ 를 사용하고, Selection 전략은 Random, Crossover는 1-point, mutation은 uniform 을 사용하였다.
 뚝뚝.....

Table 7: 실험 1 결과

케이스	평균 결과	최고 결과	표준편차
<i>Unweighted50</i>	95.9	96	0.2
<i>Unweighted100</i>	349.5	350	1.3
<i>Unweighted500</i>	3070.3	3080	18.1
<i>Weighted500</i>	4340.6	4360	36.3
<i>WeightedChimera297</i>	8035.4	8512	1084.4

3 결과 분석

실험 결과 Population size와 Replacement size는 품질에 크게 영향을 주지 않는다는 것을 확인할 수 있었다. 너무 급진적인 변수만 아니라면.

그 대신, 그 외에 여러 가지 Selection 전략이라던가, Crossover 전략을 바꾸었을 때 품질에 대한 영향을 주는 요소가 더 많았다.

전통적인 GA를 돌렸을 때의 한계점.

마지막으로 가장 최종적으로 우수한 해를 나타낸 $n=20$, $k=4$, MutationProb=0.015, Roulette Wheel Selection, One-point Crossover와 g++ -O3 옵션을 통해 실험한 결과에서 수행된 Population의 진화 양상을 분석하였다. 초기 해의 수렴. 그 후에....

뚝뚝.....

4 논의

사실 보고서에 안썼지만, 품질 변화가 눈에 띄게 보이는 품질 향상을 만들어내기는 어려웠다.

여러번의 시행착오를 겪어서 가장 최적의 우수한 해를 찾아내는 변수들을 어느 정도 찾아낼 수 있었다.

실험 관찰 결과 재미있었던 것은 초기 initial population 생성된 해의 품질이 결과적으로 나타난 해에 영향을 많이 미친다는 것이었다. 변수의 변화보다..

초기에 Population을 문제에 대한 정보를 이용하여 잘 줄 수 있으면 좀 더 좋은 해를 얻을 수 있을 것 같다.

가장 전통적인 GA 연산자들로만 사용하여 우수한 해를 찾아내는 것에 대해 어려움이 많았다.

최적해에 어느정도 도달하였지만 그래도 많이 부족한 품질을 보임.

최적화와 지역 최적화 등 여러가지 방법을 사용하면 더 좋은 품질을 낼 수 있을 것 같다.

눔눔.....