

2022 GA Project 3. Max-Cut with Hybrid GA

22조 (서윤형, 서현규, 이재선)

Abstract

본 보고서는 local optimization heuristic algorithm을 갖는 Hybrid GA를 이용하여 max-cut 문제의 최적해를 도출하고자 한 과정 및 결과를 기술하고 있다. Local optimization은 Kim et al.이 제안한 MAX-FM 알고리즘을 사용했다. 8개의 인스턴스에 대하여 최적해의 99% 수준에 가까운 해를 찾을 수 있었다. Introduction, methodology, result, conclusion & discussion 순으로 문제에 대한 정의, hybrid GA 구조, 실험 결과 및 결론에 대하여 설명하고 있다.

Introduction

일반적인 GA는 넓은 공간에서의 해를 탐색하는 능력을 갖고 있으나 지역 최적점으로의 미세조정이 낮은 편이기 때문에 이러한 단점을 보완하기 위해서 혼합형태인 Hybrid GA가 사용된다. 본 과제에서는 GA에 지역최적화 알고리즘을 추가하고 GA 연산자를 수정 및 보완하여 성능을 개선하였다. 다음 절부터 이러한 방법에 대해 기술하고 샘플로 주어진 8개의 인스턴스를 기반으로 실시된 몇가지 실험에 대한 결과를 통해 알고리즘의 성능 및 각 요소의 기능 및 효과에 대해 설명하였고 마지막에 본 과제에서 사용된 Hybrid GA에 대한 느낀 점과 분석한 내용을 서술하였다.

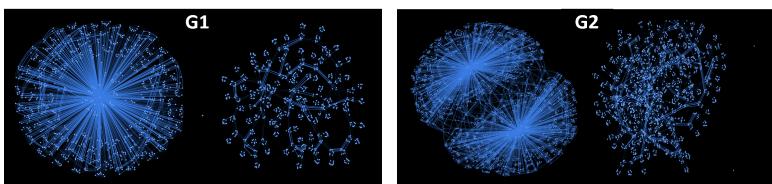
Methodology

실험환경은 지난 번과 동일하며 몇가지 실험을 통해 Hybrid GA를 검증하였다.

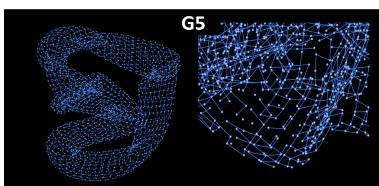
【입력 인스턴스 분석】

8개의 인스턴스가 입력값으로 주어지는데, 정점(vertex)의 수, 간선(edge)의 수, 간선의 가중치값, 정점간 연결형태 그리고 모든 정점과 연결되는 공통정점의 유무 등에서 차이가 있었다.

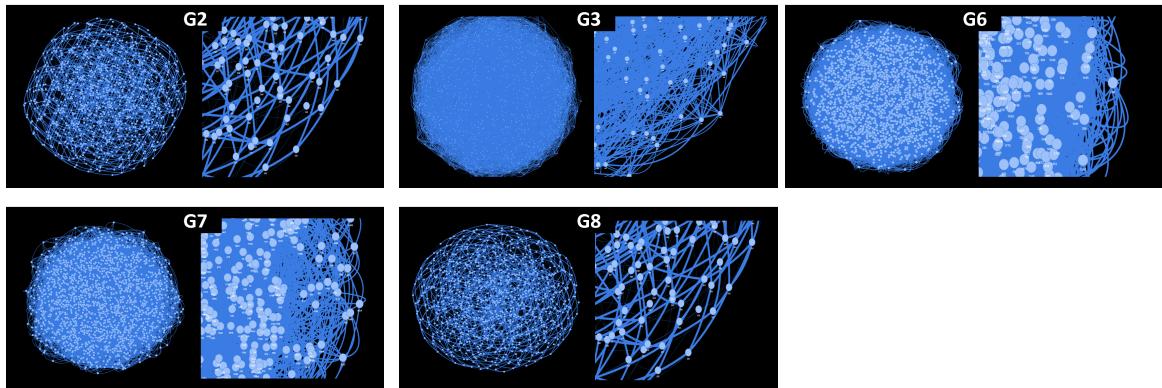
→ 분류 1. 공통정점이 있고 인접한 정점끼리 연결된 클러스터가 넓게 분포된 형태 : G1, G4



→ 분류 2. 인접한 정점끼리만 연결된 형태 : G5



→ **분류 3.** 인접성과 상관없이 정점간 복잡하게 연결된 형태 : 그 외 나머지



분류	분류 1		분류 2		분류 3			
	G1	G4	G5	G2	G3	G6	G7	G8
인스턴스								
정점	946개	1,852개	2,000개	1,000개	1,000개	2,000개	2,000개	2,744개
간선	3,240개	6,480개	4,000개	3,000개	10,000개	19,990개	11,778개	8,232개
정점간 연결형태	10개 이내 인접끼리 연결	10개 이내 인접끼리 연결	2~4개의 인접끼리 연결	10개 이내 무작위 연결	수십 개 무작위 연결	수십 개 무작위 연결	수십~수백 개 무작위 연결	10개 이내 무작위 연결
공통접점	1개	2개	-	-	-	-	-	-
가중치	100, -44, -100	100, -44, -100	1, -1	1, -1	3, 2, 1, -1	1	1	1, -1

G1, G4 그리고 G5는 정점간 연결이 인접성에 의존적이고 나머지 인스턴스의 경우 정점, 간선, 그리고 정점간 연결형태 등에서 다양성을 갖고 있었지만 인접성과 무관하게 복잡하게 연결된 형태를 보여주었다.

(분류1) G1은 경우 공통 정점이 1개이지만 G4는 두 개의 G1을 연결한 형태와 같으므로 복잡성이 증가하여 알고리즘 성능에 영향을 미칠 것으로 예상했다. (분류2) G5는 인접한 정점끼리만 연결된 특이한 형태로 마치 3차원 폴리곤처럼 정점간의 인접성이 매우 강한 특성을 갖는다. (분류3) G3, G6, G7 그리고 G8은 G2와 비교하여 정점이나 간선의 수, 간선의 가중치값 그리고, 정점간 연결의 복잡성 등의 차이가 있었지만 실험결과는 거의 비슷할 것으로 예상했다.

[Hybrid GA 전체 구조]

본 과제에서 사용된 Hybrid GA는 다음과 같은 구조를 갖는다.

1. 초기 해집합 무작위 생성 (해집합 크기 : 100)
2. 제한시간동안 Hybrid GA 수행
 - a. 선택 연산 : 룰렛휠 (선택압 : 3.0)
 - b. 교차 연산 : 균등교차 (임계확률 : 0.249)
 - c. 지역최적화 : MAX-FM heuristic algorithm
 - d. 대치 연산 : 엘리티즘 with 부모해 대치
3. 최적해 도출

[선택 연산, 교차 연산]

다양한 선택 및 교차 연산자를 실험한 결과 큰 효과가 없어서 지난 번과 동일한 방법을 사용하였다.

【지역최적화 알고리즘】

Kim et al.이 제안한 FM 알고리즘의 변형인 MAX-FM을 사용했다[1]. 처음에는 수업시간에 배운 KL 알고리즘을 사용하려 했으나, KL 알고리즘을 최적화하기 위해서는 max heap 등의 자료구조를 사용해야 하는 반면 MAX-FM은 보다 구현이 간단하고 실행 속도가 KL보다는 다소 빠르기 때문에 최종적으로 MAX-FM 알고리즘을 이용하여 자녀해에 대한 heuristic을 제시했다.

【대치 연산】

지난 프로젝트에서 엘리티즘 방식을 사용했는데 실험결과 수렴이 빨라서 수렴을 늦추기 위해 다음과 같이 수정하였다 먼저, 생성된 자녀해를 부모해와 비교한다. 두 부모 중 적합도가 작은 부모와 교체하고, 두 부모보다 자녀해가 항상 클 경우, 부모 해 중 자녀와 유사한 해와 교체하여 해 집단의 다양성을 유지시켰다. 만약 자녀해가 두 부모보다 적합도가 작을 경우에는 해 집단 중 worst solution과 교체했다 (=엘리티즘 대치).

Results

구현한 Hybrid GA에 대한 성능을 측정 및 분석하기 위하여 다음 세 가지 실험을 진행했다.

【샘플 인스턴스에 대한 실험결과 (가장 좋은 결과, 평균 결과 및 표준편차)】

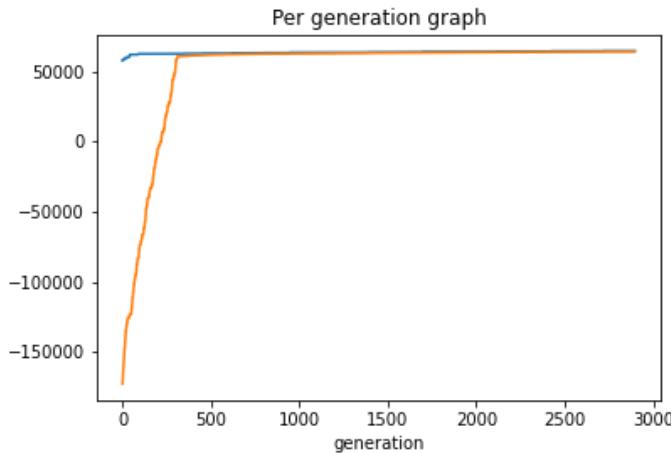
인스턴스 G1 ~ G8에 대하여 각각 20번씩 실행하여 얻은 결과는 다음과 같다.

Instance	G1	G2	G3	G4	G5	G6	G7	G8
Best	31,856 (100%)	894 (100%)	9,524 (100%)	65,700 (99.8%)	1,410 (100%)	13,308 (99.8%)	7,660 (99.6%)	2,442 (99.3%)
Average	31,856 (100%)	893 (99.9%)	9,492 (99.7%)	65,054 (98.8%)	1,406 (99.7%)	13,288 (99.7%)	7,650 (99.5%)	2,434 (98.9%)
S.D.	0	0.98	15	298.99	1.93	13.4	5.48	4.37

인스턴스 G1은 항상 최적해를 찾을 수 있었고 다른 인스턴스들도 대부분 최적해를 찾거나 최적해에 근사한 값(99% 수준)을 찾을 수 있었다. 다만 정점간 연결이 복잡한 G4와 정점수가 가장 많은 G8의 경우 최적해에 근사하게 접근하지만 주어진 시간 내 수렴을 못하는 결과를 보여주었다. G4의 표준편차가 큰 이유는 간선의 가중치값 변동폭(-100, -44 또는 100)이 커서 해에 따라 가중치 합의 차이가 많이 나기 때문에 판단이 된다. 실험결과 제안된 Hybrid GA가 최적해에 근접한 좋은 성능을 가짐을 알 수 있었다.

【가장 큰 그래프(G8)에 대해 세대 진행에 따른 population 수령 상황 및 Multi-start와의 비교】

먼저, 가장 큰 그래프 (G8)의 세대에 따른 population 수령 상황을 그려보면 다음과 같다.

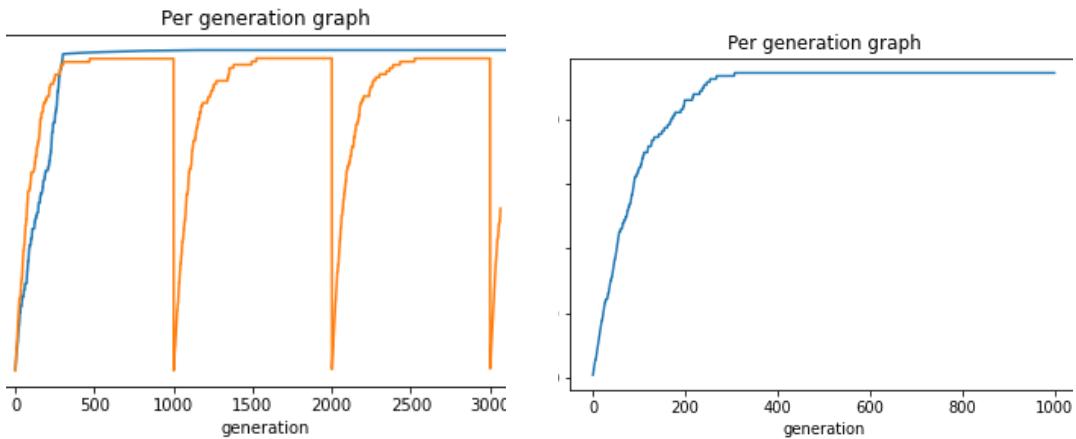


파란 그래프는 **best** 품질을, 주황 그래프는 평균 품질을 의미한다. 지역 최적화 heuristic 알고리즘 덕분에 GA 수행 초기부터 높은 **best** 품질을 기록하지만, 전체적인 평균 품질은 낮은데 시간이 지날 수록 평균 품질이 수렴해감을 알 수 있다.

GA는 최적화를 위한 전역탐색 능력을 갖고 있으나 지역탐색 능력은 상대적으로 낮기 때문에 지역최적점 근처에서의 미세조정 능력을 보완하기 위해 지역최적화 알고리즘을 추가하기도 한다. 본 실험에서는 지역최적화 알고리즘이 포함된 Hybrid GA와 GA 연산자 없이 local optimization만 사용하는 Multi-start 방식을 비교한 것으로 이를 통해 최적해를 찾는 문제에서 GA 연산자가 어떤 영향을 미치는지 알 수 있다. Multi-start와 같이 해집합의 초기화를 반복하여 지역최적점을 찾는 방식은 초기해의 품질이 매우 중요하지만 넓은 탐색공간에서의 좋은 해들을 모으는 것은 간단하지 않다. 반면에 GA연산자를 활용하는 경우, 초기해가 다소 품질이 좋지 않더라도 진화를 통해서 좋은 해를 얻을 수 있고 이를 통해 궁극적으로 최적해를 탐색할 수 있다.

Instance	G8	
Algorithm	Hybrid GA	Multi-start
Best	2,442 (99.3%)	2,400 (97.6%)
Average	2,434 (98.9%)	2,391 (97.2%)
S.D.	4.37	5.16

다음의 그래프는 Hybrid GA와 Multi-start의 시간 경과에 따른 해의 수렴을 관찰한 것으로, 푸른색 그래프는 hybrid GA, 주황색 그래프는 multistart 를 의미한다. 전자의 경우 일정 세대가 지나면 최적해에 근접하게 수렴하는 반면 후자는 어느 정도 우수한 해에 근접할 수는 있지만 그 이상의 성능 향상이 보장되지 않음을 알 수 있다. 이것은 multi-start의 특성상 초기 해집합에 최적해까지 인도해줄 수 있는 해들의 포함 유무에 따라 결과가 정해지기 때문이다.



[그 외 GA 작동 메카니즘을 관찰할 수 있는 실험]

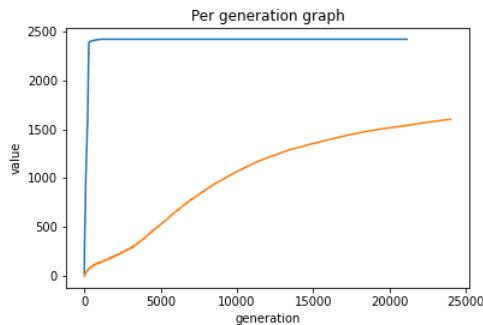
지역최적화 알고리즘의 효과

본 실험은 Hybrid GA에서 지역최적화 알고리즘 부분만 Project 2의 변이연산(1-bit flip)으로 대체하여 출력 해의 성능 그리고 시간에 따른 해집단의 변화를 비교한 것으로 이를 통해 새로 도입된 지역최적화 알고리즘의 효과를 확인하기 위한 것이다.

(출력 해의 성능) Hybrid GA의 지역최적화 알고리즘을 1-bit flip mutation과 대체한 결과 지역최적화 알고리즘으로 인해 해의 품질이 30% 넘게 향상되고 표준오차도 $\frac{1}{6}$ 수준으로 감소됨을 알 수 있는데 변이와 같은 진화에만 의존해서는 좋은 해를 얻을 수 없다는 것을 확인할 수 있었다.

Instance	G8	
Algorithm	Hybrid GA	Hybrid GA w/ 1-bit flip instead of local optimization
Best	2,442 (99.3%)	1,686 (68.5%)
Average	2,434 (98.9%)	1,652 (67.2%)
S.D.	4.37	25.2

(시간에 따른 해집합의 변화) G8에 대해 세대 진행에 따른 해집합의 변화를 비교한 결과 지역최적화 없이 변이연산만으로는 최적해에 수렴하기가 어려움을 확인할 수 있었다. 파란색 그래프는 Hybrid GA를, 주황색 그래프는 local optimization 을 사용하지 않은 GA를 의미한다.

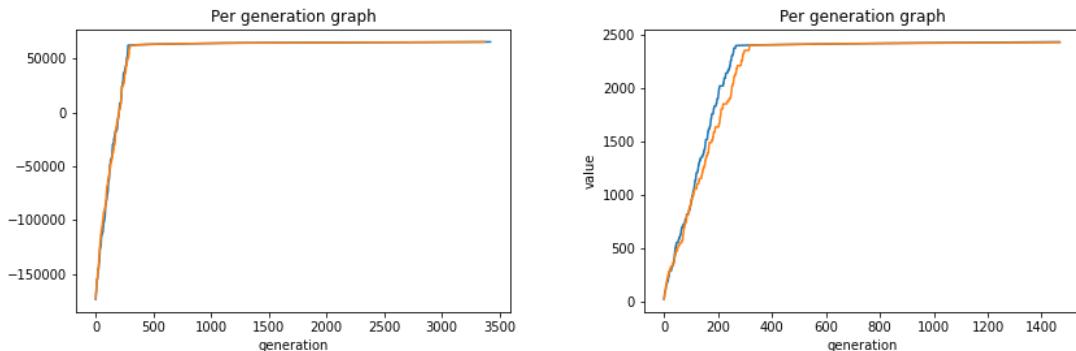


□ 유전자 재배치 효과

유전자 재배치는 유전자들간의 관계를 이용하여 유전자의 원래 배치를 바꾸는 것으로 이를 통해 해의 우수한 스키마를 세대를 거듭하는 동안 생존할 수 있게 하는 것이 목적이다. 본 과제에서는 DFS 기반의 유전자 재배치 알고리즘을 추가하여 실험하였고 다음의 결과를 얻을 수 있었다. 성능차이는 거의 없었으나 G4와 같이 정점간 인접성이 강한 경우 유전자 재배치를 통해 보다 안정적으로 최적해에 근사한 해를 찾을 수 있는 반면에 정점간 연결이 무작위성을 갖는 경우 오히려 표준편차가 소폭 하락함을 확인할 수 있었다. 실험결과 큰 효과가 없었기 때문에 최종적으로는 유전자 재배치를 사용하지 않았다.

Instance	G4		G8	
Reordering	X	O	X	O
Best	65,700 (99.8%)	65,712 (99.8%)	2,442 (99.3%)	2,446 (99.4%)
Average	65,054 (98.8%)	65,435 (99.3%)	2,434 (98.9%)	2,436 (99.0%)
S.D.	298.99	134.69	4.37	4.81

아래의 그림은 세대 진행에 따른 G4(좌) 및 G8(우)의 세대 진행에 따른 해의 수렴률을 나타낸 것으로 성능에 큰 차이가 없음을 보여준다.



Conclusion & Discussion

본 프로젝트에서는 지난 프로젝트에서 처럼 파라미터를 미세 조정하는 접근 방식 보다는 거시적으로 heuristic 알고리즘을 사용하는데 주력했다. 여러 경로를 통해 survey 를 한 결과 [1] Kim et al. 에서 제시된 MAX-FM 알고리즘을 도입했는데 이 heuristic 알고리즘을 사용한 결과 해의 품질이 매우 크게 증가한 점이 인상깊었다. 이를 통해 어느정도의 heuristic 알고리즘이 GA 의 성능을 크게 높일 수 있다는 점을 다시 한번 알게 되었다.

최적해를 구하는 과정은 주어진 문제를 고찰하는 것에서 시작이 되며 입력되는 데이터의 특성에 따라 사용되는 알고리즘도 달라지게 마련이다. 그러나, 본 과제에서 주어진 데이터의 경우 특성이 동일하지 않기 때문에 GA 연산자 등을 여러가지로 시도했을 때 데이터에 따라 성능이 향상되기도 하지만 모든 데이터에 잘 맞는 알고리즘을 정하는 것이 쉽지 않았다. 만약 입력된 데이터가 하나로 고정된다면 여러 가지의 GA 연산자의 효과 및 가장 적합한 파라미터를 찾는 것이 좀 더 수월했을 것으로 예상되는데 GA를 활용한 실험의 소요시간이 적지 않기 때문에 한정된 시간 내에 여러가지 시도를 해보지 못한 것을 아쉽게 생각하며 향후 기회가 된다면 다양한 옵션을 통해 진화에 기반한 알고리즘에 대해 고찰할 수 있으면 좋을 것 같다.

팀 기여도

- 이재선 ($\frac{1}{3}$), 서윤형 ($\frac{1}{3}$), 서현규 ($\frac{1}{3}$)

References

- [1] Kim, S.H., Kim, Y.H. and Moon, B.R., 2001, July. A hybrid genetic algorithm for the max cut problem. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (pp. 416-423).