Algorithm, 2023
Homework for the latter part of the class

MAX-CUT PROBLEM WITH HYBRID GENETIC ALGORITHM

디카페인 바닐라 라떼

2019-1130
김영은

1 해의 표현 및 사용한 GA 전체 구조

1.1 해의 표현

A, B 를 그래프 노드 수만큼 나열한 문자열로 표현한다. 문자열 인덱스와 노드 번호가 같고, 각 자리 글자가 노드가 속한 그룹을 의미한다. 'AABB', 'BBAA'처럼 서로 뒤집어 동치가 되는 경우는 고려하지 않으며 서로 다른 것으로 취급한다.

1.2 GA 구조

1. 노드 수만큼 A, B 를 50% 확률로 뽑아 해를 생성하고, 유효한 해인지 검사함과 동시에 cost 를 계산하고 대륙을 부여해 부모 집합을 만든다.

2. 서로 같은 대륙끼리 교배한다. female 이 cost 토너먼트로 먼저 뽑히고, female 의 cost 에 따라 male 이 선택된다. 부모의 cost 차이가 서로 일정량 이하인 경우만 교배하도록 하며, 낮은 확률로 cost 차이가 큰 부모 쌍이 생성될 수 있고, 자식도 더 많이 생성한다. 자식은 생성 직후 돌연변이를 시도하고, 생성된 자식을 검사해 유효하지 않은 것은 바로 제거한다.

3. 일정 수의 부모 쌍을 선택하여 생성된 자식들은 자신보다 cost 가 약간 작은 유전자를 대체한다. 대체할 유전자가 없는 경우 낮은 확률로 아무 조건 없이 pool 에 편입될 수 있다.

4. 제한 시간 내에 대륙 내 진화가 수렴하면 대륙마다 지역 최적화를 잠깐 시도한 후 위와 같은 과정으로 대륙 통합 진화를 시작한다. 수렴 후 남은 시간동안 지역 최적화를 시도한다.

2 DYNAMIC PROGRAMMING 을 활용한 방안

* 지역 최적화를 할 때는 비슷한 해의 cost 를 자주 계산하게 된다. 지역 최적화 내에서의 cost 계산에 대해 해를 key 로 하고 cost 를 value 로 하는 memo map 을 추가해 중복된 cost 계산을 줄였다. 이후 모든 해의 validation 과정에 memo 를 사용해 전체적인 cost 계산 시간을 줄였다.

* 지역 최적화를 할 때 해를 한 자리 변경해도 cost 가 유지되는 경우가 있다. 이때 바뀐 해와 이전 해 중 무엇을 이용하는 게 더 나은지 두 가지 경우에 대해 모두 지역 최적화 함수를 다시 호출하고 결과가 더 좋은 해를 사용하고자 했었다. 다만 재귀 호출이 누적되면 스택 오버플로우가 발생하여 재귀 호출은 한 번으로 제한한다. 이 방법으로 테스트한 결과, 성능이 하락하여 지금은 폐기하였다.

3 사용한 GA 연산자에 대한 설명과 DP 알고리즘 설명

* validate: 해의 유효성을 검사하며 cost 를 계산한다. 계산한 모든 cost 는 저장 후 재사용한다.

* selection: 전체 유전자 풀을 두 대륙으로 나누어, 서로 같은 대륙끼리 교배한다. 부모의 cost 차이를 일정 수치로 제한하나 낮은 확률로 제한되지 않은 쌍이 생성될 수 있다. female 이 먼저 선택되고, 그 결과 따라 male 이 선택된다.

* crossover: 해의 각 자리를 60% 확률로 부모 중 우월한 쪽의 것으로 선택한다.

* mutation: 해의 길이를 기준으로 변이 시도 횟수를 정하고, 무작위 발생 위치를 선택한다. 선택된 자리마다 일정 확률로 다시 선택하며, 이전의 값과 똑같을 수 있다. 발생 확률은 해의 길이에 맞춰 일정하게 유지한다.

* replacement: generational GA 방식 이용, cost 가 약간 작은 것과 대체한다. 대체 실패한 해는 낮은 확률로 그대로 pool 에 포함된다.

* local optimization: 현재 보유한 해 중 가장 좋은 것을 이용한다. 해의 각 자리를 무작위 순서로 선택하고 flip 하여 cost 가 같거나 커지는 쪽으로 해를 바꾼다. Simulated Annealing 방식을 사용하여 cost 가 낮아진 경우의 미래 가능성도 함께 고려하도록 하며, 최적화를 진행할수록 cost 가 낮아지는 방향으로 나아갈 확률이 감소한다. cost 가 같거나 커지면 해당 확률을 약간 증가시킨다. validation 시간을 줄이기 위해 계산한 모든 해의 cost 는 map 에 저장되고 재사용된다.

4 테스트 결과 분석

지난 과제에서 제공했던 세 개의 샘플 인스턴스와 weighted_chimera_297.txt 에 대해 GA 를 각각 최소 30 번씩 수행하여 가장 좋은 결과, 평균 결과, 표준편차를 테이블로 기록하고, 하나의 run 을 선택해 분석한다.

4.1 Sample run 통계

[Table 1]은 지난 과제에서 제공했던 세 개의 샘플 인스턴스와 weighted_chimera_297.txt 에 대해 GA 를 수행하고 기록한 결과이다. GA 의 성능 향상으로 결과가 안정된 세 개의 샘플 인스턴스는 31 번씩 실행하고, 그래프의 복잡도가 높아 편차가 큰 chimera 데이터는 61 번 실행하였다.

mean time 은 각 데이터별 평균 결과 도출 시간으로 weighted 500 데이터를 제외하면 대부분 제한 시간보다 훨씬 짧은 시간 내에 결과를 출력한 것을 볼 수 있다. 이로부터 대부분 데이터는 진화와 지역 최적화 모두 종료 조건을 충족하고 실행을 종료했다는 것을 알 수 있으나 아직 평균 cost 가 최고의 값으로 수렴하지 못했다는 점에서 본 보고서의 GA 가 아직 미숙함을 확인할 수 있다.

Table 1 Sample Test 통계

사용 데이터	runs	mean time(s)	min cost	max cost	mean cost	cost std
unweighted 50	31	0.792258	96	99	98.677419	0.791079
unweighted 100	31	4.210355	352	358	355.354839	2.429655
weighted 500	31	178.902903	4541	4670	4608.709677	32.513785
weighted chimera 297	61	20.334902	7300	8608	8088.065574	281.369500

4.2 Sample run 세대 분석

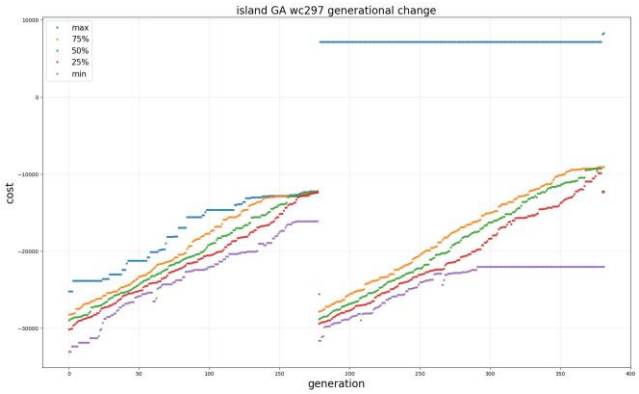

Figure 1 weighted chimera 297 generational change

[Figure 1]은 weighted chimera 297 데이터의 세대별 cost 변화를 나타낸 것이다. 본 보고서의 GA 구조에 따라 이 그래프는 총 세 구역으로 나뉜다. 그래프가 크게 끊어지는 지점을 기준으로 왼쪽, 가운데, 오른쪽이 각각 첫 번째와 두 번째 대륙 진화, 대륙 통합 진화에서의 세대 변화를 나타낸다. 첫 번째 대륙은 178 세대, 두 번째 대륙은 202 세대, 마지막 진화는 2 세대를 기록하여 총 382 세대가 그래프에 표현되었고, 각각의 지역 최적화가 하나의 세대로 기록되었으므로 실제로 진화를 시도한 세대는 하나씩 차감하여 총 379 세대이다. 그래프의 가로축은 세대 수, 세로축은 pool 을 구성한 해의 cost 값을 나타내며, 그래프는 위에서부터 max, 75%, 50%, 25%, min 값을 나타낸다. 첫 번째 대륙의 초기 pool cost 는 -33020 ~ -25192 사이였고 마지막엔 -22028 ~ 8304 로 진화를 종료했다. 평균은 -29116.4776 에서 -10340.2236 로 상승했다.

각 대륙 진화가 끝나고 한 번씩 지역 최적화를 시도하여 생성된 해를 모든 대륙에 추가했기 때문에 max 값이 첫 번째 대륙 진화 이후 급상승하는 것을 볼 수 있고, 이 수치는 지역 최적화가 아닌 진화만으로는 메꿀 수 없는 차이를 보인다. 두 번째 대륙 진화에서는 지역 최적화의 영향으로 순수하게 대륙에서 생성된 해의 최댓값을 알 수 없지만, 첫 번째 대륙 진화의 추이로 미루어 보건대,

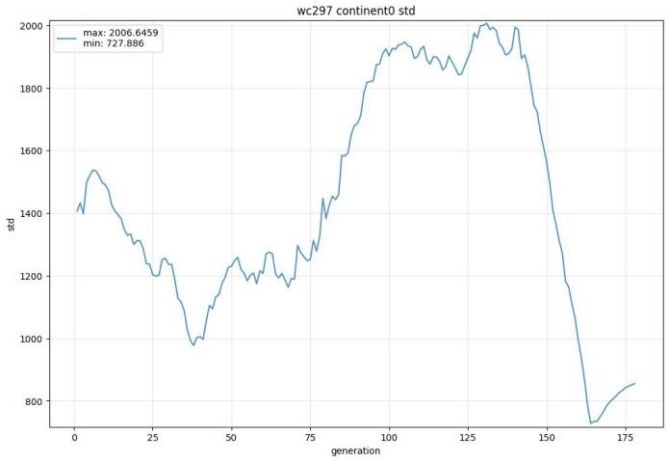

Figure 2 chimera 297 세대별 pool cost 표준편차

75% 값과 큰 차이가 나지 않았을 것이라 짐작할 수 있다. 이 그래프에 나타난 세 번의 진화 중 두 번의 진화에서는 공통적으로 75%, 50%, 25% 값이 max 값을 향해 수렴할 때 min 값은 잘 수렴하지 않는 것을 관찰할 수 있는데, 이는 세대 교체 시 자식의 cost 와 일정량 이하의 차이가 나는 해만 교체할 수 있도록 했기 때문에 품질 좋은 자식이 늘어날수록 cost 가 낮은 해가 교체될 기회를 적게 받아 끝까지 교체되지 못하고 남기 때문인 것으로 볼 수 있다. 또한 앞서 두 대륙이 이미 진화가 수렴한 상태로 통합되어 마지막의 대륙 통합 진화에서는 사실상 진화라고 할 만한 과정이 없었던 것을 볼 수 있는데, 이 또한 본 보고서의 GA 에 개선이 필요한 점이라고 할 수 있다.

[Figure 2]는 첫 번째 대륙이 진화한 178 세대의 세대별 pool 을 구성하는 해의 cost 표준편차를 그래프로 나타낸 것이다. 첫 세대에서 1405.6741 로 시작하여 마지막 세대에서 855.2889 로 끝났고, 최댓값과 최솟값은 각각 2006.6459, 727.886 이다. 이 그래프에 대해서는 어떻게 해석해야 할지 잘 모르겠다. [Figure 1]에 나타난 세대별 pool 의 5 분위 값을 보면 수렴하는 모양이 보이는데 표준편차는 중간에 급증하는 구간이 있다. 두 그래프를 겹쳐 대고 보면 5 분위 그래프의 간격이 약간 벌어지는 부분에서 표준편차가 증가한 것으로 보이는데, 갑자기 cost 구성이 다양해지고 표준편차가 증가한 이유는 알지 못했다. 두 번째 대륙과 마지막 진화 pool 은 첫 번째 대륙에서의 지역 최적화의 영향으로 순수하게 진화로만 구성된 pool 관찰이 어려워 분석하지 않았다.

5 시도해본 개선안

* 지역 최적화를 추가하면서 다양한 방식으로 해의 품질을 높이려 해봤는데, 제한 시간이 있기 때문에 짧고 여러 번 실행되는 지역 최적화 보다는 길게 한 번에 실행되는 지역 최적화가 훨씬 효과가 좋았다.

* 지난 과제 피드백에서, 세대 교체 과정을 좀 더 개선할 수 있을 것 같아 보인다는 말에 랜덤 없이 cost 가 가장 나쁜 것부터 차례로 대체하게 해봤는데 결과 품질이 떨어졌다. 이후 원래의 방법을 유지하되 큰 의미가 없는 랜덤성만 제거하는 것으로 수정했다. 교체 대상 cost 는 랜덤으로 뽑되 구체적인 교체 대상은 벡터의 맨 마지막 요소로 고정한다. cost 가 다르면 유의미한 차이지만 cost 가 같은 해끼리는 그다지 유의미한 차이가 나지 않는다고 생각했다.

* 토너먼트 승률을 두 대상의 cost 차이에 비례하게 해봤는데, 해의 품질이 떨어져서 다시 고정수치로 바꿨다. 그리고 토너먼트 참가자 수를 2 의 거듭제곱으로 맞추려다 보니 참가자 수를 정하는 코드가 복잡해져서 구현 방식을 큐로 바꿨다.

* 지역 최적화 중 변이 전후 cost 가 동일한 경우에 대해 재귀 호출을 이용해 더 전망이 좋은 것을 선택하면 결과도 더 좋아질 줄 알았는데 의외로 결과의 평균이 떨어졌다.

* 초기 풀 크기를 조작해보니 풀이 클수록 결과가 잘 나오는 경향이 있는 건 맞는 것 같지만, 그래프의 크기가 클수록 수렴과 진화에 걸리는 시간이 길어져 시간 내에 충분한 결과를 내지 못해 100 을 상한으로 잡기로 했었다. 이후 수정을 거쳐 최종 제출본은 초기 pool 크기의 최솟값만을 보장하도록 되어 있다. 보장된 최솟값은 20 이고, 그래프의 크기에 비례해 노드 수가 적을수록 큰 pool 을 생성한다. 그래프 크기가 클 경우 진화에 시간이 더 오래 걸리는 점을 감안해 작은 그래프는 다양한 해를 이용하여 수렴에 더 집중하고 큰 그래프는 작은 pool 에서 빠르게 수렴한 후 지역 최적화에 더 집중하게 했다. 교배 및 진화보다 지역 최적화로 개선되는 cost 변화량이 훨씬 큰 것 같아 이렇게 했다.

* cost 가 같은 해들의 공통된 부분을 모아 스키마로 삼고 교배 시 이 부분을 보호하게 해보려고 했는데, 막상 디버깅해보니 보호되는 글자 수는 지극히 적었고, cost 는 있지만 스키마는 생성되지 않는 오류가 발생해 이 방안은 폐기했다. 이 부분에서, 스키마가 거의 나오지 않은 이유로 AABB 와

BBAA 처럼 글자는 정 반대지만 실제로는 동일한 해가 pool 에 함께 존재하여 단순히 공통된 글자를 찾는 것만으로는 스키마를 만들 수 없었기 때문이라는 결론을 내렸었다. 그러므로 뒤집으면 서로 같아지는 해를 배제하기 위해 모든 해의 첫 글자를 A 로 고정하고 보호해봤는데, 가능한 해의 다양성이 절반으로 줄어서 그런지 성능이 떨어졌다. 스키마 파악과 더불어 그래프에 존재하는 노드의 A 집단과 B 집단의 구분 방향이 고정되면 그 안에서 최적의 해를 찾는 데 도움이 될 거라 생각해서 시도한 것인데 결과는 그렇지 않았다.

* 지역 최적화에 Simulated Annealing 도입. 복잡도가 큰 그래프 데이터를 사용하면 결과 편차가 너무 크게 나오는 문제가 있었는데, 이 방법으로 weighted chimera 297 데이터의 테스트 편차가 1300 대에서 200 중후반으로 줄었다. 평균도 약 2000 가량 상승했다. 처음엔 cost 가 나빠지면 온도를 감소시키기만 했는데, 그래도 수렴 속도가 너무 빠른 것 같아 cost 가 같거나 좋아지면 약간 증가시키는 코드를 추가했고, 도움이 되었다. 이후엔 cost 를 증가시키는 비율을 조정하면서 결과 편차를 줄이려고 했다. 이 알고리즘에 대해서는 chatGPT 에게 배웠다. 온도와 그 변동폭은 그래프의 복잡도를 반영하기 위해 pool 에 존재하는 최대 cost 와 최소 cost 의 차이 및 지역 최적화 전후 cost 차이에 비례하게 했다.

6 DISCUSSION - 프로젝트 리뷰

프로젝트 진행 중 느낀 점, 잘 안 되는 점, 의외의 현상, 예상대로 된 점 등을 서술한다.

6.1 프로젝트 후기

* 대륙을 분리하여 따로 진화시켜 수렴한 후에 다시 섞어서 교배하면 좀 좋은 해가 나올까 했는데 순수 GA 와 크게 다르지 않아 약간 실망했다. 뭐가 문제인지는 잘 모르겠지만, 지역 최적화가 해의 품질을 상당히 높여서 다행이었다.

* 지난 과제부터 비주얼 스튜디오의 빠른 실행을 위해 Release 상태로 실행을 해왔는데, 이 경우 실행 도중 오류가 나서 중단되었을 때 오류를 보여주지 않고 꺼진다는 것을 알게 되었다. 디버깅 모드가 따로 존재하는 이유가 있었다.

* 돌연변이가 구현만 되어 있고 사용되지 않고 있었다. 확인 즉시 수정했다.

* 여러 가지 하이퍼 파라미터들을 그래프 크기에 비례하는 랜덤 숫자로 해보니까, 랜덤성이 과하면 오히려 품질이 떨어지는 것 같다.

* 지역 최적화에 시간을 많이 투자하는 게 좋다고 생각해 여러 자잘한 과정에서 시간을 줄이려는 고민을 많이 했다.

* 지역 최적화 과정 중 Simulated Annealing 의 온도를 다시 높이는 부분에서, 해의 cost 가 높아지면 온도를 더 높이고 그대로이면 온도를 조금 올리는 게 좋은 해를 찾는 데 도움이 될 거라 생각했는데 실제로는 반대로 하는 게 평균도 좋고 표준편차도 작았다.

6.2 더 할 수 있는 것

* 대륙마다 진화를 끝낸 후 대륙을 통합할 때, 새로운 자식들을 무작위로 대거 생성해서 편입시키면 pool 의 수렴 정도는 떨어뜨리고 해의 다양성은 높일 수 있지 않을까? 가장 처음 순수 GA 를 구상할 때 했던 생각인데, 그때는 프로젝트 요건에 안 맞기도 했고, 처음 써본 알고리즘이라 능력의 한계로 구현하지 못했고, 이번에는 하이브리드 GA 구상을 크게 개편하여 잊고 있다가 프로젝트 기한이 다 되어서야 생각나서 실천해보지 못했다.

* 대륙을 분리하여 초기 해를 생성할 때, 생성된 해의 첫 글자를 기준으로 대륙을 분리하면 좀 다른 결과가 나올 수 있을 것 같다. 서로 다른 대륙끼리는 해의 첫 글자가 무조건 다르게 되어 두 대륙의 차이가 극명해지니 진화가 수렴한 후 대륙을 합쳐도 다양성이 남아 있는 효과가 있을 것으로 기대되고, 따라서 대륙 통합 진화도 어느 정도 세대를 거칠 수 있을 것으로 예상한다.