

# Symmetric Traveling Salesman Problem을 풀기 위한 Genetic Algorithm에서 유전자의 중복을 제거한 세대 구성 방법

김행수, 정태충

경희대학교 전자계산공학과 인터넷&지능시스템 연구실

## The Generation Organization Technique Removing Redundancy of Chromosome on Genetic Algorithm for Symmetric Traveling Salesman Problem

HeangSu Kim, TaeChoong Chung

Internet&Intelligent System Lab. Dept of Computer Engineering, KyungHee Univ

### 요 약

조합 최적화 문제인 Traveling Salesman problems(TSP)을 Genetic Algorithm(GA)과 Local Search Heuristic인 Lin-Kernighan(LK) Heuristic[2]을 이용하여 접근하는 것은 최적해를 구하기 위해 널리 알려진 방법이다. 이 논문에서는 LK를 이용하여 주어진 TSP문제에서 Local Optima를 찾고, GA를 이용하여 Local Optima를 바탕으로 Global Optima를 찾는데 이용하게 된다. 여기서 이런 GA와 LK를 이용하여 TSP문제를 풀 경우 해가 점점 수렴해가면서 중복된 유전자가 많이 생성된다. 이런 중복된 유전자를 제거함으로써 탐색의 범위를 보다 넓고 다양하게 검색하고, 더욱 효율적으로 최적해를 찾아내는 방법에 대해서 논하겠다. 이런 방법을 이용하여 rat195, gil262, lin318의 TSP문제에서 효율적으로 수행된다.

### 1. 서 론

의미론 상의 순회 판매원 문제(Traveling Salesman Problem 이하 TSP라고 한다)는 여러 개의 도시에 각각 도시간의 거리가 주어지고, 그 도시간의 최단거리를 찾는 문제이다. 여기서 각각의 도시는 정확히 한번 방문을 해야하며, 시작한 도시로 다시 돌아와야 한다. 물론 거리 대신에 시간이나 비용등을 이용해도 된다. 여기서는 Symmetric TSP 문제만을 다루겠다. Symmetric TSP (STSP) [5]는 임의의 두 도시 A와 B사이의 거리와 B와 A의 사이의 거리도 같아야 한다.

<STSP 제약 조건>

1. Tour를 이루어야 한다. (complete graph)
2. 모든 도시를 꼭 한번 들러야 한다.
3. 도시 A,B에서 Distance(A,B) = Distance(B,A) 위와 같은 TSP문제는 널리 알려져 있는 NP-hard combinatorial optimization 문제이다. 이런 문제를 풀기 위해서는 측정되는 시간은 N승의 계급으로 시간이 증가한다. 많은 휴리스틱 기술(genetic algorithm[3][4], simulated annealing[4])은 이런 문제에 대하여 합리적인 시간안에 최적해에 더욱 가까운 해를 찾기 위해 많이 제안되었다. 이 논문에서는 TSP문제를 풀기위해서 GA와 Local Search 기술(LinKernighan Heuristic)을 조합하여 접근하는 방법에서의 문제점을 제시하고 해결방법에 대해서 논하겠다. 일반적으로, 문제를 개선하기 위해서는 재 조합 방법이나 변이를 개선하는 것이 대부분이지만, 여기서는 각 세대마다의 중복 유전자를 제거하는 방법으로 일반적인 개선방법과는 다른 방법으로 이

문제를 풀어나가겠다.

이 논문의 구성은 2장에서 기존의 방법에 대해서 다룰 것이고, 3장에서는 새로운 적용방법에 대해서 논한다. 그리고 4장 실험 결과, 5장은 이 논문의 결론 및 앞으로의 방향에 대해서 다룬다.

### 2. 기존의 접근 방법들

아래 그림 1은 TSP 문제를 풀기위해 Local Search Heuristic 과 GA와의 결합에 대한 일반적인 방법들을 요약한 것이다. 초기해를 구성하기 위한 방법으로 Nearest Neighbor Heuristic 과 Insertion Heuristic[6] 등이 있다. Improvement tour heuristic으로 잘 알려진 것으로는 2-opt, 3-opt, 그리고 LinKernighan Heuristic이 있다. 이런 Heuristic은 한 세대의 각각의 유전자들에 수행되어 적당히 개선을 시킨다.

여기서 교배 연산자로 DPXCrossover, Multi-parent 재 조합 [7] 일반적으로 알려진 ranking, tournament, elitist 재 조합 방법 등도 있다.

이 논문에서 수행되어진 교배 연산자로는 DPXCrossover를 이용하였는데, 이 교배 연산자는 두 부모의 거리 만큼 새로운 자식 유전자와의 거리를 일정하게 유지시켜준다. 즉 한 세대에서 각각의 유전자를 교배하고 변이를 수행 한 후에 다시 LinKernighan Heuristic을 수행하는데, 여기서 일반적인 방법으



그림 1. 휴리스틱 GA의 표준유형[1]

로 Crossover를 할 경우 부모의 형질과 거의 비슷하거나 똑같은 유전자가 나올 확률이 높다. 이런 내용에 대한 언급은 이미 Bernd Freisleben와 Peter Merz의 논문 "New Genetic Local Search Operator for the Traveling Salesman Problem"에서 다루워지고 개선을 위해서 DPXCrossover를 제안한 것이다.

3. 제안된 방법

이 논문에서 새롭게 제안된 방법은 TSP문제에서 Global optimal을 찾기 위해 사용되어지는 GA와 Local optimal을 찾기 위한 방법으로 LK를 이용한다. LK는 TSP문제를 풀어나가는데 있어서 좋은 Heuristic이라는 것은 이미 널리 알려졌다. 그러나 이런 종류의 Heuristic을 이용한 GA에서는 한가지 문제점을 찾을 수 있다. 각 세대의 유전자들을 LinKernighan Heuristic을 수행하면서, 한 세대

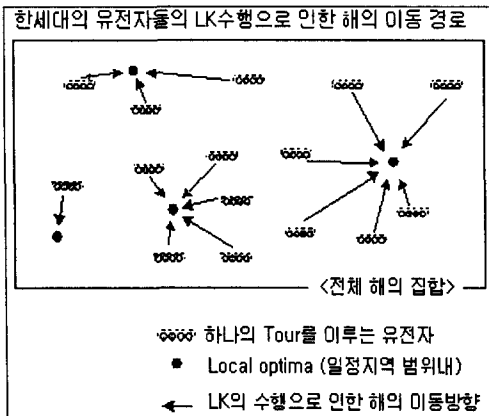


그림 2 한 세대의 유전자 이동

에서 다음 세대로 수렴할 때까지 계속해서 세대의 전이가 일어난다. 여기서, 각 유전자들은 많은 유사성을 갖게 된다. 그 이유는 전체 해의 집합들 속에 퍼져 있는 각각의 유전자들이 LK Heuristic을 수행하면서 주변에 있던 유전자들이 한곳(그림 2)으로 모이기 때문이다. 이런 유사성이 많은 유전자와

Crossover나 변이를 해도 결론은 부모의 유전자와 매우 유사한 유전자를 생성하게 될 가능성이 높게된다. 물론, 이런 현상을 피하기 위해서 Crossover이나 Mutation이 있지만, 이것으로는 부족하다는 것을 다음 4장에 보였다. 그래서 이런 중복 유전자들을 제거하여 각 유전자들의 다양성을 높임으로써 보다 효율적으로 해를 찾을 수 있다. 이 논문에서 새로 제안하려는 방법은 두 Tour간의 거리 (Hamming Distance[10])를 측정하여 거리가 일정 거리 이하이면, 하나의 노드를 여러개의 Fragment로 나누어 다시 재 조합하는 방법을 제안하고 있다. 여기서 Fragment 재 조합하는 방법은 그림 3과 같이 수행된다. 일단 각각의 세대마다 세대 재구성을 수행하여 유사한 Tour들을 변경함으로써, 유사한 Tour간의 교배나 변이로 인하여 부모와 같은 또 하나의 유전자를 만드는 것을 제거된다.

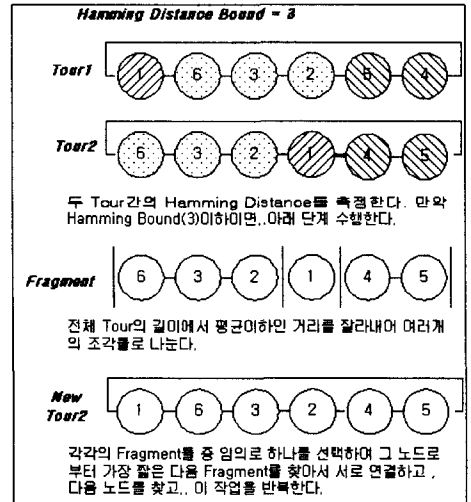


그림 3 개체 재구성 방법

또한 이런 방법은 다양한 해를 검색함으로써 Local optima에 빠지지 않게 도와준다.

```

Procedure 세대 재구성
begin
  Fragment K=RandomSelect(모든 Fragment)
  repeat
    for T:=0 to # of 나머지 Fragment do
      if Fragment T==ShortestFragment then
        break;
      endif
      Fragment K = Fragment K + Fragment T
    endfor;
  until Tour;
end;
    
```

그림 3. 세대 재구성을 위한 Fragment 조합 방법

그림 4에서 초기 유전자의 집단은 Nearest Neighbor Tour Construction heuristic[6]를 이용하여 생성하고, 여기서 초기의 d전자들을 각각 다른 Tour를 구성하기 위해서 임의로 시작노드를 정하여 각각의 다른 Tour를 구성하게 된다. Crossover Operator로는 DPXCrossover[1]를 수행한다. 이 Crossover Operator는 두 개의 부모들의 Hamming Distance[1][10]를 측정하여 자식 유전자의 부모와 일정

```

Procedure TSP-GA
begin
  NNH를 이용한 초기해 집단 구성
  foreach 유전자 do Linkernighan(유전자)
  repeat
    for i:=0 to # of crossover do
      Parent 1, Parent 2=RandomTwoSelect(유전자)
      Child=DPXCrossover(Parent1, Parent2);
      Mutation(Child);
      Linkernighan(Child);
    endfor;
    for i:=0 to 세대당 유전자의 수C2 do
      Dist=Hamming거리(TourI, TourII)
      if Dist<=Hamming_Bound then
        세대 재구성(TourI);
      endif
    endfor
  until 수렴;

```

그림 4 전체 수행도 및 새로운 접근법

한 거리를 유지함으로써, Local optima에 빠지지 않게 노력한다. 여기서 Hamming Distance란 두 개의 노드간에 거리를 측정하는 단위로서 두 Tour간에 다른 edge의 개수를 의미한다. 이런 거리 측정법은 K.D.Bose의 "Cost versus Distance in the Traveling Salesman Problem"에서 사용된다. TSP문제에서 최적의 local search로 알려진 Iterated Linkernighan[2]을 사용한다. 특히 세대 재구성을 함으로써 다음과 같은 효과를 본다. 즉 비슷한 해들에 대한 각각의 개체를 다른 곳으로 이동시킴으로써 해를 보다 다양하게 찾을 수 있고, 보다 효율적으로 찾을 수 있다.

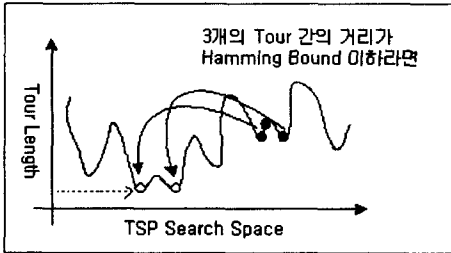


그림 5 재 조합으로 인한 해의 이동

그림 5는 이런 세대 재 조합으로 인한 해의 이동모습을 가상적으로 나타낸 것이다.

4. 수행과 계산 결과

위의 제안된 방법을 실험하기 위해서 Red Hat Linux 6.0 Tornado (PentiumII 233MHz, 64M)에서 C language로 구현되었다. Iterated Linkernighan Heuristic의 구현은 CONCORDE Library를 이용하였고, GA에 관련된 사항은 직접 프로그래밍 하였다.

실험에서는 TSP 예제로 널리 알려진 TSPLIB[5]에서 추출하여 실험을 하였다. 두 번째 컬럼은 TSPLIB에서 추출한 TSP예제의 이름이고, 세 번째는 최적 값/실험 결과 값이다. 즉, 최적 값은 그 TSP예제에서 알려진 최적 값이고 그 다음은 실험에서 얻어진 결과 값이다. 네 번째 컬럼은 실험 결과 값이 나오기까지 세대수의 평균값이다. 그 다음 컬럼은 결과 값이 나오기까지의 평균 소요 시간이다.

비교 대상을 위해서 (표1) 홀수 행의 '표준'은 2장에서 이미 언급한 표준적인 방법으로 구현하였고, 짝수 행의 '제안'은 여기

서 제안한 세대 재구성 방법(3장에서 제안한)을 이용한 것이다. 이것 외의 나머지 parameter들은 동일하다.

이 실험에서는 다음과 같은 parameter를 사용했다.

- ▶ Population Size : 30
- ▶ Crossover rate : 0.5 (즉 15개의 자손 생성)
- ▶ Mutation rate : 0.03
- ▶ HammingDistance Bound : PopulationSize/10
- ▶ 최대 세대 수 : 300

	TSP예제	최적값/결과값	평균 세대수	수행 시간(초)	
표준	rat195	2323	2323	56.6	59.79
제안			2323	35.35	55.55
표준	gil262	2378	2378	38	89.35
제안			2378	27	83.48
표준	lin318	42029	42029	153	677.3
제안			42029	98	650.25

표 1 제안된 GA 실험 결과

'표준'과 '제안'에서 평균세대수가 많이 차이가 났음에도 불구하고 수행 시간이 많이 개선이 없던 이유는 제안된 방법을 사용하기 위해서는 많은 작업(LK를 거의 1.3배 이상 더 수행한다.)을 거치기 때문이다. 더 큰 도시의 문제에서는 더 나쁜 결과도 나왔다. 좀더 수정 및 보완이 필요하다고 생각한다. 도시의 수가 100이하인 경우에는 제안된 방법이나 표준적인 방법 모두 10세대 이내에서 최적해를 구하기 때문에 실험적 가치가 없었다.

5. 결론과 앞으로의 연구 사항

이 논문에서 제안한 세대의 재 구성 방법은 비교적 작은 도시의 크기(48-400)에서 더 좋은 결과를 나타냈지만 도시의 사이즈가 커지면서 큰 차이점을 느낄 수 없었고, 더 나쁜 경우도 나타났다.

앞으로의 연구대상으로 좀더 재구성 방법을 효율적으로 구성하고 코드를 optimize한다면 나아질 수 있을 것이다. 보다 동적인 Hamming Distance Bound값을 알아내어 자동적으로 이 값을 변화하는 루틴을 추가한다면 더욱 좋아질 수 있을 것이다.

[ Reference ]

- [1] Bernd Freisleben and Peter Merz "New Genetic Local Search Operator for the Traveling Salesman Problem"
- [2] S.Lin and B.W. Kernighan "An Effective Heuristic Algorithm for the Traveling Salesman Problem"
- [3] MICHALEWICZ "유전자 알고리즘"
- [4] 기타노 히로아키 "유전자 알고리즘"
- [5] Gerhard Reinelt "TSPLIB 95"
- [6] Stephan Mertens, "TSP Algorithms in Action Animated Examples of Heuristic Algorithms"
- [7] A.E.Eibe, P-E.Raue, Zs Ruttkey "Genetic Algorithm with multi-parent recombination"
- [8] Bernd Freisleben and Peter Merz "A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems"
- [9] William M. Spears "Adapting Crossover in a Genetic Algorithm"
- [10] K.Bose "Cost versus Distance in the Traveling Salesman Problem"