

대칭 순회 판매원문제를 위한 Subtour 보존 교차 연산자

석상문^{1*} · 이홍걸² · 변성철¹

¹특허청, 정보심사팀 / ²경남대학교, e-비즈니스 학부

Subtour Preservation Crossover Operator for the Symmetric TSP

Sang-Moon Soak¹ · Hong-Girl Lee² · Sung-Cheal Byun¹

¹Information Systems Examination Team, Korean Intellectual Property Office, Daejeon 302-701

²Division of e-business, Kyungnam University, Gyeongsangnam-do 631-701

Genetic algorithms (GAs) are very useful methods for global search and have been applied to various optimization problems. They have two kinds of important search mechanisms, crossover and mutation. Because the performance of GAs depends on these operators, a large number of operators have been developed for improving the performance of GAs. Especially, many researchers have been more interested in a crossover operator than a mutation operator. The reason is that a crossover operator is a main search operator in GAs and it has a more effect on the search performance. So, we also focus on a crossover operator.

In this paper we first investigate the drawback of various crossovers, especially subtour-based crossovers and then introduce a new crossover operator to avoid such drawback and to increase efficiency. Also we compare it with several crossover operators for symmetric traveling salesman problem (STSP) for showing the performance of the proposed crossover. Finally, we introduce an efficient simple hybrid genetic algorithm using the proposed operator and then the quality and efficiency of the obtained results are discussed.

Keywords: Genetic Algorithm, Crossover Operator, Symmetric TSP

1. 서론

존 홀랜드가 1970년대 초반에 유전 알고리즘(Genetic Algorithm: GA)을 처음 소개한 이래로, 많은 연구자들이 최적화 문제를 해결하기 위한 새로운 기법으로 유전 알고리즘에 많은 관심을 보여왔다. 알고리즘의 단순함에 비해 복잡한 최적화 문제를 해결하는 유전 알고리즘의 능력이 관심을 끌기에 충분 했었다. 기본적으로 유전 알고리즘은 자연 선택(natural selection)과 자연 유전학(natural genetics)의 메카니즘에 기반을 둔 일종의 탐색 알고리즘이다(Goldberg, 1989). 따라서 유전 알고리즘에서는 기존의 어떤 알고리즘에서도 시도하지 않았던 해 집단(population)을 이용해서 해를 탐색해 나가는 특징을 지니고 있으며, 해 집단 내에 존재하는 해들을 개선 시켜 나가기 위해 선

택(selection), 교차(crossover)와 돌연변이(mutation) 연산자를 이용한다. 결국 이러한 연산자들이 자연계의 적자생존(survival-of-the-fittest)의 개념을 대신하기 위해 프로그램되고, 많은 진화 과정을 거친 후 해 집단 내의 해들이 최적해에 가까운 해를 찾을 때까지 이러한 과정을 반복 수행하게 된다.

유전 알고리즘에는 중요한 두 가지 탐색(search) 메카니즘이 있는데, 교차 연산자(crossover operator)와 변이 연산자(mutation operator)가 바로 그것들이다. 일반적으로 교차 연산자의 역할은 부모해들 간의 유전인자의 교환을 통해 어떤 해 주변의 해를 정밀하게 탐색하는 것이고, 변이 연산자의 역할은 하나의 해 자체 내에서 유전인자의 변화를 통해 이전의 진화 과정에서 탐색하지 못했던 새로운 탐색공간을 폭넓게 탐색하게 하는 것이다. 따라서 결국 최적화 문제들을 해결하기 위한 유

본 연구 결과물은 2007학년도 경남대학교 학술연구장려금 지원을 통해 이루어졌음.

* 연락처: 석상문, 302-701 대전광역시 서구 둔산동 920번지 정부대전청사 4동 특허청 정보심사팀 1306호, Fax : 042-472-3584,

E-mail : soakbong@hotmail.com

2006년 08월 접수; 2006년 11월 수정본 접수; 2007년 01월 게재 확정.

전 알고리즘의 성능은 이러한 연산자들에 의해 좌우되기 때문에 보다 더 우수한 해를 찾기 위해 그동안 수많은 연산자들이 개발되어왔다. 특히, 많은 연구자들이 이 두 연산자들 중에 교차연산자에 더 많은 관심을 보여 왔는데 그것은 위에서 언급한 것처럼 교차 연산자가 각 개체(individual)들 사이에 우수한 정보를 교환하는 역할을 수행하는 중요한 요소일 뿐만 아니라 최적해(global optimal solution)가 일반적으로 지역 최적해(sub-optimal solution) 근처에 존재하는 것으로 알려져 있기 때문이다(Boese *et al.*, 1994).

반면에 몇몇 연구자들은 단지 변이 연산자만을 사용하는 유전 알고리즘을 제안하기도 했었다(Morikawa *et al.*, 1992; Tao and Michalewicz, 1998; Falco *et al.*, 2002). 하지만 본 논문에서는 앞에서 언급한 것처럼 일반적으로 우수한 해 근처에 보다 더 좋은 해들이 많이 있는 것으로 알려져 있고 이를 탐색하는 기능이 교차 연산자에 의해 구현되기 때문에 변이 연산자 대신 교차 연산자의 개발에 초점을 맞추며 특히, 대칭 순회 판매원 문제(the symmetric traveling salesman problem: STSP)를 해결할 수 있는 교차 연산자 개발 및 기존의 연산자들과의 비교에 초점을 맞춘다.

오랫동안 유전 알고리즘은 대칭 순회 판매원 문제를 해결하기 위해 많이 응용되어 왔다(Davis, 1985; Starkweather *et al.*, 1991; Mathias *et al.*, 1992; Bui and Moon, 1994; Maekawa *et al.*, 1996; Yang, 1997; Katayama *et al.*, 1998; Nguyen *et al.*, 2000). 일반적으로 대칭 순회 판매원 문제는 NP-complete 문제로 알려져 있으며(Karp, 1972), 이는 결국 이러한 문제가 아주 넓은 탐색공간을 지니며 해결하기 아주 어렵다는 것을 의미한다.

대칭 순회 판매원 문제를 유전 알고리즘을 이용해서 해결하려고 시도했던 최초의 연구는 Grefenstette 등(Grefenstette *et al.*, 1985)과 Goldberg 등(Goldberg and Lingle, 1985)의 연구였다. 그들이 유전 알고리즘을 이용해서 대칭 순회 판매원 문제를 해결하는 방법을 소개한 이후, 근사해(sub-optimal solution)나 최적해(optimal solution)를 찾기 위한 많은 노력들이 연산자 개발에 기울여져 왔다.

대부분 유전 알고리즘의 사용 초기에는 단순히 치환에 기반을 둔 교차연산자(permutation-based crossover operator)의 개발이 주를 이루었으며, 일반적으로 알려진 치환에 기반을 둔 교차 연산자들로는 사이클 교차 연산자(Cycle crossover: CX) (Oliver *et al.*, 1989), 순서 교차 연산자(Order Crossover: OX) (Davis, 1985; Shang and Li, 1991; Murata and Ishibuchi, 1996), 위치 기반 교차 연산자(Position-Based Crossover : PBX) (Syswerda, 1991), 순서 기반 교차 연산자(Order-Based Crossover: OBX) (Syswerda, 1991), 부분 맵핑 교차연산자(Partial-Mapped Crossover : PMX) (Goldberg and Lingle, 1985) 등이 있다. 그리고 최근에는 연구자들의 관심이 subtour(본 논문에서는 subtour를 하나 이상의 노드가 연속적으로 결합된 운송 경로를 의미한다) (Yamamura, *et al.*, 1992; Maekawa *et al.*, 1996; Yamamura *et al.*, 1996; Katayama *et al.*, 1998)나 에지(edge)(Whitley *et al.*, 1989; Starkweather *et al.*, 1991; Mathias *et al.*, 1992), (Maekawa *et al.*,

1996; Nguyen *et al.*, 2000) 또는 에지와 두 도시 사이의 거리 (Grefenstette *et al.*, 1985; Yang, 1997; Qu and Sun, 1999; Yang and Stacey, 2001)와 같은 부모해가 지니고 있는 정보를 사용하는 방향으로 바뀌어 왔다.

또한 최근에는 작은 해집단을 사용하여 순회 판매원 문제를 해결하기 위한 시도(Baraglia *et al.*, 2001)와 본 논문에서 사용하는 순서(order)기반 인코딩 방법과는 달리 2차원 평면상에서 도시들이 분포하는 이미지 자체를 해 표현법으로 사용하는 것에 대한 연구가 있었다(Jung and Moon, 2002; Seo and Moon, 2002). 비록 세 논문 모두 아주 우수한 결과를 보여 주었지만, 본 논문의 목적인 동일한 조건에서 교차 연산자의 우수성을 비교하기 위해서는 적합하지 않아 고려하지 않았다.

앞에서 언급한 것처럼 유전 알고리즘의 기본 원리가 적자생존(survival-of-the-fittest)이기 때문에 만약 어떤 해가 다른 해들보다 더 우수하다면, 그 해는 자신의 유전 정보를 사용해서 더 많은 자손(offspring)을 다음 세대에 생성을 하게 될 것이고, 이러한 가치 있는 정보들이 더 많은 자손들에게 전이되어 질 것이다. 이러한 과정을 반복함으로써 유전 알고리즘은 더 우수한 해를 생성하고 결국 목적을 달성하게 된다. 따라서 Starkweather *et al.*(Whitley *et al.*, 1989; Starkweather *et al.*, 1991)와 Moscato(Moscato, 1989)가 지적인 것처럼, 부모해의 정보(parental information)를 사용하는 것이 우수한 해를 찾기 위한 좋은 대안이 될지도 모른다.

따라서 본 논문에서는 부모해가 지니고 있는 이웃하는 유전 인자들의 정보를 이용하는 것이 중요하며 이러한 정보를 보존해야만 한다는 것을 보여 주기 위해 새로운 subtour 기반 교차 연산자(subtour-based crossover operator)를 제안한다. 그리고 제안하는 연산자가 기존에 대칭 순회 판매원 문제를 해결하기 위해 제안된 치환에 기반을 둔 교차 연산자(permutation-based crossover), 에지 또는 subtour에 기반을 둔 교차 연산자(edge or subtour-based crossover)들과의 비교 실험을 통해 제안하는 새로운 교차 연산자의 우수성을 보인다. 마지막으로, 제안하는 교차 연산자를 사용하는 단순한 혼합형 유전 알고리즘(hybrid genetic algorithm)을 소개하고 찾아낸 결과의 우수성에 대해 간단히 언급한다.

2. 순회 판매원 문제를 위한 교차 연산자들

오랫동안 많은 교차 연산자들이 순회 판매원 문제를 해결하기 위해 개발되어 왔다. 그중 몇몇 교차 연산자들은 부모들 사이의 부분적인 정보만을 사용하였으며, 또 다른 연산자들은 교차 연산자 적용시 선택된 부모해가 지니고 있는 에지 정보나 subtour 정보를 사용하여 개발되었다.

본 절에서는 그동안 개발되어온 많은 교차 연산자들을 자손(offspring)을 생성하기 위해 사용되는 기본 데이터 정보에 기초해서 네 그룹으로 분류한다.

2.1 치환 기반 교차 연산자(Permutation-Based Crossover Operators)

유전 알고리즘을 이용해서 순회판매원 문제를 해결하려고 시도한 초기에 치환 기반 교차연산자들이 많이 등장하였다. 주로 이 그룹에 속하는 교차 연산자들은 교차 연산을 위한 기본 정보로써 각 부모해에 있는 유전 인자의 순서(order)나 위치(position) 정보를 사용하였다. 이러한 치환 기반 교차 연산자의 주된 장점은 아주 간단하고 명확하기 때문에 다른 분류의 연산자들보다 구현이 쉽고 계산 시간이 상대적으로 짧다는 것이다. 또한 이러한 치환 기반 교차 연산자들은 유전 알고리즘의 가장 기본 교차 연산자인 1점, 2점 교차 연산(one or two point) 또는 다점(multi-point) 교차 연산자의 변형들이 많이 있기 때문에 이러한 연산자들을 순회 판매원 문제에 적용하기 위해서는 생성된 자손의 유효성을 유지시키기 위해서 일종의 수정(repair)과정을 필요로 한다. 특히, Poon과 Carter에 의해 제안된 Tie-Breaking 교차 연산자(TBX) (Poon and Carter, 1995)의 경우에는 기존의 2점 교차 연산자와 매우 유사하다.

따라서 이러한 부류의 교차 연산자들은 기존의 1점 또는 다점 교차 연산자들처럼 단점을 지니고 있는데, 교차 연산자를 통해 생성된 자손이 그 부모보다 우수한 해일 가능성이 낮다는 것이다. 왜냐하면 이러한 연산자들이 유효하지 않은 경로를 생성하는 것을 피하기 위해 수정 과정을 거침으로써 부모의 유전 정보를 깨뜨린다거나 부모해가 지나고 있는 우수한 정보들 중에서 단지 부분적인 정보만을 이용하기 때문이다.

이러한 부류에 속하는 교차 연산자로는 PMX(Goldberg and Lingle, 1985), OX(Davis, 1985; Murata and Ishibuchi, 1996), Enhanced Order Crossover(EOX) (Shang, and Li, 1991), CX (Oliver *et al.*, 1989), PBX(Syswerda, 1991), OBX(Syswerda, 1991), TBX(Poon and Carter, 1995), Maximal Preservative Crossover(MPX) (Mühlenbein, 1992), Schleuter 교차연산자(SCHX) (Kzuber and Whitley, 1994)와 상대적 순서 교차 연산자(Relative Order Crossover : ROX) (Moscato, 1989) 등이 있다.

2.2 에지 기반 교차 연산자(Edge-Based Crossover Operators)

치환 기반 교차 연산자들이 제안된 이후에 연구자들의 관심은 점점 단순한 치환에서 부모해들이 오랜 진화과정을 통해서 습득한 유전 정보를 이용하는 쪽으로 바뀌게 되었다. 어쨌든 이러한 유전 정보를 사용하게 됨으로써 해의 개선은 이루어졌지만 알고리즘이 보다 더 복잡해지게 되었고, 그러므로 해서 계산 시간은 치환 기반 연산자들에 비해 더 요구되었다.

부모해가 지니는 에지의 정보를 사용한 최초의 교차 연산자는 휴리스틱 교차 연산자(Heuristic Crossover: HX)였다 (Grefenstette *et al.*, 1985). 그 이후 탐욕적 교차연산자(Greedy Crossover: GX) (Yang, 1997), 탐욕적 선택 교차 연산자(Greedy Selection Crossover: GSX) (Qu and Sun, 1999) 그리고 50 : 50 탐

욕적 교차 연산자(Fifty-fifty Greedy Crossover) (Yang and Stacey, 2001)와 같은 다양한 변형들이 등장하게 되었다.

대표적인 에지 기반 교차 연산자는 Whitley 등에 의해 제안된 에지 재결합 교차 연산자(Edge Recombination Crossover : ER) (Whitley *et al.*, 1989)이다. 이 연산자는 부모해 내에 있는 이웃하는 유전 인자 정보만을 재결합하는데 초점을 맞춰 개발되었다. 그리고 ER이 제안된 이후에 Starkweather 등(Starkweather *et al.*, 1991)은 이러한 점을 보완하기 위해 강화된 ER 연산자(Enhanced Edge Recombination Crossover : EER)를 제안하였다. 기본 개념은 ER 연산자에서 고려되지 않았던 두 부모해가 가지는 공통 에지 정보를 보존하는 것이었다. 이를 위해 EER 연산자는 공통 에지에 대한 정보를 지니는 표를 생성해서 이러한 에지들에게 자손해를 형성하는 과정에서 더 빨리 선택될 수 있는 우선권(priority)를 주었다. 이 이외에도 ER 연산자를 개선하기 위한 많은 노력들이 있었다(Mathias *et al.*, 1992; Nguyen *et al.*, 2000).

다른 에지 기반 교차 연산자로는 에지 교환 교차 연산자(Edge Exchange Crossover : EEX)(Maekawa *et al.*, 1996)가 있는데 이는 부모가 지니고 있는 에지들을 서로 교환하는 방법으로 새로운 자손을 생성하였다.

2.3 매트릭스 기반 교차 연산자(Matrix-Based Crossover Operators)

매트릭스 기반 교차 연산자들은 엄밀히 따지면 에지 기반 교차 연산자에 포함이 될지도 모른다. 하지만 본 장의 서두에서 언급한 것처럼 교차 연산자를 위해 사용되는 기본 데이터 정보에 따라 교차 연산자들을 분류하고 있기 때문에 이를 다른 절로 분류해서 간단하게 다룬다.

몇몇 연구자들은 매트릭스 기반 교차 연산자들을 제안해 왔다(Fox and McMahan, 1991; Bui and Moon, 1994). 하지만 이러한 연산자들이 많은 연구자들의 관심을 끌지는 못했었다. 그 이유는 Fox와 McMahan(Fox and McMahan, 1991)이 보여준 것처럼 다른 연산자들이 45분 동안 80회에서 3,000번의 세대(generation)를 진화시켜 나가는 반면에 그들이 제안한 횡단 연산자(intersection crossover operator)와 통합 연산자(union crossover operator)는 대략 80회의 세대(generation)를 수행했었다. 결국 이는 매트릭스 기반 교차 연산자가 지니고 있는 단점으로 해 표현법 자체가 매트릭스 형태를 유지하고 있기 때문에 교차 연산자 수행에 상대적으로 막대한 계산시간을 요구하게 된다. 따라서 본 논문에서는 매트릭스 기반 교차 연산자들은 다루지 않는다.

2.4 Subtour 기반 교차 연산자(Subtour-Based Crossover Operators)

Subtour 기반 교차 연산자들은 에지 기반 교차 연산자들보다 좀 더 진화된 연산자들로 에지 기반 교차 연산자들과 비교

해서 다른 특징들을 지닌다.

우선, 이러한 분류에 속하는 연산자들은 교차 연산자를 위해 선택되는 부모해가 지니고 있는 공통 에지 정보뿐만 아니라 공통 subtour 정보들을 찾아내고 비록 약간의 차이는 있을 지라도 그 이후 이러한 정보들을 서로 교환하거나 전도(invert) 시킴으로 해서 많은 자손들을 생성하고, 생성된 자손들 중에서 거리비교를 통해서 두 자손을 선택하는 과정을 수행한다.

대표적인 subtour 기반 교차 연산자로는 subtour 교환 교차 연산자(Subtour Exchange Crossover : SXX) (Yamamura, *et al.*, 1992; Yamamura *et al.*, 1996), 완전한 subtour 교환 교차 연산자(Complete Subtour Exchange Crossover : CSEX) (Katayama *et al.*, 1998)과 거리 보존 교차 연산자(Distance Preserving Crossover : DPX) (Freisleben and Merz, 1996)가 있다.

기존에 유전 알고리즘을 이용해서 순회판매원 문제를 해결하기 위해 사용된 교차 연산자 중에서 가장 우수한 성능을 보이는 것은 DPX 연산자이다. DPX 연산자도 역시 subtour를 열거하는 방법을 사용하고 탐욕적 재결합 방법(greedy reconnection procedure)을 사용해서 완성된 경로를 형성한다(Freisleben and Merz, 1996). 어쨌든, DPX 연산자도 완전한 경로를 형성하기 위해서 부모해의 에지 정보 대신에 모든 도시들의 거리 정보를 사용하였다.

이러한 subtour 기반 교차 연산자들 또한 심각한 결점을 지니고 있다. 만약 교차 연산자를 위해서 선택된 부모해가 동일한 해라면 그들의 부모와 다른 자손을 형성할 수 없다는 것이다. 왜냐하면 이러한 경우에 추출되는 subtour 자체가 완성된 경로가 되고 결국에는 부모해가 자손이 되기 때문이다. 어쨌든 진화과정이 점점 더 증가할수록, 선택전략(selection strategy) 때문에 해 집단 내에 동일한 해의 개수는 기하급수적으로 증가하게 될 것이고, 이 때문에 해의 개선은 점점 더 어려워지게 된다. 따라서 subtour 기반 교차 연산자를 사용하는 유전 알고리즘은 전적으로 해의 개선을 변이 연산자에 의존하게 된다. 그러므로 이러한 부류의 교차 연산자를 순회판매원 문제에 적용하기 위해서는 보완적인 방법이 필요하다.

3. Subtour 보존 교차 연산자(Subtour Preservation Crossover Operator)

본 논문에서 제안하는 subtour 보존 교차 연산자(Subtour Preservation Crossover Operator : SPX)는 기존의 다른 subtour 기반 교차 연산자들과 유사한 subtour 탐색 및 추출 방법을 사용하지만 유효한 경로를 형성하기 위해 사용하는 방법에서 차별성을 지닌다. SXX와 CSEX의 경우는 부모들이 지니고 있는 공통의 subtour를 서로 교환하거나 전도(invert)시키는 과정을 통해 생성될 수 있는 모든 가능한 자손들을 생성하고 생성된 자손들 중에서 가장 짧은 경로를 형성하는 두 자손을 다음 자손으로 선택하게 된다. 그리고 DPX는 subtour의 끝 점들끼리

연결 가능한 가장 가까운 끝점들을 연결시켜 나가는 탐욕적 재결합 과정(greedy reconnection procedure)을 사용한다.

이에 반해 SPX는 진화 과정을 통해 전파된 부모의 유전 정보를 사용하여 원하는 만큼 많은 자손을 생성할 수 있을 뿐만 아니라 부모해가 지니고 있는 유전 정보를 최대한 활용하여 subtour를 연결 시켜나간다.

또한 앞 장에서 언급한 것처럼 만약 동일한 부모가 교차 연산자를 위해 선택될 경우, 어떤 subtour 교차 연산자도 부모해와 다른 자손을 생성하지 못한다. 그래서 이러한 연산자들은 해의 성능을 개선 시켜 나갈 수 없으며 이 이후에는 해의 개선을 위해 변이 연산자에 의존해야만 한다. 하지만 SPX의 경우 이러한 문제점을 해결하기 위해 부모해 대신에 새로운 우수한 해를 해 집단에 소개하는 방법을 사용한다. 좀 더 자세한 내용은 다음 절에서 소개한다.

그리고 만약 SXX와 CSEX의 경우 $Subtour_1(1, 2, 3, 4)$ 와 $Subtour_2(1, 2, 3, 4)$ 와 같이 동일한 순서를 가진 도시의 subtour를 찾았다면, 이들은 현재의 부모와 동일한 자손을 생성하게 되는 단점 또한 지니고 있다. 하지만 SPX의 경우는 동일한 subtour를 찾을지라도 subtour와 도시들을 재결합하는 각 과정에서 다른 시작점을 선택하게 함으로써 다른 자손을 형성할 수 있다. 또한 SXX와 CSEX는 진화과정을 통해 축적된 부모해가 지니고 있는 많은 정보들 중에서 단지 subtour 만을 사용한다. 즉, SXX와 CSEX는 subtour 만을 사용한다는 점에서 치환 기반 교차 연산자와 차이가 있을 뿐이며 나머지 과정은 치환 기반 교차 연산자들이 사용하는 치환 과정과 아주 유사하다. 반면에 SPX는 부모가 지니고 있는 정보의 손실을 최소화하기 위해 부모가 지니고 있는 가능한 모든 정보를 이용하려고 시도한다.

3.1 Subtour 분석

Katayama 등(Katayama *et al.*, 1998)은 해의 평균 성능, 공통 subtour의 평균 개수 및 평균 길이와 진화과정(generation) 사이의 관계를 $pop_size = 32$ (해 집단의 크기)를 사용해서 4가지 문제에서 조사하였다. 하지만 진화 과정이 증가함에 따라 subtour의 개수가 어떻게 변화하는지 그리고 얼마나 많은 동일한 부모가 교차 연산자를 위해서 선택되는지는 조사하지 않았다. 이러한 것들은 앞에서 언급한 것처럼 알고리즘의 성능과 관련되어 있는 아주 중요한 척도이다. 따라서 본 논문에서는 이러한 척도들을 조사하였다.

본 실험에서는 SPX 연산자 적용시 동일한 부모해가 선택될 경우 다른 우수한 해로 대체하는 과정을 채택하지 않고 실험을 하였다(<Figure 3> 단계 4 참조). 그 이유는 본 실험의 목적이 기존의 subtour 기반 교차 연산자들이 지니고 있는 단점을 보이기 위함이기 때문이다.

<Figure 1>은 SPX를 이용해 30번의 반복 실험에서 1000세대(generation) 동안 subtour의 평균 개수와 길이의 변화를 보여준

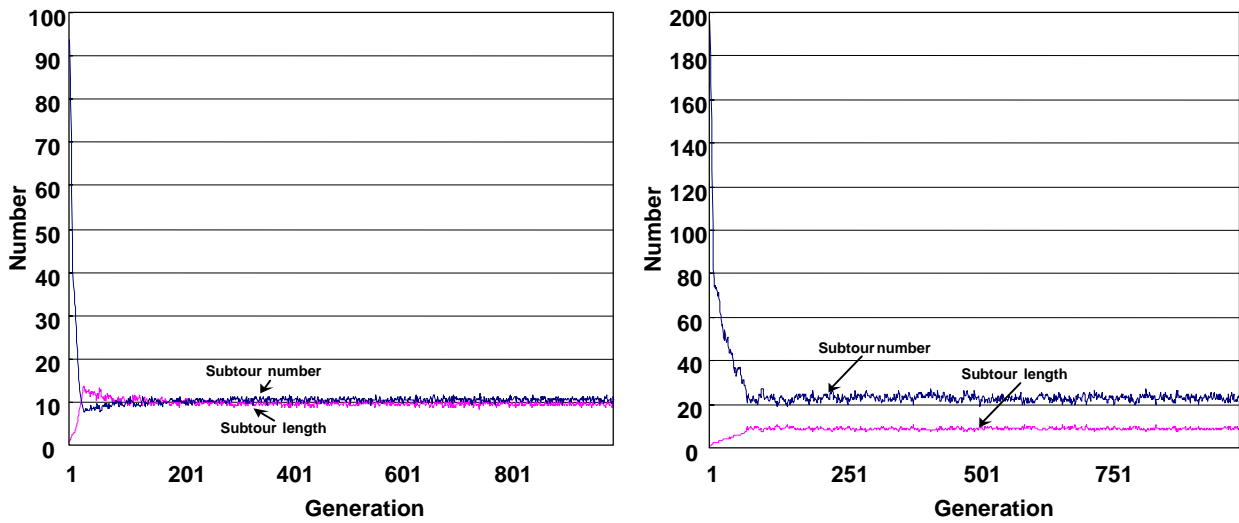


Figure 1. The average number and length of subtour : (a) Instance (left) = kroA100, (b) Instance (right) = kroB200, (pop_size = 200, Pc = 0.6, Tournament selection, Total generation = 1000, Iteration = 30)

다. <Figure 1>에서 subtour의 개수는 Katayama 등이 사용한 것과 달리 하나의 도시도 subtour로 간주하였다.

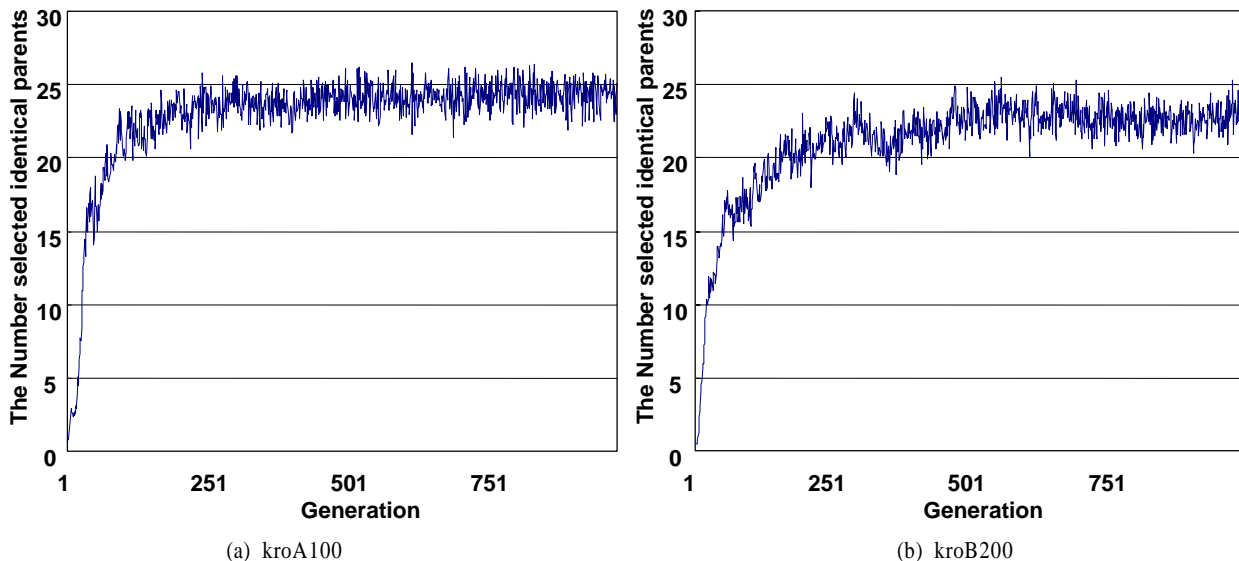
<Figure 1>에서 확인할 수 있는 것처럼 초기 진화 과정에서는 평균 subtour의 개수가 아주 높게 나타났지만 빠르게 감소해서 100세대 이후에서는 일정한 상태를 유지함을 확인할 수 있다. 그리고 subtour의 평균 길이는 (a)의 경우에는 약 10이었으며, (b)의 경우는 약 9였다.

<Figure 1>을 통해서 진화과정이 증가함에 따라 subtour의 개수는 감소하는 반면 subtour의 길이는 증가함을 확인할 수 있는데, 이는 아주 흥미롭게도 유전 알고리즘의 기본 이론인 “schema theorem” and “building block hypothesis”와 일치하는 결과이다(Michalewicz, 1992). 즉, 이는 SPX와 같은 subtour 기반 교차 연산자가 유전 알고리즘의 기본 메카니즘과 일치하는

아주 합당한 교차 연산자임을 보이는 결과이다.

<Figure 2>는 교차 연산자가 적용될 때 동일한 두 부모가 선택되는 평균횟수를 보여준다. 200세대가 넘어갈 경우, 동일한 부모가 선택될 횟수가 (a)와 (b)의 경우는 약 20을 넘고 있음을 확인할 수 있다. 즉, 이는 동일한 부모가 전체 교차 연산자 과정 동안에 30% 이상 선택됨을 보여주는 결과이다(교차 확률 Pc = 0.6). 또한 (c)의 경우는 편차가 있지만 대부분의 경우에서 약 20을 상회하였다. 그리고 (d)의 경우는 진화과정이 200을 넘어감에 따라 약 25를 상회함을 확인할 수 있다.

그리고 만약 이러한 문제점을 해결하기 위해 부수적인 방법을 고려하지 않는다면 교차 연산자가 새로운 자손을 생성할 수 없기 때문에 해의 개선은 점점 더 어려워질 것이라는 것을 보여준다.



(a) kroA100

(b) kroB200

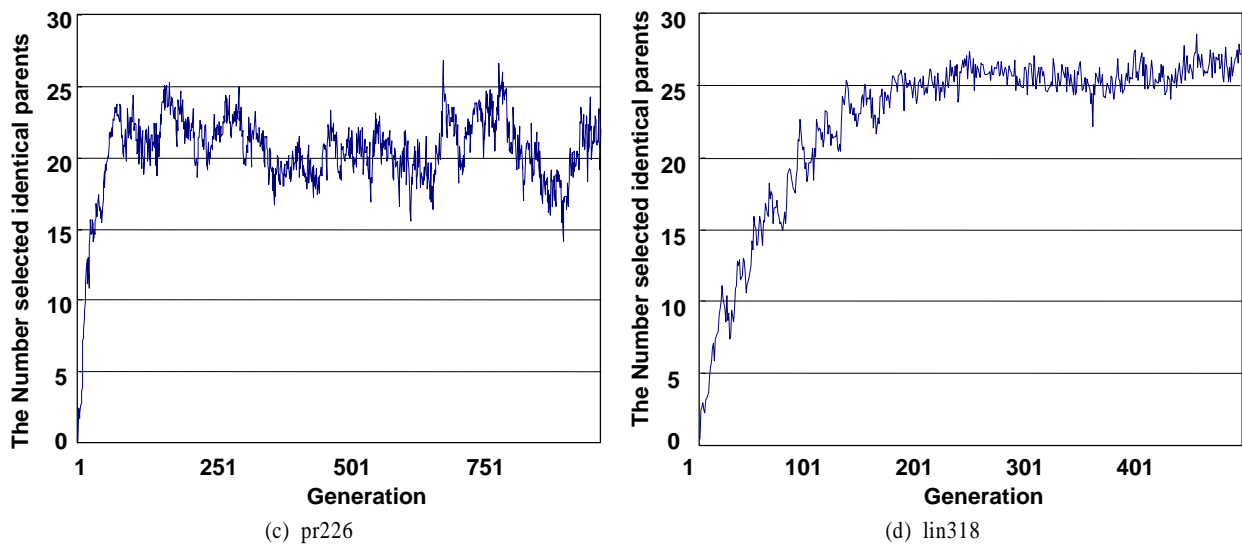


Figure 2. The average number of times that identical parents are selected: pop_size = 200, Pc = 0.6, Proposed Tournament selection, Total generation = 500 to 1000, Iteration = 30 (in case of (a) and (b)) and 20 (in case of (c) and (d))

3.2 새로운 해 생성

만약 동일한 부모해가 교차 연산자를 위해 선택된다면 SPX는 변이 연산자 또는 치환 방법과 같은 다양한 방법들을 이용해서 새로운 해들을 생성할 수 있다. 하지만 만약 단순한 교차 연산자나 치환 방법에 의해 생성된 자손들을 해집단에 새로운 해로 대체한다면, 이러한 해들의 생성단계가 대부분 진화 과정의 후반부에서 빈번하게 발생할 것이기 때문에 이미 해집단 내에는 좋은 적합도 값을 지닌 해들에 의해서 생존경쟁에서 살아남기가 쉽지 않을 것이다. 따라서 SPX에서 동일한 부모해가 선택되었을 경우 새로운 해를 생성하는 방법은 다음과 같이 이루어진다.

만약 동일한 부모해가 SPX 과정에서 선택된다면 우선 임의

로 두 해를 생성하고, 그 다음 해 집단 내에 이미 존재하는 우수한 해들과 경쟁이 가능하도록 하기 위해 생성된 각각의 해에 지역 탐색 방법(local search method)을 적용해서 새로운 해를 생성한다. 이렇게 함으로써 동일한 부모해가 선택되었을 경우에 해의 개선 없이 진화과정이 이루어지는 것을 방지할 수 있을 뿐만 아니라 단순한 변이 연산자를 이용해서 새로운 해를 생성함으로써 발생하는 새로운 해들의 해 집단 내에서의 빠른 도태과정 또한 방지할 수 있다.

3.3 Subtour 보존 교차 연산자(Subtour Preservation Crossover Operator : SPX)

<Figure 3>은 SPX의 subtour 연결 과정을 나타낸다. SPX의

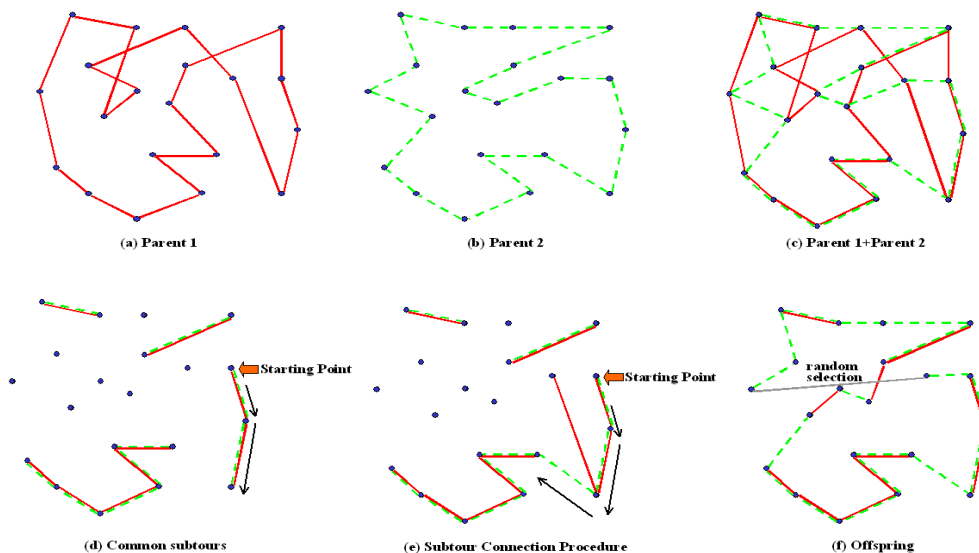


Figure 3. Subtour connection procedure

첫 번째 단계에서는 모든 가능한 subtour들을 열거하고 그 다음 subtour들을 연결하는 연결 방법을 이용해서 각각의 subtour를 연결 시켜 자손 해를 생성한다. 이때 subtour 연결 과정에서 선택 가능한 예지는 <Figure 3>의 (e)에서 확인할 수 있는 것처럼 각 부모로부터 하나의 예지씩 둘 중에 거리가 짧은 예지를 선택한다. 이때 만약 선택되는 예지와 관련된 노드가 이미 subtour 연결과정에서 선택되었다면 다른 예지를 선택하고 둘 다 이미 선택되었다면 이전에 선택되지 않은 노드들 중에서 임의로 하나의 노드를 선택해서 앞의 과정을 반복 수행한다.

<Figure 4>는 subtour 보존 교차 연산자의 전반적인 절차를 보여주고 있다. 여기서 단계 4는 교차 연산자 적용시 동일한 부모해가 선택될 경우 이를 개선하기 위한 단계이며 단계 5는 위에서 언급한 subtour 연결 과정을 나타내고 있다.

3.4 변이 연산자

본 논문에서처럼 대칭 순회 판매원문제를 위해서 순서기반 해 표현법(order based encoding method)를 사용할 경우 순서에 대한 제약 즉, 모든 도시가 스트링(string)에 한 번씩만 존재해야 한다는 조건을 깨뜨리지 않는 한 어떤 변이 연산자를 사용해도 무방하다.

단계 0. 생성하기를 원하는 자손의 개수를 설정한다.

단계 1. 두 부모해로부터 모든 subtour를 추출한다.

단계 2. 단계 0에서 설정된 자손의 개수만큼 자손이 생성되면 종료하고, 그렇지 않으면 다음 단계를 반복 수행한다.

단계 3. 추출된 subtour의 끝점(subtour의 길이가 1인 경우는 하나의 끝점)들 중에서 임의로 하나의 끝점을 출발 지점(C_i)으로 선택하고 생성하는 tour에 삽입하고 현재의 위치를 $C_i \leftarrow C_i$ 로 갱신한다.

단계 3.1 만약 선택된 끝점의 subtour의 길이가 1보다 크면 subtour 내에 존재하는 모든 점들을 tour에 삽입하고 현재의 위치를 subtour의 다른 편 끝점(C_i)으로 설정한다($C_i \leftarrow C_i$).

단계 4. subtour의 개수가 1이면, 임의로 두 개의 새로운 해를 생성하고 생성된 새로운 해에 지역 탐색 방법(local search method)을 적용하여 새로운 자손해를 생성하고 단계 2로 간다.

단계 5. 완전한 tour가 생성될 때까지 다음 과정을 반복 수행한다.

단계 5.1 현재의 도시에서 선택 가능한 도시를 두 부모해에서 찾아낸다(<Figure 3>의 (e) 참조).

단계 5.2 만약 선택 가능한 도시가 하나라면 그 도시를 생성 중인 tour에 추가하고, 현재의 위치를 $C_i \leftarrow C_i$ 로 갱신한다.

단계 5.3 만약 선택 가능한 도시가 두 개라면 둘 중에 거리가 짧은 쪽을 선택한다.

단계 5.4 만약 선택하는 도시가 이미 이전에 선택된 도시라면, 나머지 하나의 도시를 선택하고 이 도시 또한 이전에 선택된 도시라면, 이전 단계에서 tour에 포함되지 않은 선택 가능한 도시들 중에서 임의의 한 도시를 선택한다. 그리고 선택된 도시를 tour에 삽입하고, 선택된 도시가 포함된 subtour의 길이가 1 이상이면 모든 subtour에 있는 도시들을 차례대로 tour에 삽입하고 subtour의 끝점(C_i)을 현재위치로 갱신한다($C_i \leftarrow C_i$).

단계 5.5 단계 2로 돌아간다.

Figure 4. Subtour Preservation Crossover (SPX)

기존에 대칭 순회 판매원문제를 적용되었던 다양한 변이 연산자들이 있는데 그 중 세 가지 변이 연산자들에 대해 설명하도록 한다.

우선 교환(swap) 변이 연산자는 가장 단순한 형태로 해 표현법 내에 임의의 두 도시를 선택해서 서로 바꾸는 형태로 구현되고, 역순서변환(inversion) 변이 연산자는 해 표현법 내에 임의의 두 지점을 선택하여 그 구간 안에 있는 도시들의 순서를 앞뒤로 바꾸는 형태로 구현된다. 그리고 마지막으로 더블 브릿지 킥 무브(double bridge kick move) 변이 연산자는 <Figure 5>처럼 임의의 네 예지(edge)를 선택해서 서로 바꾸는 연산자이다. 더블 브릿지 킥 무브(double bridge kick move) 변이 연산자의 경우에는 Martin 등(Martin *et al.*, 1991)이 밝힌 바와 같이 표준(standard) Lin-Kernighan 알고리즘과 함께 사용될 경우 효과적인 것으로 알려져 있다.

하지만, 본 논문의 경우 교차연산자 사이의 성능비교가 목적이기 때문에 비교적 구현이 간단한 역순서변환(Inversion) 변이 연산자를 모든 실험을 위해 사용하였다.

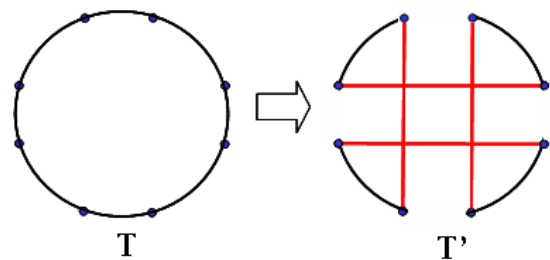


Figure 5. The double bridge kick move

4. 실험 결과

본 논문에서는 두 종류의 실험을 수행하였다. 첫 번째 실험은 기존의 여러 교차 연산자들과 제안하는 SPX 연산자와의 성능을 비교하기 위한 실험이고, 두 번째 실험은 SPX를 사용하

는 간단한 혼합형 유전 알고리즘(simple hybrid genetic algorithm)을 통해 제안하는 SPX 연산자를 이용할 경우 해의 개선의 정도와 가능성을 보이기 위한 실험이다

모든 알고리즘들은 Visual C++ 프로그램으로 구현이 되었으며, Pentium IV 1.6Ghz 개인용 컴퓨터에서 실험이 이루어졌다. 그리고 모든 실험 문제들은 참고문헌에 있는 TSPLIB (TSPLIB)사이트를 통해서 얻었다.

4.1 성능 비교

SPX의 우수성을 보이기 위해 SPX의 성능과 기존에 TSP를 위해 적용된 우수한 교차 연산자들과 비교 실험을 하였다. 각각의 교차 연산자들을 공정하게 비교하기 위해 본 논문에서는 단순 순수 유전 알고리즘(simple pure genetic algorithm)을 사용하였다. 그리고 본 논문에서는 공정한 교차연산자 비교를 위해 변이연산자, 교차확률, 변이확률 등 별도의 언급이 없는 한 아래에 설명한 것과 동일한 환경에서 실험을 수행하였다.

실험에서 사용된 유전 알고리즘의 구조를 간단하게 설명한다

초기해는 임의로 생성되었고, 선택전략을 위해서 토너먼트 크기가 2인 토너먼트 전략이 사용되었으며, 엘리트 보존 전략 또한 사용되었다. 즉, 새로운 진화 단계(generation)에서 가장 우수했던 해가 이전 진화 단계에서 가장 우수했던 해보다 못하다면, 이전 진화 단계에서 가장 우수했던 해를 현 단계에서 계속해서 유지시켜 나가며 이때 유지시켜 나가는 엘리트 해의 개수는 3개로 하였다. 그리고 변이 연산자로 역순서변환(Inversion)

변이 연산자(inversion mutation operator)가 사용되었다. 그리고 해 집단의 크기(pop_size), 교차 확률(P_c), 변이 확률(P_m)은 각각 200, 0.6, 0.4를 이용하였고, 종료조건으로는 최적해를 찾거나 또는 최종해가 N 세대 동안 변화가 없을 경우 종료하는 것으로 하였으며, 여기서 N 는 1,000으로 설정하였다. 또한 모든 실험은 30번씩 반복해서 이루어졌으며, SPX에 의해 생성되는 자손의 개수는 기존의 연산자들과 형평성을 고려해서 2로 설정을 하였으며, 동일한 부모해가 선택될 경우 적용하는 지역 탐색 방법(local search method)으로는 2-opt 기법을 사용하였다. 2-opt 기법을 사용한 이유는 이 기법이 TSP를 위한 다른 지역 탐색 방법들에 비해 상대적으로 간단할 뿐만 아니라 계산시간 또한 적게 소요되기 때문이며, 만약 이렇게 적용되는 지역 탐색 방법에 의해 SPX의 성능이 좌우된다면 더 좋은 지역 탐색 방법을 통해 더 우수한 결과를 얻을 수 있을 것이다.

실험결과에서 보여지는 해의 성능(quality)은 다음의 식 (1)에 의해 계산되었으며, 이는 최적해를 초과하는 퍼센티지를 의미한다.

$$quality = \frac{Fitness - Optimal}{Optimal} \times 100(\%) \quad (1)$$

<Table 1>은 각 교차 연산자의 실험 결과를 보여준다. 여기서 *는 해 집단의 크기(pop_size)를 100으로 두고 SPX를 적용하여 실험한 결과이며, 2-opt의 경우에는 임의로 생성한 해에 2-opt를 적용한 결과로 각각 200번씩 수행한 결과를 가지고 평균한 것이다. SPX 연산자가 동일해가 선택되었을 때 2-opt를

Table 1. Simulation results of various crossover operators

	eil51(426)						rat99(1211)					
	Min	Avg.(quality)	Max	Avg. Gen.	Avg. CPU(sec)	STD	Min	Avg.(quality)	Max	Avg. Gen.	Avg. CPU(sec)	STD
MPX	432	440.2(3.3)	450	2019.2	22.7	5.33	1286	1334.6(10.2)	1382	2396.1	85.6	24.03
SCH	430	448.8(5.4)	464	1520.5	6.6	8.94	1260	1337.3(10.4)	1431	1947.4	20.6	34.94
ROX	427	437.9(2.8)	451	1734.8	5.7	5.32	1249	1289.6(6.5)	1332	2631	19.3	23.9
HX	426	439.2(3.1)	453	1743.3	8.3	6.19	1240	1300.0(7.4)	1382	2135.4	24.7	29.89
GX	431	443.9(4.2)	466	1414.4	6.7	7.8	1264	1333.7(10.1)	1430	1643.2	19.2	37.54
ER	430	443(3.9)	459	1794.3	12.6	6.72	1285	1329.9(9.8)	1394	2172.9	38.4	30.06
EER	433	448.3(5.2)	464	1506.6	9.7	7.92	1282	1339.8(10.6)	1432	1744.2	28.9	34.41
SXX	438	449.4(5.5)	470	1638.5	15.5	7.32	1280	1341.1(10.7)	1403	1863.1	41.8	29.78
CSEX	433	447.8(5.1)	471	1500.5	15.3	8.76	1286	1340.1(10.6)	1412	1930.9	45.7	29.9
DPX	428	433.8(1.8)	443	1225.2	8.9	4.18	1213	1250.9(3.3)	1287	1384.0	47.6	18.15
SPX	426	429.0(0.7)	435	1321.9	35.1	2.79	1212	1227.4(1.3)	1283	1518.7	142.9	16.14
*	426	430(0.9)	437	1546.5	23.1	3.41	1212	1228.1(1.4)	1283	1798.3	94.1	15.31
2-opt	442	468.6(10.02)	511			14.9	1291	1426.6(17.8)	1600			52.86

	kroA100 (21282)						rat195(2323)					
	Min	Avg.(quality)	Max	Avg. Gen.	Avg. CPU(sec)	STD	Min	Avg.(quality)	Max	Avg. Gen.	Avg. CPU(sec)	STD
MPX	21552	22683.0(6.6)	23779	2482.5	90.1	575.07	2551	2623.0(12.9)	2697	4089	528.5	37.11
SCH	22026	23073.9(8.4)	25262	1766.6	18.7	728.25	2577	2638.7(13.6)	2761	3474.6	106.8	50.9
ROX	21315	22528.2(5.8)	23947	2525.6	18.5	638.62	2466	2532.7(9.0)	2617	5315.5	106.7	41.13
HX	21391	22258.9(4.6)	23285	1837	21.6	493.82	2483	2593.7(11.6)	2707	3064.7	105.9	54.45
GX	21668	22755.2(6.9)	24865	1685.1	19.7	815.18	2524	2627.4(13.1)	2725	3051	104.6	49.15
ER	21785	22939.4(7.8)	24355	1815.8	32.5	652.75	2548	2625(13.0)	2727	3371.6	175	49.23
EER	21638	23115.0(8.6)	24837	1686.3	28.1	700.95	2526	2633.9(13.4)	2737	3218.6	159.2	50.48
SXX	21807	22775.7(7.0)	23954	1989.8	44.3	579.9	2522	2635.3(13.4)	2715	3618.3	409	44.53
CSEX	21721	22919.5(7.7)	24945	1998.4	47.3	713.67	2558	2642.9(13.8)	2717	4142.9	490	42.04
DPX	21292	21940.5(3.1)	23059	1292.6	48.1	410.35	2370	2400.9(3.4)	2471	1818.5	357.8	23.44
SPX	21282	21445.1(0.8)	21841	1412.5	140.1	191.58	2334	2378.2(2.4)	2466	2746.5	1120.0	36.21
*	21282	21418.3(0.6)	21841	1632.5	91.7	203.36	2345	2385.3(2.7)	2465	3401.3	701	28.03
2-opt	21870	24828.9(16.6)	27513			1202.04	2616	27470.(18.25)	2985			69.9

() : optimal solution, * : pop_size = 100, STD indicates the average of the standard deviation of fitness values.

지역 탐색 방법으로 사용함으로 2-opt 기법만을 수행한 결과가 제안하는 알고리즘의 성능 비교를 위한 기초가 될 것이다.

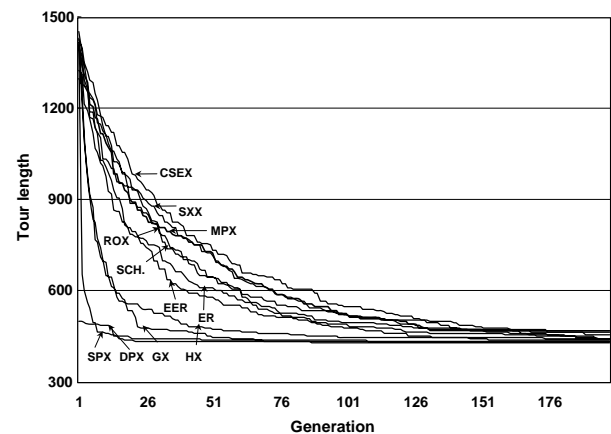
<Table 1>에서 확인할 수 있는 것처럼 2-opt 기법만을 사용한 경우의 결과는 모든 문제에서 가장 나쁜 결과를 보여주었으며, SPX와 비교해 볼 때 해의 성능(quality) 측면에서 최소 9.3%(eil51에서)에서 최대 16.4%(rat99에서)로의 차이를 보였다.

일반적으로, SPX가 다른 교차 연산자들에 비해 상대적으로 많은 계산 시간을 필요로 하였지만, 성능적인 측면에서는 모든 실험 문제들에서 다른 교차 연산자들보다 우수한 결과를 보여 줌을 확인할 수 있었다. 특히, eil51과 kroA100 문제에서, SPX를 적용한 유전 알고리즘은 최적해를 찾아내었지만 다른 교차 연산자들을 적용한 유전 알고리즘들은 최적해를 찾지 못하였다. 그리고 제안하는 알고리즘의 경우 모든 실험 문제들에서 최적해와의 차이가 3%를 넘지 않는 성능을 보여주었다. 또한 실험 결과를 분석해 본 결과 SPX가 많은 계산 시간을 필요로 한 이유는 SPX 연산자를 적용하는 동안 동일한 부모해가 선택되는 횟수의 증가 때문인 것으로 분석되었다. 따라서 계산시간을 줄이기 위해서, 해 집단의 크기를 절반으로 줄여서 실험을 수행하였으나, *로 표시된 실험 결과에서 확인할 수 있는 것처럼 성능은 거의 변하지 않음을 확인할 수 있었다.

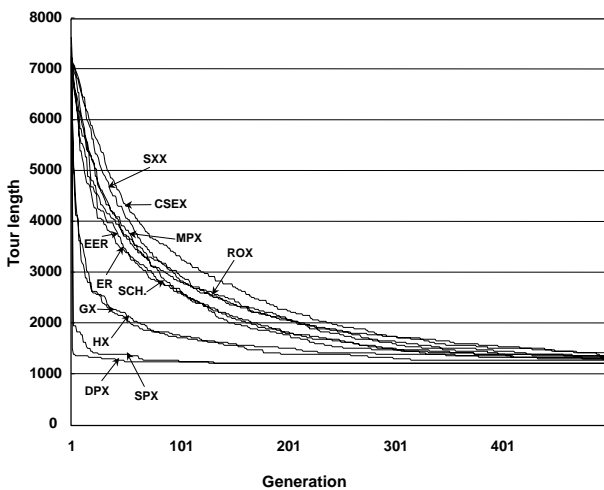
특히 본 실험을 통해서 주목할 사항은 기존에 순회 판매원 문제를 해결하기 위해 적용된 유전 알고리즘에서 가장 우수한 결과를 보여주었던 교차 연산자인 DPX 교차 연산자보다도 더 우수한 성능을 보여 준다는 것이다. 이는 매우 고무적인 결과이며, 부모해가 지니고 있는 공통의 정보를 최대한 이용한 결과일 뿐만 아니라 동일한 부모해가 선택될 경우 이를 해결하

기 위한 대안을 마련한 결과이기도 하다

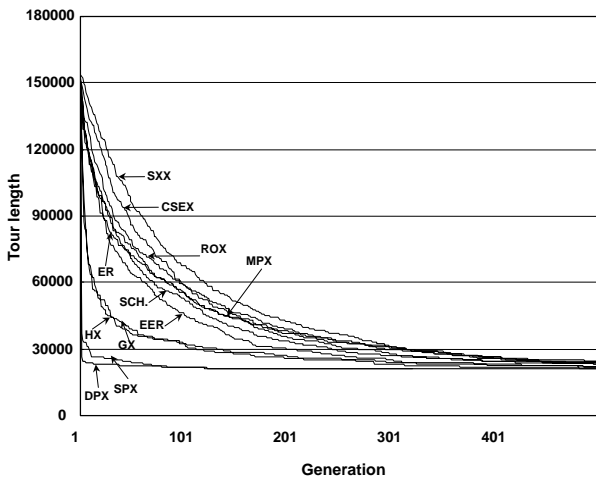
<Figure 6>는 eil51, rat99, kroA100과 rat195의 실험문제에서 30번 동안 반복 수행한 결과에서 가장 우수한 성능을 보인 결과의 수렴과정을 보여준다. <Figure 6>에서 흥미로운 사실은 부모해의 이웃하는 정보를 사용하는 교차 연산자들이 상대적으로 빠른 수렴과정을 보여주고 더 우수한 결과를 찾는다는 것이다. 이는 단순한 치환과 같은 방법을 이용하는 치환 기반 교차 연산자에 비해 부모해가 지니고 있는 정보를 이용하는 연산자들이 더 짧은 시간 내에 더 우수한 해를 찾을 가능성이 높다는 것을 보여주는 결과이다. 그리고 그림에서 확인할 수 있는 것처럼 SPX의 경우 대부분의 계산 시간이 종료 조건을 만족시키는 과정에 일어났음을 확인할 수 있다.



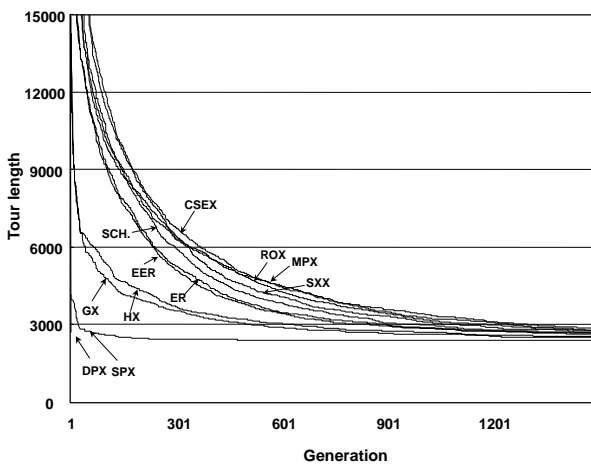
(a) Instance : eil51



(b) Instance : rat99



(c) Instance : kroA100



(d) Instance : rat195

Figure 6. Convergence processes of crossover operators

또 다른 흥미로운 사실은 유사한 특성을 이용하는 연산자들 - 즉 DPX와 SPX, HX와 GX, ER과 EER 그리고 다른 연산자

들 - 수렴과정에서 유사한 특성을 보여 준다는 것이다. 그리고 문제의 크기가 커질수록 그 차이는 더욱더 명확해진다. 비록 ER, EER과 다른 치환 기반 교차 연산자들을 정확하게 구분하는 것이 힘들어 보일지라도, 분명한 것은 ER과 EER이 항상 치환 기반 교차 연산자들 보다 더 빨리 수렴함을 확인할 수 있다.

그리고 <Figure 6>를 보면 마치 SXX와 CSEX 연산자가 치환 기반 교차 연산자처럼 수렴하는 것을 확인할 수 있는데 이는 2장에서 언급한 것처럼 이들이 자손들을 생성할 때 치환 기반 교차 연산자와 비슷한 특성을 보이기 때문이다.

4.2 SPX를 사용하는 혼합형 유전 알고리즘

두 번째 실험에서는 혼합형 유전 알고리즘(hybrid genetic algorithm : HGA)을 위해 SPX 교차 연산자와 함께 지역 탐색 방법(local search method) 그리고 최근접 이웃 휴리스틱 기법(nearest neighbor heuristic : NN)을 같이 사용하였다. 여기서 사용된 지역 탐색 방법은 수정된 Lin-Kernighan(Modified Lin-Kernighan : MLK) 방법이다(Mak and Morton, 1993). MLK는 단순히 2-opt 알고리즘의 확장이지만 2-opt에 비해 훨씬 우수한 결과를 주는 것으로 알려져 있다. 본 실험에서 사용된 혼합형 유전 알고리즘의 대략적인 개요는 다음과 같다.

초기해가 생성된 이후, NN 방법을 생성된 초기해에 적용하여 새로운 초기해를 생성하였으며, 이 이후 SPX와 변이 연산자가 적용된 이후 생성된 자손들에 MLK를 적용하여 해를 개선 시켰다. 이 이외의 다른 조건은 해 집단의 크기, 변이 확률, 반복 실험 횟수를 제외한 모두 첫 번째 실험과 동일한 조건에서 실험을 수행하였다. 그리고 해 집단의 크기는 20으로 하였으며, 변이 확률은 0.6으로 설정하였고, 반복회수는 20번으로 설정하였다. 본 실험에서 반복회수를 20번으로 한 이유는 <Table 2>에서 확인할 수 있듯이 최소값 뿐만 아니라 최대값의 성능(quality)이 2%를 넘지 않았으며, 표준편차 또한 최적해와 비교해서 0.5%도 넘지 않았기 때문에 20번의 결과로도 실험결과의 유효성을 보이기에 충분하다고 판단을 하였기 때문이다. 그리고 일반적으로 혼합형 유전 알고리즘에서는 해의 다양성을 유지하는 것이 가장 중요하므로 일반적인 변이 확률보다 상당히 높은 값을 이 실험에서 채택하였다.

<Table 2>는 HGA와 NN과 MLK를 결합한 알고리즘의 실험 결과를 보여준다. HGA의 경우 모든 실험 문제들에서 해의 성능(quality)이 최적해의 1% 이내에 있음을 확인할 수 있으며, 도시의 수가 300개 이하인 문제들에서는 모든 경우에서 최적해를 찾았다. 혼합형 유전 알고리즘의 목적이 보다 우수한 해를 찾는 것이기 때문에 이는 매우 고무적인 결과이다. 비록 본 논문에서 사용한 혼합형 유전 알고리즘이 단순히 기존의 방법들의 결합에 의존하는 단순한 알고리즘이지만 실험 결과는 아주 우수했으며 이는 좀 더 우수한 결과를 얻기 위해 제안된 혼합형 유전 알고리즘을 세부적으로 튜닝을 한다면 더 좋은 결과를 얻을 수 있을 것으로 판단된다.

Table 2. The Quality over the optimal tour distance by HGA and MLK

	SPX(Quality (%))							NN+MLK	
	Min.	Avg.	Max.	Best	Avg. Gen	Avg. CPU(sec)	STD	Avg.	STD
eil101(629)	0.0	0.0	0.0	629	45.75	1.45	0	3.46	10.19
kroA100(21282)	0.0	0.0	0.0	21282	15.65	0.5	0	2.06	220.16
kroB100(22141)	0.0	0.0	0.0	22141	73.75	2.3	0	0.99	92.2
kroA150(26524)	0.0	0.0007	0.003	26524	593.3	44.1	0.41	4.24	242.45
kroB150(26130)	0.0	0.007	0.04	26130	495.5	36.9	3.96	2.72	159.01
kroA200(29368)	0.0	0.004	0.04	29368	592	80.2	4.3	2.64	128.64
kroB200(29437)	0.0	0.05	0.66	29437	1047.0	142.1	47.07	5.12	425.59
pr226(80369)	0.0	0.0	0.0	80369	160.0	26.25	0	2.74	284.47
lin318(42029)	0.14	0.35	0.57	42091	1654.6	654.3	37.97	7.16	222.31
pr439(107217)	0.029	0.045	0.069	107249	2204.8	1859.5	14.59	6.03	1313.93
rat783(8806)	0.23	0.69	1.05	8827	3333.5	8883.7	23.45	7.58	69.27
pr1002(259045)	0.66	0.96	1.48	260762	4248.4	21898.5	594.75	9.27	1785.48

() : Optimum solution, STD indicates the average of the standard deviation of fitness values.

5. 결론

본 논문에서 저자들은 순회판매원문제를 위한 새로운 효과적인 subtour 기반 교차 연산자인 subtour 보존 교차 연산자를 제안하였고, SPX의 특징 및 성능을 실험하였다. 다양한 실험들로부터 SPX가 유전 알고리즘에 적합한 특성을 보인다는 것을 확인할 수 있었고, 기존의 subtour 기반 교차 연산자들이 지니고 있는 결점을 보완함으로써 해서 보다 우수한 결과를 얻을 수 있음을 보여주었다.

또한 다양한 실험을 통해 SPX가 기존에 TSP를 위해 개발된 다른 변이 연산자들 보다 우수한 결과를 산출해냄을 확인할 수 있었고, 또한 가장 우수한 교차 연산자로 알려진 DPX 연산자보다도 더 우수한 결과를 찾아냄을 보여주었다. 비록 제안하는 연산자가 다른 연산자들에 비해 더 많은 계산 시간을 필요로 했지만 찾아낸 성능으로 보아 이는 상쇄 할만한 수준으로 판단된다.

그리고 MLK와 NN알고리즘을 SPX와 함께 사용하는 혼합형 유전 알고리즘 또한 제안하였는데, 비록 알고리즘의 성능을 향상시키기 위한 세부적인 튜닝 과정이 이루어지지 않는 않지만 제안하는 방법이 모든 실험문제들에서 최적해에 1% 이내에 존재하는 평균적인 해를 찾아냄을 확인할 수 있었다.

앞으로는 SPX를 이용한 알고리즘의 계산 시간을 줄이는 문제와 보다 정교하게 혼합형 유전 알고리즘을 구현하여 대규모의 문제들에 적용할 필요성이 있다고 판단된다.

참고 문헌

Baraglia, R., Hidalgo, J. I., and Perego, R. (2001), A hybrid heuristic for the traveling salesman problem, *IEEE Trans. Evolutionary Computation*, 5(6), 613-622.

Boese, K. D., Kang, A. B., and Muddu, S. (1994), A new adaptive multi-start technique for combinatorial global optimizations, *Operations Research Letters*, 16(2), 101-113.

Bui, T. N. and Moon, B-R. (1994), A New Genetic Approach for the Traveling Salesman Problem, *In IEEE Conf. on Evolutionary Computation*, 7-12.

Bui, T. N. and Moon, B-R. (1994), A new genetic approach for the traveling salesman problem, *In IEEE Conf. on Evolutionary Computation*, 7-12.

Davis, L. (1985), Applying adaptive algorithms to domains. *Proc. of the Int. Joint Conf. on Artificial Intelligence*, 162-164.

Falco, I. D., Cioppa, A. D., and Tarantino, E. (2002), Mutation-based genetic algorithm : performance evolution, *Applied Soft Computing*, 1, 285-299.

Fox, B. R. and McMahon, M. B. (1991), Genetic Operators for Sequencing Problems. *In Foundations of Genetic Algorithms*, Kaufmann, 284-300.

Freisleben, B. and Merz, P. (1996), A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesmen Problems, *IEEE Int. Conf. on Evolutionary Computation*, 616-621.

Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, MA.

Goldberg, D. and Lingle, R. (1985), Alleles, loci and the traveling salesman problem, *Proc. of the 1st Int Conf. on Genetic Algorithms*, 154-159.

- Grefenstette, Gopal, R., Rosmaita, B., and Gucht, D. (1985), Genetic Algorithms for the Traveling Salesman Problem, *First Int. Conf. on GA and Their Applications*, 160-168.
- Jung, S. and Moon, B-R. (2002), Toward Minimal Restriction of Genetic Encoding and Crossovers for the 2D Euclidean TSP, *IEEE Transactions on Evolutionary Computation*, **6**(6), 557-565.
- Karp, R. M. (1972), Reducibility among combinatorial problems. *Complexity of Computer Computations, Advances in Computing Research*, Plenum Press.
- Katayama, K., Hirabayashi, H., and Narihisa, H. (1998), Performance Analysis of a New Genetic Crossover for the Traveling Salesman Problem. *IEICE TRANS. FUNDAMENTALS*, E81-A **5**, 738-750.
- Kzuber, J. and Whitley, D. (1994), Advanced Correlation Analysis of Operators for the Traveling Salesman Problem, *In parallel Problem Solving from Nature III*, 68-77.
- Maekawa, K., Mori, N., Tamaki, H., Kita, H., and Nishikawa, Y. (1996), A Genetic Solution for the Traveling Salesman Problem by Means of a Thermodynamical Selection Rule, *Proc. of IEEE Int. Conf. on Evolutionary Computation*, 529-534.
- Mak K. T. and Morton, A. J. (1993), A modified Lin-Kernighan traveling-salesman heuristic, *Operations Research Letters*, **13**, 127-132.
- Martin, O. C., Otto, S. W., and Felten, E. W. (1991), Large-step Markov chains for the traveling salesman problem, *Complex Systems*, **5**, 299-326.
- Mathias, K. and Whitley, D. (1992), Genetic operators, the fitness landscape and the traveling salesman problem. *Proc. Parallel Problem Solving from Nature II*, 219-228.
- Michalewicz, Z. (1992), Genetic Algorithms+Data Structures = Evolution Programs.
- Morikawa, K., Furuhashi, T., and Uchikawa, Y. (1992), Single Populated Genetic Algorithm and its Application to Jobshop Scheduling, *Proc. of the 1992 Int. Conf. on Industrial Electronics, Control, Instrumentation, and Automation*, **2**, 1014-1019.
- Moscato, P. (1989), On Genetic Crossover Operators for Relative Order Preservation. *Caltech Concurrent Computation Program*, C3P Report 778.
- Mühlenbein, H. (1992), Parallel Genetic Algorithm In Combinatorial Optimization, *Computer Science and Operations Research*, 441-456.
- Murata, T. and Ishibuchi, H. (1996), Positive and Negative Combination Effects of Crossover and Mutation Operators in Sequencing Problems, *IEEE Int. Conf. on Evolutionary Computation*, 170-175.
- Nguyen, H. D., Yoshihara, I., and Yasunaga, M. (2000), Modified Edge Recombination Operators of Genetic Algorithms for the Traveling Salesman Problem, *20th Annual Conf. of the IEEE on Industrial Electronics Society*, **4**, 2815-2820.
- Oliver, I., Smith, D., and Holland, J. (1989), A Study of Permutation Crossover Operators on the Traveling Salesman Problem. *Proc. of the 2nd Int. Conf. on GA*, 224-230.
- Poon, P. W. and Carter, J. N. (1995), Genetic Algorithm Crossover Operators For Ordering Application, *Computers & Operations Research*, **22**(1), 135-147.
- Qu, L. and Sun, R. (1999), A synergetic approach to genetic algorithms for solving traveling salesman problem, *International Journal of Information Sciences*, **117**, 267-283.
- Seo, D-I. and Moon, B-R. (2002), Voronoi Quantized Crossover for Traveling Salesman Problem, *Genetic and Evolutionary Computation Conference*, 544-552.
- Shang, Y. and Li, G. J. (1991), New Crossover Operators In Genetic Algorithms, *Proc. of the 1991 IEEE Int. Conf. on Tools for Artificial Intelligence*, 150-153.
- Starkweather, T., McDaniel, S., Mathias, K., Whitley, D., and Whitley, C. (1991), A Comparison of Genetic Sequencing Operators, *Proc. of the 4th Int. Conf. on Genetic Algorithm*, 69-76.
- Syswerda, G. (1991), Schedule Optimization Using Genetic Algorithms, *In L. Davis, ed., Handbook of Genetic Algorithms*, New York, 332-349.
- Tao, G. and Michalewicz, Z. (1998), Inver-over Operator for the TSP, *Proc. of the 5th Int. Conf on Parallel Problem Solving from Nature, Lecture Notes in Computer Science*, 1498, Springer-Verlag, 803-812.
- TSPLIB. <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html>.
- Whitley, D., Starkweather, T., and Fuquay, D. (1989), Scheduling problems and traveling salesman : the genetic edge recombination and operator, *Proc. 3rd Int. Conf. GA and their Applications*, 133-140.
- Yamamura, M., Ono, I., and Kobayashi, S. (1992), Character-preserving genetic algorithms for traveling salesman problem. *Journal of Japan Society for Artificial Intelligence*, **6**, 1049-1059.
- Yamamura, M., Ono, I., and Kobayashi, S. (1996), Emergent Search on Double Circle TSPs using Subtour Exchange Crossover, *Proc. of the 3rd IEEE Conf. on Evolutionary Computation*, 535-540.
- Yang, L. and Stacey, D. A. (2001), Solving the Traveling Salesman Problem Using the Enhanced Genetic Algorithm, *Lecture Notes in Computer Science*, 2056, Springer-Verlag, 307-316.
- Yang, R. (1997), Solving Large Traveling Salesman Problems with Small Populations, *Genetic Algorithms in Engineering Systems: Innovations and Applications*, 157-162.