

# Performance Analysis of Distributed Genetic Algorithms for Traveling Salesman Problem

Young Nam Kim · Min Jung Lee · Chunghun Ha<sup>†</sup>

School of Information & Computer Engineering, Hongik University

## 순회판매원문제를 위한 분산유전알고리즘 성능평가

김영남 · 이민정 · 하정훈<sup>†</sup>

홍익대학교 정보컴퓨터공학부 산업공학전공

Distributed genetic algorithm (DGA), also known as island model or coarse-grained model, is a kind of parallel genetic algorithm, in which a population is partitioned into several sub-populations and each of them evolves with its own genetic operators to maintain diversity of individuals. It is known that DGA is superior to conventional genetic algorithm with a single population in terms of solution quality and computation time. Several researches have been conducted to evaluate effects of parameters on GAs, but there is no research work yet that deals with structure of DGA. In this study, we tried to evaluate performance of various genetic algorithms (GAs) for the famous symmetric traveling salesman problems. The considered GAs include a conventional serial GA (SGA) with IGX (Improved Greedy Crossover) and several DGAs with various combinations of crossover operators such as OX (Order Crossover), DPX (Distance Preserving Crossover), GX (Greedy Crossover), and IGX. Two distinct immigration policies, conventional noncompetitive policy and newly proposed competitive policy are also considered. To compare performance of GAs clearly, a series of analysis of variance (ANOVA) is conducted for several scenarios. The experimental results and ANOVAs show that DGAs outperform SGA in terms of computation time, while the solution quality is statistically the same. The most effective crossover operators are revealed as IGX and DPX, especially IGX is outstanding to improve solution quality regardless of type of GAs. In the perspective of immigration policy, the proposed competitive policy is slightly superior to the conventional policy when the problem size is large.

**Keywords** : Traveling Salesman Problem, Distributed Genetic Algorithm, Improved Greedy Crossover, Immigration Policy

### 1. 서론

순회판매원문제(Traveling Salesman Problem, TSP)는 대표적인 조합적 최적화(combinatorial optimization) 문제로서 주어진 N개의 도시 중 한 도시로부터 출발하여 모든 도시를 한 번씩만 방문한 다음 출발도시로 돌아오는 가장

짧은 경로를 찾는 문제이다[4, 14]. TSP는 해밀턴 경로(Hamiltonian path) 문제, 충족 가능성(satisfiability) 문제, 그래프 색칠(graph coloring) 문제와 더불어 대표적인 NP-hard 문제로서 전역 최적해 보다는 근사해를 구하는 것이 선호되어 왔다[4, 16]. 그 중에서도 담금질 기법(Simulated Annealing), 타부 서치(Tabu Search), 유전알고리즘(Genetic Algorithm, GA), 개미군집최적화(Ant Colony Optimization)와 같은 메타휴리스틱 방법이 선호되고 있다[20]. TSP에 관한 수리적 모형과 메타 휴리스틱을 포함한 해법은 김기태와 전건욱의 연구[11]에 잘 정리되어 있다.

Received 20 September 2016; Finally Revised 7 December 2016;  
Accepted 8 December 2016

<sup>†</sup> Corresponding Author : chungun.ha@hongik.ac.kr

그 동안 TSP에 GA를 적용한 많은 연구가 진행되었다 [2, 19]. GA는 자연진화를 모방한 확률적 탐색기법으로서 여러 개의 지역해(local optimum)가 존재하는 문제(multimodal), 해가 일정한 규칙성을 가지고 있어 염색체로 표현이 가능한 문제(regularity), 지역해 간에 상대적인 우위 관계가 존재하는 문제 등에 매우 적합하다고 알려져 있으며, 특히 NP-hard 군에 속하는 조합적 최적화 문제에서 우수한 성과를 보여주고 있다[23]. 그러나 단일 개체군을 이용하는 GA는 전체 개체군(population)을 대상으로 유전 연산을 시행하므로 개체군의 크기가 작을 경우 초개체(super individual)가 발생하면 지역해에 빠지기 쉬운 단점이 있다. 반면, 해의 다양성을 유지하기 위하여 개체군의 크기를 크게 하면 계산시간이 오래 걸리는 상충관계가 존재한다[12]. 이를 보완할 수 있는 대표적인 방법이 개체군을 독립적이고 부분적으로 이용하는 병렬유전알고리즘이다.

병렬유전알고리즘(parallel GA)은 원래 병렬 컴퓨팅을 이용한 계산시간 단축을 목적으로 제안되었으나 계산시간뿐만 아니라 해의 품질도 우수함이 입증되면서 주목을 받았다[5, 12]. 병렬유전알고리즘은 크게 이웃모델과 섬모델로 분류할 수 있다[6, 3, 10, 12]. 이웃모델은 cellular GA, fine-grained model로도 불리는 데, 하나의 개체군 내에서 각 개체가 독립적으로 이웃 개체와 선택과 유전 연산을 병렬로 수행하며 진화하는 방식을 취한다. 지역적 탐색을 위주로 하므로 초개체가 나타나더라도 전체 개체군으로 급속히 확산되지 않아 다양성을 유지할 수 있다. 반면, multi-deme 모델, coarse-grained 모델, 분산유전알고리즘으로도 불리는 섬모델은 하나의 개체군을 다수의 부분개체군으로 분리하고 각 부분개체군을 독립적으로 진화시키며 지역적 탐색을 강화하고 주기적으로 개체를 다른 부분개체군으로 이전시키며 다양성을 보완하는 방법이다.

기존의 TSP에 관련된 GA 연구는 새로운 유전연산자의 개발, 새로운 휴리스틱의 개발, 2-OPT나 LK(Lin-Kernighan) 같은 휴리스틱과의 결합, 분산유전알고리즘의 적용 등을 통하여 해의 품질을 개선하는 연구에 집중되어 왔다[4, 20]. 실제로 유전알고리즘의 해의 품질과 계산시간은 위에 언급한 구조적인 측면 외에도 개체군의 크기, 세대 수, 사용된 선택, 교배, 변이의 종류와 적용확률 등 적용 파라미터의 영향을 크게 받는다[8]. 따라서, GA에 사용되는 각종 파라미터의 영향에 대한 비교 분석[8], 교차연산자의 영향에 대한 비교 분석[13, 22], 선택연산의 영향에 대한 비교 분석[17] 등의 연구가 수행되었다.

본 연구에서는 TSP에 분산유전알고리즘을 적용시킴에 있어서 여러 요인 중에서 가장 영향이 크다고 알려진 분산유전알고리즘의 적용여부, 교차연산자의 종류, 그리고 이주정책의 종류에 따른 해의 품질과 계산시간의 차이에 대

한 분석을 하였다. 이를 위하여 섬모델 중 부분개체군이 독립적인 교차연산자를 사용하는 CoPDEB(Co-operating Populations with Different Evolution Behaviors) 모델[1, 18]을 적용하였다. 또한 교차연산자로서는 OX(Order Crossover), DPX(Distance Preserving Crossover), GX(Greedy Crossover), 그리고 GX를 개선한 IGX(Improved Greedy Crossover)[9]를 고려하였으며, 이주정책으로는 기존의 무경쟁편입[18]과 대응되는 경쟁편입 방식을 새로 제안하고 이를 적용하였다. 본 논문은 기존 논문과 다음과 같은 차별점을 갖는다. 우선 직렬 GA 뿐만 아니라 분산유전알고리즘도 고려하여 분석을 수행하였으며, 파라미터의 변화에 따른 비교가 아닌 구조적인 측면에서 교차연산자와 이주정책을 동시에 고려하였다. 이때, 새로운 이주정책을 추가로 설계하여 반영하였으며, 200개 도시 미만의 작은 문제뿐만 아니라 1,173개 도시를 가진 중형문제도 검증에 포함하였다. 또한 해의 품질과 계산시간의 비교분석을 위하여 평균값의 단순 비교가 아닌 분산분석(ANOVA)을 통하여 통계적 검증을 실시하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 연구에서 적용하는 분산유전알고리즘과 교차연산자에 대하여 상세히 기술한다. 제 3장에서는 기존의 이주정책과 본 논문에서 새롭게 제안하는 이주정책을 비교하여 설명한다. 이어서 제 4장에서는 본 연구의 비교실험에 대한 실험환경을 정리하고 제 5장에서는 실험결과를 제시하고 이에 대한 논의를 한다. 마지막으로 제 6장에서는 본 논문의 결론과 향후 연구방향에 대하여 논한다.

## 2. 기존 연구

### 2.1 분산유전알고리즘

유전알고리즘은 1975년에 Holland가 저서 “Adaptation in Natural and Artificial Systems”에 처음 소개한 자연도태의 원리를 기초로 개발한 최적화 방법이다. 한 세대(generation)는 개체들의 집합, 즉 개체군으로 구성되어 있으며, 각각의 개체(individual)는 복수개의 유전자(gene)의 집합으로 구성된 유전자형(genotype)인 염색체(chromosome)를 가지고 있다. 개체군 중에서 환경에 대한 적합도(fitness)가 높은 표현형(phenotype; 유전자형에 의해 결정된 외부로 나타나는 개체의 형질)을 가진 개체들은 자연선택을 통해 높은 확률로 살아남아 교차(crossover)나 변이(mutation) 등의 과정을 통하여 자식을 생성함으로써 다음 세대의 개체군을 형성하고 적합도가 낮은 개체는 자연적인 도태를 하게 된다. 유전알고리즘은 이러한 진화 과정을 여러 세대에 걸쳐 진행하여 적합도가 높은 개체를 근사해로 결정한다.

분산유전알고리즘(distributed GA : 이하 DGA)은 일반적으로 다음의 절차에 의하여 수행된다[12, 18].

- Step 1 : 초기 개체군을 생성하고  $k$ 개의 부분모집단으로 분리
- Step 2 : 종료 조건을 만족하면 종료
- Step 3 : 이주 조건을 만족하면 다음을 실행
  - Step 3-1: 각 부분개체군으로부터 이주시킬 개체를 선택
  - Step 3-2: 유전연산을 실행하여 자식을 생성
  - Step 3-3: 자식을 이웃 부분개체군으로 편입(이때 부분개체군의 크기를 유지하기 위하여 기존의 부분개체군에서 적합도가 낮은 개체를 교체될 자식의 수와 동일하게 선택하여 교체)
- Step 4 : 각 부분개체군에 대하여 선택, 교차, 변이, 편입을 순차적으로 수행하고 Step 2로 이동

분산유전알고리즘의 기존의 유전알고리즘과의 차이점은 Step 1의 개체군 분리과정과 Step 3의 개체 이주에 있다. 분산유전알고리즘은 부분개체군을 유지함으로써 초기 개체의 무분별한 확산을 방지하여 해가 지역해로 조기 수렴하는 것을 방지하고 알고리즘이 다양한 해공간을 탐색할 수 있도록 한다.

### 2.2 교차연산자

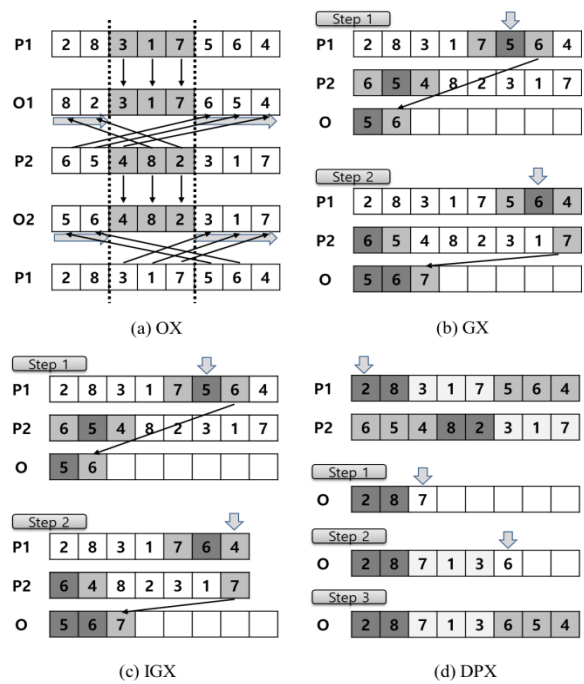
GA의 기본 연산으로는 선택, 교차, 변이가 있다. 선택은 자식을 생성하기 위하여 교차와 변이를 수행하기 전에 부모 개체를 선택하는 과정이다. 선택 연산은 룰렛휠 선택(Roulette Wheel Selection), 순위기반 선택(Rank-based Selection), 토너먼트 선택(Tournament Selection) 등이 주로 사용되고 있다[17]. 변이는 해가 지역해에 고착되는 것을 막기 위해 수행하는 연산이며, 중심 역위 변이(Centre Inverse Mutation)와 역순 서열 변이(Reverse Sequence Mutation) 등이 있다.

해의 품질에 가장 많은 영향을 미치는 교차연산자는 선택과정을 통하여 선별된 부모 개체로부터 자식을 생성하는 연산이다. 그 동안 많은 교차연산자가 개발되었지만 TSP에서는 Cycle Crossover(CX), Partially-Mapped Crossover(PMX), Order Crossover(OX), Greedy Crossover(GX) 등이 주로 사용되어 왔다[13, 22, 24]. 본 연구는 기존의 OX와 GX 외에 DPX와 IGX를 적용하였다. 각 교차연산자에 대한 개략적인 동작원리를 <Table 1>의 거리행렬과 <Figure 1>의 개념도를 통하여 설명하면 다음과 같다.

OX에서는 우선 부모 개체의 염색체(P1, P2)에서 임의의 두 교차 지점(<Figure 1(a)> 점선)을 선정한다. 자식의 생성과정은 우선 P1로부터 두 교차 지점 사이의 유전자

<Table 1> Distance Matrix for Example

city	1	2	3	4	5	6	7	8
1	0	12	14	14	20	15	10	11
2	12	0	17	15	18	13	21	6
3	14	17	0	13	9	11	17	18
4	14	15	13	0	8	9	13	13
5	20	18	9	8	0	7	12	17
6	15	13	11	9	7	0	22	20
7	10	21	17	13	12	22	0	7
8	11	6	18	13	17	20	7	0



<Figure 1> Various Crossover Operations

(gene) 집합(3-1-7)을 그대로 복사하고, 나머지 빈 공간은 P2로부터 P1에서 복사된 유전자(3, 1, 7)를 제외하고 두 번째 교차선 이후로 순차적으로 복사한다. 다른 자식은 이 과정을 P1과 P2를 바꾸어 생성한다.

GX는 순차적으로 유전자를 추가하는 방식으로 자식의 염색체를 생성한다. 우선, 부모 개체의 염색체에서 하나의 유전자를 무작위로 선택한다(<Figure 1(b)> Step 1 유전자 5). 다음에 각각의 부모 염색체에서 선택된 유전자(5)를 찾아 인접한 유전자들(4, 6, 7) 중 가장 거리가 가까운 유전자(6)를 다음 유전자로 선택한다. 이때, 만약 인접한 유전자들이 이미 자식의 염색체에 포함되어 있다면 배정되지 않은 잔여 유전자 중 무작위로 선택하거나 가장 거리가 인접한 유전자를 선택한다. 이 과정을 자식 염색체의 모든 자리가 채워질 때까지 반복한다.

IGX는 선택된 유전자에 인접한 유전자가 모두 이미 자식

의 염색체에 배정이 되어 있다면 모든 잔여 유전자를 고려하여 선택하여야 하므로 매우 비효율적이다. IGX는 GX의 이러한 단점을 개선하기 위하여 제안되었다[9]. IGX는 연결된 리스트 형태의 자료구조로 구성된다. GX의 과정 중에서 선택된 유전자(5)가 자식의 염색체로 복사되면 해당 유전자를 부모 염색체의 연결된 리스트로부터 제거한다 (<Figure 1(c)> Step 2). 이는 항상 배치 가능한 인접 유전자가 존재하도록 유지해 주므로 GX의 모든 잔여 유전자를 검색해야 하는 비효율성을 개선할 수 있는 장점이 있다.

마지막으로 DPX는 부모 염색체의 좋은 스키마를 그대로 상속하는 것을 중점으로 하는 교차연산이다[7]. 부모 염색체들에서 공통된 유전자 집합을 서브패스(<Figure 1(d)>에서 2-8, 3-1-7, 5-6-4)라고 하고, 이를 목록으로 관리한다. 이때, 서브패스는 유전자의 구성이 동일하면 동일한 서브패스로 간주한다. 자식의 염색체는 서브패스 단위로 추가하며 생성하는 데, 생성 규칙은 다음과 같다. 우선, 임의로 첫 유전자(2)를 선택한다. 해당 유전자(2)가 포함된 서브패스(2-8) 선택해서 자식염색체에 복사한다. 해의 다양성을 확보하기 위하여 선택된 서브패스(2-8)와 인접한 유전자(3, 4)를 제외하고 잔여 서브패스의 끝 유전자들(3, 5, 7) 중 가장 가까운 거리에 있는 유전자(7)를 선택한다. 선택된 유전자(7)을 포함하는 서브패스를 자식염색체에 추가하고 이 과정을 염색체가 완성될 때까지 반복한다.

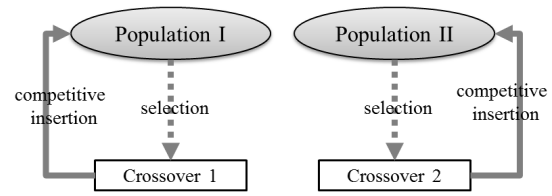
### 3. 이주 정책

#### 3.1 기존 이주정책 : 무경쟁편입

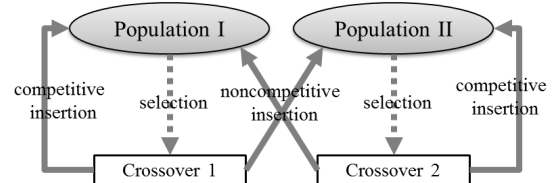
DGA에서 이주방식은 해의 품질과 계산시간에 많은 영향을 미친다. 이주방식은 다양하게 설계될 수 있으나, 본 연구에서는 박유석의 연구[18]를 기본으로 하여 연구를 수행하였다. 기존 연구에서의 이주 방식은 전형적인 DGA의 과정을 따르고 있는데, 개체군은 다수의 부분개체군으로 분리하고 각 부분개체군은 일정세대 동안 독립적으로 진화하다가 이주주기가 도래하면 이주율에 따라 생성된 개체를 상대방의 개체군에 편입한다. 명확한 설명을 위하여 비이주주기와 이주주기에 과정을 그림으로 요약하면 <Figure 2>와 같다.

비이주주기(<Figure 2(a)>)에서는 각 부분개체군은 선택과정을 통하여 선별된 부모개체로부터 개별적으로 정해진 교차연산자를 적용하여 자식을 생성한 후 이를 경쟁편입을 통하여 각 부분개체군으로 환원한다. 이주주기(<Figure 2(b)>)가 도래하면 선택된 부모 개체로부터 정해진 교차연산자를 적용하여 자식을 생성한 후 자신의 개체군으로 경쟁 편입을 하고 인접한 부분개체군으로는 이주율에 따라 무경쟁 편입을 한다. 예를 들면, 이주주기가

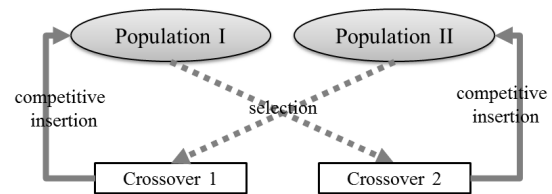
15세대이고 이주율이 5%라면, 매 15세대마다 각 개체군에서 생성되는 자식 개체 중 5%는 인접한 부분개체군으로 편입되며, 이때 적합도가 낮은 개체는 무조건 대체된다. 이러한 이주정책은 새로운 유전자의 개체를 무조건 편입시킴으로 해의 다양성을 증가시키는 데 유리한 것처럼 보인다. 하지만, 이주하는 자식개체의 해의 품질을 보장할 수 없으므로 다음 세대의 진화과정에서 이주된 개체가 탈락될 가능성도 높다. 따라서 이러한 방식은 해의 다양성 증가에 대한 기여도가 불확실하다. 이주율을 높이면 이러한 단점을 보완할 수 있지만, 이 경우에는 부분개체군이 보유하고 있는 우수한 해도 대체될 가능성이 높아지므로 적절한 이주율을 결정해 주어야 한다.



(a) immigration policy at non-immigration period



(b) immigration policy of conventional GAs at immigration period



(c) immigration policy of the proposed GA at immigration period

<Figure 2> Immigration Policy

#### 3.2 제안 이주정책 : 경쟁편입

기존의 무경쟁편입 정책의 가장 큰 단점은 부분개체군의 일부를 무조건 다른 부분개체군으로 편입시키므로 해의 다양성이 증가한다는 장점이 분명히 존재한다. 그러나 부분개체군간 해의 품질의 차이가 클 경우 품질이 낮은 부분개체군으로부터 이주한 개체는 품질이 우수한 부분개체군의 품질을 전체적으로 저하시킬 수 있고, 이는 수렴속도를 늦추게 된다. 따라서 본 논문에서는 무경쟁편입을 대체할 수 있는 경쟁편입 정책을 제안한다. 새로운 이주정책은 적절한 수준을 결정하기 어려운 이주율을 따로

설정하지 않고, 이주주기가 되면 상대방의 개체군에서 부모 개체를 선택하고 정해진 교차연산을 수행하여 자식 개체를 생성한 뒤 자신의 부분개체군에 경쟁 편입하는 구조(<Figure 2(c)>)를 갖는다. 즉, 상대 부분개체군으로부터 생성된 자식 개체가 자신의 적합도 최하위 개체보다 좋은 해를 가질 경우에만 대체가 이루어진다. 이러한 이주정책은 인접한 부분개체군의 개체가 부모로 선택되는 것을 보장함으로써 해의 다양성을 확대할 수 있고 동시에 경쟁 개체를 통하여 해의 품질도 향상시킬 수 있는 장점이 있다.

## 4. 실험 구성

### 4.1 유전알고리즘 구성

본 연구에서는 GA를 위한 염색체 구조로서 일반적으로 가장 많이 사용하는 해밀토니안 경로에 대한 순열표현을 적용하였다. 적합도는 주어진 해에 대하여 해밀토니안 경로 길이의 역수를 사용하였다. 초기 개체군을 형성하기 위한 개체 생성방법은 2-OPT나 최단이웃검색(nearest neighborhood search)과 같은 휴리스틱을 적용하는 방법도 있지만 본 논문에서는 초기해의 영향을 상쇄하기 위하여 무작위로 순열을 생성하는 방식을 적용하였다.

개체군에서 부모 개체를 선택하는 방법은 이진 토너먼트 방식(Binary Tournament)을 공통적으로 적용하였다. 이것은 룰렛휠 선택이나 순위 기반 선택에 비하여 수렴성이 크기 때문에 해의 품질을 떨어뜨리는 경향이 있지만 빠른 실행속도를 가지고 있고 실험결과 해의 품질에 있어서도 큰 격차를 보이지 않아 적용하였다. 교차연산은 제 3.1절에서 설명한 바와 같이 OX, GX, IGX, DPX 4가지 교차연산자를 고려하였다. 변이 연산자로는 역순서열 변이를 공통적으로 적용하였다. 또한 개체군 내의 우수한 개체의 염색체가 변이 연산을 통해 훼손되지 않게 하기 위하여 상위 20%의 개체는 변이 연산을 거치지 않고 다음 세대로 전달되는 엘리티즘을 적용하였다. 이주정책은 제 3.2절에서 설명한 바와 같이 기존의 무경쟁

편입방식(noncompetitive insertion)과 본 연구에서 새롭게 제안한 경쟁편입(competitive insertion)방식을 고려하였다.

<Table 2>에는 본 논문에서 실험을 위해 사용한 7가지 GA의 구조적 차이를 요약하였다. GA의 이름은 각 GA의 특징을 대변하고 있는데, 첫 번째 항은 Serial(S)과 Distributed(D) GA를 의미하고, 두 번째 항은 사용된 교차연산자의 조합을 나타낸다. 마지막 항은 DGA에서 무경쟁편입(N)과 경쟁편입(C) 이주정책을 나타낸다.

### 4.2 파라미터 설정

GA에 적용하는 파라미터는 해의 품질과 계산시간에 영향을 미친다. 공정한 평가를 위하여 모든 GA에 대하여 동일한 파라미터는 동일한 값으로 설정하였다. 교차율(mutation rate)과 변이율(mutuation rate)은 각각 0.6과 0.7을 적용하였다. 개체군의 크기는 40으로 설정하였고, 부분개체군의 개수는 2로 고정하였다. 따라서 SGA의 개체군은 40개의 개체로 구성되고, DGA는 각각 개체가 20개인 2개의 부분개체군으로 구성된다. DGA의 경우 이주주기는 기존 연구[18]를 참조하여 개체군크기의 0.15배로 설정하였고, 이주율(immigration rate)도 개체군 크기의 0.05배로 설정하였다. 알고리즘의 종료 조건은 진화 세대가 각 문제의 도시 수의 10배가 되면 종료하는 것으로 설정했다.

## 5. 실험 결과 및 분석

### 5.1 실험 결과

GA의 평가는 TSPLIB[21]에 있는 대칭 TSP(symmetric TSP) 문제 중에서 도시의 수에 따라 eli51, kroA100, kroA200, pcb442, pcb1173 5개의 문제를 임의로 선택하여 실시하였다. 모든 알고리즘은 python으로 작성하였고, PyPy[15]를 적용하여 실행속도를 개선하였다. 실험에는 i7-5700HQ 2.7GHz 컴퓨터를 사용하였다.

각 문제에 대해서 <Table 2>에 열거한 7가지 알고리즘을 각각 10회 실행하였다. 실험결과에서 얻어진 각 문제별, 알고리즘별 최적해(Best L), 평균해(Ave. L), 그리고 평균계산시간(Ave. CT)을 <Table 3>에 정리하였다. <Table 3>에서 각 문제의 알려진 최적해는 각각 문제의 이름아래 표기하였으며, 각 문제 별 항목에 대한 가장 우수한 결과는 회색으로 음영처리 하였다. 표에서 확인할 수 있듯이 최적해 관점에서는 IGX와 DPX 그리고 제안한 경쟁편입을 적용한 DGA(D/ID/C)가 가장 우수한 성과를 보이고 있다. 반면, 평균해 관점에서는 특별히 두각을 나타내는 알고리즘이 없었다. 반면, 계산시간에서는 분산유전알고리즘 중 제안한 경쟁편입을 도입한 DGA가 우수함을 확인할 수 있었다.

<Table 2> Various GAs for Experiments

Name	Type of GA	Crossover	Immigration Policy
S/I	Serial	IGX	N/A
D/GO/N	Distributed	GX+OX	noncompetitive
D/IO/N	Distributed	IGX+OX	noncompetitive
D/ID/N	Distributed	IGX+DPX	noncompetitive
D/GO/C	Distributed	GX+OX	competitive
D/IO/C	Distributed	IGX+OX	competitive
D/ID/C	Distributed	IGX+DPX	competitive

<Table 3> Experimental Results for 5 Selected Problems

Problem (optimal L)	GA Type	Best L	Ave. L	Ave. CT [sec]
eil51 (426)	S/I	426	428.2	0.89
	D/GO/N	429	434.5	0.47
	D/IO/N	426	428.7	0.42
	D/ID/N	426	428.7	0.33
	D/GO/C	428	434.9	0.71
	D/IO/C	426	431.8	0.29
D/ID/C	426	429.8	0.55	
kroA100 (21282)	S/I	21282	21396.9	4.93
	D/GO/N	21515	22035.4	3.30
	D/IO/N	21282	21370.8	2.94
	D/ID/N	21282	21383.4	2.32
	D/GO/C	21431	21968	2.86
	D/IO/C	21355	21402	2.26
D/ID/C	21282	21401.9	2.49	
kroA200 (29368)	S/I	29551	29676.2	17.58
	D/GO/N	30639	31887.7	16.02
	D/IO/N	29549	29866	10.27
	D/ID/N	29533	29725.7	10.94
	D/GO/C	30490	31324.8	15.32
	D/IO/C	29477	29739.9	12.53
D/ID/C	29445	29600.2	10.04	
pcb442 (50778)	S/I	51459	51976.1	92.31
	D/GO/N	55578	56336.5	75.88
	D/IO/N	51400	51865	57.85
	D/ID/N	51386	51665.7	59.41
	D/GO/C	53918	55124.2	76.71
	D/IO/C	51512	52056.8	76.35
D/ID/C	51215	51768.5	48.52	
pcb1173 (56892)	S/I	58358	59264	1022.61
	D/GO/N	67063	69053.3	595.41
	D/IO/N	58785	59498.7	472.03
	D/ID/N	59121	59358.7	483.84
	D/GO/C	65552	67524.9	609.22
	D/IO/C	58890	59270.4	461.37
D/ID/C	58847	59354.7	427.49	

알고리즘 성능의 정확한 판별을 위하여 해의 품질과 계산시간의 품질에 대한 박스도표를 <Figure 3>에 나타내었다. 해의 품질(Solution Quality)과 계산시간의 품질(CT Quality)은 각각 다음의 식에 의하여 계산하였다.

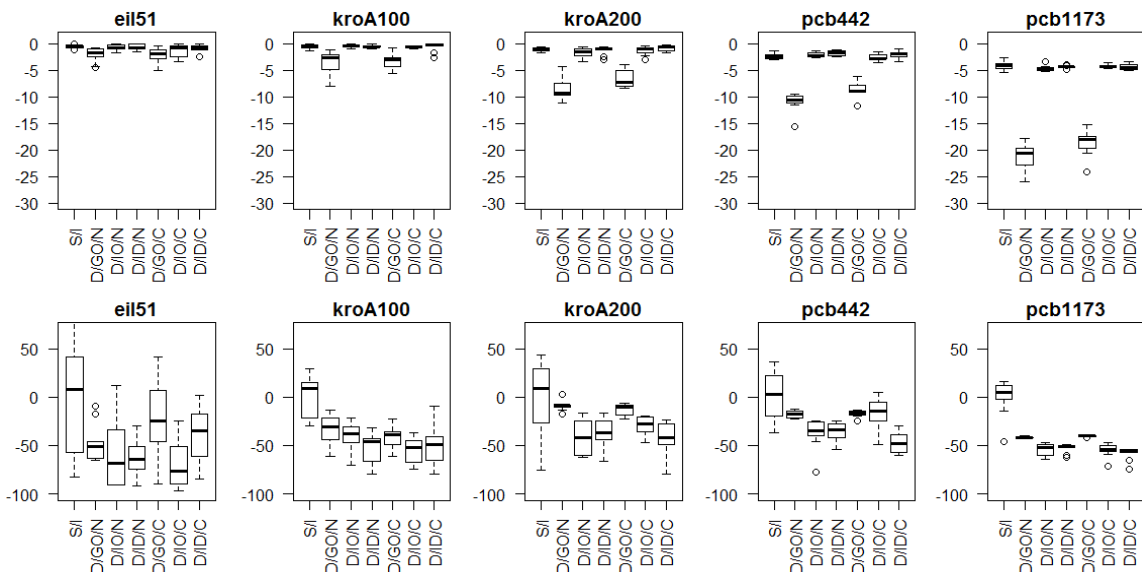
$$\text{Solution Quality} = \frac{\text{optimal L} - \text{tour L}}{\text{optimal L}}$$

$$\text{CT Quality} = \frac{\text{mean(CT of S/I)} - \text{CT of GA}}{\text{mean(CT of S/I)}}$$

여기서 optimal L은 알려진 최적 투어의 길이이고, tour L은 각 GA의 결과로 나온 투어의 길이이다. CT는 계산시간을 의미하고 S/I의 평균계산 시간을 기준으로 하여 각 GA 알고리즘의 계산시간을 비교한다. <Figure 3>에서 명확히 확인할 수 있는 것은 해의 품질 측면에서는 SGA나 DGA와 관계없이 IGX를 사용한 경우 매우 우수하고 그 차이가 크지 않다는 점이다. 또한, 계산시간의 품질측면에서는 SGA가 DGA에 비하여 계산시간이 길고 DGA 간에는 적용한 교차연산자에 따라 차이가 보인다는 점이다.

### 5.2 분산분석

제 5.1절을 통하여 대체적인 경향은 분석하였지만 분산 GA의 적용, 교차유전자의 종류, 이주정책에 따른 효과를 명확히 검증하기 위하여 분산분석(Analysis of Variance : one-way ANOVA)을 실시하였고 필요한 경우 Tukey의 HSD (honestly significant difference) test를 통하여 집단 간 평균 다중비교(multiple comparison)를 실시하여 차이에 대한 검증



<Figure 3> Boxplots of Solution Quality(1st Row) and CT Quality(2nd Row) for 5 Selected Problems

을 실시하였다. 분산분석을 실시하기 전 등분산 검정(Levene Test)과 실시한 후 잔차에 대하여 정규성검정(Shapiro Test)도 실시하였다. 일부 경우에 등분산과 정규성에 대한 귀무가설을 기각하여 만족하지 못하는 경우도 나왔으나 본 실험의 경우 표본의 개수가 동일한 균형표본(balanced sample)이므로 이들에 대한 영향은 크지 않아 분산분석 결과를 그대로 사용하였다.

우선 DGA의 효과를 확인하기 위하여 IGX를 적용한 SGA와 IGA를 적용한 DGA(D/IO/N, D/ID/N, D/IO/C, D/ID/C)에 대하여 분산분석을 실시하였고 결과 중 p-value에 대하여 <Table 4>에 정리하였다. <Table 4>에서 TL(tour length)은 투어의 길이를 의미하고 CT(computation time)는 계산시간을 의미한다. 결과에서 볼 수 있듯이 투어의 길이는 알고리즘에 따른 차이가 있다는 귀무가설을 기각하지 못하므로 IGX를 적용한 경우 SGA와 DGA의 해의 품질이 다르다고 할 수 없다. 반면, 계산시간에서는 DGA가 SGA보다 우수하다고 통계적인 결론을 내릴 수 있다. 결론적으로 동일수준의 해를 도출하는 데 DGA를 적용하는 것이 계산시간 관점에서 유리함을 확인할 수 있었다.

<Table 4> SGA vs. DGAs with IGX

Prob.	TL	CT
eli51	0.1546	0.00014
kroA100	0.8341	< 0.0001
kroA200	0.4837	< 0.0001
pcb442	0.2079	< 0.0001
pcb1173	0.3271	< 0.0001

\*Gray cell : statistically significant(p-value < 0.05) and DGAs are better than SGA.

다음으로는 DGA에서 동일한 이주정책 하에서 개체군에서 사용하는 교차연산자의 종류에 따른 차이가 있는지 분산분석을 실시하였다. 분산분석결과 GX와 OX에 의한 차이는 유의수준 0.05에서 통계적으로 유의하지 않았으며, DPX와 IGX는 유의하였다. 분산분석에 따른 p-value는 <Table 5>에 정리하였다. 이주정책(Policy)과 관계없이 몇몇 경우만 제외하고는 DPX와 IGX의 사용이 해의 품질이나 계산시간에 모두 긍정적임을 확인할 수 있었다. 단, 계산시간보다는 해의 품질에 대한 영향이 더욱 크다.

마지막으로 DGA에서 동일한 교차연산자 조합에서 이주정책에 따른 차이가 있는지 검정하였다. 분산분석 결과는 <Table 6>에 정리하였으며, 이주정책에 따른 차이는 크게 존재하지 않았다. 여기서 주목할 점은 문제의 크기가 큰 경우 경쟁편입 정책이 비경쟁 편입보다는 해의 품질 또는 계산시간에서 다소 우위를 보인다는 점이다.

<Table 5> GX vs. OX vs. IGX vs. DPX Under the Same Immigration Policy

Policy	Mea.	Prob.	DPX	IGX
NC	TL	eli51	0.0563	0.0018
		kroA100	0.0052	< 0.0001
		kroA200	< 0.0001	< 0.0001
		pcb442	< 0.0001	< 0.0001
		pcb1173	< 0.0001	< 0.0001
	CT	eli51	0.2534	0.6475
		kroA100	0.0091	0.2911
		kroA200	0.0308	< 0.0001
		pcb442	0.0669	0.0004
		pcb1173	0.0079	< 0.0001
C	TL	eli51	0.0571	0.1444
		kroA100	0.0011	< 0.0001
		kroA200	< 0.0001	< 0.0001
		pcb442	< 0.0001	< 0.0001
		pcb1173	< 0.0001	< 0.0001
	CT	eli51	0.6657	0.0023
		kroA100	0.8198	0.0846
		kroA200	< 0.0001	0.0044
		pcb442	< 0.0001	0.9423
		pcb1173	< 0.0001	< 0.0001

\*Gray cell : statistically significant(p-value < 0.05) and use of DPX (IGX) is better than no-use.

<Table 6> Noncompetitive vs. Competitive

Crossover	Prob.	TL	CT
GX+OX	eli51	0.8783	0.0547
	kroA100	0.7	0.168
	kroA200	0.0299	0.1208
	pcb442	0.0047	0.5486
	pcb1173	0.0298	0.001
IGX+OX	eli51	0.0791	0.3763
	kroA100	0.1696	0.0257
	kroA200	0.3155	0.0559
	pcb442	0.1568	0.0151
IGX+DPX	eli51	0.328	0.0366
	kroA100	0.7703	0.6635
	kroA200	0.1446	0.4962
	pcb442	0.4283	0.0201
	pcb1173	0.973	0.0462

\*Gray cell : statistically significant(p-value < 0.05) and competitive is better than noncompetitive

\*Black cell : statistically significant(p-value < 0.05) and noncompetitive is better than competitive

## 6. 결 론

TSP에 대하여 단일 유전알고리즘과 다양한 분산유전 알고리즘을 구성하고 이들에 대한 해의 품질과 계산시간에 대하여 분산분석을 통하여 통계적 검증을 실시하였다. DGA의 경우, SGA보다 해의 우수성을 특별히 보장할 수는 없지만 계산시간이 단축되는 특징이 있었다. 해의 품질은 사용하는 교차연산자의 영향을 많이 받으며, IGX와 DPX가 해의 품질을 개선하는 데 기여도가 큼을 확인할 수 있었다. 이주정책은 해의 품질이나 계산시간에 뚜렷한 차이가 있다고 하기는 어려우나, 본 논문에서 제안한 경쟁편입은 문제의 크기가 큰 경우에 해의 품질이나 계산시간에서 기존의 비경쟁 편입에 비하여 다소 우수한 성능을 보이고 있다. 본 연구의 한계점으로는 우수한 성능을 보이고 있는 2-OPT나 LK와 같은 휴리스틱을 결합한 memetic GA를 고려하지 못하였고, 다양한 이주정책을 설정하여 비교하지 못한 점이 아쉬움으로 남는다. 이는 향후 연구를 통하여 보완할 계획이다.

## Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(2015R1D1A1A01060391).

## References

- [1] Adamidis, P. and Petridis, V., Co-operating populations with different evolution behaviors, in *Proceedings of IEEE International Conference on Evolutionary Computation*, 1996, pp. 188-191.
- [2] Ahmed, Z.H., Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, *International Journal of Biometrics and Bioinformatics*, 2010, Vol. 3, No. 6, pp. 96-105.
- [3] Alba, E. and Troya, J.M., A survey of parallel distributed genetic algorithms, *Complexity*, 1999, Vol. 4, No. 4, pp. 31-52.
- [4] Applegate, D.L., Bixby, R.E., Chvátal, V., and Cook, W.J., *The traveling salesman problem : a computational study*, 2006.
- [5] Belding, T.C., The distributed genetic algorithm revisited, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995, pp. 114-121.
- [6] Cantu-Paz, E., A survey of parallel genetic algorithms, *Calculateurs Paralleles*, 1998, Vol. 10, No. 2, pp. 141-171.
- [7] Freisleben, B. and Merz, P., New genetic local search operators for the traveling salesman problem, in *Volume 1141 of the series Lecture Notes in Computer Science*, 1996, pp. 890-899.
- [8] Grefenstette, J., Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man, and Cybernetics*, 1986, Vol. 16, No. 1, pp. 122-128.
- [9] Ismkhan, H. and Zamanifar, K., Developing improved greedy crossover to solve symmetric traveling salesman problem, *International Journal of Computer Science Issues*, 2012, Vol. 9, No. 4, pp. 121-126.
- [10] Kim, K.-T. and Jeon, G.-W., A path planning to maximize survivability for unmanned aerial vehicle by using A×PS-PGA, *Journal of the Society of Korea Industrial and Systems Engineering*, 2011, Vol. 34, No. 3, pp. 16-23.
- [11] Kim, K.-T. and Jeon, G.-W., Hybrid parallel genetic algorithm for traveling salesman problem, *Journal of the Korea Safety Management and Science*, 2011, Vol. 13, No. 3, pp. 107-114.
- [12] Kim, Y.K., *Evolutionary algorithms*, Kwangjoo : Chonnam University, 2010.
- [13] Kumar, N. and Karambir, R.K., A comparative analysis of PMX, CX and OX crossover operators for solving travelling salesman problem, *International Journal of Latest Research in Science and Technology*, 2012, Vol. 1, No. 2, pp. 98-101.
- [14] Laporte, G., The traveling salesman problem : An overview of exact and approximate algorithms, *European Journal of Operational Research*, 1992, Vol. 59, No. 2, pp. 231-247.
- [15] Lutz, M., *Learning python*, O'Reilly Media, 2013.
- [16] Merz, P. and Freisleben, B., Memetic algorithms for the traveling salesman problem, *Complex Systems*, 1997, Vol. 13, pp. 297-345.
- [17] Noraini, M. and Geraghty, J., Genetic algorithm performance with different selection strategies in solving TSP, in *International Conference of Computational Intelligence and Intelligent Systems*, 2011.
- [18] Park, Y.S., Distributed genetic algorithms for the TSP, *Journal of the Korea Safety Management and Science*, 2001, Vol. 3, No. 3, pp. 191-200.
- [19] Potvin, J.-Y., Genetic algorithms for the traveling salesman problem, *Annals of Operations Research*, 1996,



Vol. 63, No. 3, pp. 337-370.

- [20] Rego, C., Gamboa, D., Glover, F., and Osterman, C., Traveling salesman problem heuristics : Leading methods, implementations and latest advances, *European Journal of Operational Research*, 2011, Vol. 211, No. 3, pp. 427-441.
- [21] Reinelt, G., TSPLIB-A traveling salesman problem library, *INFORMS Journal on Computing*, 1991, Vol. 3, No. 4, pp. 376-384.
- [22] Starkweather, T., McDaniel, S., Mathias, K., and Whitley, L., A comparison of genetic sequencing operators, in *Proceedings of the fourth International Conference on Genetic Algorithms*, 1991, pp. 69-76.
- [23] Whitley, D., A genetic algorithm tutorial, *Statistics and Computing*, 1994, Vol. 4, No. 2, pp. 65-85.
- [24] Yim, D.-S., A genetic algorithm for cell-to-switch assignment problem in cellular mobile networks, *Journal of the Korean Institute of Plant Engineering*, 2005, Vol. 10, No. 1, pp. 91-102.

#### ORCID

Young Nam Kim | <http://orcid.org/0000-0002-0762-6896>

Min Jung Lee | <http://orcid.org/0000-0002-8224-5364>

Chunghun Ha | <http://orcid.org/0000-0002-4222-2555>