유전 알고리즘을 이용한 Maximal Covering 문제의 해결

박태진* · 이용환* · 류광렬*

A Genetic Algorithm for the Maximal Covering Problem

Tae Jin Park · Yong Hwan Lee · Kwang Ryel Ryu

요 약

Maximal Covering 문제(MCP)란 행렬 상에서 n개의 열(column) 중 p개를 선택하여 m개의 행(row)중 최대한 많은 행을 cover하는 문제로 정의된다. 본 논문에서는 MCP를 유전 알고리즘(Genetic Algorithm)으로 해결하기 위해 문제에 적합하게 설계된 교차 연산자(crossover operator)와 비발현 유전인자(unexpressed gene)를 가진 새로운 염색체 구조를 제시한다. 해결 하고자 하는 대상 MCP의 규모가 매우 큰 경우 전통적인 임의교차(random crossover) 방법으로는 좋은 결과를 얻기가 힘들다. 따라서 본 연구에서는 그리디 교차(greedy crossover) 방법을 제시하여 문제를 해결한다. 그러나 이러한 그리디 교차를 사용하더라도 조기 수렴 등의 문제로 인해 타부 탐색 등의 이웃해 탐색 방법에 비해 좋은 결과를 얻기가 힘들다. 본 논문은 이러한 조기 수렴 문제를 해결하고 다른 이웃해 탐색 방법보다 더 좋은 결과를 얻기 위해 비발현 유전인자(unexpressed gene)를 가진 염색체를 도입하여 해결함을 특징으로 한다. 비발현 유전인자는 교차 과정에서 자식 염색체의 유전인자로 전달되지 않은 정보 중 나중에라도 유용할 가능성이 보이는 정보를 보존하는 역할을 하여 조기 수렴 문제를 해결하는데 도움을 주어 보다 나은 결과를 얻을 수 있게 해준다. 대규모 MCP를 해결하는 실험에서 새로운 비발현 유전인자를 적용한 유전 알고리즘이 기존의 유전 알고리즘 뿐만 아니라 다른 탐색 기법에 비해 더욱 좋은 성능을 보여줌을 확인하였다.

Key words : 유전알고리즘, Maximal Covering 문제, 비발현 유전인자, 최적화

1. 서 론

Maximal Covering 문제(MCP)는 잘 알려진 문제 중의 하나인 Set Covering 문제(SCP)와 유사하다. 많은 수의 열중 일부를 선택하여 행을 cover하는 문제라는 공통점이 있으며, 차이점은 SCP의 경우 반드시 모든 행을 cover해야 하는 제약 조건을 가지고 최소한의 열을 선택하는 문제인 반면 MCP의 경우는 선택할 수 있는 열의 개수가 미리 정해져 있는 제약 조건을 가지고 최대한의 행을 cover하는 문제이다. SCP, MCP는 모두 crew scheduling 문제, facility location 문제 등의 모델링을 위하여 많이 사용되며 SCP의 경우는 필요한 최소 자원수를 구할 경우 사용되며[1], MCP의 경우는 고정된 자원으로 최대한의 서비스를 수행해야 할 경우 많이 사용된다.[2] 본 논문에서는 이중 MCP를 유전 알고리즘을 이용하여 해결하려 한다.

기존 연구 [2]는 MCP를 유전 알고리즘을 이용해 해결할 경우 교차연산(crossover)을 통해 생성되는 자손들이 부모에 비해 더욱 좋지 않음을 지적하고 있다. 이는 결과적으로 전체집단이 교차연산을 통해 개선될 수 없음을 의미하고 결국 유전 알고리즘을 이용해 MCP를 해결할 경우 좋은 결과를 얻을 수 없음을 의미한다. 이는 SCP의 경우도 마찬가지이며 기존 연구 [3]에서는 이를 해결하기 위해 특수한 교차 연산자(crossover operator)를 개발하는 등의 방법을 동원하였다. 기존 연구 [2]에서도 이와 마찬가지로 그리디 교차 연산자(greedy crossover operator)를 개발하여 단순한 교차 연산자를 사용할 때보다 결과가 더욱 좋아짐 밝히고 있다. 그러나 이러한 그리디 교차 연산자를 동원할지라도 타부 탐색(Tabu Search), 언덕 오르기 탐색(hill-climbing search) 등의 이웃해 탐색 기법보다는 훨씬 좋지 않은 결과를 보여줌 또한 지적하고 있다. 이에 본 논문에서는 유전 알고리즘을 이용해 MCP를 해결하기 위해 그리디 교차 연산자를 사용할 경우 발생하는 문제를 해결하고 이웃해 탐색 기법보다 더 좋은 결과를 얻기 위해 비발현 유전인자(unexpressed gene)를 가진 염색체 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 대상

* 부산대학교 컴퓨터 공학과

문제인 MCP에 대하여 설명한다. 3장에서는 유전 알고리즘을 이용한 구현 방법에 대해서 소개한다. 4장에서는 실제 대규모 MCP에 대해서 타부 탐색과 비교 실험을 한다. 5장에서는 결론을 맺고 앞으로의 향후 연구방향에 대해서 토의한다.

2. Maximal Covering 문제

Maximal Covering 문제(MCP)는 행렬 상에서 n개의 열(column) 중 p개를 선택하여 m개의 행(row) 중 최대한 많은 행을 cover하는 문제로 정의된다. 이를 수식으로 표현하면 다음과 같다.

$$\text{Max} \quad \sum_{i=1}^{m} y_i \tag{2.1}$$

subject to :

$$\sum_{j=1}^{n} a_{ij} x_j \geq y_i \ , i = 1,2,...,n \tag{2.2}$$

$$\sum_{j=1}^{n} x_j = p \tag{2.3}$$

$$y_i \in (0,1) \ , i = 1,2,...,n \tag{2.4}$$

$$x_j \in (0,1) \ , j = 1,2,...,m \tag{2.5}$$

y_i, x_j는 0 또는 1의 값을 가지는 결정 변수로서 y_i의 경우 i번째 행이 cover되면 1의 값을 가지며 x_j의 경우 j번째 열이 선택될 경우 1의 값을 가지게 된다. a_{ij}는 m개의 행과 n개의 열로 이루어진 행렬로서 j번째 열이 i번째 행을 cover할 경우 1의 값을 취하게 된다. 목적 함수 (2.1) 은 cover되는 행의 수를 최대화한다. 제약 조건 (2.2)는 열과 행의 관계를 결정지어 준다. 제약 조건 (2.3)은 선택될 열의 개수가 p가 됨을 보장해 준다. 제약 조건 (2.4), (2.5)는 결정변수 x, y 가 0 또는 1을 가짐을 보장해 준다.

MCP의 예로 [그림 1]을 6개의 열중 3개를 선택하여 5개의 행을 최대한 cover하는 MCP라 가정하면 열의 집합 {1, 2, 3}은 하나의 해가 될 수 있다. 이때 이 해에 의해 cover되는 행의 개수는 {1, 2, 3}의 3개이며 1번 행은 1, 3번 열에 의해 2번 행은 2번 열에 의해 3번 행은 1번 열에 의해 cover되었다고 한다.

[그림 1] maximal covering 문제의 예

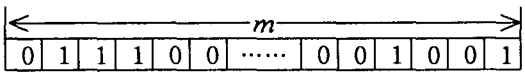
	1	2	3	4	5	6
1	1		1		1	
2		1				
3	1			1		
4				1		1
5					1	

3. 유전 알고리즘의 적용

3.1 염색체 표현

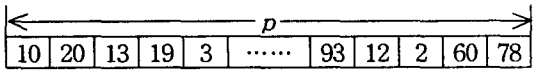
일반적인 유전 알고리즘의 경우 염색체의 구성에 이진표현을 많이 사용하고 이러한 경우 가장 좋은 성능을 보여주는 것으로 알려져 있다. MCP를 이진 표현을 통해 해결 할 경우 [그림 2]와 같이 각 염색체가 열의 수 m 만큼의 유전인자(gene)를 가지게 하고 각 각의 유전인자는 해당 열이 해의 구성을 위해 선택되었는지를 1 또는 0의 값으로 표현하게 된다. 그러나 염색체 내의 1또는 0의 개수가 p개로 고정되어야 하는 제약 조건이 있으며 이러한 제약조건이 교차연산(crossover)이나 돌연변이 연산(mutation)을 통해 깨어지기가 쉽다는 문제가 있다. 또한 대규모 MCP의 경우 열의 개수가 몇 만에서 몇 십만인 경우가 많으며 이때 염색체의 길이 또한 몇 만에서 몇 십만이 되어야 한다. 이럴 경우 염색체의 길이가 너무 길어져 좋은 결과를 기대하기 어렵게 된다.

[그림 2] 염색체의 이진 표현



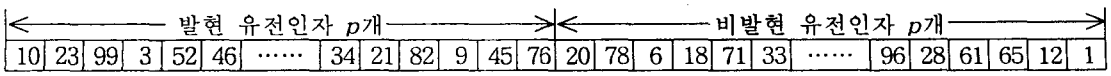
이러한 문제 때문에 본 연구에서는 [그림 3]과 같이 염색체를 정수표현을 사용하여 구성한다. 정수표현 상에서 각 각의 염색체는 정확히 p개의 유전인자를 가지며 각 각의 유전인자는 선택된 열의 인덱스를 가지게 된다.

[그림 3] 염색체의 정수 표현



3.5에서 다시 언급하게 되겠지만 본 논문에서는 MCP를 유전알고리즘을 이용해 효율적으로 해결하기 위해 교차연산자로 그리디 교차연산자를 사용한

[그림 4] 비발현 유전인자를 가진 염색체



다. 그리디 교차 연산자를 사용할 경우 교차연산 과정에서 전체집단이 빠르게 지역 최적해로 수렴해서 좋은 결과를 얻지 못하는 조기수렴 문제가 나타나는 경우가 많다. 본 논문에서는 이러한 조기 수렴 문제를 해결하는 한 방법으로 비발현 유전인자를 도입한다.

비발현 유전인자를 가진 염색체는 [그림 4]와 같이 $2p$개의 유전인자를 가지게 되며 이들 중 염색체의 평가와 후보해의 구성을 위해 사용되는 p개의 유전인자는 발현 유전인자라고 하며, 나머지 p개의 유전인자는 해의 평가에 사용되지 않으므로 비발현 유전인자라고 한다. 실험을 통해 밝히게 되겠지만 비발현 유전인자의 경우 후보해의 구성에는 사용되지 않지만 나중에라도 유용할 가능성이 보이는 열의 정보를 보존하는 역할을 한다.

3.2 적합도 평가(Fitness Evaluation)

각 염색체의 적합도는 각 염색체가 구성하는 해가 cover하는 행의 수가되며, 그 수가 클수록 적합도는 높아진다. 비발현 유전인자를 가진 염색체의 경우 발현 유전인자에 의해 cover되는 행의 수가 적합도가 된다.

3.3 초기 집단 생성

각각의 염색체는 greedy adding 휴리스틱을[4] 사용하여 초기화되며 이 알고리즘은 기존 연구 [5]에서 타부 탐색에 사용된 초기해 생성 방법과 같으며 자세한 알고리즘은 [그림 5]에 주어져 있다. 초기에 해의 집합 x에는 선택된 열이 하나도 존재하지 않으며 제약조건에 의해 지정된 p개의 열이 선택될 때까지 하나씩 추가하는 과정을 반복적으로 수행한다.

[그림 5] 초기해 생성 알고리즘

```
Algorithm Initialize()
  C : Set of Parings
  x : Current solution
  p : The number of parings to be selected
Begin
  x := ∅;
  while |x| != p Do
    find j which has maximum Δj where j ∈ C
    x = x ∪ {j}
    C = C - {j}
  End While
  Return x
End Begin
```

C는 전체 열중 현재 x에 포함되어 있지 않은 열들의 집합을 의미하는 것이며 Δ_j는 현재해의 상태에서 j번째 열에 의해서만 cover되는 행의 개수를 의미한다. 즉, 현재 남아있는 열들 중 다른 열이 cover하지 않은 행을 가장 많이 cover하는 열이 우선 적으로 선택되는 것이다. 만약 Δ_j값이 동일한 열이 하나 이상 존재할 경우 임의의 열이 선택된다. 전체 집합의 크기만큼 위 알고리즘을 반복 수행하여 전체 집합을 초기화한다.

비발현 유전인자를 포함한 염색체의 경우 발현 유전인자는 위와 같은 방법으로 초기화되고 비발현 유전인자의 경우 p개의 열을 임의로 선택한다.

3.4 선택(Selection)

교차연산을 위한 부모 염색체의 선택은 Stochastic Universal Sampling(SUS) 방법을 사용하며 그 알고리즘은 [그림 6]과 같다.[5]

[그림 6] Stochastic Universal Sampling

```
prt = Rand() /* return random number uniform
              distributed in [0, 1] */
for(sum = i = 0; I < N; i++)
    for(sum += ExpVal(i,t); sum > ptr; ptr++)
        Select(i);
```

$ExpVal(i,t)$는 집단 내 i번째 염색체가 t세대에 선택될 기대 값이다. 이 값의 횟수만큼 각 염색체가 선택되기를 기대할 수 있다.

SUS하에서 각 염색체는 최소 $\lfloor \mathrm{Exp}Val(i,t) \rfloor$ 에서 최대 $\lceil \mathrm{Exp}Val(i,t) \rceil$ 횟수만큼 선택됨이 보장되어 Roulette Wheel selection 과 같은 단순 random selection 방법에서 나타나는 selection error를 줄여준다.

3.5 그리디 교차연산 (Greedy Crossover)

해결하려는 문제의 규모가 매우 크고 각 유전인자의 위치가 의미를 가지지 않기 때문에 전통적인 일점교차, 이점교차 방법으로는 좋은 결과를 얻기가 힘들다. 따라서 여기서는 MCP에 적합하게 설계된 그리디 교차(greedy crossover)방법을 제안한다.

그리디 교차 방법은 2개의 부모 염색체로부터 1개의 자식 염색체를 생성하며, 기본적으로 초기 집단 생성 방법과 유사하다. 즉, 교차연산 초기에는 염색체 내에 선택된 열이 하나도 존재하지 않으며 열의 개수 제약조건에 지정된 개수 p개가 선택될 때까지 하나씩 추가하는 과정을 반복적으로 수행한다. 각 과정에서 아직 cover되지 않은 행을 가장 많이 cover하는 열이 우선 적으로 선택된다. 그러나 초기해 생성의 경우 전체 열 n개중 p개를 선택하는 문제이지만 그리디 교차의 경우 2개의 선택된 부모 염색체 안에 존재하는 $2p$개의 열중 p개를 선택하는 문제로 간소화된다. 또한 초기집단 생성 시에는 2개 이상의 열이 같은 수의 아직 cover되지 않은 행을 cover할 경우 임의로 하나를 선택하였지만 교차연산 시에는 식 (3.1)과 같은 $similar_{all}$ 값을 구해서

이 값이 가장 작은 열을 우선적으로 선택한다. 즉, 우선적으로 cover되지 않은 행을 cover하는 수를 비교하여 큰 열을 선택하고 cover하는 수가 같은 열이 있을 경우 $similar_{all}$ 값이 작은 것을 우선적으로 선택하는 것이다.

$$similar_{all}(c) = \sum_{v \in G} c \cdot v = c \cdot \sum_{v \in G} v \qquad (3.1)$$

$$P = P_1 \cup P_2 \qquad (3.2)$$

c는 $similar_{all}$ 값을 구하고자 하는 열의 벡터이며 $P1$, $P2$는 2개의 부모 염색체들이다. 즉, $similar_{all}$ 값은 해당 열과 부모 염색체 내에 존재하는 모든 열들과의 유사도의 합이다. 결국 $similar_{all}$ 값이 작을수록 다른 열들과의 유사도가 낮다는 의미가 된다. 다른 열들과 유사도가 낮은 열의 경우 다른 열들이 잘 cover하지 못하는 행을 많이 cover하게 됨으로 이런 열들을 우선적으로 선택한다. 예들 들어, [그림 7]과 같은 상황에서는 c5를 가장 우선 적으로 선택하게 된다.

비발현 유전인자를 가진 염색체의 경우 발현 유전인자는 위와 동일한 방법으로 결정한다. 단, 비발현 유전인자가 없는 경우 $2p$개의 열중 p개를 결정하는 반면 비발현 유전인자를 가진 경우 비발현 유전인자까지 포함해서 $4p$개의 열중 p개를 선택하게 된다. 자식 염색체의 비발현 유전인자의 경우 발현 유전인자를 결정하는 것과 같은 방법을 사용하지만 이 경우 $similar_{all}$값 대신 $similar_{gene}$ 값을 사용한다.

$$similar_{gene}(c) = \sum_{v \in G} c \cdot v = c \cdot \sum_{v \in G} v \qquad (3.3)$$

여기서 G는 발현 유전인자에 포함된 열의 집합이다. $simailar_{all}$과의 차이점은 유사도 계산을 하는 대상이 부모 염색체에 포함된 모든 열이 아니라 발현 유전자에 포함된 열과의 유사만을 계산한다는 점이다. 결과적으로 비발현 유전인자에 포함되는 열은 발현 유전인자에 포함된 열들과 유사도가 가장 낮은 열이 우선 적으로 선택된다.

[그림 7] *similar* 값의 예

	c_1	c_2	c_3	c_4	c_5	Σ
r_1	0	0	0	0	1	1
r_2	0	1	1	0	1	2
r_3	1	1	1	1	0	4
r_4	1	0	1	0	0	2
r_5	0	0	1	0	1	2
r_6	0	0	0	1	0	1
r_7	0	0	0	0	0	0
r_8	1	1	0	1	0	3
r_9	0	0	0	0	1	1
r_{10}	1	1	0	1	0	3
$c_i \cdot \Sigma$	12	12	10	11	6	

3.6 돌연변이(Mutation)

돌연 변이 연산의 경우 기존 연구 [4]에서 타부 탐색을 이용한 MCP 해결 방법 중 이웃해 생성 방법인 k-exchange neighbor 방법과 유사하다. 실제 지역 탐색이란 교차연산 없이 돌연변이만으로 탐색하는 방법이라고 생각할 수도 있으므로 타부 탐색의 이웃해 생성방법을 돌연변이 연산으로 사용하는 것은 합당하다고 볼 수 있다.

여기서 사용한 돌연변이 연산은 k-exchange mutation이라고 한다. 일반적인 돌연변이 방법은 각각의 유전인자가 일정 확률로 돌연변이를 일으키는 반면 k-exchange mutation의 경우 각 염색체가 일정 확률로 돌연변이를 일으킨다. 각 각의 염색체에 대해서 돌연변이 테스트를 한 후 염색체가 돌연변이 할 것으로 결정될 경우 k개의 열을 선택해서 제거한 후 초기해 생성과 같은 그리디한 방법으로 k개를 추가한다.

제거할 k개를 선택하는 방법은 현재해의 상태를 고려하여 확률적으로 선택하는 방법을 사용하였다. 각 열의 제거 확률은 해당 열이 제거됨으로써 줄어드는 coverage 값에 비례하게 설정하였다. coverage 값이 적게 줄어들수록 개선될 가능성이 높기 때문이다. 예를 들어 줄어드는 coverage 값에 따라 제거될 확률을 [표 1]과 같이 설정할 수 있다.

k개를 추가하는 방법은 초기해 생성 시 열을 추가하는 방법과 완전히 동일하다.

비발현 유전인자를 가진 염색체의 경우 발현 인자에 대해서는 위와 동일한 돌연변이 방법을 사용한다. 그러나 비발현 유전인자에 대해서는 간단한 point mutation 방법을 사용한다. 즉 각각의 유전인자가 일정확률로 돌연변이를 일으키며 돌연변이가 결정될 경우 해당 유전인자의 열은 임의의 열로 치환된다.

[표 1] 열 제거 시 확률적 선택을 위한 제거 확률의 예

줄어드는 cover 수	제거 확률
10	0.95
9	0.75
8	0.5
7	0.3
6	0.2
5	0.2
4	0.2
3	0.2
2	0.2
1	0.1
0	0.1

4. 실험

본 논문에서 제안한 유전 알고리즘을 평가하기

위해 다른 휴리스틱 알고리즘인 타부 탐색 방법과 비교하였다. 타부 탐색은 기존 연구 [4]에 나온 방법에 다각화를 좀더 개선하여 사용하였다. 초기해의 생성 방법은 유전 알고리즘의 초기집단 생성방법과 동일하며 평가함수 또한 동일하다. 이웃해의 경우는 유전 알고리즘의 k-exchange mutation과 유사한 k-exchange neighbor 생성 방법을 사용한다. 다각화는 탐색이 진행되는 동안 각 열이 cover하는 행의 정보를 누적 시켜 두었다가 다각화시 비교적 많이 cover되었던 행을 많이 cover하는 열을 우선적으로 제거하고 적게 cover되었던 행을 많이 cover하는 열을 우선 적으로 추가하는 row frequency based 메모리 방법을 사용한다.

실험에 사용된 데이터는 국내 모 도시의 지하철 승무일정 데이터이다. 지하철 승무일정 계획 문제는 고정된 숫자의 승무원으로 전체 열차 계획을 운행하는 문제로서 MCP로 모델링 된다. 모델링된 MCP는 약 18만개의 열중 83개를 뽑아서 814개의 행을 최대한 cover하는 문제이다. 본 데이터의 경우 문제 특성상 18개의 열은 미리 고정되어 있어 실제는 65개의 열만을 선택하면 됨으로 해의 길이는 65가 된다. 각 열은 실제 한 사람의 승무원에 의해 운행되는 근무표이며 다음과 같은 제약 조건들이 존재한다.

1. 각 열은 승무원의 대기소에 따라 2종류로 분류되며 각 종류별로 아래 개수만큼 선택되어야 한다.
   a. 대기소 A : 38
   b. 대기소 B : 27
2. 평균 근무 시간의 제한 : 각 열은 근무시간 정보를 가지고 있으며 선택된 열들의 평균 근무시간은 일정시간 이하여야 한다.

제약 조건을 만족시키기 위해서 유전 알고리즘의 초기집단 생성, 교차연산, 돌연변이 연산 타부 탐색 알고리즘의 초기해 생성, 이웃해 생성, 다각화 과정에서 열을 해에 추가 할 때 제약조건을 반드시 지키도록 수정하였다.

모든 실험은 PentiumIV 2G, 1G Ram, 운영체제로 FreeBSD 4.5를 사용한 서버 상에서 1시간의 시간제한을 두고 수행하였다. 유전 알고리즘의 경우 비발현 유전인자와 k-exchange mutation이 유전 알고리즘 내에서 성능에 미치는 영향을 확인하기 위해 3가지 서로 다른 설정을 사용하였다. 자세한 설정은 아래와 같다.:

GA :
전체집단 크기 : 1500
3-exchange mutation 사용
k-exchange mutation 확률 : 1%
비발현 유전인자 돌연변이 확률 : 10%

GA without unexpressed gene :
전체진단 크기 : 3000
3-exchange mutation 사용
k-exchange mutation 확률 : 1%

GA without k-exchange mutation :
전체집단 크기 : 1500
비발현 유전인자 돌연변이 확률 : 10%

비발현 유전인자를 사용하지 않은 유전 알고리즘의 경우 염색체의 길이가 다른 두 경우에 비해 반이므로 전체 집단의 크기를 2배인 3000으로 설정하였으며 유전 알고리즘을 위한 공통적인 설정들은 실험을 통해 최적의 설정을 사용하였다.

비교 대상으로 사용된 타부탐색의 자세한 설정은 아래와 같다.:

Tabu :
이웃해 : 19개 생성 {4, 5, 5, 5, 5}
Tabu tenure : 10
Aspiration condition : simple aspiration
Diversification condition : not improved 100 iteration
Diversification period : 10 iteration

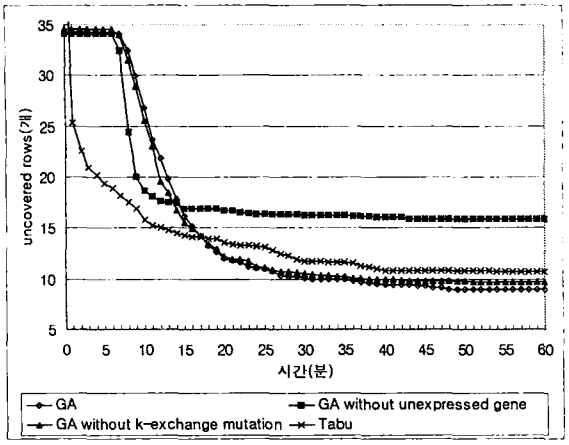
이웃해 생성 방법에서 {4, 5, 5, 5, 5}의 의미는 i번째 숫자가 생성될 i-exchange 이웃해의 개수이다. 즉, 1-exchange 의 경우 4개가 생성되며 2에서 5-exchange 이웃해의 경우 5개씩 생성된다. tabu tenure가 10 임으로 한번 해에 추가되었거나 제거되는 정보는 탐색이 10회 진행되는 동안 다시 제거되거나 추가되지 못한다. 단 aspiration condition에 의해서 현재까지 찾은 가장 좋은 해 보다 더 좋아지는 경우 타부 리스트에 있더라도 해에 추가되거나 제거 될 수 있다. diversification condition에 의해 100회 동안 최적해의 개선이 없으면 다각화 상태로 돌입하며 다각화는 10회 동안 수행된다.

[표 2] 지하철 승무 일정 문제 적용 결과

	최대	평균	최소
GA	10	9.2	8
GA without unexpressed gene	17	15.9	15
GA without k-exchange mutation	12	9.7	8
Tabu Search	12	10.7	9

[표 2]는 지하철 승무일정계획 문제에 유전 알고리즘과 타부 탐색을 1시간씩 10회 적용시켜 본 결과이다. 숫자가 뜻하는 것은 cover하지 못한 row의 수로서 낮을수록 좋다. 실험 결과에 의하면 비발현 유전인자와 k-exchange mutation을 모두 적용한 유전 알고리즘이 가장 좋으며, 그 다음으로 k-exchange mutation을 사용하지 않은 유전 알고리즘, 타부 탐색, 비발현 유전인자를 사용하지 않은 유전 알고리즘 순으로 좋다는 것을 알 수 있다. 실험 결과에서 비발현 유전인자를 사용한 경우와 사용하지 않은 경우 특히 성능차이가 많이 나는 것으로 보아 비발현 유전인자가 중요한 역할을 하고 있다는 것을 확인할 수 있다.

[그림 8] 시간에 따른 최적해 변화



[그림 8]은 시간 변화에 따른 최적해의 변화를 그래프로 그려본 것이다. 유전 알고리즘 초기에 일정 시간 동안 최적해의 변화가 없는 것은 초기집단을 그리디 하게 생성했기 때문에 나타나는 현상이다. 탐색의 초반에는 타부 탐색이 좋은 결과를 보여 주지만 후반으로 갈수록 유전 알고리즘의 성능이 더 좋아짐을 보여 주고 있다. 또한 비발현 유전인자를 사용하지 않은 유전 알고리즘의 경우 해가 초기에 너무 빠르게 개선되어 좋은 결과를 얻지 못하는 조기 수렴 문제가 나타남을 확인할 수 있다.

[표 3]은 비발현 유전인자를 사용한 경우와 사용하지 않은 경우 전체 모집단 안에 있는 서로 다른 열의 개수 변화를 세대수에 따라 나타낸 것이다. 2번째, 3번째 열은 비발현 유전인자를 사용한 유전 알고리즘에서 전체 집단 내에 포함된 서로 다른 열의 개수를 보여 주고 있으며, 2번째 열은 발현 유전인자 내에 포함된 열의 개수만을 3번째 열은 비발현 유전인자 내에 포함된 것까지 합쳐서 보여 주고 있다. 4번째 열의 경우는 비발현 유전인자를 사용하지 않은 유전 알고리즘에서 전체 집단에 포함된 서로 다른 열의 개수 변화를 보여 주고 있다. [그림 9]는 이를 그래프로 나타낸 것이다.

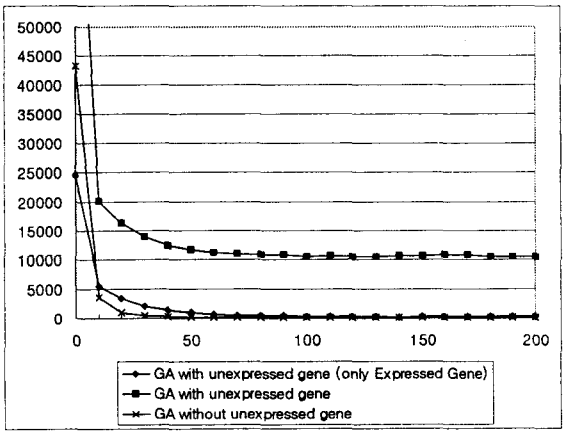
비발현 유전인자의 역할은 이 그래프에 의하면 2가지로 볼 수 있다.

그 첫째는 수렴 속도를 줄여 주는 것이다. 비발현 유전인자를 사용한 경우와 비발현 유전 인자를 사용하지 않은 경우의 발현인자를 비교해 보면 비발현 유전인자를 사용하지 않은 경우 전체집단의 수가 2배이므로 거의 2배정도 많은 서로 다른 열들을 가지고 탐색을 시작하지만 10세대를 진행한 후에는 이미 그 개수가 10분의 1이하로 줄어들어 있는 것을 확인 할 수 있다. 이는 그리디 교차연산이 2개의 부모 염색체로부터 하나의 자식 염색체를 생성하기 때문에 교차연산 과정에서 해의 구성을 위해 선택되지 못한 절반의 열 정보가 버려지기 때문에 엄청 나게 빠른 속도로 수렴이 일어나고 있기 때문이다. 그러나 비발현 유전인자를 사용한 경우에는 교차연산 과정에서 해의 구성을 위해 선택되지 못하더라도 이후의 탐색과정에서 유용하게 쓰일 수 있다고 판단되면 바로 버려지지 않고 비발현 유전인자를 통해 열 정보가 보존되기 때문에 버려지는 열의 정보가 훨씬 적어져 수렴 속도가 느려지게 된다. 이로 인해 시작은 비발현 유전인자를 사용하지 않은 경우 보다 적은 개수의 서로 다른 열들을 가지고 시작했지만 이미 10세대에서 더 많은 서로 다른 열들을 전체 집단 내에 가지고 있음을 알 수 있다. 또한 [그림 9]의 그래프를 보아도 비발현 유

[표 3] 세대에 따른 서로 다른 열의 개수

세대수	GA		GA without unexpressed gene
	발현 유전인자만	비발현 유전인자 포함	
0	24445	92611	43330
10	5579	20225	3560
20	3463	16357	989
30	2128	14044	433
40	1308	12497	318
50	1019	11772	250
60	714	11179	197
70	544	10967	201
80	532	10873	174
90	464	10835	178
100	408	10584	156
110	338	10726	137
120	331	10574	165
130	321	10596	145
140	245	10632	120
150	293	10733	116
160	282	10936	101
170	272	10776	111
180	287	10522	110
190	267	10513	109
200	269	10592	110

[그림 9] 세대에 변화에 따른 서로 다른 열의 개수



전인자를 사용한 경우가 사용하지 않은 경우에 비해 수렴 속도가 더욱 느림을 확인 할 수 있다.

비발현 유전인자의 두 번째 역할은 수렴 후에도 전체집단의 다양성을 유지해 준다는 것이다. 비발현 유전인자를 가지지 않은 경우 수렴 후 전체집단 내에 포함된 서로 다른 열의 개수가 무척 적기 때문에 교차연산에 의한 해의 개선은 거의 기대할 수 없는 상태가 되어 돌연변이 연산에 의한 해의 개선만 이루어지게 된다. 그러나 비발현 유전인자를 가진 경우에는 수렴 후에도 발현 유전인자들 내에 비교적 많은 수의 서로 다른 열들을 가지고 있으며 비발현 유전인자 내에는 충분한 양의 서로 다른 열들을 가지고 있으므로 돌연변이뿐만 아니라 교차연산에 의한 개선 또한 기대할 수 있게 된다.

결과적으로 비발현 유전자의 이러한 두 가지 역할에 의해 비발현 유전인자를 가진 유전 알고리즘이 그렇지 못한 경우에 비해 훨씬 좋은 결과를 보여주게 되는 것이다.

[표 4] 인공적으로 생성한 MCP 데이터

	rows	columns	p	covered rows per column
mcp634	634	142265	65	10
mcp520	520	92139	53	9~10
mcp453	453	72094	45	9~10
mcp384	384	54738	42	8~9
mcp312	312	55807	39	7~8

[표 4]는 다양한 크기의 MCP를 해결함에 있어 유전 알고리즘이 타부 탐색에 비해 더 좋은 결과를 보여주는지 확인하기 위해 사용한 추가 실험 데이터이다. 지하철 승무일정계획 데이터에서 행과 열의 수를 줄여서 변형한 데이터이다.

[표 5] MCP 해결 결과

Data	GA			Tabu		
	최소	평균	최대	최소	평균	최대
mcp634	4	6.3	8	8	8.8	10
mcp520	4	5.3	6	5	6.2	7
mcp453	12	12.3	13	11	12	13
mcp384	6	7.3	8	7	7.9	9
mcp312	2	2.7	3	3	4	5

[표 5]는 이 데이터에 유전 알고리즘과 타부 탐색을 적용시켜 본 결과이며 모든 데이터에 대해 1시간의 시간 제약을 두고 10회씩 실험하였다. 유전 알고리즘과 타부 탐색의 설정은 지하철 승무일정계획 데이터에서 사용한 것과 동일한 설정을 사용하였다.

mcp453을 제외하고는 유전 알고리즘이 더 좋은 결과를 보여 주고 있음을 확인 할 수 있다. 특히 문제의 규모가 가장 큰 mcp634 에 대해서는 아주 큰 차이로 유전 알고리즘이 더 좋은 결과를 보여주고 있음을 확인 할 수 있다. 또한 타부 탐색이 더 좋은 결과를 보여준 mcp453의 경우에는 타부 탐색과 유전 알고리즘의 성능차이가 무척 작다는 것을 알 수 있다.

5. 결론 및 향후 연구 방향

본 논문에서 대규모 MCP를 효과적으로 해결하기 위한 유전 알고리즘을 제안하였으며, 실험을 통해 MCP의 규모가 매우 큰 경우 타부 탐색에 비해 더 좋은 결과를 얻을 수 있음을 보였다.

MCP를 유전 알고리즘을 이용해 해결할 경우 전통적인 교차 방법으로는 좋은 결과를 기대하기 힘들기 때문에 그리디 교차연산을 사용하였다. 그러나 그리디 교차연산을 사용하더라도 조기 수렴 등의 문제로 인해 타부 탐색 등의 이웃해 탐색 기법들에 비해 좋지 못한 결과를 보여주게 된다. 본 논문에서는 이러한 문제를 해결하고 이웃해 탐색 기법들 보다 더 좋은 결과를 얻기 위해 비발현 유전인자를 가진 염색체 구조를 제시하였으며 실험을 통해 조기수렴 문제의 해결에 도움이 되며 수렴 후에도 전체 집단의 다양성을 유지시키는데 도움이 됨을 보였다. 또한 실험을 통해 이러한 비발현 유전인자를 도입한 유전 알고리즘이 Maximal Covering 문제의 해결에 있어 더 좋은 결과를 얻을 수 이 있음을 보였다.

본 논문에서 제안한 비발현 유전인자를 가진 염색체 구조는 본 논문에서 해결한 대상 문제 외에도 다양한 MCP의 해결에 적용이 가능할 것으로 판단되며, 나아가 MCP외의 SCP등의 다른 최적화 문제에도 적용이 가능할 것으로 보인다. 따라서 향후 다른 MCP를 비롯한 다양한 최적화 문제로의 적용을 통해 본 논문에서 제시한 방법의 효과를 검증해 볼 필요가 있을 것으로 사료된다.

참고 문헌

[1] B.M. Smith, A. Wren, *A Bus Crew Scheduling System Using a Set Covering Formulation*, Transportation Research, 22A, 97-108, 1988

[2] 황준하, *휴리스틱 탐색기법과 정수계획법의 결합에 의한 승무일정계획의 최적화*, 박사 학위논문, 부산대학교, 2002.

[3] J.E. Beasley and P.C. Chu, *A Genetic Algorithm for the Set Covering Problem*, European Journal of Operational Research, 94, 392-404, 1996

[4] Church, R., and ReVelle, C., *The Maximal Covering Location Problem*, Naval Research Logistics, 43, 435-461, 1996

[5] 강창성, *Maximal Covering 문제를 위한 효율적인 이웃해 생성 전략*, 석사학위 논문, 부산대학교, 2001

[6] Melanie Mitchell, *An Introduction to Genetic*

Algorithms, MIT Press, 1997
[7] D. Hochbaum, *Approximation Algorithms for Set Covering and Vertex Cover Problem, SIAM Journal on Computing*, 11(3), 555-556, 1982.
[8] R.L. Church, C.S. ReVelle, *The Maximal Covering Location Problem*. Regional Science, 32, 101-118, 1974
[9] Elon Santos Correa, *A Genetic Algorithm for the P-Median Problem*, GECCO-2002