유전알고리듬을 이용한 속성의 중복 허용 파일 수직분할 방법

김 재 련* · 유 종 찬**

An Attribute Replicating Vertical File Partition Method by Genetic Algorithm

Jae-Yearn Kim* · Jong-Chan Yoo**

Abstract

The performance of relational databases is measured by the number of disk accesses necessary to transfer data from disk to main memory. The paper proposes to vertically partition relations into fragments and to allow attribute replication to reduce the number of disk accesses. To reduce the computational time, heuristic search method using genetic algorithm is used. Genetic algorithm used employs a rank-based-sharing fitness function and elitism. Desirable parameters of genetic algorithm are obtained through experiments and used to find the solutions.

Solutions of attribute replication and attribute non-replication problems are compared. Optimal solutions obtained by branch and bound method and by heuristic solutions(genetic algorithm) are also discussed. The solution method proposed is able to solve large-sized problems within acceptable time limit and shows solutions near the optimal value.

※ 본 연구는 1998년도 한양대학교 교내연구비에 의해서 연구된 것임.
* 한양대학교 공과대학 산업공학과
** LG 생산 기술원

1. 서 론

1.1 연구의 배경과 목적

데이터베이스 시스템에서 수행도 문제는 주요한 연구의 대상이 되고 있다. 수행도(performance)는 하나의 트랜잭션(transaction)을 처리하기 위해 소요되는 응답시간(response time)을 뜻하며, 일반적으로 하나의 트랜잭션을 처리하기 위해 디스크 접근 해야 하는 디스크 I/O 횟수에 의해 결정된다. 트랜잭션을 처리할 때 주기억장치와 디스크간에 이동해야 하는 데이터의 양이 많으면 디스크의 접근 횟수가 증가한다. 트랜잭션은 대상 릴레이션의 모든 속성을 필요로 하지 않는 경우가 대부분이므로 릴레이션의 속성을 몇 개의 단편으로 나누어 저장하면, 불필요한 속성을 디스크에서 주기억장치로 옮길 필요가 없다. 이와 같이 속성을 단편단위로 저장하는 문제가 수직분할문제이다. 속성단위로 분할하는 수직분할문제는 데이터의 이동 양을 줄이는 하나의 방법으로써, 속성의 중복을 고려하지 않은 비중복 알고리듬[윤병익, 김재련 1996]은 한 개의 속성을 반드시 한 개의 단편에 할당하는 제약을 갖고 있다. 그러므로 빈번하게 접근되는 속성을 한 개의 단편에만 할당하는 경우 트랜잭션에서 필요로 하는 특정속성 한 개를 액세스하기 위해 단편 모두를 접근해야 한다. 따라서 단편에 할당되는 속성의 수에 관한 제약을 완화하여 속성의 중복을 허용하는 수직분할 방법[유종찬, 김재련 1998]을 사용하면 문제에 따라서 디스크 접근 횟수를 줄일 수 있다.

트랜잭션의 응답시간은 디스크 접근 횟수에 비례하므로, 불필요한 속성의 이동을 줄이기 위한 방법으로 속성의 중복을 허용하는 수직분할 모형을 고려한다. 그러나 이와 같은 문제는 조합 최적화 문제로써 속성의 수에 따라 지수적으로 증가하는 계산량을 가진다. 그러므로 고려하는 속성의 수가 커지면 최적해를 구하는데 시간이 많이 소요된다. 따라서 본 연구에서는 중복허용 수직분할 문제의 최적해에 근접하는 해를 유효한 시간내에 풀 수 있는 해법으로 유전알고리듬을 이용한 발견적 기법을 제시하였다.

1.2 수직 분할문제의 기존 연구

수직분할문제에 대한 기존연구로는 데이터베이스를 설계할 때, 논리적단계와 물리적단계에서 분할하는 연구의 두 부류로 나눌 수 있다. 전자의 경우 친밀도(affinity)를 최대화하는 연구[Cheng 1994], [Hoffer, Severace 1975], [Navathe, Wiederhold, Dou 1984], [Vavathe, Ra 1989]가 있고, 후자는 디스크 접근 횟수를 최소화하는 연구[윤병익, 김재련 1996], [Chu, Ieong 1993], [Cornell, Yu 1990], [Cornell, Yu 1987]가 있다. Hoffer와 Severance [1975]는 속성들을 조합하여 친밀도가 높은 속성들로 그룹핑하여 수직분할하는 알고리듬을 개발하였고, Navathe[1984]는 [Hoffer, Severance 1975]의 연구를 확장하여 2단계 이진 수직분할 알고리듬을 제안하였다. Navathe와 Ra[1989]는 [Navathe, Wiederhold, Dou 1984]의 수직분할방법의 친밀도 행렬에서 두 속성들 사이의 친밀도를 edge값으로 하는 친밀도 그래프 알고리듬을 개발하였다. Cheng [1994]은 속성사용행렬의 mutually separable cluster를 찾아내기 위한 분지한계법과 발견적 기법을 제안하였으나, 입력변수로 속성사용행렬만을 사용하여 분할하므로 최적 디스크 접근 횟수를 구한다는 보장이 없다. 위의 연구에서는 구체적인 물리적 디스크 접근 비용을 고려하지 않았다. Cornell과 Yu[1987, 1990]는 속성들을 물리적인 단편에 할당하여 디스크 접근 횟수를 최소화하는 이진분할 선형 정수계획법을 개발하였다. 또한 재분할에 관한 수리모형을 제안하였으나, 분할의 수가 커지면 계산시간이 많이 걸린다. Chu[1993]는 Cornell and Yu의 문제를 트랜잭션에 근거한 접근 방법으로 이진 수직분할을 연구하였다. 최적 이진 수직분할은 reasonable cut에서만 존재함을 보이고, 이진분할에 대한 해법으로 분지한계법과 발견적 기법을 제시하였다. 위의 모든 연구는 속성들을 두 개의 단편으로만 분할하는 알고리듬을 개발하여, 두

개 이상의 단편으로 분할할 경우, 알고리듬을 반복적으로 적용하여 해를 구하므로 최적단편이 여러 개인 문제에서는 최적해를 구한다는 보장이 없다. 윤[1996]은 디스크 접근 횟수를 줄이기 위한 수직분할문제의 0-1정수 계획모형을 개발하고, 모형의 해법으로 N-ary분지한계법을 제안하여 최적해를 구하였다. 또한 유[1998]는 [윤병익, 김재련 1996]의 연구를 확장하여 속성의 중복을 허용하는 0-1정수 계획모형과 모형에 대한 최적해법으로 분지한계법을 제안하였다. 그러나 [유종찬, 김재련 1998]이 최적해법으로 제시한 분지한계법은 속성의 수에 따라 계산량이 지수적으로 증가하므로, 큰 문제를 푸는데는 많은 시간이 소요되어 현실적으로 해법을 적용하는데는 한계가 있다. 따라서 본 연구에서는 [유종찬, 김재련 1998]의 중복허용 수직분할 모형을 사용하여 큰 문제를 풀기 위해 유전해법을 제시한다.

1.3 중복허용 수직분할 문제

릴레이션을 몇 개의 단편으로 수직분할할 때, 동일한 속성을 두 개 이상의 단편에 배치하는 것을 속성의 중복할당이라 한다. 속성을 중복하는 것은 트랜잭션을 처리하기 위해 액세스되는 단편의 수를 감소시켜 디스크의 접근횟수를 줄이는 방법이다. 만일 여러 개의 속성을 포함하는 단편이 트랜잭션에서 요구하는 하나의 속성을 위해서 접근되면, 같은 단편에 포함된 필요로 하지 않는 속성도 같이 접근되므로 데이터의 이동량이 증가한다. 이와 같이 필요치 않은 속성의 이동을 줄이기 위하여 속성을 여러 단편에 중복하여 저장하는 것이 유리한 경우가 있다. 중복할당의 대상이 되는 속성은 조회트랜잭션에서 자주 필요로 하는 속성이고 이러한 속성을 어느 한 단편에만 할당하면 그 단편이 여러 트랜잭션에서 접근되어야 한다. 따라서 필요하다면 속성을 단편에 중복할당하여 수직 분할하는 문제를 중복허용 수직 분할문제라 한다. 갱신트랜잭션의 경우 데이터의 무결성(integrity)을 유지하기 위해 하나의 속성이 갱신되면 중복 할당된 모든 속성을 갱신해야 한다. 따라서 갱신트랜잭션이 필요로 하는 속성을 단편들에 중복할당하면 중복배치속성이 포함된 모든 단편을 접근해야하므로 디스크 접근 횟수가 증가한다. 유[1998]의 실험에서는 갱신트랜잭션의 수가 증가하면 중복속성의 수가 감소되고, 비용 절감율도 감소하는 것으로 나타났다. 그러므로 트랜잭션의 타입에 따라 디스크 접근 비용의 감소율이 다를 수 있다.

관계형 데이터베이스에서 사용하는 스캔방법들에 대해 디스크 접근 횟수를 추정해 보면 아래의 식 (1.1)에서 (1.3)과 같다.

* Segment scan method

$$No.of\ access \approx \frac{(No.\ of\ tuples)\cdot(length\ of\ tuple)}{(page\ size)\cdot(prefetch\ blocking\ factor)} \quad (1.1)$$

* Clustered index scan method

$$No.of\ access \approx \frac{(No.\ of\ tuples)\cdot(selectivity)\cdot(legnth\ of\ tuple)}{(page\ size)} \quad (1.2)$$

* Unclustered index scan method

$$No.of\ access = (No.of\ tuples)\cdot(selectivity) \quad (1.3)$$

위의 식에서 디스크 접근 횟수는 Segment scan 방법과 Clustered index scan 방법에서 터플의 길이에 비례하므로, 본 연구에서는 터플의 길이를 사용하여 디스크의 접근 횟수를 최소로 하는 속성의 중복허용 수직분할문제를 고려한다.

2장에서는 터플의 길이를 고려하여 디스크의 접근 횟수를 최소화시킬 수 있는 수리모형을 제시한다. 3장에서는 유전알고리듬을 이용하여 중복허용 수직분할문제의 발견적 해를 구하는 방법을 제시하고, 4장에서는 유전해법의 실험 및 분석을 수행한다. 마지막으로 5장에서는 결론을 제시한다.

2. 중복을 허용한 수직분할모형

2.1절에서는 사용 기호를 정의하고, 2.2절에서는

속성의 중복을 허용하는 N-ary 수직분할문제의 0-1 정수계획법[유종찬, 김재련 1998] 수리모형을 설명한다.

2.1 사용 기호

속성의 중복을 허용하는 수직분할 문제에서 사용하는 기호와 입력변수 및 결정변수는 다음과 같다.

- 사용기호

a : 릴레이션의 속성 수
t : 릴레이션을 액세스하는 트랜잭션의 수
f : 편의 개수
j : 속성 번호 $(j = 1, \ldots, a)$
I : 트랜잭션 번호 $(i = 1, \ldots, t)$
l : 단편 번호 $(l = 1, \ldots, f)$
R_t : 조회 트랜잭션의 집합
U_t : 갱신 트랜잭션의 집합
N_l : 단편 l의 터플 길이 (byte)
R_i : 조회 트랜잭션 i의 디스크 접근 비용
U_i : 갱신 트랜잭션 i의 디스크 접근 비용
IDL : 키 속성의 길이 (byte)
ATL_j : 속성 j의 길이
CDL : 릴레이션의 터플의 수
W : 갱신트랜잭션의 추가적 디스크 접근 비용을 나타내는 가중치
$Freq_i$: 트랜잭션 i의 단위 시간당 발생 횟수
SEL_i : 트랜잭션 i의 명제 만족율

AUM_{ij} =

$$\begin{cases} 1, & \text{트랜잭션 } i\text{가 속성 } j\text{를 사용하는 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases}$$

UTR_{il} =

$$\begin{cases} 1, & \text{갱신트랜잭션 } i\text{가 단편 } l\text{를 필요로 하는 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases}$$

RTR_{il} =

$$\begin{cases} 1, & \text{조회트랜잭션 } i\text{가 단편 } l\text{를 필요로 하는 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases}$$

- 결정변수

$$X_{jl} = \begin{cases} 1, & \text{속성 } j\text{가 단편 } l\text{에 할당될 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases}$$

중복허용 수직분할문제에 대한 본 연구의 수행척도(performance measure)는 트랜잭션을 처리할 때 필요한 디스크의 접근 비용이다. 릴레이션이 f개의 단편에 속성이 중복허용되어 분할되었다면, 조회트랜잭션 i의 디스크 접근비용은 $R_i = \sum_{l=1}^{f}\{CDL \cdot SEL_i \cdot (N_l + IDL)\} \cdot RTR_{il}$이 된다. 이 식은 트랜잭션 i가 단편 l을 액세스할 때 필요한 디스크 접근 횟수를 계산하는 비용이다(디스크 접근 비용은 (R_i/page size)로 표시하는 것이 더 정확하다. 그러나 이것은 모든 단편에 적용되어 수리모형에서 동일한 효과를 나타내므로 생략하여 표시한다.). 조회트랜잭션인 경우에 사용되는 변수 RTR_{il}는 트랜잭션 i가 속성 j를 필요로 하고 속성 j가 단편 l에 포함되어 있을 때 3.2절의 비용계산 알고리듬에 의해서 가장 적은 디스크 접근비용으로 트랜잭션을 처리할 수 있도록 단편을 선정할 때 사용되는 변수이다. 그러나 갱신트랜잭션의 경우에는 데이터의 무결성을 유지하기 위하여 변수 UTR_{il}를 사용하여 트랜잭션 i가 요구하는 속성 j가 포함된 단편을 모두 액세스 하도록 한다. 또한 단위시간에 발생하는 디스크 접근비용은 $R_i \cdot Freq_i$와 $U_i \cdot Freq_i$가 된다. 따라서 단위시간에 발생하는 모든 트랜잭션의 디스크 접근 비용은 $\sum_{i \in R_t} R_i \cdot Freq_i + \sum_{i \in U_t} U_i \cdot Freq_i$이 된다.

속성의 중복을 허용하여 디스크 접근 비용을 최소화하는 0-1정수 계획모형은 2.2절과 같다.

2.2 수리모형

중복을 허용한 수직분할 문제의 0-1정수계획법의 수리모형은 다음과 같다.

$$Min\ TC = \sum_{i \in R_t} R_i \cdot Freq_i + \sum_{i \in U_t} U_i \cdot Freq_i \quad (2.1)$$

Subject to

$$R_i = \sum_{l=1}^{f}\{CDL \cdot SEL_i \cdot (N_l + IDL)\} \cdot RTR_{il}\ \forall i \quad (2.2)$$

$$U_i = \sum_{j=1}^{f}\{CDL \cdot SEL_i \cdot (N_l + IDL)\} \cdot UTR_{il} \cdot W \quad \forall i \tag{2.3}$$

$$N_l = \sum_{j=1}^{a} ATL_j \cdot X_{jl} \quad \forall l \tag{2.4}$$

$$1 \le \sum_{l=1}^{f} X_{jl} \le f \quad \forall j \tag{2.5}$$

RTR_{il} =

$$\begin{cases} 1, & \text{트랜잭션 } i\text{가 단편 } l\text{를 필요로 하는 경우 (비용계산 알고리듬(3.2절)에서 결정)} \\ 0, & \text{그렇지 않은 경우} \end{cases} \tag{2.6}$$

UTR_{il} =

$$\begin{cases} 1, & \sum_{j=1}^{a} AUM_{ij} \cdot X_{jl} > 0 \quad \text{인 경우} \quad \forall i, l \\ 0, & \sum_{j=1}^{a} AUM_{ij} \cdot X_{jl} = 0 \quad \text{인 경우} \end{cases} \tag{2.7}$$

목적식 (2.1)은 모든 트랜잭션의 디스크 접근 비용의 합이다. 조회트랜잭션과 갱신트랜잭션으로 구성되고, 각 트랜잭션이 접근해야 하는 데이터의 양이 감소하면 총 디스크 접근 비용은 감소한다. 제약식 (2.2)는 조회트랜잭션 i의 디스크 접근 비용을 나타낸다. 제약식 (2.3)은 갱신트랜잭션 i의 디스크 접근 비용을 나타내며, 갱신트랜잭션 i가 요구하는 속성이 포함된 모든 단편을 액세스 하는 비용을 계산한다. 갱신트랜잭션을 처리하기 위해서는 터플을 주기억 장치로 이동하여 수정한 후 이것을 다시 보조기억장치에 보관하는 작업이 필요하다. 그러므로 갱신트랜잭션을 처리할 경우에는 속성을 갱신하는데 필요한 추가적인 디스크 접근 비용이 포함되므로 디스크 접근 비용이 증가한다. 따라서 W는 갱신트랜잭션을 수행하는데 필요한 추가적인 디스크 접근 비용을 나타내는 가중치이다. 일반적으로 W = 2로 하여 푸는 경우가 대부분이다. 제약식 (2.4)는 단편 l의 터플 길이를 나타낸다. 식 (2.5)은 각 단편에 포함시킬 수 있는 속성의 수를 나타내는 식으로 본 연구에서는 중복할당을 고려하므로 단편의 수 만큼 속성을 할당하는 것이 가능하다. 식 (2.6)은 조회트랜잭션 i를 처리하기 위해 필요한 단편 l을 최소의 비용으로 액세스하기 위해 3.2절의 비용계산 알고리듬에 의해 단편을 선정한다. 식 (2.7)은 갱신트랜잭션 i가 단편 l에 할당된 속성들 중에서 적어도 하나의 속성을 필요로 하면 단편 l을 액세스해야 하는 조건이다.

3. 유전알고리듬을 이용한 해법개발

중복허용 수직분할 문제의 크기가 커지면 최적 알고리듬을 적용하여 해를 구하는 데는 많은 시간이 소요된다. 그러므로 본 장에서는 최적해에 가까운 해를 빠른 시간내에 풀 수 있는 발견적기법으로 알고리듬의 해법을 제시한다.

3.1 유전알고리듬의 개념

유전알고리듬(Genetic Algorithm, GA)은 Holland (1975)가 최초로 제안한 방법으로 적응적 탐색(adaptive search)방법에 속하는 일반적인 발견적 해법(meta heuristic)으로써 자연계의 진화과정의 법칙인 적자생존(survival of fitness)원리와 유전정보의 교환을 통한 세대교체의 원리를 문제를 풀기 위한 해법절차로 사용하는 것이다. 유전알고리듬에서는 개별해(single solution)가 아닌 해집단(population of solution)을 처리하고, 해집단의 반복적인 세대교체 과정을 통하여 적응적으로 진화한다.

다음세대의 해집단을 구성하기 위하여 현세대의 해집단으로부터 우수한 유전자, 즉 문제의 목적식을 만족시킬 수 있는 적합도가 큰 개별해들을 확률적으로 선발하여 부모해를 생성하고 이러한 부모해들에 대하여 유전연산(genetic operation)을 사용하여 해의 재조합을 수행하면 자손해가 생성된다. 이렇게 생성된 자손해로 현세대의 해집단을 대치(replace)하므로써 새로운 세대의 해집단을 구성한다. 위의 과정은 한 세대를 교체하는 과정이다. 이러한 세대교체의 과정을 반복적으로 수행하면 해집단은 우수한 해집단으로 수렴해 간다.

유전알고리듬은 개념적으로 단순하며, 반복적인 형태를 갖고 있다. 그러나 효율적인 탐색절차로

사용되고 있는 이유는 탐색원리가 체계적이고 이론적이기 때문이다. 유전알고리듬의 탐색 원리는 스키마 정리(schema theory)로 알려져 있다[Goldberg 1989].

3.2 개별해(FAMs)의 비용계산 방법

본 절에서는 유전해법을 적용할 때 해를 평가하기 위하여 비용계산 알고리듬을 제시한다[유종찬, 김재련 1998]. 하나의 개별해의 비용을 구하기 위해서 필요한 자료는 단편-속성 행렬(Fragment-Attribute Matrix, FAM)로 나타낸다. FAM은 단편에 속성이 배치된 형태를 나타내며, 일차배치 속성과 중복배치 속성을 구분하는데 사용될 수 있다. FAM의 예는 (그림 1)과 같이 0-1 행렬로 되어 있다.

트랜잭션이 요구하는 모든 속성은 단편에 반드시 배치되어야 하며 (그림 1)과 같이 속성들이 단편에 배치되었을 때 속성을 구분하는 방법은 다음과 같다. 일차배치 속성(First-Incident Attribute, FIA)이란 위의 (그림 1)에서 원으로 표시된 속성으로써 단편 1에서 시작해서 처음으로 나타난 속성들을 일차배치 속성이라 하며 다음 단편에 다시 동일 속성이 할당되면 이를 중복배치속성(Replicated-Incident Attribute, RIA)으로 부른다. 그밖의 속성 즉, 단편에 첫 번째로 배치되지 않고 중복되어 배치된 속성을 중복배치 속성이라 하며, 중복배치 속성의 수는 단편의 수를 f라 할 때, (f-1)개씩 할당할수 있으므로, 최대로 (f-1)*a개가 될수 있다. 또한 트랜잭션에서 필요로 하는 속성을 일차배치 속성으로 포함하는 단편들을 후보단편(Candidate Fragment)이라 한다. 한 세대에 속하는 개별해(FAM)의 비용(Cal_N_C)들을 계산하는 절차는 다음과 같다.

● 비용 계산 알고리듬

[단계 0] 트랜잭션의 구분

- If 트랜잭션 i가 조회트랜잭션이면 Then [단계 1]로 간다.
- If 트랜잭션 i가 갱신트랜잭션이면 Then [단계 6]으로 간다.

[단계 1] 일차배치 속성의 탐색

- FAM에서 각 속성(column)에 대하여 단편(row)을 조사하여 첫 번 째로 발생된 단편의 속성을 일차배치속성으로 한다.

[단계 2] 단편의 제거

- 일차배치 속성을 포함하지 않는 단편의 디스크 접근 횟수를 "0"으로 놓는다.

[단계 3] 후보단편의 선정

- 조회트랜잭션 i에서 요구하는 속성 j를 일차배치속성으로 가지고 있는 단편들을 찾는다.

[단계 4] 조회트랜잭션에 의해 액세스되는 단편을 선정

- If 후보단편들 중에서 하나의 단편으로 조회트랜잭션 i에서 필요로 하는 속성 j를 모두 포함하면 ($_{cf}C_1$인 경우)
  Then 그 단편에 조회트랜잭션 i의 디스크 접근 횟수를 저장한다

Frag. \ ATT.	Attribute1	Attribute2	Attribute3	Attribute4	Attribute5	Attribute6
Fragment 1	①		①			
Fragment 2	1	①			①	
Fragment 3	1	1	1	①	1	
Fragment 4		1				①

(그림 1) 속성사용행렬(Fragment-Attribute Matrix)의 예

Else 조회트랜잭션 i에서 필요로 하는 속성 j를 모두 포함하는 단편들의 조합을 찾기 위해서 사용할 단편의 수를 증가시키면서 후보 단편들을 조합한다. ($_nC_r$ (r≥2)인 경우)

If 단편의 조합이 2가지 이상이면

Then 속성의 총 길이가 작은 조합에 포함되는 단편에 조회 트랜잭션 i의 디스크 접근 횟수를 저장한다.

[단계 5] 조회트랜잭션에 대해서 FAM에 속하는 모든단편들의 디스크 접근 횟수계산

- 모든 조회트랜잭션에 대해서 [단계 3]에서 [단계 5]를 반복하고 FAM의 각 단편에 트랜잭션 횟수를 저장한다. 고려되지 않은 조회 트랜잭션이 없으면 [단계 8]로 간다.

[단계 6] 갱신트랜잭션에 의해 액세스 되는 단편을 선정

- FAM에서 갱신트랜잭션 i에서 필요로 하는 속성 j를 포함하는 모든 단편에 갱신트랜잭션 i의 디스크 접근횟수를 저장한다.

[단계 7] 갱신트랜잭션에 대해서 FAM에 속하는 모든단편들의 디스크 접근 횟수계산

- 모든 갱신트랜잭션에 대해서 [단계 6]을 반복하고 고려되지 않은 갱신트랜잭션이 없으면 [단계 8]로 간다.

[단계 8] 가능해(FAM)의 비용 (Cal_N_C)계산

- 단편에 저장된 디스크 접근 횟수와 각 단편의 속성길이를 곱하여 각 단편의 디스크 접근 비용을 구하고, 모든 단편의 디스크 접근 비용을 합하여 하나의 가능해(FAM)에 대하여 비용(Cal_N_C)를 구한다

위의 Cal_N_C는 식 (2.1)의 TC와 동일한 값을 나타낸다.

3.3 유전알고리듬의 요소기법 설계

본 절에서는 속성의 중복을 허용하는 화일 수직분할문제의 해를 발견적인 기법으로 구하기 위하여 유전알고리듬을 사용한다. 유전알고리듬으로 정해진 문제를 해결하기 위한 해법의 개발과정은 각각의 요소기법을 설계한 후 결합하므로써 종합적인 유전알고리듬의 해법이 개발된다. 요소기법으로는 이진 암호화 방법, 초기해집단의 생성방법, 해의 최적성 평가 및 가능해 여부의 판정방법, 유전자의 재조합을 위한 부모해의 선발방법, 유전자의 재조합을 위하여 사용되는 연산자의 선택, 세대교체의 진행방법 및 종료조건이 있다. 수직분할 문제를 풀기 위하여 본 연구에서 제시하는 유전알고리듬의 요소기법은 다음과 같다.

3.3.1 해의 암호화

본 연구에서 수직분할 문제의 개별해, 즉 하나의 가능해는 모든 속성이 단편에 배치된 경우의 단편-속성 행렬(FAM)로 나타낸다. 식 (3.1)과 (3.2)와 같이 정의하는 이진 행렬(binary matrix)의 형태인 FAMs와 같다. 한 세대에 속하는 개별해 FAMs에 대하여 속성이 임의의 단편에 배치된 경우는 1, 그렇지 않은 경우는 0으로 나타낸다.

$$FAM_S = FAM_S(l, j) \quad s = 0, 1, 2, \cdots\cdots, L \quad \forall 1, j \qquad (3.1)$$

$$FAM_S(l, j) = \begin{cases} 1, & \text{속성 } j\text{가 단편 } l\text{에 할당될 경우} \\ 0, & \text{그렇지 않은 경우} \end{cases} \qquad (3.2)$$

3.3.2 초기해 집단의 생성 방법

유전알고리듬에서는 세대교체의 과정을 반복적으로 수행한다. 세대교체의 진행을 나타내기 위하여 세대교체 주기의 시점 t를 세대교체 횟수의 상한 T까지 증가시킨다. 시점 t에서의 해집단 S(t)는 FAMs의 집합으로 구성되며, L은 해집단을 구성하는 개별해의 총 개수이다.

$$S(t) = FAM_S : s = 1, 2, \cdots\cdots, L, \quad t = 0, 1, 2, \cdots\cdots, T \qquad (3.3)$$

초기해집단을 나타내는 S(0)는 전통적인 유전해법에서 무작위로 발생된 해를 사용한다. 그러나

본 연구에서는 해의 수렴속도와 최적성을 모두 향상시키기 위하여 비중복 알고리듬의 발견적 기법[윤병익, 김재련 1996]으로 구해진 해와 무작위로 발생시킨 해를 혼합하여 사용한다. 발견적 기법으로 구해진 비중복 문제의 해를 L_h로 나타낸다. 무작위로 FAMs에 "1"을 발생시킨 해들을 L_r로 나타내고, 비중복 문제의 발견적기법으로 구해진 해를 일차배치속성으로 하여 고정시키고, 중복배치속성에 대하여 무작위로 "1"을 발생시켜 사용하는 해들, 즉 FAMs에서 중복배치속성에 대해서만 무작위적 형태로 단편에 배치하여 생성된 해를 L_{h+r}로 나타낼 때, 초기 해집단 S(0)에 포함되는 개별해들의 총 개수 $L = L_h + L_{h+r} + L_r$로 나타낼 수 있다.

3.3.3 적합도 평가

해의 최적성을 평가하기 위한 방법으로 3.2절에서 제시한 비용계산 알고리듬을 사용한다. 즉, 모든 속성이 할당된 가능해 FAMs들의 디스크 접근 비용을 계산하므로써 해를 평가한다. S(t)에 포함된 모든 개별해들에 대하여 비용계산을 한다. 유전해법에서는 해의 적합도를 평가하기 위하여 비용함수를 평가한 후 최소화 문제를 풀기 위해서는 비음의 적합도 함수로 변환하여야 한다. 해의 적합도를 결정하는 방법에 따라 유전해법의 효율성에 크게 영향을 주기 때문에 바람직한 탐색성능을 유도하기 위하여 적합도의 조절방법으로 사용되는 것은 첫째로, 적합도의 척도(scaling)를 조절하는 방법과 둘째로, 순위(ranking)에 근거하여 적합도 함수를 만들어내는 방법이 있다.

본 연구에서는 위에서 제시한 방법을 모두 사용한다. 개별해들의 비용을 계산하여 순위를 결정하고 적합도 함수의 척도 계수로써 f_{max}와 f_{min}을 사용하여 식 (3.4)와 같이 적합도 함수 f(s)를 결정한다. 여기서 f_{max}와 f_{min}은 매개변수로써 해들의 적합도 차이를 스케일링하는 변수이다. R(s)는 해집단에 속한 개별해들의 비용계산 순위이다.

$$f(s) = f_{max} - (f_{max} - f_{min}) \frac{R(s)-1}{L-1} \qquad (3.4)$$

유전해법에서 세대교체가 진행됨에 따라 초기해의 해집단은 우수한 해집단으로 수렴해 간다. 해집단의 진화과정에서 빈번하게 동일한 적합도를 갖는 해가 생성된다. 이러한 동일한 적합도를 갖는 해들의 중복현상을 감소시키기 위하여 Goldberg가 제안한 적합도 공유(fitness sharing)의 개념을 적용한다. 동일한 적합도 (f(s))값을 갖는 d개의 해에 대하여 공유적합도 함수 $f_{fs}(G)$는 식 (3.5)와 같다.

$$f_{fs}(s) = \frac{f(s_k)}{d} \quad k = 1, 2, \ldots\ldots, d \qquad (3.5)$$

유전자의 재조합에 참여할 부모해를 선발하기 위하여 각 개별해들간의 상대적인 적합도에 근거한 우수한 해들을 선택해야 한다. 이를 위한 상대적 적합도 함수 $f_{rf}(s)$는 식 (3.6)과 같다.

$$f_{rf}(s) = \frac{f_{fs}(s)}{\sum_{s=1}^{L} f_{fs}(s)} \quad s = 1, 2, \ldots\ldots, L \qquad (3.6)$$

3.3.4 부모해의 선발과 복제

현 세대의 해집단으로부터 자손해를 생성하는 과정으로 상대적 적합도($f_{rf}(s)$)에 근거하여 유전자의 재조합과정에 참여할 부모해를 확률적으로 선발한다. 본 연구에서는 부모해를 선발하는 샘플링 기법으로 최근 가장 널리 사용되고 있는 추계적 비복원 잔여 샘플링(stochastic reminder sampling without replacement)방법을 사용한다. 선발된 해는 모든 유전정보를 복제하여 재조합에 참여시킨다.

3.3.5 재조합을 위한 연산

확률적으로 선발된 부모해는 자손해를 생성하기 위하여 유전자 재조합 과정을 수행한다. 본 연구에서는 부모해를 결합하여 유전정보를 재조합하기 위하여 상호교차(crossover)연산을 사용하고, 보조연산으로 돌연변이(mutation)연산을 사용한다.

상호교차 연산은 부모해로부터 상호교차 연산 적용율에 따라 선발된 $FAM_1(l, j)$와 $FAM_2(l, j)$의 유전암호를 식 (3.7)와 (3.8)에 의하여 자손해 $FAM_1'(l, j)$와 $FAM_2'(l, j)$를 생성한다. 식에서

P_c는 상호교차의 적용율, l (l = 1, ... f)은 단편 번호, j (j = 1, ... a)는 속성 번호, G_c는 상호교차 연산의 기준점이다.

$FAM_1'(l, j)$ =

$$\begin{cases} FAM_1(l, j), & a \cdot (l-1)+j \le G_c \\ FAM_2(l, j), & a \cdot (l-1)+j > G_c \end{cases} \quad (3.7)$$

$FAM_2'(l, j)$ =

$$\begin{cases} FAM_2(l, j), & a \cdot (l-1)+j \le G_c \\ FAM_1(l, j), & a \cdot (l-1)+j > G_c \end{cases} \quad (3.8)$$

돌연변이 연산은 식 (3.9)와 같고 자손해를 구성하는 모든 개별해들의 이진수 값, 즉 bit의 총 개수를 대상으로 적용율 P_m의 값에 따라 수행된다. 따라서 해집단 전체에 대하여 돌연변이연산의 대상이 되는 유전인자의 총 개수는 a * f * L * P_m이 된다. 돌연변이 연산을 수행할 유전인자의 개수가 결정되면 무작위적으로 유전인자를 선택하여 (3.9)식에 의하여 연산을 수행한다.

$$FAM_s(l, j) = \begin{cases} 1, & FAM_s(l, j) = 0 \\ 0, & FAM_s(l, j) = 1 \end{cases} \quad (3.9)$$

3.3.6 세대교체의 진행방법과 종료조건

본 연구에서 사용하는 세대교체의 진행방법은 현재 세대에서 가장 우수한 해는 다음세대에도 항상 유지되도록 보장하는 엘리트 주의(elitism)방법을 사용하고, 종료조건은 세대교체 횟수 상한을 결정하는 방법과 일정 횟수만큼의 세대교체 동안 우수한 해가 발견되지 않는 경우 유전 알고리듬을 종료하는 방법을 모두 사용하여 두 조건 중 하나만 만족되어도 알고리듬을 종료시키는 방법을 사용한다.

3.4 유전 알고리듬의 절차

3.3절에서 제시한 요소기법을 종합하여 속성의 중복을 허용하는 수직분할 문제의 최소비용을 구하는 유전알고리듬의 절차를 나타내면 다음과 같다. 해법절차에서 $FIT_{best}(t)$는 시점 t까지 나타난 모든 해들 중에서 가장 적합도가 우수한 해이며, $FIT_{max}(t)$는 시점 t에서 발견된 가장 열등한 해를 나타낸다.

● 유전해법 알고리듬

[단계 0] 초기화

t = 0으로 놓는다.
식 (3.1)과 (3.2)에서 정의하는 L개의 초기해집단 S(t)를 구성한다.

[단계 1] 개별해(FAM_S)의 비용계산

3.2절에서 제시한 비용계산 알고리듬을 사용하여 S(t)를 구성하는 모든 개별해들의 비용을 계산한다.

[단계 2] 상대적 적합도로 변환

개별해들의 비용계산 값에 따라 순위(R(s))를 결정하고 식 (3.4), (3.5), (3.6)에 따라 f(s), $f_{fs}(s)$를 계산하고, 상대적 적합도 $f_{rf}(s)$를 구한다.

[단계 3] 적합도가 우수한해의 존속

엘리트 주의의 구현방법으로, $FIT_{max}(t)$가 $FIT_{best}(t)$보다 열등하면, $FIT_{best}(t)$로 $FIT_{max}(t)$를 교체한다.

[단계 4] 자손해의 생성

* [단계 4-1]에서 [단계 4-4]는 새로운 자손해의 생성과정이다.

[단계 4-1] 부모해의 선발

$f_{rf}(S)$에 따라 추계적 비복원 잔여 샘플링 기법을 사용하여 부모해를 선발한다.

[단계 4-2] 부모해의 복제

적합도에 따라 선발된 부모해의 유전자를 복제한다.

[단계 4-3] 상호교차

복제된 해들을 대상으로 상호교차 적용

비율 P_c에 따라 무작위로 FAM_s들을 선발하여 쌍(pair)을 지어 식 (3.7)와 (3.8)와 같이 상호 교차 연산을 수행한다.

[단계 4-4] 돌연변이

복제된 해들을 대상으로 돌연변이 적용비율 P_m에 따라 무작위로 유전 인자를 선발하여 식 (3.9)와 같이 돌연변이 연산을 수행한다.

[단계 5] 세대교체와 종료조건

- t = t + 1로 놓는다
- 복제와 재조합 과정을 통하여 생성된 자손해(S(t+1))로 현세대의 집단해 S(t)를 교체한다.
- 반복횟수의 상한 T에 도달하거나 일정 횟수 동안 우수한 해로 갱신되지 않으면 종료한다. 그렇지 않으면, [단계 1]로 간다.

3장에서는 중복허용 수직분할 문제를 풀기 위한 유전해법의 요소기법과 알고리듬절차를 제시 하였다. 4장에서는 유전해법을 이용하여 실험을 수행한다.

4 실험 및 분석

본 장에서는 속성의 중복을 허용하는 수직분할 문제를 풀기 위해 설계된 유전 해법절차의 성능을 분석하기 위하여 4.1절에서는 문제의 크기가 각각 다른 5개의 문제를 선택하여 실험을 수행한다. 이 실험의 목적은 실험에 사용된 5개의 문제에 대하여 평균적으로 해의 탐색성능이 우수한 매개변수의 수준 조합을 찾는 것이다.

4.2절에서는 4.1절의 실험에 의해 얻어진 해의 탐색성능이 가장 우수한 매개변수 수준 조합값으로 유전해법의 매개변수를 설정하여 구한 해를 분지한계법으로 구한 중복허용, 비중복 최적해 및 비중복 발견적해와 비교한다. 본 장에서 문제에 사용되는 모든 속성사용행렬(Attribute Usage Matrix, AUM)에서 트랜잭션은 모두 조회 트랜잭션으로 하였으며, 키 속성의 길이 IDL = 4, 트랜잭션 i의 명제 만족율 SEL_i = 1, 릴레이션의 터플 수는 CDL = 1 (CDL (cardinality)의 값은 모든 단편에서 동일하고 해를 구하는 과정에 영향을 미치지 않으므로)로 하였다.

알고리즘을 시작할 때 단편의 수(f)는 속성의 수(a)와 동일하게 놓는다. 알고리듬의 수행 후에 속성이 배치되지 않은 단편과 중복배치속성만으로 구성된 단편이 존재하게 되므로 이러한 단편들은 제외한다. 따라서 알고리듬 초기에 단편의 수 f를 큰 값으로 하여 풀어도 최적 단편 수 f*개를 제외한 나머지 단편에는 속성이 할당되지 않는다[유종찬, 김재련 1998].

본 연구에서 제안한 유전해법의 성능실험을 위해 알고리듬을 C/C++로 프로그램 하였으며, Pentium 150 Mhz 기종의 PC에서 수행하였다.

4.1 유전해법의 매개변수 최적화

4.1.1 매개변수의 선택

유전해법의 탐색성능은 매개변수값에 따라 차이가 있다. 본 연구에서 제시하는 유전해법의 매개변수는 해 집단의 크기(L), 세대 교체 횟수의 상한(T), 적합도의 척도 조절상수 $F = (F_{max}, F_{min})$, 상호교차 연산자의 적용비율(P_c), 돌연변이 연산자의 적용비율(P_m), 발견적해의 혼합율(P_h), 발견적 기법의 해[윤병익,김재련 1996]를 일차배치속성으로 단편들에 할당하고 중복배치속성을 무작위로 단편들에 할당한 개별해(L_{h+r})들과 모든 속성을 무작위로 발생시킨 개별해(L_r)들, 즉 L_{h+r}과 L_r는 3.2절의 비용계산 방법에서 설명한 FAM_s에서 "1"의 발생 밀도이다.

본 연구에서 해 집단의 크기 L = 100, 세대교체의 상한 T = 2000으로 하였다.

4.1.2 매개변수 최적화를 위한 실험

유전해법에서 매개변수 수준의 최적화에 관한 연구는 De Jong[1975]에 의하여 처음 시도 되었고, Greffenstte[1986]에 의해 유전해법의 최적 매개변

수 수준 조합을 찾기 위해 유전해법을 이용하여, De jong의 연구를 개선 하였다. 또한 Schaffer et al.[1989]은 실험계획법적인 접근방법으로 매개변수 최적화 작업을 수행하였다. Davis[1991]에 의하면 유전해법의 절차가 수정되면 위의 선행 연구에서 얻어진 최적매개 변수 값을 동일하게 사용할 수 없다. 본 연구에서 제시하는 유전해법은 전통적인 유전해법과는 다른 기법들을 사용하고 있다. 따라서 전통적인 유전해법에서 사용되고 있는 매개변수의 값을 사용할 수 없고, 새로운 매개변수의 수준조합을 찾아야 한다.

기존연구와 예비적인 실험을 통하여 각 매개변수 수준의 흥미영역에 대한 실험계획 수준은 <표 1>과 같다. <표 1>과 같이 실험에서 고려되는 매개변수의 수는 모두 6개이고 해집단에 포함되는 개별해의 개수(L)와 세대교체의 횟수(T)는 각각 L = 100, T = 2000으로 고정하였다. 각각의 매개변수에 대한 수준은 두 가지씩 설정하였다. 매개변수 최적화 실험에 사용되는 매개변수의 모든 수준 조합을 나타내면 <표 3>과 같다. <표 3>은 64개의 수준조합이 있으나 일부분만을 나타내었다. 마찬가지로 <표 4>~<표 7>까지도 일부분만을 나타내었다. 실험은 각 수준조합에 대하여 <표 2>와 같

〈표 1〉 매개변수 최적화 실험에 사용되는 실험 수준

척도조절 상수(F)	Crossover Rate(P_c)	Mutation Rate(P_m)	발견적 해의 혼입율(P_h)	L_{h+r}/L	L_r/L
(205, 95)	1.0	0.002	0.2	0.4	0.4
(275, 25)	0.8	0.005	0.4	0.6	0.6

〈표 2〉 매개변수 최적화 실험에 사용되는 문제

Prob. No.	1	2	3	4	5
Size of Prob (transaction *attribute)	6*6	8*8	10*10	12*12	14*14

이 문제의 크기가 다른 5개의 문제에 대하여 각각 3회씩 반복하여 수행하였다. 따라서 총 실험의 횟수는 64*5*3 = 960회이다.

<표 4>는 각 문제들에 대하여 64개의 수준조합 값들을 3회 반복 수행하여 얻어진 결과의 평균값들을 나타낸다. <표 5>는 <표 4>와 같이 실험결과로 얻어진 평균값을 오름차순으로 순위에 의하여 정열한 것이다. 즉 유전해법에 의해서 구해진 해를 각 문제에 따라 실험결과가 우수하게 나타낸 매개변수 수준 순으로 정리한 것이다. 실험에 의해 <표 5>에서 나타난 것과 같이 실험 대상 문제에 따라 가장 우수한 탐색성능을 나타내는 수준조합이 다르다는 것을 알 수 있다.

4.1.3 최적 매개변수 수준의 결정

4.1.2절의 실험에 의해 문제에 따라 가장 우수한 탐색성능을 나타내는 매개변수의 수준조합은 각기 다르다. 본 연구에서 제시한 유전해법을 일반적인 문제의 해법으로 제안하기 위해 실험에 사용된 모든 문제에 대하여 평균적으로 우수한 성능을 나타내고 있는 수준조합을 결정해야 한다. 이러한 수준 조합을 찾아내기 위하여 각 문제들에 대하여 실험결과값이 나타내는 분포가 서로 다르므로, 중

〈표 3〉 매개변수의 수준 조합

수준번호	척도조절 상수(F)	Crossover Rate(P_c)	Mutation Rate(P_m)	발견적 해의 혼입율(P_h)	L_{h+r}/L	L_r/L
1	(205, 95)	1.0	0.002	0.2	0.4	0.4
2	(205, 95)	1.0	0.002	0.2	0.4	0.6
⋮	⋮	⋮	⋮	⋮	⋮	⋮
62	(275, 25)	0.8	0.005	0.4	0.4	0.6
63	(275, 25)	0.8	0.005	0.4	0.6	0.4
64	(275, 25)	0.8	0.005	0.4	0.6	0.6

〈표 4〉 각 매개변수의 수준 조합으로 3회 반복 실험한 해의 평균값

Level \ Prob.no.	1	2	3	4	5
1	9740.33	19791.33	21871.00	23787.67	60521.00
2	9887.33	20098.00	22150.67	24084.67	59409.67
3	9664.00	19764.67	21845.33	23801.00	60076.33
⋮	⋮	⋮	⋮	⋮	⋮
63	9666.67	19708.67	21793.33	24051.00	60116.33
64	9662.67	19542.00	21280.00	24097.33	60516.00
평균	9749.95	19890.08	21826.24	24190.24	60125.85
표준편차	93.19	232.22	271.50.	249.81	813.80

〈표 5〉 각 문제들에 대한 수준 조합별 탐색성능 순위

Ranking \ Prob.no.	1	2	3	4	5
1	13	25	8	18	50
2	58	43	37	47	35
3	26	14	64	1	52
⋮	⋮	⋮	⋮	⋮	⋮
63	34	53	28	31	13
64	42	54	39	29	16

심극한 정리를 이용하여 이들 결과값을 정규화 한다. <표 6>은 <표 4>의 실험결과의 평균값을 표준 정규화한 수치로 나타낸 것이고, 이러한 표준 정규화된 수치를 이용하여 각 수준조합에서의 실험결과를 문제 상호간에 비교할 수 있다. 문제에 따라서 각기 다른 결과값의 분포를 정규화 하기 위해 본 연구에서는 매개변수 수준들에 따른 실험결과의 관측값을 $Ex(i, j)$ ((i= 1, 2, 3, · · · · · , 64), (j= 1, 2, · · · · · · , 5))로 정의하고, 평균값 및 표준편차를 각각 $Aver(j)$와 $STD(j)$로 정의 한다. $Ex(i, j)$, $Aver(j)$, $STD(j)$에 따라 식 (4.1)과 같이 각 수준조합과 문제에 대하여 표준 정규화 변수 $Norm(i, j)$를 구한다.

$$Norm(i,j) = \frac{Ex(i,j) - Aver(j)}{STD(j)} \qquad (4.1)$$

$Norm(i, j)$는 수준조합 i로, 문제 j에 대하여 탐색한 결과값 $Ex(i, j)$를 표준정규화한 값이다. 즉, 문제 j에 대한 모든 수준조합의 평균결과값 $Aver(j)$와 비교하여 $Ex(i, j)$가 어느 정도의 탐색성능을 나타내는가를 표준편차 $STD(j)$에 대한 상대적인 크기로 나타낸다. 그러므로 $Norm(i, j)$가 "0"에 가까울수록 문제 j에 대한 수준조합 i의 탐색성능이 평균값에 근사하고, 본 연구는 최소화 문제이므로 수치가 적을수록 다른 수준조합에 비해 탐색성능이 우수함을 나타낸다.

<표 6>은 각 문제와 수준조합에 따라 Norm(i, j)의 값과 평균값을 나타낸다. 또한 <표 7>에서는

〈표 6〉 반복실험에서 얻어진 평균값을 표준 정규화한 값과 평균

Level \ Prob.no.	1	2	3	4	5	$Norm(i, j)$의 평균
1	-0.10	-0.43	0.16	-1.61	0.49	-0.30
2	1.47	0.90	1.19	-0.42	-0.88	0.45
3	-0.92	-0.54	0.07	-1.56	-0.06	-0.60
⋮	⋮	⋮	⋮	⋮	⋮	⋮
62	0.25	-0.41	-1.66	-0.31	-0.05	-0.44
63	-0.89	-0.78	-0.12	-0.56	-0.01	-0.47
64	-0.94	-1.50	-2.01	-0.37	0.48	-0.87

〈표 7〉 표준 정규화 수치의 평균으로 순위 정렬한 매개변수의 수준조합 번호

순위	수준번호	$Norm(i,j)$의 평균	순위	수준번호	$Norm(i,j)$의 평균
1	18	-0.01	33	44	0.00
2	58	-0.96	34	49	0.02
3	64	-0.87	35	22	0.04
⋮	⋮	⋮	⋮	⋮	⋮
30	21	-0.04	62	39	0.82
31	41	-0.03	63	28	0.98
32	55	-0.01	64	29	1.09

표준 정규화된 수치의 평균값을 오름차순으로 정리한 것이다. <표 7>에 따라서 본 연구에서 제시한 유전해법의 탐색성능이 가장 우수한 수준조합은 18번이다. 이 수준 조합의 매개변수값은 각각 적도계수 F = (205, 95), 상호교차연산 적용율 P_c = 0.8, 돌연변이 연산 적용율 P_m = 0.002, 발견적해의 혼합율 (P_h) = 0.2, L_{hr} = 0.4, L_r = 0.6으로 나타났다.

4.2 최적매개변수 수준에서의 실험

4.1절에서는 중복허용 수직분할문제를 풀기 위한 유전해법의 매개변수들에 대한 최적 수준조합을 결정하였다. 본 절에서는 각 매개변수의 값을 4.1.3절에서 제시한 수준조합 (F = (205, 95), P_c = 0.8, P_m = 0.002, P_h = 0.2, L_{hr} = 0.4, L_r = 0.6)으로 설정하고, 문제의 크기가 다른 12개의 문제를 임의로 만들어 기존의 분지한계법을 이용하여 구한 중복허용 최적해, 비중복 최적해 및 발견적 기법을 사용하는 비중복 문제의 발견적해[윤병익, 김재련 1996]와 비교 실험을 수행한다. 실험에 사용되는 문제의 크기와 문제 번호는 아래의 <표 8>에 나타나 있다.

<표 8> 실험에 사용되는 문제의 번호와 크기

Prob. no.	1,2	3,4	5,6	7,8	9,10	11,12
Size of Prob	5*5	6*6	8*8	10*10	12*12	14*14

실험에서 매개변수인 해집단의 크기 L = 100, 세대교체 횟수의 상한 T = 2000으로 설정하고, 1000번의 세대교체 과정 동안 해의 개선이 나타나지 않으면 해법절차를 종료하도록 설정한다. 유전해법에 의해 구해지는 해들은 알고리듬을 수행할 때마다 나타나는 해의 값이 다를 수 있으므로, 각 문제에 대하여 유전해법 절차를 5회 반복하여 가장 우수한 해를 기존연구의 해와 비교 하였다. 실험결과는 <표 9>에 나타나 있다. <표 9>의 수치는 화일을 수직분할하지 않은 경우와 비교한 비중복, 중복허용 최적해, 비중복 발견적 해와 중복허용 유전해법을 사용한 해의 비용 절감율을 나타낸다. <표 9>에서 분지한계법을 사용하는 비중복, 중복허용 문제의 경우 문제의 크기에 따라 계산량이 지수적으로 증가한다. 따라서 문제의 크기가 큰 문제는 과다한 시간이 소요되므로 실험에서 제외하였다.

<표 9>에서 나타난 것과 같이 과다한 시간 소요로 풀 수 없는 분지한계법을 이용한 비중복, 중복허용의 최적해를 본 연구에서 제시한 유전해법으로 유효한 시간내에 풀 수 있었으며 작은 문제의 경우 최적해와 차이가 없는 것으로 나타났다. 또한 문제에 따라서 비용절감율이 다소 차이가 날 수 있으나, 비중복 알고리듬의 발견해법와 비교하여 비용절감율이 7번 문제에서 최대 10.68%, 12번 문제에서 최소 1.57%로 평균 5.92%로 나타났다.

<표 9> 기존연구의 알고리듬과의 비용절감율(%) 비교(수직 분할하지 않았을 때와의 비용비교)

Algorithm / Prob.no.	(1)비중복 최적해의 경우	(2)비중복 발견적해의 경우	(3)중복허용 최적해의 경우	(4)중복허용 유전해법의 경우
1	68.06%	68.06%	71.10%	71.10%
2	64.74%	64.74%	68.82%	68.82%
3	27.87%	27.87%	·	34.83%
4	39.32%	39.32%	·	45.59%
5	33.47%	32.54%	·	42.49%
6	36.00%	36.00%	·	41.00%
7	·	22.54%	·	33.23%
8	·	32.35%	·	36.00%
9	·	29.50%	·	39.41%
10	·	38.47%	·	42.52%
11	·	32.53%	·	38.39%
12	·	48.07%	·	49.64%

비용 절감율을 수식으로 표현하면 다음과 같다.

(1) 비중복 최적해 경우의 비용 절감율(%) = ((A-비중복 최적해)/A)*100
(2) 비중복 발견적해 경우의 비용 절감율(%) = ((A- 비중복 발견적해)/A)*100
(3) 중복허용 최적해 경우의 비용 절감율(%) = ((A-중복허용 최적해)/A)*100
(4) 중복허용 유전해법 경우의 비용 절감율(%) = ((A-중복허용 유전해법 해)/A)*100
- A : 분할전 비용, 즉 원래의 릴레이션을 분할하지 않고 사용할 때의 비용임

5 결 론

트랜잭션은 릴레이션의 모든 속성을 필요로 하지 않으므로 릴레이션을 수직 분할하여 단편으로 저장하므로써 트랜잭션을 처리할 때 가장 적은 수의 단편만을 접근하여 디스크 접근 횟수를 최소화 시킬 수 있다.

본 연구에서는 속성의 중복을 허용하여 디스크의 접근 횟수를 감소시킬 수 있는 중복허용 수직분할모형을 제시하고, 이 문제의 최적해를 구하는 경우 지수적으로 증가하는 계산량으로 인해 큰 문제를 풀기 어려운 단점이 있으므로 발견적 기법인 유전해법을 이용하여 유효한 시간내에 좋은해를 얻을 수 있는 방법을 제안 하였다.

유전해법의 탐색성능을 향상시키고 문제의 크기에 관계없이 중복허용 수직분할 문제에 적용할 수 있는 일반화된 매개변수의 수준조합을 찾기 위하여 매개변수의 최적화 작업을 수행하였다. 실험대상으로 선정된 모든 문제에 대하여 평균적으로 가장 우수한 탐색 성능을 나타낸 수준조합은 F = (205, 95), P_c = 0, P_m = 0.002, P_h = 0.2, L_{hr} = 0.4, L_r = 0.6로 설정한 경우였다. 매개변수 최적화 실험의 결과로 구해진 위의 매개변수 값으로 설정한 상태에서의 탐색성능을 12개의 문제에 대하여 기존 연구의 해법과 비교하였다. 중복허용 유전해법이 중복허용 문제의 최적해와는 속성의 수가 작은 문제에서는 동일한 결과를 나타내었으며, 비중복 발견적 해와는 비용면에서 평균 6%의 절감효과가 있었다.

참 고 문 헌

[1] 유종찬, 김재련, "속성의 중복을 허용한 화일 수직분할 방법," 「한국 데이터 베이스 학회」, 제4권, 2호, 1998, pp.43-19.

[2] 윤병의, 김재련, "관계형 데이터베이스의 성능향상을 위한 수직분할방법", *한양 대학교 박사학위논문*, 1996.

[3] 윤병익, 김재련, "데이터베이스의 물리적설계에서 분지한계법을 이용한 N-ary수직분할문제", 「대한산업공학회」, 제22권, 4호, 1996, pp. 567-578.

[4] Bethke, A. D., "Genetic Algorithms as Function Optimizers," Doctoral Dissertation, University of Michigan, Ann Arbor, Michigan, 1981.

[5] Booker, L. B., "Improving the Performance of Genetic Algorithms in Classifier Systems," *Proceedings of the International Conference on GAs and Their Applications*, 1985, pp. 80-92.

[6] Brindle, A., "Genetic Algorithms for Functions Optimization," *Doctoral Dissertation, University of Alberta, Edmonton*, 1981.

[7] Ceri, S., S. Navathe and G. Wiederhold, "Distribution design of logical database schema," *IEEE Trans. Software Eng.*, No1. SE-9, No.4, 1983, pp.487-503.

[8] Cheng, C., "Algorithm for vertical partitioning in database physical design,"*Omega, Int. J. Mgnt. Sci.*, Vol.22, No.3, 1994, pp.291-303.

[9] Chu, W. W and I. T. Ieong, "A Transaction-based approach to vertical partitioning for relational database systems," *IEEE Trans. on Software.*, Vol.19, No.8, 1993, pp. 804-812.

[10] Cornell, D. W. and P. S. Yu, "An effective approach to vertical partitioning for physical design of relational database," *IEEE Trans. Software Eng.*, Vol.16, No.2, 1990, pp.248-258.

[11] Cornell, D. W. and P. S. Yu, "Vertical partitioning algorithm for relational databases," *in Proc. 3th IEEE Data Eng.*, 1987.

[12] Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[13] Dewan, R. M. and B. Gavish, "Model for the combined logical and physical design of database." *IEEE Transaction on Computers*, Vol.38, No.7, 1989, pp.955-967.

[14] De Jong, K. A., "An Analysis of the Behavior of a class of Class Genetic Adaptive Systems," Doctorial Dissertation, University of Michigan, Ann Arbor, Michigan, 1975.

[15] Gen, M and Cheng, R., *Genetic Algorithms and Engineering Design*, John Wiley, 1996.

[16] Goldberg, D. E., *Genetic Algorithms in search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.

[17] Goldberg, D. E., "Genetic Algorithms and Rule Learning in Dynamic System Control," *Proceedings of the International Conference on GAs and their Applications*, pp.8-15, 1985.

[18] Goldberg, D. E. and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Functions Optimization," *proceedings of the 2nd International Conference on GAs and their Applications*, 1987, pp.41-49.

[19] Grefenstette, J. J., "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, man, and Cybernetics*, Vol.SMC-16, No.1, 1986, pp.122-128.

[20] Hammer, M. and B. Niamir, "A heuristic approach to attribute partitioning," in Proc. ACM-SIGMOD International Conf, On Management of data, 1979.

[21] Hoffer, J. A. and D. G. Severance, "The use of cluster analysis in physical database design," *in Proc. First VLDB*, 1975.

[22] Hoffer, J. A., "An integer programming formulation of computer database design problems," *Inform. Sci.* Vol.11, 1976, pp.29-48.

[23] Holland, J. H., *Adaptation in Natural and the Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.

[24] Holland, J. H., "Genetic Algorithms and the Optimal Allocations of Trials," *SIAM Journal of Computing*, Vol.2, No.2, 1973, pp.88-105.

[25] Holland, J. H., "Schemata and Intrinsically Parallel Adaptation," *Proceedings of the NSF Workshop on Learning System Theory and its Applications*, 1973, pp.43-46.

[26] Hwang, H. and Sun, J. U., "A Genetic-Algorithm-Based Heuristic for The GT Cell Formation Problem," *Computers ind. Eng*, Vol.30, No.4, 1996, pp.941-955.

[27] Kusiak, A. And C. H. Cheng, "A branch-and-bound algorithm for solving the grouping technology problem," *Annals of Operations Research*, Vol.26, 1990, pp.415-431.

[28] Kusiak, A. And W. S. Chow "An efficient cluster identification algorithm," *IEEE Trans. on Syst., Man and Cybern.*, Vol.SMC-17, No.4, 1987, pp.696-699.

[29] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, Springer-Verlag Berlin Heidelberg, 1992.

[30] Navathe, S., K. Kamalakav, and M. Y. Ra, "A Mixed fragmentation methodology for the initial distributed database design," *College of Computing Georgia Institute of Technology*, 1992.

[31] Navathe, S., S. Ceri, G. Wiederhold, and J.

Dou, "Vertical partitioning algorithm for database design," *ACM Trans. Database Syst.*, Vol.9, No.4, 1984, pp.680-710.

[32] Navathe, S. and M. Ra, "Vertical Partitioning for database design : A graphical algorithm," *in Proc. ACM SIGMOD Int. Conf. Management Data*, 1989.

[33] Ra, M., "A graph-based horizontal partitioning algorithm for distributed of database design," *Journal of the Korea Information Science Society*, Vol.19, No.1, 1992.

[34] Schaffer, J. D., R. A. Caruna, L. J. Eshelman, and R. Das, "A Study of Control Parameters Affecting On-line performance of Genetic Algorithms for Function Optimization," *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989.

■ 저자소개

김 재 련

서울대학교 공과대학 전자공학과를 졸업하고, Oregon State Univ.에서 산업공학 석사, 그리고 Virginia Tech.에서 산업공학 박사학위를 취득하였다. 현재 한양대학교 공과대학 산업공학과 교수로 재직하고 있으며 주 관심분야는 MIS, 데이터베이스 및 전문가시스템이다.

유 종 찬

공동저자 유종찬박사는 한양대학교 산업공학과를 졸업하고, 동대학원에서 산업공학 석사, 산업공학 박사학위를 취득하였다. 현재 LG 생산 기술원에 재직 중이다.