비대칭 외판원 문제를 위한 새로운 분지기법
- New Branching Criteria for the Asymmetric Traveling Salesman Problem -

지 영 근*
G, Young Gun
강 맹 규*
Kang, Maing Kyu

Abstract

Many algorithms have been developed for optimizing the asymmectric traveling salesman problem known as a representative NP-Complete problem. The most efficient ones of them are branch and bound algorithms based on the subtour elimination approach. To increase efficiency of the branch and bound algorithm, number of decision nodes should be decreased. For this the minimum bound that is more close at the optimal solution should be found or an effective bounding strategy should be used.

If the optimal solution has been known, we may apply it usefully to branching. Because a good feasible solution should be found as soon as possible and have similar features of the optimal solution. By the way, the upper bound solution in branch and bound algorithm is most close at the optimal solution. Therefore, the upper bound solution can be used instead of the optimal solution and information of which can be applied to new branching criteria.

As mentioned above, this paper will propose an effective branching rule using the information of the upper bound solution in the branch and bound algorithm. And superiority of the new branching rule will be shown by comparing with Bellmore-Malone's one and Carpaneto-Toth's one that were already proposed.

1. 서론

외판원 문제(traveling salesman problem)는 한 외판원이 본사를 출발하여 그의 고객이 있는 모든 도시를 두 번 이상 방문하지 않고 반드시 한 번은 방문하고 본사로 돌아오는 최소비용의 순환로를 찾는 문제이다[1]. Dantzig[6]에 의해 제안된 외판원 문제의 수리모형은 다음과 같다. c_{ij}는 도시 i에서 j로 가는 비용이고, 호(i, j)가 사용되었을 때 x_{ij} = 1, 사용되지 않았을 때 x_{ij} = 0이라 한다.

* 한양대학교 산업공학과

최소화 $$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{1.1}$$

제약조건

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j=1, \ldots, n, \tag{1.2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i=1, \ldots, n, \tag{1.3}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S|-1, \ \forall S \subset \{1, 2, \ldots, n\}, \ S \neq \phi, \tag{1.4}$$

$$x_{ij}\in\{0, 1\}, \ i, j = 1, \ldots, n. \tag{1.5}$$

식(1.1)은 도시를 방문하는 비용이 최소가 되어야 한다는 목적식이다. 식(1.2)와 식(1.3)은 각 도시를 반드시 한 번만 방문해야 한다는 조건이고, 식(1.4)는 하나의 헤밀톤 순환로(Hamiltonian path)여야 한다는 조건이다. 비대칭 외판원 문제(asymmetric traveling salesman problem)란 서로 다른 두 도시 i, j $(i \neq j)$에 대하여 도시 i에서 j로 가는 비용 c_{ij}와 도시 j에서 도시 i로 가는 비용 c_{ji}가 서로 다른 외판원 문제이다. 비대칭 외판원 문제는 현재까지 다항식 계산량을 갖는 알고리듬이 알려져 있지 않은 대표적인 NP-Complete 문제이다. 비대칭 외판원 문제의 최적해를 찾기위해 많은 알고리듬이 개발되어 왔는데, 그 중 가장 효율적인 것들은 부분순환로 제거법(subtour elimination approach)에 바탕을 둔 분지한계법(branch and bound algorithm)이다[3,5].

분지한계법은 각 단계에서 최적해가 포함되어 있을 가능성이 가장 커 보이는 부분집합을 선정하여, 이 부분집합을 다시 두 개 이상의 부분문제로 분지(branch)한 다음, 분지된 부분문제의 하한(minimum bound)을 계산한다. 만약 하한을 구하는 중에 최적해가 나오면 끝낸다.

분지한계법에서 하한을 구하는 방법은 원문제의 제약을 완화시킨 문제를 푸는 방법을 사용한다. 분지한계법으로 외판원 문제를 풀 때 하한을 구하는 방법으로는 외판원 문제를 완화시킨 배정문제(assignment problem)가 잘 알려져 있다[2]. 배정문제는 위의 수리모형에서 식(1.4)가 없는 문제이다.

최근에는 이러한 분지한계법을 이용하여 500,000 여개의 도시를 갖은 문제를 풀였음이 보고되어 있다[8]. 그러나 분지한계법은 기본적으로 열거법이기 때문에 그 효율을 향상시키기 위해서는 문제를 푸는 동안 생성되는 의사결정노드(decision node)의 수를 줄이는 것이 관건이다. 의사결정노드의 수를 줄이기 위해서는 두 가지 측면에서 접근이 가능하다. 첫째는 하한을 되도록 최적해에 가까운 값으로 구하는 방법으로 Held-Karp의 하한 등 여러 논문[4,9]에서 제시되었다. 둘째는 선정된 의사결정노드에서 적절한 분지 기준을 사용하여 하한을 구해야 할 의사결정노드의 수를 줄이는 방법으로 Bellmore-Malone[3]의 분지법이 대표적이며, 이방법보다 좀 더 개선된 방법으로 Carpaneto-Toth[5]의 분지법이 있다. 본 연구에서는 두 번째 접근 방법으로 새로운 분지기법을 제시하고, 실험을 통하여 위의 두 방법보다 우수함을 보인다.

2. 기존의 분지기법

이제까지 비대칭 외판원 문제를 풀기 위한 분지기법들이 많이 개발되어 왔다. 그 중에서 부분 순환로 제거법에 근거한 분지기법이 우수하다고 알려져 있다[3,5]. 이 장에서는 부분순환로 제거법에 바탕을 둔 분지기법인 Bellmore-Malone의 분지기법과 Carpaneto-Toth의 분지기법에 대해 설명한다.

2.1 기호 설명

본 연구에서 앞으로 사용하게 될 기호는 다음과 같다.

$G(N, L)$: 도시집합 N과 호집합 L을 갖는 그래프

n: 도시의 수

$N = \{1, 2, \ldots, n\}$

$L = \{(i, j) \mid i \neq j, i \in N, j \in N\}$

c_{ij}: 도시 i에서 도시 j로 가는 비용

$N_t = \{r_1, r_2, \ldots, r_p\}$, $N_t \subseteq N$

$L_t = \{(r_1, r_2), (r_2, r_3), \ldots, (r_p, r_1)\}$, $L_t \subseteq L$

$\tilde{G}(N_t, L_t)$: 도시집합 N_t와 호집합 L_t을 갖는 새로운 그래프

E_k: 의사결정노드 k에서 사용하지 않기로 한 모든 호의 집합

I_k: 의사결정노드 k에서 사용하기로 한 모든 호의 집합

2.2 Bellmore-Malone의 분지기법

Bellmore와 Malone[3]은 부분순환로 제거법에 근거한 분지기법을 발표하였다. 이 방법은 제거할 부분순환로를 선택함에 있어, 분지한계법을 수행하는 동안 생성되는 의사결정노드의 수를 감소시키기 위해서 호의 개수가 가장 적은 부분순환로를 선택한다. 이 방법의 분지절차는 다음과 같다.

[단계 1] 선정된 의사결정노드 k에 대해 배정문제를 풀어 하한을 구한다.

[단계 2] 단계 1에서 구한 배정문제의 해에서 가장 적은 개수의 호를 포함하는 부분 순환로 $\tilde{G}(N_1, L_1)$를 선택한다. 이때 호의 개수를 p개라 하자.

[단계 3] 모든 부분문제 j (j=1, 2, ..., p)에 대해 각각 다음의 제약을 갖도록 분지한다.

$$E_j = E_k \cup \{(r_j, r_{j+1})\} \quad (2.1)$$
$$I_j = I_k \cup \{(r_u, r_{u+1}) \mid u = 1, \ldots, j-1\} \quad (2.2)$$
$$r_{p+1} = r_1$$

식(2.1)에 의해서 부분순환로 $\tilde{G}(N_1, L_1)$은 새로 분지되는 의사결정노드 j에서 구성될 수 없으며, 식(2.2)에 의해 의사결정노드 k는 p개의 의사결정노드 j (j=1, 2, ..., p)로 분할된다.

2.3 Carpaneto-Toth의 분지기법

Carpaneto와 Toth[5]는 Bellmore-Malone의 분지한계법에서 분지기법과 하한을 구하는 새로운 기준을 제시한 논문을 발표하였다. 이 방법 역시 부분순환로 제거법에 기초한 방법으로서,

부분순환로를 선택할 때 사용하기로 한 호의 집합 I_k에 포함되지 않은 호가 가장 적은 부분순환로를 선택하여 의사결정노드의 수를 줄인다. 이 방법의 분지절차는 다음과 같은 절차로 수행된다.

[단계 1] 선정된 의사결정노드 k에 대해 배정문제를 풀어 하한을 구한다.

[단계 2] 단계 1에서 구한 배정문제의 해에서 집합 I_k에 포함되지 않은 호가 가장 적은 부분순환로 $\breve{G}(N_q, L_q)$를 선택한다. 이때 포함되지 않은 호의 집합을 $\overline{L}_q=\{(i_1, f_1), (i_2, f_2), \ldots, (i_v, f_v)\}$라 하자.

[단계 3] 집합 $\overline{L}_q$에 속한 모든 호 (i_j, f_j) $(j=1, 2, \ldots, v)$에 대해 다음의 파라미터를 계산한다.

h_j: 호 (i_j, f_j)에 직접 연결되어 있는 집합 I_k에 포함된 호의 개수,

$$w_j = h_j(v-1) + h_{j+1}(v-2) + \ldots + h_v(j-1) + h_1(j-2) + \ldots + h_{j-2},$$
$$h_0 = h_{-1} = 0$$

[단계 4] w_j값이 가장 큰 호가 앞으로 오도록 집합 $\overline{L}_q$안의 호들을 회전 이동시킨다.

[단계 5] 모든 부분문제 j $(j=1, 2, \ldots, v)$에 대해 다음의 제약을 갖는 부분문제를 만들어 분지한다.

이 방법의 중요한 개념은 집합 I_k에 포함되지 않은 호의 개수가 적은 부분순환로를 선정한다는 것이다. 이렇게 하는 이유는 I_k에 포함된 호가 식(2.1)의 제약에 의해 사용되지 않게 될 경우는 가능해가 존재하지 않게 되어, 하한을 구할 필요가 없기 때문이다.

3. 새로운 분지기법

3.1 분지한계법에서의 상한

이 절에서는 새로운 분지기법을 제시하기 전에 분지한계법에서 상한을 이용하는 방법과 그 이유를 설명한다. 원래 비대칭 외판원 문제를 분지한계법을 이용해 풀 때, 하한을 구하는 방법으로 외판원 문제를 배정문제로 완화하여 푼다. 이때 배정문제를 푸는 알고리듬으로는 여러 가지가 있지만, 주로 헝가리안 알고리듬(Hungarian algorithm)을 많이 사용한다. 그런데 이 알고리듬의 계산량은 잘 알려진 것처럼 $O(n^3)$[8]으로, 비대칭 외판원 문제를 분지한계법을 이용해 풀 때 대부분의 시간이 배정문제를 푸는 데 소요된다. 따라서, 분지한계법의 효율을 높이기 위해서 하한을 다음과 같은 두 단계로 나누어 구할 필요가 있다.

첫 번째 단계는 주어진 비용행렬의 각 행과 열에 적어도 0이 하나 이상 들어가도록 행렬의 값을 감소시키는 방법[7]을 이용한다. 두 번째 단계는 배정문제를 풀어 하한을 구한다. 첫 번째 단계에서 구한 하한이 현재의 상한보다 높게 되면 계산량이 많은 두 번째 단계의 하한을 구하

지 않아도 된다. 따라서, 가능한 한 낮은 상한을 구해서 현재의 상한을 갱신할 필요가 있다. 이런 이유로 분지한계법에서 상한을 구하는 것이 효율을 높이는 데 도움이 된다.

상한을 구하는 방법에는 여러 가지 발견적 해법들이 있다[8]. 이러한 여러 가지 발견적 해법들 중 어떠한 것이라도 사용할 수 있는데, 본 연구에서는 새로 고안한 다음과 같은 발견적 해법을 사용하여 상한을 구한다.

[단계 1] 의사결정노드 k의 배정문제의 해를 $\tilde{G}(N, L_k)$라고 하자.

[단계 2] $\tilde{G}(N, L_k)$의 첫 번째 부분순환로 $\tilde{G}(N_1, L_1)$을 선택한다.
단, $\overline{N_1} = N - N_1$.

[단계 3] $\max_{\forall i \in N_1, \forall j \in \overline{N_1}} \{c_{(i)(i+1)} - c_{ij} + c_{(j)(j+1)} - c_{(j+1)(i+1)}\}$인 도시 i와 도시 j를 선택해 호$(i, i+1)$와 호$(j+1, i+1)$을 넣어 두 개의 부분순환로를 합친다.

[단계 4] $\tilde{G}(N, L_k)$가 하나의 헤밀톤 순환로이면 끝내고, 그렇지 않으면 단계 1로 간다.

3.2 새로운 분지기법

만약 최적해 $\tilde{G}(N, L^*)$가 주어져 있다고 가정을 하자. 해 $\tilde{G}(N, L^*)$가 최적해임을 확인하고자 한다면 두 가지 방법이 있다. 첫 번째 방법은 최적해를 구하는 알고리듬을 사용하여 최적해를 구해 $\tilde{G}(N, L^*)$와 비교를 하는 방법이고, 두 번째 방법은 알고 있는 최적해를 이용해서 그보다 더 좋은 해가 없음을 증명하는 것이다.

본 연구에서 제시하고자 하는 분지기법은 후자의 방법에 착안하여 고안된 것이다. 즉, 최적해라고 짐작되는 해(여기서는 상한을 이루는 해)의 정보를 최대한 활용하여 분지기준으로 삼는 것이다.

어떤 의사결정노드 k의 배정문제의 해 $\tilde{G}(N, L_k)$가 있다. 이 해에 t개의 부분순환로가 있고 q번째의 부분순환로를 $\tilde{G}(N_q, L_q)$라 하자. 단, q = 1, 2, ..., t. 새로운 분지기법은 Carpaneto-Toth의 분지기법과 같은 방법으로, 집합 I_k에 포함되지 않은 호가 가장 적은 부분순환로를 선정한다. 이 때 선정된 부분순환로를 $\tilde{G}(N_{\bar{q}}, L_{\bar{q}})$라 하고, 집합 I_k에 포함되지 않은 호들의 집합을 다음과 같이 정의한다.

$$\begin{aligned} \overline{L}_{\bar{q}} &= L_{\bar{q}} - I_k \\ &= \{(i_1, f_1), (i_2, f_2), \ldots, (i_v, f_v)\} \end{aligned}$$

분지한계법은 최대한 빨리 우수한 가능해에 도달하는 것이 관건이다[5]. 따라서, 분지를 할 때 분할되는 부분문제들의 제약이 되도록 최적해에 만족될 수 있는 것들이어야 유리하다. 그러므로 가능한 한 빨리 최적해를 이루고 있는 호가 집합 I_k에 포함 되어야 한다. 그러기 위해서는 집합 $\overline{L}_{\bar{q}}$와 집합 L^*에 공통으로 포함된 호를 대체비용에 대해 내림차순으로 앞쪽에 정렬

하고 집합 $\overline{L}_{\bar{q}}$에는 포함되어 있으나 집합 L^*에는 포함되지 않은 호를 대체비용에 대해 내림차순으로 뒤쪽에 정렬한다. 이렇게 정렬한 집합을 $\overline{L}_{\bar{q}}' = \{(i_1', f_1'), (i_2', f_2'), \ldots, (i_v', j_v')\}$라 하자. 그리고 $\overline{L}_{\bar{q}}'$에 대해 식(2.1)과 식(2.2)를 포함하는 부분문제로 분지한다. 이렇게 함으로써, 알고리듬을 수행하는 초반에 최적해를 이루고 있는 호가 사용하는 호로 많이 확정되어 우수한 가능해에 빨리 도달하게 되는 것이다.

최적해 $\tilde{G}(N, L^*)$로 사용될 해는 최대한 실제 최적해에 가까운 값이어야 한다. 그러므로 분지한계법을 적용해 가는 과정에서 그에 근사하는 값을 구해 사용할 수밖에 없다. 분지한계법을 적용해 가는 과정에서 가장 $\tilde{G}(N, L^*)$에 가까운 해는 당연히 현재 상한을 이루고 있는 해이다. 따라서, 이 해를 이용해 $\tilde{G}(N, L^*)$를 대신한다. 결국 이 분지 기준이 더욱 큰 효과를 거두기 위해서는 상한을 구하는 방법이 될 수 있는 한 좋은 것이어야 한다.

이상 설명한 분지기법을 정리하면 다음과 같다.

[단계 1] 의사결정노드 k에 대해 배정문제를 풀어 하한을 구한다. 그리고 위의 상한을 구하는 해법을 적용해 현재의 상한을 갱신한다.

[단계 2] 단계 1에서 구한 배정문제의 해에서 집합 I_k에 포함되지 않은 호가 가장 적은 부분순환로 $\tilde{G}(N_{\bar{q}}, L_{\bar{q}})$를 선택한다. 이 때 포함되지 않은 호의 집합을 $\overline{L}_{\bar{q}} = \{(i_1, f_1), (i_2, f_2), \ldots, (i_v, j_v)\}$라 하자.

[단계 3] 집합 $\overline{L}_q$의 호들을 위의 설명에서 제시한 정렬기준에 따라 정렬한 호의 집합을 $\overline{L}_{\bar{q}}' = \{(i_1', f_1'), (i_2', f_2'), \ldots, (i_v', j_v')\}$라 하자.

[단계 4] 모든 부분문제 j ($j=1, 2, \ldots, v$)가 다음의 제약을 갖도록 분지한다.

$$E_j = E_k \cup \{(i_j', f_j')\}$$
$$I_j = I_k \cup \{(i_u', f_u') | u = 1, \ldots, j-1\}$$

단계 3에서 대체비용에 대해 내림차순으로 정렬하는 이유는 대체비용이 큰 호가 최적해에 포함될 확률이 크기 때문이다[5,7].

4. 새로운 분지기법을 적용한 분지한계법

이상 설명한 상한을 구하는 방법과 새로운 분지기법을 적용한 알고리듬의 수행절차는 다음과 같다. 다음 절차에서는 하한을 구하는 첫 번째 단계인 행렬을 감소시켜 구한 하한을 min1이라 하고, 배정문제를 푼 하한을 min2라 한다.

[단계 1] 리스트 Q를 초기화 한다.

$E_1 = \phi, I_1 = \phi$인 의사결정노드 D_1을 만들어 min2를 구한다.

이 때 구한 배정문제의 해를 3.1절에서 제시한 절차를 적용하여 상한을 구하고 이를 U^*라 하자. D_1를 Q에 저장한다.

[단계 2] Q가 비었으면 단계 7로 간다.
Q에서 가장 낮은 하한을 갖는 D_k를 꺼낸다.
만약 D_k의 하한이 U^*보다 크거나 같으면 단계 7로 간다.

[단계 3] 3.2의 분지기법을 이용하여 의사결정노드 D_j를 만든다. 단, j = 1, 2, ..., v.
j = 1.

[단계 4] D_j에 대해 min1을 구한다.
만약 min1이 U^*보다 크거나 같으면 D_j를 삭제한다. $j = j + 1$. 단계 4로 간다.
그렇지 않으면, 단계 5로 간다.

[단계 5] D_j에 대해 min2를 구한다.
만약 min2가 U^*보다 크거나 같으면 D_j를 삭제한다. $j = j + 1$. 단계 4로 간다.
그렇지 않으면, 단계 6으로 간다.

[단계 6] D_j를 Q에 저장한다.
D_j의 배정문제를 푼 해로 3.1절의 상한을 구하는 절차를 사용하여 새로운 상한 U를 구한다.
만약 $U^* > U$이면 $U^* = U$, $j = j + 1$.
만약 $j > v$이면 단계 2로 간다.
그렇지 않으면, 단계 4로 간다.

[단계 7] 현재의 상한 U^*이 최적해 이다. 그러므로 모든 절차를 끝낸다.

5. 실험 결과

도시 수 50, 100, 150, 200인 문제를 비용범위 1에서 1000에 대해 각각 40, 30, 30, 30개의 문제를 무작위로 만들어 기존의 두 방법(Bellmore-Malone; ATSP1, Carpaneto-Toth; ATSP2)과 새로운 방법(ATSP3)으로 실험했다. 분지기법의 차이에 의한 효과만을 측정하기 위해 기존의 방법에도 상한을 구하는 해법을 추가로 적용했다.

표 1은 배정문제를 푼 평균 횟수이고, 표 2는 배정문제를 푼 최대 횟수이다. 그리고 그림 1은 Bellmore-Malone의 분지기법을 사용한 알고리듬의 수행시간을 100으로 보았을 때 다른 두 방법의 상대시간이다.

프로그래밍 언어는 C++를 사용했으며 컴파일러는 MS사의 Visual C++ 2.0을 사용했다. 수행

한 기계의 중앙처리장치는 펜티엄 100Mhz이다.

표 1. 배정문제를 푼 평균 횟수

도시 수	ATSP1	ATSP2	ATSP3
50	37.3	36.6	34.2
100	106.4	104.8	100.1
150	121.3	125.8	110.7
200	202.1	205.6	177.5

표 2. 배정문제를 푼 평균 횟수

도시 수	ATSP1	ATSP2	ATSP3
50	163	154	125
100	381	361	346
150	462	511	265
200	534	710	493

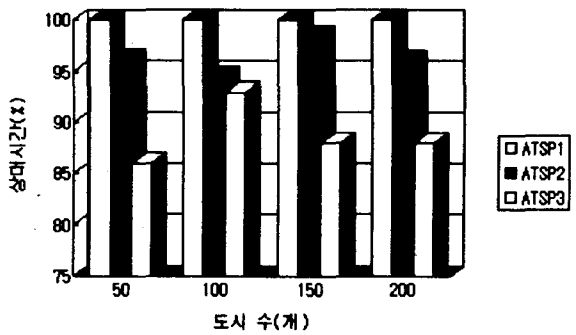

그림 1. ATSP1에 대한 상대 수행 시간

이상의 실험결과에서 보여 주듯이 새로운 분지기법을 사용한 알고리듬(ATSP3)이 Bellmore-Malone의 분지기법을 사용한 알고리듬(ATSP1)보다 배정문제를 푼 회수에서는 약 8%에서 10%의 효율 향상을 보이고, 수행 시간에 있어서는 약 10%에서 14%의 효율 향상이 있음을 알 수 있다. 특히, 배정문제를 푼 최대 회수에서 도시 수 150개인 문제에 대해서는 약 46% 정도까지 줄인 것을 볼 수 있다.

6. 결론

비대칭 외판원 문제는 대표적인 NP-Complete 문제이다. 이제까지 이 문제의 최적해를 찾기 위해 많은 노력이 경주되어 왔다. 그것들 중 가장 우수한 방법들로 알려진 것은 부분순환로 제거법에 근거한 분지한계법이다. 이러한 분지한계법으로는 이미 Bellmore-Malone의 분지한계법과 Carpaneto-Toth의 분지한계법이 있다. 그러나 분지한계법 역시 근본적으로는 열거법이기

때문에 그효율을 더욱 향상시킬 필요가 있다.

분지한계법의 효율을 향상시키기 위해서는 최적치에 가까운 하한을 구하는 방법과 효과적인 분지기법을 사용하여 의사결정노드의 수를 줄이는 방법이 있다. 본 연구에서는 후자의 방법으로 새로운 분지기법을 제시 하였다.

본 연구에서는 분지한계법을 수행하는 도중 갱신되는 상한을 이루는 해의 정보를 최대한 이용하여 분지의 기준으로 삼았다. 즉, 최적해라고 짐작되는 해를 알고 있을 때 그 해를 이루는 값들을 분지기법에 반영하는 것이다. 상한을 이루는 해는 분지한계법이 수행되는 도중에 발생하는 가장 최적해에 가까운 해이다. 따라서, 이 해를 최적해로 가정하는 것이다. 본 연구에서 새로 제시한 분지기법 역시 부분순환로 제거법에 근거한 것이다. 그러나 기존의 방법들과는 달리 상한을 이루는 해의 정보를 활용하여 분지를 할 때 사용하는 호들을 정렬함으로써, 가능한 한 빨리 우수한 가능해가 나오도록 유도하였다. 그리고 순수하게 분지기법만 다르게하여 기존의 두 방법과 함께 실험을 하였다.

실험 결과에서 보여 주듯 기존의 두 가지 분지기법보다 새로운 분지기법이 많은 경우 46% 더 적은 의사결정노드에 대해 배정문제를 풀었음을 알 수 있으며, 시간은 평균적으로 약 10%에서 14%를 줄였다. 실제로 본 연구에서 제시한 분지기법은 그 방법의 특성상 가능한 한 최적해와 가까운 상한을 이루는 해를 구할수록 그 결과가 좋았다. 따라서, 본 연구에서 상한을 구할 때 사용한 발견적 해법보다 더욱 우수한 해법을 사용한다면 더 좋은 결과를 기대할 수 있다. 그리고 새로 제시한 분지기법의 기본 개념은 가능한 한 빨리 좋은 가능해가 나와야 한다는 분지한계법의 특성에 잘 맞는다. 따라서, 외판원 문제를 풀기 위한 분지한계법에서 뿐만이 아니라 다른 문제를 풀기 위한 분지한계법에도 잘 적용될 수 있을 것으로 기대된다.

참 고 문 헌

1. 강맹규, *네트워크와 알고리듬*, 박영사, 서울, 1991.

2. Balas, E. and P. Toth, "Branch and Bound Methods," in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, (eds.) E. L. Lowler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Wiley, New York, pp. 361-701, 1985.

3. Bellmore, M. and J. C. Malone, "Pathology of Traveling Salesman Subtour Elimination Algorithms," *Operations Research*, Vol. 19, pp. 278-307, 1971.

4. Carpaneto, G., M. Fischetti, and P. Toth, "New Lower Bounds for the Symmetric Traveling Salesman Problems," *Mathematical Programming*, Vol. 45, pp. 233-254, 1989.

5. Carpaneto, G. and P. Toth, "Some New Branching and Bounding Criteria for the Asymmetric Traveling Salesman Problem," *Management Science*, Vol. 26, No. 7, pp. 736-743, 1980.

6. Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson, "Solution of a Large-Scale Traveling Salesman Problem," *Operations Research*, Vol. 2, pp. 393-410, 1954.

7. Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel, "An Algorithm for the Traveling Salesman Problem," *Operations Research*, Vol. 11, pp. 972-989, 1963.

8. Pekny, J. F., D. L. Miller, and D. Stodolsky, "A Note on Exploiting the Hamiltonian Cycle Problem Substructure of the Asymmetric Traveling Salesman Problem," *Operations Research Letters*, Vol. 10, pp. 173-176, 1991.
9. Williamson, D. P., "Analysis of the Help-Karp Lower Bound for the Asymmetric TSP," *Operations Research Letters*, Vol. 12, pp. 83-88, 1992.