Journal of the Korean Institute
of Industrial Engineers
Vol.22, No.3, September, 1996

차량경로문제에 대한 발견적 해법

A Heuristic for the Vehicle Routing Problem

노인규*, 예성영*

In Kyu Ro*, Sung Young Ye*

Abstract

This study is concerned with developing a heuristic for the vehicle routing problem(VRP) which determines each vehicle route in order to minimize the transportation costs, subject to meeting the demands of all delivery points.

VRP is known to be NP-hard, and it needs a lot of computing time to get the optimal solution, so that heuristics are more frequently developed than optimal algorithms. This study aims to develop a heuristic which can give a good solution in comparatively brief time.

Finally, the computational tests were performed using the benchmark problems and the proposed heuristic is compared with the other existing alrogithms. The result of computational tests shows that the proposed heuristic gives good solutions, in much shorter time, which are not 1% more expensive than the best known solutions, which are same as the best known solutions in the previous researchs.

1. 서론

차량경로문제(Vehicle Routing Problem)란 재화 및 서비스를 수집 또는 배분하기위하여 차량을 할당하고 그 운행경로를 결정하는 문제이다. 일반적으로 차량경로문제의 최적화 목표는 운행 비용의 최소화이며 제약 조건은 가용 차량의 대수와 적재 요구량, 수요지점의 수요량 및 차량에 대한 각종 운행 조건 등이다. 차량경로문제는 처음 Dantzig와 Ramser[2]에 의해 제기된 이후 관련된 제약조건을 다양화하여 광범위한 연구가 꾸준히

* 한양대학교 산업공학과

진행되고 있으며, 최근에 이르러서는 물류와 관련한 수송비 절감의 필요성에 대한 인식으로 그 중요성을 더해 가고 있다.

본 연구는 하나의 창고와 적재 용량이 동일한 차량, 수요량이 알려진 수요지점, 그리고 단일의 수요지점에 대해서는 오직 한대의 차량만이 일회 방문한다는 일회방문원칙의 제약 조건하에서, 모든 수요지점의 수요량을 만족시키면서 총 운행 비용을 최소화하는 차량의 운행 경로를 구하는 발견적 해법을 개발하는 데 그 목적이 있다.

차량경로문제는 수리적으로 NP-hard 문제에 속하는 계산량이 매우 많은 문제이다. 또한 문제의 크기가 커질수록 계산량이 급속히 증가하는 등 규모가 큰 문제에 최적화 해법을 적용하기에는 어려움이 많아 발견적 해법이 상대적으로 활발히 연구되고 있다. Clarke와 Wright[3]는 saving 개념을 도입한 발견적 해법을 개발하였다. 이때 saving이란 두 대의 차량이 각각 서로 다른 두 수요지점을 방문하고 돌아오는 것에 비해 한 대의 차량이 두 수요지점을 한번에 경유하여 돌아올 때 얻게 되는 거리(또는 비용)의 절약을 말한다. Holmes와 Parker[4]는 savings 기법을 개량한 발견적 해법을 개발하였다. savings 기법으로 구한 경로내의 arc중 해의 질을 떨어뜨리는 arc를 가려내 제거하여 새로운 경로를 구성하므로써 개선된 해를 구하는 것을 그 내용으로 한다. Altinkemer와 Gavish[5]는 수요지점간의 saving 개념을 그대로 이용하면서 수요지점간의 결합에 matching 모형을 적용하여 해를 구하는 발견적 해법을 개발하였다. Osman[6]은 simulated annealing 개념을 도입하여 발견적 해법을 개발하였고, Gendreau, Hertz와 Laporte[7]는 tabu search 개념을 VRP에 도입하여 발견적 해법을 개발하였다. 이렇게 simulated annealing이나 tabu search의 개념을 도입한 발견적 해법들은 최근에 보다 활발히 연구되고 있으며, 이러한 해법들은 보다 우수한 해를 구해낼 수 있다고 알려져 있다. 그러나 해법의 수행시간이 매우 오래 걸린다는 점이 단점으로 지적되고 있다. 그 외에도 유전자 알고리듬을 차량경로문제에 도입하여 기본적인 차량경로문제의 일회방문원칙의 제약을 완화하여 수요지점에 대한 다회방문을 허락하는 경우에 대해서도 적용이 가능한 발견적 해법을 개발한 신[1]의 연구 등이 있다.

본 연구는 서론에 이어 제 2 장에는 차량경로문제의 전통적 수리 모형 및 관련 기호를 설명하고, 제 3 장에서는 본 연구에서 개발한 해법을 소개한다. 제 4 장에서는 Christofides[8,9]가 제시한 기준문제(benchmark problem)에 본 연구에서 개발한 해법을 적용하여 해를 구한 후, 이와 동일한 문제에 대하여 해를 구한 기존의 연구들과 그 결과를 비교, 분석한다. 제 5 장에서는 본 연구의 결론을 유도하고 추후 연구 과제를 제시한다.

2. 수리 모형

2.1 모형의 정의 및 가정

차량경로문제의 전통적 모형은 단일의 본점과 N개의 수요지점,그리고 차량 용량이 U인 M대의 차량으로 구성되며, 수요지점의 수요량은 미리 알려져 있고 차량의 운행 거리(또는 시간)에 제한이 있을 수 있다. 또한 본 연구는 전통적 차량경로문제의 기본 가정들

을 따른다.

2.2 기호의 정의

N : 수요지점의 수

D_i : 수요지점 i의 수요량 (i=1,2,⋯,N)

M : 차량 수

U : 차량의 용량

C_{ijk} : 차량 k가 수요지점 i와 수요지점 j를 운행하는 경로 거리(또는 시 간) (i=0,1,⋯,N; j=0,1,⋯,N; k=1,2,⋯,M; i,j가 '0'일때 본점을 의미)

W : 차량의 최대 운행 거리(또는 시간)

수리 모형은 다음과 같다.

$$\min \sum_{k=1}^{M}\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ijk}\, x_{ijk} \tag{2.1}$$

subject to

$$\sum_{k=1}^{M}\sum_{i=0}^{N} x_{ijk} = 1 \;\forall\; j \tag{2.2}$$

$$\sum_{k=1}^{M}\sum_{j=0}^{N} x_{ijk} = 1 \;\forall\; i \tag{2.3}$$

$$\sum_{j=0}^{N} x_{ijk} - \sum_{j=0}^{N} x_{jik} = 0 \;\forall\; i,k \tag{2.4}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ijk}\, x_{ijk} \le W \;\forall\; k \tag{2.5}$$

$$\sum_{i=1}^{N} D_i \Big(\sum_{j=0}^{N} x_{ijk}\Big) \le U \;\forall\; k \tag{2.6}$$

$$\sum_{j=1}^{N} x_{0jk} \le 1 \;\forall\; k \tag{2.7}$$

$$\sum_{i=1}^{N} x_{i0k} \le 1 \;\forall\; k \tag{2.8}$$

$$y_i - y_j + N \cdot \sum_{k=1}^{M} x_{ijk} \le N-1$$

$$\forall\; i,j (i \neq j) \tag{2.9}$$

$$y_i : real\ number \;\forall\; i \tag{2.10}$$

$$x_{ijk} = \{0,1\} \;\forall\; i,j,k \tag{2.11}$$

수리 모형의 식(2.1)은 총 운행 거리(또는 시간)를 최소화하는 목적식이다. 이때 거리(또는 시간)는 비용과 일대일 대응하므로 식(2.1)은 곧 총 운행 비용을 최소화하는 목적식이기도 하다. 식(2.2)와 식(2.3)은 각 수요지점은 한 대의 차량에 의해 한번만 방문되어야 한다는 것을 의미한다. 식(2.4)는 차량이 수요지점을 방문한 후 반드시 떠나야 한다는 것을 의미하며, 식(2.5)는 경로에 포함된 지점들을 방문하는데 드는 총 운행 거리(또는 시간)는 차량의 최대 운행 거리(또는 시간)를 초과할 수 없음을 의미한다. 식(2.6)은 각 경로에 포함된 수요지점의 총 수요량은 차량의 적재용량을 초과할 수 없다는 것을 의미한다. 식(2.7)과 식(2.8)은 모든 수요지점을 방문하는 데 필요한 차량 수는 주어진 차량 수 M 대를 넘을 수 없다는 것을 의미하며, 식(2.9)와 식(2.10)은 모든 차량들이 본점을 통과한다는 것을 의미한다. 식(2.11)은 0,1 정수 조건이다.

3. 발견적 해법

3.1 내용

본 연구에서 제시하는 발견적 해법의 내용은 다음과 같다.

① Clarke와 Wright[3]의 saving 개념을 이

용한다.

② 초기해는 Clarke와 Wright[3]의 savings 기법으로 구한다.

③ savings 기법을 이용함에 있어 Holmes와 Parker[4]의 접근과는 다음의 차이가 있다. 즉, Holmes와 Parker[4]는 경로내 임의의 arc의 제거 여부의 결정에 촛점을 맞추어 해법을 진행한 것에 반해 본 해법은 이웃해 집합을 구성하고 그 과정에서 보다 개선된 해를 구하는 집합적인 접근 방식을 취한다. 따라서 집합을 구성하는 조건과 집합의 크기를 결정하는 것을 중요시 하며 본 연구의 실험에서는 집합을 구성하는 조건과 집합의 크기에 따른 해의 질적 개선 및 수행 시간에 대한 영향을 함께 분석하고자 한다. 이렇게 집합적 접근을 취함으로써 Holmes와 Parker[4]가 해법의 수행중 얻은 해보다 더 많은 해를 탐색하여 보다 개선된 해를 구할 수 있다.

④ 초기해의 각 차량 경로는 수요지점간의 arc를 연결한 형태로 구성된다. 이때 연결되어 있는 임의의 한 arc를 경로에서 제거하고 savings 기법을 적용하여 경로를 재구성하므로써 새로운 경로를 얻게 되며, 이것을 초기해의 이웃해 생성 절차로 한다. 이때 arc수만큼의 이웃해들을 얻는다.

⑤ 내용 ④의 절차를 다시 이웃해에 적용하여 해당 이웃해의 이웃해들을 구하며, 이러한 이웃해 생성 절차를 k 번 반복하므로써 이웃해의 범위를 계속 넓혀 갈 수 있다. 이때 k를 이웃해 생성 제한 횟수라 하여 k에 따라 이웃해의 범위가 정해진다. 이웃해 생성 제한 횟수 k는 이웃해 집합의 범위를 정하는데 매우 중요한 역할을 하므로 3.3 절에서 자세히 다루고자 한다.

⑥ 이웃해 생성 단계에서 추가로 arc를 제거할 때는 이미 앞서 제거된 arc들에 비해 saving 값이 작은 arc를 제거한다. 또한 saving 기법을 적용할 때도 제거된 arc보다 saving 값이 작은 arc들만을 대상으로 하며 saving 값이 큰 arc들은 그대로 기존 해에서 가져온다.

⑦ 모든 해들은 자신의 이웃해 생성 절차를 수행하기 전에 이웃해를 생성해도 좋은지에 대한 여부를 가리고, 이웃해를 생성해도 좋은 경우에만 이웃해를 생성하게 되며 그 기준은 다음과 같다. 즉, 자신이 임의의 해로부터 이웃해로서 유도되어 나왔을 때 그 해보다 값이 좋은 해만이 자신의 이웃해를 생성할 수 있다.

3.2 알고리듬

본 연구에서 개발한 해법의 알고리듬내에 사용된 기호는 다음과 같다.

K : 이웃해 생성 제한 횟수

k : 이웃해 생성 횟수, k=1,…,K

V_k : 이웃해 생성 절차를 k번 수행하여 얻은 이웃해의 값, k=1,…,K

A_k : 이웃해 생성 절차를 k번 수행하여 얻은 이웃해의 arc의 집합, k=1,…,K

S_k : 이웃해 생성 절차를 k번 수행하여 얻은 이웃해의 arc 수, k=1,…,K

i_k : A_k에 있는 arc 번호, i_k=1,…,S_k

본 연구에서 개발한 해법의 절차는 다음과 같다.

단계 0 : 초기화

모든 수요지점간의 saving을 계산하고 이를 내림차순으로 정렬한다.

savings 기법으로 초기해를 구한다.

이때 구한 해와 경로를 최선해, 최선 경로로 저장한다.

k=1

구한 해, 경로, arc 수를 각각 V_k, A_k, S_k에 저장한다.

i_k=1.

단계 1 : 이웃해 생성

i_k번째 arc를 경로에서 제거한다.

i_k번째 arc보다 saving 값이 큰 arc들은 그대로 두고, 그 보다 saving 값이 작은 arc들에 대해 savings 기법을 다시 적용하여 새로운 이웃해를 구한다.

이때 구한 해, 경로, arc 수를 V_{k+1}, A_{k+1}, S_{k+1}에 저장한다.

단계 2 : 최선해, 최선 경로 수정

단계 1에서 구한 해가 최선해보다 좋은 경우 최선해, 최선 경로를 수정한다.

단계 3 : 이웃해를 생성할 것인지의 결정

k가 미리 내정된 이웃해 생성 제한 횟수 K를 넘지 않고, V_{k+1}이 V_k보다 좋은 경우에만 이웃해를 생성한다. 이웃해를 생성하기로 한 경우 단계 4로 가고, 아니면 단계 5로 간다.

단계 4 : k의 증가

k=k+1.

i_k=i_{k-1}.

단계 1으로 간다.

단계 5 : i_k의 증가

i_k=i_k+1.

i_k가 S_k보다 크면 단계 6으로, 아니면 단계 1로 간다.

단계 6 : k의 감소

k=k-1.

이때 k가 1 이상이면 단계 5로, k가 0 일때 단계 7로 간다.

단계 7 : 종료

3.3 이웃해 생성 제한 횟수의 결정

본 해법에서 해의 질적 개선과 수행 시간은 이웃해 생성 제한 횟수 k에 크게 영향 받으며, 이때 이웃해 생성 제한 횟수 k의 역할은 다음과 같다. 즉, 본 해법이 새로운 이웃해들을 생성하는 절차를 반복하여 이웃해 집합을 계속 넓혀 가는 과정에서 이웃해를 생성하는 절차의 횟수에 제한을 두므로써 이웃해 집합의 범위를 결정하고 이로 인하여 해의 개선정도나 수행 시간은 달라지게 된다. 이웃해 생성 절차가 많을수록 더욱 개선된 해를 얻을 수 있으며 이때 수행 시간은 더욱 늘어난다. 본 연구에서는 해법의 개발과 함께 모수 이웃해 생성 제한 횟수 k에 대한 분석을 제시한다. 구체적인 이웃해 생성 제한 횟수 k의 적절한 값에 대해서는 본 해법을 기준문제에 적용하여 해를 구한 후 4.4절에서 다룬다.

4. 수치 예제

이 장에서는 본 해법의 성능을 분석하기 위하여 Chritofides[8,9]가 제안한 기준 문제에 해법을 적용하여 그 해를 도출하고, 동일한 문제에 대하여 해를 도출하였던 기존의 연구들과 비교하고자 한다. 이때 비교 대상은 기준문제의 해의 값과 해법의 수행시간이다. 실험은 Intel 80586 60MHz CPU를 장착하고 WINDOWS 3.1을 OS로 하는 IBM PC을 이용하여 수행하였다.

4.1 기준 문제

실험에 사용한 기준 문제는 Christofides [8,9]가 제안하였으며 표 1과 같다. 이때 지

표 1. 기준 문제

문제번호	지점수	수요량범위	차량용량	운행시간제한	하역시간
1	50	3 ~ 41	160	-	-
2	75	1 ~ 37	140	-	-
3	100	1 ~ 36	200	-	-
4	120	2 ~ 35	200	-	-
5	100	10 ~ 50	200	-	-
6	50	3 ~ 41	160	200	10
7	75	1 ~ 37	140	160	10
8	100	1 ~ 36	200	230	10
9	120	2 ~ 35	200	720	50
10	100	10 ~ 50	200	1040	90

점수 150, 199인 문제는 실험기기의 용량부족으로 인하여 본연구에서는 제외되었다. 기준문제에 대하여 본 연구의 해법과 함께 비교할 기존 연구들은 표 2와 같다.

4.2 실험 결과

본 연구에서 개발한 해법으로 구한 기준문제의 해와 해법의 수행 시간을 실험의 결과치로서 제시한다. 본 연구에서 개발한 해법은 이웃해 생성 제한 횟수 k에 따라 해의 질과 해법의 수행시간이 달라진다. 따라서 기준 문제 각각에 대해서 k에 따라 구분하여 결과치를 제시한다.

1) 차량의 운행 시간에 제한이 없는 경우

문제 1~5는 차량의 운행시간에 제한이 없는 경우에 대한 문제이다.

표 3은 여러 가지 k 값에 따른 운행비용이며 표 4는 해법의 수행시간이다. 표 3과 표 5를 살펴보면 k 값이 커질수록 더욱 개선된 해를 구할 수 있음을 알 수 있다. 그러나 k가 일정 값 이상으로 커지면 더 이상 해가 개선되지 않는다. 표 3과 표 5에 공백으로 남아 있는 부분은 해가 더 이상 개선되지 않았음을 의미한다.

2) 차량의 운행 시간에 제한이 있는 경우

문제 6~10은 문제 1~5와 본점이나 수요지점의 위치, 수요지점의 수요량, 차량의 용량 등의 제약은 동일하나 차량의 운행 시간에 제한이 있다는 점이 다르다.

표 5는 여러 가지 k 값에 따른 결과치이며 표 6은 해법의 수행시간이다.

4.3 결과 분석

이 절에서는 본 연구에서 개발한 해법으로 구한 4.2절의 결과를 분석하고 기존 연구들의 결과와 비교한다. 본 연구의 기존 연구들과의 결과 비교는 표 7과 같다. 표 7중 '본 연구의 해법'을 제외한 나머지 결과들은 Gendreau et al.[7]에서 인용하였다.

k 값을 늘려 감에 따라 해의 개선 정도는 일반적으로 발견적 해법에서 나타나는 것과 같이 하강하는 지수곡선 형태를 띈다. 또한 결과를 살펴 보면, k 값을 늘려도 일정 값 이상에서는 더 이상 해가 개선되지 않고 수렴하는 것을 볼 수 있다. 이때 해의 수렴 정도는 해법내에 존재하는 이웃해의 생성 여부를 결정하는 판단 기준에 결정적으로 영향 받는다. 즉, 본 해법의 수행 절차에서는 "자신을 생성했던 전 단계의 해보다 값이 더 좋아진 이웃해만이 새로운 이웃해를 생성할 수 있음"을 그 기준으로 하였는데, 이 기준을 달리 하면 결국 이웃해 집합의 생성이 달라져

표 2. 기준 문제에 대한 기존 연구

기 호	해 법	년 도
CW	Clarke & Wright의 savings 해법[3]	1964
GM	Gillet & Miller의 SWEEP 해법[10]	1974
MJ	Mole & Jameson의 generalized savings 해법[11]	1976
CMT1	Christofides,Mingozzi,Toth의 two-phase 해법[9]	1979
CMT2	Christofides,Mingozzi,Toth의 트리 탐색 해법[9]	1979
FJ	Fisher & Jaikumar의 two-phase 해법[12]	1981
DV	Desrochers & Verhoog의 MBSA 해법[13]	1989
AG	Altinkemer & Gavish의 PSA-T 해법[9]	1991
PF	Pureza & França의 tabu search 해법[14]	1991
OSA	Osman의 simulated annealing 해법[6]	1993
TABUROUTE	Gendreau, Hertz, & Laporte의 tabu search 해법[7]	1994
BEST	기존의 최선해	

표 3. 차량의 운행 시간에 제한이 없는 문제에 대한 해

K	문제 1	문제 2	문제 3	문제 4	문제 5
1	548.49	867.22	863.94	1051.47	824.68
2	546.43	863.84	853.23	1049.82	823.25
3	944.97	859.31	851.48	1049.55	822.32
4	541.06	852.28	850.31	-	820.89
5	538.99	844.55	849.62	-	819.97
6	-	840.57	848.47	-	819.56
7	-	839.76	848.06	-	-

표 4. 차량의 운행 시간에 제한이 없는 문제에 대한 해법의 수행시간

K	문제 1	문제 2	문제 3	문제 4	문제 5
1	2.857	19.230	54.170	75.329	45.714
2	4.725	38.956	99.615	115.659	60.484
3	8.351	99.340	244.175	244.505	96.318
4	13.516	234.065	660.989	-	150.164
5	17.250	465.549	1603.901	-	200.604
6	-	770.549	3145.164	-	232.032
7	-	1083.186	5285.549	-	-

표 5. 차량의 운행 시간에 제한이 있는 문제에 대한 해

K	문제 6	문제 7	문제 8	문제 9	문제 10
1	579.98	951.58	948.85	1568.32	868.52
2	557.64	942.04	922.33	1566.70	866.95
3	555.43	936.58	919.65	1564.72	-
4	-	936.14	917.05	1558.73	-
5	-	-	916.46	1557.29	-
6	-	-	-	-	-
7	-	-	-	-	-

표 6. 차량의 운행 시간에 제한이 있는 문제에 대한 해법의 수행시간

(단위 : 초)

K	문제 6	문제 7	문제 8	문제 9	문제 10
1	3.186	14.615	43.186	75.714	44.318
2	6.040	23.846	68.846	130.109	50.602
3	10.604	45.659	149.340	360.274	-
4	-	76.538	332.802	812.637	-
5	-	-	639.870	2368.131	-
6	-	-	-	-	-
7	-	-	-	-	-

표 7. 기준문제에 대한 본 연구의 해법 및 기존 연구의 결과 비교

문제	지점수	본 연구의 해법	CW	GM	MJ	CMT1	CMT2	FJ	DV	AG	PF	OSA	TABU ROUTE	BEST
1	50	538.99	585	532	575	547	534	524*	586	556	536	528	524.61	524.61**
2	75	839.76	900	874	910	883	871	857	885	855	842	838.62	835.32	835.26**
3	100	848.05	886	851	882	851	851	833	889	860	851	829.18	826.14	826.14**
4	120	1049.44	1079	1266	1100	1066	1092	-	1058	1047	1049	1176	1042.11	1042.11
5	100	819.56	831	937	879	827	816*	824	828	834	826	826	819.56	819.56**
6	50	555.43	619	560	599	565	560	560	593	577	560	555.43	555.43	555.43
7	75	936.14	976	933	969	969	924	916	963	939	929	909.68	909.68	909.68
8	100	916.46	973	888	999	915	885	885	914	913	887	866.75	865.94	865.94
9	120	1557.29	1634	1770	1590	1612	1608	-	1562	1551	1631	1545.98	1545.93	1545.93
10	100	866.95	877	949	883	876	878	848	882	874	866	890	866.37	866.37

* : 정수 해로서 실수 해의 반올림 또는 버림 등의 연산 과정에서 실수 최적해보다 작게 나왔음.
** : Fisher[16]에 의해 최적해(optimal solution)임이 증명된 값
본 표는 '본 연구의 해법'을 제외하고Gendreau et al.[7](*Management Science*, Vol.40,No.10,1994) 의 pp.1282에 수록된 table 1에서 인용하였음.

이웃해 집합이 커질 수도 작아질 수도 있으며 이것은 곧 해법의 수행 속도 및 해의 수렴과 바로 직결된다.

표 3~7을 참고로 하여 결과치의 질적인 측면에서 분석하면, 문제 1,2,3,5의 최선해는 Fisher[16]에 의해 최적해(optimal solution)임이 증명되었다. 대부분의 기준 문제들에 대한 기존의 최선해는 Gendreau et al.[7]의 tabu search 해법을 위시하여 tabu search나 simulated annealing 개념을 도입한 해법들이 최선해를 도출하였다. 한편 본 연구에서 개발한 해법은 위의 10개의 기준문제중 문제 5와 문제 6, 두 문제에서 상대적으로 빠른 시간안에 최선해와 동일한 결과를 내었다. 특히 문제 5의 해가 최적해임은 이미 지적한 바와 같다. 기존의 최선해와 동일한 해를 도출한 기준 문제는 표 8과 같다.

표 8. 기존 최선해와 동일한 해를 도출한 기준 문제

(시간 단위 : 초)

문 제 5			문 제 6		
해	819.56		해	555.43	
시간	본 해법	232.03	시간	본 해법	10.60
	BEST*	960		BEST*	820

* : Gendreau et al.[7]의 tabu search 해
사용된 실험기기 : Silicon Graphics Workstation, 36MHz, 5.7Mflops

본 연구에서 개발한 해법은 tabu search 나 simulated annealing 등과 같이 해의 질적인 측면에 집중하여 개발된 해법들(PF, OSA, TABUROUTE - 표 2 참조)을 제외한 다른 해법들에 비해 비교적 우수한 해를 도출하였음을 확인할 수 있다. 특히 Christofides et al.[9]이 보다 실제적인("practical")한 문제라 언급한 문제 4,5,9,10에 있어서는 본 연구의 해법의

해가 더욱 우수한 결과를 나타내고 있다.

여기서 Gendreau et al.[7]는 표 7에 나온 기존 해법의 결과 수치에 대해 다음과 같은 문제점이 있음을 지적하였다. 즉, TABUROUTE 해법과 같이 수요지점간 거리(또는 시간) 및 각 차량의 경로 거리(또는 시간)를 실수로 계산하여 해를 구한 해법들과 그 거리(또는 시간)를 정수로 계산한 해법들사이에 해의 정확한 비교에 문제가 있음을 지적하였다. 예를 들어, 문제 1의 경우 거리(또는 시간)를 실수로 계산할 경우 자신의 TABUROUTE 해법이 구한 524.61이 최적해(optimal solution)임을 증명하고(Hadjiconstantinou & Christofides[15]), FJ 해법이 구한 수치 524에 의문을 제기하였다. 또한 문제 5의 경우에서는 CMT2 해법은 결과치가 816이라 하였으나 Fisher[16]는 문제 5의 실수 최적해가 819.56임을 증명하여 실수해와 정수해의 비교에 문제가 있다는 Gendreau et al.[7]의 지적에 일리가 있음을 간접적으로 입증하였다. 따라서 본 연구의 해법 역시 실수로 해를 구하였으므로 Gendreau et al.[7]의 경우와 같이 실수해로서 최선해를 선정하고 본 해법의 해와 비교하기로 한다.

한편 tabu search 해법 등은 해의 질적인 측면만을 고려하고 해법의 수행 시간의 신속함은 고려하지 않으므로 수행 속도는 다른 발견적 해법에 비해 매우 느리다고 할 수 있다. 따라서 해의 질과 함께 시간적 측면 역시 중요시되어 보다 신속한 응답을 요구하는 상황에서는 tabu search 해법 등은 보다 좋은 해를 구할 수 있음에도 불구하고 부적절하다고 할 수 있다.

해법의 수행 속도 측면에서 살펴 보면, 표 8에서 보면 동일한 최선해를 구하는데 소요된 수행 시간은 본 연구의 해법이 문제 5에서는 약 4분의 1, 문제 6에서는 약 80분의 1 정도로, 본 연구의 해법이 상대적으로 매우 빠른 해법임을 알 수 있다. 그 밖에 기존의 해법들중 FJ, DV, AG (표 2 참조) 등은 최적화 기반 발견적 해법(optimization-based heuristic)으로서 보다 좋은 해를 구하는 대신 수행 시간이 상당히 긴 발견적 해법임을 감안한다면 본 연구의 해법은 수행 속도 측면에서 다른 해법에 비해 우월하다고 할 수 있다.

문제 5와 문제 6을 제외한 다른 기준 문제들에 대하여 본 해법의 결과치와 최선해와의 상대 오차 및 그를 구하는데 소요된 시간의 비교는 표 9와 같다.

표 9. 본 해법의 결과치와 최선해의 상대 오차 및 수행 시간 비교

문 제	상대오차(%)*	시간(배)**
1	2.67	0.05
2	0.53	0.10
3	2.58	4.79
4	0.70	0.18
5	0.00	0.24
6	0.00	0.01
7	2.83	0.02
8	5.51	0.42
9	0.82	0.67
10	0.07	0.01

* : (결과치 - 최선해) / 결과치 * 100

** : 본 해법의 수행시간 / 최선해의 수행시간

- 문제 2 : 본 해법의 수행시간 / Fisher[16]의 수행시간
- 그 외 문제 : 본 해법의 수행시간 / Gendreau et al.[7]의 수행시간

표 9에서 본 해법은 최선해와 0.00% ~

5.51% 오차의 해를 구하는 데 소요되는 시간은 0.01 배에서 4.79 배의 시간이 소요되며 반 이상의 문제에서 1%미만의 차이의 해를 구하는데 소요되는 시간은 4분의 1정도이다.

4.4 이웃해 생성 제한 횟수에 대한 결과 분석

본 해법에서 사용되는 이웃해 생성 제한 횟수 k는 k-optimal 해법의 모수 k와 유사하다. 즉, 표 3~6에 나타나 있듯이 본 해법은 k 값이 커짐에 따라 해는 더욱 개선되지만 그 개선되는 폭이 점차 작아지며 하강하는 지수 곡선 형태를 띈다. 이때 해법의 수행 시간은 k 값이 커짐에 따라 전 단계에 비해 2배 이하의 작은 폭으로 증가한다. 이것은 수행 시간이 기하급수적으로 증가하여 '3'을 경계로 해법의 효율성이 확연히 달라지는 k-optimal 해법의 경우와 뚜렷한 차이를 보이는 것으로 결과적으로 수행 시간에 있어서는 확연히 구분되는 경계가 없어 k 값을 결정하기가 어렵다.

따라서 보다 빠른 응답을 요구하는 상황에서는 k 값을 작게 하여 해를 보다 빠른 시간안에 구할 수 있고 수행 시간보다 해의 질이 더욱 중요한 상황에서는 k 값을 크게 하여 보다 개선된 해를 얻을 수 있다. 이때 k 값을 크게 하여도 수행 시간은 급속히 증가하지 않을 것이므로 k 값의 선정에는 일반적인 발견적 해법에 비해서 그 선택의 폭이 넓다고 할 수 있다. 또한 선택의 폭이 넓음으로 인해 보다 적절한 k 값의 선정에 대해서는 앞으로 좀더 깊이 있는 연구가 필요하다.

결론적으로 본 해법의 이웃해 생성 제한 횟수 k는 '2'에서 '5' 사이의 값으로 처한 상황에 맞추어 정하기를 제안하고자 한다. 이러한 성질은 본 해법내의 이웃해의 생성 여부를 결정하는 판단 기준에 결정적으로 영향받아 도출된 것이다. 그 판단 기준을 달리 한다면 해법의 해의 개선 정도와 수행 속도는 지금과는 다른 형태를 띄게 될 것이며, 나아가 이웃해 생성 제한 횟수 k의 적정 값도 지금과는 다르게 선정될 것이다.

5. 결 론

본 연구에서는 하나의 본점에서 다수의 수요지점들의 수요량을 충족시키면서 차량의 운행 비용의 최소화를 목적으로 하여 차량의 운행 경로를 결정하는 전통적인 차량경로문제에 대한 발견적 해법을 개발하였다. 또한 개발한 해법을 기준 문제에 적용하여 해를 구하고, 동일한 문제에 대해 해를 도출하였던 기존의 연구 결과들과 비교하였다.

수요지점의 수가 50~120인 총 10개의 기준문제중 2개의 문제에서 기존연구의 최선해와 동일한 해를 매우 빠른 시간안에 얻었으며, 다른 기준 문제에 대해서도 tabu seach 해법이나 simulated annealing 해법 등 해의 질적인 측면만을 강조한 해법을 제외하고는 기존의 연구 결과에 비해 우수한 해를 구하였으며 수행 속도의 측면에서도 상대적으로 빠른 결과를 나타내고 있다. 또한 이웃해를 구성하는 기준과 이웃해 생성 횟수에 제한을 두어 해법의 해의 질이나 수행속도를 조절할 수 있도록 하였다.

추후 연구 과제로는 보다 우수한 이웃해를 가려낼 수 있는 기준에 대한 연구와 이웃해 생성 제한 횟수 k와 해의 개선의 관계에 대

해 심도 깊은 연구를 통해 보다 빠르면서 좋은 해를 보장하는 해법을 개발하는 것이다.

참 고 문 헌

[1] 신해웅, "혼합형 유전해법을 이용한 배송차량의 경로 결정," 박사 학위 논문, 한양대학교 대학원, 서울, 1994

[2] Dantzig, G. B. and J. H. Ramser, "The Truck Dispatching Problem," *Management Science*, Vol. 6, No. 1, pp.80~91, 1959

[3] Clarke, G. and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, Vol. 12, No. 4, pp.568~581, 1964

[4] Holmes, R. A. and R. G. Parker, "A Vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme," *Operations Research Quarterly*, Vol. 27, No. 1, pp.83~92, 1976

[5] Altinkemer, K. and B. Gavish, "Parallel Savings Based Heuristic for the Delivery Problem," *Operations Research*, Vol. 39, pp.456~469, 1991

[6] Osman, I. H., "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem," *Annals of Operations Research*, Vol. 41, pp.421~451, 1993

[7] Gendreau, M., A. Hertz and G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem," *Management Science*, Vol. 40, No. 10, pp.1276~1290, 1994

[8] Christofides, N. and S. Eilon, "AN Algorithm for the Vehicle Dispatching Problem," *Operational Research Quarterly*, Vol. 20, No. 3, pp.309~318, 1969

[9] Christofides, N., A. Mingozzi and P. toth, "The Vehicle Routing Problem," *Combinatorial Optimization*, N. Christofides, A. Mingozzi, P. Toth, and C. Sandi(Eds.), Wiley, Chichester, pp.315~338, 1979

[10] Gillett, B and L. Miller, "A Heuritic Algorithm for the Vehicle Dispatching Problem," *Operations Research*, Vol. 22, pp.340~349, 1974

[11] Mole, R. H. and S. R. Jameson, "A Sequential Routing-Building Algorithm Employing a Generalized Savings Criterion," *Operational Research Quartery*, Vol. 27, pp.503~511, 1976

[12] Fisher, M. L. and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing," *Networks*, Vol. 11, pp. 109~124, 1981

[13] Desrochers, M. and T. W. Verhoog, "A Matching Based Savings Algorithm for the Vehicle Routing Problem," Cahier du GERADG-89-04. Ecole des Hautes Etudes Commerciales de Montréal, 1989

[14] Pureza, V. M. and P. M. França, "Vehicle Routing Problems via Tabu Search Metaheuristic," Punlication CRT-747, Centre de recherche sur les transports, Montréal, 1991

[15] Hadjiconstantinou, E. and N. Christofides, "An Optimal Procedure for Solving Basic

Vehicle Routing Problems," presented at the 35th Annual Conference of Canadian Operational Research Society, Halifax, Canada, 1993

[16] Fisher, M. L., "Optimal Solution of Vehicle Routing Problems using Minimum K-Trees," *Operations Research*, Vol. 42, No. 4, pp.626~ 642, 1994

96년 1월 최초 접수, 96년 5월 최종 수정