발견적 승무계획 해법의 연구

A Heuristic Algorithm for Crew Scheduling Problems

김 정 식 *

Abstract

This paper presents a heuristic algorithm for a crew scheduling problem with dead head flights.

This paper modifies and improves saving method for finding the Multiple Salesman tours in the graph.

The results show that the computing time from this algorithm is implemented very much than those from general crew scheduling algorithms by set covering models.

1. 서 론

승무계획문제는 계획된 운항스케쥴을 기본으로 계획기간동안 모지기를 출발하여 일련의 구간을 승무한 후 모지기로 귀착할 때까지의 최적승무계획을 수립하는 것이다.

이 문제에서 소요되는 시간 및 비용을 최소화하는 것이 목적이며, 제약조건은 운항에 따른 조건 및 모지기와 기항지의 수, 편승구간 등이다.

본 연구는 보편적인 승무계획문제를 전통적인 수리계획모형과는 달리 복수 외판원모형과 비슷한 모형을 개발하여 적정 승무원수와 승무계획을 구함을 목적으로 한다.

이와 같은 경우 일반적으로 다음과 같은 Set Covering Solution 기법으로 다룬다.

$$\text{Min } Z = \sum_j C_j X_j$$

$$\text{s.t.} \sum_j a_{ij} X_j \geq 1, \quad V_i$$

단, X_j = 1 : Tour j가 선택
0 : 기타

a_{ij} = 1 : 구간 i가 Tour j에서 포함
0 : 기타

C_j = Tour j의 체류시간(비용)

* 청주대학교 산업공학과

Pierce & Lasky [1] 는 Tour를 가장 빠른 구간을 기준으로 정렬하고 초기실행가능해를 정하여 Branch and Bound Search 기법으로 해결하는 기법을 개발하였으며, Lemke [2] 는 선형계획법과 Branching 기법을 이용하여 Set Covering 문제를 해결하였으며 Javier Etcheberry [3] 는 Lagrangian Relaxation을 이용 목적함수의 하한치 L(u)를 구해 Subgradient Optimization을 적용 L(u)값을 가능한 한 증가시키는 u값을 효율적으로 구하였다.

위에서 소개된 기법들은 제약조건들을 고려하여 구간들을 조합하며 후보계획을 만들고 이에 대한 비용 등을 미리 계산한 후 각종 combinational 기법을 적용하여 해를 구하고 있다.

이 들에 비해 Baker [4] 는 차량운행문제를 이용한 heuristic 기법을 만들어 근사해를 효율적으로 구하였고 Giannessi and Nicoletti [4]는 Asymmetric directed graph 상에서 총 체류시간을 최소화시키는 복수외판원문제로 변환 후 이를 다시 일반외판원문제로 변형시켜 nearest unvisited vertex 해법으로 초기해를 구하고 배정문제해법을 이용, 총거리의 하한값을 구하는 분지 한계법으로 근사해를 구하고 있다.

2. Net—Work 모형

본 연구에서는 승무계획문제를 directed

Network로 표현하여 복수외판원모형으로 변형시키기로 하였다. 변화시의 제약은 다음과 같다고 가정한다.

(1) 연속승무시간은 10시간 이하이다.

(2) 승무후 승무시간의 1.5배 이상 체류 후 연결승무한다.

(3) 체류시간은 72시간 이내이다.

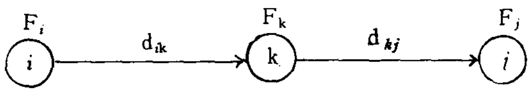
(4) Dead-Head(B팀)의 승무시간은 주승무(A팀)시간을 제외한 시간이다.

(5) 구간내에서 교대된 B팀은 체류시간외 5시간을 더 체류한다.

이에 따라 구간은 node로 표현하고 구간들 사이의 연결여부는 arc로 표현한다.

node의 집합 N에 대해서는 $N_i(i=2\cdots N)$으로 표현하고 각 구간의 승무시간을 기입한다.

다음 arc의 집합 A에 대해서는 N_i와 N_j가 조건에 합당하면 arc(i, j)를 만들며 이때의 값은 연결시 소요되는 체류시간을 기재한다. 또한 비승무 편승구간을 K로 연결한다.



단, F_i : 구간 i에서의 승무시간

d_{ik} : 구간 i에서의 구간 k로 연결시 체류시간

D/H : 편승시간

모지기 밖에서 연결 될시 d_{ij}는 다음과 같다.

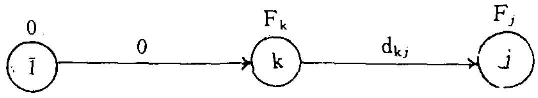
d_{ij} = (운항구간 j의 출발시각) − (운항구간 i의 도착시각)

단, d_{ik}는 F_i이상이며 $d_{ik}+d_{ki}$는 F_i의 1.5배이상

모기지에서 출발하는 경우에는 d_{ij}는

d_{ij} = (운항구간 j의 출발시각) − (비승무 편승구간 K의 출발시각)

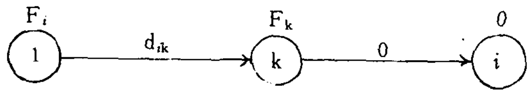
단, d_{kj}는 F_k 이상이야 함



모기지로 귀환하는 경우 d_{i1}을 다음과 같이 한다.

d_{i1} = (비승무구간 K의 도착시각) − (구간 i의 도착시각)

단, d_{ik}는 F_i이상이어야 한다.



D/H

상기와 같이 그림 1의 예제를 Network G(N, A)로 표현하면 그림 2와 같고 체류시간 행렬은 표 1과 같다.

복수외판원 문제를 해석한다는 것은 승무계획문제를 해석한다는 것과 같으므로 변환이 가능하나 이경우 편승운항구간이 문제가 되므로 node를 달리 처리해야 한다.

3. 편승구간

편승(Dead Head)은 10시간 이상의 승무시간이 소요되는 구간을 뜻한다. 이제까지 Set Covering 모형에서는 그림이 함께 승무한 것으로 간주하여 후보 Tour들을 generate시켜 처리하였으나 본 연구에

운항 구간표

운항구간	출발지	도착지	요일	출발 시간	도착 시간	운항 시간	비 고
2	서울	HNL	일	11 : 30	20 : 00	8 : 30	
3	서울	L.A	월	9 : 30	20 : 00	11 : 00	Dead Heading
4	HNL	L.A	월	22 : 00	화)3 : 00	5 : 00	
5	L.A	HNL	화	17 : 00	22 : 30	5 : 30	
6	HNL	서울	수	8 : 00	17 : 00	9 : 00	Dead Heading
7	L.A	서울	수	10 : 00	21 : 00	11 : 00	

(주) 1) 각 승무원팀의 주간 최대 승무시간 U=15시간.

2) 연속 승무시간은 10시간 이하이며, 승무 후 승무시간의 1.5배 이상 체류 후 연결 승무.

3) 체류시간은 72시간 이내.

4) 운항구간 내에서 승무가 교대된 편승팀은 원래 체류시간외 5시간을 연장 체류.

※ 시간 : GMT(Greenwich Mean Time)

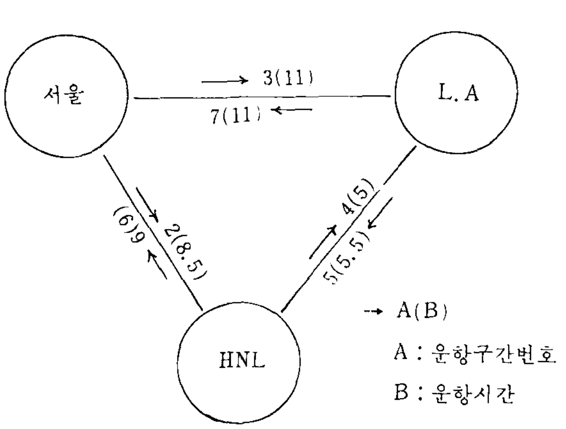


그림 1. 운항노선

서는 dead head 구간에 해당하는 node를 각각 팀수대로 2개의 node로 나누어서 1개의 node에 1개팀만 승무하도록 조정하여 확장된 Network G(N', A')상 표현하여 처리하고자 한다. 편승구간을 두 개의 node로 조정하여 확장시키면 node수는 dead head 구간수 만큼 acr수는 node r_B에 해당하는 확장된 node들과 연결이 가능해진 arc수 만큼 확장된 비대칭의 유방향 Network G(N', A')가 된다.

상기한 방법에 의해 그림 3과 같이 편승구간을 조정한 확장된 그래프상의 복수외판원 모형으로 변환되고 체류행렬시간은 표 2와 같이 변환되어 적정승무계획의 집합을 보다 효율적으로 구할 수 있다.

4. 새로운 Heuristic 기법

전 장에서와 같이 편승구간을 처리한 복수외판원 문제는 Vehicle Routine Problem과 같은 문제가 되므로 각 수요처의 수요와 각종 조건을 만족시키면서 총 비용이 최소로 되는 최적경로를 구하고 적정차량댓수를 구하는 문제로 변혁 가능하다. 이 경우 Saving Approach가 발견적 기법들 중에서 많이 이용되고 있다.

복수외판원 모형을 이용한 Savings Algorithm은 다음 단계와 같이 처리한다.

단계 1 : 편승구간을 고려하여 확대한다(체류시간행렬 및 승무시간수정)

단계 2 : Saving list 작성

이때의 Savings를 Descending Order로 정리한다.

단계 3 : Best Savings의 arc을 선택한다.

$$S_{ij} = \frac{\max}{ij} [S_{ij}$$

단계 4 : node i와 j를 Tour로 연결

단계 5 : 계획의 형성여부를 점검

단계 6 : 짧은 Tour들을 연결 최종 Tours를 결정한다.

단계 7 : 최적해를 산출한다.

상기한 수준으로 각 팀의 주간 최대승무시간 28시간, 체류시간 $D=(d_{ij})$을 행렬 $D^0=(d_{ij}^0)$로 만든 후 여기에 dead head 구간을 조정하여 새로운 체류시간행렬 $D=(d_{ij})$을 구하고 이에 따른 절약 list는 표 3과 같고, 후보 Tours의 집합은 표 4와 같고, 최적해는 표 5와 같다.

따라서 다음과 같은 결론을 얻었다.

첫째로 본 예제에서는 Savings의 Descending Order에 따라 Savings가 가장 큰 Arc부터 연결하였다. 그리고 둘째로 후보 Tours의 집합에서 보신 바와 같이 No 1~No 6 이외에는 경로가 형성되지 않는다.

마지막으로 결과식에서 경로 1-3-7-1이 중복되는 것은 정상적인 운항구간표가 아니고, 임의로 만든 가상 운항구간표로 인하여 결과가 중복되었다.

표 1. 체류시간 행렬 $[D^0=(d_{ij}^{\,0})]$

$i \backslash j$	1	2	3	4	5	6	7	운항구간	승무시간
1		0	0	34.5	32	68.5	49	1	
2	69			26	45	60	62	2	8.5
3	49				21	36	38	3	11
4	42				14	29	31	4	5
5	18.5					9.5		5	5.5
6	0							6	9
7	0							7	11

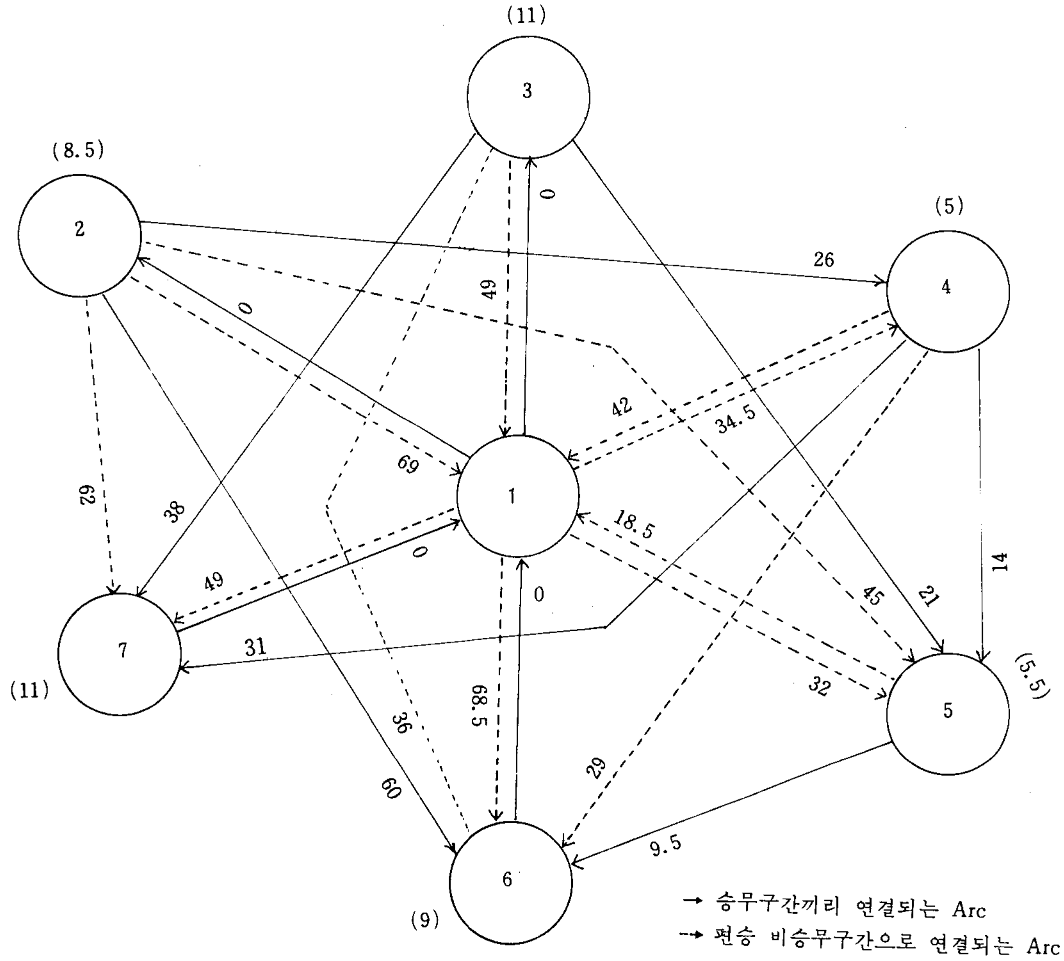


그림 2. Graph로 전환된 승무계획모형

표 2. 편승구간을 조정 확장한 네트워크상의 체류 시간 행렬 D=(d_{ij})

i \ j	1	2	3	31	4	5	6	7	71	운항구간	승무시간
1		0	0	0	34.5	32	68.5	49	54	1	
2	69				26	45	60	62	67	2	8.5
3	49					21	36	38	43	3	10
31	49					21	36	38	43	31	1
4	42					14	29	31	36	4	5
5	18.5						9.5			5	5.5
6	0									6	9
7	0									7	10
71	0									71	1

(빈칸 : +M), M>0큰수, Node 상의수 : 승무시간 Arc상의수 : 체류시간

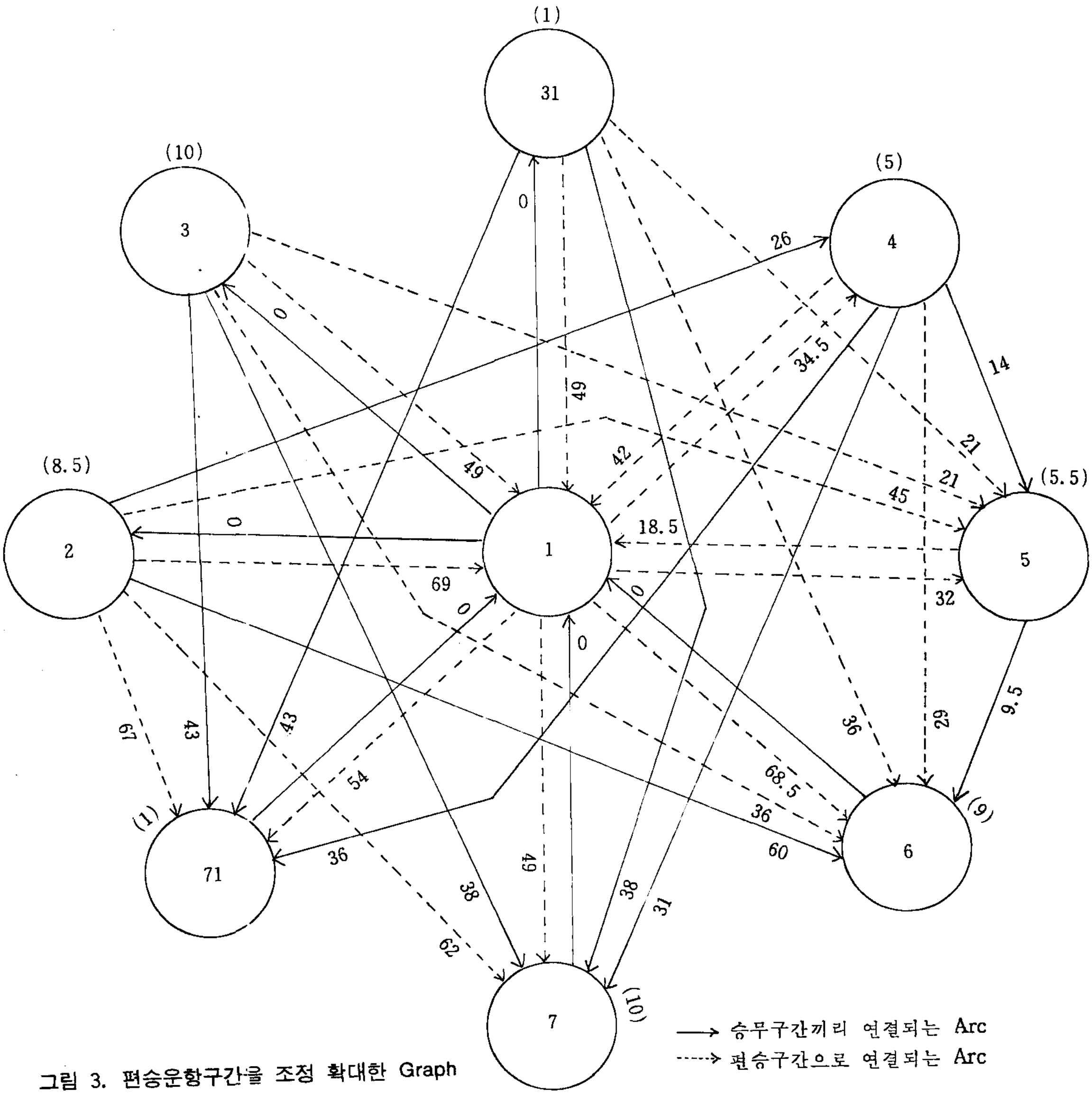


그림 3. 편승운항구간을 조정 확대한 Graph

표 3. Savings List

$S_{ij} = d_{i1} + d_{1j} - d_{ij}$

i \ j	1	2	3	31	4	5	6	7	71
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	1.000				77.5	56	77.5	56	56
3	1.000					60	81.5	60	60
31	1.000					60	81.5	60	60
4	1.000					60	81.5	60	60
5	1.000						77.5		
6	1.000								
7	1.000								
71	1.000								

(빈칸 : minus 부호의 상수값)

Savings의 Descending Order

순서	Arc	Savings	순서	Arc	Savings
1	3−6	81.5	10	31−5	60
2	31−6	81.5	11	31−7	60
3	4−6	81.5	12	31−71	60
4	2−4	77.5	13	4−5	60
5	2−6	77.5	14	4−7	60
6	5−6	77.5	15	4−71	60
7	3−5	60	16	2−5	56
8	3−7	60	17	2−7	56
9	3−71	60	18	2−71	56

표 4. 후보 Tours의 집합

No. 1 최대승무시간 : U=15시간

Tour	경 로	체류 시간	승무 시간	비 고
1	1−4−6−1	34.5+29+0=63.5	5+9=14	
2	1−3−5−1	0+21+19.5=39.5	10+5.5=15.5	infeasible
3	1−31−7−1	0+38+0=38	1+10=11	
4	1−2−71−1	0+67+0=67	8.5+1=9.5	
시간 :		28		

No. 2

Tour	경 로	체류 시간	승무 시간	비 고
1	1−4−6−1	34.5+29+0=63.5	5+9=14	
2	1−3−5−1	0+21+18.5=39.5	10+5.5=15.5	infeasible
3	1−31−71−1	0+43+0=43	1+1=2	
4	1−2−7−1	0+62+0=62	18.5	
시간 :		208		

No. 3

Tour	경 로	체류 시간	승무 시간	비 고
1	1－4－6－1	34.5＋29＋0＝63.5	5＋9＝14	
2	1－3－7－1	0＋38＋0＝38	10＋10＝20	infeasible
3	1－31－5－1	0＋21＋18.5＝39.5	1＋5.5＝6.5	
4	1－2－71－1	0＋67＋0＝67	8.5＋1＝9.5	
시간：		208		

No.4

Tour	경 로	체류 시간	승무 시간	비 고
1	1－4－6－1	34.5＋29＋0＝63.5	5＋9＝14	
2	1－3－7－1	0＋38＋0＝38	10＋10＝20	infeasible
3	1－31－71－1	0＋43＋0＝43	1＋1＝2	
4	1－2－5－1	0＋45＋18.5＝63.5	8.5＋5.5＝14	
시간：		208		

No. 5

Tour	경 로	체류 시간	승무 시간	비 고
1	1－4－6－1	34.5＋29＋0＝63.5	5＋9＝14	
2	1－3－71－1	0＋43＋0＝43	10＋1＝11	infeasible
3	1－31－5－1	0＋21＋18.5＝39.5	1＋5.5＝6.5	
4	1－2－7－1	0＋62＋0＝62	8.5＋10＝18.5	
시간：		208		

No. 6

Tour	경 로	체류 시간	승무 시간	비 고
1	1－4－6－1	34.5＋29＋0＝63.5	5＋9＝14	feasible
2	1－3－71－1	0＋43＋0＝43	10＋1＝11	D/H：7
3	1－31－7－1	0＋38＋0＝38	1＋10＝11	D/H：3
4	1－2－5－1	0＋45＋18.5＝63.5	8.5＋5.5＝14	
시간：		208		

표 5. 적정해의 산출

Tour	경 로	체류 시간	승무 시간	비 고
1	1－4－6－1	63.5	14	
2	1－3－71－1	43	11	D/H：7
3	1－31－7－1	38	11	D/H：3
4	1－2－5－1	63.5	14	
시간：		208	50	

총 체류시간＝208시간

운항 경로：1－4－6－1

1－3－71－1

1－31－7－1

1－2－5－1

원래 Graph 상에서의 운항 경로는

1－4－6－1

1－3－7－1 (편승구간：7)

1－3－7－1 (편승구간：3)

1－2－5－1

적정 승무원 팀수：m＝4

5. 결 론

승무계획문제는 모기지를 출발한 후 각 기지에서 다음 구간을 탑승하기 까지 체류에 소요되는 제반조건들을 최소화하는 문제로서 본 연구에서는 복수의 판원 모형을 사용하여 적정승무계획과 승무원팀 수를 산출하는 발견적 기법을 개발하였다.

이 기법을 더 발전시키면 차량운행문제 등에 활용할 수 있으리라 사료된다.

앞으로의 연구방향은 Savings Approach에 의해 구하여진 해를 효율적(Dynamic Approahc)으로 개선하는 방법을 모색하는 것이 될 것이다.

참 고 문 헌

1. C.E. Lemke, H.M. Salkin, and K. Spielberg, "Set Covering by single Branch Enumeration with Linear Programming Subproblems", Oper. Res, 19, 498−1022(1971)
2. E.K. Baker, L.D.Bodin, W.F. Finnegan, and R.J. Ponder, "Efficient Heuristic Solutions to Airline Crew Scheduling Problem", AIEE Trans, 79−85(1979)
3. J. Etcheberryy, "The Set covering Problem : A New Implict Enumeration Algorithm" Oper. Res, 25, 760−772(1977)
4. J.F. Pierce "Applications of Combinatorial Programming to a Class of All–Zero–One Integer Programming Problems", Manage, sci, 15, 191−209(1968)