Journal of The Korea Society of Computer and Information
Vol. 20 No. 11, pp. 69-75, November 2015
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.11.069

A Heuristic Polynomial Time Algorithm for Crew Scheduling Problem

Sang-Un Lee*

Abstract

This paper suggests heuristic polynomial time algorithm for crew scheduling problem that is a kind of optimization problems. This problem has been solved by linear programming, set cover problem, set partition problem, column generation, etc. But the optimal solution has not been obtained by these methods. This paper sorts transit costs c_{ij} to ascending order, and the task i and j crew paths are merged in case of the sum of operation time Σo is less than day working time T. As a result, we can be obtain the minimum number of crews $_{\min}K$ and minimum transit cost $z={}_{\min}c_{ij}$. For the transit cost of specific number of crews $K(K>{}_{\min}K)$, we delete the maximum c_{ij} as much as the number of $K-{}_{\min}K$, and to partition a crew path. For the 5 benchmark data, this algorithm can be gets less transit cost than state-of-the-art algorithms, and gets the minimum number of crews.

▸ Keywords : Tasks, Crews, Transit cost, Ascending sort, Path merge and partition

• First Author: Sang-Un Lee, Corresponding Author: Sang-Un Lee
*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Engineering, Gangneung-Wonju National University
• Received: 2015. 08. 07, Revised: 2015. 08. 27, Accepted: 2015. 09. 21.

I. Introduction

승무원 근무계획 문제 (crew scheduling problem, CSP)는 실세계에 있는 항공기나 버스, 철도, 지하철 등과 같은 대량수송 수단에서 빈번하게 발생하고 있다[1]. 각 수송 수단별로 특수한 제약사항들이 다르게 적용되지만 본 논문에서는 포괄적으로 적용할 수 있는 공통된 제약사항만을 다룬다.

CSP를 풀기 위해 2가지 방법이 적용되고 있다. 첫 번째로는 초기 승무원 계획 후보들을 생성하고, 집합 피복 문제 (set cover problem, SCP)나 집합 분할 문제 (set partitioning problem, SPP)로 최소 비용 부분집합을 선택하고, 분기한정법 (branch-and-bound, BB)에 기반한 선형계획법 (linear programming, LP)로 해를 구한다. 두 번째로는 선형계획 완화법 (LP relaxation, LPR)으로 얻은 초기 후보 승무원 계획에 대해 열 생성법 (column generation, CG)을 적용한다. 이 2가지 방법을 혼용한 연구 결과들도 있다.

Falkner와 Ryan[2]과 Graves et al.[3]는 SPP를, Gershkoff[4]은 SPP와 BB를, Hoffman과 Padberg[5]는 SPP와 LPR을, Cranic과 Rousseau[6], Desrochers와 Soumis[7], Lavoie et al.[8]는 SCP와 CG를, Smith[9]은 SCP를 Cheddad[10]는 LP와 LPR을 적용하였다.

또 다른 방법으로, Ball et al[11]와 Ball과 Roberts[12]는 초기비용 매칭의 일련순서로, Martello와 Toth[13]는 0-1 정수계획법을, Beasley와 Cao[1]는 1996년에 포괄적인 CSP에 대해, 0-1 정수 선형계획법과 BB를 적용한 트리 탐색법 (tree search algorithm, TSA)을 제안하였다.

이와 같이 다양한 방법으로 CSP를 해결하고자 하였으나 1996년 이후에는 보다 심도 있는 연구가 수행되지 않고 있으며, 최근 들어 Qiao[14]는 상자 채우기 (bin packing)법을 적용하였으며, Chen과 Niu[15]는 2차원 정수배열 부호화 기법을 적용하기도 하였다.

CSP는 다항시간으로 최적 해를 찾는 알고리즘이 알려져 있지 않아 NP-완전 (non-deterministic polynomial time-complete) 문제로 분류되고 있다.

본 논문에서는 CSP에 대해 $O(m \log m)$의 다항시간으로 최적 해를 찾을 수 있는 휴리스틱 알고리즘을 제안한다. 2장에서는 CSP의 개념을 고찰해 본다. 3장에서는 최적 해를 다항시간으로 찾을 수 있는 휴리스틱 알고리즘을 제안한다. 4장에서는 OR-LIB[16]에서 인용된 5개 데이터에 대해 제안된 알고리즘의 적합성을 검증해 본다.

II. Description of CSP

승무원 근무계획 문제 (CSP)에 대한 개념은 Beasley와 Cao[1]에서 인용된 그림 1에서 알 수 있듯이 다음과 같이 정의된다.

(1) $K(K \le n)$명의 승무원이 수행하는 n개의 업무 (tasks)가 있다. 각 승무원은 자신의 근무일 (1일 8시간)의 근무 시작과 종료를 동일한 차고 (depot)에서 수행한다.

(2) 각 승무원은 차고로부터 업무 i를 수행하는데 필요한 이동시간 b_i과 이동 비용 c_{oi}이 발생한다. 각 업무 i는 수행 비용 d_i, 시작시간 s_i, 종료시간 f_i와 수행시간 $o_i = f_i - s_i$를 갖고 있다.

(3) 업무 i를 수행하고, 이어서 시간적으로 수행이 가능한 다음 업무 j로 전이되는 경우 업무 전환비용 c_{ij}가 발생하며, 전환이 가능한 경우 수는 m개가 있다.

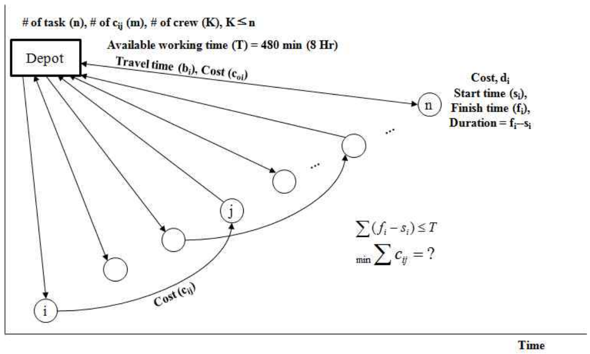


Fig. 1. Crew Scheduling Problem

CSP는 $K \le n$명의 승무원으로 n개의 업무를 수행하기 위해 식 (1)의 최소 업무전환비용 z를 충족하는 K개의 승무원 경로 (crew path)인 업무 집합을 형성해야한다. 여기서, 승무원 경로는 동일 승무원이 다수의 업무를 순차적으로 수행하는 경로이며, Σo_i는 일일 업무시간 $T = 480\min$을 초과할 수 없는 제약조건을 갖는다[1].

$$z = {}_{\min}\Sigma c_{ij} \tag{1}$$

where $\Sigma o_i \le T$, for each crew i

III. Polynomial-time algorithm for CSP

본 장에서는 승무원 일정계획 문제의 최적 해를 $O(m \log m)$의 다항시간으로 찾을 수 있는 알고리즘을 제안한다. 먼저, 각 업무의 수행시간 (기간)을 o_i를 계산하고, 업무별 전이비용을 오름차순으로 정렬한다. 오름차순으로 정렬된 전이비용 순으로 $\Sigma o_i + \Sigma o_j \le T$ 즉, i와 j 작업 그룹의 수행비용 합이 일일 업무시간 한계치 $T = 480\min(=8\,hour)$를 초과하지 않는 경우 한 운전기사가 i와 j의 작업 그룹을 수행하는 방식을 제안한다. 이와 같이 수행하면 최소로 필요한 운전기사 인원수를 구할 수 있으며, 보다 적은 업무전환비용으로 보다 많은 인원을 운영하고자 할 경우, 업무 전환비용이 가장 많은 호를 절단하여 업무를 분할하면 된다. 제안된 알고리즘을 CSPA라 하며 그림 2와 같이 수행된다. 제안된 CSPA는 단순히 $O(m \log m)$의 다항시간으로 수행됨에도 불구하고, 최소 승무원 수 ${}_{\min}K$를 구할 수 있을 뿐 아니라 $K = [{}_{\min}K, n]$의 각 승무원 수에 대한 최적 해 z도 함께 구할 수

있는 특징을 갖고 있다.

Step 1. for $i=1$ to n /* 수행 복잡도 : $O(n)$ */
각 업무 (tasks)의 시작시간 t_s와 종료시간 t_e에 대한 수행시간 $o_i=t_e-t_s$계산.
end
Step 2. for $j=1$ to m /* 수행 복잡도 : $O(m\log m)$ */
c_{ij}를 오름차순으로 정렬.
end
Step 3. for $j=1$ to m /* 수행 복잡도 : $O(m)$ */
i와 j 작업 그룹의 o_i와 o_j에 대해 $\Sigma o_i+\Sigma o_j \le T$이면 호 (i,j)를 연결하고, c_{ij}를 저장.
end
연결되거나 독립적인 작업 수= $_{\min}K$ (최소로 필요한 승무원 수), Σc_{ij}= $_{\min}K$ 승무원으로 운영되는 최소 비용.
Step 4. for $i={}_{\min}K$ to n
Step 3에서 연결된 c_{ij}들 중에서 $_{\max}c_{ij}$를 삭제하면서 $i=i+1$로 치환.
end
남은 Σc_{ij}는 해당 승무원 수 $K=i$로 수행시 최소 비용.

Fig. 2. Crew scheduling problem algorithm (CSPA)

제안된 알고리즘이 이와 같이 최적 해를 구할 수 있는지 검증하기 위해, OR-LIB[14]에서 인용된 표 1의 csp50 데이터에 대해 CSPA를 적용하여 보자.

Table 1. csp50 instance

Number of tasks n = 50, Time limit T= 480, $m=173$															
Task No.	시작 시간	종료 시간	Task i	Task j	전환 비용	Task i	Task j	전환 비용	Task i	Task j	전환 비용	Task i	Task j	전환 비용
1	1	144	1	10	169	10	21	82	19	25	48	35	40	232
2	14	217	1	11	131	10	23	245	19	28	190	35	43	568
3	60	212	1	13	286	10	25	163	19	29	249	35	44	457
4	97	118	1	16	371	10	26	303	19	31	301	35	47	667
5	134	365	1	17	457	10	27	327	19	32	367	36	43	294
6	138	325	1	18	409	10	28	311	19	33	507	36	44	228
7	143	272	2	13	155	11	22	26	20	32	286	36	47	339
8	144	364	2	17	346	11	27	142	21	32	220	37	44	182
9	220	255	2	18	300	11	28	208	21	35	412	37	47	280
10	231	400	3	13	219	11	29	200	22	35	298	38	42	218
11	264	478	3	14	245	12	24	23	23	33	124	38	45	268
12	301	531	3	16	274	12	27	83	23	34	193	38	46	392
13	324	444	3	17	319	12	29	233	23	35	249	38	47	493
14	342	534	3	18	360	12	31	320	24	34	181	39	41	68
15	402	576	3	19	347	13	23	128	24	35	265	39	46	199
16	405	465	4	10	187	13	24	209	25	31	160	39	47	254
17	410	462	4	11	178	13	25	170	25	33	304	40	46	132
18	414	463	4	12	287	13	26	175	25	34	287	40	47	165
19	419	507	4	13	288	13	27	184	25	35	435	40	48	244
20	445	606	4	14	390	13	28	280	26	33	104	40	49	182
21	461	652	4	15	349	13	29	238	26	35	273	40	50	218
22	499	724	4	16	529	13	31	443	27	33	86	41	50	21
23	541	693	4	17	462	13	32	394	27	34	279	47	50	27
24	553	747	4	18	586	14	24	22	27	35	308			
25	554	615	4	19	471	14	31	192	27	38	481			
26	574	689	5	19	77	14	32	410	28	35	364			
27	576	708	6	15	130	15	32	301	28	36	409			
28	632	691	6	16	121	15	33	205	28	38	361			
29	656	726	6	17	109	16	23	137	29	34	226			
30	674	786	6	25	309	16	24	144	29	35	171			
31	711	765	7	15	148	16	25	125	29	38	329			
32	766	791	7	16	203	16	29	217	29	39	374			
33	775	882	7	17	154	16	30	278	30	35	101			
34	871	951	7	18	170	16	31	305	30	36	151			
35	886	927	7	19	254	16	32	410	30	38	419			
36	903	1104	7	20	307	16	33	374	30	39	340			
37	932	1168	7	25	502	17	22	58	31	35	171			
38	997	1055	8	17	91	17	23	152	31	38	334			
39	1023	1131	8	25	324	17	25	150	31	39	282			
40	1156	1190	9	12	61	17	27	181	31	40	750			
41	1195	1370	9	15	203	17	29	232	32	35	121			
42	1216	1445	9	16	236	17	31	410	32	36	167			
43	1259	1308	9	17	297	17	32	552	32	38	321			
44	1261	1298	9	19	285	17	33	511	32	39	298			
45	1278	1405	9	20	342	18	24	170	32	40	707			
46	1297	1433	9	21	303	18	25	97	33	40	490			
47	1333	1362	9	23	427	18	28	206	34	43	545			
48	1340	1543	9	25	560	18	31	486	34	44	528			
49	1351	1539	9	26	571	18	32	343	35	38	96			
50	1388	1528	9	28	713	18	33	556	35	39	164			

csp50 데이터는 $n=50, m=173, T=480\text{min}$이다. 여기서는 모든 이동시간과 비용을 고려하지 않고 있으며 단지 업무 i를 수행 완료 한 후 업무 j를 시작할 수 있는 가능한 경우에 대해서만 m개의 호 (i,j)의 비용 $c_{ij}(i,j)$만을 제시하고 있다.

Beasley와 Cao[1]는 분기한정법(BB)에 기반한 트리 탐색 알고리즘 (TSA)을 적용하여 $K=[27,31]$의 5 구간 승무원 수에 대한 최소 업무전환비용 z만을 구하였다. 반면에, 제안된 CSPA로 수행한 결과는 그림 3과 같이 $_{\min}K=20$, $\Sigma c_{ij}=4,503$을 얻었다. 여기서 K는 승무원 수, Σc_{ij}는 한 업무를 수행한 후 다른 업무로 전환하는 경우의 소요 비용 합으로 C01의 68+21=89, C02는 22+181=203, ... 으로 C15까지의 업무 전환 비용 (호의 가중치)을 모두 더한 값이다. 결론적으로, CSPA는 csp50에 대해 최소 20명의 운전기사로도 1일 운행 일정을 충족시킬 수 있음을 보였다. 반면에, Beasley와 Cao[1]의 TSA는 최소 27명이 소요됨을 보여 인력운영 효율성이 CSPA에 비해 좋지 않음을 알 수 있다.

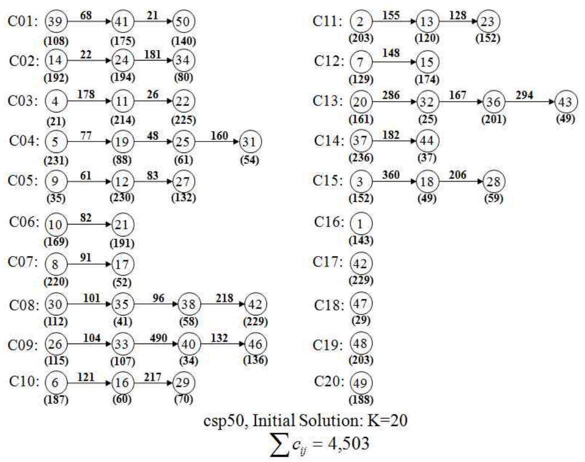


Fig. 3. Result of CSPA for csp50

CSPA를 Beasley와 Cao[1]가 제시한 TSA의 $K=[27,31]$과 비교하기 위해 $K=[21,31]$에 대해 삭제되는 호의 c_{ij}를 그림 4에 표현하고 있다.

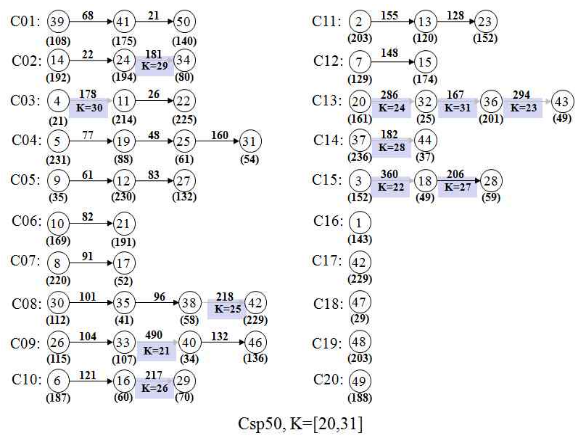


Fig. 4. $K=[21,31]$ for csp50

표 2에서는 csp50에 대해, 그림 3의 결과로부터 $K=[20,50]$의 운전기사 인원수 증가에 따라 삭제 (절단)되는 c_{ij}로 업무가

분할되는 경우와 업무전환비용 합 Σc_{ij}를 나타내고 있다. 즉,

Table 2. Σc_{ij} of $K=[20,50]$ for cap50 data

	K=20	K=21	K=22	K=23	K=24	K=25	K=26	K=27	K=28	K=29	K=30	K=31	K=32	K=33	K=34	K=35	K=36	K=37	K=38	K=39	K=40	K=41	K=42	K=43	K=44	K=45	K=46	K=47	K=48	K=49	K=50
1	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68	68						
2	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	
3	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22		
4	181	181	181	181	181	181	181	181	181																						
5	178	178	178	178	178	178	178	178	178	178																					
6	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26	26			
7	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77	77							
8	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48				
9	160	160	160	160	160	160	160	160	160	160	160	160																			
10	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61					
11	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83	83									
12	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82	82								
13	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91	91										
14	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101												
15	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96	96											
16	218	218	218	218	218																										
17	104	104	104	104	104	104	104	104	104	104	104	104	104	104	104	104	104	104													
18	490																														
19	132	132	132	132	132	132	132	132	132	132	132	132	132	132	132																
20	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121	121														
21	217	217	217	217	217	217																									
22	155	155	155	155	155	155	155	155	155	155	155	155	155																		
23	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128	128															
24	148	148	148	148	148	148	148	148	148	148	148	148	148	148																	
25	286	286	286	286																											
26	167	167	167	167	167	167	167	167	167	167	167																				
27	294	294	294																												
28	182	182	182	182	182	182	182	182																							
29	360	360																													
30	206	206	206	206	206	206	206																								
Σc_{ij}	4,503	4,013	3,653	3,359	3,073	2,855	2,638	2,432	2,250	2,069	1,891	1,724	1,564	1,409	1,261	1,129	1,001	880	776	675	579	488	405	323	246	178	117	69	43	21	0
${}_{\max}c_{ij}$	490	360	294	286	218	217	206	182	181	178	167	160	155	148	132	128	121	104	101	96	91	83	82	77	68	61	48	26	22	21	

이와 같이 단순히 남은 업무전환비용들 중에서 최대 업무전환비용 호를 가진 두 작업을 분할할 때마다 인원수가 1명씩 증가하며 이 경우 삭제되는 최대 업무전환비용만큼 비용을 절감할 수 있다.

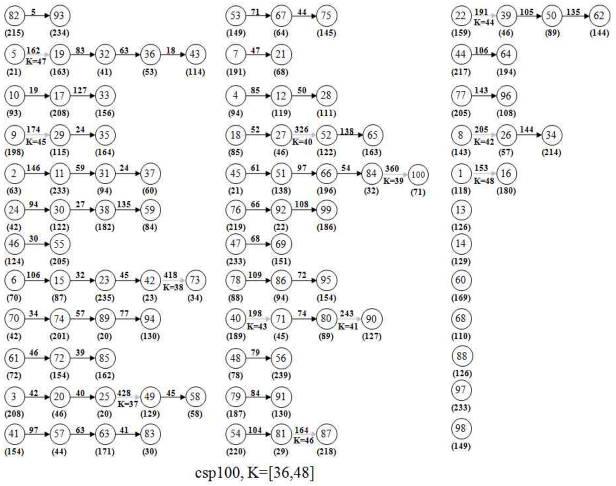
Beasley와 Cao[1]의 TSA는 $K=[27,31]$의 인원수에 대한 업무전환 소요비용을 각각 3129, 2706, 2399, 2092와 1872를 얻었는데 반해 제안된 CSPA는 각각 2432, 2250, 2069, 1891과 1724를 얻어 Beasley와 Cao[1]의 TSA에 비해 업무전환비용을 각각 697, 456, 330, 201과 148을 감소시킬 수 있었다.

Jeroslow[17]에 따르면 0-1 정수계획법의 BB는 $2^{(m+1)/2}$회의 순방향과 역방향 탐색을 수행해야 해를 얻을 수 있으나 제안된 알고리즘은 $O(m\log m)$의 오름차순 정렬 후 m회만으로 해를 구할 수 있어 보다 단순한 방법임을 알 수 있다.

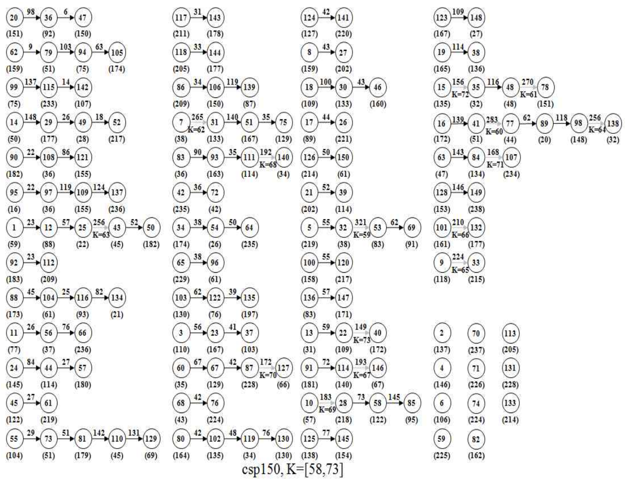
IV. Experimental Results and Analysis

본 장에서는 OR-LIB[14]에서 인용된 csp100, csp150, csp200과 csp250에 대해 제안된 CSPA를 적용하여 본다. 제안된 CSPA를 실험 데이터들에 적용한 결과는 그림 5와 같다. csp 100의 ${}_{\min}K=36$, csp150의 ${}_{\min}K=58$, csp200의 ${}_{\min}K=79$, csp250의 ${}_{\min}K=90$을 얻었으며, 각 승무원의 경로를 보여주고 있다. 또한, Beasley와 Cao[1]가 구한 ${}_{\max}K$ 승무원까지의 해를 얻기 위해 삭제되는 호를 표시하였다.

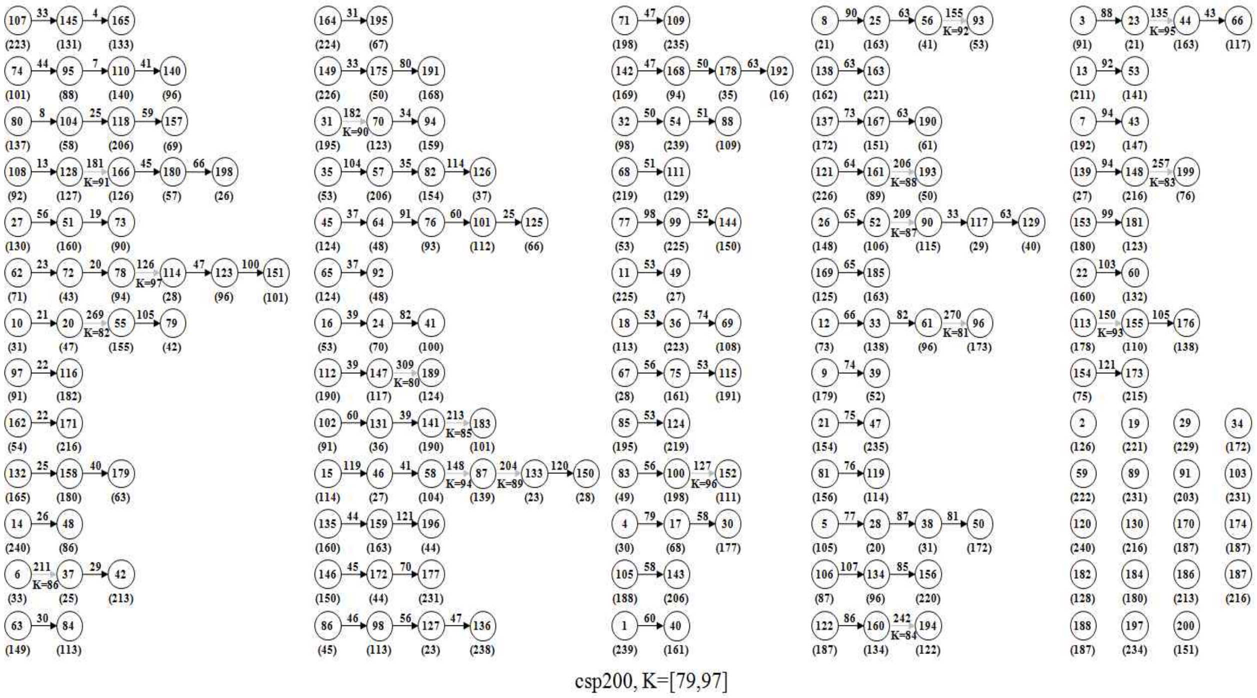
표 3에서는 Beasley와 Cao[1]의 TSA와 제안된 CSPA의 성능을 비교한 결과를 제시하고 있다.



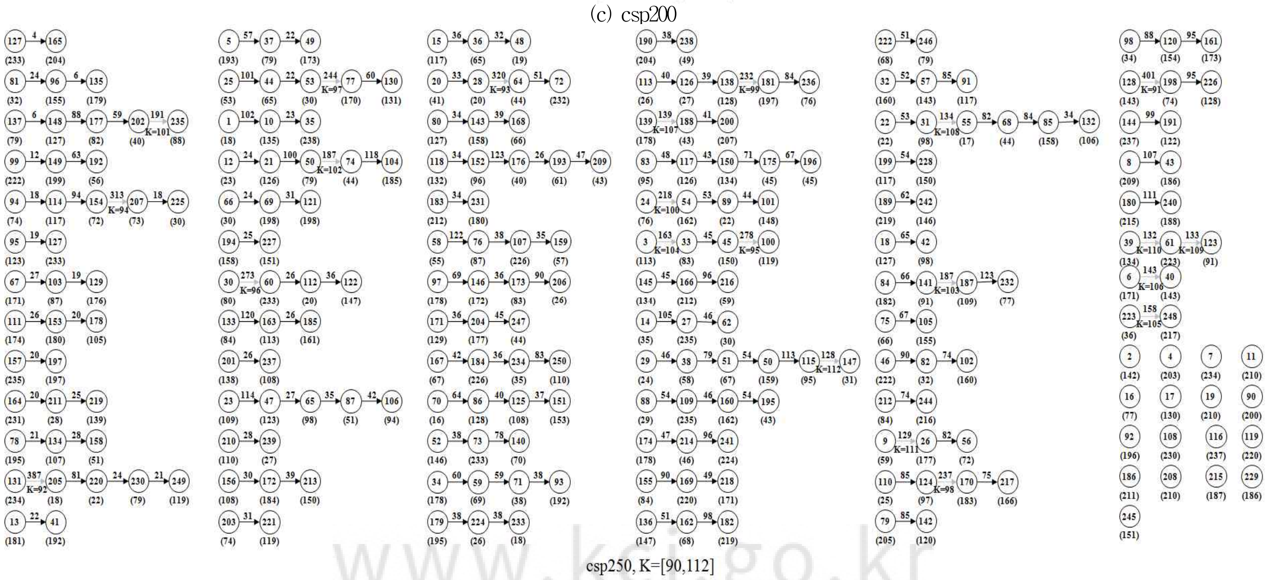
(a) csp100



(b) csp150



(c) csp200



(d) csp250

Fig. 5. CSPA for Experimental data

Beasley와 Cao[1]의 TSA는 특정 5개 승무원 구간에 대해서만 해를 구하였는데 반해 제안된 CSPA는 $_{\min}K$ 부터 Beasley와 Cao[1]의 $_{\max}K$까지의 해를 제시하고 있다. 예로, csp50 문제는 $n=50$회/1일의 버스 운행 일정이 계획되어 있으며, $m=173$의 가능한 차량간 운전기사 이동 (전환)이 가능하다. 여기서 K는 n회 차량 운행일정을 K명의 운전기사가 담당함을 의미하며, 차량 전환 비용 Σc_{ij}는 운전기사가 차량을 바꾸어 운행하는 경우의 이동 에 소요되는 비용이다. 삭제 호 $c_{ij}(i,j)$는 어느 운전기사의 작업량을 분할하여 추가되는 운전기사에게 배정할 것인가를 나타내고 있다. 여기서, TSA와 CSPA의 Σc_{ij}를 비교하면 CSPA가 보다 저렴한 업무전환 소요비용(비생산적 비용)으로 일정계획을 수립하였음을 알 수 있다.

따라서 본 논문에서는 CSP에 대해 최적 해를 구할 수 있는 휴리스틱 다항시간 알고리즘이 존재함을 보였다.

V. Conclusions

CSP는 최적 해를 다항시간으로 찾을 수 있는 규칙을 가진 휴리스틱 알고리즘이 제안되지 않고 있어 NP-완전인 난제로 분류되고 있다.

이 문제는 버스, 철도, 항공, 선박 등 다양한 교통수단에 대한 운행계획표를 경제적으로 작성하는데 필수적으로 적용되는 문제로, 급박한 변경이 발생하면 최적 해를 다항시간으로 찾는 알고리즘이 절실히 요구된다.

Table 3. Comparison of Algorithm's Performance

문제	n	m	K	알고리즘 TSA Σc_{ij}	알고리즘 CSPA Σc_{ij}	CSPA 삭제 호 $c_{ij}(i,j)$
csp50	50	173	20	-	**4,503**	
			21	-	4,103	490(33,40)
			22	-	3,653	360(3,18)
			23	-	3,359	294(36,43)
			24	-	3,073	286(20,32)
			25	-	2,855	218(38,42)
			26	-	2,638	217(16,29)
			27	3,139	**2,432**	206(18,28)
			28	2,706	**2,250**	182(37,44)
			29	2,399	**2,069**	181(24,34)
			30	2,092	**1,891**	178(4,11)
			31	1,872	**1,724**	167(32,36)
csp100	100	715	36	-	**6,376**	
			37	-	6,308	428(25,49)
			38	-	5,890	418(42,73)
			39	-	5,530	360(84,100)
			40	-	5,204	326(27,52)
			41	-	4,961	243(80,90)
			42	-	4,756	205(8,26)
			43	-	4,558	198(40,71)
			44	4,812	**4,367**	191(22,39)
			45	4,514	**4,193**	174(9,29)
			46	4,310	**4,029**	164(81,87)
			47	4,107	**3,867**	162(5,19)
			48	3,905	**3,714**	153(1,16)
csp150	150	1,355	58	-	**8,363**	
			59	-	8,042	321(32,53)
			60	-	7,759	283(41,77)
			61	-	7,489	270(48,78)
			62	-	7,224	265(7,31)
			63	-	6,968	256(25,43)
			64	-	6,712	256(98,138)
			65	-	6,488	224(9,33)
			66	-	6,278	210(101,132)
			67	-	6,085	193(114,146)
			68	-	5,893	192(111,140)
			69	6,275	**5,710**	183(10,28)
			70	5,999	**5,538**	172(87,127)
			71	5,754	**5,370**	168(84,107)
			72	5,551	**5,214**	156(15,35)
			73	5,347	**5,065**	149(22,40)
csp200	200	2,543	79	-	9,817	
			80	-	9,508	309(147,189)
			81	-	9,238	270(61,96)
			82	-	8,969	269(20,55)
			83	-	8,712	257(148,199)
			84	-	8,470	242(160,194)
			85	-	8,257	213(141,183)
			86	-	8,046	211(6,37)
			87	-	7,837	209(52,90)
			88	-	7,631	206(161,193)
			89	-	7,427	204(87,133)
			90	-	7,245	182(31,70)
			91	-	7,064	181(128,166)
			92	-	6,909	155(56,93)
			93	6,914	6,759	150(113,155)
			94	6,747	6,611	148(58,87)
			95	6,583	6,376	135(23,44)
			96	6,430	6,349	127(100,152)
			97	6,288	6,223	126(78,144)
csp250	250	4,152	90	-	**12,214**	
			91	-	11,813	401(128,198)
			92	-	11,426	387(131,205)
			93	-	11,106	320(28,64)
			94	-	10,793	313(154,207)
			95	-	10,515	278(45,100)
			96	-	10,242	273(30,60)
			97	-	9,998	244(53,77)
			98	-	9,761	237(124,170)
			99	-	9,529	232(138,181)
			100	-	9,311	218(24,54)
			101	-	9,120	191(202,235)
			102	-	8,933	187(50,74)
			103	-	8,746	187(141,187)
			104	-	8,583	163(3,33)
			105	-	8,425	158(223,248)
			106	-	8,282	143(6,40)
			107	-	8,143	139(139,188)
			108	8,406	**8,009**	134(31,55)
			109	8,212	**7,876**	133(61,123)
			110	8,023	**7,744**	132(39,61)
			111	7,863	**7,615**	129(9,26)
			112	7,707	**7,487**	128(115,147)

이러한 필요성에 따라, 본 논문은 NP-완전 문제인 CSP에 대해 $O(m\log m)$의 다항시간으로 최적 해를 찾아 갈 수 있는 규칙을 가진 휴리스틱 알고리즘을 제안하였다.

제안된 방법은 단순히 c_{ij}를 오름차순으로 정렬하고, c_{ij}가 존재하는 두 업무 i,j 그룹들에 대한 업무 수행시간 o_i,o_j의 합 Σo가 T를 초과하지 않으면 하나의 그룹으로 병합하는 방법을 적용하였다. 이 결과, n개 업무를 수행하는 최소의 승무원수 $_{\min}K$와 최소 소요 비용 $_{\min}\Sigma c_{ij}$를 얻었으며, 특정 승무원 수를 운영하고자 할 경우 원하는 K에 도달할 때까지 $_{\max}c_{ij}$를 갖는 호를 삭제하여 그룹을 분할하는 방법을 적용하였다.

5개의 실험 데이터에 적용한 결과, 제안된 알고리즘은 수학

적 방법인 선형계획법이나 NP-완전으로 알려진 SCP나 SPP에 비해 매우 간단하면서도 최적 해를 구할 수 있는 장점을 갖고 있음을 보였다.

REFERENCES

[1] J. E. Beasley and B. Cao, "A Tree Search Algorithm for the Crew Scheduling Problem," European Journal of Operational Research, Vol. 94, No. 3, pp. 517-526, Nov. 1996.

[2] J. C. Falkner and D. M. Ryan, "Aspects of Bus Crew Scheduling System Using a Set Partitioning Model," Computer-Aided Transit Scheduling, Lecture Note in Economics Mathematical Systems, Vol. 308, pp. 91-103, 1988.

[3] G. W Graves, R. D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara, "Flight Crew Scheduling," Management Science, Vol. 39, No. 6, pp. 736-745, Jun. 1993.

[4] I. Gershkoff, "Optimizing Flight Crew Schedules," Interfaces, Vol. 19, No. 4, pp. 29-43, Jul. 1989.

[5] K. L. Hoffman and M. Padberg, "Solving Airline Crew Scheduling Problems by Branch-and-Cut," Management Science, Vol. 39, No. 6, pp. 657-682, Jun. 1993.

[6] T. G. Crainic and J. M. Rousseau, "The Column Generation Principle and the Airline Crew Scheduling Problem," INFOR, Vol. 25, pp. 136-151, 1987.

[7] M. Desrchers and F. Soumis, "A Column Generation Approach to the Urban Transit Crew Scheduling Problem," Transportation on Science, Vol. 23, nO. 1, pp. 1-13, Feb. 1989.

[8] S. Lavoie, M. Minoux, and E. Gdier, "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation," European Journal of Operational Research, Vol. 35, No. 1, pp. 45-58, Apr. 1988.

[9] B. M. Smith, "IMPACS - a Bus Crew Scheduling System Using Integer Programming," Mathematical Programming, Vol. 42, No. 1-3, pp. 181-187, Apr. 1988.

[10] H. Cheddad, "Algorithms for Crew Scheduling Problems," Ph.D thesis, Imperial College, London, 1987.

[11] M. Ball, L. Bodin and R. Dial, "A Matching Based Heuristic for Scheduling Mass Transit Crews and Vehicles," Transportation Science, Vol. 17, No. 1, pp. 4-31, Feb. 1983.

[12] M. Ball and A. Roberts, "A Graph Partitioning Approach to Airline Crew Scheduling," Transportation Science, Vol. 19, No. 2, pp. 107-126, May. 1985.

[13] S. Martello and P. Toth, "A Heuristic Approach to the Bus Driver Scheduling Problem," European Journal of Operational Research, Vol. 24, No. 1, pp. 106-117, Jan. 1986.

[14] W. Qiao, "An Algorithm for Crew Scheduling Problem with Bin Packing Features," Master of Science, Department of Civil and Environmental Engineering, Faculty of the Graduate School of the University of Maryland, pp. 1-88, 2008.

[15] M. Chen and H. Niu, "A Model for Bus Crew Scheduling Problem with Multiple Duty Types," Discrete Dynamics in Nature and Society, Vol. 2012, Article ID: 649213, pp. 1-11, Aug. 2012.

[16] J. E. Beasley, "OR-Library: Crew Scheduling," http://people.brunel.ac.uk/~mastjjb/jeb/orlib/csinfo.html, 2013.

[17] R. G. Jeroslow, "Trivial Integer Programs Unsolvable by Branch-and-Bound," Mathematical Programming, Vol. 6, No. 1, pp. 105-109, Dec. 1974.

Authors

Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively. He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.