

## 운전기사 일정계획 문제의 교환-삽입 알고리즘

이 상 운\*

### Swap-Insert Algorithm for Driver Scheduling Problem

Sang-Un Lee \*

#### 요 약

본 논문은 NP-완전인 DSP에 대해  $O(m)$ 의 다항시간으로 근사 해를 찾는 규칙을 제시한 휴리스틱 알고리즘을 제안하였다. 제안된 알고리즘은  $m$ 개의 주어진 운행계획에 대해, 최소의 운전기사인  $n$ 명을 배정한 초기 배정 결과를 얻는다. 다음으로, 교환 또는 삽입의 5개 규칙들을 적용하여 초과시간 (OT)과 유휴시간 (IT)를 감소시켜 최소의 비용 (TC)을 얻었다. 제안된 알고리즘은 최적 (또는 근사) 해를 찾는 규칙을 제안한  $O(m)$  복잡도의 휴리스틱 다항 시간 알고리즘임에도 불구하고, 5개의 실험 데이터에 적용한 결과 메타 휴리스틱 기법들과 필적하는 결과를 얻었다. 결론적으로, 본 논문에서는 CSP에 있어서 최적 해를 찾아가는 규칙이 전혀 없는 NP-완전이 아닌 다항시간의 규칙이 존재하는 P-문제가 될 수 있음을 보였다.

▶ Keywords : 운전기사 일정 문제, 평행사변형, 교환, 삽입, 유휴시간

#### Abstract

This paper suggests  $O(m)$  polynomial time heuristic algorithm to obtain the solution for the driver scheduling problem, DSP, that has been classified as NP-complete problem. The proposed algorithm gets the initial assignment of  $n$  minimum number of drivers from given  $m$  schedules. Next, this algorithm gets the minimum total time (TC) using 5 rules of swap and insert for decrease of over times (OT) and idle times (IT). Although this algorithm is a heuristic polynomial time algorithm with  $O(m)$  time complexity rules to be find a optimal (or approximate) solution, this algorithm is equal to metaheuristic methods for the 5 experimental data. To conclude, this paper shows the DSP is not NP-complete problem but Polynomial time (P)-problem with polynomial time rules.

▶ Keywords : Driver scheduling problem, Parallelogram, Swap, Insert, Idle time

•제1저자 : 이상운

•투고일 : 2014. 08. 19. 심사일 : 2014. 09. 14. 게재확정일 : 2014. 09. 17.

\* 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Eng., Gangneung-Wonju National University)

## I. 서론

운전기사 일정계획 문제 (driver scheduling problem, DSP)는 버스 기사 일정계획 문제 (bus driver scheduling problem, BDSP) 또는 승무원 일정계획 문제 (crew scheduling problem, CSP)라고도 한다[1-3].

DSP는  $S_i, (i=1,2,\dots,m)$ 의 다른 경로를 운행할 버스 (트럭, 또는 택시) 운행계획이 있을 때, 최소의 비용  $\min TC$ 으로 최소의 운전기사  $D_j, (j=1,2,\dots,n)$ 를 배정하는 문제이다 [4,5]. 이 경우, 운전기사의 근무시간은 정상 8시간, 최대 10시간을 초과할 수 없으며, 두 대의 버스를 중첩 운행할 수 없다.

DSP에 대해 Lourenço et al.[2]은 Tabu search (TS) + GA (genetic algorithm)에 기반한 메타 휴리스틱 기법을, Lourenço et al.[6]은 GRASP (greedy randomize adaptive search procedure)에 기반한 메타 휴리스틱 기법을, Mauri와 Lorena[4,5]는 PTA/LP (population training algorithm/linear programming) 기법을, Shen과 Kwan[7]은 TS를, Dias et al.[8]은 GA를, Bai et al.[9]은 Nash Equilibrium 기법을, Portugal et al.[10]과 Leone et al.[11]은 집합 분할/피복에 기반한 수학적 기법을, Laurent et al.[12]은 담금질 기법 (simulated annealing, SA)을, Shung et al.[13]은 진화 알고리즘 (evolutionary algorithm, EA)을 연구하였다.

이와 같은 다양한 메타휴리스틱 기법 연구들에도 불구하고, 집합피복 문제 (set cover problem, SCP)는 Karp[14]가 제시한 21개 NP-완전 (nondeterministic polynomial-complete) 문제들 중 하나로 아직까지 정확한 해를 다항시간으로 구할 수 있는 알고리즘이 알려져 있지 않고 있다.

메타휴리스틱 기법은 랜덤한 초기치 설정으로 지역 최적점 (local minima)을 찾아가는 기법으로, 일단 지역 최적점에 수렴하면 이 지점을 탈출할 수 없는 단점이 있으며, 운 좋게도 전역 최적점 (global minima) 근방에 초기치가 설정되면 전역 최적점에 도달하기도 한다. 따라서 메타휴리스틱 기법은 전역 최적점인 최적 해를 찾아가는 규칙을 제시하지 못하며, 동일한 랜덤 초기치를 얻기 어려운 검증이 불가한 단점이 있다. 반면에, 휴리스틱 기법은 지역이나 전역 최적점을 찾아가는 규칙을 제시한다는 장점이 있다.

본 논문에서는 DSP에 대해 메타휴리스틱 기법에 필적할 만한 휴리스틱 기법을 제안한다. 2장에서는 DSP의 정의를 고찰해 본다. 3장에서는 DSP의 근사 해를 다항시간으로 찾는 휴리스틱 알고리즘을 제안한다. 4장에서는 DSP의 실험

데이터들에 대해 제안된 휴리스틱 알고리즘의 적합성을 메타휴리스틱 기법들과 비교하여 검증해 본다.

## II. 운전기사 일정계획 문제

운전기사 일정계획 문제 (DSP)는 그림 1로 설명되며, 다음과 같이 정의된다[5].

- (1) 운전기사의 정상 근무시간 (normal working time, NWT)은 480분(8시간)이며, 최대 근무시간 (maximum working time, MWT)은 600분(10시간)으로, 다수의 버스 운행 계획을 배정시 600분을 초과할 수 없다.
- (2)  $i$ 번째 버스 운행종료 시간  $t_{e(i)}$ 과  $i+1$ 번째 버스 운행 시작시간  $t_{s(i+1)}$ 의 중첩시간 (overlapping time, OV)은 허용되지 않는다.
- (3) 운전기사  $j$ 의 유희시간 (idle time, IT)는 자신에게 배정된 운행계획상의 모든 버스들에 대해 NWT까지  $i$ 번째 버스 운행종료 시간  $t_{e(i)}$ 과  $i+1$ 번째 버스 운행 시작시간  $t_{s(i+1)}$ 의 합으로 계산된다.
- (4) 초과근무시간 (overtimes, OT)은 MWT를 초과한 시간으로 계산된다.
- (5) MWT를 초과하는 시간 (exceeds time, ET)는 허용되지 않는다.
- (6) 비용 (total cost, TC)은  $IT+OT$ 로 계산된다.

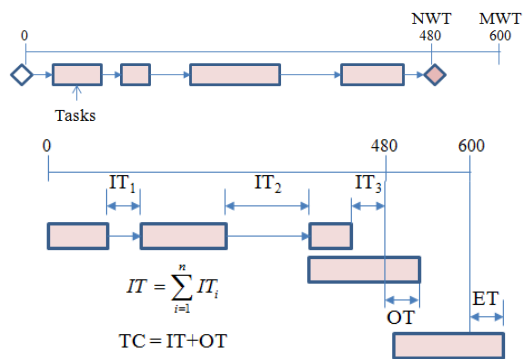


그림 1. 운전기사 일정계획 문제  
Fig. 1. Driver scheduling problem

위와 같은 6가지 정의에 기반하여 DSP는 식 (1)을 만족하도록  $n$ 명의 운전기사들에게  $m$ 개의 버스 운행계획을 배정하는 문제이다.

$$\min \sum_{j=1}^n c_j x_j \quad (1)$$

$$\text{s.t. } \sum_{j=1}^n a_{ij} x_j = 1, i = 1, 2, \dots, m$$

$$x_j \in \{0, 1\}, j = 1, 2, \dots, n$$

$$c_j = (IT_j + OT_j)$$

$$IT = \max(0, [NWT - t_{e(last)}]) + \sum_{i=1}^m \max(0, [t_{s(i+1)} - t_{e(i)}])$$

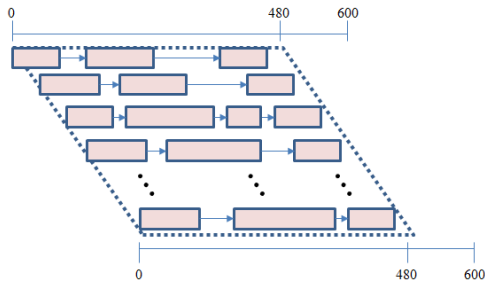
$$OT = \max(0, [t_{e(last)} - t_{s(1st)}] - NWT)$$

식 (1)은 집합피복 문제 (set-covering problem, SCP)와 같다[6]. SCP는 Karp[14]가 제시한 21개 NP-완전 문제들 중 하나로 알려져 있다. 즉, 정확한 해를 구하는 다항시간 알고리즘이 알려져 있지 않고 있다.

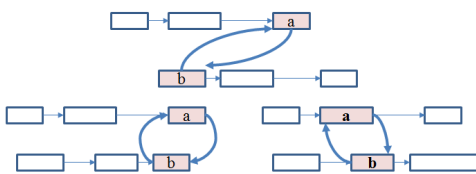
3장에서는 DSP에 대해 다항시간으로 근사 해를 찾을 수 있는 휴리스틱 알고리즘을 제안한다.

### III. 교환-삽입 알고리즘

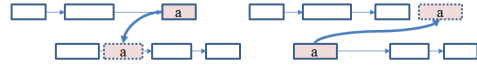
본 장에서는 주어진  $S_j, (i = 1, 2, \dots, m)$  운행계획에 대해, 그림 2의 (a)와 같이 각 운전기사  $j$ 에 NWT 또는 MWT를 초과하지 않도록 운행계획 버스들을 평행사변형법으로 초기 배정 결과를 얻는다. 초기 배정 결과에 대해 (b)와 같이 버스 운행계획을 상호 교환 (swap)하거나 (c)와 같이 삽입 (insert)하여 OT 또는 IT를 감소시키는 간단한 규칙을 제안한다. 제안된 알고리즘을 교환-삽입 알고리즘 (swap-insert algorithm, SIA)이라 하자.



(a) Initial Assignment



(b) Swap method



(c) Insert method  
 그림 2. 교환-삽입 알고리즘  
 Fig. 2. Swap-Insert algorithm

제안된 SIA는 다음과 같이 수행된다.

- Step 1. 초기 배정 /\* 수행 복잡도 :  $O(m)$  \*/  
 주어진  $S_j, (i = 1, 2, \dots, m)$  운행계획에 대해, 각 운전기사  $j$ 에 NWT 또는 MWT를 초과하지 않도록 운행계획 버스들을 평행사변형법으로 배정한다.
- Step 2. 교환과 삽입 최적화 /\* 수행 복잡도 :  $O(m)$  \*/  
 다음의 5가지 규칙을 적용한다.
- (규칙 1) NWT를 초과하는  $i$ 번째 운행계획을  $t_{e(i)last} > t_{e(k)1st}$  인  $k$ 번째 운행계획과 상호 교환하여 OT를 감소시킨다.  
 if  $\exists_i, t_{e(i)last} > NWT$  and  
 $\exists_k, t_{e(i)last} > t_{e(k)1st}$  then  $i \leftrightarrow k$
- (규칙 2)  $t_{s(i)1st} < t_{s(k)1st}$  에 대해  $t_{e(i)last} > t_{e(k)last}$  이면 상호 교환하여 IT를 감소시킨다.  
 if  $\exists_i, k, t_{s(i)1st} < t_{s(k)1st}$  and  $t_{e(i)last} > t_{e(k)last}$   
 then  $t_{e(i)last} \leftrightarrow t_{e(k)last}$
- (규칙 3) 중간에 위치한 2개의 운행계획을 상호 교환하여 IT를 감소시킨다.  
 if  $\exists_i, k, i, k$  positioned at middle sequence  
 then  $i \leftrightarrow k$
- (규칙 4)  $j$ 번째 운전기사의 일정계획의 마지막에 위치한 운행계획을  $k$ 번째 운전기사의 IT에 추가할 수 있으면 이동시켜 IT를 감소시킨다.  
 if  $\exists_j, [t_{s(j)last}, t_{e(j)last}] \in k$   
 then  $k \leftarrow [t_{s(j)last}, t_{e(j)last}]$
- (규칙 5)  $j$ 번째 운전기사의 일정계획의 첫 번째에 위치한 운행계획을  $k$ 번째 운전기사의 마지막 일정에 추가할 수 있으면 이동시켜 IT나 OT를 감소시킨다.  
 if  $\exists_j, [t_{s(j)first}, t_{e(j)first}]$  and  $t_{s(j)first} > t_{e(k)last}$   
 then  $k \leftarrow [t_{s(j)first}, t_{e(j)first}]$

메타휴리스틱 기법은 규칙 없이 랜덤한 초기치 설정으로 인해 대부분은 지역 최적 점에 수렴하기도 하며, 요행이도 전역 최적 점 근방에 위치하면 전역 최적점인 최적 해에 도달한다. 반면에, 제안된 SIA는 고정된 초기치에 대해 교환-삽입



dsp500에 대해 제안된 SIA를 적용하여 보고, Mauri와 Lorena[4,5]의 PTA/LP와 SA 결과와 비교하여 본다. dsp25, dsp50, dsp100, dsp250과 dsp500에 대해 SIA를 적용한 결과 운전기사별로 운행계획을 배정한 결과는 표 3에 제시하였다.

dsp25는  $n = 12$ , dsp50은  $n = 20$ , dsp100은  $n = 40$ , dsp250은  $n = 81$ , dsp500은  $n = 145$ 의 운전기사가 필요함을 알 수 있다.

표 3의 결과에 대해 Mauri와 Lorena[4,5]의 PTA/LP와 SA와 성능을 비교한 결과는 표 4에 제시되어 있다.

5개 데이터 모두에서 SIA는 최소의 운전기사를 얻었음을 알 수 있다. TC 에 대해서는 dsp25는 최적의 결과를, dsp50과 dsp100은 PTA/LP, SA와 동일한 결과를, dsp250과 dsp500에 대해서는 mPTA/LP 다음으로 좋은 성능을 나타냄을 알 수 있다.

표 3. SIA 결과  
Table 3. Result of SIA

dsp25									
$D_j$	Schedule	WT	OT	IT	$D_j$	Schedule	WT	OT	IT
D1	1-2	315	0	220	D7	12-17	420	0	142
D2	3-7	325	0	274	D8	13-20-24	425	0	205
D3	4-6-10	479	0	56	D9	14-21	381	0	175
D4	5-8	372	0	184	D10	15-23	400	0	181
D5	9-16	467	0	170	D11	22-25	288	0	255
D6	11-18	495	15	166	D12	19	152	0	328
계								15	2028

dsp50									
$D_j$	Schedule	WT	OT	IT	$D_j$	Schedule	WT	OT	IT
D1	1-8-12	450	0	96	D11	25-42	381	0	160
D2	2-13	471	0	191	D12	26-41	395	0	160
D3	3-14	449	0	195	D13	28-38-48	372	0	115
D4	4-9-17	480	0	28	D14	29-39	355	0	145
D5	5-11-15-19	478	0	45	D15	30-40	310	0	170
D6	6-10-18	477	0	27	D16	31-44	325	0	165
D7	7-16	415	0	123	D17	33-45	350	0	140
D8	20-22-32-43	470	0	50	D18	34-47-49	413	0	119
D9	23-27-36	423	0	69	D19	37-50	403	0	112
D10	21-24-35	440	0	65	D20	46	55	0	425
계								0	2600

dsp100									
$D_j$	Schedule	WT	OT	IT	$D_j$	Schedule	WT	OT	IT
D1	1-16	476	0	10	D21	25-49	438	0	300
D2	2-35-44	444	0	94	D22	28-50	420	0	310
D3	3-21-37	440	0	58	D23	29-52	475	0	189
D4	4-22-30	318	0	165	D24	54-67	415	0	142
D5	5-23-39	455	0	48	D25	56-69	420	0	140
D6	6-26-40-45	400	0	105	D26	57-72-91	465	0	134
D7	7-27-42	458	0	42	D27	58-64-83	336	0	270
D8	8	130	0	350	D28	59-74-94	467	0	124
D9	9-31-36	324	0	205	D29	60-75	373	0	185
D10	11	102	0	378	D30	61-76	370	0	195
D11	10-32-41-43	446	0	118	D31	62-81-85-90	455	0	165
D12	12	125	0	355	D32	63-77-93	465	0	109
D13	13-33-38-48	469	0	85	D33	65-84-88-97	458	0	119
D14	14	140	0	340	D34	66	154	0	326
D15	15-34	290	0	196	D35	68-86-99	480	0	65
D16	17-55	474	0	246	D36	70-78-87-98	325	0	200
D17	18-47	480	0	175	D37	71	140	0	340
D18	19-53	466	0	247	D38	73-82-89-96	363	0	210
D19	20-46	462	0	172	D39	80-92	295	0	200
D20	24-51	447	0	248	D40	79-95-100	455	0	35
계								0	7395

dsp250													
$D_j$	Schedule	WT	OT	IT	$D_j$	Schedule	WT	OT	IT				
D1	1-3-34-50	464	0	51	D42	106-134-174	474	0	32				
D2	2-29-68	445	0	50	D43	107-142-172	410	0	80				
D3	4-40-79	440	0	48	D44	111-139-173	467	0	15				
D4	5-73-101	475	0	9	D45	112-147-154-206	408	0	91				
D5	6-42-53-86	465	0	33	D46	113-137-175	423	0	60				
D6	7-49-81	459	0	40	D47	116-133-168	425	0	77				
D7	8-51-60-93	479	0	64	D48	119-152-199	468	0	15				
D8	9-47-85-104	449	0	67	D49	115-148-185-194	350	0	145				
D9	10-52-83	446	0	59	D50	123-157-166	370	0	119				
D10	11-56	420	0	79	D51	124-158-205	474	0	8				
D11	12-48-55	243	0	242	D52	127-163-204	450	0	30				
D12	13-30-69-98-108	425	0	85	D53	131-164-209	395	0	85				
D13	14-57-61-83	470	0	25	D54	128-167-211	435	0	52				
D14	15-59-100	443	0	65	D55	132-170-216	417	0	79				
D15	16-54	280	0	210	D56	136-171	344	0	138				
D16	17-63-89	474	0	32	D57	138-176-221-230	471	0	72				
D17	18-62-87	445	0	57	D58	141-165-210	408	0	80				
D18	19-58-90	478	0	34	D59	140-156-179-212	364	0	180				
D19	20-64-91	477	0	37	D60	144-186-241	525	45	42				
D20	21-80-96	389	0	206	D61	145-187-227	521	41	30				
D21	22-78	366	0	216	D62	143-178-192-229	548	68	83				
D22	23-71	356	0	185	D63	149-159-207-213-330-242	530	30	38				
D23	24-65-92	480	0	24	D64	150-195-228	515	55	50				
D24	25-67-92-95	479	0	110	D65	151-193-235	538	56	116				
D25	26-66-97	478	0	100	D66	153-203-217	535	46	42				
D26	27-76-94	478	0	94	D67	155-180-232	520	40	129				
D27	28-74-105	514	34	69	D68	160-189-201-233	511	31	66				
D28	31-102-129	540	60	135	D69	161-218-237	515	35	90				
D29	32-109	500	20	300	D70	162-208-219-238	512	32	32				
D30	33-77-117	523	43	69	D71	169-214-240	493	13	57				
D31	35-110-146	530	50	280	D72	177-197-215-244	475	0	85				
D32	36-72-125	518	38	134	D73	183-223-235	445	0	148				
D33	37-70-114	528	48	75	D74	181-226	465	0	45				
D34	38-75-120	537	57	100	D75	182-191-239	465	0	280				
D35	39-118	529	49	230	D76	184-222-246	469	0	55				
D36	41-84	522	42	377	D77	188-224-245	463	0	52				
D37	43-99-135	521	41	186	D78	190-225-248	475	0	76				
D38	44-122	525	45	355	D79	198-231-247	466	0	69				
D39	45-126	535	55	243	D80	200-234-249	480	0	60				
D40	46-121	536	56	246	D81	202-243-250	545	65	105				
D41	103-130-196	441	0	56									
계										1195	7843		

dsp500													
$D_j$	Schedule	W	OT	IT	$D_j$	Schedule	WT	OT	IT				
D1	1-2-19-99	460	0	29	D74	233-297-354-379	468	0	18				
D2	3-59-89-102-164-182	460	0	29	D75	235-298-388	477	0	15				
D3	4-58-77-101-124-177	482	2	12	D76	235-299-315-380	465	0	29				
D4	5-63-103-168	485	5	10	D77	237-303-395	472	0	20				
D5	6-70-112-175	435	0	62	D78	239-307-370-389	485	5	8				
D6	7-60-143	440	0	42	D79	241-282-369	435	0	50				
D7	8-71-191	460	0	25	D80	242-314-394	457	0	25				
D8	9-142-156	469	0	19	D81	244-311-343	399	0	93				
D9	10-78-128-200	482	2	12	D82	245-316-408-436-452	465	0	33				
D10	11-79-118-179-207	462	0	61	D83	246-319-381	440	0	36				
D11	12-80-97-116-174-202	469	0	51	D84	247-266-328-414	430	0	50				
D12	13-81-158	450	0	90	D85	248-315-329-409	475	0	10				
D13	14-86-158-188	460	0	45	D86	249-317-405-457	470	0	25				
D14	15-67-148-165	464	0	41	D87	250-318-367-401	455	0	70				
D15	16-91-161	463	0	57	D88	251-323-410-480	473	0	66				
D16	17-90-162	450	0	53	D89	252-330-420	473	0	15				
D17	18-92-171-180-217	475	0	81	D90	253-325-373-398	470	0	15				
D18	20-94-167	458	0	54	D91	254-320-347-448	440	0	73				
D19	21-93-155	423	0	90	D92	255-332-434	476	0	22				
D20	22-95-170	462	0	62	D93	257-331-413	463	0	45				
D21	23-96-169-222	484	0	90	D94	258-329-375-415	473	0	20				
D22	24-53-138-176	470	0	69	D95	260-333-440	480	0	2				
D23	25-104-140-183	470	0	98	D96	261-341-432-448	426	0	74				
D24	26-105-141-212-231	475	0	96	D97	262-335-374	402	0	85				
D25	27-172	454	0	47	D98	263-331-436	465	0	50				
D26	28-106-166	428	0	57	D99	264-349-363-390	417	0	70				
D27	29-107-173	428	0	47	D100	265-321-404-469	451	0	50				
D28	30-114-178	476	0	47	D101	266-304-393-429	480	0	25				
D29	31-119-184	476	0	35	D102	267-342	323	0	158				
D30	32-113-181	480	0	36	D103	269-350-453	494	14	31				
D31	33-111-144-192	479	0	65	D104	272-356-460	505	25	70				
D32	34-180-188	484	4	47	D105	273-355-436	468	0	19				
D33	35-117-149-230	482	2	89	D106	274-346-402-471	525	45	10				
D34	36-110-185	486	6	60	D107	275-348-387-419-450	505	25	36				
D35	37-120-187	488	9	53	D108	277-336-455	503	23	84				
D36	38-125-137-194	481	1	28	D109	278-352-445	480	0	5				
D37	39-122-198	486	6	56	D110	280-364-458	505	25	34				
D38	40-108-201	486	6	75	D111	283-371-418	543	63	3				
D39	41-123-145-193	488	5	104	D112	283-377-491	521	41	23				
D40	42-61-146-203	490	10	17	D113	284-388-371-467	530	50	45				
D41	43-128-189	487	7	52	D114	285-327-422-449-482-488	545	65	28				
D42	44-121-190	482	2	54	D115	286-378-463	515	35	55				
D43	45-129-204	484	5	108	D116	287-380-462	521	41	28				
D44	46-130-225	484	4	89	D117	291-							

D59	67-214	498	9	186	D132	326-407-477-492	525	45	40
D60	68-147-232	491	17	64	D133	334-411-465	530	50	10
D61	69-150-238-258-271	495	15	98	D134	338-431-486	487	7	12
D62	72-228	488	9	155	D135	340-433-442	474	0	15
D63	73-154-221	490	10	28	D136	344-386-461	504	24	163
D64	74-224	487	7	175	D137	346-424-493	495	15	55
D65	75-205-268	491	11	163	D138	351-446-487	493	13	13
D66	76-109-220-279	515	35	236	D139	353-447-495	490	10	50
D67	82-163	514	34	29	D140	357-428-498	520	40	0
D68	83-157-240	520	40	75	D141	362-439-490	480	0	60
D69	84-209-276	538	56	135	D142	365-456-496	490	10	15
D70	85-152-227-289	523	48	15	D143	366-451-494	481	1	26
D71	100-216-270	558	75	139	D144	369-443-466-499	500	20	85
D72	226-288-368	460	0	20	D145	372-426-500	510	30	0
D73	229-290-376	476	0	5					
계								1957	8451

표 4. 알고리즘 성능 비교  
Table 4. Compare with algorithm performance

Problem	Model	Number of Drivers (ND)	Over Time (OT)	Idle Time (IT)	Total Cost(TC) =OT+IT	Rank
dsp25	mPTA/LP	12	15	2356	2371	2
	sPTA/LP	12	15	2356	2371	2
	sSA	12	15	2356	2371	2
	mSA	12	15	2356	2371	2
	SIA	12	15	2028	2043	1
dsp50	mPTA/LP	20	0	2600	2600	1
	sPTA/LP	20	0	2600	2600	1
	sSA	20	27	2627	2654	2
	mSA	20	0	2600	2600	1
	SIA	20	0	2600	2600	1
dsp100	mPTA/LP	40	0	7395	7395	1
	sPTA/LP	40	0	7395	7395	1
	sSA	40	0	7395	7395	1
	mSA	40	0	7395	7395	1
	SIA	40	0	7395	7395	1
dsp250	mPTA/LP	81	1103	7751	8854	1
	sPTA/LP	82	1058	8186	9244	3
	sSA	85	815	9383	10198	5
	mSA	83	1112	8720	9832	4
	SIA	81	1195	7843	9038	2
dsp500	mPTA/LP	145	1601	8115	9716	1
	sPTA/LP	151	1567	10961	12528	3
	sSA	153	1254	11608	12862	5
	mSA	153	1096	11450	12546	4
	SIA	145	1957	8451	10408	2

### V. 결론 및 추후 연구과제

본 논문은 NP-완전인 DSP에 대해  $O(m)$ 의 다항시간으로 근사 해를 찾는 규칙을 제시한 휴리스틱 알고리즘을 제안하였다.

가장 일찍 시작하는 운행계획부터 한 명의 운전기사에게  $NWT=480$ 분 또는  $MWT=600$ 분을 초과하지 않는 운행계획들을 연속적으로 배정하고, 다음 운전기사에게 남아 있는 가장 일찍 운행할 계획부터 다시 이 과정을 수행하여 평행사변형 형태로 최소의 운전기사인  $n$ 명을 배정한 초기 배정 결과를 얻는다. 다음으로, 교환 또는 삽입의 5개 규칙들을 적용하여 OT와 IT를 감소시켜 최소의 TC를 얻었다.

제안된 알고리즘은 최적 (또는 근사) 해를 찾는 규칙을 제안한  $O(m)$  복잡도의 휴리스틱 다항시간 알고리즘임에도 불구하고, 5개의 실험 데이터에 적용한 결과 메타 휴리스틱 기법들과 필적하는 결과를 얻었다. 즉, csp25에 대해서는 최적의 결과를, csp50과 csp100에 대해서는 mPTA/LP와 동일

한 결과를, csp250과 csp500에 대해서는 mPTA/LP 다음으로 좋은 결과를 얻었다.

결론적으로, 본 논문에서는 CSP에 있어서 최적 해를 찾아가는 규칙이 전혀 없는 NP-난제가 아닌 다항시간의 규칙이 존재하는 P-문제가 될 수 있음을 보였다.

본 논문에서 제안한 SIA는 최적 해를 찾아가는 규칙인 교환-삽입법을 제안하였지만 csp250과 csp500과 같이 데이터 크기가 클 경우 mPTA/LP 기법으로 얻은 최적 해를 찾지 못한 단점을 갖고 있다. 따라서, 추후 대용량 데이터에 대해 mPTA/LP 기법으로 얻은 최적 해를 찾을 수 있는 방법을 연구할 예정이다.

### 참고문헌

- [1] A. Wren, S. Fores, A. Kwan, R. Kwan, M. Parker, and L. Proll, "A Flexible System for Scheduling Drivers," Journal of Scheduling, Vol. 6, No. 5, pp. 437-455, Sep. 2003.
- [2] H. R. Lourenço, J. P. Paixão, and R. Portugal, "Multiobjective Metaheuristics for the Bus-Driver Scheduling Problem," Transportation Science, Vol. 35, No. 3, pp. 331-343, Jun. 2001.
- [3] S. Groot and D. Huisman, "Vehicle and Crew Scheduling Solving Large Real-World Instances with an Integrated Approach," 9th International Conference on Computer-Aided Scheduling of Public Transport, San Diego, California, Lecture Notes in Economics and Mathematical Systems, Vol. 600, pp 43-56, 2008.
- [4] G. R. Mauri and L. A. N. Lorena, "Driver Scheduling Generation Using a Population Training Algorithm," Brazilian Symposium in Neural Networks, Sao Luis, Maranhão, Brazil, pp. 1-6, Sep. 2004.
- [5] G. R. Mauri and L. A. N. Lorena, "A New Hybrid Heuristic for Driver Scheduling," International Journal of Hybrid Intelligent Systems, Vol. 4, No. 1, pp. 39-47, Jan. 2007.
- [6] H. R. Lourenço, J. P. Paixão, and R. Portugal, "Metaheuristics for the Bus-Driver Scheduling Problem," Department of Economics and Management, Universitat Pompeu Fabra, Vol. 35, pp. 1-26, Jul. 1998.
- [7] Y. Shen and R. S. K. Kwan, "Tabu Search for Driver Scheduling," Computer-Aided Scheduling of Public

- Transport, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, Vol. 505, pp. 121-135, 2001.
- [8] T. G. Dias, J. P. Sousa, and J. F. Cunha, "A Genetic Algorithm for the Bus Driver Scheduling Problem," 4th International Conference of Metaheuristics, pp. 35-40, Jul. 2001.
- [9] R. Bai, J. Li, J. Atkin, and G. Kendall, "A Novel Approach to Independent Taxi Scheduling Problem Based on Stable Matching," Journal of Operational Research Society, pp. 1-10, Sep. 2013.
- [10] R. Portugal, H. R. Lourenço, and J. P. Paixão, "Driver Scheduling Problem Modeling," Public Transportation, Vol. 1, No. 2, pp. 103-120, Jun. 2009.
- [11] B. Laurent, V. Guilhaire, and J. K. Hao, "A Heuristic Solution for a Driver-Vehicle Scheduling Problem," Proceedings of Operations Research, Vol. 2005, pp. 703-708, 2005.
- [12] R. Leone, P. Festa, and E. Marchitto, "A Bus Driver Scheduling Problem: A New Mathematical Model and GRASP Approximate Solution," Journal of Heuristics, Vol. 17, No. 4, pp. 441-466, Aug. 2011.
- [13] L. T. Shung, R. Ramli, and H. Ibrahim, "An Evolutionary Algorithm Approach to a Bus Driver Scheduling Problem with Break-Time Consideration," Proceedings of the 5th Asian Mathematical Conference, Malaysia, pp. 652-658, Jun. 2009.
- [14] R. M. Karp, "Reducibility Among Combinatorial Problems," In Complexity of Computer Computations, Plenum Press, New York, pp. 85-103, 1972.
- [15] L. A. N. Lorena, "Problem Instances: Driver Scheduling," <http://www.lac.inpe.br/~lorena/nstancias.html>, 2014.

## 저자 소개



### 이 상 운

1983년 ~ 1987년 :  
한국항공대학교 항공전자공학과 (학사)  
1995년 ~ 1997년 :  
경상대학교 컴퓨터과학과 (석사)  
1998년 ~ 2001년 :  
경상대학교 컴퓨터과학과 (박사)  
2003.3 ~ 현재 : 강릉원주대학교  
멀티미디어공학과  
부교수  
관심분야 : 소프트웨어 프로젝트 관리,  
소프트웨어 개발 방법론,  
소프트웨어 신뢰성, 그래프  
알고리즘  
e-mail : [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)