

Maximum Options-Equiped Class First-Production Algorithm for Car Sequencing Problem

Sang-Un Lee *

Abstract

This paper suggests $O(n)$ linear-time algorithm for car sequencing problem (CSP) that has been classified as NP-complete because of the polynomial-time algorithm to solve the solution has been unknown yet. This algorithm applies maximum options-equipped car type first production rule to decide the car sequencing of n meet the $r:s$ constraint. This paper verifies thirteen experimental data with the six data are infeasible. For thirteen experimental data, the proposed algorithm can be get the solution for in all cases. And to conclude, This algorithm shows that the CSP is not NP-complete but the P-problem. Also, this algorithm proposes the solving method to the known infeasible cases. Therefore, the proposed algorithm will stand car industrial area in good stead when it comes to finding a car sequencing plan.

▶ Keyword : Car sequencing, $r:s$, Car type, Constraints, Assignment

1. Introduction

동일한 자동차 모델을 생산하는 조립라인을 고려하여 보자. 이 생산라인에서 n 대의 자동차를 생산하는 경우, n 대의 자동차 모두 기본형 부품들이 장착된 이후, 추가로 k 개의 옵션(option)들 $O_i, i=1,2,\dots,k$ 중 고객의 요청에 의해 일부 옵션들을 다르게 장착해야 한다. 이 경우 n 대 자동차들 중에는 $k=0$ 인 자동차가 없으며, $2^k - 1$ 개의 다른 옵션 부류를 장착하는 일부 자동차 부류(car type or class) $C_j, j=1,2,\dots, 2^k - 1$ 가 존재한다. 이 문제에서 한 가지 고려할 제약사항은 조립라인의 생산성 또는 제품 생산시간(cycle time)을 고려하여 연속하는 s 대의 자동차 생산대수 중에서 r 대만이 해당 옵션을 장착할 수 있다. 이 경우 자동차 조립라인에 투입될 n 대의 순서를 결정하는 문제를 자동차 조립순서 문제(car sequencing problem, CSP)라 한다[1].

CSP는 해를 다항시간으로 구하는 알고리즘이 알려져 있지 않아 NP-완전(nondeterministic polynomial-time complete)으로 분류된 난제이다[1,2].

이에 따라 정확한 해를 찾아갈 수 있는 다항시간 규칙이 존재하지 않는 난제로 판단하여, 선형계획법(integer programming, LP)의 수

학적 프로그램 패키지 활용, 이웃탐색법(neighborhood search, NS)을 비롯하여 유전자 알고리즘(genetic algorithm, GA), 개미집단 최적화(ant colony optimization, ACO), Tabu 탐색법(tabu search, TS) 등의 메타휴리스틱 방법들을 적용하여 해를 얻고자 시도하고 있다 [3]. 예를 들면, Fliedner와 Boysen[4]은 분기한정법(branch-and-bound, BB)을, Gagné et al.[5]은 ACO를, Zufferey et al.[6]은 TS를, Cordeau et al.[7]은 반복적 TS(iterative TS)를, Thiruvady[8]는 라그랑제-ACO(Lagrangian-ACO, LACO)를, Simonis[9]는 BB를 적용한 트리 탐색법을 제안하였다.

이와 같이 다양한 방법들이 적용되고 있음에도 불구하고, Smith[10]의 CSPLib에서는 일부 실험 데이터에 대해서는 해를 찾지 못하였으며, Gent[11]도 일부 문제는 해가 존재하지 않음을 증명하였다.

본 논문에서는 CSP의 해를 $O(n)$ 의 선형시간으로 구할 수 있는 알고리즘을 제안한다. 또한, 기존에 해가 존재하지 않는다고 알려진 Smith[10]의 CSPLib와 Gent[11]이 제시한 문제들에 대해서도 해를 찾을 수 있음을 보인다. 2장에서는 예제 데이터를 대상으로 CSP의 문제를 정의한다. 3장에서는 CSP에 대해 $O(n)$ 복잡도로 해를 구할 수 있는 알고리즘을 제시한다. 4장에서는 제안된 알고리즘을 실제 데이터에 적용하여 알고리즘 적합성을 평가해 본다.

*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. 06. 17, Revised: 2015. 07. 05, Accepted: 2015. 07. 14.

II. Definitions of Car Sequencing

표 1은 $n=8$ 대의 자동차를 생산하는 경우, $k=5$ 개 옵션 부품들 중 가능한 조합인 $2^5 = 32$ 중에서 2개씩 다른 조합으로 장착하는 4개 부류가 예정되어 있다. 여기서 옵션 1 (O1)의 $r:s$ 를 예를 들면, O1은 연속된 5개 조립대수 중에서 3대의 자동차에만 장착할 수 있는 능력을 갖고 있음을 의미한다. 이 문제에서 8대의 자동차를 생산하는 순서를 결정하여 보자. 여기서, 셀의 “1”은 해당 자동차 부류가 해당 옵션을 장착하는 경우이며, “0”은 해당 옵션을 장착하지 않는 경우를 의미한다.

Table 1. Car sequencing problem of P_1

Option	Car Type (Class)				rs
	C1	C2	C3	C4	
O1	1	0	0	1	3:5
O2	0	1	0	0	1:3
O3	1	1	0	0	2:5
O4	0	0	1	1	2:3
O5	0	0	1	0	1:4
요구대수	2	2	2	2	8

Kis[1]는 P_1 에 대해 표 2와 같이 C2-C3-C1-C4-C4-C2-C3-C1, C1-C3-C2-C4-C4-C1-C3-C2와 C1-C3-C2-C4-C4-C2-C3-C1의 3가지 가능한 해를 제시하였다.

Table 2. Possible solution for P_1

Sol #1	C1	C2	C3	C4	rs	C2	C3	C1	C4	C4	C2	C3	C1
O1	1	0	0	1	3:5								
O2	0	1	0	0	1:3								
O3	1	1	0	0	2:5								
O4	0	0	1	1	2:3								
O5	0	0	1	0	1:4								
Req	2	2	2	2									
Sol #2	C1	C2	C3	C4	rs	C1	C3	C2	C4	C4	C1	C3	C2
O1	1	0	0	1	3:5	1			1	1	1		
O2	0	1	0	0	1:3			1					1
O3	1	1	0	0	2:5	1		1			1		1
O4	0	0	1	1	2:3		1		1	1		1	
O5	0	0	1	0	1:4		1						1
Req	2	2	2	2									
Sol #3	C1	C2	C3	C4	rs	C1	C3	C2	C4	C4	C2	C3	C1
O1	1	0	0	1	3:5	1			1	1			1
O2	0	1	0	0	1:3			1			1		
O3	1	1	0	0	2:5	1		1			1		1
O4	0	0	1	1	2:3		1		1	1		1	
O5	0	0	1	0	1:4		1						1
Req	2	2	2	2									

표 2의 3가지 해법 모두 각 옵션의 $r:s$ 제약조건을 충족시키는 자동차 조립순서를 알 수 있다. 따라서 CSP는 $k \times n$ 행렬에 대해 각 옵션의 $r:s$ 제약조건을 충족시키는 자동차 부류를 찾아 셀들을 채울 수 있으나 문제의 해가 결정된다. 만약, 채울 수 없으면 실행 불가능 (infeasible)으로, 채울 수 있으면 충족 가능 (satisfiable)으로 판단하며 이 경우 생산 순서도 함께 제시해야만 한다.

표 3은 Smith[20]의 CSPLib에서 예제로 제시한 P_2 문제로 $n=10, k=5$ 에 대해 6가지 자동차 부류를 갖고 있는 경우이다. Smith[10]는 6가지의 가능한 해를 제시하였다.

Table 3. Car sequencing problem and solution of P_2

Option	Car Type (Class)						rs
	C1	C2	C3	C4	C5	C6	
O1	1	0	0	0	1	1	1:2
O2	0	0	1	0	1	1	2:3
O3	1	0	0	0	1	0	1:3
O4	1	1	0	0	1	0	2:5
O5	0	0	1	0	0	0	1:5
Req	1	1	2	2	2	2	10

Sol #1: C1-C2-C6-C3-C5-C4-C4-C5-C3-C6
 Sol #2: C1-C3-C6-C2-C5-C4-C3-C5-C4-C6

Sol #3: C1-C3-C6-C2-C6-C4-C5-C3-C4-C5
 Sol #4: C5-C4-C3-C5-C4-C6-C2-C6-C3-C1
 Sol #5: C6-C3-C5-C4-C4-C5-C3-C6-C2-C1
 Sol #6: C6-C4-C5-C3-C4-C5-C2-C6-C3-C1

III. Maximum Options-Equipped Class First Production Algorithm

CSP의 자동차 부류 조립순서를 결정함에 있어 가장 중요한 사항은 $r:s$ 제약조건을 충족시키는 부류 순서 패턴이다. 이를 찾는 방법은 알려져 있지 않다. 만약, 이 부류 순서 패턴만 알 수 있다면 다항시간으로 해를 구하는 규칙을 제시할 수 있다.

본 장에서는 이 부류 순서 패턴을 쉽게 찾는 방법으로 $k \times n$ 행렬의 각 옵션에 대해 $n/s=l$ 개의 크기가 s 인 블록으로 분할하고, $p(r)$, $p=1,2,\dots,l$ 부여하는 방법을 제시한다. 이 행렬에 대해 최대 옵션 수를 장착하는 자동차 부류부터 먼저 생산하는 방식으로 n 대의 자동차에 대한 부류순서 패턴을 찾는다.

제안된 알고리즘을 최대 옵션 수 장착 자동차 부류 우선 생산 알고리즘 (maximum options-equipped class first production algorithm, MOCFPA)이라 하며, 다음과 같이 수행된다.

- n : 자동차 대수
- k : 옵션 수
- m : 부류 수, $m \leq 2^k - 1$
- $r:s$: s 개의 연속된 생산 중 옵션을 r 개만 장착할 수 있는 생산능력
- q : 각 부류별 자동차 생산 대수
- f : 각 부류별 옵션 장착 개수

Step 1. n 개 상자 준비

각 옵션별 $n/s=l$ 개의 크기가 s 인 블록에 $p(r)$, $p=1,2,\dots,l$ 부여

Step 2. for $i=1$ to $n/*$ 수행 복잡도 : $O(n) *$

- 가능한 옵션 패턴을 가진 $\max f$ 인 자동차 부류 선택, i 번째 상자에 배정
- (1) “1” 옵션의 동일 p 블록: $r=r-1$ 로 수정
- (2) “0” 패턴: $p(r)$ 삭제
- (3) 해당 자동차 부류: $q=q-1$ 로 수정

end

제안된 알고리즘은 n 대의 자동차를 생산하는 순서를 결정하는데 있어 Step 2에서 n 개 상자에 자동차들을 배정하는 방법으로 $O(n)$ 의 선형시간 수행 복잡도가 요구된다.

제안된 알고리즘을 P_1 문제에 대해 적용한 결과는 표 4와 같다. 4개의 자동차 부류가 2개씩의 옵션을 장착하므로 최대 옵션 장착 자동차 부류를 C1으로 설정하여 첫 번째로 조립한다고 가정하여 보자. 이 경우, 두 번째로 조립이 가능한 자동차 부류는 C1,C2,C3,C4 모두 가능하며, 여기서는 C3를 선택하였다. C3가 조립된 이후에는 O1,O2,O3만이 장착 가능하므로 이를 충족하는 자동차 부류는 C2이다. 따라서 세 번째로 C2가 배정된다. 이와 같은 방법으로 $n=8$ 의 순서를 결정하면 C1-C3-C2-C4-C3-C1-C4-C2의 순서를 얻을 수 있다.

제안된 MOCFPA의 성능을 요약하여 표 8에 제시하였다. 13개 데이터 중에서 6개는 해가 존재하지 않는다고 알려져 왔으나 제안된 MOCFPA는 모든 데이터에 대해 가능한 조립순서인 해가 존재함을 보였다.

기준에는 메타휴리스틱 방법으로 랜덤한 초기치를 갖는 수많은 시행을 통해 최적이나 근사 해를 얻거나 이에 해를 얻지 못한 경우도 발생하였다. 반면에, 본 논문에서 제안된 알고리즘은 $r:s$ 부류 순서 패턴을 찾는 규칙을 제시하여 이 문제의 해를 $O(n)$ 의 선형시간으로 찾을 수 있었다.

Table 8. Performance of MOCFPA

문제	Known Solution	MOCFPA
P_1	Satisfiable	Satisfiable
P_2	Satisfiable	Satisfiable
DSV88	Satisfiable	Satisfiable
4/72 (PR97)	Satisfiable	Satisfiable
CP-97	Infeasible	Satisfiable
6/76	Infeasible	Satisfiable
10/93	Infeasible	Satisfiable
16/81	Satisfiable	Satisfiable
19/71	Infeasible	Satisfiable
21/90	Infeasible	Satisfiable
36/92	Infeasible	Satisfiable
41/66	Satisfiable	Satisfiable
26/82	Satisfiable	Satisfiable

V. Conclusions

NP-완전으로 분류된 자동차 조립 순서 문제에 대해 본 논문에서는 $O(n)$ 의 선형시간 알고리즘이 존재할 수 있음을 보여 자동차 조립 순서 문제가 더 이상 난제가 아님을 실험적으로 증명하였다.

제안된 방법은 $k \times n$ 행렬에서 $r:s$ 의 제약조건을 충족시키기 위해 해당 옵션에 대해 $l=n/s$ 개의 블록으로 분할하고 $p(r), p=1,2,\dots,l$ 을 배정하였다. 다음으로, n 대의 자동차 조립 순서를 결정하기 위해 최대 옵션 수를 장착하는 자동차 부류부터 배정하고 해당 자동차 부류의 옵션을 장착하는 "1"의 셀에 대해 동일한 p 를 갖는 셀들의 $r=r-1$ 로 감소시켰으며, 해당 자동차 부류의 요구대수 $q=q-1$ 로 감소시키고, 다음으로 배정할 수 있는 자동차 부류를 결정하는 방식을 적용하였다.

실험 결과, 이와 같이 단순한 배정 규칙을 적용하여 $O(n)$ 의 선형시간으로 해를 구하였음에도 불구하고, 13개의 실험 데이터들 중에서 기존에 해가 존재하지 않는다고 알려진 6개 데이터에 대해서도 해를 구할 수 있음을 보였다.

결론적으로, 자동차 조립순서를 결정하는 문제에 대해 기존에는 다항시간 알고리즘이 알려져 있지 않으며, 또한 일부 데이터에 대해서는 조립순서를 결정하기가 불가능하다고 판단하였다. 반면에, 제안된 알고리즘은 $O(n)$ 의 선형 시간 복잡도로 자동차 조립순서를 결정할 수 있는 규칙 가진 알고리즘을 제안하였으며, 해가 존재하지 않다고 알려진 실험 데이터들에 대해서도 해를 구할 수 있음을 보였다. 따라서 제안된 알고리즘은 자동차 조립공정의 생산계획을 수립하는데 있어 실질적인 큰 도움을 줄 수 있을 것이다.

REFERENCES

- [1] T. Kis, "On the Complexity of the Car Sequencing Problem," *Operations Research Letters*, Vol. 32, No. 4, pp. 331-335, Jul. 2004.
- [2] B. Estellon and F. Gardi, "Car Sequencing is NP-hard: a Short Proof," *Journal of the Operational Research Society*, Vol. 64, No. 10, pp. 1503-1504, Oct. 2013.
- [3] C. Solnon, V. D. Cung, A. Nguyen, and C. Artigues, "The Car Sequencing Problem: Overview of State-of-the-art Methods and Industrial Case-Study of the ROADEF'2005 Challenge Problem," *European Journal of Operational Research*, Vol. 191, No. 3, pp. 912-927, Dec. 2008.
- [4] M. Fliedner and N. Boysen, "Solving the Car Sequencing Problem via Branch & Bound," *European*

Journal of Operational Research, Vol. 191, No. 3, pp. 1023-1042, Dec. 2008.

- [5] C. Gagné, M. Gravel, and W. L. Price, "Solving Real Car Sequencing Problems with Ant Colony Optimization," *European Journal of Operational Research*, Vol. 174, No. 3, pp. 1427-1448, Nov. 2006.
- [6] N. Zufferey, M. Studer, and E. A. Silver, "Tabu Search for a Car Sequencing Problem," *Proceedings of the 19th International Florida Artificial Intelligence Research Society*, pp. 457-462, 2006.
- [7] J. F. Cordeau, G. Laporte, and F. Pasin, "Iterated Tabu Search for the Car Sequencing Problem," *European Journal of Operational Research*, Vol. 191, No. 3, pp. 945-956, Dec. 2008.
- [8] D. Thiruvady, A. Ernst, and M. Wallace, "A Lagrangian-ACO Metaheuristic for Car Sequencing," *EURO Journal on Computational Optimization*, Vol. 2, No. 4, pp. 279-296, Nov. 2014.
- [9] H. Simonis, "ECLiPSE ELearning Website, Chapter 16: More Global Constraints (Car Sequencing)," <http://4c.ucc.ie/~hsimonis/ELearning/>, Cork Constraint Computation Centre, Computer Science Department, University College Cork, Ireland, 2006.
- [10] B. Smith, "CSPLib: A Problem Library for Constraints Car Sequencing, prob001: Car Sequencing," <http://ianm.host.cs.st-andrews.ac.uk/CSPLib/prob/prob001/>, 2014.
- [11] I. P. Gent, "Two Results on Car-Sequencing Problems," Technical report APES-02-1998, (<http://www.apes.cs.strath.ac.uk/apesreports.html>), pp. 1-7, Apr. 1998.

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.