

<https://doi.org/10.7236/JIIBC.2020.20.1.177>
JIIBC 2020-1-25

자동차 페인트 순서 문제의 연속된 최장 구간 색 승리 알고리즘

Sequential Longest Section Color Winning Algorithm for Car Paint Sequencing Problem

이상운*

Sang-Un Lee*

요약 본 논문은 차체가 조립되어 도장공장에 도착한 자동차들을 대상으로 동일한 색으로 최대한 그룹을 형성하여 도장 순서를 결정하는 자동차 페인트 순서 문제를 다룬다. 본 문제는 정확한 해를 다항시간으로 구하는 방법이 알려져 있지 않은 NP-완전으로 난제로 알려져 있다. 도장공장에서는 도장 색이 변경되면 이전 자동차 도장 색 페인트들을 완전히 제거하는 퍼징을 수행해야 하므로, 퍼징 횟수를 최소화시키는 것을 목표로 하고 있다. 본 논문에서는 버퍼에 도착한 자동차들의 이동 가능한 구간인 윈도우 개념에 기반하여 최소의 이동거리와 최소의 퍼징 횟수를 얻을 수 있도록, 자동차들을 동일 색, 도착 순서별로 정렬시키고, 구간 마라톤 경기를 수행하는데 있어 기본적으로는 연속적으로 가장 긴 구간을 차지하는 색 팀이 승리하는 방식을 적용하였다. 다만, 패자 팀이 더 이상 경기를 수행할 수 없는 구간이 존재하는 경우와 패자 팀에게 승리를 양보하고 이후의 경기에서 보다 많은 구간에서 승리하는 경우에는 승리의 우승컵을 해당 패자 팀에게 양보하여 모든 구간에서 모든 자동차 선수들이 한 번씩은 반드시 승리하는 방식을 적용하였다. 제안된 알고리즘은 n 대 자동차에 대해 $O(n \log n)$ 의 다항시간 복잡도로 간단하면서도 빠른 장점에도 불구하고, 다양한 사례들에 적용한 결과, 모든 실험 데이터들에 대해 최소의 이동거리와 최소의 퍼징 횟수를 얻을 수 있었다.

Abstract This paper deals with the car paint sequencing problem (CPSP) that the entrance sequence is to same colored group with maximum sequenced cars for the buffer arriving cars from the body shop. This problem classified by NP-complete problem because of the exact solution has not obtained within polynomial time. CPSP is aim to minimum pugging number that each pugging must be performs at color changing time in order to entirely cleaning the remaining previous color. To be obtain the minimum number of moving distance with window concept and minimum number of pugging, this paper sorts same color and arriving sequence. Then we basically decide the maximum length section color time to winner team using stage race method. For the case of the loser team with no more racing or yield to loser team and more longer stage in upcoming racing, the winner team give way to loser team. As a result, all cars(runners) are winner in any stage without fail. For n cars, the proposed algorithm has a advantage of simple and fast with $O(n \log n)$ polynomial time complexity, this algorithm can be get the minimum number of moving distance and purging for all of experimental data.

Key Words : buffer arriving sequence, color, window, purging, stage race

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자: 2019년 11월 23일, 수정완료: 2020년 1월 3일
게재확정일자: 2020년 2월 7일

Received: 23 November, 2019 / Revised: 3 January, 2020 /
Accepted: 7 February, 2020

*Corresponding Author: sulee@gwnu.ac.kr

Dept. of Multimedia Eng., Gangneung-Wonju National
University, Korea

1. 서 론

자동차 제조과정은 크게 프레스 가공으로 만든 패널들을 조립, 용접하여 차의 모양을 만드는 차체조립공장(body shop), 부식으로부터 소재 보호, 아름다운 외관 형성과 다른 차량과 구별하는 기능을 하는 페인팅 공장인 도장 공장(paint shop)을 거쳐, 차체에 실내 부품, 전장부품과 배선 배관작업을 하여 차량을 완성하는 조립라인(assembly line)의 의장 공장을 거쳐 최종 완성된 차량의 검수과정을 거친다^[1]. 여기서 비용을 최대로 절감할 수 있는 문제가 되는 공장은 도장 공장이다.

만약, 단일 색상이나 하루에 생산할 차량 색상별 대수 계획에 따라 동일 차종을 생산하는 제조과정이라면, 도장 공장을 선입선출로 진행하면 크게 문제가 되지 않는다. 그러나 다수의 색을 가진 다수의 차종을 생산하는 경우 차체가 조립되는 순서대로 도장 공장을 수행할 경우 심각한 문제가 발생한다. 왜냐하면, 도장 공장에서 색을 변경시키는 경우 이전 색의 페인트를 완전히 제거하는 퍼징(purging)을 수행한 이후 다시 새로운 색상의 페인트를 주입시켜야만 원하는 색상을 얻을 수 있다. 이 과정은 색상 교체에 따른 이전에 남아 있는 페인트를 버리고, 완전히 제거하기 위한 용제(solvent)의 추가적인 비용으로 인해 매우 고비용과 시간이 소요되기 때문이다^[2].

따라서 도장 공장에서는 가급적 퍼징 횟수를 최소화시키기 위해 동일 색을 최대로 가질 수 있는 일괄(batch) 형태로 도장작업의 순서를 결정한다. 이 문제에서는 차체 조립단계에서 해당 차체에 적용될 색이 이미 결정되어 있다고 가정한다^[3,4]. 또한 차체조립공장을 거쳐 나온 제품이 도장 공장에 투입되기 전에 순서를 변경시킬 수 있는 범위를 가지며 이를 윈도우(window)라 한다^[5].

이러한 차체조립공장 순서와 윈도우의 제약조건을 가진 상태에서 최소의 퍼징 횟수를 가지도록 색 배치를 하여 도장 공장 순서를 결정하는 문제를 자동차 페인트 순서 문제(car paint sequencing problem, CPSP) 또는 색 일괄 문제(colour batching problem, CBP)라 한다 [3,4]. 또한, 조립라인의 의장 공장에서 동일한 차량모델 별로 일괄 배정하는 문제를 일반적으로 자동차 순서 문제(car sequencing problem, CSP)라 한다^[6].

CSP에 대해서는 Direxl et al.^[6]와 Gagnéa et al.^[7]을 비롯하여 많은 연구가 수행되었으나 CPSP에 대해서는 연구가 활발히 이루어지지 않고 있다. 본 논문에서는 CPSP에 한정한다.

Epping et al.^[8]에 따르면 CPSP는 다항시간으로 정

확한 해를 찾을 수 있는 방법이 알려지지 않은 난제로 NP-완전(nondeterministic polynomial time-complete)으로 분류되어 있다. 따라서 근사 해를 다항시간으로 찾기 위한 다양한 메타휴리스틱 기법들을 적용하고 있는 실정이다.

실제 산업현장에서의 적용 사례를 살펴보면, BMW Plant Rosslyn의 BMW F30 3 시리즈 페인트 공장에서는 색의 민감도를 고려하여 밝은 색에서 어두운 색으로, 다시 어두운 색에서 밝은 색으로 사이클을 형성하는 방법으로 색을 배정하는데 있어 담금질기법(simulated annealing, SA)과 유전자 알고리즘(genetic algorithm, GA)을 결합한 하이브리드 메타휴리스틱 기법의 시뮬레이션-기반 최적화(simulation-based optimization)를 적용하고 있다^[9]. 또한, GM Plant에서는 이산형 사건 시뮬레이션(discrete event simulation, DES)모델과 결정최적화기(decision optimizer)를 채택하고 있으며, Decision optimizer에는 동적계획법(dynamic programming, DP)과 정수계획법(integer programming, IP)을 적용하고 있다^[10].

이론적 측면에서의 연구사례를 살펴보면, Akrotirianakis et al.^[3,4]의 제약논리계획법(constraint logic programming, CLP), Han et al.^[2]와 Moon et al.^[11]의 이산형 사건 시뮬레이션(DES), Zufferey et al.^[12]의 타부 탐색법(tabu search, TS), Cordeau et al.^[13]의 반복적 타부 탐색법(iterative TS), Sokol^[5]의 라그랑제 완화법(Lagrangian relaxation)을 적용한 선형계획 완화법(linear programming relaxation, LPR) 등이 있다. Risler et al.^[14]은 삽입법(insertion method, IM)과 순차적 방법(sequential method, SM)으로 초기 실현 가능 해를 얻고, 가변적 이웃 하강법(variable neighborhood descent, VND)과 SA로 최종 결과를 얻는 방법을 제안하였다.

이외에도 Joly와 Frein^[15]은 절삭 분기한정법(truncated branch-and-bound, TBB)을 적용한 점진적 구축-순서 알고리즘(progressive construction-sequence algorithm, PCSA)의 휴리스틱 기법과 SA, 가변적 이웃탐색법(variable neighborhood search, VNS)과 진화알고리즘(evolutionary algorithm, EA)의 3가지 메타휴리스틱 기법을 비교하였다. 또한, Solnon et al.^[16]은 정확한 해를 얻는 방법으로 제약계획법(constraint programming, CP), 정수계획법(IP)과 분기한정법(branch-and-bound, BB)이 있으며, 휴리스틱 접근법으로 근사 해를 얻는 방법으로 탐욕법(greedy algorithm), 지역탐색법(local

search, LS), 유전자 알고리즘(GA)와 개미집단최적화법 (ant colony optimization, ACO) 등이 있음을 제시하였다.

본 논문에서는 CPSP에 대해 다항시간으로 정확한 해를 찾을 수 있는 규칙을 제시한 탐욕적 알고리즘을 제안한다. 2장에서는 자동차 페인트 순서 문제를 구체적으로 살펴본다. 3장에서는 최소의 퍼징 횟수를 가진 페인트 순서를 다항시간으로 빠르게 결정하는 기법을 제안한다. 4장에서는 실험 데이터들에 제안된 알고리즘을 적용하여 알고리즘의 적합성을 검증한다.

II. 자동차 페인트 순서 문제

자동차 공장은 차체가 조립된 도장공장에 도착하는 순서대로 버퍼에 저장되며, 여기서는 동일 색상을 가질 차량들을 연속적으로 도장하기 위해 자동차의 순서를 재정렬하며, 이와 같이 재정렬된 순서로 도장 공장에 투입된다^[15]. 재정렬은 2가지 목적 중 어느 하나를 달성하고자 한다. 첫 번째는 순서 변경에 따른 이동 비용을 최소화시키고자 하며, 두 번째는 가급적 최소의 색 변경 횟수를 가지도록 하는 것이다.

본 논문에서 다루고자 하는 CPSP와 관련된 용어들은 다음과 같이 정의된다.

- 자동차 대수(number of cars): $i = 1, 2, \dots, n$
- 차체조립순서(body assembly order) 즉 도장공장에 도착하는 순서: $b = 1, 2, \dots, n$
- 색채 패턴(color pattern) : $c = 1, 2, \dots, m$
- 윈도우(이동 가능 위치) : w

여기서의 제약조건은 페인트 공장에 연속적으로 입고 되는 두 대의 자동차 색이 다를 경우 반드시 남아 있는 이전 색의 페인트를 버리고 용제로 깨끗이 세척한 이후에 다음 자동차 색의 페인트를 투입하는 퍼징을 수행해야만 한다. 이러한 제약조건에 퍼징 횟수를 최소화시키는 목적을 달성하기 위해 차체조립이 완료되고 페인트 공장을 대기하는 순서가 i 번째인 자동차의 c 색채를 도장 (paint, P)할 위치 P_i 는 윈도우 개념을 적용하여 식 (1)로 도장 공장에 투입될 위치(순서)를 변경시킬 수 있다.

$$P_{\min}^i \leq P_i \leq P_{\max}^i \quad (1)$$

where $P_{\min}^i = \max[i, i-w]$

$$P_{\max}^i = \min[i+w, n]$$

이 경우 버퍼에 도착하는 순서를 j , 도장공장에 투입될 순서로 색별로 그룹화되어 재정렬된 순서를 k 라 하면, 각 자동차 i 의 이동거리 d_i 들을 합한 총 이동거리는 식 (2)로 계산된다.

$$d = \sum_{i=1}^n d_i \quad (2)$$

where $d_i = |j-k|$

Akrotirianakis et al.^[4]로부터 인용된 표 1의 P_1 문제를 대상으로 살펴보면, 도장공장에 입고될 버퍼에 도착한 자동차 대수 $n=10$, 도장할 색채 패턴 수 c 는 R,B,G,Y의 4색, 차체조립이 완료되어 버퍼에 도착한 순서에서 도장을 위해 앞으로 이동 가능한 위치(윈도우) $w=3$ 이다. 표 1에서는 차체조립순서별 자동차의 도장을 위한 가능한 위치(possible location)는 식 (1)로 구하여 제시되어 있다.

표 1. CPSP의 예제(P_1 문제)
 Table 1. Example of CPSP(P_1 problem)

Body Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Color Pattern	R	B	G	Y	R	B	G	Y	R	B
Possible Location	1,2,3,4	1,2,3,4,5	1,2,3,4,5,6	1,2,3,4,5,6,7	2,3,4,5,6,7,8	3,4,5,6,7,8,9	4,5,6,7,8,9,10	5,6,7,8,9,10	6,7,8,9,10	7,8,9,10

possible 3-position moving($w=3$)



버퍼에 들어온 자동차의 색 순서는 R-B-G-Y-R-B-G-Y-R-B로 이 순서대로 도장을 수행하면 R↓B↓G↓Y↓R↓B↓G↓Y↓R↓B로 퍼징(↓) 횟수는 9회이다.

R(1)-B(2)-G(3)-Y(4)-R(5)-B(6)-G(7)-Y(8)-R(9)-B(10)의 순서에 대해 Akrotirianakis et al.^[4]은 CLP를 적용하여 5회 퍼징을 수행하는 87개의 가능한 해를 구하고 그 중 15개를 제시하였다. 이들은 표 2에 제시하였으며, 여기에서 동일한 결과를 나타낸 해 2,8,9와 해 3,15를 각각 하나로 취급하면 12개의 다른 결과를 얻는다. 이들 중 이동거리는 12와 14를 얻어 최소 이동거리 12를 나타낸 해 1의 RR↓GG↓BB↓YY↓R↓B, 해 2,8,9의

RR↓BB↓GG↓YY↓R↓B, 해 3,15의 R↓BB↓GG↓YY↓RR↓B, 해 4의 R↓BB↓YY↓GG↓RR↓B, 해 5의 R↓B↓GG↓RR↓YY↓BB, 해 6의 R↓B↓GG↓YY↓RR↓BB 6가지 결과들 중 어느 하나를 선택하면 된다.

III. 최대차량 색 승리 알고리즘

본 장에서는 이동거리 최소화와 퍼징 횟수 최소화의 두 가지 목적을 모두 달성할 수 있는 방법으로 연속적으로 도장 순서를 기다리는 자동차가 대수가 최대인 색채를 우선적으로 선택하여 도장 공장 입고 순서를 결정하는 최장 길이 색 승리 방법을 제안한다.

제안된 방법은 n 대의 모든 자동차들이 n 개의 어느 한 구간에서는 모두 승리자가 되는 구간 경기(race stage)의 계주경기(relay race)를 하는 방식이다. 본 알고리즘은 다음 가정들에 기반한다.

- 각 주자(자동차)가 달릴 수 있는 구간은 버퍼에 도착한 순서 i 와 이동 가능 구간 w 에 대해 $\max[i, i-w] \leq P_i \leq \min[i+w, n]$ 이다.
- 해당 구간에서 각 주자는 전력질주를 할 수 있다.
- 주자들이 연속적으로 배정되어 있는 팀은 해당 주자들이 위치한 마지막 구간까지 가장 빠른 속도로 도달할 수 있다.
- 만약, 특정 구간에 주자가 없는 팀은 해당 구간에서의 속도는 ∞ 로, 절대로 모든 구간에 주자가 있

는 팀을 이길 수 없어 패자 팀이 된다.

- 만약, 승리 팀이라 할지라도, 패자 팀의 첫 번째 주자가 더 이상 달릴 수 없어 승리자가 될 수 없는 경우 해당 구간에서의 승리의 우승점은 패자 팀에게 양보한다.
- 만약, 패자 팀에게 양보하고 이후의 구간에서 승리하는 것이 원래의 위치보다 더 가까워지면 패자 팀에게 양보한다.

승리 팀 색을 c_w , 패자 팀 색을 c_L 이라 하자. n 대의 도장 순서를 기다리는 자동차들을 m 개의 도장 색 수 레인에 도착순서대로 정렬시킨다. L_s 의 출발점(starting point)에서 각 색 레인별로 출발하여 연속적으로 최장 길이를 가진 팀을 연속적 길이의 마지막 위치 L_e 까지의 후보 승리 팀(candidate winner team)으로 결정한다. 여기서, 어느 한 주자라도 n 개 구간들 중 어느 한 구간에서는 반드시 승리할 수 있도록 하기 위해 발생하는 다음의 3가지 경우로 승리 팀을 결정하고, 다음 지점부터 동일한 방법으로 경기를 재시작하는 방법으로 종료(결승)점 L_n 까지 계속적으로 진행한다.

- 만약, 한 패자 팀 c_L 의 첫 번째 주자(자동차)가 해당 지점 전인 L_j 에서 더 이상 달릴 수 없는 경우, 승리 후보팀 c_w 를 승리 팀으로 최종적으로 결정하고, 승리 구간은 L_{j-1} 로 결정하고, 다음 경기는 L_j 부터 시작($s=j$)한다.
- 만약, 한 패자 팀 c_L 의 첫 번째 주자가 해당 지점 L_i 이후에는 더 이상 달릴 수 없으며, 패자 팀은

표 2. CLP로 구한 P_1 문제의 가능한 해

Table 2. Possible solution for P_1 problem using CLP

순서	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	d	도장 순서	퍼징 횟수	
도착순서	R(1)	B(2)	G(3)	Y(4)	R(5)	B(6)	G(7)	Y(8)	R(9)	B(10)				
도장 공장 입고 재 정렬 순서	해 1	R(1)	R(5)	G(3)	G(7)	B(2)	B(6)	Y(4)	Y(8)	R(9)	B(10)	12	RR↓GG↓BB↓YY↓T↓B	5
	해 7	R(1)	R(5)	G(3)	G(7)	B(2)	B(6)	Y(4)	Y(8)	B(10)	R(9)	14	RR↓GG↓BB↓YY↓B↓R	5
	해 2, 8, 9	R(1)	R(5)	B(2)	B(6)	G(3)	G(7)	Y(4)	Y(8)	R(9)	B(10)	12	RR↓BB↓GG↓YY↓R↓B	5
	해 10	R(1)	R(5)	B(2)	B(6)	G(3)	G(7)	Y(4)	Y(8)	B(10)	R(9)	14	RR↓BB↓GG↓YY↓B↓R	5
	해 11	R(1)	R(5)	B(2)	G(3)	G(7)	Y(4)	Y(8)	B(6)	B(10)	R(9)	14	RR↓B↓GG↓YY↓BB↓R	5
	해 13	R(1)	R(5)	B(2)	G(3)	G(7)	Y(4)	Y(8)	R(9)	B(6)	B(10)	14	RR↓B↓GG↓YY↓R↓BB	5
	해 12	R(1)	R(5)	B(2)	Y(4)	Y(8)	G(3)	G(7)	B(6)	B(10)	R(9)	14	RR↓B↓YY↓GG↓BB↓R	5
	해 14	R(1)	R(5)	B(2)	Y(4)	Y(8)	G(3)	G(7)	R(9)	B(6)	B(10)	14	RR↓B↓YY↓GG↓R↓BB	5
	해 3, 15	R(1)	B(2)	B(6)	G(3)	G(7)	Y(4)	Y(8)	R(5)	R(9)	B(10)	12	R↓BB↓GG↓YY↓RR↓B	5
	해 4	R(1)	B(2)	B(6)	Y(4)	Y(8)	G(3)	G(7)	R(5)	R(9)	B(10)	12	R↓BB↓YY↓GG↓RR↓B	5
해 5	R(1)	B(2)	G(3)	G(7)	R(5)	R(9)	Y(4)	Y(8)	B(6)	B(10)	12	R↓B↓GG↓RR↓YY↓BB	5	
해 6	R(1)	B(2)	G(3)	G(7)	Y(4)	Y(8)	R(5)	R(9)	B(6)	B(10)	12	R↓B↓GG↓YY↓RR↓BB	5	

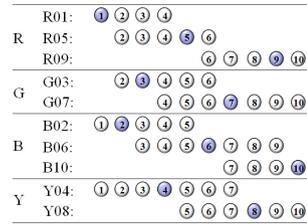
단지 주자가 1명인 경우, 이 경우에는 후보 승리 팀이 패자 팀에게 승리를 양보한다. 즉, 패자 팀이 L_s 에서 승리하고, 다음 경기는 L_{s+1} 에서 시작한다.

- 만약, 패자 팀의 어느 한 주자라도 L_e 이후에도 계속 달릴 수 있는 경우, (1) 패자 팀에게 승리를 양보하고, 이후 지점부터 승리 팀이 보다 더 긴 길이로 이길 수 있다면 이 방법을 선택한다. (2) 패자 팀에게 승리를 양보하고, 이후 지점부터 승리 팀이 보다 더 긴 길이를 선택하는 경우 원래의 위치에 보다 가까워지면 이 방법을 선택한다. (3) 그렇지 않으면 후보 승리 팀을 해당 구간 L_e 까지의 최종 승리 팀으로 결정하고, L_{e+1} 지점부터 다음 경기를 시작한다.

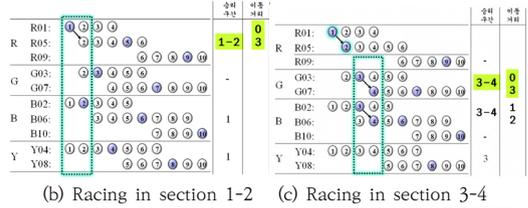
제안된 방법을 최장 길이 색 승리 알고리즘(maximum length color winning algorithm, MLCWA)이라하며, 그림 1에 제시하였다.

- Step 1. m 색 팀별로 레인을 배정하고, 버퍼에 대기 중인 차량별로 윈도우 개념을 적용하여 해당 팀에 배정한다. 즉, 색, 도장공장 버퍼 도착 순서별 오름차순 정렬.
- $s = 1$.
- Step 2. 경기 재시작 출발점(L_s)에서 각 색 m 팀별로 남아 있는 차량(선수)들을 대상으로 연속하여 배정된 차량 수 계산.
- Step 3. 가장 긴 길이 $\max l$ 를 가진 팀을 승리 후보 c_W 로 결정, c_W 가 승리한 지점은 $L_i, i = (s + \max l - 1)$ 로 잠정 결정.
- Step 4. c_W 의 승리한 위치 L_i 에 대해
 if 승리 후보 팀이 단독($|c_W| = 1$)인 경우 then
 if \exists (패자 팀 c_L 의 j 번째 자동차의 $P_{\max}^j < L_i$) then c_W 는 $P_{\max}^j - 1$ 까지의 최종 승자로 결정, $s = P_{\max}^j$
 else if \exists (1명 패자 팀 c_L 의 j 번째 자동차의 $P_{\max}^j = L_i$) then 승리 팀을 이 팀으로 교체, 현재 출발점 L_s 를 배정, $s = L_{s+1}$
 else
 if 패자 팀에게 승리를 양보하고 이후 지점부터 승리 후보 팀이 보다 더 긴 길이를 승리할 수 있는 경우 then 이 방법 선택
 else c_W 를 L_i 까지 승자 팀으로 결정, $s = L_{i+1}$.
- else if 동률 승자 팀이 u, v, \dots 로 2팀 이상($|c_W| \geq 2$)인 경우 then
 if u 의 $P_{\max}^u < L_i$ then u 팀을 승자 팀으로 결정
 else 원래 위치에 보다 가까운 팀을 승자 팀으로 결정.
- Step 5. 승자 팀의 차량들은 해당 위치 L_i 까지의 페인트 공장 순서로 확정하고, 경기에서 제외.
- Step 6. if \exists (승자로 결정되지 않은 선수들) then s 에서 출발하는 재경기를 수행하기 위해 Step 2로 복귀.
 else 경기 종료.

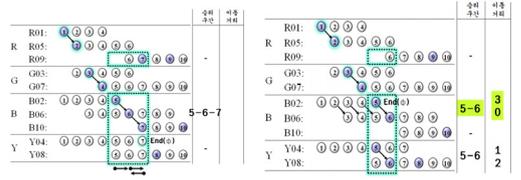
그림 1. 최대 길이 색 승리 알고리즘
 Fig. 1. Maximum length color winning algorithm (MLCWA)



(a) Car assigned to each color team for P_1 problem



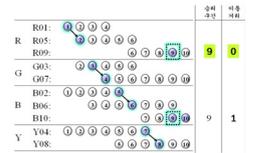
(b) Racing in section 1-2 (c) Racing in section 3-4



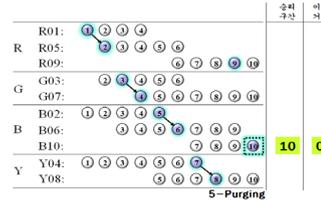
(d) Racing in section 5-6



(e) Racing in section 7-8

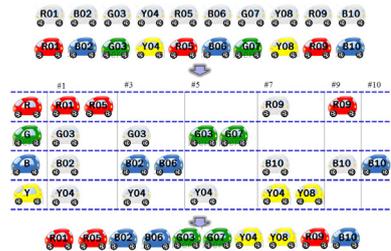


(f) Racing in section 9



(g) Racing in section 10

Car #	Assembly Location										Color	Moving Distance
	1	2	3	4	5	6	7	8	9	10		
R01											R	0
R05											R	3
R09											R	0
G03	1	1	1	1	1	1	1	1	1	1	G	2
G07											G	1
B02	1	1	1	1	1	1	1	1	1	1	B	1
B06											B	2
B10											B	0
Y04	1	1	1	1	1	1	1	1	1	1	Y	3
Y08											Y	0
Purging	P1	P2	P3	P4	P5	5 Purging				L2		



(h) Result of MLCWA

그림 2. P_1 문제에 관한 MLCWA
 Fig. 2. MLCWA for P_1 problem

Sequential Longest Section Color Winning Algorithm for Car Paint Sequencing Problem

Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9
Color Pattern	Blue(1)	Blue(2)	Red(3)	Yellow(4)	Black(5)	Green(6)	Red(7)	Blue(8)	Blue(9)
Possible Location	1,2,3,4	1,2,3,4,5	1,2,3,4,5,6	1,2,3,4,5,6,7	2,3,4,5,6,7,8	3,4,5,6,7,8,9	4,5,6,7,8,9	5,6,7,8,9	6,7,8,9

possible 3-position moving(w=3)

Blue(1)-Blue(2)-Red(3)-Yellow(4)-Black(5)-Green(6)-Red(7)-Blue(8)-Blue(9)

↓

Blue(1)-Blue(2)-Red(3)-Red(7)-Yellow(4)-Black(5)-Green(6)-Blue(8)-Blue(9)

5-purging, d:0-0-0-3-1-1-1-0-0=6

(a) P_2 problem

Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
Color Pattern	Yellow	Red	Red	Red	Red	Yellow	Green	Blue	Blue	Red	Green	Green
Possible Location	1,2,3,4	1,2,3,4,5	1,2,3,4,5,6	1,2,3,4,5,6,7	2,3,4,5,6,7,8	3,4,5,6,7,8,9	4,5,6,7,8,9,10	5,6,7,8,9,10,11	6,7,8,9,10,11,12	7,8,9,10,11,12	8,9,10,11,12	9,10,11,12

possible 3-position moving (3 point blocks, w=3)

Y(1)-R(2)-R(3)-R(4)-R(5)-Y(6)-G(7)-B(8)-B(9)-R(10)-G(11)-G(12)

↓

R(2)-R(3)-Y(1)-Y(6)-R(4)-R(5)-R(10)-B(8)-B(9)-G(7)-G(11)-G(12)

4-purging, d:1-1-2-2-1-1-3-0-0-3-0-0-14

(b) P_3 problem

Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18
Color Pattern	Black	White	Black	Black	White	Black	Black	White	Black	Black	White	Black	Black	White	Black	Black	White	Black
Possible Location	[1,7]	[1,8]	[1,9]	[1,10]	[1,11]	[1,12]	[1, 13]	[2,14]	[3,15]	[4,16]	[5,17]	[6,18]	[7,18]	[8,18]	[9,18]	[10,18]	[11,18]	[12,18]

possible 6-position moving (6 point blocks, w=6)

B(1)-W(2)-B(3)-B(4)-W(5)-B(6)-B(7)-W(8)-B(9)-B(10)-W(11)-B(12)-B(13)-W(14)-B(15)-B(16)-W(17)-B(18)

↓

B(1)-B(4)-B(6)-B(7)-B(10)-B(12)-B(3)-W(2)-W(5)-W(8)-W(11)-W(14)-W(17)-B(9)-B(15)-B(13)-B(16)-B(18)

2-purging, d:0-2-3-3-5-6-4-6-4-2-0-2-4-5-0-3-1-0=50

(c) P_4 problem

Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30
Color Pattern	B	R	B	R	B	R	R	R	R	R	R	R	B	R	B	R	B	B	B	B	B	B	B	B	B	R	B	R	B	R

possible 2-position moving (w=2)

B(1)-R(2)-B(3)-R(4)-B(5)-R(6)-R(7)-R(8)-R(9)-R(10)-R(11)-R(12)-R(13)-B(14)-R(15)-B(16)-R(17)-B(18)-B(19)-B(20)
-B(21)-B(22)-B(23)-B(24)-B(25)-R(26)-B(27)-R(28)-B(29)-R(30)

↓

B(1)-B(3)-B(5)-R(2)-R(4)-R(6)-R(7)-R(8)-R(9)-R(10)-R(11)-R(12)-R(13)-R(15)-R(17)-B(14)-B(16)-B(18)-B(19)-B(20)
-B(21)-B(22)-B(23)-B(24)-B(25)-B(27)-B(29)-R(26)-R(28)-R(30)

3-purging, d: 0-1-2-2-1-0-0-0-0-0-0-0-0-0-1-2-2-1-0-0-0-0-0-0-0-0-0-1-2-2-1-0=18

(d) P_5 problem

Assembly Order	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18
Color Pattern	G	R	G	R	G	R	R	B	R	B	R	B	B	G	B	G	B	G
Assembly Order	#19	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30	#31	#32	#33	#34	#35	#36
Color Pattern	G	R	G	R	G	R	R	B	R	B	R	B	B	G	B	G	B	G

possible 2-position moving (w=2)

G(1)-R(2)-G(3)-R(4)-G(5)-R(6)-R(7)-B(8)-R(9)-B(10)-R(11)-B(12)-R(13)-G(14)-B(15)-G(16)-B(17)-G(18)-G(19)-R(20)
-G(21)-R(22)-G(23)-R(24)-R(25)-B(26)-R(27)-B(28)-R(29)-B(30)-B(31)-G(32)-B(33)-G(34)-B(35)-G(36)

↓

G(1)-G(3)-G(5)-R(2)-R(4)-R(6)-R(7)-R(9)-R(11)-B(8)-B(10)-B(12)-B(13)-B(15)-B(17)-G(14)-G(16)-G(18)-G(19)-G(21)
-G(23)-R(20)-R(22)-R(24)-R(25)-R(27)-R(29)-B(26)-B(28)-B(30)-B(31)-B(33)-B(35)-G(32)-G(34)-G(36)

6-purging, d: 0-1-2-2-1-0-0-1-2-2-1-0-0-1-2-2-1-0-0-1-2-2-1-0-0-1-2-2-1-0-0-1-2-2-1-0-0-1-2-2-1-0=36

(e) P_6 problem

횃수를 언도록 도장 색별로 그룹을 형성시킬 수 있음을 알 수 있다.

Blue(01)Blue(02) ↓ R(03)R(07) ↓ Y(4) ↓ Black(05) ↓ G(06)
↓ Blue(08)Blue(09)= 5-purging, Moving distance=6

(a) P_2 problem

R(2)R(3)R(4) ↓ Y(1)Y(6) ↓ R(5)R(10) ↓ B(8)B(9) ↓
G(7)G(11)G(12) = 4-purging, Moving distance=14

(b) P_3 problem

B(1)B(3)B(4)B(6)B(7)B(9)B(10) ↓ W(2)W(5)W
(8)W(11)W(14)W(17) ↓ B(12)B(13)B(15)B(16)B(18)
=2-purging, Moving distance=37

(c) P_4 problem

B(1)B(3)B(5) ↓ R(2)R(4)R(6)R(7)R(8)R(9)R(10)
R(11)R(12)R(13) R(15)R(17) ↓ B(14)B(16)B(18)B(19)B(20)
B(21)B(22)B(23)B(24)B(25) B(27)B(29) ↓ R(26)R(28)R(30)
=3-purging, Moving distance=18

(d) P_5 problem

G(1)G(3)G(5) ↓ R(2)R(4)R(6)R(7)R(9)R(11) ↓
B(8)B(10)B(12) B(13)B(15)B(17) ↓ G(14)G(16)G(18)G(19)
G(21)G(23) ↓ R(20)R(22)R(24) R(25)R(27)R(29) ↓ B(26)B(28)
B(30)B(31)B(33)B(35) ↓ G(32)G(34) G(36)=6-purging, Moving
distance=36

(e) P_6 problem

B(02)B(05) ↓ Y(01) ↓ T(03)T(06)T(09) ↓
W(04)W(07)W(10)W(13) ↓ P(08)P(11)P(14)P(17) ↓
R(12)R(15)R(18)R(21) ↓ G(16)G(19)G(22)G(25) ↓
Y(20)Y(23)Y(26)Y(29) ↓ B(24)B(27)B(30)B(33) ↓
T(28)T(31)T(34)T(37) ↓ W(32)W(35)W(38)W(41) ↓
P(36)P(39)P(42)P(45) ↓ R(40)R(43)R(46)R(49) ↓
G(44)G(47)G(50)G(53) ↓ Y(48)Y(51)Y(54)Y(57) ↓
B(52)B(55)B(58) ↓ T(56)T(59) ↓ W(60)=17 Purging,
Moving distance=114

(f) P_7 problem

B(02)B(05) ↓ T(01)T(03)T(06)T(09) ↓
W(04)W(07)W(10)W(13) ↓ P(08)P(11)P(14)P(17) ↓
R(12)R(15)R(18)R(21) ↓ G(16)G(19)G(22)G(25) ↓
Y(20)Y(23)Y(26)Y(29) ↓ B(24)B(27)B(30)B(33) ↓
T(28)T(31)T(34)T(37) ↓ W(32)W(35)W(38)W(41) ↓
P(36)P(39)P(42)P(45) ↓ R(40)R(43)R(46)R(49) ↓
G(44)G(47)G(50)G(53) ↓ Y(48)Y(51)Y(54)Y(57) ↓
B(52)B(55)B(58)B(60) ↓ T(56)T(59)=15 Purging, Moving
distance=118

(g) P_8 problem

그림 4. 실험 데이터에 관한 MLCWA
Fig. 4. MLCWA for experimental data

제안된 MLCWA의 성능을 비교 분석한 결과는 표 3에 제시되어 있다. 제안된 MLCWA는 LP, CLP와 IWA로 수행한 8개의 실험 데이터 모두에 대해 최소 이동거리로 최소 퍼징 횃수를 얻을 수 있는 최적의 알고리즘임을 알 수 있다.

표 3. 알고리즘 성능

Table 3. Performance of algorithms

문제	성능	알고리즘		
		LP[3,4]/CLP[5]	IWA[5]	MLCWA
P_1	이동거리	12	-	12
	퍼징 횃수	5	-	5
P_2	이동거리	6	-	6
	퍼징 횃수	5	-	5
P_3	이동거리	-	14	14
	퍼징 횃수	-	4	4
P_4	이동거리	-	50	37
	퍼징 횃수	-	2	2
P_5	이동거리	-	18	18
	퍼징 횃수	-	3	3
P_6	이동거리	-	36	36
	퍼징 횃수	-	6	6
P_7	이동거리	-	-	114
	퍼징 횃수	-	-	17
P_8	이동거리	-	118	118
	퍼징 횃수	-	15	15

제안된 알고리즘은 n 대의 자동차에 대해 도장 색과 버퍼 도착 순서별로 오름차순 정렬하는데 $O(n \log n)$ 의 수행복잡도를 가지며, 순차적인 방법으로 구간을 색별로 그룹화시키는데 $O(n)$ 의 수행복잡도가 요구된다. 따라서 제안된 알고리즘은 $O(n \log n)$ 의 수행 복잡도를 갖는 다항시간 알고리즘임에도 불구하고, 최소 이동거리와 최소 퍼징 횃수의 2가지 목표를 달성할 수 있는 성능을 갖고 있음을 알 수 있다.

V. 결론 및 향후 연구

본 논문은 정확한 해를 다항시간으로 찾을 수 있는 규칙을 가진 알고리즘이 알려져 있지 않아 NP-완전으로 분류되어 있는 자동차 페인트 순서 문제를 다루었다.

본 논문에서는 차체 조립공장에서 나온 순서대로 버퍼에 도착한 자동차들을 대상으로 윈도우 개념의 이동 가능한 구간을 적용하여 최소의 이동거리와 최소의 퍼징 횃수를 얻을 수 있는 도장 공장 입고 순서를 결정하였다.

이와 같이 최소 이동거리와 퍼징 횃수의 두 가지 목표를 동시에 달성하기 위해, 본 논문에서는 버퍼에 도착한 자동차들을 동일 색, 도착 순서별로 정렬시키고, 구간 마

라톤 경기를 수행하는 방식을 적용하였다. 기본적으로는 연속적으로 가장 긴 구간을 차지하는 색 팀을 승리자로 결정하였다. 단, 패자 팀이 더 이상 경기를 수행할 수 없는 경우와 패자 팀에게 승리를 양보하고 이후의 경기에서 보다 많은 구간에서 승리하는 경우에는 해당 패자 팀에게 승리를 양보하고 이후의 구간에서 재경기를 수행하는 방식을 적용하여 모든 자동차 선수들이 어느 한 구간에서는 반드시 승리하도록 하였다.

n 대 자동차에 대해 제안된 알고리즘은 $O(n \log n)$ 수행 복잡도로 빠르며, 위와 같이 단순한 승리 팀 결정 규칙을 갖는 장점에도 불구하고, 8개의 다양한 사례들에 적용한 결과, 모든 데이터들에 대해 최소 이동거리와 퍼징 횟수를 얻을 수 있었다.

제안된 알고리즘을 실제의 자동차 제조공장에서 조립 라인 의장 공장에 적용되는 CSP[17]와 결합시켜 도장 순서를 결정하는데 적용하면 커다란 비용 절감 효과를 얻을 수 있을 것이다.

References

- [1] A. Nguyen and V. D. Cung, "Le Problème du Car Sequencing Renault et le Challenge ROADEF' 2005," Proceedings of Journées Francophones de Programmation par Contraintes, pp. 3-10, 2005.
- [2] Y. H. Han, C. Zhou, B. Bras, L. McGinnis, C. Carmichael, and P. J. Newcomb, "Simulation in Automotive Industries: Paint Line Color Change Reduction in Automobile Assembly Through Simulation," Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, pp. 1204-1209, 2003. ISBN:0-7803-8132-7
- [3] I. Akrotirianakis et al., "SAS/OR(R) 13.2 User's Guide: Constraint Programming, Chapter 3. The CLP Procedure, Example 3.5 Car Painting Problem," SAS Institute Inc., Cary, NC, USA, pp. 73-75, Aug. 2014.
- [4] I. Akrotirianakis et al., "SAS/OR(S) 14.1 User's Guide: Mathematical Programming, Chapter 6. The Constraint Programming Solver, Example 6.5 Car Painting Problem," SAS Institute Inc., Cary, NC, USA, pp. 231-232, July, 2015.
- [5] J. S. Sokol, "Optimizing Paint Blocking in an Automobile Assembly Line: An Application of Specialized TSP's," Ph. D. diss., Massachusetts Institute of Technology, Jun. 1999.
- [6] A. Drexler, A. Kimms, and L. Matthießen, "Algorithms for the Car Sequencing and the Level Scheduling Problem," Journal of Scheduling, Vol. 9, No. 2, pp. 153-176, Apr. 2006.
- [7] C. Gagnéa, M. Gravela, and W. L. Priceb, "Solving Real Car Sequencing Problems with Ant Colony Optimization," European Journal of Operational Research, Vol. 174, No. 3, pp. 1427-1448, Nov. 2006. doi:10.1016/j.ejor.2005.02.063
- [8] T. Epping, W. Hochstättler, and P. Oertel, "Complexity Results on a Paint Shop Problem," Discrete Applied Mathematics, Vol. 136, No. 2-3, pp. 217-226, Feb. 2004. doi:10.1016/S0166-218X(03)00442-6
- [9] S. Enslin, "Simulation-based Optimisation to solve a Car Sequencing Problem with Stochastic Rework," University of Pretoria, [http://www.up.ac.za/media/shared/101/ZP_Files/2015 Projects Evening/Posters/1-bpj420-enslin.zp73983.pdf](http://www.up.ac.za/media/shared/101/ZP_Files/2015%20Projects%20Evening/Posters/1-bpj420-enslin.zp73983.pdf), pp. 1-1, 2015.
- [10] M. M. L. Elahi, K. Rajpurohit, J. M. Rosenberger, G. Zaruba, and J. Priest, "Optimizing Real-time Vehicle Sequencing of a Paint Shop Conveyor System," Omega, Vol. 55, pp. 61-72, Sep. 2015. doi:10.1016/j.omega.2015.02.003
- [11] D. H. Moon, H. S. Kim, C. Song, and K. W. Kim, "A Simulation Analysis on the Validity of Color Rescheduling Storage in an Automobile Painting Shop," IE Interfaces, Vol. 16, No. 2, pp. 211-221, Jun. 2003.
- [12] 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 Conference, Melbourne, USA, pp. 457-462, 2006.
- [13] 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. doi:10.1016/j.ejor.2007.04.048
- [14] M. Risler, M. Chiarandini, L. Paquette, T. Schiavinotto, and T. Stiitzle, "An Algorithm for the Car Sequencing Problem of the ROADEF 2005 Challenge," TU Darmstadt, Computer Science, Intellectics Group, Technical Report AIDA-04-06, pp. 1-6, Nov. 2004.
- [15] A. Joly and Y. Frein, "Heuristic for an Industrial Car Sequencing Problem Considering Paint and Assembly Shop Objectives," Computers & Industrial Engineering, Vol. 55, No. 2, pp. 295-310, Sep. 2008. doi:10.1016/j.cie.2007.12.014
- [16] 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, 2008. doi:10.1016/j.ejor.2007.04.033
- [17] S. U. Lee, "Maximum Options-Equipped Class First-Production Algorithm for Car Sequencing Problem," Journal of KSCI, Vol. 20, No. 9, pp. 105-111, Sep. 2015. doi:10.9708/jksci.2015.20.9.105

저 자 소 개

이 상 운(정회원)



- 1987년: 한국항공대학교 항공전자공학과 (학사)
- 1997년: 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과

전임강사

- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 그래프 알고리즘
- e-mail : sulee@gwnu.ac.kr

※ 이 논문은「2019년도 강릉원주대학교 학술연구조성비」지원에 의하여 수행되었음.