

자동차 조립라인에서 부품사용의 일정을 유지할 위한 투입순서 결정

- Sequencing to keep a constant rate of part
usage in car assembly lines -

현철주*

Hyun Chul Ju

Abstract

This paper considers the sequencing of products in car assembly lines under Just-In-Time systems. Under Just-In-Time systems, the most important goal for the sequencing problem is to keep a constant rate of usage every part used by the systems.

In this paper, tabu search technique for this problem is proposed. Tabu search is a heuristic method which can provide a near optimal solution in real time. The performance of proposed technique is compared with existing heuristic methods in terms of solution quality and computation time. Various examples are presented and experimental results are reported to demonstrate the efficiency of the technique.

1. 서론

자동차 조립라인은 공장부지 및 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하고 있다. 예로서 승용차 라인에서는 여러 가지 차종을 생산하고 있으며, 하나의 차종에 대해서도 엔진, 문(door)의 형태, 자동변속기 또는 차의 색상 등의 옵션이 조합되어 각각 다른 모델이 된다. 자동차 조립라인의 효율적인 이용을 위해서는 라인밸런싱과 모델의 투입순서를 결정하는 문제가 중요하다.

* 정인대학 품질관리과

라인밸런싱은 자동차를 조립하기 위해 필요한 작업들을 선행제약을 어기지 않고 각 작업장에 균형적으로 할당하는 문제이며, 투입순서 문제는 모델의 수요량에 대한 모델의 투입순서를 다루는 문제이다. 투입순서문제는 라인밸런싱문제보다 단기적인 측면을 가지고 있으며, 각 모델의 수요율은 자주 변하기 때문에 주기적으로 해결해야 할 필요성이 있다. 본 연구는 자동차 조립라인에서 모델의 투입순서 문제를 다루고 있다.

자동차 조립라인의 투입순서에 관한 연구는 Kilbridge와 Wester[11]에 의해 처음으로 제시된 이래 라인길이의 최소화[3, 6], 가외작업의 최소화[2, 17], 라인정지의 최소화[9], 부품사용의 일정을 유지[7, 8] 등 다양한 목적에 관해 여러 연구가 이루어 졌다. 특히 Just-In-Time 생산 시스템에서는 재고비용을 줄이고 고객의 다양한 수요를 적시에 만족시키기 위해 부품사용의 일정을 유지에 관한 목적을 흔히 사용하고 있다. 부품사용의 일정을 유지하는 제품의 투입순서 결정은 완제품 및 재공품 재고의 수준을 최소로 할 수 있으며, 외주업체가 부품의 일정량을 일정간격으로 납품할 수 있고, 수요의 변화에 빠르게 대처할 수 있다는 장점을 가진다[8].

자동차 조립라인의 투입순서 문제는 조합최적화 문제로 수요변화, 결품으로 인한 투입순서의 재결정에는 실시간 처리가 요청된다. 본 연구에서는 자동차 조립라인에서 부품의 사용율을 일정하게 유지하기 위한 투입순서를 실시간에 효율적으로 구하기 위해 Tabu Search 기법을 이용하고자 한다.

Tabu Search는 인간의 기억과정을 이용한 탐색기법으로 해의 탐색과정을 일정기간 기억함으로써 해의 순환을 막고, 이웃해 중에서 가장 좋은 해로 이동함으로써 부분최적(Local Optimum)에서 벗어나 최적해 또는 근사 최적해를 구할 수 있는 기법이다. Tabu Search는 Glover에 의해 제안되어 여러 분야에서 활발히 적용되었으며, 특히 외판원문제와 Flow-shop 순서문제 등의 조합 최적화 문제에 적합한 기법으로 알려져 있다[4, 12]. 본 연구에서는 Tabu Search 기법을 다루는 문제에 적용하여 기존에 제시된 발견적 기법과 해의 탐색성능과 계산 시간의 측면에서 비교분석하고자 한다.

본 연구의 구성은 아래와 같다. 제 2장에서 Miltenburg와 Sinnamon[8]이 제시한 수리모형을 살펴보고, 제 3장에서는 Tabu Search 기법을 다룬다. 그리고 제 4장에서는 실험을 통하여 Tabu Search와 발견적 기법을 비교분석한다. 제 5장은 결론으로 구성 되어 있다.

2. 수리모형

자동차 조립라인의 투입순서문제는 Kilbridge와 Wester에 의해 처음으로 제기된 이후 투입순서의 목적으로 라인길이의 최소화, 라인정지의 최소화, 부품사용의 일정을 유지 등 다양한 목적이 제시되었다[1, 16, 17]. 특히 Just-In-Time 생산 시스템에서는 부품사용의 일정을 유지하기 위한 목적이 다른 목적보다 중요시되고 있으며, 이에 관한 최적의 투입순서를 구하기 위한 여러 가지 기법들이 제시되어 왔다. Monden[18]은

부품 사용율을 가능하면 일정하게 유지하기 위해 Toyota 자동차 회사에서 사용하는 목표추적법(Goal-Chasing Method)이라는 생산계획의 평준화방법을 설명하였다. 또한 Miltenburg[7]는 Just-In-Time 생산시스템에서 단위시간당 부품의 사용을 가능하면 일정하게 유지하는 투입순서를 구하는 수리모형과 발견적기법을 제시하였고, Miltenburg와 Sinnamon[8]는 Miltenburg[7]의 연구를 확장하여 제품뿐만 아니라 제품을 구성하고 있는 부품들의 실제 생산량과 수요량의 편차를 최소로 하는 투입순서를 구하는 수리모형과 발견적기법을 제시하였다. Sumichrast *et al.*[14]은 Just-In-Time 상황에서 제안된 여러 형태의 투입순서를 결정하는 방법들을 작업자의 유희 및 부품 사용율의 변동 등과 같은 작업의 비효율성 측면에서 비교분석하였다.

Just-In-Time 생산 시스템에서는 평준화생산을 위해 계획기간(Planning horizon)동안 생산될 모델의 개수를 반복 생산하여 계획기간의 수요를 만족시키는 사이클 생산을 흔히 사용하고 있다. 본 연구에서는 사이클 생산에서 제품뿐만 아니라 제품을 구성하고 있는 반제품, 부품, 원재료들의 사용량을 일정하게 유지하기 위한 투입순서를 구하기 위해 Miltenburg와 Sinamon[8]이 제시한 수리모형을 사용한다. 수리모형은 다음과 같다.

$$\begin{aligned} \text{MIN. } & \sum_{k=1}^{DT_1} \sum_{j=0}^4 \sum_{i=0}^{n_i} w_j (x_{ijk} - XT_{jk}r_{ij})^2 \\ \text{S.T. } & XT_{1k} = k \quad k=1, 2, \dots, DT_1 \\ & 0 \leq x_{ik} - x_{i(k-1)} \leq 1 \\ & x_{ik} \text{는 정수, } i=1, 2, \dots, k=1, 2, \dots, DT_1 \end{aligned}$$

수리모형에서 목적함수는 제품뿐만 아니라 제품을 구성하고 있는 반제품, 부품, 원재료들의 실제 생산량(x_{ijk})과 수요량($XT_{jk}r_{ij}$)의 편차 합을 나타내며, 이를 최소로 하기 위한 투입순서(x_{ik})를 구하고자 한다. 목적함수에서 w_j 는 각 수준(level)의 중요도를 나타낸다. 첫 번째 제약식은 k 개의 제품이 정확히 k 단계 동안에 생산되어야 함을 나타내며, DT_1 은 계획기간동안에 생산되어야 하는 총 제품수를 나타낸다. 두 번째와 세 번째 제약식은 각각 주어진 단계에서 각 제품은 한 단위를 초과하여 생산하거나 분할하여 생산할 수 없다는 정수조건을 나타낸다.

3. Tabu Search

Tabu Search(TS)는 부분최적에서 벗어나 최적해 또는 근사 최적해를 구할 수 있는 탐색적 방법으로 Glover[4]에 의해 연구되기 시작하여, 유전알고리즘, 신경망이론과 함께 일정계획, 고용계획, 공간계획과 구조설계, 통신경로문제 등의 조합최적화 문제에 널리 적용되고 있다. Takashi *et al.*[10]는 외판원문제에 Tabu Search를 적용하여 이

웃해의 집합을 줄이는 방법을 제시하고 있으며, Beatty[15]는 일정계획 문제에 대해 Tabu Search를 적용하여 타기법과 비교분석하고 있다. 또한 Tabu Search에 관한 이론 연구로는 이동의 유형과 그 적용방법, 순환을 막는 방법, Tabu목록의 관리방법 등에 관한 많은 연구가 이루어 졌다[5, 10, 13]. Tabu Search의 절차는 다음과 같다[4].

단계 1 : 초기화

- 1) Tabu 속성, Tabu 목록의 크기, 열망수준과 종료조건을 결정한다.
- 2) 초기가능해를 구하여 현재해와 최선해로 둔다.
- 3) Tabu 목록을 비워둔다.

단계 2 : 이웃해의 생성

- 1) 현재해로부터 Tabu 이동이 아니거나 Tabu이동이지만 열망수준을 만족하는 이동에 대해 이웃해를 생성한다.

단계 3 : 현재해와 최선해의 수정

- 1) 생성된 이웃해중에서 가장 좋은 이웃해를 현재해로 둔다.
- 2) 현재해가 최선해보다 좋은 해이면 현재해를 최선해로 둔다.

단계 4 : Tabu 목록의 수정

- 1) 새로운 현재해의 이동속성을 Tabu목록에 기록한다.
- 2) 만약 Tabu목록에 저장된 Tabu 속성의 수가 Tabu목록의 크기보다 크면 가장 먼저 기록된 Tabu속성을 삭제한다.

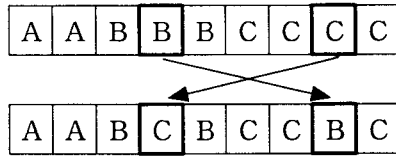
단계 5 : 종료조건이 만족하면 알고리즘의 수행을 끝내고 그렇지 않으면 단계 2로 간다.

Tabu목록은 해의 이동속성을 일정기간 기억하는 목록으로, 이 목록을 만들기 위해서는 해의 이동속성을 나타내는 Tabu속성이 정의되어야 한다. 순서문제에서 Tabu속성은 이동 원소와 그 위치 등에 따라 여러 형태가 있다[10]. 본 연구에서는 Tabu속성으로 흔히 이용되는 $(\Pi(i), \Pi(j), i, j)$ 을 사용하였다. 이것은 이동으로 인하여 i 위치에 있는 원소 $\Pi(i)$ 와 j 위치에 있는 원소 $\Pi(j)$ 가 변함으로써 새로운 이웃해가 생성된다는 것을 의미한다. 그리고 Tabu목록의 크기는 Tabu상태로 제약할 수 있는 이동의 최대수를 나타낸다.

열망수준은 어떤 이동이 Tabu상태이지만 이를 해제하여 해의 이동을 가능하게 하는 기준이다. 본 연구에서는 열망수준은 현재까지 탐색한 가장 좋은 해, 즉 최선해로 두었다. 또한 종료조건으로는 반복수, 탐색한 이웃해의 수, 최선해를 개선하지 못한 회수 등이 있으나, 본 연구에서는 현재까지 탐색한 이웃해의 개수를 종료조건으로 사용하였다.

이웃해의 생성방법은 교환(Swap), 삽입(Insertion), 역순(Inversion) 등 여러 가지 방법이 있으나, 본 연구에서는 두개의 원소의 위치를 서로 바꾸는 교환 이동을 사용함

다. 교환 이동은 서로 같지 않은 임의의 두 원소를 교환하여 이웃해를 생성하는 방법으로, <그림 1>과 같이 교환될 원소로 B원소와 C원소가 선택되었다면, 두 원소를 서로 교환하여 이웃해를 생성한다. 이때 사용된 Tabu 속성은 $(\Pi(i), \Pi(j), i, j) = (B, C, 4, 8)$ 가 된다.



<그림 1> 교환이동

4. 실험결과

이 장에서는 다양한 예제문제를 통해 Tabu Search와 Miltenberg와 Sinnamon[8]이 제시했던 두 가지 탐색적 방법(이하 MS_H1과 MS_H2로 표기)을 해의 효율과 계산시간의 측면에서 비교분석하고자 한다.

비교분석을 위한 각 기법들은 C++ 언어로 프로그램되어 Pentium 230MHz CPU를 사용하여 실험하였으며, Tabu Search는 실험의 신뢰도를 높이기 위하여 매 실험마다 초기해를 달리하여 30회 반복 실험하였다.

4.1 예제 1

본 연구에서는 첫번째 예제로 Miltenburg와 Sinnamon[8]이 제시한 예제를 사용하였다. 이 예제는 3종류의 모델을 각각 6000, 6000, 1000개를 생산한다. 따라서 한 사이클당 생산되는 모델별 생산제품 수는 각각 6, 6, 1개가 되며, 한 사이클당 생산되는 총 제품 수는 13개가 된다.

예제의 부품구성은 4수준(1수준 : 제품, 2수준 : 반제품, 3수준 : 부품, 4수준 : 원재료)으로 구성되어 있다. 즉, 각 제품은 3개의 반제품으로, 각 반제품은 4개의 부품으로, 또한 각 부품은 3개의 원재료로 이루어져 있다. 이에 관한 부품구성표는 <표 1>과 같다.

<표 1> 예제 1에 대한 부품구성표

		제 품			반 제 품			부 품							
		1	2	3				1	2	3	4				
반 제 품	1	1	0	0	부 품	1	1	0	1	원 재 료	1	1	0	0	1
	2	1	1	1		2	1	1	0		2	0	1	0	1
	3	0	0	4		3	0	1	0		3	1	0	1	0
					4	0	0	4							

<표 2>는 본 연구에서 제시한 Tabu Search, 그리고 MS_H1, MS_H2를 사용하여 구한 목적함수 값과 그때의 투입순서를 나타내고 있다. Tabu Search에서 이동방법은 교환이동을, Tabu 목록의 크기는 7로 두고 528개의 이웃해를 탐색하였다. Tabu Search의 결과치는 초기해를 달리하여 30회 실험한 결과의 평균치를 사용하였다.

<표 2> 해의 성능 비교분석

문제	가중치	MS_H1		MS_H2		Tabu Search	
		Variance	Sequence	Variance	Sequence	Variance	Sequence
I-1	(1,1,1,1)	338.183	2122123111212	324.033*	2122113112212	324.033*	2122113112212
I-2	(0,1,1,1)	352.310	2212221311112	325.662	2212213111122	315.026*	2212113112122
I-3	(0,0,1,1)	382.357	2222222311111	342.883	2222221311111	291.369*	2221113111222
I-4	(0,0,0,1)	120.173	2222223111111	116.844	2222213111112	114.180*	2221113111222
I-5	(1,0,0,0)	5.077	1212312121212	4.615*	1212123121212	4.615*	1221123122121
I-6	(0,1,0,0)	19.380*	2121213121212	19.380*	2121213121212	19.380*	2121213121212

*가장 좋은 해

문제의 크기가 작은 첫 번째 예제에서는 가중치에 상관없이 Tabu Search기법이 MS_H1, MS_H2방법보다 같거나 좋은 해를 찾아내어, Tabu Search가 다른 방법보다 더 좋은 성능을 나타냄을 알 수 있었다. 이때 각 방법의 평균 계산시간은 각각 0.002초, 0.024초, 0.223초가 소요되었다.

4.2 예제 2

두 번째 예제에서는 문제의 크기를 확장하여 각 기법에 대한 해의 효율과 계산시간을 비교분석하였다. <표 3>은 모델의 수와 한 사이클당 생산되는 모델별 생산제품 수에 따라 분류한 9개의 문제를 나타내고 있다. <표 3>에서 문제 II-2의 경우에는 5개의 모델을 생산하며, 한 사이클당 생산되는 모델별 생산제품 수는 각각 10, 5, 3, 1, 1개로, 한 사이클당 생산되는 총 제품 수는 20개가 된다.

<표 3> Tabu Search의 성능평가를 위한 문제의 설계

문제	생산모델															총수요
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
II-1	16	1	1	1	1											20
II-2	10	5	3	1	1											20
II-3	5	4	4	4	3											20
II-4	21	1	1	1	1	1	1	1	1	1						30
II-5	7	6	5	3	3	2	1	1	1	1						30
II-6	4	3	3	3	3	3	3	3	3	2						30
II-7	26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	40
II-8	9	6	5	4	3	2	2	2	1	1	1	1	1	1	1	40
II-9	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	40

<표 5> 각 기법들의 탐색성능 비교분석

가중치	문 제	MS_H1	MS_H2	Tabu Search			MS_H2 (Mean)	IR
				Best	Mean	Worst		
	II-1	207.934	207.934	161.205*	161.205*	161.205*	161.205*	28.987
	II-2	188.107	161.149	160.933*	160.933*	160.933*	160.933*	0.134
	II-3	183.331	180.922	148.778*	148.778*	148.778*	148.778*	21.605
	II-4	1140.365	1008.236	920.870*	965.789	1004.269	944.532	9.487
(1, 1, 1, 1)	II-5	1219.044	999.720	964.300*	1037.700	1078.039	999.720	3.673
	II-6	1643.501	1382.668	1088.917*	1133.401	1177.421	1314.595	26.976
	II-7	7304.571	6822.721	5483.271*	5745.936	5912.869	6631.736	24.428
	II-8	8593.404	5796.565	5566.442*	5952.324	6180.931	5796.565	4.134
	II-9	8453.959	6266.656	6044.261*	6496.030	6709.215	6154.613	3.679
	II-1	192.934	192.934	145.705*	145.705*	145.705*	145.705*	32.414
	II-2	172.607	145.363*	145.363*	145.363*	145.363*	145.363*	0.000
	II-3	197.894	166.040	135.028*	135.028*	135.028*	135.028*	22.967
	II-4	1077.532	940.789	861.269*	919.055	958.081	877.038	9.233
(0, 1, 1, 1)	II-5	1339.701	933.298	913.944*	955.599	995.991	931.760	2.118
	II-6	1523.790	1220.459	1068.943*	1098.549	1119.079	1171.018	14.174
	II-7	7135.196	7441.983	5265.103*	5558.928	5929.955	6476.111	35.519
	II-8	8395.979	5613.925	5539.109*	5922.759	6200.671	5613.925	1.351
	II-9	8126.084	6703.086	5821.760*	6096.799	6336.081	6521.567	15.138
	II-1	141.807	141.807	124.017*	124.017*	124.017*	124.017*	14.345
	II-2	187.834	139.081	113.377*	113.377*	113.377*	113.377*	22.671
	II-3	180.059	130.413	107.107*	107.142	107.284	107.107*	21.760
	II-4	966.423	841.437	777.325*	818.386	874.430	795.323	8.248
(0, 0, 1, 1)	II-5	1148.216	868.961	859.136*	891.694	922.302	841.247	1.144
	II-6	1262.327	1032.967	945.005*	978.616	1031.710	948.472	9.308
	II-7	6577.894	6963.029	5119.484*	5388.710	5700.537	6068.347	28.487
	II-8	7404.955	5212.886	5371.075	5598.141	5805.377	5124.379*	-2.945
	II-9	7330.500	6868.005	5613.611*	5970.467	6273.391	6733.307	22.346
	II-1	110.305	110.305	97.915*	97.915*	97.915*	97.915*	12.654
	II-2	128.986	128.631	88.512*	88.512*	88.512*	95.658	45.326
	II-3	115.846	113.015	83.789*	84.179	84.563	83.789	34.880
	II-4	787.501	601.283	559.243*	580.601	598.327	567.750	7.517
(0, 0, 0, 1)	II-5	902.665	794.111	606.980*	643.562	664.803	662.024	30.830
	II-6	804.652	727.248	677.524*	698.515	711.317	717.210	7.339
	II-7	6359.024	5798.445	4283.840*	4464.511	4711.100	4488.700	35.356
	II-8	6785.713	5356.486	4468.867*	4630.325	4809.196	5160.792	19.862
	II-9	6457.931	4993.135	4604.653*	4992.739	5194.539	4945.778	8.437
	II-1	13.500*	13.500*	13.500*	13.500*	13.500*	13.500*	0.000
	II-2	10.000*	10.000*	10.000*	10.000*	10.000*	10.000*	0.000
	II-3	11.350*	11.350*	11.350*	11.350*	11.350*	11.350*	0.000
	II-4	45.500*	45.500*	45.500*	45.500*	45.500*	45.500*	0.000
(1, 0, 0, 0)	II-5	31.289	31.156	30.889*	30.889*	30.889*	30.889*	0.864
	II-6	40.844*	40.844*	40.844*	40.844*	40.844*	40.844*	0.000
	II-7	93.375*	93.375*	93.375*	93.375*	93.375*	93.375*	0.000
	II-8	62.075*	62.075*	62.075*	62.075*	62.075*	62.075*	0.000
	II-9	76.625*	76.625*	76.625*	76.625*	76.625*	76.625*	0.000
	II-1	17.793	17.793	17.520*	17.520*	17.520*	17.520*	1.558
	II-2	20.358	19.358*	19.358*	19.675	20.151	19.358*	0.000
	II-3	23.985	21.956	17.601*	17.774	17.889	21.898	24.743
	II-4	72.567	70.326	62.515*	62.515*	62.515*	70.004	12.495
(0, 1, 0, 0)	II-5	83.471	83.139	63.581*	65.814	67.447	68.819	30.761
	II-6	103.652	85.463	65.245*	66.991	68.286	80.205	30.988
	II-7	185.392	179.820	177.605*	187.250	197.453	178.763	1.247
	II-8	200.431	196.036	191.648*	195.267	203.500	194.664	2.290
	II-9	234.076	210.913	188.831*	200.661	207.464	200.888	11.694

Tabu Search의 결과는 30회 실험한 결과의 가장 좋은 값(Best), 평균값(Mean), 가장 나쁜 값(Worst)을 나타내며, Tabu Search(MS_H2)는 문제 II-1, 2, 3의 경우 100개, II-4, 5, 6의 경우 200개, II-7, 8, 9는 300개의 이웃해를 탐색하였으며, 30회 실험한 결과의 평균치를 나타내었다. 또한 IR(Improved rate)은 $\{MS_H1 \text{ 또는 } MS_H2 \text{ 중 좋은 해} - Tabu Search(Best)\} / Tabu Search(Best) \times 100$ 으로 계산되었다.

9개의 문제에 대해 각각 가중치를 변화시키면서 MS_H1, MS_H2 방법과 Tabu Search에서 30회 실험결과의 평균값을 나타낸 Tabu Search(Mean)와 MS_H2방법을 비교한 결과, Tabu Search(Mean)가 총 54개의 문제 중 45개 문제에서 같거나 더 좋은 결과를 나타내었다. 또한 30회 실험결과에서 가장 나쁜 값을 나타낸 Tabu Search(Worst)와 MS_H2방법을 비교한 결과, Tabu Search(Worst)가 총 54개의 문제 중 41개 문제에서 같거나 더 좋은 결과를 나타내어 해의 질 면에서는 Tabu Search 방법이 상당히 좋음을 알 수 있다.

문제의 크기가 작은 II-1, 2, 3의 경우 Tabu Search 방법은 거의 최적해를 찾아감을 알 수 있으며, 제품만의 실제 생산량과 수요량의 편차를 최소로 하는 투입순서를 구하는 형태인 가중치 (1, 0, 0, 0)의 경우 문제가 단순해짐으로써 목적함수 값의 많은 중복을 가져와 최적해와 같은 해가 다수 발생하여 모든 방법이 최적해를 쉽게 찾을 수 있었다.

<표 6>은 각 방법에 대한 계산시간의 비교분석을 나타낸다. Tabu Search 방법이 MS_H1이나 MS_H2방법보다 계산시간이 상당히 많이 걸림을 알 수 있다. Tabu Search의 계산시간을 줄이기 위해 MS_H2방법으로 구한 해를 Tabu Search의 초기해로 사용한 Tabu Search(MS_H2)는 MS_H2방법과 계산시간이 큰 차이가 없음을 알 수 있다. 또한 Tabu Search(MS_H2)를 해의 효율측면에서 살펴볼 때, 총 54개의 문제 중 41개 문제에서 MS_H2방법으로 구한 해를 개선시키었으며, 35개 문제에서 Tabu Search(Mean)와 같거나 더 좋은 결과를 나타내어 상당히 좋은 해를 찾을 수 있었다.

<표 6> 계산시간의 비교분석

문제	MS_H1	MS_H2	Tabu Search	Tabu Search(MS_H2)
II-1, II-2, II-3	0.01	0.03	3.81	0.08
II-4, II-5, II-6	0.03	0.25	34.03	0.94
II-7, II-8, II-9	0.08	1.13	149.52	3.15

5. 결론

본 연구에서는 자동차 조립라인의 투입순서에 관한 여러 형태의 목적 중에서 Just-In-Time 시스템에서 특히 중요시하는 목적인 부품 사용의 일정율을 유지하기 위한 목적을 다루었다. 다루는 목적에 대해 Tabu Search기법을 적용하여 실시간에 효율

적인 투입순서를 구할 수 있는 방법을 제시하고, 기존에 제시된 발견적 기법과 해의 효율과 계산 시간의 측면에서 비교 분석하였다.

다양한 문제를 통해 해의 효율과 계산 시간의 측면에서 각 기법들을 비교 분석한 결과, 예제에 따라 정도의 차이는 있으나 본 연구에서 제시한 Tabu Search 기법이 해의 효율면에서 뛰어났으며, 계산시간은 더 많이 걸린 것으로 나타났다. 계산시간을 감소시키기 위해 탐색적 방법으로 구한 해를 Tabu Search의 초기해로 사용한 결과, 빠른 시간에 상당히 좋은 해를 탐색할 수 있었다.

본 연구에서 제시한 Tabu Search는 Just-In-Time 시스템에서 현장의 상황 변화에 따라 투입순서를 재결정하고자 할 때 다루는 목적에 대해 최적에 가까운 투입순서를 실시간에 구할 수 있다.

6. 참고문헌

- [1] 현철주, 김여근, "혼합모델 조립라인의 다목적 투입순서문제를 위한 유전알고리즘", 대한산업공학회지, 22권 4호, pp. 533-549, 1996.
- [2] C. J. Hyun, Y. K. Kim and Y. Kim, "A Genetic Algorithm for Multiple Objective Sequencing Problems in Mixed Model Assembly Lines", Computers & Ops Res, 25(7-8), pp.675-690, 1998.
- [3] E. M. Dar-El, "Mixed-model assembly line sequencing problems", Omega, Vol. 6, pp. 313-323, 1978.
- [4] F. Glover, "Tabu search-part I", ORSA Journal on Computing, Vol. 1, pp. 190-206, 1989.
- [5] F. Kolahan and M. Liang, "An adaptive TS approach to JIT sequencing with variable processing times and sequence-dependent setups", Eur. J. Operational Research, pp. 142-159, 1998.
- [6] J. F. Bard, A. Shtub and S. B. Joshi, "Sequencing mixed-model assembly lines to level parts usage and minimize line length", Int. J. Prod. Res., Vol. 32, pp. 2431-2454, 1994.
- [7] J. Miltenburg, "Level schedules for mixed-model assembly lines in just-in-time production systems", Mgmt. Sci., Vol. 35, pp. 192-207, 1989.
- [8] J. Miltenburg and G. Sinnamon, "Scheduling mixed-model multi-level just-in-time production systems", Int. J. Prod. Res., Vol. 27, pp. 1487-1509, 1989.
- [9] K. Okamura and H. Yamashina, "A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor", Int. J. Prod. Res., Vol. 17, pp. 233-247, 1979.

- [10] K. Takashi, K. Hiroaki and N. Masakazu, "A hybrid Search for genetic algorithms : Combining genetic algorithms, TABU Search and Simulated annealing", Proceedings of the fifth International Conference on Genetic Algorithms, pp. 641, 1993.
- [11] M. Kilbridge and L. Wester, "The assembly line model-mix sequencing problem", Proc. of the Third Int. Conf. Op. Res., Oslo, Paris: English Universities Press, pp. 247-260, 1963.
- [12] M. Laguna, J. W. Barnes and F. Glover, "Tabu search methods for a single machine scheduling problem", Journal of Intelligent Manufacturing, Vol. 2, pp. 63-74, 1991.
- [13] P. R. McMullen, "An efficient frontier approach to addressing JIT sequencing problems with setups via search heuristics", Computers & IE, pp. 335-353, 2001.
- [14] R. T. Sumichrast, R. S. Russell and B. W. Taylor III, "A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system", Int. J. Prod. Res. Vol. 30, pp. 199-214, 1992.
- [15] S. J. Beaty, "Genetic algorithm versus Tabu search for instruction scheduling", Proceedings of the fifth International Conference on Genetic Algorithms, pp. 496-501, 1993.
- [16] T. Tamura, H. Long and K. Ohno, "A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline", International Journal Of Production Economics, pp. 557-564, 1999.
- [17] Y. K. Kim, C. J. Hyun and Y. Kim, "Sequencing in Mixed Model Assembly Lines : A Genetic Algorithm Approach", Computers & Ops Res., Vol.23, No. 12, pp. 1131-1145, 1996.
- [18] Y. Monden, Toyota Production System, Institute of Industrial Engineers Press, Norcross, GA, 1983.

저 자 소 개

현 철 주 : 전남대학교에서 학사, 석사, 박사를 취득.

현재 정인대학 품질관리과 조교수로 재직중.

주요 관심분야는 최적화문제에 Genetic Algorithm 과 Tabu Search의 응용 및 관련기법 개발 등.