

준비시간이 있는 혼합모델 조립라인의 제품투입순서 결정 : Tabu Search 기법 적용[†]

김여근* · 현철주**

Sequencing in Mixed Model Assembly Lines with Setup Time :
A Tabu Search Approach[†]

Yeo Keun Kim* · Chul Ju Hyun**

ABSTRACT

This paper considers the sequencing problem in mixed model assembly lines with hybrid workstation types and sequence-dependent setup times. Computation time is often a critical factor in choosing a method of determining the sequence. We develop a mathematical formulation of the problem to minimize the overall length of a line, and present a tabu search technique which can provide a near optimal solution in real time. The proposed technique is compared with a genetic algorithm and a branch-and-bound method. Experimental results are reported to demonstrate the efficiency of the technique.

1. 서 론

혼합모델 조립라인은 공장부지, 시설 등의 투자를 줄이고 시설의 효율적인 이용을 위하여 한 라인에서 유사한 여러 모델의 제품을 생산하는 조립라인이다. 이러한 라인의 대표적인 예는 자동차 조립라인(승용차 라인, 중형차 라인, 트럭 라인 등)으로 각 라인에는 생산제품의 특성과 크기 등

이 유사한 여러 종류의 제품이 생산되고 있다[1].

혼합모델 조립라인에서는 재고비용을 줄이고 수요를 적시에 만족시키기 위하여 모델의 종류와 양을 평균화하여 끌고루 혼류로 생산하는 평균화 생산방식을 택하는 경우가 많다. 평균화 생산방식은 모델별로 작업내용, 작업방법 및 작업시간 등이 상이할 뿐 아니라 모델의 투입순서에 따라 준비시간이 변화하여 생산계획과 통제가 복잡하게

† 이 논문은 1995년도 전남대학교 자동차 연구소의 공과과제 연구비에 의하여 연구되었음.

* 전남대학교 공과대학 산업공학과

** 정읍공업전문대학 품질관리과

된다. 이러한 복잡성 때문에 흔히 작업자를 과잉 투입하거나, 작업장의 길이를 필요이상으로 길게 하는 경향이 있다. 라인의 효율적인 이용을 위해서는 모델의 투입순서(Sequencing)를 결정하는 문제가 중요하게 다루어 져야 한다.

혼합모델 조립라인의 투입순서에 관한 연구는 Kilbridge와 Wester[14]에 의해 1963년 처음으로 발표된 이후, 라인길이의 최소화[2, 3, 8], 컨베이어 정지위험의 최소화[20], 총 가외작업(Utility work)의 최소화[23], 부품사용의 일정을 유지 [17, 18] 등 다양한 목적들에 대해 여러 연구가 이루어 졌다. 라인길이를 최소화 하는 투입순서는 라인길이가 주어졌을 때 컨베이어의 정지위험과 라인에서 발생하는 가외작업을 줄일 수 있을 뿐 아니라 작업장 및 설비의 효율적인 배치수립을 가능하게 한다는 장점이 있다. 본 연구에서는 투입순서를 결정하는데 있어서 라인길이를 최소화 하는 목적을 사용한다.

조립라인에서 작업장은 흔히 폐쇄작업장과 개방작업장으로 분류된다[8]. 폐쇄작업장은 설비나 장비의 제약으로 인하여 주어진 작업장의 경계내에서 작업을 해야 하는 작업장이고, 개방작업장은 인접작업자와의 작업간섭이 되지 않는 범위내에서 앞뒤작업장의 공유가 가능한 작업장이다. 대부분의 혼합모델 조립라인은 폐쇄작업장과 개방작업장이 혼합되어 있는 혼성(Hybrid) 조립라인으로 구성되어 있다[10]. 한편 각 작업장은 선후행간의 모델의 변화에 따른 준비작업이 흔히 요구된다[6, 7]. 예를 들어 자동차 조립라인에서 차체(Body)작업장은 선후행간의 문(Door)형태에 따라, 엔진작업장은 선후행모델과의 엔진형태 등에 따라 각각 준비시간이 다르게 된다. 흔히 준비시간은 모델변화에 종속되므로 조립시간에 포함시킬 수가 없게 된다. 본 연구에서는 혼성 조립라인에서 모델의 변화에 따른 준비시간을 고려하고자

한다.

혼합모델 조립라인의 투입순서문제는 조합최적화문제로 NP-hard 이다. 수요변화, 결품으로 인한 투입순서의 재결정에는 실시간 처리가 요청된다. 본 연구에서는 폐쇄 작업장과 개방 작업장이 공존하는 혼성조립라인에서 모델변화에 따른 준비시간을 고려하여, 전체 라인길이를 최소화 하는 투입순서를 구하는 수리모형을 개발하고, 실시간에 효율적인 해를 구할 수 있는 Tabu Search 기법을 제시하고자 한다.

Tabu Search는 인간의 기억과정을 이용한 탐색기법으로 해의 탐색과정을 일정기간 기억함으로써 해의 순환을 막고, 이웃해 중 가장 좋은 해로 이동함으로써 부분최적에서 벗어나 최적해 또는 근사최적해를 구할 수 있는 기법이다. Tabu Search는 Glover [11]에 의해 제안되어 여러 분야에서 활발히 적용되어 왔으며, 특히 외판원문제(Traveling salesman problem)와 Flow-shop 순서문제 등의 조합최적화 문제에 적합한 기법으로 알려져 있다[12, 22]. 본 연구에서는 Tabu Search를 적용하기 위하여 혼합모델 조립라인의 투입순서문제에 적합한 이웃해의 생성방법과 확률적 이동방법을 다룬다. 그리고 제안된 기법과 유전알고리즘, 그리고 분지한계법(Branch and bound method)을 계산시간 및 해의 효율 측면에서 비교분석하고자 한다.

본 연구의 구성은 아래와 같다. 제 2장에서 준비시간을 고려하여 전체 라인의 길이를 최소화 하는 수리모형을 제시하고, 제 3 장에서는 Tabu Search의 절차, 이웃해의 생성방법과 확률적 이동방법을 다룬다. 그리고 제 4 장에서는 실험을 통하여 이동방법과 확률적 이동방법을 분석하며, Tabu Search, 유전알고리즘, 그리고 분지한계법을 비교 분석한다. 제 5 장은 결론으로 구성되어 있다.

2. 모형화

2.1 문제의 설정

본 연구에서 다루는 혼합모델 조립라인은 일정 한 속도(v_c)로 이동하는 컨베이어 시스템으로 유사한 여러모델의 제품들이 일정시간 간격으로 투입되고 있으며, 컨베이어의 이동에 따라 작업자가 이동하면서 작업하는 라인이다. 라인은 J 개의 혼성작업장으로 구성되어 있으며, 각 작업자는 인접 작업자와의 간섭이 발생되어서는 안된다고 본다. 작업자는 컨베이어 상의 조립작업을 마친 후, 컨베이어 밖에서 다음 제품의 준비작업을 시작한다.

혼합모델 조립라인의 설계시 고려하는 요소로는 작업자의 작업계획(Operator schedule), 사이클당 생산제품의 수, 발주간격(Launch interval) 등이 있다[2]. 첫째, 작업자의 작업을 계획하는 방법에는 늦은작업계획(Late start schedule)과 빠른작업계획(Early start schedule)이 있다[10]. 늦은작업계획은 작업자의 유휴시간없이 작업을 계획함으로써 빠른작업계획에 비해 상대적으로 라인길이가 길 수 있으며, 반면에 빠른작업계획은 작업자의 유휴시간을 허용함으로써 생산시간이 손실될 수 있다[2]. 본 연구에서는 빠른작업계획을 사용한다. 둘째, 혼합모델 조립라인에서 계획기간(Planning horizon)동안 생산될 모델의 개수를 각각 D_1, \dots, D_M 이라고 하고, D_1, \dots, D_M 의 최대공약수를 h 라 하자. 각 모델의 수요 ($D_1/h, \dots, D_M/h$)= (d_1, \dots, d_M) 를 최소부품집합(Minimum Part Set : MPS)이라고 한다[2]. 혼합모델 조립라인에서는 MPS를 h 번 반복 생산함으로써 계획기간의 수요를 만족시키는 사이클 생산을 흔히 사용하고 있다. 셋째, 발주간격(γ)은 $\gamma_1 + \gamma_2$ 로 둔다. 여기서 γ_1 은 각 작업장당 각 모델의 평균 조

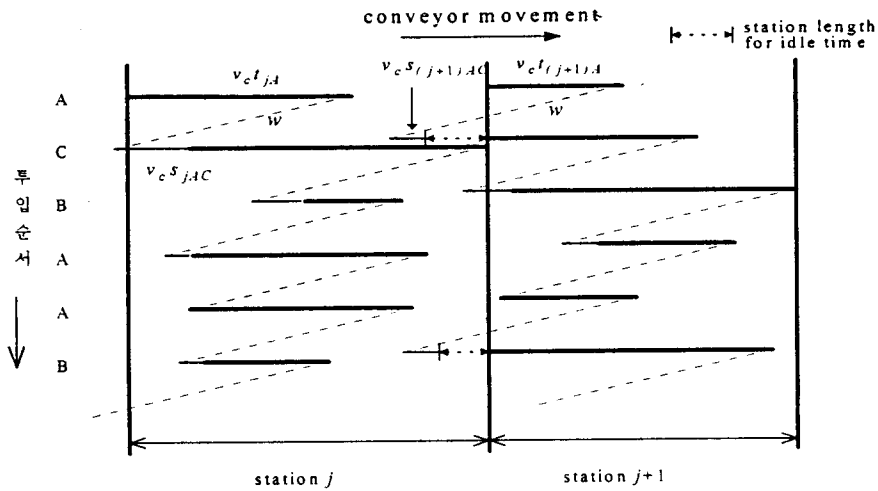
립작업시간을 나타내며, 이는 1회의 MPS를 생산하는데 요구되는 총 조립시간(T)을 제품의 총수와 작업장의 수로 나눈 값이 된다. 또한 γ_2 는 각 작업장당 모델변화에 따른 평균 준비시간을 나타내며, 여기서는 총 준비시간(S)을 모델변화의 모든 경우의 수, $iP_i (= I \times (I - 1))$, 와 작업장의 수 J 로 나눈 값으로 둔다.

혼합모델 조립라인의 모델의 투입순서결정을 위해서는 먼저 사이클타임의 결정, 라인밸런싱 등의 절차가 필요하다[19]. 본 연구에서는 사이클타임이 결정되고, 각 작업장은 라인밸런싱되었다고 본다. 또한 다음 제품의 작업을 위한 작업자의 이동거리(w)는 $v_c \gamma [v_o / (v_o + v_c)]$ 이 된다. 작업자의 이동속도(v_o)가 컨베이어 이동속도(v_c)보다 상당히 빠르다고 보고, $w = v_c \gamma$ 라 둔다.

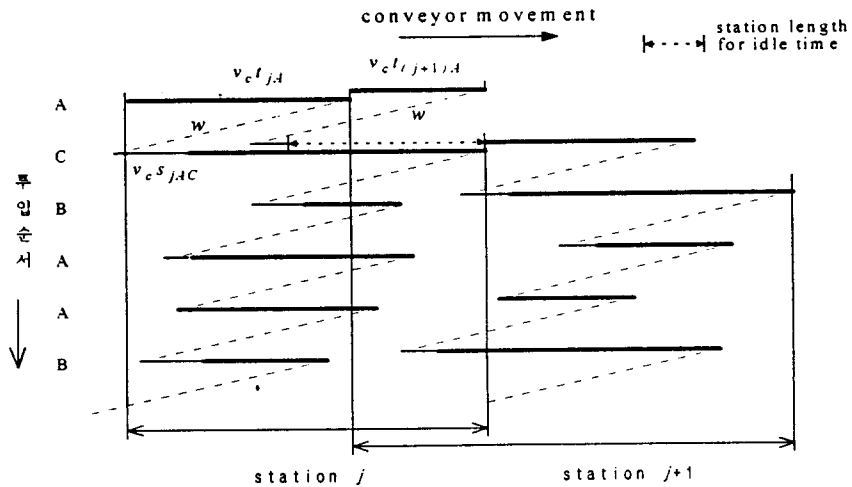
[그림 1]과 [그림 2]는 다루는 모형에서 폐쇄 작업장과 개방작업장의 길이를 각각 보여주고 있다. A, C, B, A, A, B 모델순으로 투입하는 경우 굵은 실선은 각 조립작업에 필요한 작업장길이를 나타내며, 가는 실선은 다음 제품의 준비시간에 해당하는 작업장길이를 나타낸다. 또한 점선은 제품간의 간격을 나타내고, 화살표로 표시된 점선은 작업자의 유휴시간에 해당하는 작업장 길이를 나타낸다. [그림 1]과 같이 두 작업장사이에 경계가 존재하는 경우, 작업자가 준비작업을 마친 시점에 다음 제품이 작업장 경계내에 도달해 있다면 조립작업을 실시하고($j+1$)작업장에서 세 번째제품의 모델 B), 그렇지 않다면 작업자는 제품이 작업장에 도달할 때까지 유휴하게 된다 ($j+1$)작업장에서 두번째 제품의 모델C). 또한 [그림 2]와 같이 두 작업장사이에 경계가 존재하지 않는 경우, 작업자가 준비작업을 마친 시점에 선행작업자가 조립작업을 끝냈다면 작업자는 즉시 조립작업을 실시하고($j+1$)작업장에서 세 번

제품의 모델B), 그렇지 않다면 선행작업자가
 조립작업을 마칠때까지 작업자는 유휴하게 된다
 ((j+1)작업장에서 두번째 제품의 모델 C).

수리모형화를 위해 사용된 기호를 정의하면 다
 음과 같다.



[그림1] 폐쇄작업장의 길이



[그림2] 개방작업장의 길이

첨자

i = 투입되는 제품의 위치(투입순서)첨자 ;

$$i = 1, 2, \dots, I.$$

j = 작업장 첨자 ; $j = 1, 2, \dots, J.$

m, r, p = 모델 첨자 ; m, r, p
 $= 1, 2, \dots, M.$

입력 자료

v_c = 컨베이어의 속도.

v_o = 작업자의 이동속도.

d_m = 모델 m 의 수요량.

t_{jm} = 작업장 j 에서 모델 m 의 조립시간.

s_{jmr} = 작업장 j 에서 모델 m 에서 모델 r 로의 변경에 따른 준비시간.

$u_j = 1$, 작업장 j 와 $j+1$ 이 개방작업장일때,
 $= 0$, 그외의 경우.

I = 투입순서를 정하는 제품의 수 ; I
 $= \sum_{m=1}^M d_m.$

J = 작업장의 수.

M = 모델의 수.

T = 총 조립시간 ; $T = \sum_{j=1}^J \sum_{m=1}^M t_{jm} d_m.$

S = 총 준비시간 ; $S = \sum_{j=1}^J \sum_{m=1}^M \sum_{r=1}^M s_{jmr} d_m d_r.$

γ = 투입시간 간격 ; $\gamma = \gamma_1 I + \gamma_2 \frac{m^2 r}{r}$,
 여기서 $\gamma_1 = T/(I \times J)$, $\gamma_2 = S/(I \times (I-1) \times J).$

w = 다음 제품의 작업을 위한 작업자의 이동거리 ; $w = v_c \gamma [v_o / (v_o + v_c)]$, 여기서 $v_o \gg v_c$ 이면 $w = v_c \gamma.$

결정변수

$X_{imr} = 1$, 투입순서에서 모델 m 이 i 번째에 위치하고, 모델 r 이 $(i+1)$ 번째에 위치한 경우,
 $= 0$, 그외의 경우.

Y_j = 작업장 j 의 길이.

Z_{ij} = 작업장 j 에서 i 번째 제품의 작업시작위치.

O_j = 작업장 j 와 $j+1$ 사이에 중첩되는 길이.

2.2 수리모형

혼합모델 조립라인의 투입순서문제는 Kilbridge와 Wester[14]에 의해 처음으로 제시된 이후 많은 수리모형과 발견적 기법이 개발되었다. Dar-El과 Cother[8]는 전체 라인길이를 최소화 하는 투입순서를 결정하는 발견적기법을 제시하고, 라인길이에 영향을 주는 요인들(모델의 수, 각 모델간의 수요편차, 작업장의 수 등)에 대해 분석하였다. Bard *et al.* [2]는 투입규칙(Launch discipline), 작업장 제약 등의 여러 요인을 고려하여 조립라인의 최소화와 완성시간(Throughput time)의 최소화를 목적으로 하는 수리 모형을 각각 제시하였다. 또한 Bard *et al.* [3]은 라인길이의 최소화과 부품사용의 일정율을 유지하는 투입순서를 위한 수리모형과 발견적 기법을 제시하였다. 이들은 모든 작업장을 폐쇄작업장이거나 개방작업장으로 가정하였으며 모델 변화에 따른 준비시간을 고려하지 않았다. 준비시간 또는 비용을 최소화 하기 위해 Burns와 Daganzo[7]는 모델변화에 따른 준비비용과 생산능력을 고려하여 투입순서를 결정하는 방법을 제시하였으며, Bolat *et al.* [6]은 준비비용과 가외작업비용(Utility work cost)을 최소화 하는 수리모형과 발견적기법을 제시하였다.

본 연구에서는 폐쇄 작업장과 개방 작업장이 혼합되어 있는 혼성조립라인, 고정발주, 빠른 작업계획의 상황에서 MPS의 수요에 대해 준비시간을 고려하여 전체 라인길이를 최소화 하는 투입순서를 결정하기 위한 수리모형을 개발 한다. 다루는 문제는 아래와 같이 혼합정수계획법으로 모형화 된다.

$$\text{Min. } \sum_{j=1}^J Y_j - \sum_{j=1}^{J-1} u_j O_j \quad (0)$$

S.T.

$$\sum_{m=1}^M \sum_{r=1}^M X_{imr} = 1 \quad i = 1, \dots, I \quad (1)$$

$$\sum_{m=1}^M X_{imr} = \sum_{p=1}^M X_{(i+D)p} \quad r = 1, \dots, M, \\ i = 1, \dots, I-1 \quad (2)$$

$$\sum_{i=1}^I \sum_{m=1}^M X_{imr} = d_r \quad r = 1, \dots, M \quad (3)$$

$$Z_{i+1,j} \geq Z_{ij} + v_c \sum_{m=1}^M \sum_{r=1}^M X_{imr} (t_{jm} + s_{jmr}) \\ - w \quad j = 1, \dots, J, \quad i = 1, \dots, I-1 \quad (4)$$

$$Y_j \geq Z_{ij} + v_c \sum_{m=1}^M \sum_{r=1}^M X_{imr} t_{jm} \\ j = 1, \dots, J, \quad i = 1, \dots, I \quad (5)$$

$$O_j \leq Y_j - (Z_{ij} + v_c \sum_{m=1}^M \sum_{r=1}^M X_{imr} t_{jm}) \\ + Z_{i,j+1} \quad j = 1, \dots, J-1, \quad i = 1, \dots, I \quad (6)$$

$$X_{imr} = 0 \text{ or } 1 \quad m, r = 1, \dots, M, \quad i = 1, \dots, I \quad (7)$$

$$Z_{ij} = 0, \quad Z_{ij} \geq 0, \quad Y_j \geq 0 \quad j = 1, \dots, J, \\ i = 1, \dots, I \quad (8)$$

$$O_j \geq 0 \quad j = 1, \dots, J-1 \quad (9)$$

목적함수(Objective function)

혼성 조립라인의 전체 길이를 최소로 하는 투입순서를 구하는 것을 목적으로 한다. 목적함수의 첫번째 항은 작업장의 총 길이를 나타내며, 두번째 항은 서로 인접한 개방작업장의 중첩된 작업장 길이의 합을 나타낸다.

제약식(Constraints)

제약식(1)은 투입순서의 각 위치에는 단 하나의 제품만이 있어야 한다는 위치제약이고, 제약식(2)는 투입된 모델은 라인상에서 순서가 변할 수 없음을 나타낸다. 제약식(3)은 MPS의 수요제약

이다. 제약식(4)는 작업장 j 에서 $(i+1)$ 번째 제품의 작업시작 위치($Z_{i+1,j}$)에 관한 제약이다. 제약식의 우변 첫번째 항은 작업장 j 의 i 번째 제품의 작업 시작위치를 나타내고, 두번째 항은 작업자가 i 번째 제품의 조립작업을 시작하여 다음 제품의 준비작업을 끝마칠때까지의 시간을 거리로 나타낸 것이다. 여기서 w 는 다음 제품의 작업을 위한 작업자의 이동거리이다. 제약식(5)는 각 작업장의 길이가 조립되는 모든 제품의 작업 종료위치이상 되어야 함을 나타낸다. 제약식(6)은 인접한 작업장에서 작업자의 간섭이 되지 않고 중첩될 수 있는 작업장의 길이를 나타내는 제약으로, 라인상의 모든 작업장이 폐쇄작업장이거나 서로 인접한 개방작업장이 없는 경우에는 제거할 수 있다. 제약식(5)와 (6)에서 준비시간이 포함되지 않은 것은 준비작업이 컨베이어 밖에서 행하여지기 때문이다. 제약식(7)은 0 과 1의 결정변수를 나타내며, 제약식(8)에서 $Z_{ij} = 0, \quad j = 1, \dots, J$ 은 각 작업장의 시작위치가 0이라는 것을 의미하고, 다른 제약들은 비음제약이다.

준비시간이 무시되고 ($s_{jmr} = 0$, 모든 j, m, r), 작업장이 모두 폐쇄작업장이면 이 모형은 Bard *et al.* [2]의 모형 I이 되고, 작업장이 모두 개방작업장이면 Bard *et al.* [2]의 모형 II와 같게 된다.

3. Tabu Search

본 장에서는 준비시간을 고려하여 전체 라인의 길이를 최소로 하는 투입순서를 구하기 위한 Tabu Search를 제시한다. 이를 위해 먼저 Tabu Search의 절차에 대하여 설명하고, 알고리즘의 성능에 큰 영향을 미치는 이웃해 생성방법과 해의 다양화를 위한 확률적 이동방법을 제안하고자 한다.

3.1 Tabu Search의 절차

Tabu Search(TS)는 부부최적에서 벗어나 최적해 또는 근사최적해를 구할 수 있는 탐색적 방법으로 Glover[12, 13]에 의해 연구되기 시작하여, 유전알고리즘, 신경망이론과 함께 일정계획, 고용계획, 공간계획과 구조설계, 통신경로문제 등의 조합최적화문제에 널리 적용되고 있다. Belarmino[5]는 흐름생산문제에 Tabu Search를 적용하여 이웃해의 집합을 줄이는 방법을 제시하고 있으며, Takashi *et al.* [21]는 외판원문제, Beaty[4]는 일정계획 문제를 대상으로 하여 Tabu Search를 타기법과 비교분석하고 있다. Tabu Search에 관한 이론연구로는 이동의 유형과 그 적용방법, 순환을 막는 방법, Tabu List 관리방법 등에 관한 연구가 이루어 졌다[13, 15, 22]. Tabu Search의 절차를 흐름도로 나타내면 그림 3과 같다[12]. 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[15]. 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 최대 수를 나타낸다.

열망수준은 어떤 이동이 Tabu상태이지만 이를 해제하여 해의 이동을 가능케하는 기준으로 본 연구에서는 열망수준을 현재까지 탐색한 가장 좋은해, 즉 최선해로 두었다. 또한 종료조건으로는 반복수, 탐색한 이웃해의 수, 최선해를 개선하지 못한 회수 등이 있으나, 본 연구에서는 현재까지

탐색한 이웃해의 개수를 사용하여 알고리즘을 종료하기로 한다.

3.2 이웃해의 생성

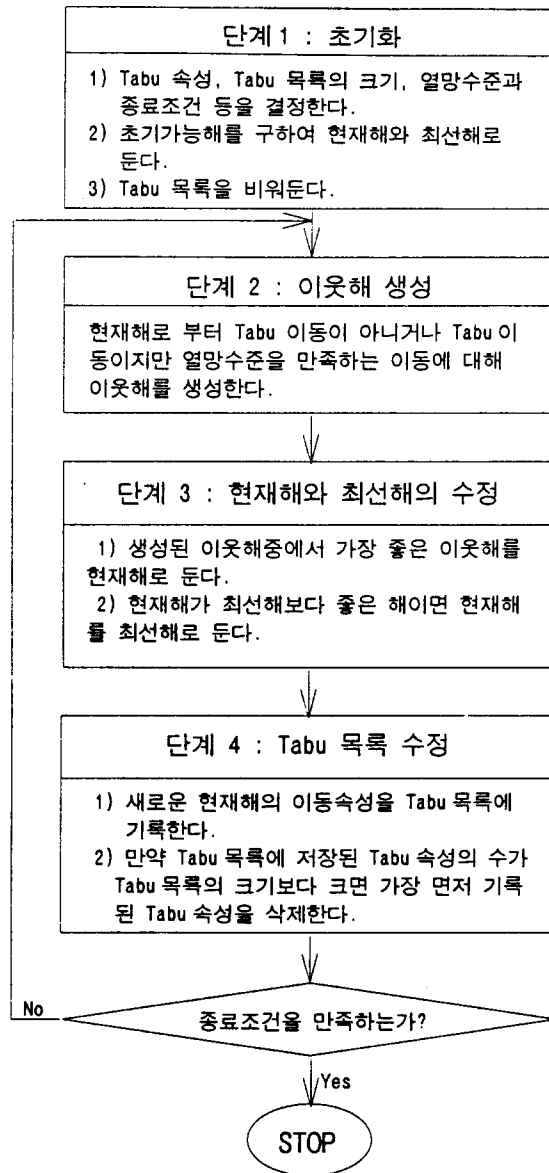
이웃해의 생성방법은 문제의 특성을 잘 반영하여 좋은 해를 효율적으로 탐색해 갈 수 있도록 개발되어야 한다. 순서문제에서는 흔히 두개의 원소의 위치를 서로 바꾸는 교환이동과 하나의 원소를 다른 위치로 이동시키는 삽입이동을 사용하고 있다[15, 22]. 교환 또는 삽입이동으로 생성된 이웃해를 현재해와 비교해 볼때, 교환이동은 위치 및 순서의 변화가 너무 작고, 삽입이동은 위치의 변화가 너무 커서 해의 탐색성능을 저하 시킬 수 있다. 본 연구에서는 교환, 삽입이동뿐 아니라 생성된 이웃해가 위치 및 순서의 변화를 적절히 가져 올 수 있는 역순, 집합이동을 제시하고 실험을 통해 비교분석한다.

3.2.1 교환 이동

교환이동은 서로 같지 않은 임의의 두 원소를 교환하여 이웃해를 생성하는 방법으로, 그림 4(a)와 같이 교환될 원소로 B원소와 C원소가 선택되었다면, 두 원소를 서로 교환하여 이웃해를 생성한다. 이때 Tabu 속성은 $(\Pi(i), \Pi(j), i, j) = (B, C, 4, 8)$ 가 된다.

3.2.2 삽입 이동

삽입이동은 한 원소를 다른 위치로 삽입하여 이웃해를 생성하는 방법이다. 예를 들어 그림 4(b)와 같이 삽입될 원소로 4번째 원소(B)가 선택되고 삽입될 위치가 (8th, 9th)의 원소사이로 선택되었다면 4번째 원소 B를 (8th, 9th)의 원소사



[그림3] Tabu Search의 절차

이로 삽입하여 이웃해를 생성한다. 이때 Tabu 속성은 $(\Pi(i), \Pi(j), i, j) = (B, C, 4, 8)$ 가 된다.

3.2.3 역순 이동

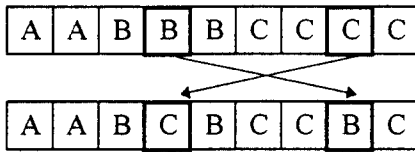
역순이동은 몇개의 연속된 원소를 선택하여, 선택된 원소들이 좌우대칭이 아니면 이 원소들을 역순으로하여 이웃해를 생성하는 방법이다. 예를 들어 [그림4] (c)와 같이 연속된 원소들이 선택되었다면, 이 원소들을 역순으로 하여 이웃해를

생성하는 방법이다. 이때 Tabu 속성은 $(\Pi(i), \Pi(j), i, j) = (B, C, 4, 7)$ 로 둔다.

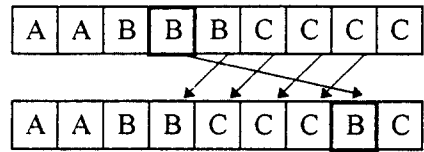
3.2.4 접합 이동

접합이동은 몇개의 연속된 원소를 선택하여,

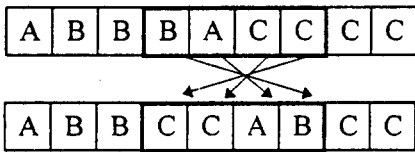
이 원소들을 마지막 위치에 접합하여 이웃해를 생성하는 방법이다. 예를 들어 [그림4] (d)와 같이 연속된 원소들이 선택되었다면, 이 원소들을 마지막 위치에 접합하여 이웃해를 생성한다. 이때 Tabu 속성은 $(\Pi(i), \Pi(j), i, j) = (B, C, 3, 6)$ 로 둔다.



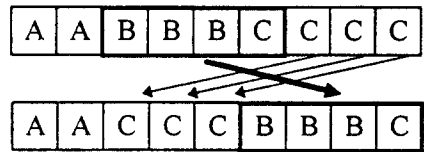
(a) 교환 이동



(b) 삽입 이동



(c) 역순이동



(d) 접합이동

[그림4] 이웃해 생성방법

3.3 확률적 이동방법

Tabu Search는 해 공간의 탐색 강화(Intensification)와 다양화(Diversification)를 적절히 조화시켜 나가는 것이 중요하다. 확률적 이동방법의 기본개념은 Simulated Annealing과 유전알고리즘에서 사용하는 확률적 개념을 도입하여 해 공간을 다양하게 탐색하고자 하는 것이다. 즉 이웃해로 이동하는데 있어서, 가장 좋은 이웃해로 이동하는 것이 아니라 이동확률을 사용하여 결정하게 된다. 확률적 이동방법은 해 공간을 다양하게 탐색할 뿐 아니라, Tabu Search에서 흔히 발생

하기 쉬운 해의 순환현상을 방지하는 효과를 가져온다.

Glover[12]는 이웃해로의 이동확률을 목적함수값에 비례하여 구하는 방법과 목적함수값의 순서를 이용하는 방법을 제안하였다. 본 연구에서 이동확률은 목적함수값이 가장 좋은 이웃해부터 차례로 나열하여 순위(Rank)를 1, 2, 3, ...으로 주어 아래와 같은 기하분포를 사용한다.

$$Prob(rank) = q(1-q)^{rank-1}, \quad rank = 1, 2, 3, \dots,$$

여기서 q 는 가장 좋은 이웃해로 이동할 확률이고 $0 < q \leq 1$ 의 값을 갖는다. q 가 1인 경우에는 이웃해 중 가장 좋은 해만으로 이동하므로 원래의 Tabu

Search기법과 같아지게 된다.

4. 실험결과

이 장에서는 실험을 통하여 이웃해의 생성방법과 확률적 이동방법을 분석하고, Tabu Search, 유전알고리즘, 그리고 분지한계법의 성능을 계산 시간과 해의 효율측면에서 비교분석하고자 한다.

Tabu Search는 C++ 언어로 프로그램되어 Pentium 60MHz CPU를 사용하여 실험하였으며, 결과분석은 실험의 신뢰도를 높이기 위하여 매 실험마다 초기해를 달리하여 60회 반복실험한 평균치를 사용하였다. 여기서 60회의 반복수는 평균결과치의 95% 오차한계를 1이내로 하기 위한 것이다.

4.1 실험 예제

본 연구에서 사용한 예제는 Dar-EI과 Nadivi [9]가 제시한 진화교환기 투입순서의 문제를 상황에 맞게 수정하여 사용하였다. 예제는 총 27개의 작업장으로 구성된 조립라인에서 8 종류의 모델을 생산한다. 한 사이클동안에 생산되는 모델별 생산제품수는 $d = \{1, 4, 2, 4, 1, 2, 2, 1\}$ 이고, 한 사이클동안에 생산할 총 제품수는 17개이다. 12개의 작업장(1, 5, 6, 7, 8, 13, 14, 20, 21, 22, 26, 27)은 폐쇄작업장이며, 그외 15개의 작업장은 개방작업장으로 구성되어 있다고 본다. 또한 10개의 작업장(1, 3, 4, 12, 13, 19, 20, 21, 25, 26)이 모델의 변화에 따라 준비시간이 발생한다고 본다. 모델별 작업장에 따른 작업시간(t_m)은 Dar-EI과 Nadivi[9의 표4]를 사용하고, 모델 변화에 따른 준비시간(s_{mr})은 $1 \leq s_{mr} \leq 5$, $m \neq r$, $j = 1, 3, 4, 12, 13, 19, 20, 21, 25, 26$ 의 범위를 갖는 일양분포에서 발생하였다. 컨베이어의 속도(v_c)는 계산의

편의를 위해 1로 두었다.

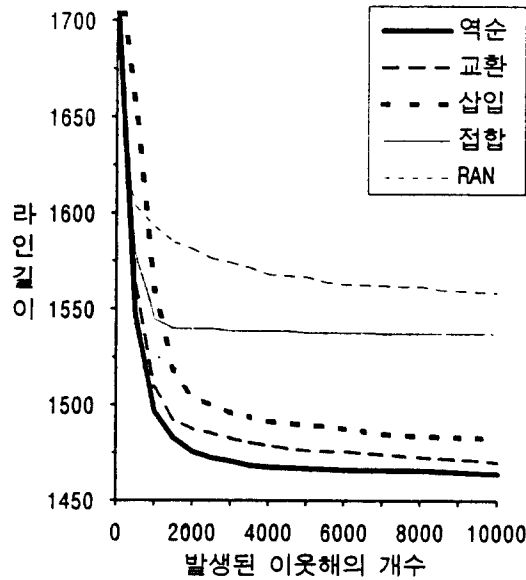
4.2 이동방법의 비교분석

교환, 삽입, 역순, 집합이동의 이웃해 생성방법들을 해의 효율측면에서 비교한다. 각 이동방법들은 가능한 모든 이웃해를 생성하며, 각 이동방법의 성능을 임의탐색(Random Search)과 비교하기 위해 매 반복마다 교환이동의 이웃해 개수($C_2 - \sum_{m=1}^M d_m C_2 = 121$)만큼 해를 임의로 발생시켜 Tabu Search를 이용한 "RAN" 방법을 사용하여 해를 탐색하였다.

각 이동방법에 적절한 Tabu목록의 크기를 찾기 위해 이동확률(q)을 1로 주고 Tabu목록의 크기를 5, 10, 15, ..., 50으로 변화시키면서 비교분석을 하였다. 다루는 문제에서 Tabu목록의 크기는 해의 질에 큰 영향을 주지 않았으나, 모든 이동방법에서 10~20 일때 비교적 해의 탐색성능이 좋았다.

[그림 5]는 각 이동방법에 대한 해의 탐색능력을 나타내고 있다. 이때 Tabu목록의 크기는 15, 이동확률(q)은 1로 두었다.

[그림 5]에서 보여주듯이 역순이동이 가장 좋은 해를 찾고, 다음은 교환, 삽입, 집합이동의 순이며, 임의 탐색(RAN)이 가장 나쁘게 나타났다. Tabu목록의 크기와 q 의 값을 변화시킨 여러 실험에도 비슷한 결과를 보였다. 역순이동은 원소의 위치 및 순서를 적절하게 변화시킴으로써 더 좋은 이웃해를 생성하는 것으로 보인다. 한편 교환, 삽입, 집합이동 순으로 성능이 떨어지는데, 이는 원소의 위치 및 순서의 변화가 심하면 이웃해의 탐색능력을 약화시키는 것으로 보인다.



[그림5] 이웃해 생성방법의 비교(Tabu 목록의 크기=15, $q=1$)

4.3 확률적 이동방법에 대한 분석

확률적 이동방법을 사용한 경우($0 < q < 1$)와 사용하지 않은 경우($q=1$)의 해의 탐색능력을 비교하기 위하여 Tabu목록의 크기와 q 의 값을 변화

시키면서 실험하였다. <표1>은 삽입, 역순이동을 사용하여 이웃해를 생성하고, 생성된 총 이웃해의 개수가 10,000개 일때의 가장 좋은 해를 나타낸 것이다.

<표1> 확률적 이동방법의 분석

이동확률→ 목록크기↓	$q=0.2$		$q=0.4$		$q=0.6$		$q=0.8$		$q=1.0$	
	삽입	역순	삽입	역순	삽입	역순	삽입	역순	삽입	역순
1	1486.5	1468.8	1486.9	1465.2	1489.3	1466.1	1492.8	1469.7	1494.7	1473.2
3	1485.7	1466.7	1484.1	1464.3	1485.3	1464.5	1487.5	1464.7	1490.2	1468.4
5	1484.1	1467.1	1480.6	1464.1	1481.9	1463.6	1483.0	1463.9	1486.9	1465.3
7	1484.7	1467.9	1480.9	1464.5	1480.1	1464.0	1481.5	1463.0	1482.2	1464.6
10	1485.9	1467.7	1482.1	1464.9	1481.5	1464.6	1479.0	1464.1	1481.2	1463.9
15	1486.1	1468.3	1483.2	1465.7	1483.2	1464.9	1482.2	1464.5	1480.3	1464.3
20	1486.4	1469.1	1484.2	1466.7	1483.6	1465.9	1483.9	1465.6	1482.5	1465.1
25	1487.1	1470.3	1485.9	1468.1	1484.5	1466.2	1484.2	1466.7	1483.1	1466.3
35	1487.7	1471.0	1486.5	1468.3	1485.6	1467.6	1485.2	1467.5	1484.2	1467.8
50	1488.9	1475.6	1487.0	1470.2	1486.7	1468.9	1485.6	1468.9	1484.9	1467.4

〈표 1〉에서 알 수 있는 결과는 다음과 같다.

1) 확률적 이동방법을 사용함으로써 해의 효율을 약간 높일 수 있으며, 상대적으로 Tabu목록의 크기를 줄일 수 있다. 즉 q 의 값을 0.6~0.8 정도로 둬으로써 해공간을 다양하게 탐색할 수 있다. 그러나 q 의 값이 너무 낮으면 해의 효율이 떨어진다는 것을 알 수 있다.

2) 해의 효율에 큰 영향을 미치는 요소는 이웃해의 생성방법이라는 것을 알 수 있다. 대부분의 경우 삽입이동보다 역순이동에서 해의 효율이 더 우수하였다.

4.4 Tabu Search, 유전알고리즘, 그리고 분지한계법의 비교분석

Tabu Search, 유전알고리즘, 그리고 분지한계법을 계산시간 및 해의 효율면에서 비교분석하기 위하여 〈표 2〉와 같이 문제를 발생하였다. 문제의 크기가 큰 경우 분지한계법의 계산시간이 너무 많이 소요되므로 문제의 크기를 비교적 작게 구성하였다. 모든 문제에서 라인은 3개의 작업장으로 구성되어 있고, 작업장 1만이 준비시간이 발생한다고 본다. 각 작업장에서의 모델별 작업시간

(t_{jm})과 작업장 1의 준비시간(s_{1m})은 다음과 같이 둔다.

$$t_{1m} = (18, 21, 19, 29, 29)$$

$$t_{2m} = (15, 27, 25, 25, 31)$$

$$t_{3m} = (20, 20, 23, 26, 29)$$

$$s_{1r} = (0, 12, 7, 5, 6)$$

$$s_{2r} = (5, 0, 11, 7, 9)$$

$$s_{3r} = (9, 8, 0, 5, 10)$$

$$s_{4r} = (7, 5, 6, 0, 5)$$

$$s_{5r} = (5, 11, 6, 9, 0)$$

여기서 $t_{1m} = (18, 21, 19, 29, 29)$ 는 작업장 1에서의 작업시간으로 문제1에서는 모델 1, 2, 3의 작업시간이 각각 18, 21, 19이고, 문제 2에서는 모델 1, 2, 3, 4에 대해 각각 18, 21, 19, 29이며 문제 3에서는 같은 방법으로 5가지 모델에 대해 차례로 18, 21, 19, 29, 29의 작업시간임을 나타낸다. 또한 $s_{1r} = (0, 12, 7, 5, 6)$ 은 작업장 1에서 모델 1 다음에 오는 모델에 따른 준비시간으로 문제 1의 경우 모델 $r=1, 2, 3$ 에 대해 각각 0, 12, 7이고, 문제2의 경우는 모델 $r=1, 2, 3, 4$ 에 대해 각각 0, 12, 7, 5이고, 문제3의 경우는 $r=1, 2, 3, 4, 5$ 에 대해 각각 0, 12, 7, 5, 6의 준비시간임을 나타낸다.

〈표 2〉 예제 문제의 형태

문제	모델의 수 (M)	생산제품의 수(I)	MPS의 수요 (d)	Total assembly content(T)	Total setup content(S)	제품간의 간격(w)
1	3	7	2, 2, 3	443	278	23
2	4	8	1, 2, 2, 3	563	316	25
3	5	10	1,2,2,2,3	750	591	27

Tabu Search는 이웃해의 생성방법으로 역순이동을 사용하였으며, Tabu 목록의 크기는 10, 가장 좋은해로 이동할 확률(q)은 0.8로 두었다. 유전알고리즘은 김여근과 현철주[1]가 제시했던 유

전알고리즘에서 평가함수(Evaluation function)만을 수정하여, 이원연산자(Binary operator)는 ISR, 일원연산자(Unary operator)는 역순을 5 : 5로 혼합하여 사용하였다. 유전알고리즘에서 모

집단의 크기는 100, 생존확률 파라미터는 모집단에
서 각 개체의 평균생존확률의 3배로 두었다. 또한
분지한계법은 Package인 Mixed Integer Linear
Programming (MILP)을 사용하였다.

〈표 3〉은 문제형태에 따라 Tabu Search, 유전
알고리즘, 그리고 분지한계법의 성능비교를 나타내
고 있다. Tabu Search와 유전알고리즘은 각 형태
의 문제를 초기해를 달리하여 30회 실험한 결과의
평균치를 사용하였다. 〈표 3〉에서 괄호안의 값은
효율을 나타내며, 이는 다음과 같이 정의된다.

$$\text{효율} = \left(1 - \frac{\text{구한해} - \text{최적해}}{\text{최적해}} \right) \times 100$$

〈표 3〉 Tabu Search, 유전알고리즘, 분지한계법의 비교

문제	분지한계법			유전 알고리즘			Tabu Search		
	최적	최소의	계산시간	재결합회수		계산시간	발생한 이웃해의 수		계산시간
	투입순서	라인길이	(sec.)	1,000	2,000	(sec.)	1,000	2,000	(sec.)
1	2211333	73	367	73(100)	73(100)	3.5	73(100)	73(100)	1.3
2	42213344	84	2,316	85.3(99.8)	85.1(99.9)	4	85.1(99.9)	85(100)	1.5
3	5334422155	95	102,644	96.9(98.0)	95.5(99.5)	4.4	96.7(98.2)	95.5(99.5)	2.0

5. 결 론

본 연구에서는 혼성작업장으로 이루어진 혼합
모델 조립라인에서 모델변화에 따른 준비시간을
고려하여, 전체 라인길이를 최소로 하는 모델의
투입순서를 구하는 수리모형을 개발하였다. 또한
실시간에 근사최적해의 투입순서를 구할 수 있는
Tabu Search 기법을 제시하였다. 이를 위해 다
루는 문제에 적합한 이웃해 생성방법을 찾고, 확
률적 이동방법을 제안하였다. 실험결과 이웃해 생
성방법 중 역순이동이 가장 좋았으며, 확률적 이
동방법을 사용함으로써 해의 효율을 향상시킬 수
있었다. 또한 Tabu Search는 실시간에 최적해에

실험결과는 다음과 같았다. Tabu Search는 유
전알고리즘과 함께 실시간에 최적해에 가까운 해
를 찾으며, 유전알고리즘보다 계산시간이 더 적게
소요됨을 알 수 있다. Tabu Search에서 계산시
간은 이웃해의 평가에, 유전알고리즘에서는 유전
연산과 개체의 평가에 대부분 소요되기 때문에
개체를 구성하는 인자의 수에 비례하여 소요되었
다. 분지한계법에서는 문제크기의 증가에 따라 계
산시간이 지수적으로 증가하였다.

가까운 해를 찾았으며, 유전알고리즘보다 계산시
간이 더 적게 소요됨을 알 수 있었다. 본 연구에
서 제시한 Tabu Search는 모델의 수요량 변화,
부품의 결품 등 현장의 상황변화에 따라 최적에
가까운 투입순서를 실시간에 구할 수 있다. 특히
자동차 조립라인에서 도장라인, 차체라인 및 의장
라인의 모델 투입 순서결정에 응용함으로써 컨베
이어 정지위험과 가외작업을 줄일 수 있다.

참 고 문 헌

1. 김여근, 현철주, “혼합모델 조립라인의 생산순
서결정을 위한 유전알고리즘”, 「대한산업공

- 학회지], 제20권, 제3호(1994), pp. 15-34.
2. Bard, J.F., Dar-El, E.M. and Shtub, A., "An analytic framework for sequencing mixed model assembly lines", *Int. J. Prod. Res.*, Vol. 30(1992), pp. 35-48.
 3. Bard, J.F., Shtub, A. and Joshi, S.B., "Sequencing mixed-model assembly lines to level parts usage and minimize line length", *Int. J. Prod. Res.*, Vol. 32(1994), pp. 2431-2454.
 4. Beaty, S.J., "Genetic algorithm versus Tabu search for instruction scheduling", *Proceedings of the fifth International Conference on Genetic Algorithms*, 1993, pp. 496-501.
 5. Belarmino, A.D., "Restricted Neighborhood in the Tabu Search for the Flowshop Problem", *European Journal of Operational Research*, Vol. 62, 1992, pp. 27-37.
 6. Bolat, A., Savsar, M. and Al-Fawzan, M. A., "Algorithms for real-time scheduling of jobs on mixed model assembly lines", *Computers Ops Res.*, Vol. 21(1994), pp. 487-498.
 7. Burns, L.D. and Daganzo, C.F., "Assembly line job sequencing principles", *Int. J. Prod. Res.*, Vol. 25(1987), pp. 71-99.
 8. Dar-El, E.M. and Cother, R.F., "Assembly line sequencing for model-mix", *Int. J. Prod. Res.*, Vol. 13(1975), pp. 463-477.
 9. Dar-El, E.M. and Nadivi, A., "A mixed-model sequencing application", *Int. J. Prod. Res.*, Vol. 19(1981), pp. 69-84.
 10. Dar-El, E.M., "Mixed-model assembly line sequencing problems", *Omega*, Vol. 6 (1978), pp. 313-323.
 11. Glover, F., "Future path for integer programming and links artificial intelligence", *Computers Ops Res.*, Vol. 13(1986), pp. 533-549.
 12. Glover, F., "Tabu search-part I", *ORSA Journal on Computing*, Vol. 1(1989), pp. 190-206.
 13. Glover, F., "Tabu search-part II", *ORSA Journal on Computing*, Vol. 2(1990), pp. 4-32.
 14. Kilbridge, M. and Wester, L., "The assembly line model-mix sequencing problem", *Proc. of the Third Int. Conf. Op. Res.*, Oslo, Paris: English Universities Press, 1963, pp. 247-260.
 15. Laguna, M., Barnes, J.W. and Glover, F., "Tabu search methods for a single machine scheduling problem", *Journal of Intelligent Manufacturing*, Vol. 2(1991), pp. 63-74.
 16. McCormick, S.T., Pinedo, M. L., Shenker, S. and Wolf, B., "Sequencing in an assembly line with blocking to minimize cycle time", *Opns Res.*, Vol. 37 (1989), pp. 925-937.
 17. Miltenburg, J. and Sinnamon, G., "Scheduling mixed-model multi-level just-in-time production systems", *Int. J. Prod.Res.*, Vol. 27(1989), pp. 1487-1509.
 18. Miltenburg, J., "Level schedules for mixed-model assembly lines in just-in-time production systems", *Mgmt. Sci.*, Vol. 35 (1989), pp. 192-207.
 19. Monden, Y., *Toyota Production System*,

- Institute of Industrial Engineers Press,
Norcross, GA, 1983.
20. Okamura, K. and Yamashina, H., "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(1979), pp. 233-247.
 21. Takashi, K., Hiroaki, K. and Masakazu, N., "A hybrid Search for genetic algorithms : Combining genetic algorithms, TABU Search and Simulated annealing", *Proceedings of the fifth International Conference on Genetic Algorithms*, 1993, pp. 641.
 22. Widmer, M. and Hertz, A., "A new heuristic method for the flow shop sequencing problem", *European Journal of Operations Research*, Vol. 41(1989), pp. 186-193.
 23. Yano, C.A. and Rachamadugu, R., "Sequencing to minimize work overload in assembly lines with product options", *Mgmt. Sci.*, Vol. 37(1991), pp. 572-586.