Journal of the Korean Institute of Industrial Engineers
Vol. 25, No. 4, pp. 411-420, December 1999.

혼합모델 U라인의 작업할당과 투입순서를 위한 공진화 알고리듬†

김여근[1] · 김선진[1] · 김재윤[1] · 곽재승[2]

[1]전남대학교 산업공학과 / [2]순천제일대학 품질경영과

A Coevolutionary Algorithm for Balancing and Sequencing Mixed-Model U-Lines

Yeo-Keun Kim[1] · Sun-Jin Kim[1] · Jae-Yun Kim[1] · Jai-Seung Kwak[2]

A mixed model production line is a production line where a variety of product models are produced. In U-shaped production lines (called U-lines) used in just-in-time production system, the strategy of mixing product models is often used to provide various types of products to customers in time. Line balancing and model sequencing problems are important for an efficient use of mixed model U-lines. Although the two problems are tightly interrelated with each other, prior researches have considered them separately or sequentially. This paper presents a new method using a coevolutionary algorithm that can solve the two problems at the same time. To promote diversity and search efficiency, in this paper the evolutionary system is based on the localized interactions within and between populations. Methods of selecting environmental individuals and evaluating fitness are developed. Efficient genetic representations and operator schemes are also provided. When designing the schemes, we take into account the features specific to the problems. The experimental results demonstrate that the proposed algorithm is superior to existing approaches.

1. 서 론

JIT(just-in-time) 생산 시스템에서 사용되는 U자형 생산라인(이하 U라인이라 함)에서는 고객에게 적시에 여러 종류의 제품을 제공하기 위하여 혼합모델 생산전략을 흔히 사용한다. 혼합모델 U라인에서 라인의 효율적인 이용을 위하여 작업할당과 투입순서문제는 매우 중요하다(Sparling and Miltenburg, 1998). 작업할당문제는 제품을 생산하기 위한 여러 작업들을 작업장에 할당하는 문제이며, 투입순서문제는 모델의 생산순서를 결정하는 문제이다. 본 연구에서는 혼합모델 U라인(Mixed-Model U-Line: MMUL)에서 두 문제를 동시에 해결할 수 있는 새로운 방법을 제안한다.

혼합모델 생산라인에서 작업할당과 투입순서문제는 서로 밀접한 관련성을 갖는다. 그러나, 기존 대부분의 연구에서는 두 문제를 서로 분리시켜 독립적으로 다루었다. 고전적인 직선라인에서 혼합모델 생산시 발생하는 작업할당문제(Chakravarty and Shtub, 1985; Macaskill, 1972; Roberts and Villa, 1970; Thomopoulos, 1970)와 투입순서문제(Hyun *et al.*, 1998; Kilbridge and Wester, 1963; Yano and Rachamadugu, 1991)에 관한 많은 연구들이 있다. MMUL의 작업할당문제는 Sparling과 Miltenburg (1998)에 의해 연구되었으며, JIT 생산시스템에서의 투입순서에 관한 연구는 Miltenburg와 Goldstein(1991), Kubiak (1993), 그리고 Monden(1993) 등에 의해 이루어졌다.

지금까지 MMUL에서 두 문제를 동시에 고려한 연구는 이루어지지 않았다. 혼합모델 직선라인에 대해서는 Thomopoulos (1967)와 Dar-El과 Nadivi(1981)가 두 문제를 계층적으로 다루었다. 즉, 한 가지 문제를 먼저 해결하고, 이로부터 나온 해를 제약으로 다른 한 문제를 해결하였다. 이러한 계층적 접근방법은 여러 개의 문제가 결합된 복합적인 문제에서 해공간을 효과적으로 탐색하는 데 한계를 갖는다.

인공 공진화(artificial coevolution)는 복잡하고 동적인 문제를

† 본 연구는 한국과학재단 특정기초연구(과제번호: 98-0200-09-01-3) 지원으로 수행되었음.

해결하는 방법론으로 인식되고 있다(Moriarty and Miikkulainen, 1997). 공진화 알고리듬(coevolutionary algorithm)은 자연계의 종들이 다른 종들과 서로 영향을 주고 받으며 진화하는 과정을 모방한 확률적 탐색기법이다(Koza, 1992). 이것은 여러 문제들이 서로 결합되어 있고, 한 문제의 해가 변함에 따라 다른 문제들의 해가 영향을 받는 상황과 유사하다고 볼 수 있다. 따라서 본 연구에서는 MMUL의 작업할당과 투입순서를 동시에 결정하는 새로운 공진화 알고리듬을 제안한다. 이 접근 방법은 계층적 방법에 비해 두 문제에 의해 형성된 해공간을 동시에 효과적으로 탐색할 수 있다는 특징을 갖는다.

본 연구의 구성은 다음과 같다. 2장에서는 본 연구에서 다루는 MMUL의 작업할당과 투입순서문제를 정의하고, 3장은 제안한 공진화 알고리듬에 관하여 기술한다. 4장과 5장은 두 문제를 위한 유전표현과 연산자에 관하여 다룬다. 6장에서는 실험을 통하여 제안한 알고리듬의 특성과 성능을 분석하고, 7장은 결론으로 구성되어 있다.

2. MMUL의 작업할당과 투입순서

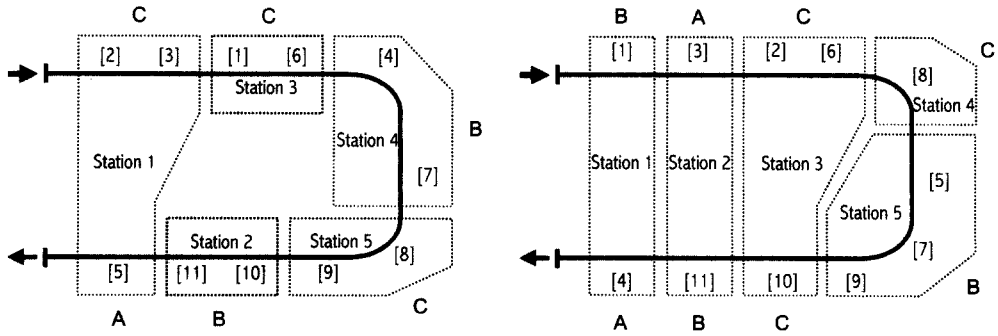
본 연구에서 다루는 MMUL은 다음과 같은 상황에서 운영된다고 가정한다. 첫째, 라인은 고정된 수의 작업장을 가지며, 각 작업장에는 단지 한 사람의 작업자가 할당되어 있다. 둘째, 결합선행공정도(Macaskill, 1972)가 주어졌다고 본다. 결합선행공정도는 작업선행관계, 작업구성, 작업시간, 그리고 생산량이 서로 다른 각 모델의 선행공정도를 하나로 통합한 것이다. 셋째, 사이클 생산방식을 취한다. 전체 계획생산기간 동안 M종류의 모델이 생산되며, 이 기간 동안 모델 m의 수요를 D_m (m = 1, 2, … , M)이라 하자. 그리고 D_1, D_2, … , D_M의 최대공약수가 b이면, $d = (D_1/b, \cdots, D_M/b) = (d_1, \cdots, d_M)$로 둔다. 이 d를 최소부품집합(minimum part set: MPS)이라 부르고, 이를 반복생산한다. 즉, MPS를 b번 반복생산하여 총 수요를 만족한다. 본 연구에서 다루는 모든 투입순서는 MPS를 기준으로 하며, 이는 혼합모델 생산라인에서 흔히 사용한다(Bard *et al.*, 1992). 그리고 작업자의 이동시간은 무시한다.

직선라인과 비교하여 U라인이 갖는 가장 큰 장점은 작업을 작업장에 할당하는데 있어 선행제약의 완화에 있다. 직선라인의 작업할당에서 할당가능 작업집합은 선행작업이 이미 할당된 모든 작업들로 구성된다. 그러나 U라인의 할당가능 작업집합은 선행작업이 모두 할당된 작업집합과 후행작업이 모두 할당된 작업집합의 합집합이 된다. 이러한 제약의 완화는 요구되는 작업장 수를 줄일 뿐만 아니라, 수요가 변경되었을 때 필요한 작업자 수를 손쉽게 조절할 수 있게 한다.

한편, 혼합모델의 투입순서에 있어서도 U라인은 직선라인과 차이점을 갖는다. 직선라인의 작업장에서 생산 사이클 동안 모든 작업자들은 하나의 모델만을 작업한다. 그러나 U라인에서 사이클 동안 작업자에게 할당된 작업은 생산라인의 앞과 뒤에 위치한 작업일 수 있다. 이러한 경우, 앞 작업의 모델과 뒷 작업 모델은 종종 다르다. 그리고 각 사이클마다 생산하는 모

표 1. 작업 선행관계와 작업시간

작업	직선행작업	작업시간		
		A	B	C
1		3	6	5
2		4	2	0
3		5	5	8
4	1,6	6	10	4
5	1	0	1	3
6	2,3	2	0	5
7	6	10	3	0
8	6	4	6	10
9	7	0	5	7
10	8	8	5	6
11	8,9	2	4	6
		44	47	54



(a) 작업할당 1 (b) 작업할당 2

그림 1. U라인의 작업할당 예.

델은 투입순서에 따라 변하게 된다. 또한, 만약 투입순서가 고정되고 작업할당이 변경된다면, 각 사이클마다 작업장에서 생산되는 모델은 달라질 수 있다. 그러므로 사이클 동안 작업장에 할당되는 작업량은 작업할당 또는 투입순서가 변함에 따라 변한다.

예를 들어, 작업선행관계와 작업시간이 <표 1>과 같이 주어졌다고 하자. <그림 1>은 <표 1>의 자료를 이용하여 U라인에서 작업장 수가 5로 주어진 경우, 가능한 두 작업할당의 예를 보인 것이다. 또한, 이러한 할당에 투입순서가 ABCBCC인 경우, 한 생산 사이클에서 작업장에 할당된 모델들을 나타내었다. <그림 1> (a)에서 작업장 1에 있는 작업자는 생산라인의 앞에서 모델 C의 작업 2와 3을 작업하고, 라인의 뒤에서 모델 A의 작업 5를 수행한다. 그러나 <그림 1>의 (b)에서 보는 바와 같이 작업할당이 달라지면 각 생산 사이클에서 각 작업장에 할당되는 모델 또한 변하게 된다.

MMUL에서 작업할당과 투입순서의 결정기준은 다양하다 (Monden, 1993; Sparling and Miltenburg, 1998). 본 연구에서는 각 사이클동안 작업장의 작업량 평활화를 목적으로 한다. 작업량 평활화는 작업자간의 작업 형평을 이룰 수 있고, 라인혼잡을 감소시킬 수 있으며, 생산량을 증가시킬 수 있다. 작업량 평활화의 평가 척도로 Sparling과 Miltenburg(1998)가 사용한 작업량 절대편차(absolute deviation of workloads: ADW)를 사용한다. ADW의 계산에 사용되는 기호는 다음 쪽에 설명한다.

주어진 작업할당과 투입순서에 대하여 ADW는 다음과 같이 계산된다.

$$ADW = \sum_{j=1}^{J} \sum_{s=1}^{S} |T_{js} - \overline{T}| \quad (1)$$

<표 1>의 자료와 <그림 1>의 작업할당을 사용하여 각 사이클마다 할당된 작업량의 변화를 보기로 하자. 예로, MPS가 $d = (d_A, d_B, d_C) = (1, 2, 3)$과 같고, MPS 제약을 만족하는 투입순서

표 2. 작업장 1의 작업량 절대 편차

(a) 투입순서가 ABCBCC 인 경우

	작업할당 1				작업할당 2			
	$IF_1 = \{2, 3\}$	$IB_1 = \{5\}$	T_{1s}	$\lvert T_{1s} - \overline{T} \rvert$	$IF_1 = \{1\}$	$IB_1 = \{4\}$	T_{1s}	$\lvert T_{1s} - \overline{T} \rvert$
Models								
cycle 1	C	A	8	2	B	A	12	2
cycle 2	A	B	10	0	C	B	15	5
cycle 3	B	C	10	0	B	C	10	0
cycle 4	C	B	9	1	C	B	15	5
cycle 5	B	C	10	0	C	C	9	1
cycle 6	C	C	11	1	A	C	7	3
			58	4			68	16

(b) 투입순서가 CACBCB 인 경우

	작업할당 1				작업할당 2			
	$IF_1 = \{2,3\}$	$IB_1 = \{5\}$	T_{1s}	$\lvert T_{1s} - \overline{T} \rvert$	$IF_1 = \{1\}$	$IB_1 = \{4\}$	T_{1s}	$\lvert T_{1s} - \overline{T} \rvert$
Models								
cycle 1	B	C	10	0	A	C	7	3
cycle 2	C	A	8	2	C	A	11	1
cycle 3	A	C	12	2	B	C	10	0
cycle 4	C	B	9	1	C	B	15	5
cycle 5	B	C	10	0	B	C	10	0
cycle 6	C	B	9	1	C	B	15	5
			58	6			68	14

I : 작업 수.
J : 작업장 수.
S : MPS생산동안 생산되는 제품의 수; $S = \sum_{m=1}^{M} d_m$.
t_{im} : 모델 m에서 작업 i의 작업시간.
$\overline{T}$: 이론적 최소사이클타임, 즉 평균 작업량;
$\overline{T} = (1/(J \times S)) \sum_{i=1}^{I} \sum_{m=1}^{M} d_m t_{im}$.
IF_j : U라인에서 작업장 j의 앞에 할당된 작업집합.
IB_j : U라인에서 작업장 j의 뒤에 할당된 작업집합.
f_j^s : 작업장 j의 앞에서 s번째 사이클에 생산된 모델.
b_j^s : 작업장 j의 뒤에서 s번째 사이클에 생산된 모델.
T_{js} : 작업장 j에서 s번째 사이클에 할당된 작업들의 작업시간 합;
$T_{js} = \sum_{i \in IF_j} t_{if_j^s} + \sum_{i \in IB_j} t_{ib_j^s}$

가 ABCBCC와 CACBCB로 주어졌다고 하자. 작업할당과 투입순서가 주어진 경우, <표 2>는 1회의 MPS생산 동안 작업장 1에서 각 사이클 동안 생산되는 모델과 작업량을 보인 것이다. 이로부터 작업할당과 투입순서는 모두 U라인의 작업량에 영향을 미침을 알 수 있다.

3. MMUL의 작업할당과 투입순서를 위한 공진화 알고리듬

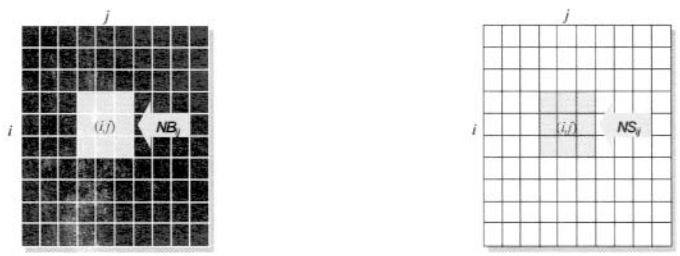
3.1 제안한 공진화 알고리듬

공진화 알고리듬은 둘 이상의 종들이 상호작용하며 진화해 가는 생물의 공진화 과정을 모방한 확률적 탐색기법의 하나이다. 공진화 알고리듬은 다양한 변형이 있지만, 협조 공진화 알고리듬과 경쟁 공진화 알고리듬으로 크게 분류된다. 협조 공진화 알고리듬(Moriarty and Miikkulainen, 1997; Potter, 1997)은 각 모집단의 개체가 서로 결합되어 하나의 해를 나타내며, 개체들이 다른 모집단이 제공하는 환경에 얼마나 잘 적응하는가를 측정하는 것을 목적으로 한다. 경쟁 공진화 알고리듬(Pagie and Hogeweg, 1997; Rosin and Belew, 1997)에서는 개체들이 다른 개체와 경쟁하고, 상대의 다양한 전략에 잘 적응하는 좋은 전략을 찾아내는데 적용된다.

본 연구에서는 협조 공진화 알고리듬을 기반으로 알고리듬을 개발한다. 이 알고리듬은 복잡하고 동적인 문제에 고전적인 진화 알고리듬에 비해 우수한 해를 제공한다(Moriarty and Miikkulainen, 1997). 여기서 고려하는 문제의 경우, MMUL의 ADW는 작업할당문제와 투입순서문제의 정보가 모두 주어져야만 계산할 수 있으며, 두 문제는 서로 특성이 다르다. 따라서, 두 문제를 각각 하나의 종으로 보고 공진화하는 모형을 개발하며, 각 문제들은 문제의 특성을 반영한 유전인자들에 의해 표현한다.

진화 알고리듬이 좋은 성능을 갖기 위하여, 모집단의 다양성 유지는 매우 중요하다. 다양성과 탐색의 효율성을 높이기 위하여 본 연구에서는 모집단내와 모집단간 지역적 진화를 기본으로 하는 전략을 사용한다. 작업할당 모집단(Pop-B)과 투입순서 모집단(Pop-S)은 <그림 2>와 같이 2차원 격자구조로 구성되며 토러스(torus) 형태를 갖는다고 본다. Pop-B와 Pop-S에서 위치 (i, j)에 있는 개체의 이웃을 각각 NB_{ij}, NS_{ij}로 정의하자. 이웃은 크기나 형태에 따라 서로 다른 방법으로 정의할 수 있으나(Sarma and DeJong, 1996), 본 연구에서는 3 × 3의 이웃구조를 사용한다. 그러면 NB_{ij}와 NS_{ij}는 <그림 2>에서 보는 바와 같이 개체 (i, j)와 주위 8개의 개체들이 이웃에 포함된다.

한 모집단은 진화하는 동안 다른 한 모집단과 다음의 과정을 반복하며 상호작용한다. 우선, 지역적 진화의 중심인 개체 (i, j)를 선택한다. NB_{ij}의 진화는 NB_{ij} 뿐만 아니라 NS_{ij}내의 개체들과도 상호작용을 통하여 이루어진다. 여기서 NS_{ij}의 가장 높은 적응도를 갖는 개체가 NB_{ij}의 환경개체로서 작용하여 적응도 평가에 참여한다. 즉, NB_{ij}의 개체들은 NS_{ij}의 가장 높은 적응도를 갖는 개체와 협조하며 적응해 나간다. NB_{ij}가 진화한 후, NS_{ij}의 진화는 진화와 환경의 역할을 변경하여 NB_{ij}에서와 유사한 과정으로 진행된다. 이러한 공진화 과정을 종료조건이 만족될 때까지 반복한다. 제안한 공진화 알고리듬의 단계별 절차는 다음과 같다.



(a) Pop-B (b) Pop-S

(a) Pop-B (b) Pop-S

그림 2. 작업할당 모집단(Pop-B)과 투입순서 모집단(Pop-S).

단계1 (초기화)
작업할당과 투입순서 개체로 이루어진 모집단 Pop-B와 Pop-S를 구성한다.

단계2 (초기 적응도 평가)
두 모집단의 개체들을 격자구조의 위치에 대응하게 짝지어 초기 적응도를 평가하고, 가장 좋은 해를 f_{best}로 둔다.

단계3 (이웃설정)
임의의 위치 (i, j)를 선택하고, NB_{ij}와 NS_{ij}를 정의한다.

단계4 (NB_{ij}의 진화)

단계4.1: NB_{ij}에서 적응도를 기준으로 두 부모개체를 선택하여 교차한 후 두 자손개체를 생산한다.

단계4.2: NB_{ij}에서 낮은 적응도를 갖는 두 개체를 선택하여 단계 4.1에서 생산된 자손개체와 대체한다.

단계4.3: 돌연변이율에 의해 NB_{ij}의 개체들에 대하여 돌연변이시킨다.

단계4.4: 새롭게 생산된 자손개체들의 적응도를 평가한다. 이때, NS_{ij}에서 가장 높은 적응도를 갖는 개체가 환경개체로 선택된다. 만약, 가장 좋은 개체의 적응도가 f_{best}보다 높으면, f_{best}를 갱신한다.

단계4.5: NB_{ij} 진화의 종료조건을 만족하면 단계5로, 그렇지 않으면 단계4.1로 간다.

단계5 (NS_{ij} 의 진화)

단계5.1: NS_{ij}에서 적응도를 기준으로 두 부모개체를 선택하여 교차한 후 두 자손개체를 생산한다.

단계5.2: NS_{ij}에서 낮은 적응도를 갖는 두 개체를 선택하여 단계 5.1에서 생산된 자손개체와 대체한다.

단계5.3: 돌연변이율에 의해 NS_{ij}의 개체들에 대하여 돌연변이시킨다.

단계5.4: 새롭게 생산된 자손개체들의 적응도를 평가한다. 이때, NB_{ij}에서 가장 높은 적응도를 갖는 개체가 환경개체로 선택된다. 만약, 가장 좋은 개체의 적응도가 f_{best}보다 높으면, f_{best}를 갱신한다.

단계5.5: NS_{ij} 진화의 종료조건을 만족하면 단계6으로, 그렇지 않으면 단계5.1로 간다.

단계6 (종료조건)
알고리듬 종료조건을 만족하면 종료한다. 그렇지 않으면 단계3으로 간다.

두 모집단 Pop-B와 Pop-S는 각각 단계 4와 5에 의해 NB_{ij}와 NS_{ij}가 다양하게 변경되는 동안 진화를 반복한다. 두 모집단의 진화는 이웃단위로 진행되며, 안정상태 유전알고리듬 (steady state genetic algorithm)의 형태를 따른다. 이 알고리듬은 세대별 유전알고리듬에 비해 해의 질과 계산시간 측면에서 성능이 우수한 것으로 알려져 있다(Syswerda, 1991; Whitley, 1989). NB_{ij}의 진화는 재생산 사이클(reproduction cycle)이라 불리는 단계 4.1부터 단계 4.4까지를 반복하며 진화하고, NS_{ij}의 진화도 유사하다. 각 이웃내의 진화 종료조건은 이러한 재생산 사이클 수로 하였다.

환경개체의 선택과 적응도 평가방법은 공진화 알고리듬의 성공적인 적용에 있어 핵심이 된다. 이것은 전통적인 유전알고리듬과 다르므로, 다음 절에서 자세히 다루기로 한다. 또한 Pop-B와 Pop-S는 서로 특성이 다른 문제를 표현하므로, 이들의 특성을 효과적으로 다룰 수 있는 방법이 요구된다. Pop-B와 Pop-S의 유전표현과 유전연산은 4장과 5장에서 각각 설명한다. 그리고 모집단의 크기, 돌연변이율, 종료조건 등을 포함한 유전 파라미터는 반복실험을 통하여 결정하며, 6장에서 설명하기로 한다.

3.2 환경개체 선택과 적응도 평가

공진화 알고리듬에서는 개체들의 적응도를 평가하기 위하여 환경개체를 결정해야 한다. Maher와 Poon(1996)은 환경개체를 선택하는 두 가지 방법을 제안하였다. 개체가 일차원으로 배열된 두 모집단 Pop-1과 Pop-2가 있다고 하자. 한 방법은 각 세대에서 Pop-1의 i번째 개체의 환경을 Pop-2의 i번째 개체로 하는 것이다. 이 모형은 강결합 공진화(tightly-coupled coevolution) 모형이라 하며, 한 개체가 특성이 서로 다른 두 염색체로 표현된 하나의 모집단을 운영하는 경우와 유사하다. 다른 한 방법으로, Pop-1에 있는 모든 개체의 환경개체는 Pop-2에서 현재 세대에 가장 높은 적응도를 갖는 개체가 된다. 그러므로, 모든 개체들은 한 세대 동안 동일한 환경을 공유한다. 이것은 약결합 공진화(loosely-coupled coevolution) 모형이라 부른다. 본 연구에서는 두 번째 모형을 변형하여 사용한다. 즉, 환경개체는 모집단의 전체 개체들에서 선택하지 않고, 각 세대마다 이웃내에서 가장 높은 적응도를 갖는 개체로 선택한다.

부모개체, 그리고 자손개체와 대체되는 개체를 선별하는데 적응도를 기준으로 한 확률바퀴 선별방법을 사용한다. NB_{ij}의 위치 (s, t)에 있는 개체의 적응도 $f_B(s, t)$는 식 (2)를 이용하여 평가한다. 식 (2)에서 $g_{es}(s, t)$는 Pop-B의 (s, t)번째 개체와 NS_{ij}의 환경개체 es가 결합되어 식 (1)에 의해 계산된 ADW이다.

$$f_B(s,t)=\frac{g_{es}(s,t)-\left\{\max_{(p,q)\in NB_{ij}} g_{es}(p,q)+1\right\}}{\min_{(p,q)\in NB_{ij}} g_{es}(p,q)-\left\{\max_{(p,q)\in NB_{ij}} g_{es}(p,q)+1\right\}} \quad (2)$$

식 (2)에 의하여 ADW가 작은 개체일수록 높은 적응도를 부여받는다. ADW는 항상 0보다 크거나 같기 때문에 적응도는 (0, 1]의 범위를 갖는다. NS_{ij}내 개체의 적응도 $f_S(s, t)$는 환경개체 eb를 NB_{ij}로 부터 선택하여 유사한 방법에 의해 계산된다. 따라서 식 (2)에서의 $g_{es}(s, t)$와 NB_{ij}를 $g_{eb}(s, t)$와 NS_{ij}로 각각 변경하면 된다.

NB_{ij}의 위치 (s, t)에 있는 개체가 자손개체와 대체될 확률은 식 (3)에 의해 결정한다. 이 식에 의하여 대체확률은 ADW가 높

을수록 더 높게 부여된다. NS_{ij}의 개체들의 대체확률도 유사하게 계산한다.

$$P_{de}(s,t)=\frac{f'_B(s,t)}{\sum_{(m,n)\in NB_{ij}} f'_B(m,n)} \quad (3)$$

여기서,

$$f'_B(s,t)=\frac{\{1+g_{es}(s,t)\}-\min_{(p,q)\in NB_{ij}} g_{es}(p,q)}{\left\{1+\max_{(p,q)\in NB_{ij}} g_{es}(p,q)\right\}-\min_{(p,q)\in NB_{ij}} g_{es}(p,q)}$$

4. MMUL의 작업할당을 위한 유전표현과 연산자

4.1 유전표현과 초기 모집단

개체는 자연스럽고 명확하면서도 중복되지 않도록 표현되는 것이 바람직하다. 또한 표현은 개체가 갖는 중요 정보가 유전연산에 의해 용이하게 추출되어 자손에게 잘 전파될 수 있어야 한다. 본 연구에서는 작업할당문제에 대하여 그룹번호표현을 사용한다. 한 개체의 길이는 I(작업의 수)이며, 각 인자는 [1, 2J-1]사이의 정수값을 갖는다. 만약, 개체의 i번째 인자값 j가 J보다 작거나 같으면, 이것은 작업 i가 U라인에서 앞쪽에 위치한 작업장 j에 할당됨을 의미한다. 그러나 i번째 인자값이 J보다 큰 2J-j이면, 작업 i는 뒤쪽에 위치한 작업장 j에 할당된다. 예로, <그림 1> (a)와 (b)의 작업할당을 개체로 표현하면 각각 (3 1 1 4 9 3 4 5 5 8 8)과 (1 3 2 9 5 3 5 4 5 7 8)와 같으며, 여기서 작업장 수 J = 5이다. 하나의 예로 두 표현에서 작업 4의 인자값은 각각 4와 9이다. 이는 작업 4가 <그림 1> (a)의 할당에서는 작업장 4의 앞 작업장에, 그리고 <그림 1> (b)의 할당에서는 작업장 1의 뒷 작업장에 할당됨을 의미한다. 이 표현은 별도의 해석방법이 필요 없으며, 각 인자가 갖는 정보를 명확히 나타내 준다는 장점을 갖는다. 반면, 가능해를 생산하는 유전연산자의 개발이 요구된다. 이러한 유전연산자는 다음 절에서 설명하기로 한다.

초기 가능해는 모든 작업이 할당될 때까지 아래의 과정을 반복하여 생성한다. 우선, 작업장 j (=1)에 대하여 할당을 고려하고, 할당가능 작업집합을 구한다. FP는 모든 선행작업이 할당된 작업들로 이루어진 작업집합이며, FS는 모든 후행작업이 할당된 작업들로 이루어진 작업집합이라 하자. U라인의 할당가능 작업집합은 FP와 FS의 합집합이다. 할당가능 작업집합으로부터 임의의 작업 i를 선택한다. 만약, $i\in FP$이면, 작업 i에 인자값 j를 할당하고, 그렇지 않으면, 인자값 2J-j를 할당한다. 그리고 다음 작업을 할당하기 위하여 할당가능 작업집합을 갱신한다. $\hat{t}_i\ (=\sum_{m=1}^{M} d_m t_m)$는 MPS생산동안 작업 i의 총 작업시간, $\hat{T}\ (=(1/J)\sum_{i=1}^{I}\hat{t}_i)$는 MPS생산 동안 각 작업장의 평균작업량으로 정의하자. 만약, MPS생산동안 작업장에 할당된 총 작업량 $\sum\hat{t}_i$가 $\hat{T}$보다 크면, 다음 작업장 j+1을 고려한다. 그리고 할당할 후보 작업은 현재 고려중인 작업장이나 이전 작업장중, 하나에 임의로 할당한다. (J-1)번째 작업장까지 작업을 할당한 후, 남아있는 모든 작업들은 작업장 J에 할당한다.

4.2 유전연산자

유전연산자는 문제의 특성을 잘 반영할 수 있는 정보를 부모로부터 추출하여 자손을 생산할 수 있어야 하며, 재생산된 개체의 가능해가 유지되도록 해야 한다. 본 연구에서는 구조교차(von Laszewski, 1991)를 작업할당문제에 적합하게 변형하여 사용하였다. 그 절차는 다음과 같다.

단계1: 범위 [1, J]에서 임의의 정수 r을 선택한다.
단계2: 부모 P1에서 작업장 1부터 r까지 표현한 [1, r]과 [2J-r, 2J-1]의 인자를 자손 O1의 동일한 위치에 상속한다.
단계3: 자손 O1의 남아있는 위치에 부모 P2에 있는 [r+1, 2J-r-1]의 인자를 상속한다.
단계4: 자손 O1의 비어있는 위치는 재할당방법을 이용하여 할당한다.

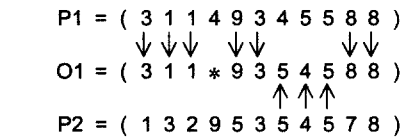


그림 3. 작업할당문제를 위한 교차.

자손 O2는 P1과 P2의 역할을 바꿔 생산한다. <그림 3>은 r = 3일 때, 교차의 예를 보이고 있다.

진화 알고리듬에서 돌연변이는 알고리듬의 탐색능력을 향상시키는데 중요한 역할을 한다. 작업할당문제에서는 개체 돌연변이율 P_m에 의해 개체들을 임의 선택하고, 선택된 개체에 대하여 인자 돌연변이율 P_g에 의해 인자를 돌연변이한다. 이들 인자의 돌연변이는 다음에서 설명할 재할당절차에 의해 이루어진다. 먼저, 재할당 과정에 사용된 새로운 기호는 다음 쪽에 설명되어 있다.

제안된 재할당기법에서는 문제가 갖는 특정 정보와 발견적 기법을 이용하여 선행제약의 만족은 물론 작업장의 작업량이 평활화되도록 한다. 여기서 작업장의 작업량 평활화는 각 작업장의 작업량 절대편차의 합($\sum_{j=1}^{J}\left|\sum_{s=1}^{S} T_{js}-\hat{T}\right|$)을 최소화하는 것이다. 이 기법의 절차는 아래와 같다.

R : 미할당되었거나 재할당이 요구되는 작업집합.

$IP(i)$: 작업 i의 직선행작업집합.

$S(i)$: 작업 i의 후행작업집합.

$NR(i)$: 집합 R의 원소 중 작업 i의 직선행작업집합; $NR(i) = R \cap IP(i)$.

Ei : 개체에서 $IP(i)$의 작업들에 해당하는 인자의 값 중에서 가장 큰 값,
만약 $IP(i) = \emptyset$ 이면, $Ei = 1$.

Li : 개체에서 $S(i)$-R의 작업들에 해당하는 인자의 값 중에서 가장 작은 값,
만약 $S(i)$-$R = \emptyset$ 이면, $Li = 2J$-1.

$AW(i)$: $\{Ei, Ei+1, \ldots, Li\}$.

$I(k)$: $\begin{cases} 1, & k \neq J \text{ 이면,} \\ 0, & \text{그렇지않으면} \end{cases}$

$\widehat{T}_k$: 이미 할당된 작업 중에서 인자값이 k인 작업들의 MPS생산시간의 합

$\widehat{T}$ $= (1/J) \sum_{i=1}^{I} \widehat{t}_i$

단계1: 집합 R과 $NR(i)$를 구하고, $\widehat{T}_k$, $k = 1, 2, \cdots, 2J$-1을 계산한다.

단계2: 집합 R의 원소이고, $NR(i) = \emptyset$ 인 작업 중 MPS생산 동안의 작업시간이 가장 큰 작업 i^*를 선택한다.

단계3: i^* 의 E_{i^*}와 L_{i^*}를 구하고, i^*의 할당가능 인자값 집합 $AW(i^*)$를 구한다.

단계4: 집합 $K = \{k \mid \widehat{T}_k + I(k)\,\widehat{T}_{2J-k} + \widehat{t}_{i^*} \leq \widehat{T} \text{ and } k \in AW(i^*)\}$이라 하자. 만약, $K \neq \emptyset$ 이면, K내에서 가장 작은 값을 k^*로 두고, 그렇지 않으면 $\widehat{T}_k + I(k)\,\widehat{T}_{2J-k}$, $k \in AW(i^*)$이 가장 작은 k를 k^*로 둔다.

단계5: 작업 i^* 의 해당 인자값을 k^*로 두고, i^* 를 집합 R에서 삭제한다. $R = \emptyset$ 이면 종료하고, 그렇지 않으면 $NR(i)$와 $\widehat{T}_k'$, $k = 1, 2, \cdots, 2J$-1을 갱신하고 단계2로 간다.

이 방법은 일종의 greedy heuristic으로서 그룹분할문제에서 좋은 결과를 얻는 것으로 알려진 best-fit-decreasing 규칙(Falkenauer, 1994)을 작업할당문제에 적합하게 수정한 것이다. 이러한 재할당방법을 사용함으로써 작업장간 작업량을 균등하게 할당하는 효과를 갖는다.

5. MMUL의 투입순서를 위한 유전표현과 연산자

5.1 유전표현과 초기 모집단

투입순서 모집단의 개체들은 MPS 생산 동안 생산되어야 하는 모델들을 투입순서의 나열로 표현한다. 예를 들어, MPS가 $d = (d_A, d_B, d_C) = (1, 2, 3)$이라 가정하고, 투입순서가 A,B,C,B,C,C라면 개체는 모델의 투입순서대로 (A B C B C C)와 같이 표현한다. 이 표현은 개체의 인자 위치에 투입순서 정보를 나타낼 수 있다. 초기 모집단은 MPS 제약을 만족하는 개체들을 모집단의 크기만큼 임의로 생성하여 구성한다.

5.2 유전연산자

투입순서문제를 위한 유전연산자로 수정순서교차(modified order crossover)와 역순(inversion)연산자를 사용한다. 혼합모델 투입순서문제에서 이들 두 연산자를 결합하여 사용함으로써 좋은 탐색 성능을 보인 것으로 나타났다(Kim *et al.*, 1996).

수정순서교차는 Davis(1985)가 제안한 순서교차를 투입순서문제에 적합하게 변형한 것이다. 두 부모로부터 두개의 절단

표 3. 실험문제

문제	작업수	모델수	작업장수	MPS	문제	작업수	모델수	작업장수	MPS
Thom1	19	3	3	1 1 1	Arcus1	111	5	12	1 1 1 1 1
Thom2	19	3	3	3 2 1	Arcus2	111	5	12	5 3 2 1 1
Thom3	19	3	4	1 1 1	Arcus3	111	5	12	1 2 4 5 8
Kim1	61	4	6	1 1 1 1	Arcus4	111	5	12	1 4 8 3 1
Kim2	61	4	6	1 3 4 5	Arcus5	111	5	15	1 1 1 1 1
Kim3	61	4	6	6 4 2 1	Arcus6	111	5	15	5 3 2 1 1
Kim4	61	4	12	1 1 1 1	Arcus7	111	5	15	1 2 4 5 8
Kim5	61	4	12	1 3 4 5	Arcus8	111	5	15	1 4 8 3 1
Kim6	61	4	12	6 4 2 1	Arcus9	111	5	27	1 1 1 1 1
					Arcus10	111	5	27	5 3 2 1 1
					Arcus11	111	5	27	1 2 4 5 8
					Arcus12	111	5	27	1 4 8 3 1

점을 임의로 선택하고, 부모 P1에 있는 절단점 사이의 인자들을 자손 O1의 동일한 위치에 상속시킨다. 부모 P2에서는 부모 P1에 있는 절단점 사이의 인자들을 임의로 삭제하고, P2에 남아 있는 인자들을 P2의 순서를 유지하면서 자손 O1의 아직 채워지지 않은 위치로 상속한다. 그리고 P1과 P2의 역할을 바꿔 자손 O2를 생성한다. <그림 4>는 수정순서교차의 예를 보인 것이다. 절단점은 " | " 로 표시되었으며, P2에서 음영으로 표현된 인자는 삭제된 인자를 나타낸다. 수정순서교차는 부모의 상대적 순서가 자손에 유지되는 연산자이다.

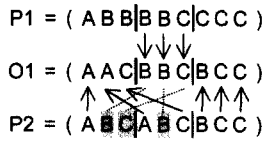


그림 4. 수정순서교차.

역순은 돌연변이 연산자이다. 부모에서 두 개의 절단점을 임의로 선택하고, 절단점 사이의 인자들을 역순으로 하여 자손을 생산한다. <그림 5>는 역순의 예를 보인 것이다. P와 O는 각각 부모와 자손개체이다.

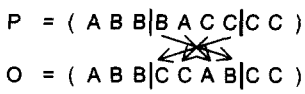


그림 5. 역순연산자.

6. 실험결과 및 분석

6.1 실험문제와 파라미터 결정

컴퓨터 실험을 통하여 제안된 공진화 알고리듬의 성능을 분석하였다. 실험은 19개의 작업을 갖는 Thomopoulos문제(Thomopoulos, 1967), 111개 작업을 가진 Arcus문제(Arcus, 1963, Appendix X-3)를 대상으로 하였다. 이들 두 문제 이외에 저자가 산학연구에서 얻은 작업이 61개인 문제를 실험문제로 사용하였다. 다양한 문제에 대한 실험을 위하여 <표 3>에서 보는 바와 같이 작업장 수와 MPS를 변화시켜 실험문제로 사용하였다. Arcus문제의 경우 작업장 수가 많은 경우를 실험하기 위하여 작업 95의 작업시간을 33491에서 6615로 변경하였다.

제안된 알고리듬은 C^{++}프로그램 언어로 구현되었으며, 166MHz Pentium CPU를 장착한 IBM-PC에서 수행되었다. 다른 파라미터는 예비실험을 통하여 결정하였다. 첫째, 모집단 크기는 100(10 × 10의 격자구조)으로 하였다. 둘째, NB_{ij}와 NS_{ij}의 재생산 사이클은 2-3회 반복한 것이 좋은 성능을 보였다. 이 횟수는 각기 확률 0.5를 사용하여 선택하였다. 셋째, Pop-B와 Pop-S의 개체 돌연변이율 P_m은 0.05, 그리고 Pop-B에서 돌연변이 되는 개체의 인자 돌연변이율 P_g는 0.1로 두었다. 끝으로 알고리듬 종료조건으로 생산된 개체수를 사용하였다. 작업 수와 모델 수에 따라 해공간이 다르므로, Thomopoulos문제, Kim문제, Arcus문제에 대하여 두 모집단에서 생산된 개체 수가 각각 7,500개, 15,000개 그리고 25,000개이면 종료하였다.

6.2 제안된 공진화 알고리듬의 성능분석

제안된 공진화 알고리듬(Proposed Coevolutionary Algorithm: PCoA)의 성능은 계층적 방법과 강결합 공진화 알고리듬(Tightly-coupled Coevolutionary Algorithm: TCoA), 그리고 약결합 공진화 알고리듬(Loosely-coupled Coevolutionary Algorithm: LCoA)과 비교하였다. 다루는 두 문제를 해결하는 계층적 방법은 일반적으로 작업할당문제를 해결한 후, 이를 제약으로 투입순서를 결정한다. 작업할당 문제는 작업장의 작업량 절대편차 최소화를 목적으로 4장에서의 표현과 유전연산을 사용하였으며, 이로부터 나온 작업할당 정보를 가지고 5장에서 제시한 표현과 유전연산을 이용하여 ADW 최소화를 목적으로 투입순서문제를 해결하였다.

TCoA와 LCoA는 다음과 같이 실험하였다. 개체의 적응도와 대체확률은 3.2절에서 소개한 식 (2)와 (3)을 사용하여 계산하였다. 적응도를 기준으로 모집단에서 55%의 개체들을 선별하여 교차하였고, 생산된 자손개체들은 낮은 적응도를 갖는 개체들과 대체하였다. 그리고 돌연변이율에 따라 개체들을 돌연변이시키고, 이러한 과정을 종료조건이 만족될 때까지 반복하였다. 두 공진화 알고리듬의 모든 파라미터와 유전연산자는 제안된 공진화 알고리듬에서와 동일하게 사용하였다. TCoA와 LCoA의 구체적인 절차는 Maher와 Poon의 연구(Maher and Poon, 1996)에서 참조할 수 있다.

<표 4>는 실험결과이다. 첫번째 열은 실험문제, 두번째 열은 계층적 방법에 의해 얻어진 ADW이다. 세번째 열부터 다섯번째 열까지는 세 가지 공진화 알고리듬에 의한 ADW로, 모든 값들은 10회 반복실험의 평균값이다. 마지막 열은 계층적 방법에 대한 제안한 공진화 알고리듬의 개선율로, {(계층적 방법-PCoA)/계층적 방법} × 100(%)의 식에 의해 구하였다.

실험 결과, 모든 실험문제에서 모든 공진화 알고리듬이 계층적 방법보다는 좋은 해를 탐색하였다. 이것은 복합적인 문제를 해결하는 고전적인 방법이 해공간을 효과적으로 탐색하지 못함을 의미한다.

공진화 알고리듬 중에서는 PCoA가 다른 알고리듬에 비해 모든 문제에서 우수한 성능을 보였다. 이들 알고리듬간에는 환경개체를 선택하는 방법에 차이가 있다. 또한, 모집단 단위로 진화하는 TCoA와 LCoA와 달리, PCoA는 이웃단위로 진화한

표 4. 제안한 공진화 알고리듬의 성능 비교

문제	계층적방법	공진화 알고리듬			개선율(%)
		TCoA	LCoA	PCoA	
Thom1	4.76	1.90	1.88	1.40	70.59
Thom2	7.29	3.50	3.44	2.78	61.82
Thom3	5.82	2.54	2.48	1.56	73.20
Kim1	70.85	39.10	35.34	22.95	67.61
Kim2	197.49	135.60	122.42	86.39	56.25
Kim3	184.31	113.30	104.20	79.78	56.72
Kim4	100.92	68.94	61.70	44.75	55.66
Kim5	298.63	218.39	203.50	152.04	49.09
Kim6	292.31	188.23	185.23	130.75	55.27
Arcus1	23583.13	23319.48	22853.61	14671.81	37.79
Arcus2	60833.33	58734.52	52557.10	33800.51	44.44
Arcus3	104325.77	71943.44	79633.43	50044.05	52.03
Arcus4	118465.36	83264.57	75710.20	49218.96	58.45
Arcus5	47478.62	32736.19	42252.11	19852.23	58.19
Arcus6	88318.83	64108.26	91380.01	48490.48	45.10
Arcus7	150535.22	110904.25	158385.30	76396.39	49.25
Arcus8	128955.78	113734.13	115206.82	69053.40	46.45
Arcus9	96583.47	94776.86	89205.85	63909.82	33.83
Arcus10	273474.48	239390.19	224071.42	150486.71	44.97
Arcus11	467006.78	392005.92	385818.74	294429.05	36.95
Arcus12	324645.25	313209.76	312184.75	233102.25	28.20

다. TCoA의 경우, 진화하는 동안 개체와 그들의 환경은 고정되어 있다. 이러한 제약은 개체들이 다양한 환경과 상호작용할 수 있는 기회를 방해할 수 있다. TCoA와 달리, LCoA에서는 진화하는 동안 개체의 환경 변화를 허용하나 한 세대 동안은 상대 모집단의 하나의 개체가 환경 개체로 작용한다. 이는 개체와 환경의 다양한 결합을 방해한다. 이러한 단점을 극복하고자, 본 연구에서 제안한 PCoA에서는 지역적 상호작용을 통한 진화전략을 사용하였다. 제안한 진화전략은 개체들이 지역적 정보에 의존하여 진화한다는 자연계의 진화개념을 따른 것이다. 이러한 진화는 빠른 지역적 수렴성과 함께 모집단 전체의 다양성을 적절히 유지시킨다(Davidor, 1991). 그리고 PCoA에서 이웃진화와 함께 안정상태 유전알고리듬의 사용은 생산된 좋은 개체가 재생산에 빨리 참여할 수 있어 탐색 효율과 지역적 수렴성을 높일 수 있다.

PCoA의 CPU시간은 Thomopoulos문제, Kim문제, 그리고 Arcus 문제에 대하여 평균적으로 각각 7, 53, 252초 정도 소요되었다. LCoA는 PCoA보다 더 적은 수의 환경개체를 선택해야 하지만, 하나의 환경개체를 선택하는데 있어 PCoA에 비해 더 많은 연산이 필요하다. 그러므로, LCoA와 PCoA의 CPU시간은 거의 차이가 없었다. 반면, TCoA는 환경개체를 선택하기 위한 별다른 절차가 필요없기 때문에 다른 두 방법에 비해 약 3.3%정도 적은 시간이 소요되었다.

7. 결 론

본 연구에서는 MMUL의 작업할당문제와 투입순서문제를 동시에 해결할 수 있는 새로운 공진화 알고리듬을 제안하였다. 공진화 알고리듬에서는 모집단의 다양성 유지와 좋은 환경의 선택이 탐색 성능을 향상시킬 수 있는 중요한 요소라고 알려져 있다. 이를 위하여 본 연구에서는 지역적인 진화전략을 적용하였으며, 환경개체를 선택하는 방법과 적응도 평가방법을 개발하였다. 또한 두 문제의 특성에 맞는 효율적인 유전표현과 유전연산자를 사용하였다.

실험 결과, 제안된 공진화 알고리듬은 고전적인 계층적 방법뿐만 아니라 다른 두 공진화 알고리듬에 비하여 우수한 성능을 보였다. 제안된 공진화 알고리듬은 기존 알고리듬과 비교하여 모집단의 다양성 유지와 함께 해공간의 효율적 탐색을 가능하게 한다.

제안된 공진화 알고리듬의 기본 구조는 진화 알고리듬이 갖는 적용의 유연성에 의해 목적이 변경되거나 제약이 추가되는 문제뿐만 아니라, 복잡하고 동적인 여러 형태의 문제에 쉽게 적용 가능하다. 특히, 4장에서 제안된 새로운 유전표현과 연산자는 주어진 투입순서에 대하여 MMUL의 작업할당문제를 해결하는데 사용될 수 있으며, 또한 5장에서 제시된 유전표현과

연산자는 고정된 작업할당하에서 MMUL의 적정 투입순서를 결정하는 문제에 각각 독립적으로 적용 가능하다.

참고문헌

Arcus, A. L. (1963), An analysis of a computer method of sequencing assembly line operations, Ph.D. dissertation, University of California.

Bard, J. F., Dar-El, E. M. and Shtub, A. (1992), An analytic framework for sequencing mixed model assembly lines, *International Journal of Production Research*, 30, 35-48.

Chakravarty, A. K. and Shtub, A. (1985), Balancing mixed model lines with in-process inventory, *Management Science*, 31, 1161-1174.

Dar-El, E. M. and Navidi, A. (1981), A mixed-model sequencing application, *International Journal of Production Research*, 19, 69-84.

Davidor, Y. (1991), A naturally occurring niche and species phenomenon: The model and first results, *Proceedings 4th International on Conference Genetic Algorithms*, 257-263.

Davis, L. (1985), Applying adaptive algorithms to epistatic domains, *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, 162-164.

Falkenauer, E. (1994), A new representation and operators for genetic algorithms applied to grouping problems, *Evolutionary Computation*, 2, 123-144.

Hyun, C. J., Kim, Y. and Kim, Y. K. (1998), A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines, *Computers & Operations Research*, 25, 675-690.

Kilbridge, M. and Wester, L. (1963), The assembly line model-mix sequencing problem, *Proceeding of the 3rd International Conference on Operations Research*, Oslo, Paris, 247-260.

Kim, Y. K., Hyun, C. J. and Kim, Y. (1996), Sequencing in mixed model assembly lines: A genetic algorithm approach, *Computers & Operations Research*, 23, 1131-1145.

Koza, J. R. (1992), *Genetic Programming*, The MIT Press, Cambridge, Massachusetts.

Kubiak, W. (1993), Minimizing variation of production rates in just-in-time systems: A survey, *European Journal of Operational Research*, 66, 259-271.

Macaskill, J. L. C. (1972), Production-line balances for mixed-model lines, *Management Science*, 19, 423-434.

Maher, M. L. and Poon, J. (1996), Modelling design exploration as co-evolution, *Microcomputers in Civil Engineering*, 11, 195-210.

Miltenburg, G. J. and Goldstein, T. (1991), Developing production schedules which balance part usage and smooth production loads for just-in-time production systems, *Naval Research Logistics*, 38, 893-901.

Monden, Y. (1993), Toyota Production System, 2nd Edn, *Institute of Industrial Engineers*, Norcross, Georgia.

Moriarty, D. E. and Miikkulainen, R. (1997), Forming neural networks through efficient and adaptive coevolution, *Evolutionary Computation*, 5, 373-399.

Pagie, L. and Hogeweg, P .(1997), Evolutionary Consequences of Coevolving Targets, *Evolutionary Computation*, 5, 401-418.

Potter, M. A. (1997), The design and analysis of a computational model of cooperative coevolution, Ph. D. dissertation, George Mason University.

Roberts, S. D. and Villa, C. D. (1970), On a multiproduct assembly line balancing problem, *AIIE Transactions*, 2, 361-364.

Rosin, C. D. and Belew, R. K. (1997), New methods for competitive coevolution, *Evolutionary Computation*, 5, 1-29.

Sarma, J. and De Jong, K. (1996), An analysis of the effects of neighborhood size and shape on local selection algorithm, *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, 236-244.

Sparling, D. and Miltenburg, J. (1998), The mixed-model U-line balancing problem, *International Journal of Production Research*, 36, 485-501.

Syswerda, G. (1991), A study of reproduction in generational and steady-state genetic algorithms, *Foundations of Genetic Algorithms*, edited by Gregory J.E. Rawlins, 94-101.

Thomopoulos, N. T. (1967), Line balancing-sequencing for mixed-model assembly, *Management Science*, 14, 59-75.

Thomopoulos, N. T. (1970), Mixed model line balancing with smoothed station assignment, *Management Science*, 16, 593-603.

Whitley, D. (1989), The Genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best, *Proceedings of the 3rd International Conference on Genentic Algorithms and their Application*, 116-121.

Yano, C. A. and Rachamadugu, R. (1991), Sequencing to minimize work overload in assembly lines with product options, *Management Science*, 37, 572-586.

von Laszewski, G. (1991), Intelligent structural operators for k-way group partitioning problem, *Proceedings 4th International on Conference Genetic Algorithms*, 45-52.