

U라인에서의 작업관련성을 고려한
작업부하 평활화에 관한 연구*
(A Study on Workload Smoothness Considering Work
Relatedness In the U-Line)

김우열, 김용주, 김동묵**

Abstract

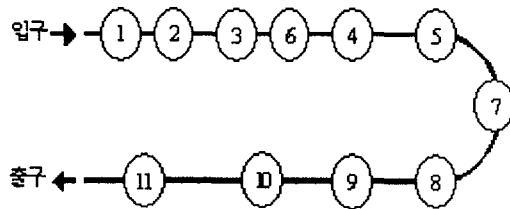
In just-in-time production systems, U-shaped production lines rather than traditional straight lines are often adopted since they have some advantages. The advantages of U-lines over straight lines are that the workstations required can be reduced and the necessary number of workers can be easily adjusted when the demand rates are changed. In this paper, we present a new genetic algorithm(GA) to minimize the number of workstations primarily and improve the work relatedness secondarily in the U-line production systems. Also, a new heuristic method is presented according to the work related factors and characteristics of U-line balancing. Some major aspects of the proposed GA are discussed, with emphasis on representation, decoding and evaluation function. Extensive experiments are carried out on well-known test-bed problems in the literature to verify the performance of our algorithm. The computational results show that our algorithm is a promising alternative to existing heuristics.

* 이 연구는 동신대학교 교내학술연구비(98-0200-09-01-3) 지원으로 수행되었음

** 동신대학교 산업공학과

1. 서 론

U자형 생산라인(U라인)은 <그림 1>과 같이 생산라인이 U자 형태로 구부러져서 제품이 들어가는 부분과 나오는 부분이 만나도록 설계된다. 이러한 형태의 생산라인에서는 가장 숙련된 작업자가 라인의 입·출구의 작업을 수행하면서 라인을 관리한다. U라인은 JIT(Just-In-Time)생산 시스템에서 자주 사용되고 있는데, 이는 U라인이 재공품 수량을 일정하게 유지할 수 있다는 장점과 작업자가 서로 근접하여 작업을 함으로 가시성이 높아 작업자 간의 작업 불균형을 쉽게 파악할 수 있어서 라인관리가 용이하다는 장점을 갖고 있기 때문이다[11].



<그림 1> U라인의 예

본 연구에서는 U라인의 작업부하 평활화 즉 라인밸런싱(U-line balancing: ULB) 문제를 다룬다. 라인밸런싱 문제는 라인에 부과된 여러 제약들을 어기지 않고 하나 또는 그 이상의 목적들이 최적이 되도록 작업들을 작업장(작업자)에 할당하는 문제이다. 본 연구에서 사이클 타임이 주어질 때 최소의 작업장의 수를 결정하면서 동시에 작업장에 할당된 작업의 관련성을 최대화하는 U라인 밸런싱 문제를 다루었다.

전통적인 직선 라인 밸런싱 기법에서는 선행 작업이 모두 할당된 작업중에서 하나의 작업을 선택하여 작업장에 할당하는 과정을 반복한다. 반면, U라인에서는 모든 선행 작업들이 할당된 작업뿐만 아니라 모든 후행 작업들이 할당된 작업까지를 할당 가능 작업으로 둔다. 따라서 U라인은 직선라인에 비해 작업할당에 대한 유연성이 높고, 이로 인해 좋은 작업할당이 가능하며, 요구되는 작업장의 수는 직선라인의 그것보다 크지 않다[10]. 또한, 작업장 수를 최소화하는 라인밸런싱 문제는 많은 대안해가 있음에도 불구하고, 작업관련성을 고려하여 해를 구하는 기법에 관한 연구는 아직 미미한 실정이다. 실제 조립라인에서는 가능한 범위에서 관련된 작업들을 동일한 작업자에 할당하는 것이 바람직하며, 이러한 작업편성은 작업자의 작업능률을 향상시킬 수 있다.

ULB문제에 관한 연구는 1994년에 Miltenburg와 Wijngaard[10]에 의해 처음 시작되었다. 그들은 동적계획 모형과 발견적 기법을 제시하였으며, Hwang et al.[6]은 시뮬레이티드 어닐링(simulated annealing)을 이용하여 라인밸런싱 하였다.

또한 Urban[12]은 정수계획모형을, 김여근 외의 연구[2]에서는 수리 모형과 분지한계법을 제시하였다. 이들 연구에서 라인 밸런싱의 목적은 작업장 수의 최소화로 두었다.

유전알고리즘(genetic algorithm)을 이용한 라인밸런싱에 관한 연구로는, Anderson and Ferris[5], Leu et al.[9], Kim et al.[7,8], 김여근 외의 연구[1] 등이 있다. 그러나 이들 연구는 직선라인의 라인밸런싱 문제를 다루고 있다. 유전알고리즘을 이용한 U라인의 밸런싱에 관한 연구는 김동욱[3]의 연구가 있으며, 작업장 수가 주어질 때 작업장간 작업량의

평활성(smoothness)을 최대화하는 문제와 복수 개의 U라인에서의 라인밸런싱 문제를 다루었다.

작업관련성을 고려한 라인밸런싱에 관한 연구로는 Agrawal[4]의 연구가 있는데, 미할당 후행작업이 없는 작업들을 고려해야 하는 U라인에는 적용하기에는 미흡한 점이 있다.

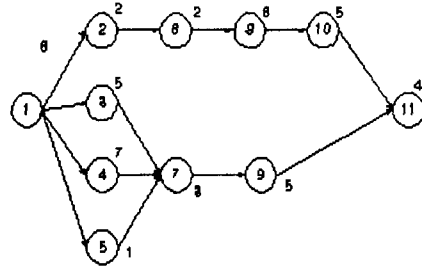
본 연구에서는 유전알고리즘을 이용하여 ULB문제의 해를 탐색하였으며, 특히 유전알고리즘의 개체 표현방법으로 랜덤키(random key) 표현을 사용하였다. 랜덤키 표현은 TSP(traveling salesman problem)와 같은 문제에서 개체 내 인자 간의 상대적 순서가 해의 중요한 정보로서 활용될 때 효율적인 것으로 알려져 있다[13].

본 연구의 목적은 U라인에서 사이클타임이 주어질 때 작업관련성을 고려하면서 작업장 수를 최소화하는 라인밸런싱 문제를 해결하기 위한 유전알고리즘을 개발하는데 있다. U라인에서의 작업관련성을 고려한 개체의 평가함수가 제시되었으며, 개체의 형질을 효과적으로 자손에 유전할 수 있고 유전 연산이 용이한 개체의 표현방법과 개체의 해석방법이 제시되었다. 컴퓨터 실험을 통하여 개발한 알고리즘의 성능을 보였다.

2. 작업관련성을 고려한 U라인 밸런싱

라인에 편성될 작업들 간에는 기술적인 제약으로 인하여 선후행 관계(precedence relationship)가 존재하고, 이는 <그림 2>와 같은 선행공정도(precedence diagram)로 표현된다. 라인밸런싱은 선후행 관계를 만족하면서 작업장에 작업을 할당하는 것으로, 한 작업장에 할당된 작업시간이 사이클타임을 넘지 않

도록 해야 한다.



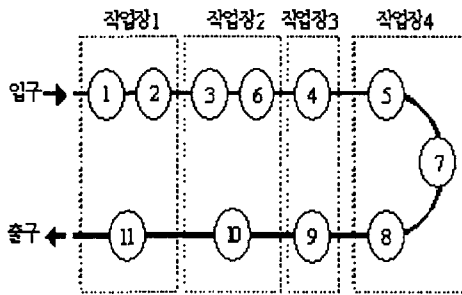
<그림 2> 선행공정도

기존 직선라인을 위한 발전적 기법에서는 작업장을 순차적으로 하나씩 추가하면서 작업을 하나씩 할당한다. 선후행 관계를 만족시키기 위하여, 미할당 선행 작업이 없고 작업시간이 작업장의 여유시간(사이클타임 - 할당된 총 작업시간)보다 크지 않은 작업들을 할당 가능작업으로 두고, 이들 중 하나를 선택하여 작업장에 할당한다. 다음은 본 연구에서 ULB 설명에 사용된 기호와 직선라인에 적용되는 발전적 기법의 절차이다.

-
- n : 작업장 수
 - M : 작업 수
 - CT : 사이클 타임
 - I : 작업의 집합 $I=1, \dots, i, \dots, M$.
 - J : 작업장의 집합 $J=1, \dots, j, \dots, n$.
 - t_i : 작업 i 의 작업시간.
 - T_j : 작업장 j 의 작업량, 즉 작업장 j 에 할당된 총 작업시간.
 - ST_j : 작업장 j 의 여유시간 $ST_j=CT-T_j$.
 - P_i : 작업 i 의 미할당 선행 작업들의 집합.
 - S_i : 작업 i 의 미할당 후행 작업들의 집합.
 - R : 할당가능 작업 집합.
 - L_j : 작업장 j 에 포함된 선행공정도에서 연결된 작업 집합의 수.
-

- 단계 1: $j=1$;
- 단계 2: 새로운 작업장 j 를 생성하고, 작업장 j 의 여유시간 ST_j 를 CT 로 둔다.
- 단계 3: 할당가능한 작업집합 $R=\{i \in I \mid P_i=\emptyset \text{ and } t_i \leq ST_j\}$ 을 구한다.
- 단계 4: $R=\emptyset$ 이면, $j=j+1$ 로 두고 단계 2로 간다.
- 단계 5: R 에서 임의 혹은 특정 기준에 따라 작업 $i^* \in R$ 을 선택하여 작업장 j 에 할당하고, $ST_j=ST_j-t_{i^*}$ 로 둔다.
- 단계 6: 모든 작업이 할당되었으면, 작업장의 수 n 을 j 로 두고 종료한다. 그렇지 않으면, 단계 3으로 간다.

U라인은 그림 1에 나타난 바와 같이 라인이 구부러져 있어서, 입구 쪽에서 들어가는 방향(순방향)의 작업과 출구 쪽으로 나오는 방향(역방향)의 작업을 한 작업장에서 모두 수행할 수 있다. 그림 3은 그림 2의 선후행관계를 갖는 ULB 문제의 해이다. 순방향의 작업 1, 2와 역방향의 작업 11이 모두 작업장 1에 할당되어 있다.



<그림 3> U라인의 작업할당 예

U라인에서는 직선라인과는 달리, 작업장에 할당 가능한 작업들의 집합에는 미할당 선행 작업이 없는

작업과 미할당 후행 작업이 없는 작업이 모두 포함되어야 한다. 예를 들면 첫 번째 작업장을 생성하는 시점에서 할당가능 집합에는 선행 작업이 없는 작업 1과 후행작업이 없는 작업 11이 모두 포함된다. 직선 라인에 적용된 발견적 기법은 약간의 수정만으로 U라인에 적용될 수 있는데, 단계 3에서 할당가능 집합을 생성하는 과정에서 미할당 후행작업이 없는 작업들에 포함하면 된다. 단계 3을 ULB에 적합하도록 바꾸면 다음과 같다.

- 단계 3 : 할당가능 작업 집합 $R=\{i \in I \mid (P_i=\emptyset \text{ or } S_i=\emptyset) \text{ and } t_i \leq ST_j\}$ 을 구한다.

발견적 기법 단계 5에서는 할당가능한 작업집합 R 의 작업 중 하나를 선택하게 되는데, 이 과정에서 어떤 작업을 선택하는가에 따라 라인밸런싱의 해가 결정된다.

직선 라인의 경우 MAXDUR(Maximum Duration), RPW(Maximum Ranked Positional Weight) 등의 할당 규칙들이 개발되었다. 본 연구에서는 유전알고리즘을 이용하여 단계 5에서 적용되는 다양한 형태의 할당 순서를 생성하여 최적해를 탐색하는 방법을 제시하고자 한다.

관련된 작업들을 가능한 범위에서 동일한 작업장에 할당하는 것은 서로 연관된 작업들이 여러 작업장에 분산된 것보다 훨씬 바람직하다. 이것은 작업자들의 작업 효율을 높일 수 있을 뿐만 아니라, 조립라인의 설계자에게 설비와 작업장을 배치하는데 있어서 보다 큰 유연성을 제공할 수 있다. 한 작업장에 할당된 작업들 간의 관련성을 평가하기 위하여 본 연구에서는 다음과 같은 척도를 사용하였다.

Index of Work Relationship(IWR)

$$= n/\sum(L_j - u_j),$$

$u_j : L_j > 1$ 이고 작업장 j 에 순방향 작업과 역방향 작업이 공존하면 1, 그렇지 않으면 0.

제시된 IWR은 각 작업장에 포함된 연결된 작업 집합의 수가 적을수록, 또한 양방향의 작업을 모두 하나의 작업장에서 수행하면 높은 값을 갖도록 제안되었다. 이는 양방향의 작업을 모두 수행한 경우, 작업관련성이 손상됨을 보상해 주는 역할을 한다.

3. U라인 라인밸런싱을 위한 유전 알고리즘

3.1 표현

라인밸런싱 문제의 해는 작업들을 작업장에 할당하는 순서에 의해 결정될 수 있고, 특히 작업들 간의 상대적인 할당순서(relative order)는 문제의 중요한 정보이다. 본 연구에서는 랜덤키 표현을 사용하여 ULB문제의 해를 개체로 표현하였다. 랜덤키 표현에서는 개체의 인자 위치가 작업번호에 해당되며, 인자값은 0부터 1사이의 실수로서 개체의 할당순서를 결정하는 값으로 사용된다. 랜덤키 표현은 유전 연산자의 개발이 용이하고, 특히 ULB문제의 유전정보인 작업들 간의 상대적 할당순서를 크게 파괴시키지 않고 자손에 유전하기 쉽다. 또한, 개체의 해석에 소요되는 계산시간 면에서도 유리하다. 다음은 <그림 2>의 선행공정도를 갖는 ULB 문제의 해를 랜덤키 표현으로 나타낸 예이다.

(0.31, 0.27, 0.14, 0.95, 0.78, 0.62,
0.44, 0.73, 0.90, 0.48, 0.53)

랜덤키 형태로 표현된 개체를 ULB 해로 해석하는 방법은 앞에서 제시한 발견적 기법을 조금 수정하면 쉽게 만들어진다. 발견적 기법의 단계 5에서 작업을 선택하는 기준으로 개체의 인자 값을 사용하면 된다. 할당가능한 작업집합에서 가장 큰 값을 갖는 인자에 해당되는 작업을 작업장에 할당한다. 따라서 개체 해석 방법 단계 5는 다음과 같이 수정되었다.

단계 5 : R에서 개체의 인자 값이 가장 큰 작업 $i^* \in R$ 을 선택하여 작업장 j 에 할당하고, $ST_j = ST_j - t_{i^*}$ 로 둔다.

위의 예에서 작업 1이 이미 작업장에 할당되어 있고 사이클 타임이 15로 주어진다면, 할당 가능 작업 집합은 2, 3, 4, 5, 11 이다. 이들 중 작업할당 우선순위가 가장 높은 것은 인자 값 0.95를 갖는 작업 4이다.

3.2 평가함수와 선별

ULB 문제에서는 작업장 수와 작업관련성이 모두 개체를 평가하는데 반영되어야 한다. 따라서, 작업장 수를 기본 척도로 두고, 작업관련성을 부가적 척도로 두는 평가함수를 본 연구에서 새롭게 개발하였다. 개체 G에 대한 평가함수 E(G)는 다음과 같다.

$$E(G) = n + (1 - IWR)$$

ULB문제는 최소화 문제이므로 E(G) 값이 적을수록 좋은 개체로 평가된다. 평가함수의 두 번째 항은 작업관련성을 나타내는 IWR이 0부터 1사이의 실수이고 클수록 좋은 해로 평가되어야 하므로, 1에 IWR을 뺀 값으로 두었다.

선별(selection)은 개체의 적응도(fitness)에 기초하여 다음 세대에 생존할 개체들을 선택하는 과정이다. 본 연구에서 사용한 선별은 확률바퀴 선별(roulette wheel selection)을 사용하였다.

3.3 유전연산자

유전알고리즘에서 유전연산자(genetic operator)는 개체 내에 포함된 해의 결정적인 형질(정보)를 추출하고 다음 세대의 자손에 전파하는 역할을 한다.

또한, 다양한 형질을 생산하는 역할을 한다. 본 연구에서는 교차 유전연산자로 일점교차(one-point crossover)를 사용하였다. 본 연구에서 도입한 랜덤

키 표현방법에서는 개체의 인자 값이 작업 할당 순서를 결정하기 위한 기준으로 사용되므로, 일점교차에 의해 생산된 자손 개체는 항상 가능해로 해석될 수 있다. <그림 4>의 (a)는 두 부모개체 P1과 P2를 일점교차 하여 새로운 자손 개체를 만드는 예이다.

그리고, 돌연변이 방법으로는 돌연변이율에 따라 임의의 개체 인자를 선택하고 0부터 1사이의 임의의 실수로 바꾸는 방법을 사용하였다. 돌연변이 된 인자는 작업 할당의 우선 순위에 변화를 주어 새로운 형질을 생산하게 된다. 이 방법 또한 생산된 개체를 항상 가능해로 유지한다. <그림 4>의 (b)는 제안된 돌연변이의 예이다.

P1=	0.31	0.27	0.14	0.95	0.78	0.62	0.44	0.73	0.90	0.48	0.53
						↓	↓	↓	↓	↓	↓
P2=	0.53	0.36	0.51	0.13	0.22	0.15	0.92	0.65	0.21	0.16	0.88
O1=	0.31	0.27	0.14	0.95	0.78	0.15	0.92	0.65	0.21	0.16	0.88
O2=	0.53	0.36	0.51	0.13	0.22	0.62	0.44	0.73	0.90	0.48	0.53

(a) 교차(crossover)

P1=	0.31	0.27	0.14	0.95	0.78	0.62	0.44	0.73	0.90	0.48	0.53
						↓					
O1=	0.31	0.27	0.14	0.95	0.78	0.15	0.44	0.73	0.09	0.48	0.53

(b) 돌연변이(mutation)

<그림 4> 유전연산자의 예

<표 1> 실험 결과

problem	cycle time	proposed GA				heuristics			
		best	worst	mean	s.d.	best	worst	mean	s.d.
S30	25	14.300	14.333	14.307	0.0141	14.645	17.839	16.685	0.6738
S30	27	13.315	13.381	13.358	0.0309	13.677	17.710	15.615	0.5551
S30	30	12.200	12.250	12.215	0.0242	12.484	15.677	13.667	0.4663
S30	36	10.166	10.167	10.167	0.0000	10.387	12.613	10.969	0.5032
S30	41	8.200	8.467	8.357	0.0797	8.452	10.677	9.534	0.1714
S30	54	7.125	7.222	7.183	0.0502	7.291	8.516	7.454	0.0601
S30	75	5.166	5.286	5.214	0.0615	5.194	5.548	5.351	0.0482
K45	57	10.375	10.444	10.421	0.0230	10.383	12.468	11.276	0.5162
K45	79	7.500	7.611	7.563	0.0350	7.340	8.936	8.600	0.1314
K45	92	6.571	6.667	6.618	0.0338	6.362	7.915	7.542	0.1324
K45	110	6.333	6.455	6.415	0.0395	6.234	6.894	6.486	0.1316
K45	138	4.333	4.500	4.469	0.0562	4.170	5.851	5.409	0.1842
K45	184	3.400	3.500	3.490	0.0316	3.128	4.787	4.278	0.3125
T70	176	21.533	21.571	21.557	0.0112	21.681	24.944	23.041	0.5210
T70	364	10.642	10.677	10.661	0.0113	10.417	11.972	10.916	0.3813
T70	410	9.625	9.690	9.667	0.0207	9.347	10.931	9.761	0.1703
T70	468	8.680	8.704	8.691	0.0088	8.361	9.847	8.722	0.1050
T70	527	7.666	7.708	7.691	0.0130	7.333	8.889	7.713	0.1442
A83	5048	16.200	16.273	16.241	0.0267	16.247	18.353	17.133	0.4093
A83	5853	14.176	14.222	14.181	0.0145	14.200	15.388	14.392	0.2783
A83	6842	12.200	12.250	12.230	0.0258	12.177	13.400	12.934	0.4741
A83	7571	11.153	11.214	11.172	0.0292	11.153	12.247	11.250	0.0293
A83	8412	10.090	10.167	10.136	0.0391	10.129	10.341	10.235	0.0265
A83	8898	9.100	9.182	9.108	0.0259	9.118	10.317	9.232	0.1355
A83	10816	8.111	8.200	8.138	0.0429	8.118	8.306	8.193	0.0262
A111	5755	27.625	27.640	27.633	0.0042	27.646	32.752	29.440	0.7062
A111	7969	19.786	20.697	20.507	0.3799	20.558	23.920	21.431	0.5583
A111	8847	18.700	18.710	18.708	0.0033	18.566	21.814	19.327	0.5553
A111	9018	17.673	17.721	17.707	0.0140	17.522	20.912	19.031	0.5144
A111	10027	16.709	16.719	16.717	0.0036	16.504	18.814	16.899	0.3845
A111	10743	15.666	15.700	15.691	0.0095	15.425	17.770	15.844	0.3509
A111	11378	14.641	14.682	14.663	0.0123	14.345	15.920	14.695	0.2538
A111	17067	9.666	9.727	9.694	0.0233	9.283	10.717	9.798	0.4707
B148	193	27.630	27.663	27.651	0.0100	27.527	30.607	28.805	0.4485
B148	238	22.620	22.662	22.642	0.0118	22.413	24.580	23.422	0.3379
B148	272	19.716	19.736	19.723	0.0071	19.440	21.593	20.510	0.1709
B148	289	18.660	18.690	18.681	0.0092	18.387	20.560	19.411	0.3005
B148	306	17.685	17.696	17.690	0.0036	17.360	18.587	18.341	0.3195
B148	323	16.709	16.738	16.724	0.0082	16.387	18.407	17.443	0.0989
B148	525	10.729	10.756	10.747	0.0083	10.260	11.440	10.394	0.1474

4. 실험 및 분석

제안한 유전알고리즘의 성능 분석을 위하여 발견적 기법과 비교 실험을 실시하였다. 비교된 발견적 기법은 임의의 해를 반복하여 생성하는 방법을

사용하였다. 실험 대상은 라인벨런싱의 벤치마크 문제로 널리 알려진 30, 45, 70, 83, 111, 148개 작업 문제[3]에 대해 다양한 사이클타임을 설정하여 실험하였다. 발견적 기법과 제안된 유전알고리즘 모두 30,000개의 해를 생성하여 가장 좋은 해를 구했으며, 이를 30회 반복 수행하여 얻은 값을<표 1>

에 나타내었다. 유전 알고리즘을 수행하기 위해서는 모집단의 크기, 교차율, 돌연변이율 등의 유전 파라미터를 결정해야 한다.

본 연구에서는 예비 실험을 통하여 적절한 유전 파라미터 값을 찾고, 모든 문제에 같은 값을 적용하여 실험하였다. 본 연구에서 사용한 유전파라미터 값으로는 모집단 크기를 30, 교차변이율을 90%, 돌연변이율을 1%를 사용하였다.

실험 결과 대부분의 경우, 제안된 알고리즘이 발전적 기법의 평균보다 좋은 해를 탐색하였다. 특히, 작업장 수에 있어서는 제안된 기법이 비교된 기법에 비해 항상 좋거나 같은 해를 구했으며, 작업관련성 또한 거의 대부분의 경우 좋은 해를 구하였다. 실험에서 제안된 알고리즘은 탐색 성능을 향상시키기 위한 파라미터 튜닝 등의 조치 없이 수행 되었음에도 불구하고 제안된 알고리즘은 비교된 기법에 비해 좋은 해를 효율적이고 체계적으로 탐색함을 알 수 있다.

5. 결 론

본 연구에서는 U라인에서 작업관련성을 고려한 작업 부하 평활화를 위한 유전알고리즘을 제시하였다. U라인의 특성에 맞는 효율적인 유전알고리즘의 개발을 위하여, 작업장에 할당된 작업들의 관련성을 고려한 평가함수를 개발하였으며, 작업장에 할당할 작업들의 우선 순위를 개체로 표현하기 위하여 랜덤키 표현 방법을 제안하였다. 랜덤키 표현 방법은 라인 밸런싱 문제의 중요한 정보인 작업 할당 순서를 적절히 표현할 수 있을 뿐만 아니라, 별도의 유전 연산자를 개발할 필요가 없이 흔히 사용

되는 1점 교차 방법을 사용할 수 있음을 보였다.

제안한 알고리즘의 성능을 평가하기 위하여 발전적 기법과의 비교 실험을 실시하였다. 실험결과, 제안한 알고리즘은 작업장 수의 측면에서 항상 좋거나 같은 해를 구했으며, 이는 지금까지 직선라인에서 알려진 최적해와 유사한 결과이다. 또한 작업관련성 측면에서의 성능도 비교된 기법보다 좋은 탐색 성능을 보였다. 따라서 제안된 알고리즘은 작업관련성을 고려한 U라인 밸런싱 문제에서 좋은 해를 체계적이고 효율적으로 탐색할 수 있음을 알 수 있다.

또한 제안된 알고리즘은 작업자의 이동시간을 고려한 경우나 특정제약이 추가된 U라인 밸런싱 문제에도 평가함수의 개발만으로 쉽게 적용할 수 있을 것으로 보인다.

참 고 문 헌

- [1] 김여근, 김용주, 김준혁, 박재승, "양면 조립라인 밸런싱을 위한 유전알고리즘," IE Interfaces, Vol. 12, No. 1, (1999), pp. 132-142.
- [2] 김여근, 김재윤, 김동목, 송원섭, "U라인 라인 밸런싱을 위한 분지한계법," 한국경영과학회지 23 권 12호, (1998), pp. 83-101
- [3] 김동목, "U형 생산 라인밸런싱을 위한 효율적인 기법", 전남대학교 박사학위 청구논문, (2000).
- [4] Agrawal, P. K., "The Related activity concept in assembly line balancing", Int. J. Prod. Res., Vol 23, (1985), pp. 403-421
- [5] Anderson, E. J. and Ferris, M. C., "Genetic Algorithms for Combinatorial Optimization:

- the Assembly Line Balancing Problem," ORSAJ. Computing, Vol. 6, (1994), pp. 161-173.
- [6] Hwang, H., Sun, J. U. and Yoon, T. H., "U-line Line Balancing with Simulated Annealing," Proceedings of the First ASIA-PACIFIC Decision Sciences Institute Conference, Hong Kong, (1996), pp.101-108.
- [7] Kim, Y. K., Kim, Y. J. and Cho, Y. K., "A Heuristic-Based Genetic Algorithm for Workload Smoothing in Assembly Lines," Computers and Operations Research, Vol.25, No.2,(1998), pp. 99-111.
- [8] Kim, Y. K., Kim, Y. J. and Kim, Y. H., "Genetic Algorithms for Assembly Line Balancing with Various Objectives," Computers and Industrial Engineering," No.30,(1996), pp. 397-409.
- [9] Leu, Y. Y., Matheson, L. A. and Rees, L. P., "Assembly Line balancing Using genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria," Decision Science, Vol. 25, (1994), pp. 581-606.
- [10] Miltenburg, G. J. and Wijngaard, J., "The U-line Line Balancing Problem," Management Science, Vol.40, (1994), pp.1378-1388.
- [11] Monden, Y., Toyota Production System (2nd Ed.), Industrial Engineering and Management Press, Institute of Industrial Engineers, Norcross, GA, (1993).
- [12] Urban, T. L., "Note. Optimal Balancing of U-Shaped Assembly Lines", Management Science, Vol. 44, No. 5, (1998), pp. 738-741.
- [13] Bean, J. C., Genetic algorithms and random keys for sequencing and optimization, ORSA Journal on Computing 6 (1994), pp. 154-160.