

Optimizing Assembly Line Balancing Problems with Soft Constraints

Seong-Hoon Choi*[†] · Geun-Cheol Lee**

*Depart. of Management Engineering, Sangmyung University

**College of Business Administration, Konkuk University

소프트 제약을 포함하는 조립라인 밸런싱 문제 최적화

최성훈*[†] · 이근철**

*상명대학교 경영공학과

**건국대학교 경영대학

In this study, we consider the assembly line balancing (ALB) problem which is known as an very important decision dealing with the optimal design of assembly lines. We consider ALB problems with soft constraints which are expected to be fulfilled, however they are not necessarily to be satisfied always and they are difficult to be presented in exact quantitative forms. In previous studies, most researches have dealt with hard constraints which should be satisfied at all time in ALB problems. In this study, we modify the mixed integer programming model of the problem introduced in the existing study where the problem was first considered. Based on the modified model, we propose a new algorithm using the genetic algorithm (GA). In the algorithm, new features like, a mixed initial population selection method composed of the random selection method and the elite solutions of the simple ALB problem, a fitness evaluation method based on achievement ratio are applied. In addition, we select the genetic operators and parameters which are appropriate for the soft assignment constraints through the preliminary tests. From the results of the computational experiments, it is shown that the proposed algorithm generated the solutions with the high achievement ratio of the soft constraints.

Keywords : Generalized Assembly Line Balancing Problem, Soft Constraints, Genetic Algorithm

1. 서 론

고객의 다양한 니즈를 수용하기 위해 다품종 생산이 대세인 지금도 컨베이어 기반의 조립라인은 흔히 볼 수 있는 프로세스이다[13]. 조립라인은 많은 자본이 투입되는 자본집약 성격의 프로세스이므로 최대한 효율이 높도록 설계해야 한다. 조립라인 밸런싱 문제(assembly line

balancing(ALB) 문제는 조립라인의 최적 설계를 다루는 중요한 의사결정 사항으로 50년 이상의 수많은 연구에도 불구하고 여전히 실무자와 연구자의 큰 관심을 끌어오고 있다[3, 22, 25].

ALB 문제를 풀기 위해서는 먼저 데이터를 준비해야 한다. 제품 생산을 위해 필요한 전체 작업을 m 개의 요소 작업(task)으로 구분하고, 각 요소작업 i 의 표준시간, t_i 를 설정한다[7]. 그리고 요소작업들 사이에는 작업순서가 존재하므로 이를 파악하여야 한다. 작업순서 제약은 흔히 선행 제약(precedence constraint)으로 불리며 선후행 관계도로 표현된다.

위의 과정을 통해 데이터가 준비되면, 작업장(workstation) 간의 작업시간이 최대한 균형을 이룰 수 있도록 m 개의 요소작업을 n 개의 작업장에 할당한다(단, $n \leq m$). 각각의 작업장에서 작업을 수행하는데 있어서 최대로 허용되는 시간은 모든 작업장에 대해 CT 로 동일하며, 이 최대허용시간은 흔히 라인 사이클 타임으로 불린다[21].

ALB 문제는 목적에 따라 4가지 문제 유형으로 구분될 수 있다[22]. ALBP-1은 n 이 주어지고, CT 를 최소화하는 문제이고, ALBP-2는 주어진 사이클 타임에 대하여 작업장의 수를 최소로 하는 문제이다. 이들 두 문제는 전형적인 ALB 문제이다. ALBP-3은 n 과 CT 를 동시에 최소화 하는 문제를 다룬다. 끝으로 ALBP-4는 주어진 n 과 CT 의 조합을 만족하는 가능한 라인 편성이 존재하는지를 확인하는 문제이다. 4가지 ALB 문제 유형에 대한 목적함수는 (1)과 같이 표현될 수 있다. 여기서, α 와 β 는 문제 유형에 따른 가중치이다. 예를 들어 ALBP-2의 경우에는 α 와 β 의 값은 각각 '0'과 '양수'로 설정된다. 그리고 $N = \{1, 2, \dots, n\}$ 은 라인을 구성하는 작업장 집합이다. 요소작업 λ 는 가상의 최종 작업으로 $t_\lambda = 0$ 이다. 만일 $x_{\lambda g} = 1$ 이면, 작업장의 수가 g 개임을 나타낸다[19].

어떤 라인 편성안이 ALB 문제의 가능해가 되려면 기본적으로 다음 세 가지 제약 사항을 만족해야 한다. 제1 제약은 각 요소작업은 하나의 작업장에만 할당되어야 하는 '단일 작업장 할당제약'이다. 제2 제약은 특정 요소작업이 수행되려면 그것보다 선행되는 모든 요소작업이 완료되어야 하는 '선행제약'이다. 마지막으로 제3 제약은 특정 작업장의 요소작업들의 작업시간 합은 라인 사이클 타임 이하이어야 하는 '사이클 타임 제약'이다. 즉, ALB 문제는 위의 세 가지 제약을 충족시키면서 식 (1)을 추구하는 문제이다. 이들 제약을 반영하여 정수계획 모델을 작성하면 다음과 같다[5]. 식 (2)~식 (4)는 각각 제1~3 제약을 모델링 한 것이다.

$$\text{Min. } \alpha CT + \beta \sum_{g \in N} g x_{\lambda g} \quad (1)$$

Subject to

$$\sum_{g \in N} x_{ig} = 1, \quad \forall i \in M \quad (2)$$

$$\sum_{g \in N} g x_{ig} \leq \sum_{g \in N} g x_{jg}, \quad \forall (i, j) \in P \quad (3)$$

$$\sum_{i \in M} t_i x_{ig} \leq CT, \quad \forall g \in N \quad (4)$$

여기에서 사용된 기호와 변수는 아래와 같이 정의된다.

기호 및 변수 1

$M = \{1, 2, \dots, m\}$: 라인을 구성하는 작업장에 할당해야 할 요소작업의 집합

$P = \{(i, j) | i \text{는 } j \text{의 직접 선행작업, 그리고 } i, j \in M\}$: 직접 선행작업 집합

t_i : 요소작업 i 의 작업시간($i \in M$)

$x_{ig} = 0$ or 1 : 요소작업 i 가 작업장 g 에 할당되면, 1, 그렇지 않으면 0($i \in M, g \in N$)

CT : 라인 사이클 타임(ALBP-2와 ALBP-4인 경우에는 상수)

지금까지 기술한 ALB 문제는 단순한 형태의 조립 라인 편성 문제(simple assembly line balancing(SALB) 문제)이다. 현실 문제에서는 위의 세 가지에 추가하여 더 많은 할당제약들이 추가로 발생한다[1]. 예를 들어 어떤 요소작업들은 작업에 필요한 설비가 설치되어 있는 특정 작업장에만 할당될 수 있다[7, 19]. 특정 요소작업들은 작업의 특성상 동일한 작업장에 배치되어야 한다. 이와 반대로 특정 요소작업들은 서로 방해하거나 작업 위치가 다르기 때문에 동일한 작업장에 할당될 수 없다. SALB 문제에 이와 같은 제약들을 추가로 반영한 문제가 일반 조립 라인 편성 문제(generalized assembly line balancing(GALB) 문제)이다. 본 연구에서는 다양한 할당제약을 포함하는 GALB 문제를 다루기로 한다.

할당제약들 중에는 반드시 만족되어야 하는 것들이 있는 반면에, 어떤 제약들은 정량적으로 정확히 표현하기는 어렵지만 되도록 충족되었으면 하는 것들이 존재한다. 예를 들어 '요소작업 i 와 j 는 가능하면 같은 작업장에 배치하지 않는 것이 바람직하다', '요소작업 k 는 작업장 g 또는 h 에 우선하여 할당하는 라인 편성안이 바람직'한데, 작업장 g 가 h 보다 우선순위가 높다'.

Choi[7]는 최초로 반드시 만족되어야 하는 엄격한 제약을 '고정 제약(rigid constraints)', 되도록 충족되었으면 하는 제약을 '유연 제약(mild constraints)'으로 표현하고 이들 제약들을 포함하는 GABLP에 대한 혼합정수계획 모델을 제시하였다. 그런데 '유연'이라는 용어는 기존의 유연 조립라인(flexible assembly line, FAL)[4, 9, 15, 21, 24]과 혼동을 줄 수 있으므로 본 논문에서는 좀 더 적합한 용어를 제시하고자 한다. 참고로 기존에 FAL은 다수의 제품이 단일 방향의 흐름만을 허용하는 작업장에서 조립되는 생산 시스템을 가리키는 용어로 많이 사용되어 왔다. 본 논문에서는 Kendall[12]이 선형계획 관련 연구에서 Choi[7]와 유사한 용도로 처음 사용한 hard and soft constraints 용어를 사용할 것을 제안한다. 즉 고정 제약과 유연 제약을 각각 '하드 제약'과 '소프트 제약'으로 부르기로 한다.

Choi[7]가 언급한 것처럼 소프트 제약을 하드 제약으로 처리하면 가능해가 존재하지 않을 수 있다. 또한 소프트 제약은 약간 포기함으로써 라인 효율이 높은 바람직한 해

를 구할 수도 있는 융통성을 부여해준다. 따라서 소프트 제약을 반영하는 GALB 문제에 대한 연구가 필요하다.

ALB 문제 해법은 전통적으로 최적해를 구하는 기법과 근사해를 구하는 기법으로 구분할 수 있다. 최적해 기법에는 혼합정수계획법, 동적계획법, 목표계획법, 네트워크 이론 적용 기법이 있다. 휴리스틱 기법으로 많은 방법이 제안되어 왔는데 최근에는 유전자 알고리즘(genetic algorithm, GA)을 중심으로 한 메타 휴리스틱 방법이 많이 보고되고 있다[3, 9, 10, 11, 15, 16, 23, 26].

Choi[7]의 연구에서는 예제 차원에서 10개의 요소작업으로 구성된 간단한 문제를 상용 최적해 패키지를 사용하여 해를 구했으나, 문제 크기가 조금만 커지더라도 NP-hard 문제의 특성상 적절한 시간 이내에 해를 구할 수 없다[2, 3]. 본 연구에서는 Choi[7]가 최초로 제시한 소프트 제약을 포함하는 혼합정수계획 모델의 오류를 보완하여 소개한다. 그리고 이를 바탕으로 개발한 GA 기반의 새로운 알고리즘을 제안한다. 참고로 하드 제약에 대한 혼합정수계획 모델은 기존에 많은 연구가 있으며 자세한 내용은 Choi[7]를 참고하기 바란다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 소프트 제약 혼합정수계획 모델을 다룬다. 제 3장은 제 2장의 혼합정수계획 모델로 표현된 GALB 문제의 해를 구하는 GA를 제안하고, 제 4장에서는 성능평가 실험 결과를 제시한다. 마지막으로 제 5장에서는 결론을 기술한다.

2. 소프트 할당제약 혼합정수계획 모델

본 장에서는 소프트 할당제약 사항을 좀 더 쉽고 유연하게 표현할 수 있는 방법론으로 Choi[7]가 최초로 제시한 관계행렬 표현법을 요약하여 소개하기로 한다. 관계행렬 표현법은 요소작업과 요소작업 사이에 존재하는 할당제약을 ‘요소작업-요소작업 관계행렬’로 표현하고 요소작업과 작업장 사이에 존재하는 할당제약을 ‘요소작업-작업장 관계행렬’로 표현하는 방법이다.

그리고 소프트 제약이 만족되지 못하는 정도에 따라 패널티를 부과하는 방식으로 개발된 Choi[7]의 혼합정수계획 모델에 존재하는 일부 오류를 보완하고 본 논문에서 제안하는 GA의 적용도 평가(fitness evaluation)에 적합하도록 변형된 모델을 제안한다.

2.1 요소작업-요소작업 할당제약

ALB-Research Group[1]이 조립라인 밸런싱 연구와 응용을 지원하기 위해 운영하고 있는 웹사이트에 언급된 다양한 할당제약들 중에서 Choi[7]는 다음 5가지를 ‘요소작업-요소작업 할당제약’으로 분류하였다.

- 친밀 요소작업군 : 작업이 함께 이루어져야 하는 것과 같은 이유로 특정 요소작업들이 동일한 작업장에 배정되어야 한다.
- 비친밀 요소작업군 : 특정 요소작업들은 서로 방해하거나 작업 위치가 다르기 때문에 같은 작업장에 할당될 수 없다.
- 최소 거리 : 어떤 요소작업은(예를 들어 페인트를 건조하기 위해) 다른 요소작업들과 최소한의 거리를 두고 배정되어야 한다.
- 최대 거리 : 어떤 요소작업은(예를 들어 어떤 물리적인 요구 조건은 특정 시간 안에서만 유지되므로) 다른 요소작업들과 최대한 어떤 거리 이내에 배정되어야 한다.
- 연속 요소작업 : 어떤 일련의 요소작업들은 연속적인 여러 개의 작업장에 할당되어야 한다. 이것은 사이클 타임보다 매우 긴 요소작업이 존재하는 경우에 발생할 수 있다.

여기서는 이들 중에서 하드 제약 성격에 강한 연속 요소작업 할당제약을 제외한 나머지 제약을 소프트 할당제약으로 표현하고 혼합정수계획으로 모델화 하는 방법을 기술한다.

먼저 ‘친밀 및 비친밀 요소작업 제약’을 쉽게 표현할 수 있는 방법을 제안한다. 요소작업과 요소작업 사이의 친밀도를 <Table 1>의 기호를 이용하여 요소작업-요소작업 관계행렬(R)로 표시하는 것이다. 이 방법은 현장의 많은 엔지니어들에게 익숙한 설비배치의 활동상관도와 유사하다. Choi[7]의 연구에서는 ‘친밀도 없음’ 제약을 표현하기 위해 기호 ‘D’를 사용하였으나, 실제로 할당제약에 들어가지 않으므로 삭제하기로 한다.

<Table 1> Symbols and Weights of the Compatibility and Distance for the Task-Task Related Assignment Constraints

Symbols	Descriptions	Weights
A	It is ‘very important’ to assign two tasks to the same workstation.	1
B	It is ‘important’ to assign two tasks to the same workstation.	0.1
C	It is ‘a bit important’ to assign two tasks to the same workstation.	0.01
E	It is ‘a bit important’ not to assign two tasks to the same workstation.	0.01
F	It is ‘important’ not to assign two tasks to the same workstation.	0.1
X	It is ‘very important’ not to assign two tasks to the same workstation.	1
$d_{i,j}$	Number of workstations between two tasks (positive and negative values mean minimum and maximum distances, respectively)	0.1

‘최소 및 최대 거리 제약’은 요소작업-요소작업 관계행렬에 최소 거리 제약인 경우에는 양의 정수로 거리를 표시하고, 최대 거리 제약일 때는 음의 정수로 거리를 표시한다. <Table 1>에서 Weights는 혼합정수계획 모델과 GA의 적용도 평가에서 사용되는 상대적인 중요도 값이다.

이제 요소작업-요소작업 관계행렬의 예로 <Table 2>를 살펴보기로 하자. 요소작업 1과 2의 친밀도가 ‘A’로 표시되어 있으므로 두 작업을 동일 작업장에 할당하는 것이 매우 중요함을 알 수 있다. 반면에 요소작업 3과 5는 최대한 동일 작업장에 할당되지 않도록 하기 위해 친밀도가 ‘X’로 표시되어 있다. 한편, 요소작업 2와 6은 서로 2개 작업장 이상 떨어져야 하고, 요소작업 4와 6은 3개 작업장 이하 거리에 있는 작업장에 배치되는 것을 선호하는 할당제약을 나타내고 있다.

<Table 2> An Example of Relationship Matrix for the Task-Task Related Assignment Constraints

		Tasks						
		1	2	3	4	5	6	...
Tasks	1		A					
	2						2	
	3				C	X		
	4					B	-3	
	5						F	
	6							
	...							

2.2 요소작업-작업장 할당제약

ALB-Research Group[1]의 웹사이트에 있는 다양한 할당제약들 중에서 Choi[7]는 요소작업과 작업장 사이에 존재하는 다음 세 가지 제약을 ‘요소작업-작업장 할당제약’으로 분류하였다.

- 작업장 고정 : 어떤 요소작업들은 특정 작업장에만 지정될 수 있다. 예를 들어, 작업에 필요한 설비가 특정 작업장에만 설치되어 있다.
- 작업장 형태 : 어떤 요소작업들은 특정 형태의 작업장에만 할당되어야 한다. 제품의 하부를 작업하는 것을 예로 들 수 있다.
- 작업장 배제 : 어떤 요소작업들은(예를 들어 필요한 자원이 해당 작업장에 설치될 수 없기 때문에) 특정 작업장에 할당될 수 없다.

요소작업-작업장 소프트 할당제약을 쉽게 표현하기 위해 <Table 3>의 친밀도 기호를 이용하여 작성할 수 있는 요소작업-작업장 관계행렬(Q)을 제안한다. 본 논문에서는 Choi[7]가 제시한 기호 체계를 수정하여 앞의 요소작업-

요소작업 관계행렬의 친밀도 기호와 동일한 기호를 적용하여 단순화 하였다. <Table 3>에서 Weights는 혼합정수계획 모델과 GA의 적용도 평가에서 사용되는 상대적인 중요도 값이다.

<Table 4>는 요소작업-작업장 관계행렬의 예이다. 요소작업 2의 경우에는 작업장 1과 2에 각각 ‘X’와 ‘C’가 표시되어 있으므로 작업장 1에는 최대한 할당되지 않도록 하고 작업장 2에 할당되는 것을 약간 선호한다는 것을 알 수 있다. 반면에 요소작업 4와 작업장 3 사이에는 ‘A’가 표시되어 있으므로 요소작업 4를 작업장 3에 할당하는 것을 매우 중요하게 고려하고 있음을 알 수 있다.

2.3 소프트 할당제약 혼합정수계획 모델

이제, 관계행렬의 데이터를 이용하여 구성된 GALB 문제에 대한 수리 모델을 제시하기로 한다. 먼저 Choi[19]의 연구를 참고하여 모델에서 사용되는 기호 및 변수에 대해 기술한다. 그리고 Choi[7]가 제시한 모델의 요소작업-요소작업 할당제약 중에서 친밀도 및 비친밀 부분에 대한 모델링 오류를 수정하고 요소작업-작업장 할당제약 부분의 모델을 단순화 한 혼합정수계획 모델을 제시한다.

<Table 3> Symbols and Weights of the Compatibility for the Task-Workstation Related Assignment Constraints

Symbols	Descriptions	Weights
A	It is ‘very important’ to assign the task to the workstation	2
B	It is ‘important’ to assign the task to the workstation	0.2
C	It is ‘a bit important’ to assign the task to the workstation	0.02
E	It is ‘a bit important’ not to assign the task to the workstation	-0.02
F	It is ‘important’ not to assign the task to the workstation	-0.2
X	It is ‘very important’ not to assign the task to the workstation	-2

<Table 4> An Example of Relationship Matrix for the Task-Workstation Related Assignment Constraints

		Workstations				
		1	2	3	4	...
Tasks	1					
	2	X	C			
	3					
	4			A		
	5					
	6				F	
	...					

기호 및 변수 2

- r_{ij} : 요소작업-요소작업 관계행렬 R 의 원소
- γ_{ij} : r_{ij} 에 대응되는 가중치(<Table 1> 참조)
- $R^O = \{(i, j) | r_{ij} = 'A', 'B' \text{ or } 'C'\}$
- $R^X = \{(i, j) | r_{ij} = 'E', 'F' \text{ or } 'X'\}$
- B : 매우 큰 양수
- $y_{ij} = 0 \text{ or } 1$: $(i, j) \in R^O$ 의 작업장이 r_{ij} 의 설정과 일치될 때, 목적함수 값이 개선되도록 하는 임시 변수
- $y_{ijg} = 0 \text{ or } 1$, $(i, j) \in R^X$ and $g \in N$: (i, j) 의 작업장이 r_{ij} 의 설정과 일치될 때, 목적함수 값이 개선되도록 하는 임시 변수
- $z_{ijg} = 0 \text{ or } 1$, $(i, j) \in R^X$ and $g \in N$: R^X 모델링에 사용되는 임시 변수
- $R^{Mn} = \{(i, j) | r_{ij} : \text{음의 정수}\}$
- $R^{Mx} = \{(i, j) | r_{ij} : \text{양의 정수}\}$
- ϵ : $(i, j) \in R^{Mn}$ or R^{Mx} 의 작업장이 r_{ij} 의 설정과 일치되지 않을 때 부과되는 페널티로 양수
- y_{ij}^+, y_{ij}^- : $(i, j) \in R^{Mn}$ or R^{Mx} 의 작업장이 r_{ij} 의 설정과 일치되지 않을 때 목적함수에 페널티를 부과하는 임시 변수(단, 변수의 범위는 0 이상의 정수)
- q_{ig} : 요소작업-작업장 관계행렬 Q 의 원소
- δ_{ig} : q_{ig} 에 대응되는 가중치(<Table 3> 참조)

$$Q^O = \{(i, g) | \delta_{ig} > 0\}$$

$$Q^X = \{(i, g) | \delta_{ig} < 0\}$$

$$\text{Min. } \alpha CT + \beta \sum_{g \in N} x_{\lambda g} - \sum_{(i,j) \in R^O} \gamma_{ij} y_{ij} - \sum_{(i,j) \in R^X} \gamma_{ij} (\sum_{g \in N} y_{ijg}) - \sum_{(i,j) \in R^{Mn}} \epsilon y_{ij}^- - \sum_{(i,j) \in R^{Mx}} \epsilon y_{ij}^+ - \sum_{(i,g) \in Q} \delta_{ig} x_{ig} \quad (5)$$

Subject to

(2)~(4)

$$\sum_{g \in N} gx_{ig} - \sum_{g \in N} gx_{jg} \leq By_{ij}, (i, j) \in R^O \quad (6)$$

$$\sum_{g \in N} gx_{jg} - \sum_{g \in N} gx_{ig} \leq By_{ij}$$

$$x_{ig} + x_{jg} \leq 2y_{ijg} + z_{ijg}, (i, j) \in R^X \text{ and } g \in N \quad (7)$$

$$\sum_{g \in N} gx_{ig} - \sum_{g \in N} gx_{jg} - d_{ij} = y_{ij}^+ - y_{ij}^-, (i, j) \in R^{Mn} \quad (8)$$

$$\sum_{g \in N} gx_{ig} - \sum_{g \in N} gx_{jg} + d_{ij} = y_{ij}^+ - y_{ij}^-, (i, j) \in R^{Mx} \quad (9)$$

먼저 요소작업-요소작업 관계행렬로 표현되는 할당 제약 부분의 모델링에 대해 알아보기로 한다. 친밀 요소작업 할당 제약(R^O 의 경우)들은 식 (5)의 3번째 항과 식 (6)으로 모델에 반영되었다. 한편, 비친밀 요소작업 할당 제약(R^X 의 경우)은 식 (5)의 4번째 항과 식 (7)로 모델링 된다. 최소 거리 제약(R^{Mn} 의 경우)은 식 (5)의 5번째 항과 식 (8)로 표현된다. 반면 최대 거리 제약(R^{Mx} 의 경우)은 식 (5)의 6번째 항과 식 (9)로 표현된다. 마지막으로 요소작업-작업장 관계행렬로 표현되는 할당 제약의 경우, $(i, g) \in Q$ 가 q_{ig} 의 설정과 일치될 때, 목적함수 값이 개선되도록 식 (5)에 7번째 항을 추가함으로써 모델링 될 수 있다.

3. 소프트 할당제약 GALB 문제를 위한 GA

본 장에서는 소프트 할당제약을 포함하는 GALB 문제를 풀기 위한 GA를 제시한다. ALB 문제를 위한 GA에 대하여 고찰을 실시한 후, 이를 기초로 소프트 할당제약 GALB 문제를 풀기 위한 GA를 제안하기로 한다. 먼저 사용되는 기호 및 변수를 정의한다. 일반 GA와 관련된 기본 기호는 Kim[15]의 제안을 따랐다.

기호 및 변수 3

- $P(t)$: 세대 t 의 집단
- N_p : 집단의 크기, 즉 집단의 개체 수
- P_c : 교차율
- P_m : 돌연변이율
- r : 필요로 하는 시점에 발생하는 범위 $[0, 1]$ 사이의 난수
- E_k : $P(t)$ 에 속하는 k 번째 개체
- E_0^* : 소프트 할당제약을 적용하지 않은 SALB 문제의 최선 해
- E^* : 소프트 할당제약을 적용한 GALB 문제의 최선해
- s_k^i : E_k 에 있는 작업 $i(i \in M)$ 의 작업장 번호
- n_k : E_k 의 작업장 수
- f_k : E_k 의 적응도
- f^* : 최선해의 적응도
- a_k : E_k 의 소프트 할당제약 충족도
- a_0^* : E_0^* 의 소프트 할당제약 충족도
- a^* : E^* 의 소프트 할당제약 충족도
- a^T : 모든 소프트 할당제약이 충족될 경우, 달성할 수 있는 최종 목표가 되는 충족도
- $ar_0^* = a_0^*/a^T \times 100(\%)$: E_0^* 의 소프트 할당제약 충족율 (achievement ratio)
- $ar^* = a^*/a^T \times 100(\%)$: E^* 의 소프트 할당제약 충족율

ALB 문제를 위한 전형적인 GA의 절차는 아래의 GAalb 알고리즘과 같다[15]. GAalb에서 알 수 있듯이 GA의 주요 구성 항목은 개체와 개체들로 구성되는 집단(population), 적응도 평가(fitness evaluation)와 선별(selection), 교차(crossover)와 돌연변이(mutation) 유전연산자(genetic operation)이다.

알고리즘 1. GAalb

- 단계 1. (초기집단) $t := 0$ 으로 두고, $P(t)$ 를 생성한다.
- 단계 2. (보수) 보수 대상인 모든 E_k 를 보수한다.
- 단계 3. (적응도 평가) 모든 E_k 의 적응도, f_k 를 평가한다.
- 단계 4. (종료 조건) 종료 조건을 만족하면 끝낸다. 그렇지 않으면 단계 5로 간다.
- 단계 5. (선별) $t := t + 1$ 로 두고, $P(t-1)$ 에서 $P(t)$ 를 선택한다.
- 단계 6. (교차)
 - 단계 6.1 모든 k 에 대해 $r < P_c$ 이면, E_k 를 교차 대상으로 둔다.
 - 단계 6.2 모든 교차 대상 개체들에 대해 임의로 쌍(두 부모)을 만든다.
 - 단계 6.3 각 쌍의 개체들을 교차하여 두 자손을 생산한다. $P(t)$ 에서 교차된 부모 개체를 제거하고 생산된 자손을 $P(t)$ 에 추가한다.
- 단계 7. (돌연변이) 모든 k 에 대해 $r < P_m$ 이면, E_k 를 돌연변이 시킨다.
- 단계 8. 단계 2로 간다.

ALB 문제를 풀기 위한 GA 관점에서 구성 항목에 대해 파악하기로 한다. 먼저 개체와 집단에 대해 알아보기로 하자. GA는 개체들로 구성된 집단을 운용한다. GA에서 풀고자하는 문제의 잠재해는 생물학의 개체에 해당된다. 개체는 염색체(chromosome)로 표현되며 염색체는 유전자(gene)라는 단위로 구성된다. 따라서 GA를 적용하기 위한 첫 번째 단계는 ALB 문제의 잠재해를 염색체라고 불리는 스트링 형태의 구조로 변환하여 표현하는 것이다[26]. 표현 방법은 유전연산자의 개발과 밀접하게 관련되어 있으며, GA의 전반적인 탐색 성능에 영향을 미치므로 문제의 특성을 잘 반영할 수 있도록 결정해야 한다[8, 26]. 요소작업 기반, 배아(embryonic), 작업장 기반, 그룹핑 기반, 휴리스틱 기반, 등 많은 표현 방법이 제시되어 있다[18, 26]. 본 연구에서는 염색체를 요소작업의 순열로 정의하는 ‘요소작업 기반 표현’을 사용하였다. 이 표

현법은 여러 형태의 ALB 문제에 적용이 가능하고 유전연산자를 선택하는데 있어서 융통성을 제공하는 장점을 가지고 있다[3].

위에서 언급한 것처럼 GA의 초기집단 $P(0)$ 을 구성하기 위해 N_p 개수만큼의 개체들을 생성해야 한다. 생성 방법에는 문제의 특성을 이용한 발견적 방법과 임의 생성법이 있다. 기존 연구들에 따르면, 발견적 방법에 의해 생성한 해들은 초기에 수렴하여 지역해에 빠지는 경향이 있어서 해공간의 폭넓은 탐색에서 불리한 것으로 밝혀졌기 때문에 임의 생성법을 더 많이 사용한다[14, 18]. Kim et al. [18]이 제안한 임의 생성법에서는 선후행 관계를 만족하는 임의의 순서로 작업들을 나열하여 개체들을 생성한다[5, 15]. 본 논문에서는 M 에 있는 요소작업들을 선후행 관계의 준수여부는 고려하지 않고 무조건 무작위로 배열하여 N_p 개의 개체를 생성하는 간단한 초기집단 구성 방법을 적용하기로 한다.

그런데 이 경우, 대부분의 개체들이 선후행 제약조건에 위배되므로 가능 개체로 수정하는 보수전략(repairing strategy)이 필요하다[15, 26]. 본 연구에서는 Kim et al.[19]의 제안 방법에 기초하여 개발한 보수 알고리즘(rearranging algorithm, 이하 RA)을 사용하기로 한다. RA의 구체적인 절차는 아래와 같다.

알고리즘 2. RA

- 단계 1. 보수 대상 개체 E_i 를 E'_i 으로 복사한다.
- 단계 2. 비어 있는 개체를 생성하여 E'_i 로 놓고, M 에서 선행작업이 없는 작업들의 집합 C 를 만든다.
- 단계 3. C 에 있는 작업 중에서 E'_i 의 가장 처음에 위치한 작업을 선택하여 E'_i 에서 삭제하고 E'_i 에 추가한다.
- 단계 4. 선택된 작업을 C 에서 제거하고, 선택된 작업의 후행작업 중에서 모든 직선행작업이 이미 개체에 할당된 작업을 C 에 추가한다.
- 단계 5. C 가 공집합이면 종료한다. 그렇지 않으면 단계 3로 간다.

적응도 평가와 선별에 대해 기술하기로 한다. 요소작업 기반 표현방식 염색체의 적응도를 계산하려면, 먼저 염색체의 유전자(요소작업)를 순서에 따라 작업장에 할당하는 절차가 필요하다[26]. 간단한 작업장 할당 방법은 개체에 표현된 순서에 따라 요소작업들을 첫 작업장부터 차례로 할당하면서 한 작업장의 작업량이 사이클 타임을 넘지 않도록 하는 것이다. 이러한 과정을 모든 요소작업이 할당될 때까지 반복한다[18].

흔히 목적함수를 사용하는 적응도는 개체의 생존 능력을 나타낸다. 본 논문에서는 식 (5)에 기초한 충족도 기반 적응도 계산법(fitness evaluation based on achievement, 이하 FA)을 제안한다. 알고리즘 FA의 구체적인 절차는 아래와 같다. 먼저 단계 1~단계 6의 절차에 따라 개체 E_k 가 할당제약을 충족하는 정도를 나타내는 a_k 를 계산한 후, 단계 7에서 E_k 의 적응도, f_k 를 계산한다. GAarc는 ALB 문제의 기본 목적을 달성할 수 있는 최선해를 구한 후, 이 해에 대한 대안해를 찾는 것이므로 알파와 베타에는 <Table 1>과 <Table 3>의 가중치보다는 월등히 큰 값을 배정해야 한다. 본 논문에서는 ‘100’을 적용하였다.

알고리즘 3. FA

단계 1. $a_k := 0$

단계 2. $a_k := a_k + \sum_{\{(i,j)|(i,j) \in R^o \wedge S_k^i = S_k^j\}} \gamma_{ij}$

단계 3. $a_k := a_k + \sum_{\{(i,j)|(i,j) \in R^x \wedge S_k^i \neq S_k^j\}} \gamma_{ij}$

단계 4. $a_k := a_k + \sum_{\{(i,j)|(i,j) \in R^{lim} \wedge |S_k^i - S_k^j| \geq |\gamma_{ij}|\}} \epsilon$

단계 5. $a_k := a_k + \sum_{\{(i,j)|(i,j) \in R^{max} \wedge |S_k^i - S_k^j| \leq |\gamma_{ij}|\}} \epsilon$

단계 6. $a_k := a_k + \sum_{(i,g) \in Q^o} \delta_{ig} x_{ig} + \sum_{(i,g) \in Q^x} \delta_{ig} (1 - x_{ig})$

단계 7. $f_k := \alpha C + \beta n_k - a_k$

한편 선별은 적자생존의 자연법칙에 따라 다음 세대에 생존할 개체를 선택하는 과정이다. 선별 방법에는 대표적으로 세 가지가 있다[15, 26]. 먼저 확률 바퀴(roulette wheel) 선별 방법은 개체의 적응도에 비례하여 개체가 선택될 확률을 부과하는 방법이다. 순위(ranking) 선별은 가장 좋은 개체부터 차례로 순위를 주어 그 순위에 따라 선별 확률을 부여한다. 끝으로 토너먼트 선별(tournament) 선별에서는 두 개 또는 그 이상의 개체들을 비교하여 그 중에서 적응도가 높은 개체를 선택한다. 본 논문에서는 토너먼트 방법을 적용하였다. 평가값의 척도를 재구성할 필요가 없고, 토너먼트의 크기에 의해 집단의 선별압력을 조정할 수 있다는 장점이 있다[18].

교차와 돌연변이 유전연산자는 선별된 개체들 중에서 일부를 짝을 지어 교배하여 새로운 자손(개체)을 생성한다. 자손은 부모로부터 좋은 유전형질을 상속받음으로써 더 우수한 잠재해들의 발현이 가능해진다. 대표적인 교차 유전연산자로 부분사상교차(partially mapped crossover), 개선된 인접인자재결합(enhanced edge recombination), 순서교차(order crossover), 균등순서교차(uniform order-based cross-

over), 순환교차(cycle crossover)을 들 수 있다[18]. ALB 문제와 같이 순서 문제와 관련된 유전연산자들의 효율성은 문제에 따라 크게 달라진다고 알려져 있으나[25], Kim et al. [16]과 Kim et al.[18]의 연구 결과에 따라 본 논문에서는 부분사상교차 방법을 적용하기로 한다.

돌연변이 유전연산자로는 교환(reciprocal exchange, RE), 역순(inversion, INV), 삽입(insertion, INS), 전위(displacement, DIS)가 대표적이다[3, 16]. SALB 문제의 경우에는 Kim et al.[18] 연구에 의하면 교환 연산자가 우수하였으나, ALB 문제에 대한 GA의 해 탐색성능은 돌연변이 유전연산자에 의해 크게 영향을 받는 것으로 알려져 있으므로 본 논문에서는 실험을 통하여 선정하기로 한다.

그런데 유전연산을 통해 생성된 개체들은 초기집단을 구성하는 개체들과 마찬가지로 선행 제약조건에 위배되는 불가능해가 될 수 있으므로 보수 절차가 필요하다. 초기집단의 경우와 동일하게 RA를 적용하였다.

유전연산자와 관련하여 추가로 고려해야 할 사항이 있는데, 최선해가 유전연산에 의해 사라질 수 있다는 점이다. 본 연구에서는 매 세대마다 가장 좋은 개체를 항상 집단 내에 보존하는 최선해 보존전략(elitist strategy)을 적용하였다[18].

이제 끝으로 소프트 할당제약 GALB 문제를 풀기 위한 GAalb 기반 알고리즘(soft allocation constraint GA, 이하 GASac)의 전체 구조를 제안하기로 한다. 소프트 할당 제약 GALB 문제는 소프트 제약이 없는 SABLBP의 다수의 최적해들(multiple optimal solutions) 중에서 소프트 할당 제약을 최대한 만족하는 해를 찾는 문제라고 할 수 있다. GASac는 먼저 E_0^* 를 구한 후, 이 해를 이용하는 GAalb를 실행하여 소프트 할당제약의 충족도를 향상시키는 해를 찾는 절차로 구성된다.

본 논문에서는 E_0^* 를 구하기 위해 Scholl과 Klein[24]이 제안한 SALOME를 사용하여 SALB 문제를 푼다. 참고로 SALOME는 새로운 지역 하한 방법(local lower-bound method)과 양방향 분기 규칙(branching strategy)을 적용하는 알고리즘으로 SALB 문제에 대해 성능이 가장 우수한 것으로 알려져 있다[4, 23]. 참고로 ALB-Research Group의 홈페이지에서 SALOME에 대한 컴퓨터 프로그램을 제공하고 있다[1].

ALBP-2 타입의 문제를 예로 들면, SALOME로 주어진 사이클 타임에 대하여 작업장의 수를 최소로 하는 문제를 푼 후, GASac를 적용하여 소프트 할당제약 GALB 문제를 푼다. 만일 최종 해의 할당제약 충족도가 만족스럽지 못하면, 주어진 제약조건을 약간 완화하여(이 예에서는 사이클 타임을 약간 증가시켜서) 충족도를 향상시키는 해를 탐색할 수 있다. 알고리즘 GASac의 구체적인 절차는 아래와 같다.

알고리즘 4. GAsac

- 단계 1. 소프트 할당제약을 적용하지 않은 SALB 문제에 대해 SALOME를 이용해서 최선해, E_0^* 를 구한다.
- 단계 2. E_0^* 에 대한 소프트 할당 제약 충족도를 평가한다. 만일 충족도가 만족스러우면 종료한다. 그렇지 않으면 단계 3으로 간다.
- 단계 3. 초기집단에 임의 생성 개체와 함께 E_0^* 를 투입하고 적응도 계산을 위해 FA를 이용하는 GAalb를 적용하여 소프트 할당제약을 포함하는 GALB 문제를 푼다.
- 단계 4. 만일 최선해의 할당 제약 충족도가 만족스러우면 종료한다. 그렇지 않으면, 하드 제약조건을 완화하고 단계 3으로 간다.

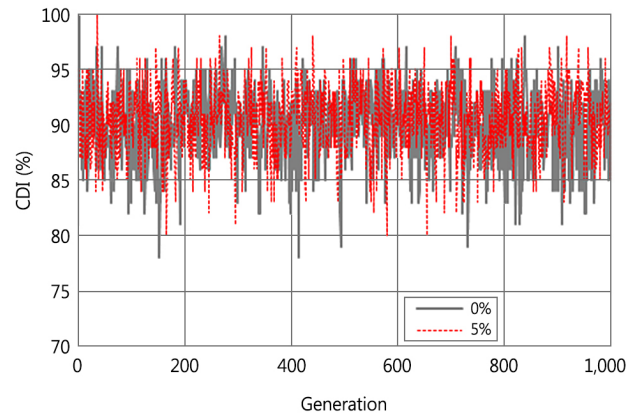
본 논문에서 제안하는 GAsac은 초기집단을 임의 생성법과 소프트 할당제약을 적용하지 않은 SALB 문제의 최적해, E_0^* 를 함께 투입하는 발견적 방법을 혼용해서 사용한다. 앞에서 언급한 바와 같이 발견적 방법에 의해 생성된 해들은 해공간의 다양한 탐색을 방해하는 단점이 있는 것으로 알려져 있다[18]. 본 논문의 제안 방법을 적용하는 경우, 개체(해) 집단의 다양성 측면에서 문제가 있는지 알아보기 위해 군집분석(cluster analysis)을 이용하여 임의로 생성한 개체만을 초기해로 투입하는 경우와 비교하였다.

먼저 특정 세대의 집단내의 전체 염색체(개체) 중에서 얼마나 다양한 염색체가 존재하는 지를 나타내는 염색체 다양성 지수(chromosome diversity index 이하 CDI)를 아래의 식 (10)과 같이 정의하였다. ‘염색체 그룹 수’는 군집분석에서 두 개체간의 거리를 계산하는데 흔히 쓰이는 유클리드 거리(Euclidean distance)를 이용하여 계산하여 구할 수 있으나 여기서는 간단하게 두 염색체 사이의 유전자 배열이 모두 동일하면(즉, 동일한 해이면) 같은 그룹에 포함하는 방식으로 계산하였다. CDI 값이 클수록 집단이 다양한 염색체들로 구성되어 있음을 의미한다.

$$CDI = \frac{\text{염색체 그룹 수}}{N_p} \times 100(\%) \quad (10)$$

WARNECKE 데이터(실험 데이터 셋에 대해서는 제 4.1절을 참조하기 바람)를 이용하여 GAsac의 세대수가 증가함에 따라 두 경우의 CDI 값을 수집한 후, 시계열 그래프를 작성한 것이 <Figure 1>이다. 그래프의 범례에서 0%가 임의로 생성한 개체만을 초기해로 투입하는 경우(즉, N_p 개의 초기집단의 개체 들 중에서 E_0^* 가 차지하

는 비율이 0%임)이고, 5%는 E_0^* 가 초기집단의 N_p 개 개체 중에서 5%를 차지하는 경우이다. 초기해로 매우 우수한 최선해를 많이 투입하는 경우에 CDI의 시계열이 얼마나 좋지 않게 형성되는지 알아보기 위해 높은 비율을 선택하였다. 그래프를 보면 세대수가 계속 진행되어도 두 경우의 CDI에 차이가 없으므로 본 논문에서 제안하는 방법을 적용하여도 잠재해 집단의 다양성이 저해되지 않음을 알 수 있다.



<Figure 1> Time Series Graph for CDIs

4. 성능평가 실험

4.1 예비 실험

성능평가를 실시하려면 먼저 문제를 준비해야 한다. 본 연구에서는 ALB-Research Group[1]이 제공하는 다양한 테스트 문제 중에서 요소작업 수가 각각 11, 30, 58, 94개인 MANSOOR, SAWYER30, WARNECKE, MUKHERJE의 4개를 임의로 선정하여 ALBP-2 타입의 문제를 대상으로 선정하였다. 각각에 대해 다양한 소프트 할당 제약이 존재하는 문제 셋을 10개씩 임의로 생성하였다. 이를 위해 전체 요소작업 중에서 임의로 30%가 친밀 또는 비친밀 제약을 갖도록 하고, 10%에 최소 및 최대 거리 제약을 부여했다. 그리고 전체 작업장의 15%가 친밀 또는 비친밀 요소작업-작업장 제약을 갖도록 했다. 그런데 임의로 할당 제약을 생성하는 경우, 최선을 다해도 충족될 수 없는 의미 없는 할당 제약이 만들어질 수 있다. 이를 방지하기 위해 요소작업 i 의 할당가능 작업장의 범위, (E_i, L_i) 를 구하여 적용하였다. 여기서 E_i 와 L_i 는 사이클 타임과 선행 제약하에서 요소작업 i 가 할당될 수 있는 가장 앞과 뒤 작업장의 번호로 구체적인 계산식은 Patterson과 Albracht[21], 그리고 Kim et al.[17]을 참고하기 바란다.

GAsac를 실행하기 위해서는 기본 모수들을 설정해야 한다. 다양한 예비실험을 통하여 N_p , P_c 그리고 P_m 의 값을 각각 100, 0.4, 0.5로 설정하였다. 2,000세대가 지나서도 해가 개선되는 경우가 발생하여 최대 세대수 2,500에 도달하거나, 해의 개선이 없이 2,000세대가 지나면 종료하는 것으로 종료조건을 설정하였다. 그리고 실험을 통하여 E_0^* 를 초기집단에 투입하는 비율은 2%로 선정하였다.

이제 소프트 할당제약 GALB 문제에 가장 적합한 돌연변이 유전연산자를 선정하기 위해 앞에서 언급한 교환(RE), 역순(INV), 삽입(INS), 전위(DIS) 4종류의 연산자에 대해 SAWYER30과 WARNECKE 두 데이터를 이용하여 각각 30회 실험을 수행하였다. 성능평가 척도로는 소프트 할당제약 충족율, ar^* 를 적용하였다. GAsac로 구한 모든 최선해(E^*)는 소프트 할당제약을 적용하지 않은 SALB 문제의 최선해(E_0^*)들 중에서 소프트 할당제약을 최대한 충족하는 해이므로 f^* 에서 차이가 나는 부분은 a^* 이다. 따라서 ar^* 값이 더 큰 해가 더 우수한 해가 된다.

돌연변이 연산자들의 성능 사이에 유의한 차이가 있는지 알아보기 위해 분산분석을 실시한 결과가 <Table 5>이다. 표에서 알 수 있듯이 유의수준 5% 하에서 두 데이터 모두 연산자들의 성능(ar^*) 사이에 유의한 차이가 있다. 연산자들 간의 성능 차이 유무를 알아보기 위해 Fisher의 최소유의차(least significant difference, LSD) 방법을 이용하여 다중비교검정(multiple range test)을 수행하였다. 검정 결과를 정리한 것이 <Table 6>이다(표에서 MGO는 돌연변이 유전연산자(mutation genetic operator)를 의미한다). 표로부터 RE가 가장 우수함을 알 수 있다. 따라서 본 연구에서는 RE를 돌연변이 유전연산자로 적용하였다.

<Table 6> Results of the LSD Multiple Range Test for the MGOs

(a) SAWYER30

MGOs	Average	Homogeneous groups		
RE	80.20	A		
INV	78.22	A	B	
DIS	74.38		B	C
INS	71.82			C

(b) WARNECKE

MGOs	Average	Homogeneous groups		
RE	64.45	A		
INS	63.99		B	
INV	63.83		B	C
DIS	63.55			C

끝으로 적응도 계산을 위해 개체의 유전자(요소작업)를 순서에 따라 작업장에 할당하는데 적용되는 Kim et al. [18]의 단순 할당 방법을 개선할 수 있는 새로운 작업장 할당 방법을 제시하고자 한다. 단순 작업장 할당 방법은 개체에 표현된 순서에 따라 요소작업들을 첫 작업장부터 차례로 할당하면서 한 작업장의 작업량이 사이클 타임을 넘지 않도록 하는 것이다. 즉, j 번째 요소작업을 현재 할당 중인 g 번째 작업장에 할당할 수 없으면, $g+1$ 번째 작업장에 할당을 진행해가는 방식이다. 그런데 만일 $j+1$ 번째 요소작업이 j 번째 요소작업과의 선행 제약을 위반하지 않고 작업시간이 g 번째 작업장의 잔여 허용시간 이하이면, 후순위인 $j+1$ 번째 요소작업을 먼저 할당함으로써 좀 더 나은 해를 얻을 수 있는 가능성을 높일 수 있을 것이다.

<Table 5> Analysis of Variance for the Performance of the MGOs

(a) SAWYER30

Source of Variation	Sum of Squared Error	DF*	Mean Squared Error	F-value	P-value
MGO	1276.383	3	425.461	3.207	0.026**
Error	15388.028	116	132.655		
Total	16664.410	119			

* DF stands for degree of freedom.

** There is a statistical difference in the effects at the significance level of 5%.

(b) WARNECKE

Source of Variation	Sum of Squared Error	DF*	Mean Squared Error	F-value	P-value
MGO	12.727	3	4.242	8.484	3.832E-05**
Error	58.002	116	0.500		
Total	70.729	119			

* DF stands for degree of freedom.

** There is a statistical difference in the effects at the significance level of 5%.

본 논문에서는 j 번째 요소작업을 현재 할당 중인 작업장에 할당할 수 없을 때, 후순위인 $j+1$ 번째와 $j+2$ 번째 요소작업의 할당을 시도하는 새로운 방법의 성능을 알아보기 위해 SAWYER30과 WARNECKE 데이터를 이용하여 각각 30회 실험을 수행하였다. 실험 결과가 <Table 7>에 정리되어 있다. 표에서 Best, Average, Worst는 각각 30회 실험 중에서 얻는 ar^* 의 최고, 평균, 최저치이다. 두 데이터 셋에 대해 제안 방법과 기존 방법 성능의 평균 차이에 대한 z -검정을 실시한 결과를 보면, SAWYER30에 대해서는 제안 방법이 유의하게 우수하였고, WARNECKE에서는 제안 방법의 평균이 더 높았으나, 유의한 차이는 없었다. 실험 결과를 종합하면, 제안 방법이 더 우수한 해를 구할 수 있는 가능성이 높으므로 새로운 제안 방법을 적용하기로 하였다.

<Table 7> Test Results for the Performance of the Workstation Allocation Methods

Data workstation allocation method	SAWYER30		WARNECKE	
	Proposed method	Existing method	Proposed method	Existing method
Best	96.02	96.02	66.27	66.27
Average	91.32	80.20	64.59	64.45
Worst	74.16	74.16	63.45	63.45
Z-value	5.220		0.743	
P-value	8.930E-08*		0.229	

*There is a statistical difference in the effects at the significance level of 5%.

4.2 본 실험

이제 소프트 할당제약 GALB 문제를 풀기 위해 본 논문에서 제안하고 있는 GAsac의 성능을 알아보는 실험을 실시하기로 한다. 앞 절의 예비 실험 결과를 적용하여 4가지 데이터에 대해 10개의 소프트 할당제약 셋(AC 1~10)을 각각 30회씩 반복 실행하였다. GAsac를 컴퓨터 프로그램으로 코딩하기 위해 엑셀 VBA를 사용하였다. 실험에는 i7-2600 3.4GHz 컴퓨터를 사용하였다.

GAsac의 4단계 절차 중에서 1~3단계를 적용한 실험 결과를 <Table 8>로 정리하였다. 표에서 SO는 SALOME로 구한 SALB 문제 최선해의 할당제약 충족율(ar_0^*)이고 IR(Improvement Ratio)은 10개의 소프트 할당제약 문제 집합 중에서 Worst, Average, Best 항목별로 SO보다 해가 향상된 비율이다. WARNECKE 데이터를 예로 들어 보면, Worst 항목에서 SO보다 개선된 해를 얻은 경우가 10개 문제 중에서 2개(회색으로 칠해진 셀)이므로 해당 IR 값이 20%이다. Average 항목의 경우에는 5개가 처음으로 SO보다 우수한 해(Worst 항목에서는 SO와 동일한 해이었지만)를 구했으므로 총 7개가 SO보다 우수하여 IR은 70%로 계산되었다. 표에서 알 수 있듯이 GAsac가 매우 우수한 성능을 보이고 있다. MANSOOR 데이터의 경우는 요소작업 수가 11개로 작은 규모의 문제이어서 상대적으로 E_0^* 에 대한 대체해(alternative solution)가 많지 않아서 소프트 할당제약을 더 많이 충족시킬 수 있는

<Table 8> Results for the Performance of the Proposed Algorithm

Data	Categories	Allocation Constraint Sets										IR (%)
		AC 1	AC 2	AC 3	AC 4	AC 5	AC 6	AC 7	AC 8	AC 9	AC 10	
MANSOOR	Best	34.95	80.51	63.21	67.31	62.26	78.04	81.97	58.68	9.91	99.50	40
	Average	34.95	80.51	63.21	67.31	62.26	78.04	81.97	58.68	9.91	99.50	40
	Worst	34.95	80.51	63.21	67.31	62.26	78.04	81.97	58.68	9.91	99.50	40
	SO	34.95	41.52	63.21	64.10	62.26	41.70	81.97	58.68	9.91	0.50	-
	ACT(sec.)	3	3	3	3	3	4	3	3	3	3	-
SAWYER30	Best	96.02	97.22	89.80	94.26	76.68	64.34	96.21	77.81	10.29	94.74	100
	Average	90.70	96.92	89.80	85.85	75.79	63.73	96.21	77.41	8.54	94.74	100
	Worst	74.16	95.89	89.80	77.04	75.58	62.06	96.21	76.72	7.43	94.74	100
	SO	10.34	42.65	88.27	56.00	65.53	37.71	58.33	62.51	1.71	48.25	-
	ACT(sec.)	16	17	13	24	22	23	19	24	23	19	-
WARNECKE	Best	64.93	58.71	38.53	61.52	60.62	72.24	76.12	51.50	39.80	56.00	70
	Average	64.27	58.71	38.39	61.52	51.28	69.79	74.76	50.55	39.80	55.44	70
	Worst	63.45	58.71	38.39	61.52	50.61	61.72	74.67	50.32	39.80	54.10	20
	SO	63.45	58.71	38.39	61.52	50.61	61.72	74.67	50.23	39.80	53.14	-
	ACT(sec.)	44	36	37	36	36	42	37	44	35	45	-
MUKHERJE	Best	70.44	46.47	47.16	62.54	52.98	75.89	81.74	78.16	57.93	64.19	100
	Average	70.09	36.33	46.77	55.39	47.46	75.15	67.37	78.07	45.52	58.94	100
	Worst	69.54	29.34	46.39	49.44	44.13	74.98	65.14	77.72	34.73	58.13	100
	SO	64.49	27.97	42.13	36.97	43.82	65.80	60.57	69.78	34.28	57.63	-
	ACT(sec.)	295	277	282	289	294	288	267	255	279	278	-

해를 찾을 수 없었을 것으로 판단된다. 표에서 ACT(Average Computing Time)는 30회 반복실험의 1회 평균 소요시간이다. 문제의 요소작업수가 증가함에 따라 실행시간이 크게 늘어나고 있는데 조립 라인의 설치한 실시간으로 해를 요구하지 않는 중장기 의사결정 문제이며 대개 대규모 자본 투자를 요구한다[3, 7]. 따라서 계산시간이 다소 걸리더라도, 좋은 해를 확보하는 것이 매우 중요하다.

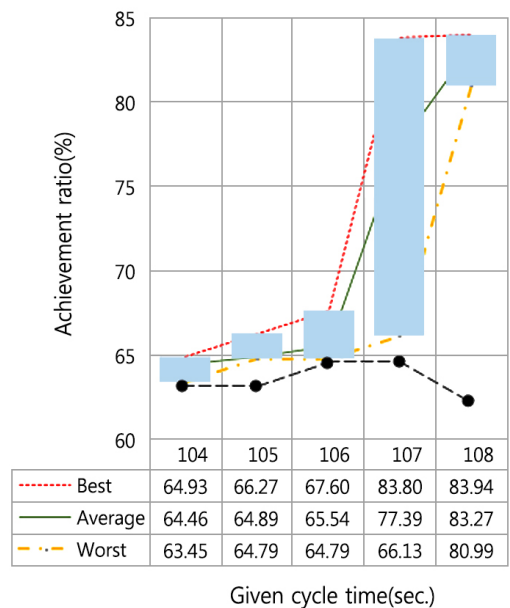
마지막으로 알고리즘 GAsac의 ‘단계 4’를 적용한 실험 결과를 제시한다. 하드 제약조건을 완화함으로써 소프트 할당제약 충족율(ar^*)을 향상시키는 단계이다. 여기서는 WARNECKE 데이터를 대상으로 ALBP-2 타입의 문제를 다루기로 한다. 30회 반복실험을 실시한 결과를 정리한 것이 <Figure 2>이다. CT를 104~108초로 하여 SALOME로 구한 SALB 문제 최선해 SO는 모두 15개의 작업장을 갖는 최적해이다. 이때 계산한 ar_0^* 를 표시한 것이 그림의 맨 아래 검은 점선의 그래프이다. 사이클 타임은 하드 제약조건으로 반드시 만족해야 하지만 동일한 목적함수값(여기서는 작업장의 수)을 유지하는 범위 안에서 사이클 타임을 완화해감에 따라 그림에서 알 수 있듯이 소프트 제약 충족율을 향상시킬 수 있다. GAsac을 이용함으로써 민감도 분석을 통해 SALB 문제의 최적해 조건을 만족하면서 소프트 제약 충족율을 만족스러운 수준까지 올릴 수 방안을 검토할 수 있다.

5. 결론

ALB 문제는 조립라인의 최적 설계를 다루는 중요한 의사결정 사항으로 본 논문에서는 정량적으로 정확히 표현하기는 어렵지만 되도록 충족되기를 바라는 소프트 할당제약을 포함하는 조립라인 밸런싱 문제를 다루었다. 많은 ALB 문제는 다양한 할당제약을 가지고 있는데, 그동안 대부분의 연구들은 만족되어야 하는 엄격한 하드 제약 문제만을 다루어 왔다. 본 논문에서는 Choi[7]가 최초로 제시한 소프트 제약에 대한 혼합정수계획 모델의 오류를 보완하여 소개하였다. 그리고 이를 바탕으로 개발한 GA 기반의 새로운 알고리즘, GAsac를 제안하였다. 성능평가를 위한 실험을 실시한 결과, GAsac가 소프트 할당제약의 충족율이 높은 해를 구할 수 있음을 알 수 있었으며, 민감도 분석을 통해 충족율을 더욱 향상시킬 수 있었다.

추후 연구과제로 기존의 엑셀 기반의 VITAMAX[6]와 같은 생산운영관리 소프트웨어에 포함시키는 것을 고려할 수 있다. 그리고 대규모 문제의 실행시간을 단축하기 위해 스크립트 언어보다 실행속도가 더 빠른 컴파일 언어로 프로그램 코드를 개발하거나, 본 논문에서 제안한

GAsac를 효율화하기 위해 긴 세대수를 단축하는 방안 등의 개선 연구가 필요하다고 사료된다. 끝으로 GAsac의 전처리 모듈로 할당제약 체크 기능에 관한 연구를 고려할 수 있다. 요소작업-요소작업과 요소작업-작업장 관계 행렬을 작성하는 시점에서 할당가능 작업장의 범위, 사이클 타임 또는 다른 제약으로 인해 특정 소프트 제약의 수용이 불가능한 것을 미리 체크할 수 있다면, 좀 더 좋은 해를 빠르게 구할 수 있을 것이다.



Note : The black dotted line at the bottom is the value corresponding to SO.

<Figure 2> Sensitivity Analysis for the Relaxation of the Hard Constraint

References

- [1] ALB-Research Group, 2018, Available from : <http://www.assembly-line-balancing.de/>.
- [2] Battaia, O. and Dolgui, A., A taxonomy of line balancing problems and their solution approaches, *Int. J. Production Economics*, 2013, Vol. 142, No. 2, pp. 259-277.
- [3] Becker, C. and Scholl, A., A survey on problems and methods in generalized assembly line balancing, *European J. of Operational Research*, 2006a, Vol. 168, No. 3, pp. 694-715.
- [4] Bukchin, J. and Tzur, M., Design of Flexible assembly line to minimize equipment cost, *IIE Transactions*, 2002, Vol. 32, No. 7, pp. 585-598.
- [5] Choi, K.H. and Kim, C.W., A U-shape Mixed Model Assembly Line Balancing Problem for Processing Time

- and Physical Workload Using the Genetic Algorithm, *J. of Society of Korea Industrial and Systems Engineering*, 2005, Vol. 28, No. 3, pp. 98-108.
- [6] Choi, S.H., An Integer Programming Model for Generalized Assembly Line Balancing Problems, *The J. of productivity*, 2011, Vol. 25, No. 1, pp. 411-434.
- [7] Choi, S.H. and Mo, C.W., A Visual Motion and Time Study Software, VITAMAX, *Journal of Society of Korea Industrial and Systems Engineering*, 2010, Vol. 33, No. 2, pp. 33-38.
- [8] Guo, Z.X., Wong, W.K., Leung, S.Y.S., Fan, J.T., and Chan, S.F., A genetic-algorithm-based optimization model for scheduling flexible assembly lines, *Int J. Adv. Manuf. Technol.*, 2008, Vol. 36, No. 1-2, pp. 156-168.
- [9] Herdy, M., Application of the Evolutions strategie to discrete optimization problems, *Proc. of the 1st Int. Conf. on Parallel Problem Solving from Nature*, 1991, pp. 187-192.
- [10] Holland, J.H.. Adaptation in natural and artificial systems, Michigan, USA : The University of Michigan Press, 1975.
- [11] Hu, S.J. and Ko, J.H., A brief review of trends in design and research of assembly line, *Industrial Engineering Magazine*, 2014, Vol. 21, No. 3, pp. 24-30
- [12] Kendall, J.W., Hard and soft constraints in linear programming, *Omega*, 1975, Vol. 3, No. 6, pp. 709-715.
- [13] Kim, J.S. and Lee, J.S., A Case of Operational Efficiency Improvement in EPS Motor Manufacturing Process Using IE Technique, *Journal of the Korea Academia-Industrial cooperation Society*, 2017, Vol. 18, No. 7, pp. 63-72.
- [14] Kim, Y.J., Kim, H.S., and Song, W.S., A genetic algorithm for flexible assembly line balancing, *Proceedings of the Korean Institute of Industrial Engineers Spring Joint Conference*, 2004, pp. 425-428.
- [15] Kim, Y.K. and Kwon, S.H., 0-1 Programming Formulations for Assembly Line Balancing of Large-Sized Product, *J. of the Korean Operations Research and Management Science Society*, 1992, Vol. 17, No. 1, pp. 55-65.
- [16] Kim, Y.K., Kim, Y.J., Kim, J.H. Kim, and Kwak, J.S., A Genetic Algorithm for Two-sided Assembly Line Balancing, *IE interfaces*, 1999, Vol. 12, No. 1, pp. 132-142.
- [17] Kim, Y.K., Metaheuristics, Gwangju, Korea : Chonnam National University Press, 2017.
- [18] Kim, Y.K., Yun, B.S, and Lee, S.B., Metaheuristic, Seoul, Korea : Yeongji Moonhwas, 1997.
- [19] Lee, H.F. and Stecke, K.E., An Integrated design support method for flexible assembly system, *J. of Manufacturing Systems*, 1996, Vol. 15, No. 1, pp. 13-32.
- [20] Lee, S.W. and Park, B.J., The development of critical node method based heuristic procedure for Solving fuzzy assembly-line balancing problem, *J. of Society of Korea Industrial and Systems Engineering*, 1999, Vol. 22, No. 51, pp. 189-197.
- [21] Patterson, J.H. and Albracht, J.J., Assembly-Line Balancing : Zero-One Programming with Fibonacci Search, *Operations Research*, 1975, Vol. 23, No. 1, pp. 166-172.
- [22] Rachamadugu, R. and Talbot, B., Improving the equality of workload assignments in assembly lines, *International J. of Production Research*, 1991, Vol. 29, No. 3, pp. 619-633.
- [23] Scholl, A. and Becker, C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *European J. of Operational Research*, 2006, Vol. 168, No. 3, pp. 666-693.
- [24] Scholl, A. and Klein, R., SALOME : A Bidirectional Branch-and-Bound Procedure for Assembly Line Balancing, *Inform J. on Computing*, 1997, Vol. 9, No. 4, pp. 319-334.
- [25] Starkweather, T., Mcdaniel, S., Mathias, K.E., Whitley, D., and Whitley, C.K., A comparison of genetic sequencing operators, *Proceedings of the fourth International Conference on Genetic Algorithms*, 1991, pp. 69-76.
- [26] Tasan, S.O. and Tunali, S., A review of the current applications of genetic algorithms in assembly line balancing, *J. Intell. Manuf.*, 2008, Vol. 19, No. 1, pp. 49-69.

ORCID

Seong-Hoon Choi | <http://orcid.org/0000-0003-0152-2356>

Geun-Cheol Lee | <http://orcid.org/0000-0002-8555-7064>