

Binary Artificial Bee Colony를 이용한 궤도 유지보수 일정계획 Railway Track Maintenance Scheduling using Binary Artificial Bee Colony

남덕희† 김기동* 김성수* 이성욱** 우병구** 이기우***
Duk-Hee Nam Ki-Dong Kim Sung-Soo Kim Sung-Uk Lee Byoung-Koo Woo Ki-Woo Lee

ABSTRACT

The objective of this paper is to propose a Binary Artificial Bee Colony (BABC) to obtain the best/optimal solution for NP-hard scheduling problem of railway track maintenance. We can greatly maximize the objective value using proposed BABC within limited computation time. The proposed BABC mechanism is very efficient to find the best solution because of employing fewer control parameters.

Keyword : Railway track maintenance, Artificial Bee Colony

1. 서론

궤도는 그것을 관리하는 단체, 사용하는 차량, 또는 그 용도 등에 따라 다양한 방법으로 유지보수 되어 지고 있다. 따라서 유지보수 업무를 수행하는 주체의 성격에 따라 일정계획 시 반영해야 하는 현장의 상황들도 다양한 차이점을 보이고 있다. 본 논문에서는 이러한 상황에 유연하게 대처하고 보다 빠르게 일정을 수립하게 위하여 휴리스틱 알고리즘을 적용한 방법론을 제안 하고자 한다. 이는 보다 수월하게 현장의 상황을 반영하여 일정 계획을 최적화 할 수 있다.

궤도 유지보수 일정 계획의 목적은 주어진 기간 내에 최대한 많은 작업을 수행하기 위하여 작업을 언제 어떤 리소스에 할당할 것인가를 결정하는 것이다. 일반적으로 NP-hard인 일정계획 문제를 최적화하기 위하여 최근에 개발된 Binary Artificial Bee Colony Algorithm (BABC)를 적용 하였다.

본 논문의 2절에서는 궤도 유지보수 일정계획 문제를 설명하고, 목적식과 제약조건을 설정하였다. 3절에서는 BABC에 대하여 설명 하였고, 4절에서는 궤도 유지보수 일정계획 문제에 BABC를 적용하는 메커니즘에 대하여 자세히 서술 하였다.

2. 궤도 유지보수 일정계획 문제

궤도유지보수 분야의 인력, 장비 등 각종 리소스의 능력과 일정을 감안하여 가장 효율적이고 효과적인 유지보수 계획을 수립하는 것을 유지보수 일정최적화라 한다[4]. 효율적이고 효과적인 유지보수 계획이란 여러 가지로 설명될 수 있다. 유지보수를 수행하는 단체의 목적이나 상황에 따라 그 기준은 달라 질 것이다. 하지만 한정된 리소스를 이용하여 적절한 시기에 더 많은 작업을 수행하는 것은 유지보수 일정 최적화의 가장 일반적이고 범용적인 목적이라 할 수 있다.

† 비회원, 강원대학교, 산업공학과, 석사과정
E-mail : stir0221@hanmail.net
TEL : (033)250-6280 FAX : (033)255-6281
* 비회원, 강원대학교, 산업공학과, 정교수
* 비회원, 강원대학교, 산업공학과, 정교수
** 정회원, 철도공사, 팀장
** 정회원, 철도공사, 차장
*** 비회원, (주)휴니드테크놀로지스, 수석연구원

현재 여러 분야에서 사용되고 있는 자원 할당문제나 일정계획 문제는 궤도 유지보수 일정 최적화 문제의 한 방법이 될 수 있다. 본 논문에서 다루고 있는 궤도 유지보수 일정 계획의 문제에서는 단일 공정 작업을 수행할 수 있는 다수개의 리소스가 존재한다. 다수의 리소스가 수행할 수 있는 작업들은 각각 다른 작업 시간을 가지고 있으므로, 일정 계획 문제의 병렬 기계 문제 중 비동일 병렬기계(non-identical parrallel machine) 문제에 해당한다[9]. 병렬 기계의 일정계획에 대한 연구는 60년대 이후부터 활발히 진행 되어 왔다. Li[10]는 비동일 병렬기계의 총 가중 완료시간을 최소화 하는 문제의 모델에 대하여 정리 하였다.

하지만 궤도유지보수 분야는 현장의 많은 상황들이 제약으로 존재하고 있고, 이를 기존의 문제에 적용하기에는 많은 어려움이 있다. 또한 현장의 종류나 특성에 따라 현장의 상황이 모두 다르기 때문에 적용되어야 할 제약에도 차이가 있다. 본 논문에서는 현장 조사와 관련 자료의 분석을 통하여 각 현장에서 공통적으로 반영되어야 할 제약에 대하여 정의 하였다. 이는 각 현장의 상황에 맞는 일정계획의 기본이 될 수 있으며, 정의된 제약을 적용한 방법론 역시 현장의 상황에 따라 발생하는 제약을 적용할 수 있는 바탕이 될 수 있다.

2.1 파라미터 설정

먼저 목적 함수와 제약식을 표현하기 위한 결정 변수와 파라미터를 설정한 후 목적 함수와 제약식에 대하여 기술 한다.

■ 목적 함수에 이용된 결정변수

$$x_{jtr} = j \text{ 작업이 } t \text{ 기간에 } r \text{ 자원을 이용하여 실시되면 } 1, \text{ 아니면 } 0$$

■ 제약 함수에 이용된 결정변수

$$ld_{rt} = r \text{ 자원이 } t \text{ 기간에 실시 할 작업 중 가장 먼 거리에 위치한 작업 구간의 종점}$$

$$sd_{rt} = r \text{ 자원이 } t \text{ 기간에 실시 할 작업 중 가장 가까운 거리에 위치한 작업 구간의 시점}$$

■ 제약 함수에 이용된 파라미터

$$j = \text{작업 번호}(1 \leq j \leq J)$$

$$t = \text{기간 인덱스}(1 \leq t \leq T)$$

$$r = \text{자원(resource) 번호}(1 \leq r \leq R)$$

$$F_j = j \text{ 작업 구간의 시점}$$

$$T_j = j \text{ 작업 구간의 종점}$$

$$D_j = j \text{ 작업의 완료 납기일}$$

$$P_{rj} = r \text{ 자원을 이용하여 } j \text{ 작업을 수행할 경우 수행 시간(단위 : 분)}$$

$$SR_j = j \text{ 작업을 수행 할 수 있는 자원들의 집합}$$

$$SR_r = r \text{ 자원을 이용하여 수행할 수 있는 모든 작업들의 집합}$$

$$C_{rt} = r \text{ 자원의 } t \text{ 기간 작업 가능 시간(단위 : 분)}$$

$$RA_h = \text{인력 자원의 작업 여유 시간(준비시간 + 이동시간 + 정리시간)}$$

$$RA_m = \text{장비 자원의 작업 여유 시간(준비시간 + 이동시간 + 정리시간)}$$

$$SR_m = \text{장비 자원들의 집합}$$

$$SR_h = \text{인력 작업들의 집합}$$

$LT_r =$ r 자원의 최대 이동 가능 거리, 인력 리소스만 해당 됨

2.2 목적함수

작업 시기를 놓친 구간은 이상상태의 진전으로 인하여 많은 추가비용을 발생 시킨다. 뿐만 아니라 할당된 작업을 모두 수행하지 못할 경우 추가 인력이나 장비에 대한 비용이 발생한다. 케도 유지보수 일정 최적화 문제의 목적함수는 비용최소화이다. 따라서 적절한 시기에 최대한 많은 작업을 수행하는 것은 이와 일맥상통 한다 할 수 있다[1].

■ 목적 함수

$$\text{Maximize } \sum_{j=1}^I \sum_{t=1}^T \sum_{r=1}^R x_{jtr}$$

2.3 제약조건

케도 유지보수 작업에는 일정 계획 시 고려해야하는 많은 현실적인 상황이 존재 한다. 케도 유지보수 일정 최적화를 위하여 현장의 일반적인 상황을 제약으로 표현 하였다.

■ 보수 작업 할당 제약[1]

각 보수 작업은 일정 수립기간 중 단 하루에만, 그리고 단 하나의 리소스에만 할당 되어야 한다는 제약.

$$\sum_{t=1}^T \sum_{r=1}^R x_{jtr} \leq 1, \quad \forall j$$

■ 해당 자원 이용 제약[1]

각 보수 작업은 해당 작업을 수행 할 수 있는 자원에 의해서만 수행 될 수 있다는 제약.

$$\sum_{t=1}^T \sum_{r \in SR_j} x_{jtr} \leq 1, \quad \forall j$$

$$\sum_{t=1}^T \sum_{r \notin SR_j} x_{jtr} = 0, \quad \forall j$$

■ 완료 요구일 제약[1]

할당된 모든 작업은 완료 요구일 안에 이루어 져야 한다는 제약.

$$\sum_{t \leq D_j} \sum_{r=1}^R x_{jtr} \leq 1, \quad \forall j$$

$$\sum_{t > D_j} \sum_{r=1}^R x_{jtr} = 0, \quad \forall j$$

■ 인력 작업의 용량 제약

인력 작업 시 하루 최대 이동거리 내에서 모든 작업이 이루어 져야 하고, 작업 준비 시간과 작업 정리시간 그리고 작업개소 간 이동시간을 포함한 하루의 작업시간은 최대 작업시간 내에 이루어 져야 한다는 제약.

$$\sum_{j \in SR_r} P_{jr} \cdot x_{jtr} \leq \max(C_{rt} - RA_h, 0) \quad \forall j, \forall t$$

$$ld_{rt} - sd_{rt} \leq LT_r$$

■ 장비 자원의 용량 제약

작업 준비 시간과 작업 정리시간 그리고 작업을 위한 총 이동 시간을 포함한 하루의 작업 시간은 최대 작업시간 내에 이루어 져야 한다는 제약.

$$\sum_{j \in SJR_r} P_{jr} \cdot x_{jtr} \leq \max(C_{rt} - RA_m, 0) \quad \forall j, \forall t$$

3. BABC 메커니즘 설명 [2, 3]

휴리스틱 알고리즘은 최적 해를 보장할 수는 없지만, 빠른 시간 내 또는 사용자가 제시하는 시간 내에 최적해/최선해 제시 할 수 있을 뿐만 아니라 다양한 문제를 보다 쉽게 적용하여 풀이할 수 있다는 장점도 가지고 있다. 그 중에서 Artificial Bee Colony Algorithm (ABC)은 2005년 이후부터 활발히 연구가 진행되고 있고 그 성능에 대하여 입증되었으며 자원할당과 일정계획 분야에도 그 연구가 활발히 진행 중이다[11, 12, 13].

본 논문은 앞서 말한 ABC를 활용한 Binary ABC를 케도 유지보수 일정계획 문제에 적용 하여 어떻게 최적해/최선해를 탐색해 낼 수 있는지 설명하였다. <그림1>은 Binary Artificial Bee Colony Algorithm을 이용한 케도 유지보수 일정계획의 흐름도를 나타낸 것이다.

단계1에서는 Food source , Employed bee, Onlooker bee의 수, 이웃해 탐색 수, 종료 조건을 결정한다. 단계2에서는 초기 가능해군을 생성하기 위하여 특정 자원이 특정 작업에 할당 되었을 때는 1, 아니면 0으로 하는 초기 해(BABC에서 food source)들을 생성하여 초기해 군(initial population)을 생성한다. 초기 해들을 생성할 때, 몇 개의 해를 생성할 것인지가 중요한 요소인데, 되도록 많이 생성하면 해 탐색의 다양성을 고려할 수 있으나 상대적으로 최적해 탐색을 위한 계산시간이 많이 소요된다. 단계3에서는 각 초기 해에 대하여 Employed bee를 결정하여 각 초기 해들의 지역탐색을 위하여 이웃 해들을 탐색한다. 이 단계에서는 이웃 해를 어느 범위까지로 할 것인지 정확한 정의가 필요하고 어떻게 몇 개의 이웃 해를 생성할 것인지 중요한데, 본 논문에서는 해 표현에서 임의로 한 부분을 선택하여 “0” 이면 “1로,” 1 “이면 ” 0 “으로 수정하여 이웃해를 생성한다. 각 해에 대하여 더 좋은 해가 생성되면 더 좋은 해로 업데이트 한다. 단계4에서는 전역 해 탐색을 위하여 단계3의 모든 해의 이웃해 탐색 후 갱신된 초기 해군의 모든 해를 사용하여 Onlooker bee가 평가 값에 비례하는 확률로 새로운 해(food source)를 생성하여 해 군에 합류시키고 이 해의 평가 값을 계산하여 이 해보다 나쁜 해는 탈락시킨다. 단계5에서는 탈락시킨 해 수 만큼의 해들을 Scout bee를 통하여 임의적으로 생성하여 지역 해(local solution)에 빠지지 않고 다양한 해 탐색을 추구함으로써 전역 해(global solution)를 탐색할 수 있도록 한다. 단계2, 3, 4, 5를 진행할 때 생성된 해가 가능해(feasible solution)인지 확인하고 가능해가 아닐 경우는 필요시 수정 보완하여 가능해로 보정한다. 이 모든 단계 수행 후 종료 조건(세대 수, 사용자가 제시한 시간 제한 등)에 맞으면 종료하고 그렇지 않으면 Employed bee의 이웃해 생성부터 반복 시행한다.

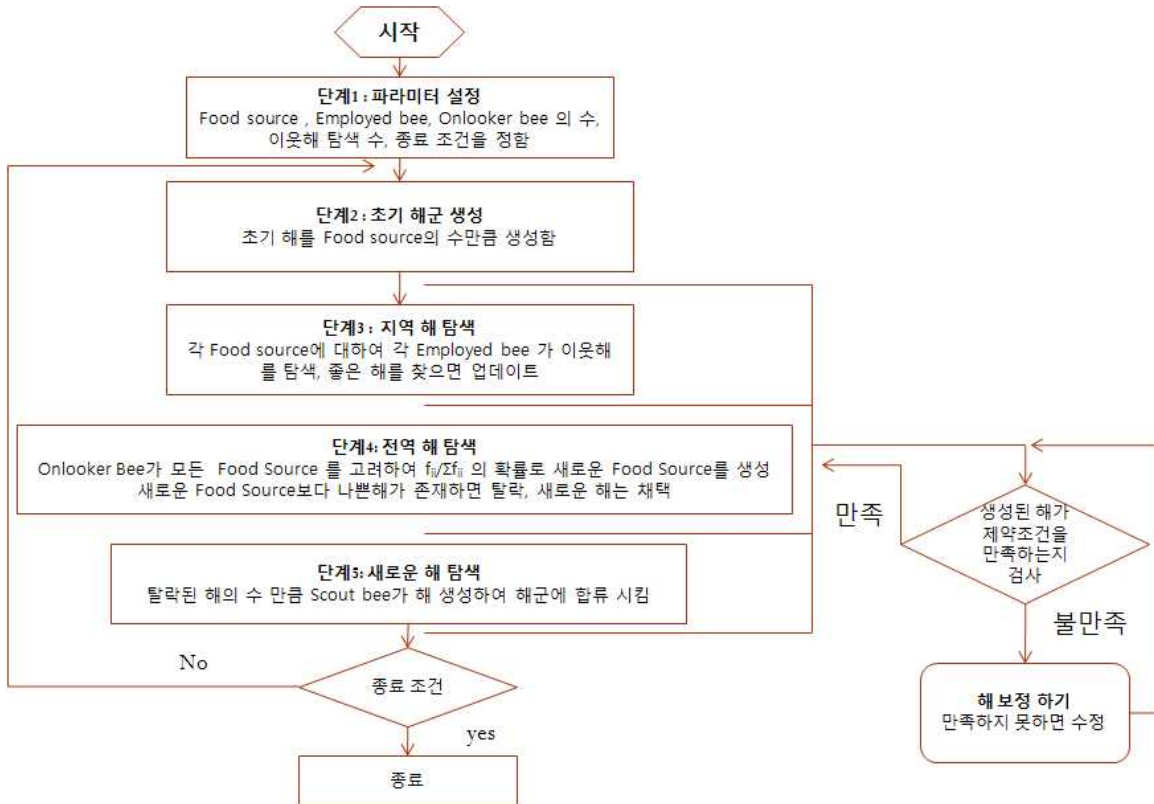


그림1. Binary Artificial Bee Colony Algorithm을 이용한 궤도 유지보수 일정계획의 흐름도

4. 궤도 유지 보수 일정 문제에 BABC 적용

본 절에서는 3절에서 설명한 BABC메커니즘을 간단한 예제를 이용하여 궤도 유지보수 일정최적화 문제에 어떻게 적용할 수 있는지에 대하여 기술 하였다. 예제는 한국 철도 공사 오송 시설사업소의 고속선 궤도 보수이력 데이터의 형식을 바탕으로 하여 설명 하였다.

4.1 궤도 유지보수 일정 예제 문제

궤도 유지보수 작업은 각각 작업 종류와 완료 요구일, 작업구간 그리고 작업시간이 존재 한다. 작업 종류는 궤도의 이상 상태에 따라 결정된다. 그 종류로는 인력템핑, 장비템핑(Spot, 총다지기 등), 자갈정리, 장대 재설정, 레일 육성 용접 등 여러 가지가 있지만 인력 및 장비 템핑이 작업량의 대부분을 차지하고 있다. 이러한 작업들은 그 종류별로 수행할 수 있는 리소스의 종류가 정해져 있다. 그러므로 일정 계획 결과 작업에 할당되는 리소스는 작업 종류와 수행 가능한 리소스와의 관계에 의하여 결정된다고 할 수 있다.

완료 요구일은 작업이 수행 되어야 하는 최종일로써 리소스의 용량이 허용하는 한도 내에서는 반드시 완료 요구일 이내에 작업이 이루어 져야 한다. 작업구간은 선로점검차량 또는 선로점검자의 궤도 점검 결과 보수가 필요하다고 생각되는 구간의 시점과 종점을 의미 한다. 시점부터 종점까지의 거리를 연장이라 하며, 이는 작업의 시간과도 관련이 있다. 대부분의 작업 시간은 작업종류별 작업속도와 작업의 연장을 고려하여 정해지는데 궤도의 특성이나, 작업 환경 등을 고려하여 유동적으로 설정 될 수 있다. 일정 수립 기간은 작업을 수행 하고자 하는 기간을 의미하는데, 이 기간에 완료요구일이 포함되어 있는 작업들은 일정수립의 대상이 된다.

아래는 예제 문제의 정보를 정리한 데이터 이다.

■ 일정 수립 기간

3월1일 ~ 3월2일

■ 가용 리소스

장비 - MTT_01(1대), STT_01(1대)

인력 - 인력1팀(1개팀)

■ 리소스 별 가능 작업

MTT_01 - 장비템핑, 자갈치기

STT_01 - 자갈치기

인력1팀 : 인력템핑

도표1. 예제 데이터

작업번호	작업종류	시점(m)	종점(m)	연장(m)	작업시간 (분)	완료요구일	완료요구일 (net)
J1	인력템핑	10	13	3	30	3월2일	2
J2	인력템핑	5	9	4	40	3월2일	2
J3	인력템핑	1,927	1,930	3	30	3월2일	2
J4	인력템핑	422	432	10	80	3월2일	2
J5	인력템핑	2,500	2,575	25	100	3월2일	2
J6	인력템핑	617	620	3	30	3월2일	2
J7	장력템핑	3,000	3,080	80	80	3월2일	2
J8	장비템핑	105	135	30	30	3월2일	2
J9	자갈치기	990	1,000	10	10	3월2일	2
J10	자갈치기	305	310	5	5	3월2일	2
J11	인력템핑	3,044	3,053	9	90	3월2일	2
J12	장비템핑	200	219	19	20	3월2일	2
J13	자갈치기	3,077	3,097	20	40	3월2일	2
J14	인력템핑	2,233	2,243	10	30	3월2일	2
J15	장비템핑	3,120	3,125	5	50	3월1일	1
J16	자갈치기	1,997	2,020	23	100	3월1일	1

4.2 파라미터 초기화

문제 풀이에 필요한 파라미터들을 미리 설정한다. BABC알고리즘에 문제를 적용하기 위해서 Food source와 Employed bee, Onlooker bee의 개수, 그리고 Limit와 Termination criterion의 조건이 필요하다.

■ Food Source (가능해)

최적해 탐색을 위하여 표현할 해 집합의 개수(SN)로써 본 문제에서는 3으로 설정

■ i

생성된 Food source의 인덱스 ($1 \leq i \leq SN$)

■ Employed bee, Onlooker bee

해 탐색 결과의 수(Food source의 수와 동일)로 본 문제에서는 3으로 설정

■ Limit

Food source탐색의 한계(이웃해 탐색의 종료조건)로 본 문제에서는 3으로 설정

■ Termination criterion

해 탐색 종료조건으로 본 문제에서는 100세대로 설정

4.3 해 표현

케도 유지보수 일정 최적화 문제의 의사결정변수 x_{jtr} 를 표현하기 위해서는 3차원 배열 형식의 해 표현 방법이 필요하다. 따라서 본 논문에서는 작업 별 가용일과 작업 별 리소스라는 두개의 의미를 담을 수 있도록 3차원의 의사결정변수를 표현 하였으며, 이는 문제풀이에도 동일하게 적용된다.

■ BABC에 사용된 결정변수

$$x_{jtr} = j \text{ 작업이 } t \text{ 기간에 } r \text{ 리소스를 이용하여 실시되면 } 1, \text{ 아니면 } 0$$

리소스	완료일	작업															
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
M T T ₀₁	1일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}
	2일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}
S T T ₀₁	1일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}
	2일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}
이택1팀	1일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}
	2일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	x_{j7}	x_{j8}	x_{j9}	x_{j10}	x_{j11}	x_{j12}	x_{j13}	x_{j14}	x_{j15}	x_{j16}

그림2. BABC를 적용한 케도 유지보수 일정 최적화 문제의 해 표현 방법

위와 같은 해 표현 기법을 통하여 해 표현 가능 영역과 불가능 영역을 설정할 수 있다. 해 표현 불가능 영역이란 작업을 할당 할 수 없는 날이나 사용 할 수 없는 리소스의 영역에 의사결정변수 x_{jtr} 이 아닌 0의 값을 미리 할당하는 영역을 말한다. 이는 완료 요구일 제약이나 해당 자원 이용 제약뿐만 아니라 특정일 또는 특정 리소스에만 작업을 할당해야 하는 현장의 제약들을 반영할 수 있는 방법 중 하나이다.

리소스	완료일	작업															
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
M T T ₀₁	1일	0	0	0	0	0	0	x_{j7}	x_{j8}	x_{j9}	x_{j10}	0	x_{j12}	x_{j13}	0	x_{j15}	x_{j16}
	2일	0	0	0	0	0	0	x_{j7}	x_{j8}	x_{j9}	x_{j10}	0	x_{j12}	x_{j13}	0	0	0
S T T ₀₁	1일	0	0	0	0	0	0	0	0	x_{j9}	x_{j10}	0	0	x_{j13}	0	0	x_{j16}
	2일	0	0	0	0	0	0	0	0	x_{j9}	x_{j10}	0	0	x_{j13}	0	0	0
이택1팀	1일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	0	0	0	0	x_{j11}	0	0	x_{j14}	0	0
	2일	x_{j1}	x_{j2}	x_{j3}	x_{j4}	x_{j5}	x_{j6}	0	0	0	0	x_{j11}	0	0	x_{j14}	0	0

그림3. 완료 요구일 제약과 해당자원 이용 제약을 반영한 해 표현 가능 영역과 불가능 영역

4.4 초기해 생성

각 작업별로 해 표현 가능 영역중 하나의 영역을 랜덤하게 선택 한다. 예를 들어 J1의 경우 5,6행의 1열의 영역 중 하나, J10의 경우 1,2,3,4행 10열의 영역중 하나를 선택 하게 된다. 선택된 영역은 1의 값을 가지게 되고 해 표현 불가능 영역을 포함한 나머지 영역은 0의 값을 갖게 된다.

일반적으로 휴리스틱 알고리즘에서 초기해의 생성은 결과 값의 품질과 풀이 시간적인 측면에서 매우 중요한 역할을 한다. 특히 궤도 유지보수 일정 최적화의 경우 리소스의 용량보다 작업의 양이 많아 모든 작업을 할당 할 수 없는 경우가 발생하는데 이때 할당의 비율은 매우 중요한 역할을 한다. 이는 작업의 해 표현 가능영역에 대부분 1의 값이 존재하지만 그렇지 않은 경우도 있을 수 있음을 의미한다. 따라서 본 논문에서는 다음의 확률로 작업이 어떠한 날에도 어떠한 리소스에도 할당되지 않도록 제안한다.

■ 작업이 할당되지 않을 확률

$$= \max\{0.05, (\text{모든 작업 작업시간의 합} - \text{전체 리소스 용량}) / \text{모든 작업 시간의 합}\}$$

$$SRh_{size} = SR_h \text{의 크기}$$

$$SRm_{size} = SR_m \text{의 크기}$$

$$\text{모든 작업 시간의 합} = \sum_{j=1}^J \left(\sum_{r=1}^R \frac{P_{rj}}{R} \right)$$

$$\text{모든 리소스 용량의 합} = \left(\sum_{r=1}^R \sum_{t=1}^T C_{rt} \right) - (RA_h \cdot SRh_{size} + RA_m \cdot SRm_{size})$$

초기해 생성의 절차는 다음과 같다.

단계 1 : 작업별로 해 표현 가능 영역 중 랜덤하게 하나를 선택, 단 리소스의 용량보다 작업의 양이 많아 작업이 할당되지 않는 경우도 고려하여 확률적으로 하나의 영역도 선택하지 않을 경우 생성

단계 2 : 선택된 영역은 1의 값 할당

단계 3 : 선택되지 않은 영역과 해 표현 불가능 영역은 0의 값 할당

단계 4 : 마지막 작업이면 단계5, 그렇지 않으면 단계 1로

단계 5 : 마지막 Food source이면 종료, 그렇지 않으면 단계1로 이동 후 다음 Food source에 적용

4.5 해 보정 방법

위의 해 표현 방법을 통하여 보수작업 할당 제약, 완료 요구일 제약 그리고 해당 자원 이용 제약을 만족 하였지만, 장비와 인력의 용량 제약은 만족하지 못했다. 따라서 장비와 인력 리소스의 용량 제약과 같이 해 표현 불가능 영역을 통하여 표현하기 어려운 제약들은 해의 값을 보정하여 그 제약을 만족 시킬 수 있다.

인력 리소스의 용량 제약 중 작업 거리에 대한 해 보정 절차는 다음과 같다.

단계 1 : 할당된 해의 리소스 r에 대하여 t날 할당된 작업들 중 ld_{rt} 와 sd_{rt} 찾기, loop = 0

단계 2 : $ld_{rt} - sd_{rt}$ 의 값을 구하여 LT_r 보다 크면 loop에 1을 더한 후 단계 3으로, 그렇지 않으면 단계 4로

단계 3 : loop가 홀수이면 ld_{rt} 에 할당된 작업 x_{jtr} 의 값을 0으로, loop가 짝수이면 sd_{rt} 에 할당된 작

업 x_{jtr} 의 값을 0으로 할당한 후 단계 1로

단계 4 : $t = T$ 이면 단계 5로, 그렇지 않으면 단계 1로 이동 후 $t = t+1$

단계 5 : $r = R$ 이면 종료, 그렇지 않으면 단계 1로 이동 후 $r = r+1, t = 1$

장비와 인력 리소스의 용량 제약 중 작업 시간에 대한 해 보정 절차는 다음과 같다.

단계 1 : 할당된 해의 리소스 r 에 대하여 t 날 할당된 모든 작업들의 작업시간 P_{rj} 를 합을 구하기

단계 2 : 리소스 r 의 하루 최대 가용 시간(인력이면 $C_{rt} - SR_h$, 장비이면 $C_{rt} - SR_m$)을 구한 후

P_{rj} 의 합 과 비교하여, P_{rj} 이 더 크면 단계 3으로, 그렇지 않으면 단계 4로

단계 3 : 리소스 r 에 대하여 t 날 할당된 모든 작업들 중 작업시간이 가장 큰 작업 x_{jtr} 의 값을 0으로 할당한 후 단계 1로

단계 4 : $t = T$ 이면 단계 5로, 그렇지 않으면 단계 1로 이동 후 $t = t+1$

단계 5 : $r = R$ 이면 종료, 그렇지 않으면 단계 1로 이동 후 $r = r+1, t = t+1$

Food source01																
리소스	완료구	작업														
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15
M TT _01	1일	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
	2일	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
S TT _01	1일	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
	2일	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
인력 팀	1일	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	2일	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Food source02																
리소스	완료구	작업														
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15
M TT _01	1일	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0
	2일	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
S TT _01	1일	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	2일	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
인력 팀	1일	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
	2일	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

Food source03																
리소스	완료구	작업														
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15
M TT _01	1일	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2일	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
S TT _01	1일	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
	2일	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
인력 팀	1일	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
	2일	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

그림4. 초기해 보정 결과

4.6 평가 값 구하기

생성된 해에 대하여 평가 값 fit_i 을 구한다. 평가 값은 위에서 언급한대로 다음의 식을 따른다.

$$Maximize \sum_{j=1}^I \sum_{t=1}^T \sum_{r=1}^R x_{jtr}$$

[그림4]의 초기해 보정결과로 알 수 있듯이 Food source01의 평가 값 fit_1 은 12, Food source02의 평가 값 fit_2 은 11이고 Food source03의 평가 값 fit_3 은 7이다.

4.7 이웃 해 탐색

Employed bee에 의한 이웃 해 탐색은 할당 된 Food source 각각의 결과를 바탕으로 더 좋은 해를 탐색하기 위한 방법이다.

이웃 해 탐색의 절차는 다음과 같다.

- 단계 1 : 자기 자신을 제외한 Food source 중 하나를 랜덤하게 선택
- 단계 2 : 선택된 Food source의 작업 중 하나를 랜덤하게 선택
- 단계 3 : 선택된 작업에 할당된 x_{jtr} 의 값을 자기 자신의 작업에 x_{jtr} 값으로 변환
- 단계 4 : 해 보정 작업(4.4 해 보정하기) 수행
- 단계 5 : 기존의 평가 값보다 좋으면 채택, 그렇지 않으면 탈락
- 단계 6 : 반복 수(Limit) 만큼 실행 했으면 단계7, 그렇지 않으면 단계 1로
- 단계 7 : 자기 자신이 마지막 Food source(SN)이면 종료, 그렇지 않으면 단계 1로



그림5. Food source01의 이웃해 탐색 과정

4.8 전역 해 탐색

Onlooker bee의 전역 해 탐색은 모든 Food source의 결과 값을 바탕으로 새로운 해를 탐색하는 것이다. 새로운 해의 값은 작업마다 할당된 모든 x_{jtr} 에 대하여 확률적으로 Food source를 선택하여 선택된 Food source의 x_{jtr} 의 값을 따른다. Food source를 선택하는 확률은 다음과 같다.

■ 전역 해 탐색 과정에서 작업 별로 Food source(j)가 선택 될 확률(P_j)

$$P_j = \frac{fit_j}{\sum_{n=1}^{SN} fit_n}$$

위와 같은 확률을 이용하여 새롭게 찾아낸 값에 대하여 가능해 인지 확인 후, 그렇지 않으면 다시 한번 보정 작업을 거치고 평가 값을 구한다. 그리고 새로운 평가 값 보다 좋은 값은 유지 하고, 그렇지 않은 값들은 다음의 식의 의하여 낮은 값부터 버려 진다.

■ 전역 해 탐색 과정에서 버려질 평가 값의 개수

TC = Termination criterion

l = Termination criterion의 인덱스($1 \leq l \leq TC$)

LF = 새로운 평가 값보다 좋지 않은 평가 값들의 수

$$\frac{(TC-l)}{TC} \cdot LF$$

4.9 새로운 해 탐색

위의 단계에서 버려진 해의 수만큼 Scout bee에 의하여 새로운 해를 탐색한다. 새로운 해를 탐색하는 방법은 4.4의 초기해 생성과 동일하다.

4.10 반복

이 모든 단계 수행 후 종료 조건(Termination criterion, 사용자가 제시한 시간제한 등)에 맞으면 종료하고 그렇지 않으면 이웃해 생성부터 반복 시행한다.

5. 결론

본 논문에서는 케도 유지보수 일정 계획 문제에 대하여 목적식과 제한식의 수리 모형을 제시 하였다. 그리고 BABC에서 Employed bee에 의한 이웃해 탐색과 Onlooker bee에 의한 전역 해 탐색 Scout bee에 의한 새로운 해 탐색 메커니즘을 설명 하였다. 이 메커니즘을 이용하여 BABC를 적용한 케도 유지보수 일정 계획 예제를 만들었고, 이를 통하여 해 표현과 초기해 생성, 해 보정 방법, 그리고 해 평가를 설명 하였다. 추후 다양한 문제의 시뮬레이션을 통하여 본 논문에서 제시하는 방법을 검증할 것이다.

참고문헌

1. 김대영, 이성근, 이기우, 우병구, 이성욱, 김기동, “케도 유지보수를 위한 틀림진진 예측 및 일정최적화” 한국철도학회 2008년도 춘계 학술대회 논문집, 2008 June 12, pp.1359 - 1370
2. 김성수, 변지환, “Banary artificial bee colony를 사용한 센서 네트워크 최적 설계,” working paper, 강원대학교 산업공학과, 2010
3. 김성수, 변지환, “Banary artificial bee colony를 사용한 Reporting Cell 위치 추적 관리 시스템 최적 설계,” working paper, 강원대학교 산업공학과, 2010
4. 박지호, “유전자 알고리즘을 이용한 고속선 케도 유지보수 일정계획에 관한 연구,” 강원대학교 석사 학위 논문, 2009
5. 한국철도공사, “고속선 케도관리 의사결정지원 시스템 개발 : 2차년도 중간 보고서,” 2008
6. 한국철도공사, “고속선 케도관리 의사결정지원 시스템 개발 : 3차년도 중간 보고서,” 2009
7. Chao-Tang Tseng and Ching-Jong Liao, “A discrete particle swarm optimization for lot-streaming flowshop scheduling problem,” European Journal of Operational Research, Vol.191, pp.360-373, 2008
8. Chin Soon Chong, Malcolm Yoke Hean Low, Appa Iyer Sivakumar and Kheng Leng Gay, “A Bee Colony Optimization Algorithm to Job Shop Scheduling” Proc. 2006 Winter Simulation Conf, pp1954 - 1961, 2006
9. Higgins, A. (1998), “Scheduling of Railway Track Maintenance Activities and Crews,” The Journal of the Operational Research Society, Vol.49, pp.592-623, 1998
10. Li, Kai, Yang, Shan-Lin, “Non-identical parallel-machine scheduling research with minimizing

total weighted completion time: Models, relaxation and algorithms, “ Applied Mathematical Modelling, Article in Press, 2008

11. Karaboga, D and Akay, B.1, "A comparative study of Artificial Bee Colony algorithm," Applied Mathematics and Computation, Vol.214, No.1, pp108-132, 2009
12. Karaboga, Dervis, Basturk and Bahriye, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, Vol.39, No.3, pp459-471, 2007
13. Karaboga, D and Basturk, B., "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, Vol.8, No.1, pp 687-697, 2008