

Artificial Bee Colony 기법을 이용한 철도궤도 유지보수 일정계획 수립 연구

Railway Track Maintenance Scheduling using Artificial Bee Colony

남덕희¹ · 김기동¹ · 이성욱² · 김성수[†]

Duk-Hee Nam · Ki-Dong Kim · Sung-Uk Lee · Sung-Soo Kim

Abstract The objective of this paper is to propose a fast and easy Binary Artificial Bee Colony (BABC) heuristic algorithm to optimize NP-hard scheduling problem of railway track maintenance considering real conditions. The optimal or best solutions can be found using proposed BABC within very short or user specified computation time. We can greatly maximize the objective value using this proposed method in 30, 60, 100 and 200 work size railway track maintenance scheduling problems for experiment and analysis.

Keywords : Railway track maintenance, Artificial Bee Colony

초 록 본 논문의 목적은 보다 빠르고 수월하게 현장의 상황을 반영하여 NP-hard 철도 궤도 유지보수 일정계획을 최적화할 수 있는 Binary Artificial Bee Colony Algorithm (BABC) 휴리스틱 알고리즘을 제안하는 것이다. 본 논문에서 제시하는 방법을 사용하여 빠른 시간 또는 사용자가 제시하는 시간 내에 최적해 또는 최선 해를 제시할 수 있다. 본 논문에서 제안하는 방법을 사용하여 30, 60, 100, 200개 작업의 궤도 유지보수 일정계획 문제의 실험 및 분석을 통하여 목적식 값을 최대화할 수 있었다.

주요어 : 궤도 유지보수 일정계획, Artificial Bee Colony

1. 서 론

1.1 문제정의와 연구의 목적

궤도유지보수 분야의 인력, 장비 등 각종 자원의 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. 그러나 결국 한정된 자원을 이용하여 적절한 시기에 더 많은 작업을 수행하는 것은 유지보수 일정 최적화의 가장 일반적이고 범용적인 목적이라 할 수 있다.

현재 철도 궤도의 유지보수 작업 현장은 각자의 성격에 따라 다양한 방법을 통하여 이루어 지고 있다. 따라서 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.

휴리스틱 알고리즘은 최적 해를 보장할 수는 없지만, 빠

른 시간 내 또는 사용자가 제시하는 시간 내에 최적해 또는 최선 해를 제시 할 수 있다. 뿐만 아니라 다양한 문제를 보다 쉽게 적용하여 풀이할 수 있다는 장점도 가지고 있다. 따라서 다양한 철도 궤도 유지보수 현장의 다양한 상황들도 보다 수월하게 반영하여 최적화 할 수 있다. 그 중에서 Artificial Bee Colony Algorithm (ABC)은 2005년 이후부터 활발히 연구[9]가 진행되고 있고 그 성능에 대하여 입증되었으며 자원할당과 일정계획 분야에도 그 연구가 활발히 진행 중이다[2,3].

본 논문의 목적은 철도궤도 유지보수 일정 계획 문제를 수리모델화하고 BABC메커니즘을 한국 철도 공사의 고속선 궤도 보수이력 데이터[7,8]를 사용하여 궤도 유지보수 일정최적화 문제의 최선 해를 찾아내는 것이다. 본 논문의 2.1절에서는 궤도 유지보수 일정계획 문제를 설명하고, 목적식과 제약조건의 수리모델을 설정 하였다. 2.2절과 2.3절에서는 궤도 유지보수 일정계획 문제에 BABC를 적용하는 메커니즘에 대하여 서술 하였다. 그리고 3절에서 BABC를 이용하여 일정계획 문제를 풀이하였고 그 결과를 최적화 하고 분석 하였다.

1.2 관련 연구 현황

현재 여러 분야에서 사용되고 있는 자원 할당문제나 일정 계획 문제는 궤도 유지보수 일정 최적화 문제의 한 방법이 될 수 있다. 본 논문에서 다루고 있는 궤도 유지보수 일정

[†]교신저자 : 강원대학교 공과대학 산업공학과
E-mail : kimss@kangwon.ac.kr

¹강원대학교 공과대학 산업공학과

²한국철도공사

계획의 문제에서는 단일 공정 작업을 수행할 수 있는 다수 개의 자원이 존재 하므로 일정 계획 문제의 parallel machine problem (병렬 기계 문제) 문제에 해당한다[10]. 다수의 기계에서 각각의 할당 작업과 작업의 처리 순서를 결정하는 Nonpreemptive parallel scheduling problem 중 Completion time을 최대화 하는 문제는 NP-hard로 알려져 있고[11], 이는 본 논문에 풀이하고자 하는 일정 계획 문제와 일맥상통한다 할 수 있다.

병렬 기계의 일정계획에 대한 연구는 60년대 이후부터 활발히 진행 되어 왔다. Li[12]는 비 동일 병렬기계의 총 가중 완료시간을 최소화 하는 문제의 모델에 대하여 정리 하였으며, McNaughton[13]은 동일 병렬기계에서 최대 작업 완료 시간(makespan)을 최소화 하는 알고리즘을 개발 하였다.

궤도 유지보수 일정 계획 문제와 관련하여 오석문[5]은 도상 다짐작업의 장기 일정계획 문제에 대하여 상용 프로그램에 탑재할 수 있는 모형을 제시 하였고, Masashi MIWA[14]는 MTT의 도상 다짐작업의 일정계획 문제를 유지 보수 비용의 최적화 측면에서 풀이 할 수 있는 모형을 제시 하였다.

그러나 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. 문제 분석 및 알고리즘 개발

2.1 궤도 유지보수 일정계획 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 보통 그 종류별로 수행할 수 있는 자원의 종류가 정해져 있다. 인력 템핑은 일반적으로 인력 팀이 작업을 수행한다. 장비 템핑은 Spot 템핑과 총다지기로 구분할 수 있는데 이 둘을 구분하는 정확한 기준은 존재하지 않는다. 하지만 현장에서는 일반적으로 작업의 연장이 비교적 긴 경우 전체 도상다짐을 위해 총다지기를 수행하고 작업의 연장이 짧고, 부분적인 도상다짐이 필요한 경우 Spot 템핑을 수행 한다[7].

총다지기는 보통 MTT(Multiple Tie Temper)를 통하여 작업을 수행 한다. MTT란 자갈궤도의 궤도틀림 정정 작업에 가장 많이 사용되는 도상 다지기 용 장비이다. MTT는 보통 유압을 통한 Temping Bar의 상하 작용을 통하여 수평 방향의 다지기 작업을 수행한다. Spot 템핑은 보통 STT(Switch

Tie Temper)를 통하여 작업을 수행한다. STT는 원래 선로의 3대 취약점(곡선부, 이음매부, 분기부)의 하나인 분기부를 다지기 위한 장비이다. 그러나 분기기뿐 아니라 일반적인 구간에서도 작업이 가능하므로 현장에서는 전 구간에서 유용하게 사용 되고 있다[6].

이처럼 궤도 유지보수 현장에서는 각 작업종류에 따라 이를 수행할 수 있는 자원이 일반적으로 정의되어 있지만 작업 환경이나 작업자의 경험에 따라 다르게 설정 할 수도 있다. 예를 들어 Spot 템핑은 일반적으로 STT장비가 수행 하지만 작업자의 의도에 따라 인력자원으로 혹은 STT 또는 인력 자원 중 어떠한 것이 할당되어도 좋다고 설정 할 수도 있다. 따라서 궤도 유지보수 일정계획은 각 작업 종류에 따라 사용자가 미리 정해놓은 작업을 수행할 수 있는 자원을 고려 하여 계획을 수립 할 수 있어야 한다. 그러므로 일정 계획 결과 작업에 할당되는 자원은 작업 종류와 작업자가 미리 정해놓은 수행 가능한 자원과의 관계에 의하여 결정된다고 할 수 있다[7,8].

완료요구일은 작업이 수행 되어야 하는 최종일로서 자원의 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 완료 요구일이 포함되어 있는 작업들은 일정수립의 대상이 된다.

2.1.1 파라미터 설정

본 절에서는 목적 함수와 제약식을 표현하기 위한 결정 변수와 파라미터를 설정한 후 목적 함수와 제약식에 대하여 기술 하였다. 이와 관련하여 2.3에서는 30, 60개 작업 구간을 대상으로 각 작업을 j 로 설정하고 인력 팀, MTT, STT를 자원 r 로, 7일의 할당 일을 t 로 설정하여 이에 대한 실험 및 분석을 상세히 설명 하였다.

- 결정변수

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

ld_{tr} = r 자원이 t 기간에 실시 할 작업 중 가장 먼 거리에 위치한 작업 구간의 종점

sd_{tr} = r 자원이 t 기간에 실시 할 작업 중 가장 가까운 거리에 위치한 작업 구간의 시점

- 파라미터 정의

j = 작업번호($1 \leq j \leq J$)

t = 기간 인덱스($1 \leq t \leq T$)

r = 자원 번호($1 \leq r \leq R$)

F_j = j 작업 구간의 시점

T_j = j 작업 구간의 종점

- D_j = j 작업의 완료 납기일
- P_{rj} = r 자원을 이용하여 j 작업을 수행할 경우 수행 시간(단위:분)
- SR_j = j 작업을 수행 할 수 있는 자원들의 집합
- SJR_r = r 자원을 이용하여 수행할 수 있는 작업들의 집합
- C_{rt} = r 자원의 t 기간 작업 가능 시간(단위:분)
- RA_h = 인력 자원의 작업 여유 시간(준비시간 + 이동시간 + 정리시간)
- RA_m = 장비 자원의 작업 여유 시간(준비시간 + 이동시간 + 정리시간)
- SMR_m = 장비 자원들의 집합
- SHR_h = 인력 자원들의 집합
- LT_r = r 자원의 최대 이동 가능 거리, 인력 자원만 해당 됨
- $BigM$ = 대수M

$$\sum_{t \leq D_j} \sum_{r=1}^R x_{jtr} \leq 1, \forall j \quad (5)$$

$$\sum_{t > D_j} \sum_{r=1}^R x_{jtr} \leq 0, \forall j \quad (6)$$

인력 작업 시 하루 최대 이동거리 내에서 모든 작업이 이루어져야 한다는 제약을 식(7)을 통하여 나타낼 수 있고, 작업 준비 시간과 작업 정리시간 그리고 작업개소 간 이동시간을 포함한 하루의 작업시간은 최대 작업시간 내에 이루어져야 한다는 제약을 식(8)을 통하여 나타낼 수 있다[8]. 인력이나 작업을 운영할 수 없는 날 $C_{rt} - RA_h$ 는 음수 값을 가질 수 있으므로 Max를 취하여 음수 값으로 인한 제약의 오류를 방지 하였다.

$$ld_{rt} - sd_{rt} \leq LT_r \quad \forall r, \forall t \quad (7)$$

$$\sum_{j \in SJR_r} P_{rj} \cdot x_{jtr} \leq \max(C_{rt} - RA_h, 0), r \in SHR_h, \forall t \quad (8)$$

작업 준비 시간과 작업 정리시간 그리고 작업을 위한 총 이동 시간을 포함한 하루의 작업 시간은 최대 작업시간 내에 이루어져야 한다는 제약을 식(9)와 같이 나타낼 수 있다[8].

$$\sum_{j \in SJR_r} P_{rj} \cdot x_{jtr} \leq \max(C_{rt} - RA_m, 0), r \in SHR_m, \forall t \quad (9)$$

각 자원에 할당된 작업의 최대, 최소 거리를 파악하여 자원의 가용일 별 작업구간을 산출하기 위한 식을 식(10)과 식(11)과 같이 나타낼 수 있다[8].

$$T_j \cdot x_{jtr} \leq ld_{rr} \quad \forall j, \forall t, \forall r \quad (10)$$

$$F_j + BigM \cdot (1 - x_{jtr}) \geq sd_{rr} \quad \forall j, \forall t, \forall r \quad (11)$$

2.1.2 목적함수와 제약조건

본 논문에서 풀이하고자 하는 궤도유지보수 일정계획 문제는 Binary Integer Programming문제이다. 이를 목적함수와 제약조건을 통하여 아래와 같이 표현하였다.

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. 궤도 유지보수 일정 최적화 문제의 목적함수는 식(1)과 같이 적절한 시기에 최대한 많은 작업을 수행하는 것이다[1].

$$Max \sum_{J=1}^J \sum_{t=1}^T \sum_{r=1}^R x_{jtr} \quad (1)$$

궤도 유지보수 작업에는 일정 계획 시 고려해야 하는 여러 가지 현실적인 상황이 존재 한다. 궤도 유지보수 일정 최적화를 위하여 현장의 일반적인 상황을 다음과 같은 제약으로 표현 할 수 있다.

각 보수 작업은 일정 수립기간 중 단 하루에만, 그리고 단 하나의 자원에만 할당 되어야 한다는 제약을 식(2)와 같이 나타낼 수 있다[8].

$$\sum_{t=1}^T \sum_{r=1}^R x_{jtr} \leq 1, \forall j \quad (2)$$

각 보수 작업은 해당 작업을 수행 할 수 있는 자원에 의해서만 수행 될 수 있다는 제약을 식(3)과 식(4)와 같이 나타낼 수 있다[8].

$$\sum_{t=1}^T \sum_{r \in SR_j} x_{jtr} \leq 1, \forall j \quad (3)$$

$$\sum_{t=1}^T \sum_{r \notin SR_j} x_{jtr} \leq 0, \forall j \quad (4)$$

할당된 모든 작업은 완료 요구일 안에 이루어져야 한다는 제약을 식(5)와 식(6)과 같이 나타낼 수 있다 [8].

2.2 BABC 메커니즘 설명

ABC를 활용한 Binary ABC를 궤도 유지보수 일정계획 문제에 적용 하여 어떻게 최적해/최선 해를 탐색해 낼 수 있는지 Fig. 1과 같이 BABC 흐름도로 설명할 수 있다. 단계 1에서는 Food source, Employed bee, Onlooker bee의 수 (SN), Limit(이웃해 탐색 최대 수), 종료 조건을 결정한다. 단계2에서는 초기 가능해군을 생성하기 위하여 특정 자원이 특정 작업에 할당 되었을 때는 1, 아니면 0으로 하는 초기해(BABC에서 Food source)들을 생성하여 초기해 군(initial population)을 생성한다. 단계3에서는 각 초기 해에 대하여 Employed bee를 결정하고 각 초기 해들의 지역탐색을 위하여 이웃 해를 탐색한다. 각 해에 대하여 더 좋은 해가 생성되면 더 좋은 해로 업데이트 하면서 Trial(이웃해 탐색 시도 횟수)을 0으로 설정하고 그렇지 않을 경우 해당 Food source의 Trial = Trial + 1로 증가시킨다. 단계4에서는 단계3의 모든 해의 이웃해 탐색 후 갱신된 초기 해군의 모든 해를 사

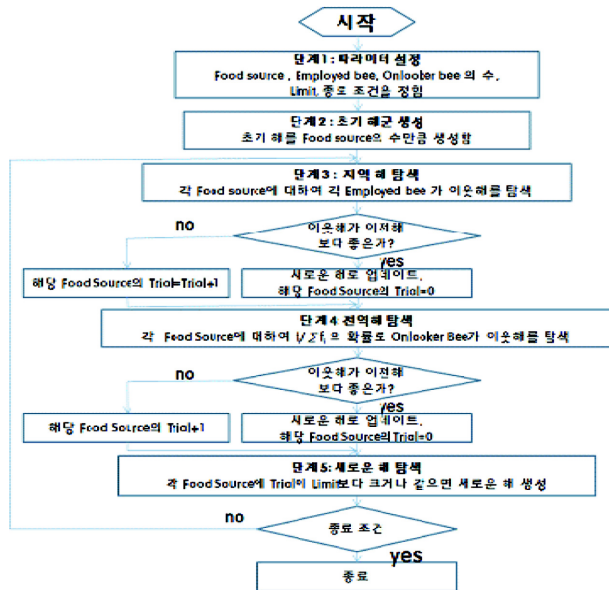


Fig. 1 Flowchart of BABC[2,3]

용하여 평가 값(i 번째 Food source의 평가 값 f_i)에 비례하는 확률로 해(Food source)를 선택하여 이 해의 이웃해와 비교하여 더 좋은 해가 생성되면 더 좋은 해로 업데이트 하면서 Trial을 0으로 설정한다. 그렇지 않을 경우 단계 3과 같이 Trial을 증가 시킨다. 단계5에서는 Trial이 Limit 보다 같거나 클 경우 해당 Food source는 더 이상 탐색하지 않고 탈락 시킨다. 탈락시킨 해의 수만큼의 새로운 해를 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. 이 모든 단계 수행 후 종료 조건(세대 수, 사용자가 제시한 시간제한 등)에 맞으면 종료하고 그렇지 않으면 Employed bee의 이웃해 생성부터 반복 시행한다[2,3].

2.3 궤도 유지보수 일정 계획 문제에 BABC 적용

본 2.3절에서는 2.2절에서 설명한 BABC메커니즘을 간단한 예제를 이용하여 궤도 유지보수 일정 최적화 문제에 어떻게 적용할 수 있는지에 대하여 기술 하였다. 예제 데이터는 한국 철도 공사의 고속선 궤도 보수이력 데이터를 참고하여 가정 한 것이다.

2.3.1 궤도 유지보수 일정 예제 문제

아래의 Table 1은 예제 문제의 정보를 정리한 데이터 이다. 먼저 일정 수립 기간은 3월 1일과 3월 2일 모두 이틀간의 일정을 수립한다. 가용 자원은 장비 자원으로 MTT와 STT 각각 한대 그리고 한 개의 인력 팀을 대상으로 한다. 각 자원의 수행 가능한 작업은 MTT는 작업종류1과 작업종류 2, STT는 작업종류 2, 인력 팀은 작업종류 3을 수행한다 가정 하였다.

2.3.2 예제 문제 풀이 절차

BABC를 적용한 궤도 유지 보수 일정 문제 풀이는 파라

Table 1 Example data for 16 work size

작업 번호	작업 종류	시점 (m)	종점 (m)	연장 (m)	작업시간 (분)	완료요구일 (net)
J1	3	10	13	3	30	3월2일 (2)
J2	3	5	9	4	40	3월2일 (2)
J3	3	1,927	1,930	3	30	3월2일 (2)
J4	3	422	432	10	80	3월2일 (2)
J5	3	2,500	2,575	25	100	3월2일 (2)
J6	3	617	620	3	30	3월2일 (2)
J7	1	3,000	3,080	80	60	3월2일 (2)
J8	1	105	135	30	30	3월2일 (2)
J9	2	990	1,000	10	10	3월2일 (2)
J10	2	305	310	5	5	3월2일 (2)
J11	3	3,044	3,053	9	90	3월2일 (2)
J12	1	200	219	19	20	3월2일 (2)
J13	2	3,077	3,097	20	40	3월2일 (2)
J14	3	2,233	2,243	10	30	3월2일 (2)
J15	1	3,120	3,125	5	50	3월1일 (1)
J16	2	1,997	2,020	23	100	3월1일 (1)

미터 초기화, 초기해 생성, Employed bee에 의한 이웃해 탐색, Onlooker bee에 의한 전역해 탐색, Scout bee에 의한 새로운 해 탐색의 절차로 이루어 진다. 가장먼저 문제 풀이에 필요한 파라미터들을 미리 설정한다. BABC알고리즘에 문제를 적용하기 위해서 Food source와 Employed bee, Onlooker bee의 개수, 그리고 Limit와 종료 조건이 필요하다.

초기 Food source 생성을 위하여 의사결정변수 x_{jtr} 을 표현하기 위해 본 논문에서는 작업 별 가용일과 작업 별 자원이라는 두 개의 의미를 담을 수 있도록 해를 표현 하였다. 아래 Fig. 2의 x_{jtr} 는 작업 j 가 t 기간에 r 의 자원을 이용하여 실시되면 1의 값을, 그렇지 않으면 0의 값을 가지게 된다. 다음 Fig. 2는 Table 1 예제 데이터의 Food source 표현 방법을 나타낸 것이다.

이와 같은 해 표현 기법을 통하여 해 표현 가능 영역과 불가능 영역을 설정할 수 있다. 해 표현 불가능 영역이란 Fig. 3과 같이 각 작업 별로 주어진 완료 요구 일의 범위 밖의 날과 해당 작업을 수행 할 수 없는 자원의 영역에 의사결정변수 x_{jtr} 가 아닌 0의 값을 미리 할당하는 영역을 말한다.

초기 해를 생성 하기 위하여 각 작업 별로 해 표현 가능 영역 중 하나의 영역을 임의적으로 선택 한다. 선택된 영역은 1의 값을 가지게 되고 해 표현 불가능 영역을 포함한 나머지 영역은 0의 값을 갖게 된다. 초기해 생성 후 장비와 인력 자원의 용량 제약을 고려하여 미리 설정하기 어려운 제약들은 해의 값이 제약을 만족 시킬 수 있도록 2.3.3의 가능해 탐색 절차를 수행 한다.

Employed bee에 의한 이웃 해 탐색은 Food source의 모든 작업 중 임의적으로 선택 된 하나의 작업에 대하여 다른 날 또는 다른 자원으로 할당결과를 변경하는 것이다. 이

자원(r)	완료일(C)	작업(j)															
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
MTT	1일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}
	2일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}
STT	1일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}
	2일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}
제한	1일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}
	2일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}

Fig. 2 Representation of Food source

자원(r)	완료일(C)	작업(j)															
		J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
MTT	1일	0	0	0	0	0	0	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	0	X _{jtr}	X _{jtr}	0	X _{jtr}	X _{jtr}
	2일	0	0	0	0	0	0	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	0	X _{jtr}	X _{jtr}	0	0	0
STT	1일	0	0	0	0	0	0	0	0	X _{jtr}	X _{jtr}	0	0	X _{jtr}	0	0	X _{jtr}
	2일	0	0	0	0	0	0	0	0	X _{jtr}	X _{jtr}	0	0	X _{jtr}	0	0	0
제한	1일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	0	0	0	0	X _{jtr}	0	0	X _{jtr}	0	0
	2일	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	X _{jtr}	0	0	0	0	X _{jtr}	0	0	X _{jtr}	0	0

Fig. 3 Representation of Food source considering constraints

때 변경 될 작업은 다른 Food source의 할당 결과를 모방하여 변경하게 된다. 이웃 해 탐색의 절차는 다음과 같다.

단계 1 : 이웃 해를 탐색 할 자기 자신을 제외한 Food source 중 하나를 임의적으로 선택

단계 2 : 선택된 Food source의 모든 작업 중 하나의 작업을 임의적으로 선택

단계 3 : 선택된 작업 j에 포함된 모든 x_{jtr}의 할당 값을 자기 자신의 Food source의 x_{jtr} 값으로 변환

단계 4 : 가능 해 탐색 절차 수행

단계 5 : 기존의 평가 값보다 좋으면 채택 후 Trial를 0으로, 그렇지 않으면 탈락 후 Trial+1

단계 6 : 자기 자신이 마지막 Food source(SN)이면 종료, 그렇지 않으면 단계 1로

Onlooker bee의 전역 해 탐색은 모든 Food source의 평가 값을 바탕으로 새로운 이웃 해를 탐색하는 것이다. 각 Food source는 아래의 식(12)에 따라 자기 자신이 선택 될 확률을 갖는다. 선택된 Food source만이 이웃 해를 탐색할 수 있으므로 결국 좋은 평가 값을 가지는 Food source는 더 많은 이웃 해 탐색의 기회를 갖게 되는 것이다. 이후의 이웃 해 탐색 과정은 Employed bee에 의한 이웃 해 탐색과 동일하다. 전역해 탐색 과정에서 Food source(i)가 선택 될 확률(p_i)는 식(12)와 같다.

$$p_i = f_i / \sum_{n=1}^{SN} f_n \quad (12)$$

위의 이웃 해 탐색 과정에서 각 Food source별로 업데이트 되고 있는 Trial을 미리 설정된 Limit와 비교한다. Trial이 Limit만큼 커지면 해당 Food source는 미리 정해진 세대 수 동안 더 좋은 해를 찾지 못한 것이므로 이 해는 버려지게 되고 버려진 Food source의 수만큼 Scout bee에 의하여 새로운 해를 탐색한다. 새로운 해를 탐색하는 방법은 초기

해 생성과 동일하다.

이 모든 단계 수행 후 종료조건(본 논문에서는 세대수 포함)에 맞으면 종료하고 그렇지 않으면 이웃해 생성부터 반복 시행한다.

2.3.3 가능해 탐색 절차

초기해와 이웃해 탐색 후 만들어진 해는 식(7), 식(8), 그리고 식(9)를 만족한다고 할 수 없다. 식(7), 식(8) 그리고 식(9)를 만족하지 못하는 해는 비 가능해라 할 수 있으므로 이러한 비 가능 해를 가능해로 만들기 위한 절차는 반드시 필요하다. 궤도 유지보수 일정계획 문제에서 순수한 BABC의 해 탐색을 통하여 얻어질 수 있는 해들은 매우 많은 비 가능 해를 포함한다. 뿐만 아니라 작업할당량을 최대화 하기 위하여 더 많은 비 가능 해를 탐색하게 될 것이다. 따라서 초기해와 이웃해 탐색 후 만들어진 해는 아래의 가능해 탐색 절차를 통하여 가능해로 변경되게 된다. 식(7)의 제약식을 만족하기 위한 가능해 탐색 절차는 다음과 같다.

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

단계 2 : ld_{rt}-sd_{rt}의 값을 구하여 LT_r보다 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 작업 x_{jtr}의 값을 0으로 할당한 후 단계 1로

단계 4 : t = T이면 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

식(8), 식(9)의 제약식을 만족하기 위한 가능 해 탐색 절차는 다음과 같다.

단계 1 : 자원 r에 대하여 t날 할당된 모든 작업들의 작

업시간 의 합 P_{ij} 를 구하기

단계 2 : 자원 r 의 하루 최대 가용 시간(인력이면 $C_{r1} - RA_{1r}$, 장비이면 $C_{r2} - RA_{2r}$)을 구한 후 t 날 자원 r 에 할당된 모든 작업의 P_{ij} 의 합 과 비교하여, P_{ij} 가 더 크면 단계 3으로, 그렇지 않으면 단계 4로

단계 3 : 자원 r 에 대하여 t 날 할당된 모든 작업들 중 작업시간이 가장 큰 작업 j 의 x_{ij} 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 $r = r + 1, t = 1$

3. 효과 실증 분석

본 논문에서 제안한 BABC알고리즘을 적용한 궤도 유지 보수 일정 계획 문제를 Microsoft사의 Visual Studio 2005의 C++를 이용하여 프로그램화 하였다. 그리고 Intel(R) Core (TM)2 Duo CPU P8600 @ 2.40GHz 그리고 RAM 2.0GB 사양의 노트북 컴퓨터로 실험을 진행 하였다. 뿐만 아니라 BABC 알고리즘을 적용한 궤도 유지보수 일정 계획 문제풀이의 성능 및 효율성을 비교하기 위하여 ILOG 제품 군의 CPLEX를 이용한 수리모형의 풀이결과와 비교 하였다. CPLEX는 수리모형을 이용하여 최적 해를 제시할 수 있다.

실험은 일정수립기간 7일 동안 30개, 60개, 100개, 그리고 200개의 작업구간을 대상으로 하였으며, 30개, 60개, 100개 작업의 경우 3개의 작업종류와 3개의 자원을 이용하였고, 200개 작업의 경우 할당률을 높이기 위하여 4개의 자원을

Table 2 Comparative analysis for BABC and CPLEX

문제 크기	풀이 방법	평균 풀이 시간(초)	최대값(할당구간) / 최적해 여부(Yes/No)	평균 (할당구간)
30개소	BABC	3.40	30 / Y	29.91
	CPLEX	5.50	30 / Y	30
60개소	BABC	15.20	48 / Y	47.163
	CPLEX	17.75	48 / Y	48
100개소	BABC	298	63 / N	61.59
	CPLEX	1748	65 / Y	65
200개소	BABC	1024	117 / -	115
	CPLEX	-	-	-

대상으로 하였다. 작업종류는 인력템핑과 장비템핑(총다지기), 장비템핑(Spot)으로 설정하였고, 인력템핑은 인력팀, 장비템핑(총다지기)은 MTT, 장비템핑(Spot)은 STT를 사용하여 작업이 가능하다. 30개, 60개, 100개의 경우 인력 팀, MTT, STT 각각 하나의 자원을 대상으로 일정을 수립하였으며, 200개 작업의 경우 인력 팀 두 팀과 각각 하나의 MTT, STT 를 대상으로 일정을 수립하였다. 자원의 용량 중 여유시간을 제외한 실제 작업 가능 시간은 장비의 경우 최대 120분이며, 인력의 경우 최대 180분이다. 인력 작업의 경우 하루 최대 이동 가능 거리는 2km로 설정 하였다. 입력 데이터는 고속선 궤도 유지보수 이력 데이터의 형식을 바탕으로 만들어 졌다 [7,8].

Limit는 작게 설정되면 Scout bee의 임의의 해 탐색의 횟수가 많아지므로 다양한 해를 탐색할 수 있고, 크게 설정되

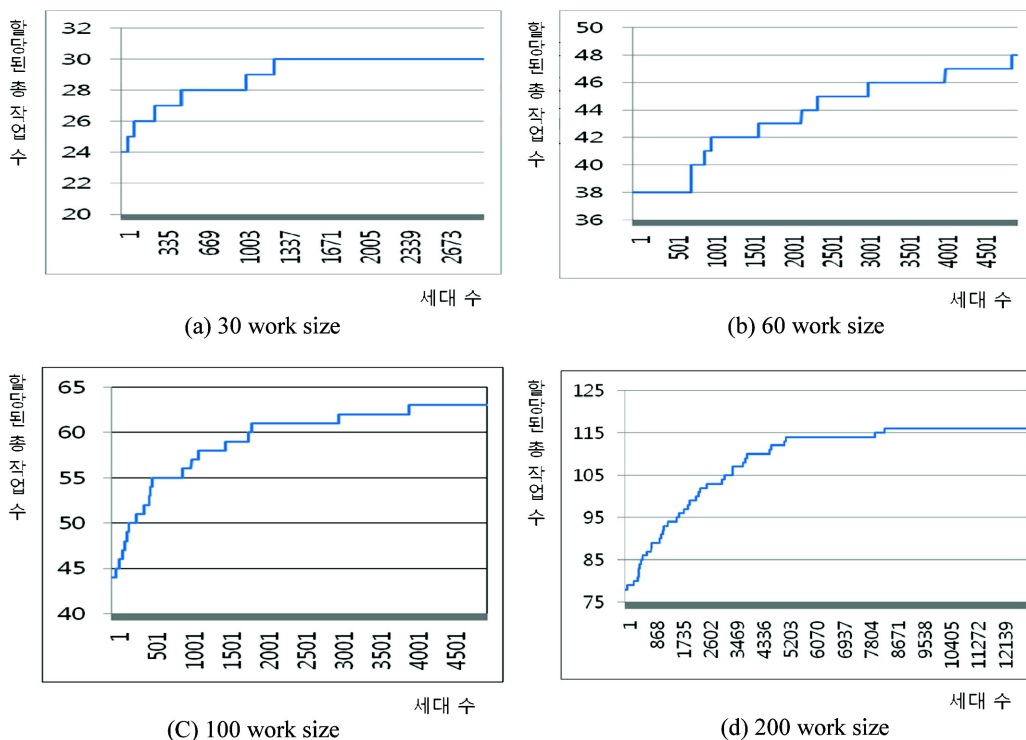


Fig. 4 Trend of convergence for the results using BABC

면 이웃해 탐색의 횟수가 많아 지므로 더욱 집중적인 해 탐색이 가능하다. 따라서 적절한 Limit의 설정은 BABC의 가장 중요한 파라미터 설정이라 할 수 있다. 본 논문에서는 실험을 통하여 30개 60개 100개 200개 작업의 경우 각각 100, 200, 500, 1200의 Limit를 설정 하였다. Food source의 수는 각각 60, 120, 200, 300개로 설정 하였고 종료조건은 각각 3000, 5000, 5000, 12000세대로 설정 하였다.

먼저 Table 2는 각 문제 크기에 대하여 100회의 실험을 통하여 얻어 진 최대값과 평균값을 나타낸 것이다. BABC의 성능을 비교하기 위하여 CPLEX를 이용한 풀이결과와 비교 하였다.

문제 풀이 결과 문제 크기가 30개와 60개인 경우 각각 30개 작업과 45개 작업을 할당하여 CPLEX와 동일한 결과를 풀이할 수 있었다. 문제 크기가 100개인 경우 CPLEX와 비교하여 2개의 작업이 적게 할당된 결과를 얻었지만, 풀이시간에서 약 6배 가량 빠른 풀이 결과를 얻을 수 있었다. 문제 크기가 200개인 경우 CPLEX를 이용하여 약 720분간 문제 풀이를 시도하였지만, 최적 결과 풀이에 실패하였다. BABC의 경우 약 17분의 풀이시간을 통하여 117개의 작업을 할당하는 결과를 풀이 하였다. 비록 최적 해를 보장할 수는 없지만 데이터의 양이 많아지는 경우라도 일정계획 수립이 가능하다는 것을 확인 할 수 있다.

다음의 Fig. 4는 30, 60, 100, 200개 작업의 최선 해를 탐색하기 위한 수렴 경향을 나타낸 것이다.

4. 결 론

본 논문에서는 철도궤도 유지보수 일정 계획 문제에 대하여 목적식과 제약식의 수리 모형을 제시 하였다. 또한, BABC에서 이웃해 탐색과 전역해 탐색 메커니즘을 설명 하였고 이 메커니즘을 이용하여 궤도 유지보수 30개, 60개, 100개 그리고 200개 작업구간 일정 계획 문제에 적용하고 최선의 해 탐색과정을 설명하였다. 본 논문에서 제안하는 방법론은 각기 다른 현장의 장비 및 인력 운영 룰에 맞추어 현장 문제를 풀이 할 수 있다는 장점이 있다. 각 작업의 종류에 따라 작업을 수행할 수 있는 자원이 달라 지더라도 이를 반영하여 문제를 풀이 할 수 있기 때문이다. 뿐만 아니라 BABC 알고리즘의 특성상 파라미터수가 적어 다른 휴리스틱 알고리즘과 비교하여 가장 적절한 파라미터의 값을 설정하는 것이 수월하다고 할 수 있다. 이를 증명하기 위하여 한국철도공사의 고속선 궤도 보수이력 데이터를 사용하여 궤도 유지보수 일정최적화 문제의 시뮬레이션을 수행 하였다. 그 결과 본 논문에서 제시하는 방법을 통하여 최선의 해를 제시하고 이를 CPLEX 방법을 사용하여 찾아낸 해와 비교 분석하였다. 본 논문에서 정의한 제약들은 여러 현장에서 공통적으로 적용될 수 있는 대표적인 제약들을 고려한 것이다. 따라서, 현장의 특성에 맞는 추가적인 제약을 유연하게 반

영하는 연구가 필요하고, 추후 더 크고 대규모의 작업구간 문제들을 여러 가지 방법들을 적용하여 비교 분석할 것이다.

참고문헌

- [1] D.Y. Kim, K.D. Kim, S.G. Lee, K.W. Lee et al. (2008) Track Deterioration Prediction and Scheduling for Preventive Maintenance of Railroad, 2008 *Korean Society for Railway Conference*, pp. 1359-1370.
- [2] Sung-Soo Kim, J.J. Byeun (2010) Optimal Design of Sensor Network using Binary Artificial Bee Colony, working paper, Kangwon National University.
- [3] Sung-Soo Kim, J.J. Byeun (2010) Optimal Design of Reporting Cell Location Management System using Binary Artificial Bee Colony, working paper, Kangwon National University.
- [4] J.H. Park (2009) A Study on Railway Track Maintenance Scheduling Using a Genetic Algorithm, Kangwon National University.
- [5] S.M. Oh, J.H. Lee, J.H. Lee, B.H. Park, et al. (2006) A study on Mathematical Model of the Long-term Track Tamping Scheduling Problem, *Journal of the Korean Society for Railway*, 9(1), pp. 50-56.
- [6] S.W. Lee et al. (2007) Korea High Speed Track Maintenance Decision Making Support System: 1st, Korail.
- [7] S.W. Lee et al. (2008) Korea High Speed Track Maintenance Decision Making Support System: 2nd, Korail.
- [8] S.W. Lee et al. (2009) Korea High Speed Track Maintenance Decision Making Support System : 3rd, Korail.
- [9] Karaboga, Dervis, Basturk and Bahriye (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39(3), pp. 459-471.
- [10] A. Higgins (1998) Scheduling of Railway Track Maintenance Activities and Crews, *The Journal of the Operational Research Society*, 49, pp. 592-623.
- [11] J. Bruno, E.G. Coffman, Jr. and R. Sethi (1974) Scheduling independent task to reduce mean finishing time, *Comm. ACM* 17, pp. 382-387.
- [12] Li, Kai, Yang, Shan-Lin. (2008) Non-identical parallel-machine scheduling research with minimizing total weighted completion time: Models, relaxation and algorithms, *Applied Mathematical Modelling*, Article in Press.
- [13] M. McNaughton (1959) Scheduling with Deadlines and Loss Functions, *Management Science*, 6, pp. 1-12.
- [14] M. Miwa (2006) Mathematical programming model analysis for the Track Maintenance Schedule, *Quarterly reports*, 43(3), pp. 131-136.

접수일(2010년 7월 29일), 수정일(2010년 11월 11일),
게재확정일(2010년 11월 22일)