

열차경합 해소에서 유전자 알고리즘의 적용에 관한 연구

A Study on GA Approach for the Railway Traffic Conflict Resolution

오석문*, 홍순흠**, 김성호***

Oh, Seog-Moon, Hong, Soon-Heum, Kim, Seong-Ho

ABSTRACT

This paper is to investigate adoptability of the genetic algorithm to the railway traffic conflict resolution problem. We categorize the problem in two cases, one is the specific-boundary case and the other is not. GA is may be used in the first case, but not for second case. For a sample problem, an GA is designed, and the GA is being compared with Ho's[2,3].

1. 서 론

한국철도의 네트워크에서 순시적(simultaneous)으로 운행되는 열차의 수는 대략 50여대에 이르고 있다. 각각의 열차들은 미리 정해진 시간(timing)과 진로(path)에 따라 운행되도록 사전에 스케줄되어 있다. 따라서 모든 열차들은 주어진 스케줄을 준수해야 하며, 그 스케줄은 실행 가능한 것이어야 한다. 그러나 실제의 열차운행에서 열차경합(conflict)에 따른 지연발생은 피할 수 없는 상황이 된다. 각 열차제어사무소의 사령요원들(train dispatchers)은 불가피하게 발생하는 지연의 문제를 최소화하기 위해 하루에도 수많은 의사결정을 수행한다. 하지만 이러한 의사결정은 승수 시스템의 구조가 복잡해지고, 고객의 정시성(punctuality)에 대한 요구수준이 높아짐에 따라 점점 더 어려워지고 있다. 따라서 이러한 문제에 신속하게 대응할 수 있는 시스템의 개발이 필수적으로 되었다.

한국철도기술연구원에서는 이러한 문제에 효율적으로 대응하기 위해 '지능형 운행제어 시스템(iTMS, Intelligent Traffic Management System)의 핵심모듈 개발'을 위한 기본연구 과제를 수행하고 있으며, 본 논문은 이러한 과제의 진행 중에 검토되는 여러 접근법(solution approaches) 가운데 유전자 알고리즘(GA, genetic algorithm)의 적용에 관한 연구이다.

열차경합 문제에서 시뮬레이션에 의한 접근법과 관련하여 KONIG & STAHLI(1970)는 분기점(junction)과 소규모 역에서의 운영상황을 시뮬레이션할 수 있는 프로그램을 제안하였으며, Zhu & Schnieder(2000)는 열차지연의 상황을 Coloured Petri Nets으로 모형화한 시뮬레이터를 제안하였다. Schaefer & Pferdmenges(1994)는 전문가 시스템에 기초한 열차경합의 예측 및 해소 시스템을 소개하였다. Berge-Brezetz et. al.(1998)은 문제에 대해 완전 자동화된 의사결정 체계에 대한 위험성을 지적하였으며, 이에 따라 규칙기반(rule-base) 기법을 이용한 사용자 연동형 프로그램을 제안하였다. 하지만 규칙기반 시스템은 문제가 사전에 잘 정의된 형태가 아닌 경우에는 적용하기가 곤란하다. Berge-Brezetz et. al.(1998)는 이러한 단점을 극복하기 위한 Tabu Search 기반의 알고리즘(자신들이 'iterative repair'라고 명명한)을 추가적으로 제시하였다. Stolk(1998)는 네델란드 철도에서 경합 발생의 예측 및 해소를 위해 개발중인 시스템의 목적 및 개념을 소개하였다.

*** 한국철도기술연구원 주임연구원, 정회원

** 한국철도기술연구원 책임연구원, 비회원

Mazzarello & Copello(2000)은 유럽연합(EC)이 지원한 MARCO(Multilevel Advanced Railways Conflict Resolution and Operation control) 프로젝트의 내용을 소개하였다. MARCO 프로젝트에서는 열차경합 해소를 위해 HAT(High Traffic Area)와 GAN(Global Area Network)을 구분하여 문제를 인식하였다. B&B, B&C, SA, GA 등 다양한 기법을 연구하였으며 greedy rule-based 기법을 선택하였다.

T. K. Ho(Hong Kong Polytechnic University)는 열차경합 해소 문제에 대해 많은 연구(T. K. Ho, J. P. Norton and C. J. Goodman, 1997, T. H. Yeung and T. K. Ho, 2000, T. K. Ho and T. H. Yeung, 2000)를 수행하였으며, 최근 GA 알고리즘에 의한 경합해소 방법을 제시하였다.

이 논문에서는 열차경합 예측 및 해소 문제에 대한 문제정의를 2.1에서 제시한다. 선행 연구에서 제시하는 문제인식 및 문제해결을 위한 접근법들에 대해 고찰한다. 2.2에서는 열차경합 해소문제에 GA 알고리즘을 적용할 때의 장점, 결정변수, 유전자 구조 및 neighbourhood의 구조 등에 대해서 검토한다. 2.3에서는 현재 진행중인 예제 문제에 대한 GA 알고리즘의 적용에 대해 설명하고, 마지막으로 3장에서는 결론 및 향후 연구방향을 제시한다.

2. 본 론

2.1 문제정의 및 접근법

열차경합의 예측 및 해소에 관련된 기능은 트래픽 관리 시스템의 전체적인 개념을 향상시키는 중요한 기능이다. KONIG와 STAHLI(1970)는 경합이 발생하는 상황을 ‘일정한 시간대에 여러 대의 열차들이 진행을 위해 진로를 요청할 때, 어느 열차에 그 진로를 할당할 것인가?’를 결정하는 문제라고 정의하였다. 또 T. H. Teung & T. K. Ho(2000)는 ‘순서가 정해진 여러 개의 작업들이 2개의 queue에 나누어져 있고, 각 작업들이 하나의 기계에 의해서 처리될 때, 전체 작업을 수행하는데 소요되는 가중 작업시간의 합(sum of total weighted tardiness)을 최소화하는 문제’라고 정의하였다.

그러나 이와 같은 경합의 정의는 경합의 문제를 다소 단순한 형태로 인식한 결과이다. Schaefer & Pferdmenges(1994)는 경합을 ‘하나의 자원(e.g. track section)을 2개의 열차가 동시에 점유하려는 시도 또는 사전 스케줄로부터 이탈된 운행 형태’라고 정의하였다. 현재 한국철도의 본선에 운행되는 열차의 운행상황은 매우 복잡하며, 이에 따른 경합의 원인 및 형태도 여러 가지의 항목이 복합적으로 발생한다. 대표적인 경합의 원인으로는 선로의 고장, 차량의 고장, 비상차량의 운행, 운행장애(승객, 기후 및 기타 원인에 의한), 선로 및 분기부의 최대용량 초과 사용, 화물열차의 운행 등을 들 수 있으며, 이와 같은 원인에 따라 지연(delay conflict), 교행·대피 순서 변경(sequence conflict), 플랫폼 사용 변경(platform conflicts), 차량운용 변경(trainset routing conflict), 승무원운용 변경(rostering conflict), 환승 불가(synchronization conflicts) 등의 경합 형태가 발생한다.

본 연구에서는 경합을 ‘당일 운영을 위한 모든 열차·차량·승무원이 사전 운행계획(schedule)으로부터 이탈되어 운행되는 제반 상황’이라고 정의하고, 시간적인 범위를 영업운행이 수행되는 시간으로, 또 공간적인 범위로 열차·차량·승무원이 당일 운행하는 본선 및 사무소로 한다. 이렇게 포괄적으로 정의되는 경합의 문제를 하나의 모형으로 모두 처리하기는 어려우며, 현실에 적용하기 위해서는 사용자와 인터페이스(MMI)가 고려된 응용 프로그램의 형태로 구성되어야 할 것

이다. 본 연구에서는 이러한 응용 프로그램(ITMS)의 핵심모듈 개발을 목적으로 하고 있으며, 본 논문에서는 유전자 알고리즘의 적용방안에 대해 검토한다.

앞에서도 언급하였지만 경합해소 문제에 대한 접근법들은 지금까지 매우 다양한 방법들이 시도되었다. Berge-Brezetz et. al.(1998)가 소개한 전문가 시스템(e.g. Komaya & Fukuda, 1989), Branch & Bound(e.g. Higgins et al, 1996), 제약전파(e.g. Chiu et al., 1996), Fuzzy(Chang & Thia, 1996)외에도 시뮬레이션에 의한 방법, 규칙기반에 의한 방법, 메타 휴리스틱에 의한 지역탐색 방법 등의 방법이 시도되었다.

경합예측의 문제에서는 경합해소의 문제와 달리 시뮬레이션에 의한 통일된 접근법들이(Stolk, 1998), Mazzarello & Copello, 2000) 제시되고 있다. 경합예측 문제는 경합해소 문제와 마찬가지로 실시간에 해당하는 응답시간(solution time requirements)이 요구되는데, 60초 내에 10분 후의 상황을 예측하는 등의 조건이 제시되고 있다(Stolk, 1998).

경합예측 및 해소 문제에 있어서 한국철도 전체 네트워크를 대상으로 하는 문제는 필요성이 적을 것으로 판단된다. 주어지는 문제는 담당 열차제어 사무소별로 문제가 되는 일부 영역, 수송밀도가 높은 수도권 및 서울~대전간의 구간 또 장래 고속철도가 운행되는 경우 고속철도와 기존철도의 분기·합류 지점 등 선로의 부분구간이 될 가능성이 높다. 이러한 문제영역은 현재 명시적으로 그 경계가 설정되지 않은 상황이며, 그 경계의 설정방법에 대해서는 현실 데이터를 이용한 전형적인 경합분석을 통해 설정할 계획에 있다. 이와 같이 문제영역의 경계가 불분명한 경우에는 수리모형, 메타 휴리스틱, 시뮬레이션 등의 방법이 유력할 것이나, 문제영역의 경계가 불분명한 경우에는 이러한 방법보다 규칙기반 시스템의 적용을 검토해야 할 것으로 판단된다.

2.2 유전자 알고리즘의 적용

유전자 알고리즘은 복잡한 문제를 표현하는데 매우 용이하며, 간단한 형태로 정식화(formulation)되기 어려운 문제를 쉽게 표현할 수 있다(GEN & CHENG, 1996).

Ho는 유전자 알고리즘에 의한 열차경합 해소 방안[2,3]을 제시하였다. Ho는 열차경합의 문제를 순서가 정해진 작업들이 2개의 대기행렬(queue)에 들어있고, 각 대기행렬의 작업들이 하나의 기계(single process)에 의해 처리되는 것으로 모형화 하였다. Ho의 모형에서 결정변수(decision variable)는 기계에서 처리되는 각 작업의 순서를 결정하는 것이 되었고, 목적(objective function)은 가중 작업시간의 합(sum of total tardiness)을 최소화하는 문제가 되었다.

그러나 Ho의 문제설정은 열차종별 및 fitness value 계산에 있어 지나치게 단순화 되어있다. 하나의 대기행렬에 들어 있는 작업(열차)들은 각각의 종별에 따라 구분되고, 각 종별에 따른 지연시간별 가중치가 달라야 한다. 또 fitness value를 구하기 위한 착발시간의 계산에 있어 필수적으로 고려해야하는 연동로직(interlocking logic)이 대부분 고려되지 않은 상황이다.

본 논문에서는 열차종별을 구분하고, 연동로직을 고려한 fitness value를 계산하였으며, Ho가 제안한 neighbourhood 구조에 대해 추가적인 구조를 제안하였다. 제안한 neighbourhood 구조는 보다 현실적인 구조라고 판단되며, 현재 그 효율성의 비교를 수행 중이다.

1	2	3	4	5	6	7	8	9	10
HSR1	HSR2	HSR3	HSR4	SE1	SE2	SE3	MU1	MU2	MU3

그림 1 유전자 기본구조

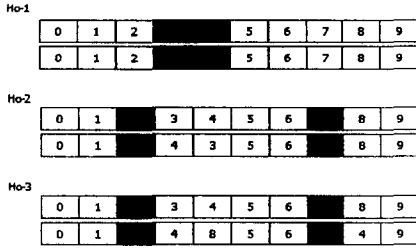


그림 2 Ho의 구조

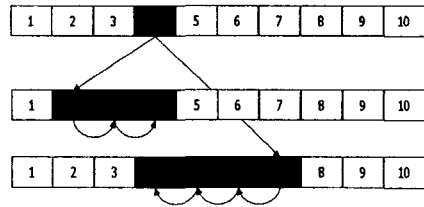


그림 3 제안한 구조

그림 1은 유전자(chromosome)의 기본구조로써 각각의 염색체(gene)의 값은 열차번호를 의미한다. 그림 2는 Ho가 제안한 neighbourhood 구조로서 첫 번째는 인접한 열차의 순서를 변경하는 구조, 두 번째는 인접하지 않는 한 쌍의 열차순서를 변경하는 구조, 세 번째는 인접하지 않는 2쌍 이상의 열차 순서를 변경하는 구조를 나타낸다. 그림 3은 본 논문에서 제안하는 구조로서 인접한 몇 개의 열차가 순연(shifting)하는 구조를 나타낸다.

개체수(population size)는 Ho의 2개와 달리 10개를 두었으며, 세대수(generations)는 50세대까지의 경우를 검토하고 있다. 각 세대간의 재생산은 그림 4와 같이 상위 4개체는 다음 세대에서 생존하고, 나머지 개체는 우수 개체에 의해 복사된 후 neighbourhood 구조에 따라 새로운 해를 구한다.

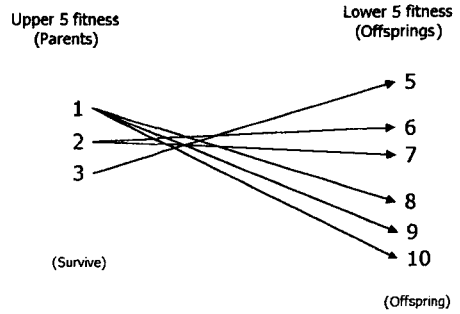


그림 4 세대간 재생산 구조

2.3 예제문제

제안된 유전자 알고리즘을 평가하기 위해 예제문제를 구성하였다. 그림 5와 같은 선로구조와 그림 6과 같은 초기 열차 시각표를 고려한다.

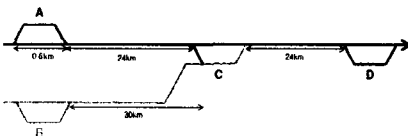


그림 5 선로구조(예제)

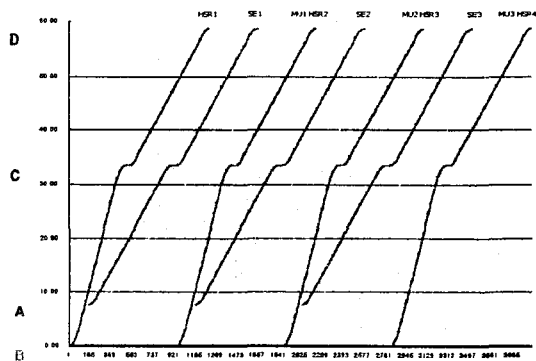


그림 6 열차 시각표(예제)

그림 1에서와 마찬가지로 1~4번 열차는 고속열차로 B역에서 출발하고, 5~7번 열차는 새마을, 8~10 열차는 무궁화로 모두 A역에서 출발한다. 주어진 스케줄에서 A/B의 임의의 역에 일정시간

의 지연이 강제적으로 설정되고, 알고리즘은 최초의 스케줄과 비교하여 가중 패널티(weighted penalty)가 최소인 유전자를 찾아낸다. 여기서 유전자가 결정하는 것은 각 열차가 C역에 도착하는 순서이고, 나머지 역들의 출발 및 도착은 정해진 규칙에 따른다. 표 1.은 KONIG와 STAHLI(1970)이 제시한 규칙들이다. 예제문제의 알고리즘에서는 FCFS(First Come First Serve)의 규칙에 따라 출발 및 도착순서가 결정된다.

표 1 자원할당 규칙

Rule	내용
Scheduled Sequence	지연상황에 관계없이 초기에 계획된 스케줄에 따라 할당한다.
FCFS	도착하는 순서대로 할당한다.
Priority	열차종별로 우선순위 표를 작성하고, 이 순서에 따라 할당한다.
Maximum Capacity	최대통과 용량이 되도록 할당한다.

알고리즘이 제시한 유전자의 C역 도착순서와 A/B역에서 출발한 열차들의 출발순서에 의한 논리가 상호 호환이 불가능한 경우에는 불가능해(infeasible solution)로서 해당하는 유전자는 매우 높은 패널티를 받게되고, 그렇지 않은 경우에는 가능해(feasible solution)로서 연동로직에 따라 착발시각이 계산되고, 이 시간으로부터 fitness value가 계산된다. 그림 7은 이러한 상황을 나타낸다.

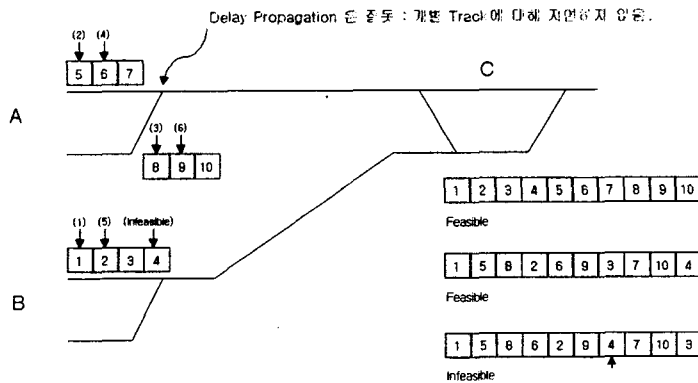


그림 7 Fitness Value 계산

3. 결론

이 논문은 열차경합 해소문제에 대한 유전자 알고리즘의 적용에 관한 연구이다. 앞에서 언급한 바와 같이 열차경합 해소문제에는 다양한 접근법이 시도되었고, 이 논문에서는 이중 유전자 알고리즘의 적용에 대해 검토하였다. 이러한 검토는 당분간 계속 검토되어야 할 사항이며, 현재까지의 결론을 제시하면 다음과 같다.

열차경합 해소문제의 문제영역을 한국철도 전체 네트워크를 대상으로 하기는 어려울 뿐만 아니라¹⁾, 그 실효성도 크지 않을 것으로 판단된다²⁾. 따라서 네트워크 상의 열차운행 실적에 대한 로그 데이터를 분석하여 문제가 되는 영역을 모두 구별해 내야 한다. 문제가 되는 영역은 다음과 같은 2가지의 경우가 될 수 있다.

- case-1 : 영역의 경계가 분명한 경우.

1) 문제범위가 커지면 실시간 응답시간 요구사항을 지키기 어렵다.
2) 국지적으로 발생하는 열차경합이 전국적인 영향을 미치는 일은 거의 없을 것으로 판단된다.

· case-2 : 영역의 경계가 애매한 경우.

case-1의 경우는 case-2에 비하여 짧은 기간에 집중적인 연구를 통하여 효율적인 접근법이 제시될 수 있을 것으로 판단되며, 이러한 접근법의 대안들로는 앞에서와 같이 수리모형, 메타 휴리스틱, 시뮬레이션 등을 들 수 있을 것을 판단되고, 이 논문에서 소개하는 유전자 알고리즘도 이에 대한 유력한 대안이 될 수 있을 것으로 판단된다.

case-2와 같이 경계가 애매한 경우에 대해서는 규칙기반의 방법론이 유력할 것으로 판단된다. 이 접근법은 각 제어사무소 사령요원들의 경험과 노하우(know-how)가 구조화된 방법에 따라 추출되어야 하고, 추출된 지식을 현장에서 발생하는 수많은 사례들과 연관지어야 하므로 case-1에 비해 장기적인 연구가 필요할 것으로 판단된다.

iTMS에서는 궁극적으로 두 가지의 경우 모두에 대해 효율적인 해(solution)를 제공할 수 있어야 하고, 따라서 각각의 경우에 대한 연구를 적절한 우선순위와 절차에 따라 수행해야 할 것으로 판단된다.

참고문헌

1. Helmut KONIG and Samuel STAHLI, "Simulation of railway operation through junctions and in small terminals", RAIL INTERNATIONAL, Nov. 1970.
2. T. K. Ho and T. H. Yeng, "Railway junction conflict resolution by genetic algorithm", ELECTRONICS LETTERS 13th vol. 36, no. 8, April 2000, p. 771~772.
3. T. H. Yeng and T. K. Ho, "Railway junction conflict resolution by local search method", Computers in Railways VII 2000, p. 769~776.
4. T. K. Ho, J. P. Norton and C. J. Goodman, "Optimal traffic control at railway junctions", IEE Proc. Electr. Power Appl., vol. 144, no. 2, March 1997.
5. Penglin Zhu and Eckehard Schnieder, "Determining Traffic Delays through Simulation", Institute of Control and Automation Engineering, Technical University of Braunschweig, Working Paper.
6. S. Berge-Brezetz, S. Benoliel, G. Bain and G. Menapace, "Computer aided methods for railway operation conflict solving", Computers in Railways, 1998, p. 563~572.
7. A. Stolk, "Automatic Conflict Detection and Advanced Decision Support for Optimal Usage of Railway Infrastructure Purpose and Concept", Computers in Railways, 1998, p. 629~638.
8. H. Schaefer and S. Pferdmeiges, "An expert system for real-time train dispatching", Computers in Railways, 1994, p. 27~34.
9. Mitsuo Gen & Runwei Cheng, "Genetic Algorithm and Engineering Design", John Wiley & Sons, Inc., 1996
10. 철도연, CTC 통합을 위한 Traffic Management 실시설계 자문, 2001. 12

APPENDIX

여객문제에 대한 Pseudo Code

```
CONGA2 {
  set initial feasible solution.
  for generation = 1:GENERATIONS
    for CHR = 1:POPSIZE
      CalFitness(CHR);
    end
    Reproduce_MOON1(generation);
  end
}

CalFitness(CHR) {
  if 0 == CalculateTime(CHR)
    return infeasibility.
  else
    calculate the fitness value.
  end
}

CalculateTime(CHR) {
  A,B departure is all same.
  for all gene
    if the gene is possible to arrive C according to CHR.
      calculate the C-arrival time of the gene.
    else
      return infeasibility.
    end
    calculate C-departure/D-arrival time of the gene with FCFS rule.
  end
}

Reproduce_MOON1(generation) {
  if the generation is the first generation
    copy the initial feasible solution to all others chromosomes.
  else
    get the chromosome whose fitness value is best and overwrite it to 8, 9,
    10th chromosomes.
    get the chromosome whose fitness value is 2nd best and overwrite it to
    6, 7th chromosomes.
    get the chromosome whose fitness value is 3th best and overwrite it to
    5th chromosomes.
    for CHR = 5:10 // 1,2,3,4th best chromosomes survive.
      PIVOT = SelectPivot_UNIF(generation,CHR);
      Pivoting(CHR,PIVOT);
    end
  end
}
```