

공통 신호주기 제약을 고려한 Particle Swarm Optimization 기반 다중 교차로 교통 신호 최적화

박진환¹ · 허재석^{2*}

¹한국공학대학교 스마트팩토리융합학과 / ²한국공학대학교 경영학부

Particle Swarm Optimization-based Traffic Signal Optimization for Multiple Intersections with Common Signal Cycle Constraints

Jinhwan Park¹ · Jaeseok Huh²

¹Department of Smart Manufacturing, Tech University of Korea

²Department of Business Administration, Tech University of Korea

Traffic congestion frequently occurs at multiple intersections with high vehicular traffic, particularly during rush hour. Inconsistent signal cycle lengths across intersections can hinder traffic flow, even when an upstream intersection grants a green light. To address this, the signal cycle lengths of multiple intersections should be identical. Furthermore, practical constraints must be considered: (1) all signal times must be integers, (2) phase sequences should remain fixed to prevent accidents and driver confusion, and (3) pedestrian safety must be ensured through sufficient crossing time. This paper proposes a novel particle swarm optimization-based method for optimizing traffic signals at multiple intersections, addressing the need for consistent signal cycle lengths and the practical constraints. The proposed method was evaluated using the traffic simulator SUMO (Simulation of Urban Mobility). Compared to existing signals, the proposed method reduced the average waiting time of vehicles passing through multiple intersections by 10% during morning rush hour, 20% during normal hour, and 31% during evening rush hour.

Keywords: Multiple Intersections, Common Signal Cycle, Traffic Signal Optimization, Traffic Congestion, Particle Swarm Optimization, Simulation of Urban Mobility

1. 서론

교통 흐름이 집중되는 교차로에서는 교통 혼잡으로 인해 사고 위험이 높으며, 운전자들이 많은 불편을 겪는다. 또한, 차량의 정체가 지속됨에 따라, 유해가스의 배출량이 증가한다. 특히, 출·퇴근 시간에는 교통량이 급증하여 평시에 비해 교차로를 통과하는데 많은 시간이 소요될 수 있다(Kim *et al.*, 2009). 이러한 문제는 교차로가 연속된 다중 교차로 환경에서 더욱 심각해진다.

교통 혼잡을 해결할 수 있는 방법에는 크게 세 가지가 있다.

첫째, 센서 및 카메라, AI기술을 활용하여 신호를 제어하는 스마트 도로를 구축하는 방법이다(LEE *et al.*, 2023). 둘째, 도로를 확장하기 위해 추가 차선을 건설하는 방법이다. 그러나 두 방법은 과도한 비용이 요구되거나 도로 건설과 관련된 정책 때문에 현실에 적용하기 어렵다. 세 번째 방식은 교통량에 따라 신호주기의 길이(Signal Cycle Length)와 현재 신호 단계인 현시(Phase)의 녹색 시간을 조절하는 신호 조정 방식이다. 우리나라에서는 주로 이 방법을 교차로에 도입하여 신호시스템을 운영하고 있다(Korea Road Traffic Authority Transportation Science Research Institute, 2023).

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(2020R1G1A1007453).

* 연락처 : 허재석 교수, 15073 경기도 시흥시 산기대학교로 237 정왕동 한국공학대학교 경영학부, Tel : 031-8041-0675, E-mail : jshuh@tukorea.ac.kr
2024년 4월 26일 접수; 2024년 7월 10일 수정본 접수; 2024년 8월 5일 게재 확정.

국내 교차로 신호체계는 고정 시간 신호 제어와 감응형 신호 제어로 분류된다. 고정 시간 신호 제어는 과거 교통량 데이터를 기반으로 하루 동안 동일한 신호를 반복하여 교통량 변동에 제한적으로 대응한다(Park and Kim, 2000). 감응형 신호 제어는 차량 검지기를 통해 동적으로 신호를 조절하여 실시간 교통 상황에 빠르게 대응할 수 있지만, 검지기 센서의 빈번한 오작동과 시스템 설치에 많은 비용이 필요하다는 한계가 있다(Pi *et al.*, 2021). 현재 국내에서는 두 방법의 단점을 보완하기 위해 교통량의 변동을 고려하여 출·퇴근 시간과 평시 또는 시간대별로 나눠 현시의 녹색 시간 및 신호주기 길이를 조절하는 시간제 방식(TOD: Time of Day)을 교차로에 적용하고 있다.

TOD방식의 신호 운영을 다중 교차로에 적용할 때는 네 가지 현실적인 제약이 존재한다. 첫째, 교차로 간의 신호 연동을 위해 다중 교차로의 신호주기 길이를 동일하게 설정해야 한다. 특히, 출·퇴근 시간대에 교차로의 처리 용량을 초과하는 교통 수요가 집중될 때, 신호주기의 길이가 다른 상태로 차량의 통행이 지속되면 하루 교차로 접근구간에서 대기하는 차량의 수가 점차 증가하게 된다. 그 결과, 상류 교차로의 녹색 신호가 등화되어도 차량이 하류 교차로를 진입할 수 없는 앞막힘 현상(Spillback)이 발생하게 된다. 이러한 앞막힘 현상이 지속되면 교차로의 차량이 진행할 수 없어 교통 혼잡이 더욱 악화될 수 있다. 이해를 돕기 위해, <Figure 1>에 차량의 진행방향과 본 연구의 대상이 되는 다중 교차로를 나타내었다. 왼쪽에서 차량이 출발할 때, 차량이 처음으로 접하는 교차로를 상류 교차로라 하고, 이후에 접하는 교차로를 하류 교차로라 한다.

둘째, 녹색 신호 시간과 황색 신호 시간은 경찰청 매뉴얼에 따라 실제 적용을 고려하여 정수로 설정되어야 한다(Korean National Police Agency, 2023). 셋째, 현시 순서의 변경은 교차로 내 동선을 변화시켜 사고 발생 가능성을 높일 뿐만 아니라, 운전자들 사이에 혼란을 초래할 수 있으므로 현시 순서를 유지해야 한다. 마지막으로, 신호 조정은 운전자뿐만 아니라 보행자의 안전도 고려하여 횡단보도를 건널 수 있는 최소한의 신호 시간을 확보해야 한다.

따라서, 본 논문에서는 SUMO(Simulation of Urban Mobility)와 메타휴리스틱(Meta-heuristic) 알고리즘을 활용하여 다중 교차로에서의 교통 신호 최적화 방법을 제안한다. 시흥 시청과 시흥경찰서로부터 수집한 신호 정보와 교통량 데이터를 기반으로

SUMO를 통해 다중 교차로를 모델링하고, 다중 교차로의 신호와 차량의 등장 시간과 경로를 설정하였다. 이후, 군집 지능 알고리즘의 일종인 PSO(Particle Swarm Optimization)를 활용하여 언급된 제약 조건을 고려하여 교통 신호 최적화를 수행하였다. 실제 데이터를 기반으로 진행된 실험을 통해, 제안 방법으로 도출된 최적 신호 설정이 기존의 신호 설정보다 차량의 평균대기시간이 더 짧다는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 SUMO와 교통 신호 최적화를 위해 메타휴리스틱을 활용한 연구를 고찰하고, 제3장에서는 다중 교차로의 교통 신호 최적화 문제를 정의하며, 제4장에서는 본 논문에서 제안하는 기법의 구체적인 작동 과정을 기술한다. 제5장에서는 실험환경 설계에 대해 설명하며, 제6장에서는 실험내용과 그 결과를 논하고, 마지막 제7장에서는 본 연구의 결론과 한계점 및 향후 연구를 제시한다.

2. 관련 연구

2.1 SUMO 기반의 교통 혼잡 해소를 위한 연구

SUMO는 오픈 소스 기반의 교통 시뮬레이션 도구로서, 실제 교통수요에 따라 도로 네트워크를 가상으로 구축할 수 있는 기능을 제공한다(Krajzewicz *et al.*, 2002). 차량의 차선, 경로, 출발 시간 및 도착 시간을 개별적으로 정의할 수 있으며, 신호등 제어를 통해 차량의 흐름을 확인할 수 있다(Krajzewicz *et al.*, 2005).

또한, SUMO는 사용자의 편의를 위한 GUI(Graphic User Interface)와 TraCI(Traffic Control Interface)를 제공하며 Python, Java 등 다른 소프트웨어와 연동이 가능하다. 이러한 특징들로 SUMO는 경로 선택, 신호 제어, 차량 추적 시스템 등의 다양한 분야에서 활용되고 있다(Behrisch *et al.*, 2011).

Heo *et al.*(2016)은 SUMO와 수리적 기법을 이용하여 단일 교차로를 통과하는 차량의 CO_2 배출량을 최소화하는 기법을 제안하였다. 실험결과, 제안 기법은 교통량이 적을 때는 기존 신호보다 효과적이었으나, 교통량이 증가함에 따라 기존 신호보다 낮은 성능을 보였다.

SUMO를 활용한 시뮬레이션을 통해, 차량속도와 교통량 밀도의 관계를 고려한 경로 탐색 알고리즘의 성능을 검증한 연구가 수행되었다(Agafonov *et al.*, 2019). 해당 연구에서 제안된 모델은 교통량이 증가할수록 기존의 경로 탐색 알고리즘에 비해 차량의 평균이동시간을 효과적으로 단축시켰다. 그러나 차량 속도의 변화량이 증가할수록 기존의 알고리즘과 큰 차이를 보이지 않았다.

한편, Pi *et al.*(2021)은 SUMO와 강화학습을 활용하여 단일 교차로를 통과하는 차량의 이동시간을 최소화하는 방법을 제안하였다. 이 방법은 고정 신호보다 차량의 이동시간을 7% 감소시키는 결과를 도출하였으나, 다중 교차로에 해당 방법을 적용할 경우 성능이 하락할 가능성이 존재한다는 한계가 있다.

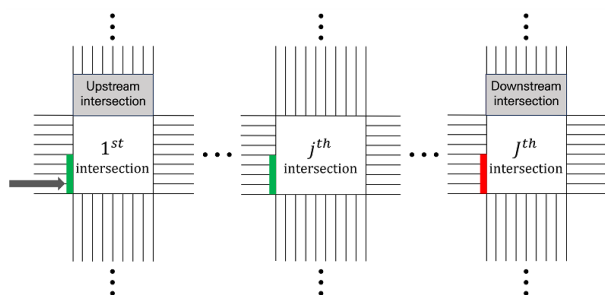


Figure 1. Example of Multiple Intersections

2.2 신호 최적화에 대한 메타휴리스틱 적용 연구

메타휴리스틱은 확률 변수를 활용하여 복잡한 최적화 문제의 근사 최적해를 빠른 속도로 찾기 위한 알고리즘이다(Blum and Roli, 2003). 알고리즘의 종류에 따라, 생물의 진화 혹은 행동에서 영감을 받아 확률적 특성과 지역 검색을 균형 있게 수행하거나, 과거 경험을 활용하는 특징을 가지고 있다(Olafsson, 2006).

해집단 기반의 메타휴리스틱 알고리즘은 초기에 무작위성을 활용하여 해공간내에서 해집단을 생성하고, 이후에는 점진적으로 해집단을 개선하면서 최적해를 탐색한다. 진화론적 알고리즘인 유전 알고리즘(GA : Genetic Algorithm)과 군집 지능 알고리즘인 입자 군집 최적화(PSO), 개미 집단 알고리즘(ACO : Ant Colony Optimization)이 경로 최적화와 신호 최적화 분야에서 주로 사용되고 있다(Fong *et al.*, 2015; Huang *et al.*, 2011; Putha *et al.*, 2012).

Tan *et al.*(2016)은 GA와 수리적 기법을 이용하여 단일 교차로를 통과하는 차량의 평균대기시간을 최소화하는 모델을 개발하였다. 제안 방법은 기존 신호 대비 우수한 성능을 보였으나, 단일 교차로를 대상으로 신호 최적화를 진행하여 다중 교차로에 적용하기에 어려움이 존재한다. 또한, 수식 기반 추정 기법은 다양한 차량 종류와 운전자의 행동 패턴과 같은 실제 교차로의 특성을 충분히 고려하기 어려우며, 신호의 성능을 검증하기 위한 시뮬레이션을 수행하지 않았다는 한계가 있다. Putha *et al.*(2012)은 다중 교차로를 통과하는 차량의 대수를 최대화하기 위한 연구에서 ACO와 GA의 성능을 비교하였다. 교통량이 많은 교차로에서는 ACO가 GA보다 성능이 우수하였으나, 그 외의 상황에서는 GA와 비슷한 성능을 보였다.

한편, PSO는 새나, 물고기 등이 무리를 지어 행동하는 패턴에서 영감을 받아 개발된 해집단 기반의 메타휴리스틱 알고리즘이다(Kennedy and Eberhart, 1995; Ko and Huh, 2022). 입자(particle)의 위치는 해, 군집(swarm)은 해집단을 나타내며, 각 입자는 개인과 집단의 경험을 이용하여 최적의 위치를 찾기 위해 해공간 내에서 이동한다. 모든 입자는 위치와 속도를 가지며, 위치에 따라 입자의 적합도가 산출된다.

입자의 속도는 입자가 이동하던 관성(inertia), 입자가 탐색했던 가장 우수한 해, 군집이 발견한 우수한 해를 활용한 벡터 연산으로 산출되며, 계산된 속도를 이용하여 입자의 위치가 업데이트된다. 알고리즘은 이러한 과정을 반복하며 해공간을 탐색하다가 종료 조건에 도달하면 군집 내의 전체 최적해를 반환한다. 해가 연속적인 실수 값으로 표현되는 최적화 문제에서 PSO의 성능은 다른 메타휴리스틱 알고리즘보다 우수한 결과를 보였다(Ezugwu *et al.*, 2020; Aghababa *et al.*, 2012).

위와 같은 특성으로 인해, PSO는 교통 신호 최적화 문제에 활발하게 활용되고 있다(Wu and Wang, 2016; Chentoufi and Eillaia, 2018; Tan *et al.*, 2021; Deng *et al.*, 2021; Silva *et al.*, 2022). Dabiri *et al.*(2016)은 PSO와 교통 시뮬레이터를 이용하

여 다중 교차로에서 차량의 총 대기시간을 최소화하는 기법을 제안하였다. 제안된 방법은 교통량이 증가할수록 기존 신호보다 우수한 성능을 보였으나, 적은 교통량을 기준으로 실험하였기 때문에 교차로의 처리 용량을 초과하는 정도의 많은 교통량에서는 성능이 하락할 가능성이 존재한다는 한계가 있다.

Jia *et al.*(2019)은 PSO와 수리적 기법을 이용하여 단일 교차로를 통과하는 차량의 총 대기시간과 CO배출량을 최소화하는 연구를 수행하였다. 제안 신호는 기존 신호에 비해 차량의 총 대기시간과 CO배출량을 감소시키는 결과를 도출하였다. PSO의 업데이트 방식에 다른 입자들의 영향력을 고려한 SL-PSO(Social Learning PSO)를 수리적 기법과 함께 활용하여 단일 교차로를 통과하는 차량의 평균이동시간을 최소화하는 시도도 존재하였다(Celtek *et al.*, 2020). 제안 알고리즘은 모든 교통 흐름에 대해 차량의 평균이동시간을 감소시켰고, PSO보다 10% 빠른 속도로 PSO와 동일한 최적 신호를 찾는 결과를 보였다.

Deng *et al.*(2021)은 다중 교차로를 통과하는 차량의 평균대기시간을 최소화하면서 교차로 처리량은 최대화하는 연구를 수행하기 위해 PSO의 업데이트 방식에 다른 세대 입자들의 최적해를 반영한 MEL-PSO(Multi-Exemplar Learning PSO)를 제안하였다. 제안 알고리즘은 모든 교통 흐름에 대해 PSO보다 우수한 성능을 보였다.

그러나 대부분의 다중 교차로의 신호 최적화 연구들은 신호를 결정할 때 가장 중요한 요소인 교차로 간의 신호주기 길이를 고려하지 않았다. 다중 교차로에서 신호주기 길이가 동일하지 않으면, 상류부 교차로의 녹색 신호가 등화 되어도 차량이 하류부 교차로에 진입이 불가능 할 수 있으며, 운전자들의 혼란이 가중될 수 있다. 또한, 선행 연구들은 TOD방식에 부합하는 신호를 제안한 것이 아닌 하나의 신호를 하루 동안 반복하기 때문에 모든 교통 흐름에 효과적으로 대응하는 것에 한계를 보였다. 따라서, 본 논문에서는 TOD방식의 신호체계를 활용하는 다중 교차로를 대상으로, 교차로 간의 신호주기 길이를 고려한 메타휴리스틱 기반의 신호 최적화 기법을 제안하고자 한다.

3. 문제 정의

<Table 1>은 다중 교차로를 정의하는데 필요한 변수에 대한 설명을 나타내고 있다. 하나의 교차로 안에서 현시들의 황색 시간을 제외한 녹색 시간의 합을 공동 신호주기의 길이(SC)라고 정의하며, d 번째 현시는 각 교차로에서 d 번째로 점등되는 녹색 신호를 의미한다. 또한, <Figure 2>는 연속된 2개의 교차로에서 각각 4개의 현시로 구성된 신호등을 가진 다중 교차로의 신호체계 예시를 나타내며, 두 교차로는 공통주기의 값으로 SC를 갖는다. 교차로마다 현시 순서와 현시 개수의 차이는 존재할 수 있다.

Table 1. List of Main Variables for Multiple Intersections

j	the j^{th} intersection ($j=1,2,\dots,J$)
d	the d^{th} phase ($d=1,2,\dots,D_j$)
J	the number of intersections
D_j	the number of phases for j^{th} intersection
SC	common signal cycle length
$T(j,d)$	traffic of volume the d^{th} phase at the j^{th} intersection

j	1				2			
Cycle length	SC				SC			
d	1	2	3	4	1	2	3	4
Phase	↓ ↓	↘ ↘	⇐ ⇐	↗ ↗	⇐ ⇐	↗ ↗	↘ ↘	↓ ↓

Figure 2. Example of Multiple Intersections' Signal System

현시는 차량의 이동 경로를 나타내므로, 차량이 향하는 방향에 따른 교통량 $T(j,d)$ 를 파악하여 현시에 맞는 적절한 신호를 부여해야 교통 혼잡을 최소화할 수 있다. 교차로들의 현시 순서를 유지하면서 공통주기를 갖는 다중 교차로를 통과하는 차량의 평균대기시간을 감소시키는 것이 본 논문의 목표이다.

다중 교차로의 신호를 최적화하기 위해 고려해야 하는 제약 조건은 다음과 같다. 첫째, 교차로 간의 신호 연동을 위해 각 교차로의 신호주기 길이가 모두 같아야 한다. 둘째, 녹색 신호의 시간은 현실 신호체계에서 적용할 수 있도록 정수로 정의한다. 셋째, 현시 순서는 현실의 다중 교차로에서 운영하고 있는 현시 순서와 동일하게 설정되어야 한다. 각 교차로의 현시 순서를 변경하는 것은 교차로 내의 신호체계의 혼동을 일으켜 사고 발생 가능성을 증가시킬 수 있기 때문이다. 마지막으로, 보행자가 횡단보도를 건널 수 있도록 현실적인 최소 녹색 시간을 보장해야 한다.

4. PSO 기반의 다중 교차로 신호 최적화 알고리즘

본 논문에서는 다중 교차로의 제약조건을 고려한 PSO 기반의 다중 교차로 신호 최적화 알고리즘을 제안한다. PSO는 녹색 시간과 같은 연속적인 실수 값을 다루는 최적화 문제에서 우수한 성능을 도출할 수 있는 알고리즘이기 때문에, 신호 최적화 알고리즘으로 PSO를 채택하였다(Ezugwu *et al.*, 2020; Jia *et al.*, 2019). 목적함수는 SUMO를 통해 다중교차로를 통과하는 차량의 평균대기시간을 최소화하도록 설정하였다. <Table 2>는 PSO의 주요 변수에 대한 정의를 나타내며, Algorithm 1은 제안 알고리즘의 전체적인 구조를 포함하고 있다. 또한, Algorithm 1에는 녹색 신호를 정수형태로 변환해주는 Algorithm 2가 내재되어 있고, Algorithm 2는 교차로 간의 신호주기 길이를 동일하

Table 2. List of Main Variables for the Proposed Method

i	the index of particles ($i=1,2,\dots,N$)
t	the iteration of the PSO algorithm ($t=1,2,\dots,Maxiter$)
N	the number of particles
$Maxiter$	the maximum iteration steps
$LB_{j,d}, UB_{j,d}$	the maximum and minimum green signal time of d^{th} phase at j^{th} intersection
P	the number of selected particles for validation
R	the maximum validation step
X_i^t	the i^{th} particle in iteration t
$X_i^t(j,d)$	the green signal time of d^{th} phase at j^{th} intersection for i^{th} particle in iteration t
$V_i^t(j,d)$	the velocity of $X_i^t(j,d)$
w_{min}, w_{max}	the maximum and minimum weight of inertia
c_1	the cognitive coefficient
c_2	the social coefficient
$F(X_i^t)$	the average waiting time of vehicles during X_i^t



Figure 3. Example of X_i^t

게 유지해주는 Algorithm 3을 포함하고 있다.

먼저, lines 2-9는 초기해 생성 단계로, $\sum_{j=1}^J D_j$ 차원(녹색 신호의 총 개수)의 크기를 갖는 위치 벡터 형태의 해들을 정해진 범위 안에서 무작위로 초기화하여 해집단을 형성한다. <Figure 3>에 나타나듯이, $X_i^t(j,d)$ 는 t 시점의 i 번째 해의 j 번째 교차로의 d 번째 녹색 시간을 의미한다. 이후, 각 녹색 시간에 대응되는 속도는 -1과 1 사이의 값에서 초기화된다. 한편, 랜덤으로 생성된 위치 벡터의 성분은 정수형태가 아닐 가능성이 높으며, 다중 교차로의 교차로 간의 신호주기 길이가 동일한 값을 갖지 않게 된다. 따라서, Algorithm 2를 통해서 각 교차로의 신호주기 길이를 통일하고, 모든 신호의 시간을 정수형태로 조정한다. 이에 대한 자세한 내용은 Algorithm 1에 대한 설명 후에 기술하기로 한다.

Algorithm 1 Multiple intersections traffic signal optimization with PSO

Input: $N, Maxiter, J, D_j, P, R, w_{min}, w_{max}, c_1, c_2$

Output: min_gbest

- 1: **Initialization:**
- 2: **for** each particle $i = 1, \dots, N$ **do**
- 3: **for** (j,d) where $(j,d) = (1,1), \dots, (J,D_j)$ **do**
- 4: Initialize the particle's position as

```

1:  $X_i^t(j,d) \sim U([L_{j,d}, U_{j,d}])$ 
5: end for
6: Initialize the particle's velocity as
    $V_i^t(j,d) \sim U([-1,1])$ 
7: Adjust the particle's position using Algorithm 2
8: end for
9: Evaluate each particle's position using SUMO
10: Update:
11: while  $t < Maxiter$  do
12: Update inertia weight using Equation (1):
    $w = w_{max} - (w_{max} - w_{min}) \times \frac{t}{Maxiter}$ 
13: for each particle  $i = 1, \dots, N$  do
14:   for  $(j,d)$  where  $(j,d) = (1,1), \dots, (J,D_j)$  do
15:     Pick random numbers :  $r_1, r_2 \sim U([0,1])$ 
16:     Update particle's velocity using Equation (2):
        $V_i^{t+1}(j,d) = w \cdot V_i^t(j,d)$ 
        $+ c_1 \cdot r_1 (pbest_i^t(j,d) - X_i^t(j,d))$ 
        $+ c_2 \cdot r_2 (gbest^t(j,d) - X_i^t(j,d))$ 
17:   end for
18: Update particle's position using Equation (3):
        $X_i^{t+1} = X_i^t + V_i^{t+1}$ 
19: Adjust particle's position using Algorithm 2
20: Evaluate particle's position using SUMO
21: if  $F(X_i^{t+1}) < F(pbest_i^t)$  then
22:    $pbest_i^t \leftarrow X_i^{t+1}$ 
23: end if
24: if  $F(pbest_i^t) < F(gbest^t)$  then
25:    $gbest^t \leftarrow pbest_i^t$ 
26:   Store  $gbest^t$  in  $gbest$  history
27: end if
28: end for
29: end while
30: Validation:
31:  $min\_gbest = None$ 
32:  $min\_fit = Inf$ 
33: for  $gbest$  in  $gbest$  history[:  $P$ ] do
34:    $fit\_sum = 0$ 
35:   for repetitions  $r = 1, \dots, R$  do
36:      $fit\_sum += F(gbest)$ 
37:   end for
38:   if  $\frac{fit\_sum}{R} < min\_fit$  then
39:      $min\_fit \leftarrow \frac{fit\_sum}{R}$ 
40:      $min\_gbest \leftarrow gbest$ 
41:   end if
42: end for
43: return  $min\_gbest$ 

```

$$w = w_{max} - (w_{max} - w_{min}) \times \frac{t}{Maxiter} \quad (1)$$

$$V_i^{t+1}(j,d) = w \cdot V_i^t(j,d) + c_1 \cdot r_1 (pbest_i^t(j,d) - X_i^t(j,d)) + c_2 \cdot r_2 (gbest^t(j,d) - X_i^t(j,d)) \quad (2)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (3)$$

해집단의 모든 벡터는 SUMO를 이용한 시뮬레이션을 수행하여 각 초기해의 적합도인 차량의 평균대기시간 $F(X_i^t)$ 를 부여받는다. 이는, 해집단의 i 번째 입자가 경험했던 최고 적합도를 가진 벡터의 위치 $pbest_i^t$ 와 군집 전체의 경험에서 가장 적합도가 높았던 벡터 위치 $gbest^t$ 를 찾는 데 사용된다.

line 10-29는 초기화 단계가 끝난 후에 진행되는 속도 및 위치 업데이트 단계이다. w 는 입자의 속도에 대한 가중치로서 식 (1)을 통해 업데이트가 진행됨에 따라 w 의 값은 점점 감소한다. i 번째 위치 벡터의 녹색 시간에 대한 다음 시점의 이동 속도 $V_i^{t+1}(j,d)$ 는 식 (2)를 통해 계산된다. c_1, c_2 는 초매개변수이며, r_1, r_2 는 0과 1사이의 무작위 값을 갖는다. 모든 녹색 시간의 속도 업데이트가 끝나면, 식 (3)에 의해 $t+1$ 시점의 i 번째 위치 벡터인 X_i^{t+1} 이 산출된다. 이와 같이, 벡터의 속도 및 위치 업데이트가 수행되고 나면, 초기화 단계처럼 Algorithm 2를 이용하여 제약조건을 준수하도록 해를 수정한다.

이후, $pbest_i^t$ 와 $gbest^t$ 의 업데이트가 수행되며, 후자는 $gbest$ history에 따로 저장된다. 이는, 업데이트 종료 후 검증 단계를 진행하기 위해 군집 내의 우수한 벡터를 보관하기 위함이다. 속도 및 위치 업데이트의 과정이 반복되며 종료조건에 도달하면 해 업데이트는 종료된다.

마지막으로, line 30-43은 메타휴리스틱과 실험환경의 확률적인 특성을 고려하여, 제안 기법으로도 도출되는 신호의 강건성을 보장하기 위한 Validation 단계이다. 실험환경에는 확률적인 특성이 존재하여 하나의 벡터는 시뮬레이션마다 매번 차량의 평균대기시간이 다르게 도출된다. 따라서, $gbest$ history에 저장된 상위 P 개의 해에 대해서 SUMO를 통해 R 번의 시뮬레이션을 수행하여 적합도의 평균을 구한다. 이 평균값이 가장 작은 해를 min_gbest 에 할당하고 알고리즘이 종료된다. 이렇게 도출된 min_gbest 는 교통량 변동에 강건하게 대응하면서 차량의 평균 대기시간을 최소화는 신호가 된다.

Algorithm 2는 신호의 실제 적용을 위해 녹색 시간을 정수 값으로 변환하는 과정을 나타낸다. 하나의 벡터 안에서 각 교차로에 대응되는 벡터의 신호주기 길이 $Changed_Cycle$ 이 SC 와 같지 않을 경우, line 5와 같이 녹색 시간을 벡터의 신호주기 길이로 나눔으로써 각 신호의 비율을 도출한 뒤, SC 를 곱하고 내림연산을 수행하여 녹색 시간을 업데이트 해준다. 이 과정을 통해서 모든 녹색 시간은 정수값을 갖게 되지만, 여

전히 신호주기 간의 오차 E 가 존재한다. 오차를 제거하여, 모든 교차로의 신호주기 길이를 SC 와 동일하게 유지하기 위해 Algorithm 3이 수행된다.

Algorithm 2 Adjusting signal time to integers

Input: *particle's position, J, D_j, SC*

Output: *particle's position*

```

1: for each intersection number  $j = 1, \dots, J$  do
2:    $Changed\_Cycle = \sum_{d=1}^{D_j} X_i^t(j, d)$ 
3:   if  $Changed\_Cycle \neq SC$  then
4:     for  $d = 1, \dots, D_j$  do
5:        $X_i^t(j, d) \leftarrow \left\lfloor \left( \frac{X_i^t(j, d)}{Changed\_Cycle} \right) \cdot SC \right\rfloor$ 
6:     end for
7:      $E = \sum_{d=1}^{D_j} X_i^t(j, d) - SC$ 
8:     Adjust the particle's position using Algorithm 3
9:   end if
10: end for
11: return particle's position

```

Algorithm 3 Maintaining consistent signal cycle lengths for multiple intersections

Input: *particle's position, T, E*

Output: *particle's position*

```

1:  $Total\_T = sum(T)$ 
2: if  $E > 0$  then
3:   for  $e = 1, \dots, E$  do
4:      $j', d' = argmin_{j, d} T(j, d)$ 
5:      $X_i^t(j', d') -= 1$ 
6:      $T(j', d') += Total\_T$ 
7:   end for
8: else
9:   for  $e = 1, \dots, |E|$  do
10:     $j', d' = argmax_{j, d} T(j, d)$ 
11:     $X_i^t(j', d') += 1$ 
12:     $T(j', d') -= Total\_T$ 
13:  end for
14: end if
15: return particle's position

```

시흥 시청에서 제공받은 교통량 데이터를 바탕으로 각 교차로의 현시에 따른 평균 교통량 $T(j, d)$ 를 사전에 정의하였으며, Algorithm 3은 T 를 기반으로 현시 시간을 조정하여 다중 교차로의 신호주기 길이를 통일하는 과정이다. 구체적으로, E 가 양수이면 교통량이 적은 현시의 녹색 시간부터 1초씩 감소시키며, 모든 현시에 대해 순차적으로 녹색 시간을 오차만큼 감소시킨다. 반대로, E 가 음수이면 교통량이 많은 현시의 녹색 시간부터 우선으로 녹색 시간을 1초씩 조정해 나아간다. 이

러한 조정은 교통량이 많은 현시에 신호를 더 부여하고 교통량이 적은 현시에 신호를 적게 부여하기 위한 과정이다. Algorithm 2와 Algorithm 3을 통해 해집단의 모든 벡터는 정수 값을 갖게 되고, 모든 교차로들은 같은 신호주기 길이를 갖게 된다.

5. 실험환경 설계

<Figure 4>는 3개의 교차로가 연속적으로 구성된 다중 교차로를 나타내고 있다. 교차로 1은 주변에 공단과 대학교가 위치하여 출·퇴근 시간에 차량들의 통행량이 많은 곳이며, 교차로 1→교차로 3의 방향은 퇴근 시간대에 운전자들이 필수적으로 통과해야 하는 도로이다. 반대로, 출근 시간에는 교차로 3→교차로 1로 향하는 차량들이 많고, 평시에는 모든 방향에 대해서 교통량이 비교적 균일하다는 특징을 가지고 있다.

시흥 경찰서에서 제공한 다중 교차로의 신호체계는 <Figure 5>와 같이 구성되어 있다. 3개의 교차로는 각각 신호등이 존재하고, 각 신호등은 5개의 현시로 이루어져 있으며, 이들의 신호주기 길이는 165초이다.

제안 기법의 파라미터는 실험을 통해 차량의 평균대기시간을 최소화하는 파라미터를 찾아 다음과 같이 설정하였다. $N=20$, $Maxiter=100$, $w_{max}=1$, $w_{min}=0.5$, $c_1=1$, $c_2=1$ 로 설정하였으며, $P=5$, $R=10$ 으로 설정하여 *gbest history*에서 *gbest*를 5개 뽑아서 10번의 시뮬레이션을 반복하여 차량의 평균대기 시간을 최소화하는 신호를 도출하도록 하였다. 파라미터의 값을 결정하기 위한 실험의 결과는 Appendix에 첨부되어 있다.

제안 알고리즘의 성능 평가를 위해 SUMO와 Python을 연동하여 다중 교차로의 교차로 및 신호를 모델링하였다. SUMO를 이용한 시뮬레이션 환경을 설계하기 위해 교차로 크기와 신호 정보 및 교통량 데이터를 종합하는 과정은 <Figure 6>과

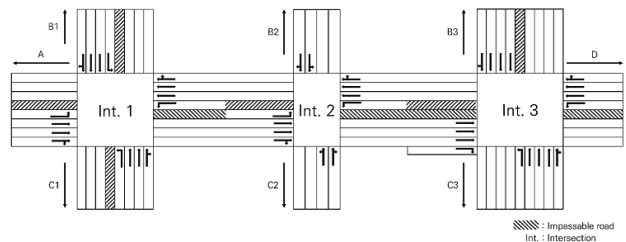


Figure 4. Example of Multiple Intersections with Three Consecutive Junctions

Intersection	Intersection 1					Intersection 2					Intersection 3				
Cycle length	165 (sec)					165 (sec)					165 (sec)				
Sequence	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Phase															

Figure 5. Illustration of Phase Sequence and Common Signal Cycle for Multiple Intersections

같다. 교차로들의 꼭짓점을 정의하는 nod.xml과 nod.xml에 정의된 node와 node를 이어주면서 차선의 수를 정하는 edg.xml을 정의하였으며, 신호등에 신호 정보를 부여할 수 있는 tls.xml를 생성하였다. 교차로와 신호등이 정의된 파일을 통합하여 net.xml 만들어 시뮬레이션을 수행할 수 있는 환경을 구축하였다.

net.xml에 차량을 등장시키기 위해, Python을 사용하여 차량의 등장 시간, 속력, 경로를 정의하는 rou.xml을 생성하였다. Traffic Data.py는 시흥 시청에서 제공받은 교통량 데이터를 기반으로 차량의 등장 시간을 지수분포로 생성하고, 개별적으로 차량의 경로를 지정하는 파일이다. rou.xml에 정의되어 교차로에 등장하는 차량은 시뮬레이션에서 차량의 평균대기시간을 계산하는데 활용하였다. 또한, 매 시뮬레이션마다 rou.xml을 다시 생성하여 차량의 등장 시간을 다르게 설정하였다.

실제 도로에서 통행하는 차량의 크기와 종류를 시뮬레이션에 반영하기 위해 교차로에 등장하는 차량을 4~5m 길이의 승용차와 11m 길이의 버스로 설정하였으며, 차량 간 간격은 2~2.5m로 가정하였다. 또한, 차량은 횡단보도가 녹색 신호이면 우회전을 할 수 없다고 가정하였다. 직진 신호에는 보행자의 통행이 가능하다는 것을 가정하여 모든 직진 현시에 해당하는 $L_{j,d}$ 를 25초로 제한하며, 그 외 현시의 $[LB_{j,d}, UB_{j,d}]$ 는 기존 신호의 녹색 시간을 기준으로 ± 10 초 이내로 설정하였다.

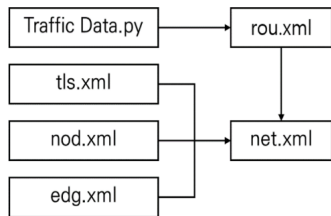


Figure 6. Data Flowchart of the SUMO Framework

6. 비교 실험 및 결과

본 논문에서 신호 최적화 알고리즘으로 PSO를 채택한 정당성을 입증하기 위해, Putha *et al.*(2012)의 연구에서 제안된 GA와 ACO를 구현하여 제안 알고리즘과 비교 실험을 수행하였다. 선행 연구에서는 신호주기의 길이를 동일하게 유지하는 제약을 고려하지 않았기 때문에, GA와 ACO에도 Algorithm 2와 Algorithm 3을 적용하여 제약을 준수할 수 있도록 구현하였다. 또한, PSO의 연산 시간과 동일한 시간 조건 내에서 GA와 ACO의 실험을 수행하였다. 시흥 시청에서 제공받은 교통량 데이터를 활용하여 퇴근 시간(17~20) 동안 3시간의 시뮬레이션을 수행하였으며, 현실과 유사한 교차로 상황을 재현하기 위해 시뮬레이션의 시작 전 1시간을 warm-up기간으로 가정하고, 이후 3시간에 대한 시뮬레이션을 통해 차량의 평균대기시간을 평가하였다.

Table 3. Comparisons of PSO, GA and ACO for Traffic Signal Optimization

	PSO	GA	ACO
Evening rush hour	95.017	98.658	101.223

<Table 3>은 세 가지 알고리즘으로 도출된 신호에 대해 각각 100번의 이산 사건 시뮬레이션을 수행하여 산출된 평균대기시간을 나타낸다. 실험 결과를 통해, PSO가 GA와 ACO보다 우수한 성능을 보이는 것을 확인할 수 있었다. GA는 해공간이 이산적이거나 순서를 결정하는 문제에 강점을 가지고 있으며, ACO는 경로 최적화 문제에 적합한 알고리즘으로 알려져 있다. 하지만 해공간이 연속적인 실수 값으로 표현되는 신호 최적화 문제에서는 PSO 대비 우수한 해를 탐색하기에 한계가 존재하는 것으로 보인다. 따라서, 본 논문에서는 신호 최적화 알고리즘으로 PSO를 사용하였다.

<Table 4>는 제안 기법을 이용하여 도출된 신호와 시흥 경찰서에서 제공받은 기존 신호를 비교 실험한 결과이다. 구체적으로, 시흥 시청에서 제공받은 교통량 데이터를 활용하여 출근 시간(06~09)과 평시(11~14), 퇴근 시간(17~20)으로 시간대를 구분하여 각각 3시간의 시뮬레이션을 진행하였다. 세 가지 메타휴리스틱의 실험 환경과 동일하게 warm-up기간을 설정하였으며, 기존 신호와 제안 신호를 이용하여 각각 100번의 이산 사건 시뮬레이션을 수행하여 차량의 평균대기시간을 산출하였다.

실험 결과를 통해 제안 신호가 모든 시간대에 기존 신호보다 더 짧은 차량의 평균대기시간을 기록하는 것을 확인하였으며, 기존 신호 대비 제안 신호의 평균대기시간 감소율은 출근 시간은 약 10%, 평시는 약 20%, 퇴근 시간은 약 31%로 측정되었다.

기존 신호와 제안 신호의 녹색 신호 시간을 나타내고 있는 <Table 5>를 보면, 제안 신호가 기존 신호보다 차량의 평균대기시간을 단축시킨 이유를 확인할 수 있다. 구체적으로, A↔D 방향으로 향하는 교차로 1의 4번 현시, 교차로 2의 4번 현시, 교차로 3의 5번 현시의 녹색 신호의 시간은 증가하였고, B↔C 방향으로 향하는 교차로 1의 1번 현시, 교차로 2의 1번 현시, 교차로 3의 1번 현시의 녹색 신호 시간은 감소하였다. 이런 결과는 제안 기법이 각 방향의 교통량에 따라 녹색 신호 시간을 늘이거나 줄이는 방향으로 해를 도출한 것으로 해석된다.

Table 4. Comparisons of the Performance(sec) between Proposed and Current Signals

	Morning rush hour	Normal hour	Evening rush hour
Current signal	70.881	60.368	137.346
Proposed signal	63.531	48.156	95.248
Reduction ratio(%)	10.37	20.229	30.651

Table 5. Case Results of the Traffic Signal Optimization

Time zone	Intersection	Intersection 1					Intersection 2					Intersection 3				
	Cycle length	165(sec)					165(sec)					165(sec)				
	Sequence	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Morning rush hour	Current signal	35	20	52	38	20	33	15	65	32	20	47	25	18	24	51
	Proposed signal	35	20	48	47	15	24	12	73	41	15	47	22	23	27	46
Normal hour	Current signal	45	17	27	58	18	41	17	27	52	28	66	22	18	4	55
	Proposed signal	37	17	24	68	19	35	15	29	60	26	56	19	17	8	65
Evening rush hour	Current signal	31	17	27	38	52	33	17	22	32	61	66	22	18	4	55
	Proposed signal	24	21	22	47	51	32	12	27	42	52	56	18	21	6	64

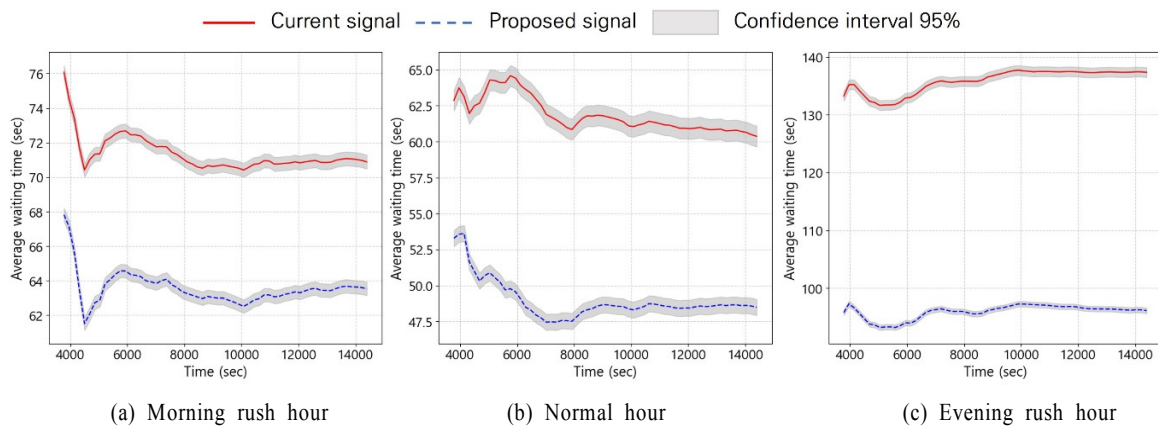


Figure 7. Change in Average Waiting Time during Simulation

한편, 출근 시간대의 A←D 방향은 교통량이 많기 때문에 교차로 1의 3번 현시의 녹색 신호 시간이 증가할 것으로 예상되었으나, 제안 신호에서는 해당 현시의 녹색 신호 시간이 감소하였다. 또한, 교통량이 많은 퇴근 시간대의 A→D 방향에 해당하는 교차로 2와 3의 3번 현시의 녹색 신호 시간을 감소시키는 결과를 기대하였으나, 제안 신호는 해당 현시의 녹색 신호 시간을 증가시키는 경향을 보였다. 이러한 변화에도 A↔D 방향의 평균대기시간이 감소할 수 있었던 이유는 해당 방향의 직진 신호에 해당하는 교차로 1, 2의 4번 현시와 교차로 3의 5번 현시의 녹색 신호 시간이 크게 증가했기 때문인 것으로 추측된다.

<Table 5>에 정의된 신호를 대상으로 100번의 이산 사건 시뮬레이션을 수행한 후, 차량의 평균대기시간의 신뢰구간 95%를 그래프로 나타낸 결과를 <Figure 7>에 도시하였다. 평시 시간을 나타내는 (b) 그래프를 보면, 제안 신호가 시뮬레이션의 4,500~8,000초 구간에서 기존 신호보다 차량의 평균대기시간을 급격하게 감소시키는 것을 확인할 수 있다. 퇴근 시간의 (c) 그래프에서는 기존 신호가 시뮬레이션 초반인 5,000초부터 차량의 평균대기시간을 점차 증가시키는 반면, 제안 신호는 차량의 평균대기시간을 안정적으로 유지시키는 것을 볼 수 있다. 전반적으로 제안 신호의 평균대기시간 그래프는 기존 신호의 값들을 y축 기준으로 아래 방향으로 평행이동한 형태를

보이며, 특정 구간에서는 감소폭이 크게 나타나는 것을 볼 수 있다. 또한, 제안 신호의 그래프는 기존 신호보다 신뢰구간의 폭이 좁기 때문에 신호의 기댓값에 대한 불확실성을 줄일 수 있다.

<Figure 8>은 다중 교차로의 방향에 따른 차량의 평균대기 시간을 나타낸다. 제안 신호는 A↔D 방면과 같은 차량의 통행이 많은 도로의 차량의 평균대기시간을 크게 감소시켰다. 한편, B↔C 방면은 차량의 평균대기시간이 늘어났지만, 이러한 손실을 감내하더라도 교통량이 많은 방향의 평균대기시간 감소폭이 훨씬 컸기 때문에 종합적인 성능이 더 우수하게 도출된 것으로 보인다. 또한, 제안 신호가 기존 신호에 비해 차량 통행이 많은 A↔D 방면에서 차량의 평균대기시간의 분산이 낮은 것을 확인할 수 있다. 이는 제안 기법이 차량의 정체가 심한 구간의 교통량 변화에 강건한 신호를 도출할 수 있음을 시사한다.

본 연구의 실험은 Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz, 32개 CPU, 256GB RAM을 갖춘 서버 환경에서 수행하였다. SUMO를 통해 1회 시뮬레이션을 수행하는데 소요되는 시간을 *simulation time*이라고 하며, 본 연구의 서버 환경에서 *simulation time*은 약 117±2초가 소요된다. 구체적으로, 시뮬레이션에서 하나의 스텝은 1초를 뜻하며, 1시간의 warm-up기간과 3시간의 시뮬레이션은 총 14400 스텝으로 이

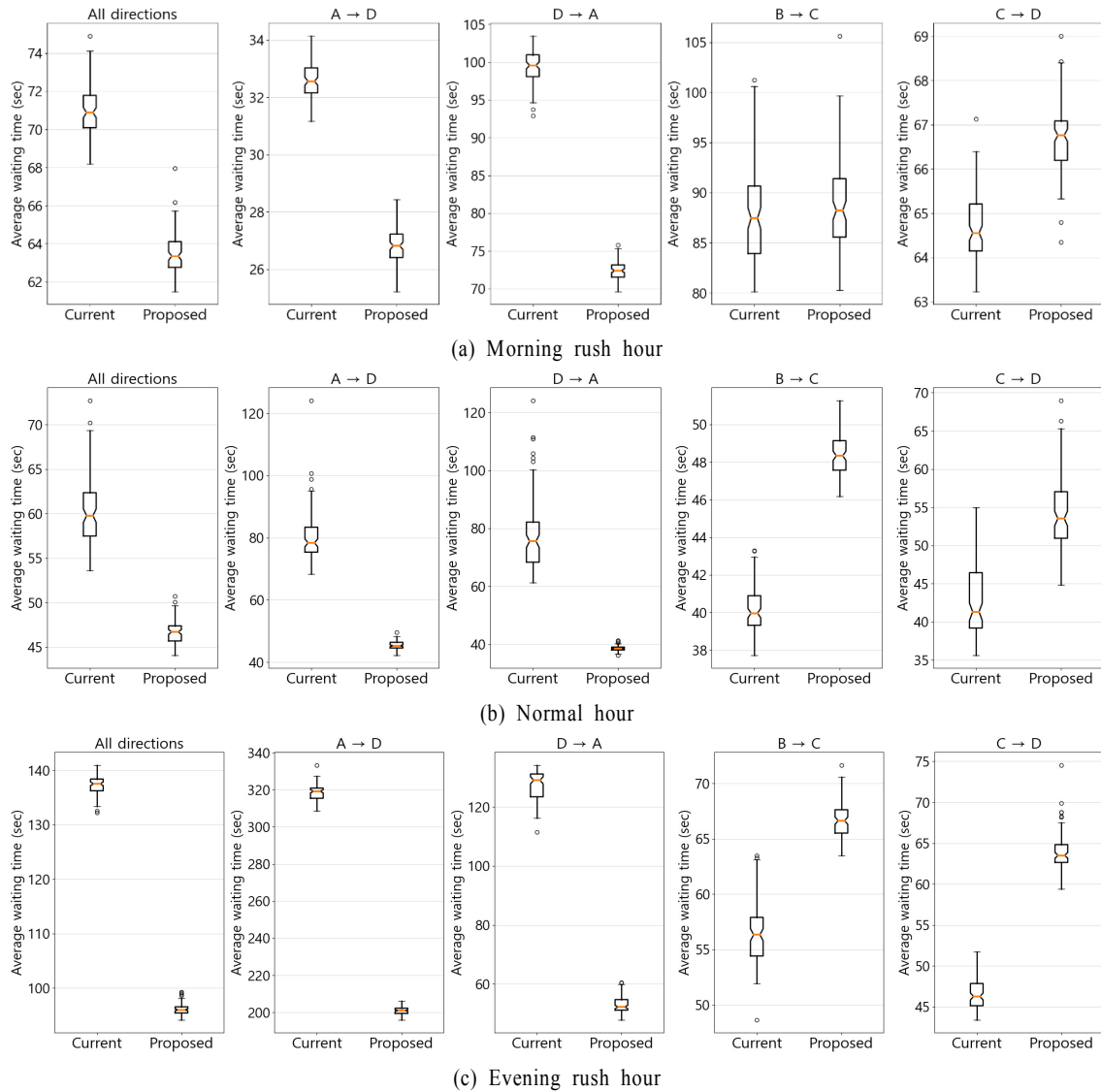


Figure 8. Distribution of Average Waiting Time Across Directions at Multiple Intersections

루어져 있고, 스텝 당 연산 시간은 약 0.08 ± 0.02 초이다. 또한, 시뮬레이션이 수행되는 동안 다중 교차로를 통과한 차량의 총 수는 출근 시간은 약 24,200대, 평시는 약 16,500대, 퇴근 시간은 약 19,500대이다.

Table 6. Computation time(min) comparisons

N	$Maxiter$	P	R	Time	Performance(sec)
10	50	3	15	1064	109.425
		5	10	1072	99.728
	100	3	15	2030	98.566
		5	10	2046	95.901
20	50	3	15	2036	98.287
		5	10	2047	97.748
	100	3	15	3988	96.476
		5	10	3998	95.562

제안 기법은 $simulation\ time \times (N \times Maxiter + P \times R)$ 초의 연산 시간이 요구되며, 총 연산 시간은 $N, Maxiter, P, R$ 값에 따라 달라진다. <Table 6>은 $w_{min} = 0.5, c_1 = 1, c_2 = 2$ 를 고정하여 $N, Maxiter, P, R$ 값의 변화에 따른 연산 시간을 나타내고 있다. 제안 기법은 긴 연산 시간이 필요하지만, 교통량에 큰 변화가 없을 경우 사전에 도출한 신호를 다중 교차로에 적용할 수 있기 때문에 연산 시간은 큰 문제가 되지 않는다.

7. 결론 및 향후계획

본 논문에서는 다중 교차로를 통과하는 차량의 평균대기시간을 최소화하기 위해, PSO 기반의 교통 신호 최적화 기법을 제안하였다. 제안 기법은 도출된 신호를 실제 신호체계에 적용하기 위해, PSO의 초기해 생성 및 해 업데이트 단계에서 모든

교차로의 신호주기 길이를 동일하게 유지하고, 녹색 신호 시간을 정수형태로 조정하는 과정을 수행한다. 실험결과, 제안 신호는 기존 신호보다 차량의 평균대기시간이 출근 시간은 약 10%, 평시는 약 20%, 퇴근 시간은 약 31% 감소함을 보였다.

제안 기법은 최적 신호를 도출하여 차량의 평균대기시간을 최소화하여 사고 발생과 운전자들의 불편함을 줄일 수 있을 뿐만 아니라, 유해가스 배출량을 감소시켜 환경에 긍정적인 영향을 미칠 것으로 기대된다. 또한, 본 논문은 교통 정체 문제에 대한 현실적인 해결책으로써, 다중 교차로 신호 설정을 활용하여 지역 사회의 민원을 해결하고 교통 관련 문제를 정량적으로 분석할 수 있는 도구를 확보하였다.

그러나 본 연구에서는 교통량 데이터를 활용할 때 요일에 관계없이 평균 교통량을 사용하였다. 따라서 평일과 주말, 또는 요일별로 교통량을 분석하지 않았기 때문에 요일에 따른 최적의 신호가 도출될 가능성이 남아 있다. 향후 연구에서는 요일에 따른 신호를 제안할 뿐만 아니라 실시간으로 교통량 수집이 가능한 카메라가 설치된 교차로를 대상으로 강화학습을 적용해 볼 계획이다.

참고문헌

- Agafonov, A., Myasnikov, V., and Borodinov, A. (2019), Anticipatory Vehicle Routing in Stochastic Networks Using Multi-Agent System, *2019 XXI International Conference Complex Systems: Control and Modeling Problems*, 91-95.
- Aghababa, M. P., Amrollahi, M. H., and Borjkhani, M. (2012), Application of GA, PSO, and ACO algorithms to path planning of autonomous underwater vehicles, *J. Marine. Sci. Appl.*, 378-386.
- Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011), SUMO - Simulation of Urban MObility: An Overview, *The Third International Conference on Advances in System Simulation*, 23-28.
- Blum, C. and Roli, A. (2003), Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Comput. Surv.*, 35(3), 268-308.
- Celtek, S., Durdu, A., and Ali, M. (2020), Real-time traffic signal control with swarm optimization methods, *Measurement*, 0263-2241.
- Chentoufi, M. A. and Ellaia, R. (2018), A Hybrid Particle Swarm Optimization and Tabu Search algorithm for adaptive traffic signal timing optimization, *2018 IEEE International Conference on Technology Management, Operations and Decisions*, 25-30.
- Dabiri, S. and Abbas, M. (2016), Arterial traffic signal optimization using Particle Swarm Optimization in an integrated VISSIM-MATLAB simulation environment, *2016 IEEE 19th International Conference on Intelligent Transportation Systems*, 766-771.
- Deng, Z.-J., Luo, L.-Y., Zhan, Z.-H., and Zhang, J. (2021), Knowledge Embedding-Assisted Multi-Exemplar Learning Particle Swarm Optimization for Traffic Signal Timing Optimization, *2021 IEEE Congress on Evolutionary Computation*, 248-255.
- Deng, Z. -J., Zhan, Z. -H., Kwong, S., and Zhang, J. (2021), Multi-Exemplar Learning Particle Swarm Optimization for Regional Traffic Signal Timing Optimization with Multi-Intersections, *2021 IEEE International Conference on Systems, Man, and Cybernetics*, 2918-2923.
- Ezugwu, A. E., Adeleke, O. J., Akinyelu, A. A. *et al.* (2020), A conceptual comparison of several metaheuristic algorithms on continuous optimization problems, *Neural Comput & Applic*, 32, 6207-6251.
- Fong, S., Deb, S., and Chaudhary, A. (2015), A review of metaheuristics in robotics, *Computers & Electrical Engineering*, 278-291.
- Heo, N., Youn, H.-S., Lee, J.-I., and Lee, H.-C., (2016), Development of Optimal Traffic Signal Control Algorithm Using each Vehicle Characteristics at the Downtown Intersection, *The Korean Society Of Automotive Engineers*, 802-808.
- Huang, H.-C. and Tsai, C.-C. (2011), Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm, *SICE Annual Conference 2011*, 1338-1343.
- Jia, H., Lin, Y., Luo, Q., Li, Y., and Miao, H. (2019), Multi-objective optimization of urban road intersection signal timing based on particle swarm optimization algorithm, *Advances in Mechanical Engineering*, 11(4), 1-9.
- Kennedy, J. and Eberhart, R. (1995), Particle swarm optimization, *International Conference on Neural Networks*, 1942-1948.
- Kim, J. H., Suh, S., and Kim, T. (2009), Estimation of Total Travel Time for a Year on National Highway Link with AADT, *KSCE Journal of Civil and Environmental Engineering Research*, 29(1), 11-16.
- Ko, G.-J. and Huh, J.-S. (2022), An Effective Hybrid Meta-heuristic Algorithm for Weight Optimization of Artificial Neural Networks, *Journal of the Korean Institute of Industrial Engineers*, 48(2), 227-234.
- Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002), SUMO (Simulation of Urban Mobility) - An open-source traffic simulation, *4th Middle East Symposium on Simulation and Modelling*, 183-187.
- Krajzewicz, D., Brockfeld, E., Mikat, J., Ringel, J., Rössel, C., Tuchscheerer, W., Wagner, P., and Wöslner, R. (2005), Simulation of modern Traffic Lights Contr-ol Systems using the open source Traffic Simulation SUMO, *3rd Industrial Simulation Conference 2005*, 299-302.
- Lee, J.-H., Choi, M.-H., and Kang, W.-E. (2023), Real Road Demonstration Case of Artificial Intelligent Incident Detection Platform Technology in Gangneug, *Korean Society of Transportation*, 20(2), 54-60.
- Marini, F. and Walczak, B. (2015), Particle swarm optimization (PSO), *A tutorial, Chemometrics and Intelligent Laboratory Systems*, 153-165.
- Olafsson, S. (2006), Chapter 21 Metaheuristics, in *Handbooks in Operations Research and Management Science*, Elsevier, 633-654.
- Park, Y.-S. and Kim, C.-O. (2000), Optimal Traffic Signal Control Using an Efficient Dynamic Programming, *Journal of the Korean Institute of Industrial Engineers*, 26(4), 315-324.
- Pi, M.-G., Lee, H.-S., and Chung, M.-Y. (2021), Reinforcement Learning-based Traffic Signal Control under Real-World Constraints, *Journal of KIISE*, 48(8), 871-877.
- Putha, R., Quadrioglio, L., and Zechman, E. (2012), Comparing Ant Colony Optimization and Genetic Algorithm Approaches for Solving Traffic Signal Coordination under Oversaturation Conditions, *Computer-Aided Civil and Infrastructure Engineering*, 14-28.
- Silva, G. O., Rocha, A. M. A. C., Witeck, G. R., Silva, A., Durães, D., and Machado, J. (2022), On Tuning the Particle Swarm Optimization for Solving the Traffic Light Problem, *ICCSA 2022. Lecture Notes in Computer Science*, 68-80.
- Tan, M. K., Chuo, H. S. E., Chin, R. K. Y., Yeo, K. B., and Teo, K. T. K. (2016), Optimization of Urban Traffic Network Signalization

- Using Genetic Algorithm, *2016 IEEE Conference on Open Systems*, 87-92.
- Tan, M. K., Ladillah, M. R., Chuo, H. S. E., Lim, K. G., Chin, R. K. Y., and Teo, K. T. K. (2021), Optimization of Signalized Traffic Network using Swarm Intelligence, *2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology*, 1-6.
- Wu, B. and Wang, D. (2016), Traffic signal networks control optimize with PSO algorithm, *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, 230-23.
- Korean National Police Agency (2023), Traffic Signal Installation Operation Guidelines, accessed Oct 10 2023, https://www.koroad.or.kr/main/board/21/87215/board_view.do?cp=1&listType=list&bdOpenYn=Y&bdNoticeYn=N.
- Korea Road Traffic Authority Transportation Science Research Institute (2023), Development of Multi Intersection Convergence Technology for Real-Time Signal Intersection Operation, accessed Oct 12 2023, <https://www.koroad.or.kr/file/FileDown.do?atchFileId=FID00122273&fileSn=4>.

<Appendix>

PSO Hyperparameter Analysis

하이퍼파라미터에 따른 PSO의 성능을 비교하기 위해, $N=20$, $Maxiter=100$, $w_{max}=1$, $P=5$, $R=10$ 으로 고정하고, w_{min} , c_1 , c_2 값을 변화시키며 실험을 수행하였다. PSO의 각 입자는 w 와 v_1 의 값이 클수록 exploration의 경향성이 증가하며, c_2 의 값이 증가할수록 exploitation에 집중한다(Marini and Walczak, 2015).

퇴근 시간대를 대상으로 파라미터 변화에 따른 차량의 평균 대기시간을 나타내는 <Table A1>을 확인해보면, 파라미터를 $w_{min}=0.5$, $c_1=1$, $c_2=1$ 로 설정하는 것이 가장 좋은 성능을 보였다. 한편, c_1 과 c_2 의 값의 차이가 클 경우, 성능이 감소되는 경향이 나타났다.

결론적으로, 본 연구의 실험에서는 파라미터를 $w_{min}=0.5$, $c_1=1$, $c_2=1$ 로 설정하여 모든 실험을 수행하였다. 해당 파라미터를 활용하여 출근 및 평시 시간대의 실험을 수행한 결과는 본문의 <Table 4>에 작성하였다.

Table A1. Comparison of PSO Hyperparameter Effects on Performance (sec) during Evening Rush Hour

w_{min}	c_1	c_2	Performance
0.2	0.5	2	99.403
	1	2	96.145
	1	1	95.186
	2	1	95.292
	2	0.5	99.751
0.5	0.5	2	106.436
	1	2	95.562
	1	1	95.017
	2	1	96.045
	2	0.5	99.251

저자소개

박진환 : 한국공학대학교 경영학부에서 2024년 학사학위를 취득하고 한국공학대학교에서 스마트팩토리융합학과 석사과정에 재학 중이다. 연구분야는 최적화, Scheduling이다.

허재석 : 서울대학교 산업공학과에서 2013년에 학사, 2019년에 박사학위를 취득하였다. 현재 한국공학대학교 경영학부의 조교수로 재직 중이며, 연구분야는 시스템 최적화, Scheduling, AI의 산업응용이다.