

사용자의 효율적인 엘리베이터 탑승 대기시간을 위한 Hybrid HRN Algorithm 연구

백진우, 염기훈, 정성욱*

Study on the Hybrid HRN Algorithm for Efficient Elevator Boarding Considering the Users' Waiting Time

Jin-Woo Baek, Gi-Hun Yeom, Sung-Wook Chung*

요약 현재 Collective Control 알고리즘은 가장 대중화되어있는 엘리베이터 알고리즘이다. Collective Control 알고리즘은 엘리베이터의 움직임 방향과 사용자의 도착지 방향이 같을 때 사용자가 엘리베이터를 이용할 수 있다. 하지만, 해당 알고리즘은 사용자의 엘리베이터의 대기시간과 여러 대의 엘리베이터를 사용할 경우, 주로 한 엘리베이터만 사용자들의 호출에 반응하는 문제점이 있다. 이러한 문제점을 해결하고자 본 논문에서는 HRN (Highest Response Ratio Next) 알고리즘에서 착안한 새로운 Hybrid HRN 알고리즘을 제안하고자 한다. 일반적으로 HRN 알고리즘은 사용자의 탑승 시간과 하차 시간이 필요하지만 엘리베이터 특성상 사용자의 호출이 언제 발생할지 예측이 어렵다. 따라서 이러한 한계를 극복하고자 본 논문에서는 사용자의 호출 위치와 도착 위치의 거리를 고려하는 Hybrid HRN 알고리즘을 제안한다. 본 논문에서는 실험을 통하여 제안한 Hybrid HRN 알고리즘이, 임의의 100명에 대한 사용자의 평균 대기 시간은 23.34초, 표준 편차는 11.86, 엘리베이터의 총 움직인 거리는 535.2m, 총 운행 시간 84sec, 두 엘리베이터 사이의 운행 Balance는 92m의 성능을 나타냄으로써, 사용자 탑승 대기 시간을 고려했을 때 기존에 제시된 Collective Control, Zoning, 3-Passage 알고리즘보다 보다 우수함을 보여준다.

Abstract Currently, the Collective Control Algorithm is the most popular elevator algorithm. The Collective Control Algorithm allows the user to use the elevator when the direction of movement of the elevator and the direction of the user's destination are the same. However, the algorithm has a problem in that only one elevator responds to a user's call when the user's waiting time and using multiple elevators. To solve this problem, this paper proposes a new hybrid HRN algorithm based on the highest response ratio next (HRN) algorithm. In general, HRN Algorithm requires a user's boarding time and getting off time, but due to the nature of the elevator, it is difficult to predict the user's call in advance. Therefore, to overcome these limitations, this paper proposes Hybrid HRN Algorithm that considers the distance between the user's call location and the arrival location. This paper shows that Hybrid HRN Algorithm, proposed through experiments, has an average waiting time of 23.34 seconds, a standard deviation of 11.86, a total moving distance of 535.2m, a total operating time of 84sec, and a driving balance between the two elevators is 92m, which is superior to the previously suggested Collective Control, Zoning, and 3-Passage Algorithm.

Key Words : Elevator Algorithm, Hybrid HRN, Zoning, 3-Passage, Waiting Time

This research is financially supported by Changwon National University in 2021~2022.

*Corresponding Author : Department of Computer Engineering, Changwon National University (swchung@changwon.ac.kr)

Received February 15, 2022

Revised February 18, 2022

Accepted February 22, 2022

1. 서론

현대의 세상은 컴퓨터에 의해 움직인다. 컴퓨터가 없는 세상은 상상할 수도 없고, 주변을 보면 컴퓨터가 자리 잡지 않은 곳은 없다. 하지만 우리가 하루에도 몇 번씩 타게 되는 엘리베이터 역시 작은 컴퓨터에 의해 움직인다는 사실을 아는 사람은 그다지 많지 않다.

엘리베이터 또는 승강기(이하 엘리베이터)는 동력을 사용하여 빌딩, 대형 선박 또는 구조물에서 사람이나 화물을 나르는 수직 운송 장치(Vertical Transportation Machine)이다. 행정안전부고시 제 2020-75호 승강기 안전운행 및 관리에 관한 운영규정에 따르면 승객용 엘리베이터는 500 ~ 2700kg 또는 450 ~ 1,600kg(6인승~24인승)을 운반할 수 있으나, 화물용 엘리베이터는 45,000kg까지 나를 수 있다. 승객용 엘리베이터는 화재에 견딜 수 있는 강철로 된 이중 문을 달며, 내부에 비상 인터폰, 전화 설비 등이 설치되어 있어 위급 상황에 대처할 수 있게 규정되어 있다[1].

최근 컴퓨터의 발전과 다양한 스케줄링 방법을 통해 엘리베이터 운행 거리 및 시간의 단축, 사용자의 평균대기 시간, 유전 알고리즘과 디지털 카메라를 이용한 엘리베이터군 제어[2], 음성인식 기반 엘리베이터 제어시스템 설계[3], 스마트 빌딩을 위한 센서 기반의 효율적인 엘리베이터 스케줄링[4] 그리고 엘리베이터

의 효율적인 전력 소모 등 다양한 방면에서 연구가 이루어지고 있다. 일반적인 엘리베이터 스케줄링은 사용자가 호출 버튼을 누르면 엘리베이터 방향과 사용자의 움직임을 고려하여 움직인다. 만약 엘리베이터의 방향과 사용자 움직임 방향이 같지 않다면 사용자는 엘리베이터 움직임 방향과 같을 때까지 대기하여야 한다. 이러한 한계 때문에 사용자의 엘리베이터 대기시간이 증가하고 엘리베이터 전력 소모까지 낭비된다. 예를 들어 그림1과 같이 엘리베이터가 1층에서 15층으로 움직이고 있을 때 사용자는 8층에서 1층으로 가고자 하면 엘리베이터가 15층을 갔다가 다시 8층으로 내려올 때까지의 엘리베이터 왕복 시간을 기다려야 한다. 또한, 두 대의 엘리베이터가 있다면 한 대의 엘리베이터가 사용자의 호출을 모두 처리하는 경우도 발생할

다. 이러한 문제점을 해결하기 위해 본 논문에서는 기존의 CPU 스케줄링 기법의 하나인 HRN 알고리즘 [5][6][7]에서 새로이 착안한 Hybrid HRN 알고리즘을 제안하고자 한다. HRN 알고리즘은 운영체제가 여러 프로세스 입력이 들어왔을 때 프로세스 실행 우선순위를 정하기 위한 기법이다. HRN 스케줄링 기법은 SJF(Shortest Job First) 알고리즘의 약점인 긴 작업과 짧은 작업 사이의 불평등을 보완하려는 방법이다. 우선순위를 계산하는 공식은 (대기 시간 + 서비스 실행 시간)/서비스 실행시간 = 시스템 응답 시간을 통해 구

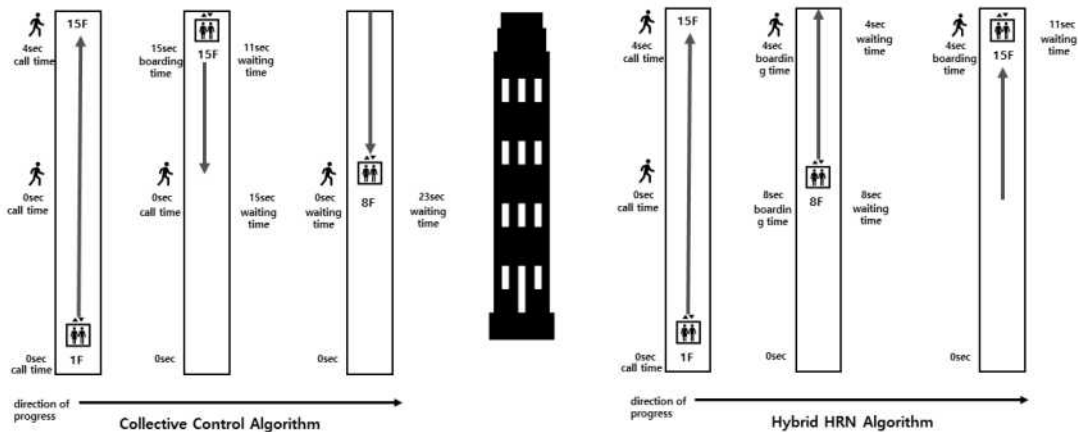


그림 1. 엘리베이터 움직임 전체 구성도
Fig. 1. Elevator movement overall diagram.

해진다. 즉, 시스템 응답 시간이 커질수록 우선순위가 높아진다. 하지만 서비스 실행 시간을 구하는 과정에서 엘리베이터는 사용자가 엘리베이터를 타고 목적지에 도착했을 때까지의 시간이 필요하다. 하지만 엘리베이터의 특성상 엘리베이터에 탑승하려는 탑승자의 호출이 언제 들어올지 알 수 없다. 이러한 문제점을 해결하기 위해 본 논문을 통해 새롭게 제안하는 Hybrid HRN 알고리즘에서의 서비스 실행 시간은 사용자가 탑승할 층에서 도착할 층의 거리 차이이다. 도착된 우선순위에 따라서 엘리베이터 탑승자를 탐색하면 탑승자의 대기시간을 줄일수 있다. 또한, 엘리베이터의 현재 층과 사용자의 도착 층을 고려하지 않아 한 대의 엘리베이터가 사용자의 탑승을 독점하는 것을 방지한다.

2. 관련 연구

엘리베이터 알고리즘을 조사하고 이에 대한 특징들을 조사한다. 엘리베이터 알고리즘의 종류 및 특징은 다음과 같다.

2.1 엘리베이터 시스템

엘리베이터 시스템에서 이용자의 서비스 요구는 승강 호출(Hall Call)과 차내 호출(Car Call)로 구성된다. 승강호출은 각 층의 승강에서 이용자가 엘리베이터 사용의사를 등록하는 호출로 상향승강호출과 하향승강호출의 두 종류가 있다. 차내 호출은 등록된 승강호출이 서비스됨에 따라 승강기에 승차한 이용자에게 의하여 발생하는 목적 층을 포함한 이송요구가 된다[8].

2.2 First In First Out(FIFO)

엘리베이터 알고리즘에 사용할 수 있는 가장 쉬운 접근법으로써 엘리베이터를 이용하려는 사용자가 요청이 들어온 순서대로 엘리베이터가 출발지로 이동해 사용자를 태우고 사용자의 목적지에 내려준 후 다음 요청을 수행하는 것을 반복하는 기본적인 알고리즘이다. 특징은 엘리베이터에 요청을 보낸 순서대로 엘리베이터가 한명만 태우고 동작하기 때문에 엘리베이터의 이동 거리와 사용자의 대기시간이 극단적으로 안 좋게 나타난다.

2.3 Collective Control

우리가 실생활에서 자주 접하는 엘리베이터 알고리즘은 “Collective Control”[9][10], “Elevator Algorithm” 등으로 불리는데 아래와 같이 2가지 규칙으로 구성된다. 첫째, 엘리베이터 내에 탑승객이 있거나 현재 진행 방향의 앞쪽에 같은 진행 방향으로 이동하고자 하는 승객이 있으면, 현재의 방향을 유지한다. 둘째, 현재 진행 방향의 요청들을 전부 처리하고 나면, 반대 방향의 요청을 처리하기 위해 방향을 전환한다. 만약 반대 방향의 요청이 없다면 멈추어 요청을 기다린다.

16층에서 1층으로 엘리베이터가 움직이는 도중 14층을 지날 때 15층에서 1층으로 가는 승강 호출이 발생하면 1층으로 가는 차내 호출을 수행한 후 15층으로 가는 승강 호출이 명령을 수행하기 때문에 에너지 전력 소모와 전체적인 사용자 대기시간이 효율적이지 못하다.

2.4 FUZZY Algorithm

퍼지는 어렵 못하고 애매모호하다는 뜻으로, 퍼지이론은 애매모호한 인간의 언어를 컴퓨터 언어로 표현하고자 한 이론이다. 자동 제어에서는 ‘경계가 불명확한’이라고 이해하는 것이 좋다.

1965년 미국 UC버클리대학의 L. Zadeh 교수가 퍼지집합(Fuzzy Sets)논문을 발표했고[11], 1973년 시스템 자동 제어에 응용할 수 있다고 얘기하였다. 1974년 영국 Mamdani교수가 스티프 엔진 제어에 처음으로 응용하여 실용가능성을 보여 주었다[12].

퍼지 알고리즘은 모호한 상태를 수식 화하여 시스템을 구축하는데 사용하며, 컴퓨터의 0과 1로 이루어진 이진 논리의 한계를 극복할 수 있다.

퍼지 이론을 사용한 관련 연구로 예전 퍼지 제어방식은 임의의 제어명령에 대한 제어결과를 실시간에 예견하는 실시간예측 기법과 대상 시스템에 대한 복수개 제어목적들을 퍼지 제어규칙을 통하여 평가하는 퍼지이론을 결합한 지능형 제어방식이다[13]. 다음으로 퍼지 추론 기법을 이용한 지능형 엘리베이터로 사람이 아쉽다고 느껴지는 상황에서 현재 층과 새로운 승강

호출의 차이를 입력변수로 사용자의 대기시간에 대한 출력변수를 이용해 운행 중인 경로 상에 존재하지 않는 승강 호출에 대해서 경유할지 여부를 결정한 후 제어하는 방법을 사용하였다[14]. 또 퍼지 논리는 승강기 시스템이 매우 복잡하고 그 결과 시스템을 정확히 모델링하는 것이 쉬운 일이 아닐 때 근사적인 시스템 모델링을 결정하는 데에 사용된다. 각 층에서 승강 호출이 발생할 때마다 적절한 if-then 규칙에 의해 선택된다. 퍼지 규칙을 승강기 시스템에 적용하면 장기 대기자들의 최소화, 승객의 평균대기 시간 최소화, 승객의 평균 서비스 시간 최소화, 에너지 소비 감소, 도착 예정자의 즉시 통보가 있다[15].

3. 엘리베이터 알고리즘

엘리베이터 알고리즘에는 Collective Control 알고리즘, Zoning 알고리즘[16][17], 3-Passage 알고리즘이 있다. 하지만 사용자의 대기시간에 초점을 맞춘 새로운 알고리즘을 제안하고자 한다. 제안한 Hybrid HRN 알고리즘은 HRN 알고리즘에서 착안하였다. HRN 알고리즘은 운영체제가 여러 프로세스 입력이 들어왔을 때 프로세스 실행 우선순위를 정하기 위한 기법으로 사용한다. HRN 알고리즘의 우선순위를 구하기 위해서는 대기시간 및 서비스 시간이 필수적으로 필요하다. 대기시간은 사용자가 엘리베이터 호출한 시간부터 탑승한 시간까지이며 서비스 시간은 사용자가 엘리베이터 탑승 시간부터 하차 시간까지이다. 구해진 대기시간과 서비스 시간을 이용하여 표1과 같이 우선순위를 구한다. 하지만 엘리베이터 특성상 사용자의 호출이 언제 발생할지 예측하는 것이 어렵다. 예를 들어 임의의 사용자가 엘리베이터를 탑승한 후 이동 과정에 새로운 사용자가 발생하면 서비스 시간을 예측하기 어렵다. 이러한 한계를 극복하고자 Hybrid HRN 알고리즘은 표1과 같이 서비스 시간을 임의의 사용자 호출 위치와 도착 위치 사이의 거리로 정의한다.

표 1. HRN과 Hybrid HRN 알고리즘의 우선 순위 구하는 공식
Table 1. HRN and Hybrid HRN Algorithm Priority Calculation Formula

	HRN	Hybrid HRN
Waiting Time	(Elevator) Ride Time - Call Time	
Service Time	(Elevator) Drop Off Time - Call Time	(User) Arrival Location - Call Location
Priority	(Waiting Time+Service Time) / Service Time	

3.1 Collective Control Algorithm

Collective Control 알고리즘은 우리가 일상생활에서 사용하는 엘리베이터의 가장 기본적인 알고리즘이다. Collective Control 알고리즘은 엘리베이터의 이동 방향과 사용자의 목적지 방향이 같을 때 우선순위가 높다. 해당 알고리즘의 스케줄링은 그림1과 같다. 각각의 엘리베이터는 현재 층에서 내리거나 탑승할 사람이 있는지 탐색한다. 조건에 만족하는 탑승자가 존재하면 엘리베이터의 상태가 정지로 전환되며 탑승자는 승하차한다. 엘리베이터 사용자가 탑승하면 사용자들은 목적지를 설정한다. 엘리베이터의 최종 목적지는 탑승한 사용자들의 목적지와 엘리베이터 위치의 거리 차이가 가장 많이 나는 층을 최종 목적지로 설정한다. 또한, 엘리베이터의 진행 방향이 결정되면 반대 방향에 대한 호출은 현재 진행 방향의 사용자 호출을 완료한 후 다음 방향으로 전환된다. 엘리베이터가 현재 같은 위치에 존재할 동일한 호출을 받는 경우 난수로 하나의 엘리베이터만 사용자에게 할당한다. 또한, 엘리베이터에 사용자들의 호출이 할당되면 해당 호출은 표2와 같이 다른 엘리베이터에 영향을 미친다. 엘리베이터에 사용자가 탑승하고 최종 목적지가 설정되면 엘리베이터에 사용자가 없을 때까지 위의 과정을 반복한다. 엘리베이터에 탑승한 사용자가 없을 경우 우선순위에 따라 탑승할 사용자의 탑승을 탐색한다.

표 2. Collective Control Algorithm에서 엘리베이터의 관계
Table 2. The relationship of elevators in Collective Control Algorithm.

		Elevator_A	Elevator_B
1 sec	Current Floor	1 st floor	1 st floor
	User Call	7 th floor	7 th floor
	State	move	wait
2 sec	Current Floor	2 nd floor	1 st floor
	User Call	3 rd floor	3 rd floor
	State	move	move
3 sec	Current Floor	3 rd floor	2 nd floor
	User Call	7 th floor	none
	State	wait and move (Users on board)	wait

표 3. Collective Control Algorithm
Table 3. Collective Control Algorithm.

Collective Control Algorithm	
State = going_up or going_down	
Destination = inelevator user's farthest destination	
while	
{	
destination set = floor check	
state set =	
destination - currentlayer >	
0 ? going_up : going_down	
}	

3.2 Zoning Algorithm

Zoning 알고리즘의 기본 동작 방식은 Collective Control 알고리즘과 동일하다. 하지만 해당 알고리즘의 경우 두 대의 엘리베이터의 시작 위치와 엘리베이터가 접근할 수 있는 층의 구역을 분할한다. 예를 들어 Elevator_A의 시작 층이 1층이면 Elevator_B의 시작 층은 10층으로 고정된 층의 값을 가진다. 따라서, Elevator_A가 건물에서 움직일 수 있는 구간은 1층에서 10층이며 Elevator_B가 움직일 수 있는 구간은 10

층에서 20층이다. 사용자가 사용하는 엘리베이터의 범위를 벗어나는 층으로 이동하고자 하는 경우 엘리베이터 사용자는 환승이 가능한 층(10층)에서 이용하는 엘리베이터를 변경해야한다.

해당 알고리즘의 우선순위는 분할된 각 층에 대해서 엘리베이터의 진행 방향과 사용자의 목적지의 방향이 같아야 우선순위가 높다. 또한, 엘리베이터의 방향과 사용자 목적지 방향이 같을 때 사용자의 호출 시간이 빠르면 우선순위가 높다.

표 4. Zoning Algorithm에서 엘리베이터 간의 움직임
Table 4. Movement between elevators in the Zoning Algorithm.

	Elevator_A	Elevator_B
Start Floor	1 st floor	10 th floor
The range of the Floor	1 st floor - 10 th floor	10 th floor - 20 th floor
Transfer Floor	10 th floor	

표 5. Zoning Algorithm
Table 5. Zoning Algorithm.

Zoning Algorithm	
Elevator starting point A = 1	
Elevator starting point B = 10	
A elevator zone = 1 <= A <= 10	
B elevator zone = 10 <= B <= 20	
If (people i starting point = 15 end point= 3){	
first i ride B elevator and 10 floor i	
get off B elevator	
second i ride A elevator go to end point	
}	

3.3 3-Passage Algorithm

3-Passage 알고리즘[18]의 스케줄링 방식은 사용자가 엘리베이터를 호출하면 비용을 계산하여 탑승할 엘리베이터를 할당한다. 비용 계산은 다음과 같다. 첫째 엘리베이터에 탑승한 사람들과 현재 엘리베이터 높이를 곱한다. 다음으로 엘리베이터의 열고 닫히는 시간을 더한 후 계산된 값과 앞서 구한 엘리베이터에 탑승한 사람들과 현재 엘리베이터 높이의 곱의 결과 값을 더한다. 최종적으로 나온 결과 값에 순차적으로 호

출된 수를 곱하여 비용을 계산한다. 3-Passage 알고리즘에서는 계산 비용이 낮을수록 우선순위가 높다.

표 6. 3-Passage Algorithm
Table 6. 3-Passage Algorithm

3-Passage Algorithm
Elevator state = going_up going_down
Layer[] = floor waiting users
Floor cost = cost open_door + cost close_door + cost guest * layer.length
Destination. Set = Floor cost lowest layer
State set = (destination - current layer) >
0 ? going_up : going_down

3.4 Hybrid HRN Algorithm

본 논문을 통해 새롭게 제안하는 Hybrid HRN 알고리즘은 CPU 스케줄링 방식인 HRN 알고리즘에서 새로이 착안하여 엘리베이터 환경에서 사용할 수 있도록 변경한 알고리즘이다. HRN 알고리즘을 사용하기 위해서는 표8와 같이 필수적으로 서비스 실행 시간을 파악해야 한다. 하지만 엘리베이터의 특성상 사용자의 호출이 언제 들어오는지 예상하기가 어려운 문제점이 있다. 이러한 문제를 해결하고자 새롭게 제안하는 Hybrid HRN 알고리즘에서 서비스 실행 시간을 표9와 같이 사용자의 도착 위치와 사용자의 출발 위치 거리 차이로 정의하였다. 또 한, 대기시간은 사용자가 엘리베이터 탑승을 하기 위하여 호출한 후 엘리베이터 탑승 전까지의 기다린 시간 차이를 의미한다. 따라서 Hybrid HRN 알고리즘에서 우선 순위를 구하는 식은 '(대기시간 + 서비스 실행 시간) / 서비스 실행 시간'이다. 기존의 Collective Control 알고리즘 경우 엘리베이터 움직임 방향과 사용자의 움직임 방향과 도착지가 같다면 엘리베이터 위치 기준 더 먼 거리에 있는 사용자를 탑승시킨다. 하지만 Hybrid HRN 알고리즘의 경우 엘리베이터 현재 위치와 사용자들의 거리의 차이는 고려하지 않고 각 엘리베이터 사용자들의 대기시간이 높은 사용자를 먼저 탑승시킨다.

표 7. 각 알고리즘의 우선순위
Table 7. Priority of each algorithm.

	Priority
Collective Control	The largest difference value between the elevator location and the current user's location.
Zoning	The users at the divided floor with the largest difference value between the elevator location and the current user's location.
3 - Passage	The lowest calculated cost
Hybrid HRN	The value where the user's waiting time is highest

표 8. HRN Algorithm
Table 8. HRN Algorithm

HRN Algorithm
User waiting time = elevator absolute time - user elevator call time
User return time = user drop off time - user boarding time
Priority[user]
For (int i = 0; i < user; i++) {
Priority[i] =(user return time + user waiting time) / user return time
}
For(int l = 0; i < user - 1 ; i++) {
If(highest value < priority[i]){
Highest value = priority[i]
Select = l;
}
}
Destination set = select user's starting point
State set = (destination - current layer) >
0 ? going_up : going_down

표 9. Hybrid HRN Algorithm
Table 9. Hybrid HRN Algorithm

Hybrid HRN Algorithm
User waiting time = elevator absolute time - user elevator call time
User return time = user end floor - user start floor

```

Priority[user]
For (int i = 0; i < user; i++) {
Priority[i] =(user return time + user waiting
time) / user return time)
}
For(int l = 0; l < user - 1 ; i++) {
    If(highest value < priority[i]){
        Highest value = priority[i]
        Select = l;
    }
}
Destination set = select user's starting point
State set = (destination - current layer) >
0 ? going_up : going_down
    
```

4. 수행 결과

본 실험의 수행에 사용된 시스템은 CentOS7 기반의 Visual Studio Code IDE 툴의 사용하였으며 JAVA 언어를 사용하였다. 자세한 실험 환경 사양은 표10과 같다.

한 건물에 엘리베이터는 두 대가 있다고 가정한다. 표12와 같이 엘리베이터의 최고층은 20층이다. 사용자는 100명으로 가정한다. 사용자의 출발지는 Random이다. 도착지는 외부에서 건물로 들어온 경우 출발지가 1층이며 도착지는 Random이다. 반대의 경우 Random한 도착지로부터 출발하며 목적 도착지는 1층이다.

본 실험에서 정의한 엘리베이터의 상태는 정지, 위로 이동, 아래로 이동으로 총 3가지이다. 건물의 층간 거리 2.4m이며 엘리베이터의 속도는 1.2m/sec이다. 따라서 엘리베이터가 한 층을 움직일 때 걸리는 소요 시간은 2초이다. 사용자의 엘리베이터 승하차 시간은 총 1초가 걸리며 한 층에서 엘리베이터를 이용하려는 사용자의 수는 승하차 시간에 영향을 주지 않는다. 즉, 현재 층에서 엘리베이터 승하차하는 사용자가 N명이어도 승하차 시간은 총 1초이다. 엘리베이터의 탑승 가능 인원은 총 15명이며, 탑승 가능 인원을 넘기는 경우 다음 사용자는 탑승 정원을 만족할 때까지 대기

하여야 한다.

측정 기준은 첫째, 엘리베이터의 총 운행 거리 및 운행 시간이며 100명의 엘리베이터 사용자 모두가 엘리베이터를 이용할 때까지의 걸리는 거리 및 시간이다. 둘째, 엘리베이터 사용자의 평균대기 시간 및 대기 시간에 대한 표준 편차이다. 마지막 측정 지표는 두 엘리베이터의 사이 운행 Balance이다. 즉, 두 엘리베이터의 운행 균형을 확인하였다.

표 10. 실험 환경

Table 10. Experiment environment.

Specifications	
OS	Linux CentOS 7
IDE	Visual Studio Code 1.64.0
Program Language	JAVA 1.8.0_211

표 11. 엘리베이터 기본 정보

Table 11. Elevator Basic Information,

Elevator	Information	
State	PAUSE	
	GOING_UP	
	GOING_DOWN	
Speed	1.2m/sec	
User's getting on and off time	1sec	
Boarding capacity	15	
Elevator's starting position	Zoning	1 st floor, 10 th floor
	Others	1 st floor

표 12. 엘리베이터 최적화 실험 가정

Table 12. Elevator Optimization Experiment Assumptions.

Information		
Building	1ea	
Number of floors	20 floors	
User	100	
Starting Point	Random	
Destination	If the starting point is the first floor	Random
	If the starting point is not the first floor	1 st floor

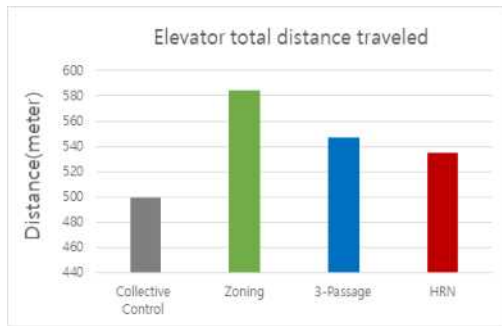


그림 2. 엘리베이터 총 운행 거리
Fig. 2. Elevator total running distance

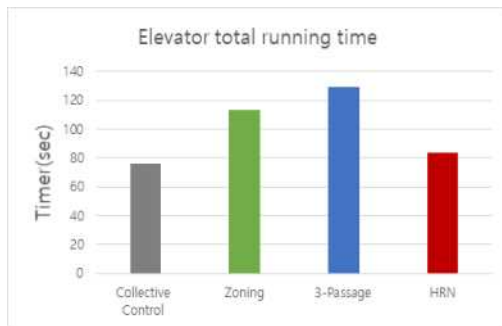


그림 3. 엘리베이터 총 운행 시간
Fig. 3. Elevator total running time

4.1 엘리베이터 총 운행 거리 및 운행 시간

엘리베이터의 운행 거리는 그림2와 같이 Collective Control 알고리즘이 499m로 제일 우수하며 다음으로는 본 논문에서 제안하는 Hybrid HRN 알고리즘이 535.2m로 우수하다. 또 총 운행 시간도 그림3과 같이 Collective Control 알고리즘 76sec이며 Hybrid HRN 알고리즘이 84sec로 가장 준수한 알고리즘인 것을 확인할 수 있다. 하지만, Collective Control 알고리즘은 사용자의 도착지 방향과 엘리베이터의 움직임 방향에 따라 움직인다. 따라서 사용자들의 도착지 방향이 같고 엘리베이터의 도착지 방향이 같으면 하나의 엘리베이터가 사용자에게 대한 탑승을 모두 수행하는 문제점이 있다. 이러한 문제를 극복하고자 추후 언급할 엘리베이터 사이의 운행 Balance를 지표로 측정하였고 본 논문을 통해 제안한 Hybrid

HRN 알고리즘이 Collective Control 알고리즘보다 엘리베이터 사이 사용자 탑승에 대한 우수한 Balance를 가지는 것을 확인할 수 있다. 즉 Collective Control 알고리즘이 제안한 Hybrid HRN 알고리즘보다 각각 평균 총 운행 거리 6% 총 운행 시간이 10% 더 효율적이지만 추후 확인할 두 엘리베이터 사이의 운행 Balance까지 고려하면 Hybrid HRN 알고리즘이 더 효율적인 엘리베이터 스케줄링 방식인 것을 확인할 수 있다.

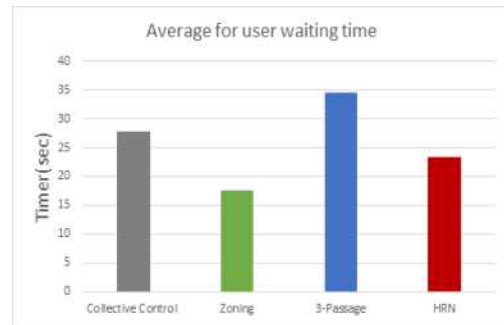


그림 4. 사용자 대기 시간에 대한 평균
Fig. 4. User's average waiting time

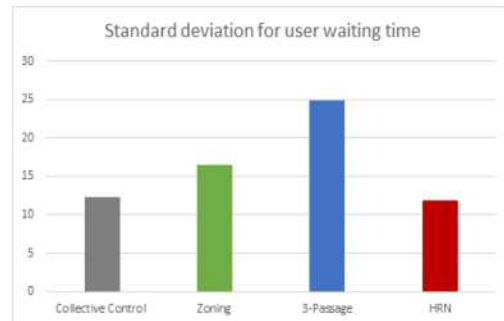


그림 5. 사용자 대기 시간에 대한 표준 편차
Fig. 5. Standard deviation for user's waiting time

4.2 사용자의 평균대기 시간 및 표준 편차

평균대기 시간이란 엘리베이터를 이용하고자 하는 사용자들이 평균 기다린 시간이다. 즉, 사용자가 엘리베이터 탑승을 위해 호출한 시점부터 엘리베이터의 탑승한 시간의 차이이다.

그림4에서 4가지 알고리즘 중 평균대기 시간은 기

존에 제안된 Zoning 알고리즘이 17.5sec로 가장 우수하고 다음으로 제안한 Hybrid HRN 알고리즘이 23.34sec로 효율적인 것을 알 수 있다. 하지만 Zoning 알고리즘은 엘리베이터의 구간을 분할하기 때문에 사용자들의 전 층을 움직이려면 환승을 해야 하는 문제점이 있다. 즉 사용자 평균대기 시간은 Zoning 알고리즘이 제안한 Hybrid HRN 알고리즘보다 평균 14.29% 더 효율적이다. 하지만, 평균 대기시간에 대한 표준 편차는 그림5와 같이 Zoning 알고리즘이 16.44이고 Hybrid HRN 알고리즘이 11.86 인 것을 확인할 수 있다. 즉, Zoning 알고리즘이 평균적인 사용자 대기 시간이 짧지만 사용자들의 개별적인 대기시간이 공정하지 않다. 하지만 제안한 Hybrid HRN 알고리즘은 제시된 다른 알고리즘보다 평균대기 시간도 짧으면서 대기시간에 대한 편차도 가장 작아 사용자들의 평균대기 시간에 대한 공정성을 가진다. 또한, 기존의 Collective Control 알고리즘보다 제안한 Hybrid HRN 알고리즘이 사용자 대기시간에 대한 평균 측면에서 17% 향상된 것을 확인할 수 있다.

4.3 두 엘리베이터 사이의 운행 Balance

두 엘리베이터 사이의 운행 Balance는 엘리베이터 사이의 움직인 거리의 차이이다. 즉, 사용자의 호출이 있는 경우 엘리베이터에 공정하게 사용자가 탑승하는 지 확인할 수 있는 지표이다.

두 엘리베이터 사이의 운행 Balance가 가장 우수

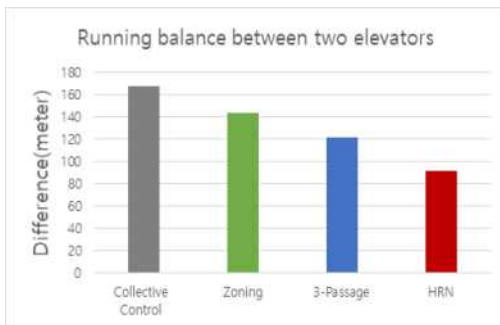


그림 6. 두 엘리베이터 사이의 운행 Balance
Fig. 6. Running balance between two elevators

한 것은 그림6과 같이 본 논문을 통해 제안한 Hybrid

HRN 알고리즘이 92m로 가장 우수하다. 일상생활에 흔히 사용되는 Collective Control 알고리즘은 엘리베이터 총 운행 거리 및 운행 시간에서 가장 우수한 알고리즘이다. 하지만 두 엘리베이터 사이의 운행 Balance 측면에서는 본 논문을 통해 제안한 Hybrid HRN 알고리즘이 Collective Control 알고리즘보다 56.84% 효율적이다. 즉, 앞서 4.1에서도 언급하였듯 총 운행 거리와 운행 시간 그리고 두 엘리베이터 사이의 운행 Balance를 고려하면 제안한 Hybrid HRN 알고리즘이 기존의 Collective Control 알고리즘보다 우수한 움직임을 보이는 것을 확인할 수 있다. 기존의 Zoning 알고리즘은 엘리베이터가 움직일 수 있는 구간을 나눈다. 따라서 사용자들이 1층부터 10층 내 범위에서 호출이 많거나 1층으로 도착지가 몰리는 경우 한 엘리베이터에 사용자가 몰리는 경우가 있다. 즉 이러한 문제점은 두 엘리베이터 사이의 운행 Balance 측면에서 제안된 Hybrid HRN 알고리즘이 기존의 Zoning 알고리즘보다 44% 성능이 향상된 것을 확인할 수 있다. 3-Passage 알고리즘은 계산된 비용을 통해 엘리베이터가 움직이기 때문에 엘리베이터 사이의 운행 Balance가 준수하다. 하지만 그 외의 지표에서 효율성이 떨어지는 것을 확인할 수 있다.

본 논문을 통해 제안된 Hybrid HRN 알고리즘의 우선순위는 사용자들의 평균대기 시간이며 엘리베이터의 현재 위치에서 거리가 더 먼 사용자들의 위치를 고려하지 않는다. 따라서 엘리베이터와 사용자들의 거리와 관계없이 평균대기 시간이 긴 사용자들의 찾아 먼저 움직이기 때문에 제안된 Hybrid HRN 알고리즘이 기존의 엘리베이터 알고리즘들 보다 두 엘리베이터 사이의 운행 Balance가 좋을 것을 확인할 수 있다.

5. 결 론

기존의 Collective Control, Zoning, 3-Passage 알고리즘과 본 논문을 통하여 제안하는 Hybrid HRN의 알고리즘을 사용하여 엘리베이터 최적화 스케줄링을 조사하고 분석하였으며 해당 알고리즘의 장점과 단점을 조사하고 설명하였다.

새롭게 제안한 Hybrid HRN 알고리즘은 두 엘리베이터 사이의 운행 Balance에서는 기존의 Collective

Control 알고리즘보다 56.84% 더 효율적인 것을 확인하였다. 추가적으로 Hybrid HRN 알고리즘이 사용자 평균대기 시간 측면에서 기존의 Collective Control 알고리즘보다 17% 향상된 것을 확인하였고 기존의 알고리즘들보다 Hybrid HRN 알고리즘 표준 편차가 11.86으로 사용자들이 균형있게 대기시간을 가지는 것을 확인하였다. 총 운행 거리 및 운행 시간 측면에서 제안한 Hybrid HRN 알고리즘이 기존의 알고리즘들 보다 향상된 것을 확인할 수 있고 두 엘리베이터 사이의 Balance에서도 기존의 알고리즘들 보다 44% 향상된 것을 확인할 수 있다.

Hybrid HRN 알고리즘은 엘리베이터의 총 운행 거리 및 시간, 사용자의 평균대기 시간 및 표준 편차, 두 엘리베이터의 운행 Balance를 종합적으로 고려하였을 때 가장 효율적인 움직임을 나타내는 것을 본 논문을 통해 확인하였다.

추후 제안한 스케줄링 알고리즘을 활용한 다양한 연구수행이 기대되며 또한 최적화 및 성능향상을 위한 C-SCAN, 퍼지 이론 등을 적용한 추가적인 연구가 필요할 것으로 예상된다.

REFERENCES

- Korea Law Information Center [online] <https://www.law.go.kr/행정규칙/승강기안전운영및관리에관한운영규정>
- J. Kim, B. Moon, "Elevator Group Control using Genetic Algorithms and Digital Cameras", The Korean Institute of Information Scientists and Engineers, vol. 26, No. 1B, pp.233-235, 1999
- D. Song, D. Jeong, "Design of a Voice-Recongized Elevator Control System", Proceedings of KIIT Conference, pp. 240-243, 2010
- H. Bahn, "Efficient Scheduling of Sensor-based Elevator Systems in Smart Buildings", Journal of the Korea Academia-Industrial cooperation Society, vol. 17, pp. 367-372, 2016
- W. F.Dowling, J. H.Gallier, "Linear-time algorithms for testing the satisfiability of propositional horn formulae", The Journal of Logic Programming, vol. 1, pp. 267-284, 1984
- A. Moallemi, M. Asgharilarimi, "A Fuzzy Scheduling Algorithm Based on Highest Response Ratio Next Algorithm", Innovations and Advanced Techniques in Systems, Springer Science+Business Media, pp. 75-80, 2008
- D. Kmiec, "Private Control of Collective Property Rights: Robert H. Nelson, Zoning and Property Rights", [online] <https://scholar.valpo.edu/vulr/vol13/iss3/6>
- C. Park, J. Kim, J. Im, Y. Woo, I. Lee, "Intelligent Elevator System Using Fuzzy Inference Method", Proceedings of the Korean Institute of Electrical Engineers, pp.317-320, 2011
- S. Janne, H. Henri, S. Marja-Liisa, "ELEVATOR SELECTION WITH DESTINATION CONTROL SYSTEM", [online] <https://www.researchgate.net/publication/31597205>
- O. Kwon, H. Bahn, K. Koh, "A Context-Aware Elevator Scheduling System for Smart Apartment Buildings", ICHIT, pp. 362-372, 2007
- X. Gu, Z. Cao, A. Jolfaei, P. Xu, D. Wu, T. Jung, C. Lin, "EEG-Based Brain-Computer Interfaces (BCIs): A Survey of Recent Studies on Signal Sensing Technologies and Computational Intelligence Approaches and Their Applications", IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, pp.1645-1666, 2021
- Q. Wu, Z. Han, Q.E. Wu, "Application of Fuzzy Automata Decision-Making System in Target Control", Journal of Computer and Communications, vol. 5, pp.16-25, 2017
- D. Choi, J. Park, H. Park, K. Woo, "A predictive fuzzy control algorithm for elevator group control" Institute of Control Robotics and Systems, pp.731-736, 1993
- H. Kim, "A Design for Elevator Group Controller of Building Using Adaptive Dual Fuzzy Algorithm", KSME international journal, vol. 15, pp.1664-1675, 2001
- C. Kim, K. Seong, H. Kwang, J. Kim, Y. Lim, "A fuzzy approach to elevator group control system", IEEE TRANSACTIONS ON SYSTEM, vol.

25, pp.985-990, 1995
 A. Kumar, "Dynamically generating problems on static scope", ACM SIGCSE Bulletin, vol. 32, pp.09-12, 2000
 Z. Li, Z. Mao, J. Wu, "Research on dynamic zoning of elevator traffic based on artificial immune algorithm", IEEE, pp. 2170-2175, 2005
 X. Liu, Y. Li, "Simulation and analysis of pedestrians in the station of urbanrail transit", IOP Publishing, pp.01-06, 2020

정 성 옥 (Sung-Wook Chung)

[종신회원]



- 2005년 5월 : CISE dept. Univ. of Florida, USA, (MS)
- 2010년 8월 : CISE dept. Univ. of Florida, USA, (Ph.D)
- 2010년 10월 ~ 2012년 2월 : KT 종합기술원 중앙연구소 선임 연구원
- 2012년 3월 ~ 현재 : 창원대학교 컴퓨터공학과 부교수

〈관심분야〉

IoT, 스마트모빌리티, 실시간 분산 멀티미디어시스템, 홈네트워크

저자약력

백 진 우 (Jin-Woo Baek)

[학생회원]



- 2016년 3월 ~ 현재: 창원대학교 학부생

〈관심분야〉

IoT, 실시간 분산 멀티미디어시스템

염 기 훈 (Gi-Hun Yeom)

[학생회원]



- 2016년 3월 ~ 현재: 창원대학교 학부생

〈관심분야〉

IoT, 실시간 분산 멀티미디어시스템