Research in Transportation Economics 92 (2022) 101095

Contents lists available at ScienceDirect

Research in Transportation Economics

journal homepage: www.elsevier.com/locate/retrec

A public transport network design using a hidden Markov model and an optimization algorithm

Yun Zhang [a], Weichu Xue [b], Wei Wei [a,*], Habibeh Nazif [c]

[a] School of Finance, Shanghai Lixin University of Accounting and Finance, Shanghai, 201209, China
[b] Department of Architecture, Korea University, Seoul, 02841, South Korea
[c] Department of Mathematics, Payame Noor University, Iran

ARTICLE INFO

Keywords:
Transport
Network design
Hidden markov model
Equilibrium optimizer
Algorithm

ABSTRACT

Transportation Network Design Problem (TNDP) includes making the right choices possible when deciding a collection of design criteria to develop a current transportation network in response to rising traffic demand. Traffic congestion, higher maintenance and fuel prices, delays, accidents, and air emissions stem from the general rise in flow volume. Because of the NP-hard nature of this problem, a hidden Markov model and an Equilibrium Optimizer (EO) are employed in this paper to solve it. Each particle (solution) behaves as a search agent in EO, with its position. To reach the equilibrium condition, the search agents change their focus at random regarding the best-so-far approaches, including equilibrium candidates. A well-defined "generation rate" concept has been shown to elevate EO's capacity in avoiding local minima. This article provides a new method to lower the feasible travel time and the public travel cost using the hidden Markov model and EO algorithm. The suggested method's performance was compared to the performance of other algorithms on a test network. The related numerical outcomes show that it is more effective.

1. Introduction

At present, the demand for public transport has increased rapidly around the world (Sumpavakup, Suwannakijborihan, Ratniyomchai, & Kulworawanichpong, 2018). Works are superimposed on physical networks of highways, parking systems, and trains in public transportation networks (Liu, Chen, Liu, Jermsittiparsert, & Ghadimi, 2020). Public transportation networks are relatively complex schemes due to the significance of transfers, multimodality, transit hubs, and intermediate walking ties; however, in contrast to the personal car option, connectivity is usually weak, and service quality varies over time (Kim, Jeong, Kim, & Park, 2020). Besides, workers (e.g., drivers), electric power (Hosseini Firouz & Ghadimi, 2016), telecom, vehicles, and other factors all play a role in the performance of operations. Many of these aspects add up to public transportation being less mobile and potentially less robust than driving one's vehicle. Besides, public transportation networks' multimodality can allow alternate modes to have backup power in the event of failures to decrease the whole risk (Bagal et al., 2018; Basnak et al., 2020). In the event of a disturbance, there are also more resources for organized reconstruction and mitigation activities (Cats and Jenelius, 2015). In today's public transit setting, the ever-increasing number of electric vehicles poses a threat (Lu et al., 2020). There are *2n* potential networks for a road network with n junctions. As a consequence, discovering an optimum path is a vital criterion of traffic optimization. However, in many situations, there is a restriction or unavailability of a road intersection. It is often likely that a specific link that seems to be shorter is inaccessible or heavily disputed at a certain time. Another profitable way of dealing with it is to reduce the time spent waiting at traffic signals. It saves drivers valuable time and alleviates traffic, increases public safety, and facilitates the flow of health demands (Srivastava & Sahana, 2019). Congestion has been a major impediment to economic prosperity as well as a hazard to life quality. In several cities, traffic congestion reduction proposals have been met with cynicism. The Network Design Problem (NDP) entails making the best decision to improve transportation infrastructure in response to rising travel needs while taking network users' route preference behavior into account (Xu, Yang, & Gao, 2010, pp. 513–517).

In the process of real-time regulation, multiple-objective optimization (regular interval time, transit time, route time, quality of service, etc.) is simultaneously known as the NP-hard multi-objective optimization problem that cannot be solved by traditional methods (Sidi, Hammadi, Hayat, & Borne, 2008). Since this problem aims to achieve

* Corresponding author.
E-mail address: weiwei202104@163.com (W. Wei).

https://doi.org/10.1016/j.retrec.2021.101095
Received 22 February 2020; Received in revised form 16 April 2021; Accepted 13 May 2021
Available online 31 May 2021

the minimum cost (total commute time) and the maximum of feasible public travels per unit length of the NP-hard problems, it requires a nonlinear algorithm. In the present article, the public transport network design is proposed using the Equilibrium Optimizer (EO) and hidden Markov model. This algorithm makes use of an exploratory search mechanism and the Gaussian mutation based on the reconstruction and division of the population. The population is developed in each iteration of the suggested algorithm utilizing these structures and the EO's regular search protocol. These methods aim to preserve various solutions throughout the search to prevent stagnation against suboptimal solutions and increase the convergence rate to achieve more precise optimum solutions (Gupta, Deep, & Mirjalili, 2020).

This article also uses the Markov model. A Markov model is a statistic that assumes the modeled structure is a Markov procedure with hidden (unseen) states (Zhu et al., 2020). The most straightforward dynamic Bayes' network is the hidden Markov model (Heidari Gharehbolagh et al., 2017; Sidi, Hammadi, Hayat, & Borne, 2008). The objectives of this paper are as follows:

- Reducing cost (total displacement time) in public transport networks using EO algorithm and hidden Markov model
- Increasing feasible travels in the public transport networks using the EO algorithm and hidden Markov model

The residual sections of the article are as follows. In Section 2, some of the important approaches to public transportation networks are discussed. Section 3 prepares a complete overview of the proposed method and details of implementing the suggested method with the outcomes. Experiments, simulations, data sets, and graphs are illustrated in Section 4. Eventually, Section 5 summarizes the results and proposals for upcoming work.

2. Related work

Sakharov, Chernyi, Saburov, & Chertkov, 2021 addressed the tactical issue of deciding the shortest paths for a group of vessels with meeting a hypothetical goal in specific coordinates and a constrained area. In this paper, the algorithm's practical application was demonstrated by estimating a network model utilizing an iterative technique employing MATLAB codes. The findings indicated that using computer technologies applied in MATLAB, they were able to remove the constraints related to the existence of negative network cycles and weights. They also could automate the estimation of the shortest node routes.

Manser, Becker, Hörl, & Axhausen, 2020 improved an agent-based simulation platform to compass large-scale public transit networks. It exceeds the prior methods by including a complex request-response to network shifts and exogenous factors. The system was evaluated in Zurich city using an agent-based (MATSim) simulation. The findings demonstrated that it recommends a sparser network of fewer cars and slightly higher speeds than the current public transit infrastructure. As a result, the strategy expects greater transit ridership at lower subsidy levels. Furthermore, it regularly recognizes corridors that could benefit from capacity improvements.

As a classifier, Kuznetsov et al. (2020, pp. 1–4) regarded the approach of automating the learning mechanism of a neural network. This approach is built on using newly designed programs and algorithms that automatically produce training instances. They employed a complete factorial experiment using this technique. The classifier is a software framework that contains a multilayer perceptron, a series of programs, and a database for automated perceptron learning. The findings revealed that using a neural network as a classifier would substantially decrease the time spent creating a constructive base for complex transportation systems.

In today's urban transportation situation, with the ever-increasing number of cars, it is critical to address network issues. Srivastava & Sahana, 2019 used Bat Algorithms (BA) to address the transportation network design issue. This study aims to find the best waiting time at traffic lights using a discrete microscopic model. The findings are compared to Genetic Algorithms (GA) and Ant Colony Optimization (ACO). They revealed that BA outperforms all the described strategies.

Also, Jiang et al. (2017) investigated the design issue of the Highway Passenger Transport-Based Express Parcel Service (HPTB-EPS) network. They also analyzed the time-space features related to this kind of network. The service decision, network flow propagation, and frequency are integrated into a mixed-integer programming model. A heuristic algorithm has been presented to address the model by decomposing the issue into three stages: service network creation, service route choosing, and network flow propagation. The effectiveness of our algorithm and model is demonstrated by a numerical trial utilizing actual data from our partner firm. The findings indicated that their solution would save up to 16.3 percent on total costs. Besides, the sensitivity study shows the model's robustness and stability.

Xu, Yang, & Gao, 2010 suggested the Particle Swarm Optimization (PSO) algorithm to tackle transportation Continuous Network Design Problems (CNDPs). They also recommended a sensitivity examination for the PSO parameters. The CNDP has been designed as a model that is bi-level programming. One-At-a-Time Designs (OATD) is utilized to examine the influence of the parameters. The outcomes indicated that when the parameters are set correctly, PSO is an efficient algorithm to tackle CNDP.

3. Suggested method

This section described public transport network definition, Markov clustering, equilibrium optimizer and fitness function.

3.1. Public transport network definition

A directed graph $G(S,E)$ can define the physical public transport networks. The node set S illustrates rail stations and stops. The link set $E \subseteq S \times S$ indicates direct associations among stops.

One or numerous public transport lines may execute each link $e \in E$. Subsequent stops $l = (s_{l,1}, s_{l,2}, \ldots, s_{l,|l|})$ $^{\text{define}}$ a line l ($s_{l,1}$ = origin terminal; $s_{l,|l|}$ = destination terminal). The collection of the whole lines has been denoted by L. $e \in l$ states that link e has been located in l, i.e., $e = (s_{l,i}, s_{l,i+1})$ for some i. So, every e link has been related to a collection of lines $L_e = \{l \in L | e \in l\}$.

Every link e has a riding time related to it that appears when the upstream stops and arrives at the downstream stop. Considering the present traffic circumstances, the riding time can differ among days and trips. Likewise, each stop has a dwell period, which is the amount of time it takes for a machine to come to a full stop for alighting and boarding. The dwell times, such as riding times, may differ among days and trips based on the vehicle type, existing travelers, and other factors, therefore called stochastically.

Every line l has a series of vehicle trips that run along with a schedule. A trip going through line l departing from the source terminal is totally a function related to a planned departure and the arrival time associated with the prior trip, determined by the train's actual riding and dwell times. Travel demand has been associated with the network via a subset of Origin-Destination (OD) nodes $S_{OD} \subseteq S$. The collection of travelers from $o \in S_{OD}$ to $d \in S_{OD}$ during the (t, $t+\tau$) is signified $N_{od}(t,\tau)$. The need for public transit is believed to be inelastic in this situation, meaning that it is unaffected by shifts in commuting times or other causes. On the other hand, travel demand amounts may differ throughout the day, and the number of passengers over this period may be stochastic to indicate day-to-day fluctuations.

The physical passenger path is defined by subsequent stops starting from the origin to the destination like the vehicle lines, i.e., $j = (s_{j,1}, s_{j,2}, \ldots, s_{j,|j|})$ where $s_{j,1}, s_{j,|j|} \in S_{OD}$. In general, a given traveler's physical route on a special day and day time would be determined by the features of the various public transportation lines and the circumstances that day, based on the individual's preferences. The possibility that traveler n utilizes path ^{j} is illustrated by $^{pn\ (j)}$, taking the system stochasticity and dynamics into account (Cats and Jenelius, 2015).

3.2. Markov clustering

Clustering is the procedure of classifying a set of objects into clusters in which the inner members of each cluster are most similar to each other and the least similar to members of other clusters (Azhir et al., 2021a; Sadrishojaei et al., 2021). In general, there may be different ways to specify the clustering of two neighboring nodes, but most do so locally (Azhir et al., 2021b, Zanbouri & Jafari Navimipour, 2020). Markov clustering is a simple and general solution to identify and differentiate clusters from each other (Sefati and Navimipour, 2021). This method is based on differentiating operators that are repeatedly applied to the graph and leads to identifying clusters. This action causes the edge values inside the clusters to increase and the edge value among the clusters to decrease, and as a result, leading to identifying clusters. These operators are simpler, faster, and more natural to cluster and are more useful in many applications [11].

Differentiation is based on the similarity of the neighborhood among node i with other nodes. In fact, the value (weight) is the edge from i to its neighbors. Hence, the neighborhood of i with $P^{i}{}_{visit}(i)$ is illustrated. In most cases, the small value of k is used to estimate the value of $P^{k}{}_{visit}(v)$ and $P^{k}{}_{visit}(u)$ (similarity of two nodes u and v). The smaller the similarity difference between u and v, the greater the weight of the relationship.

3.3. Equilibrium optimizer

The Equilibrium Optimizer's (EO) mathematical model, inspiration, and algorithm are presented in this section (Faramarzi, Heidarinejad, Stephens, & Mirjalili, 2020).

3.3.1. Inspiration

A basic dynamic mass balance is the source of inspiration for the EO method. In this balance, an equation related to the mass balance is utilized to characterize the focus of a nonreactive constituent on a control volume due to the different relevant sink and source strategies. A differential equation expresses the related equation:

$$V\frac{dC}{dt} = QC_{eq} - QC + G \tag{1}$$

C_{eq} indicates the concentration at an equilibrium state where no generation exists in the control volume, C demonstrates the concentration into the control volume (V), Q is the rate of the volumetric flow into, $V\frac{dC}{dt}$ is the mass shift rate, and G is the mass generation rate. If $V\frac{dC}{dt}$ is zero, a steady state has been attained. A rearrangement of Eq. (1) leads to tackling $\frac{dC}{dt}$ as a function of $\frac{Q}{V}$; where $\frac{Q}{V}$ indicates the residence time inverse (λ) or the rate of the turnover $\left(i.e.,\ \lambda = \frac{Q}{V}\right)$. Afterward, in the control volume *(C)*, we can rearrange Eq (1) in order to tackle the concentration as a time function (t):

$$\frac{dc}{\lambda C_{eq} - \lambda c + \frac{G}{V}} = dt \tag{2}$$

Eq. (3) illustrates Eq. (2) integration over time:

$$\int_{c_0}^{c} \frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = \int_{t_0}^{t} dt \tag{3}$$

It leads to the equation below:

$$C = C_{eq} + \left(C_0 - C_{eq}\right)F + \frac{G}{\lambda V}(1 - F) \tag{4}$$

F is calculated as below using Eq. (4):

$$F = exp[-\lambda(t - t_0)] \tag{5}$$

Where, based on the integration interval, C_0 and t_0 are the initial concentration and start-time. Eq. (4) may be utilized in measuring the overall turnover rate by employing a linear regression. Also, in the control volume, it can be used in approximating the concentration utilizing a given rate of the turnover.

The stated equations serve as the overarching basis for designing EO in this section. A particle in EO is equivalent to a solution, and a concentration in the PSO algorithm is equivalent to a particle's location. As seen in Eq. (4), a particle's updating principles and concentration are described by three terms. The equilibrium concentration is described as a crucial solution chosen at random from a pool (equilibrium pool). A concentration differential between a particle and the equilibrium state is the second term that serves as a straightforward search process. This concept allows particles to serve as explorers by exploring the domain worldwide. The 3rd term is the generation rate that primarily serves as a solution refiner or exploiter, especially with small steps, albeit it may also serve as an explorer. The following sections describe each term and how it influences the search pattern.

3.3.2. Initialization and function evaluation

EO, like the majority of meta-heuristic algorithms, begins the optimization mechanism with the initial population. The beginning concentrations are determined depending on the measurements and particles in the search space, with uniform random initialization:

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) \quad i = 1, 2, \ldots, n \tag{6}$$

$C_i^{initial}$ is the concentration vector of the *i-th* particle, C_{min} and C_{max} are the minimum and maximum values for the dimensions, Rand_i is a random vector in the interval of [0,1], and n is the number of particles in each population. The fitness function of particles is assessed, and then the candidates for equilibrium are categorized.

3.3.3. Equilibrium pool and candidates (C_{eq})

The algorithm's ultimate convergence state that is intended to be the universal optimum is the equilibrium state. Nothing is evident about the equilibrium state at the outset of the optimization procedure, so just the candidates are selected to have a search pattern. They are the 4 best-so-far particles found throughout the entire optimization procedure—these four applicants aid EO's exploration capabilities. The number of candidates is determined at random and is dependent on the form of optimization issue. They may be used as long as they are compatible with the literature [6]. GWO, for instance, changes the positions of the other wolves (alpha, beta, and gamma wolves) using three best-so-far candidates. Utilizing less than four candidates, on the other hand, degrades the method's output in multi-modal and composition functions while enhancing outcomes in unimodal functions. It will have the reverse result if there are more than four contestants. These particles are designated as candidates, and they are employed to build the pool:

$$\vec{C}_{eq.pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\} \tag{7}$$

Each particle updates its concentration in each iteration by choosing candidates at random from a pool of candidates of the same likelihood. For example, the first particle updates the whole of its concentrations in the first iteration based on $\vec{C}_{eq\ (1)}$; afterward, it may update the concentrations in the second iteration based on $\vec{C}_{eq\ (ave)}$. When the optimization procedure is finished, each particle will go through the updating phase.

3.3.4. Exponential term (F)

F helps to update the key concentration rule. The precise meaning of this concept would aid EO in striking a fair balance between exploitation and exploration. Because the turnover rate in a real control volume can differ over time, λ is believed to be a random vector in the range [0,1].

$$\overrightarrow{F} = e^{-\overrightarrow{\lambda}(t-t_0)} \tag{8}$$

t is obtained via Eq. (9):

$$t = \left(1 - \frac{Iter}{Max\_iter}\right)^{\left(a_2 \frac{Iter}{Max\_iter}\right)} \tag{9}$$

Where *Iter* and *Max_iter* illustrate the present and the highest number of iterations. This research addresses the following factors to ensure convergence and optimizing the algorithm's exploration and exploitation ability:

$$\overrightarrow{t}_0 = \frac{1}{\overrightarrow{\lambda}} In(-a_1 sign(\overrightarrow{r} - 0.5)[1 - e^{-\overrightarrow{\lambda} t}]) + t \tag{10}$$

Where a_1 manages exploration capability. The higher the a_1 , the better the exploration ability. Likewise, the higher the a_2, the better the exploitation ability. $sign(\overrightarrow{r} - 0.5)$ impacts the exploitation and exploration direction. r is a random vector. For the whole issues subsequently tackled in the present article, a_1 and a_2 are equal to 2 and 1. These constants are chosen after a subset of test functions is empirically tested. These parameters, on the other hand, can be tweaked as required for different issues.

Eq. (11) illustrates the revised version of Eq. (8) using Eq. (10):

$$\overrightarrow{F} = a_1 sign(\overrightarrow{r} - 0.5)[1 - e^{-\overrightarrow{\lambda} t}] \tag{11}$$

3.3.5. Generation rate (G)

One of the most critical words in the suggested algorithm to supply accurate solutions by optimizing the exploitation process is the generation rate. Several models can describe the generation rate as a function of time in numerous engineering applications (Guo, 2002). One multipurpose model mechanism is:

$$\overrightarrow{G} = \overrightarrow{G}_0 e^{-\overrightarrow{k}(t-t_0)} \tag{12}$$

Where G_0 is the initial value and k illustrates a decay constant. The current investigation considers $k=\lambda$ and utilizes the priorly extracted exponential term to have a more managed search pattern and restrict random variables. As a result, here is the final series of generation rate equations:

$$\overrightarrow{G} = \overrightarrow{G}_0 e^{-\overrightarrow{k}(t-t_0)} = \overrightarrow{G}_0 \overrightarrow{F} \tag{13}$$

Where:

$$\overrightarrow{G_0} = \overrightarrow{GCP}\left(\overrightarrow{C_{eq}} - \overrightarrow{\lambda}\overrightarrow{C}\right) \tag{14}$$

$$\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{15}$$

Where r_1 and r_2 are random numbers in [0,1], and GCP vector is built using Eq. (15). Another term called Generation Probability (GP) determines the likelihood of this contribution, which describes how many particles utilize the generation term to change their states. Eq. (15) is utilized at each particle level. If GCP is zero, for instance, G is zero, and the dimensions of that particular particle are modified without a generation rate term. A good balance between exploitation and exploration is attained with $GP = 0.5$. Eventually, the following will be the EO updating rule:

$$\overrightarrow{C} = \overrightarrow{C}_{eq} + \left(\overrightarrow{C} - \overrightarrow{C}_{eq}\right).\overrightarrow{F} + \frac{\overrightarrow{G}}{\overrightarrow{\lambda} V}\left(1 - \overrightarrow{F}\right) \tag{16}$$

Where F is defined in Eq. (11), and V is regarded as the unit.

The 1st term in Eq. (16) is an equilibrium concentration, in which the 2nd and 3rd terms demonstrate the changes in concentration. The 2nd term is in charge of scanning the whole room for an optimal point. This concept refers further to exploration, allowing vast differences in focus to be exploited. The third term adds to the accuracy of the solution by finding a point. As a result, this term refers more to exploitation and gains from minor concentration fluctuations regulated by the generation rate term. The 2nd and 3rd terms can have the same or opposite signs based on parameters like particle concentrations and equilibrium candidates and the turnover rate (λ). The same sign increases the variation, which aids in searching the entire field, and the opposite sign decreases the variation, which aids in local searches.

While the 2nd term seeks solutions that are comparatively far from candidates, and the 3rd term seeks to optimize the solutions near to the candidates, this does not necessarily occur. Small turnover ratios in the denominator of the 3rd term (e.g.,0.05) enhance variety and aid discovery in certain dimensions. Fig. 1 demonstrates a 1-D version of how the terms contribute to exploitation and exploration. $C_1 - C_{eq}$ is representative of 3$^{\text{nd}}$ term in Eq. (16) while $C_{eq} - \lambda C_1$ represents the third term. The generation rate terms (Eqs. (13)–(15)) manage these differences. Since differences in each dimension produce large variations, this large variance only occurs in dimensions with small λ values. This function, which works similarly to a mutation operator, aids EO in exploiting the solutions.

In the suggested algorithm, Fig. 2 depicts the cooperation of the whole candidates on a sample particle. It also indicates the way they impact the updating of the concentration one by one. Because the topological locations of candidates vary in beginning iterations, the particles will cover the whole field in their search using this step-by-step updating method. In the final iterations, the candidates circle the optimal point with identical setups, resulting in the opposite situation. The exponential term produces small random amounts at these moments by having smaller phase sizes.

3.3.6. Particle's memory saving

Using memory-saving techniques makes it easier for each particle to follow its coordinates in space, affecting its fitness value. This procedure illustrates the *pbest* in PSO. Each particle's fitness value in the present iteration is compared to that of the prior iteration, and if it attains an improved fit, it is overwritten. This process helps extraction capacity, but if the system does not profit from global exploration ability, it may raise the risk of being stuck in local minima (Parouha & Das, 2016). Fig. 3 shows the suggested EO algorithm's pseudo-code and memory saving feature.

3.3.7. Exploration ability of EO

The following are some of the parameters and pathways in EO that contribute to exploration:

- a_1: it manages the exploration quantity of the algorithm and establishes the distance between the current position and the candidate. The higher it, the higher the exploration ability. Since a_1 can enlarge the concentration difference, it must be high enough to boost the exploration ability. Nevertheless, experimental research has shown that values larger than three encourage agents to look for boundaries. This suggestion is close to other algorithms' recommendations for free parameters. In PSO, for instance, the number of the social and cognitive parameters must be below or equal to 4 (Beielstein, Parsopoulos, & Vrahatis, 2002).
- *sign* $(r - 0.5)$: it controls the exploration direction.
- *Generation probability (GP):* it manages the possibility of concentration updating involvement dependent on the generation scale. $GP = 1$ indicates that there won't be a generation rate term participating in the optimization procedure. It highlights great exploration capability

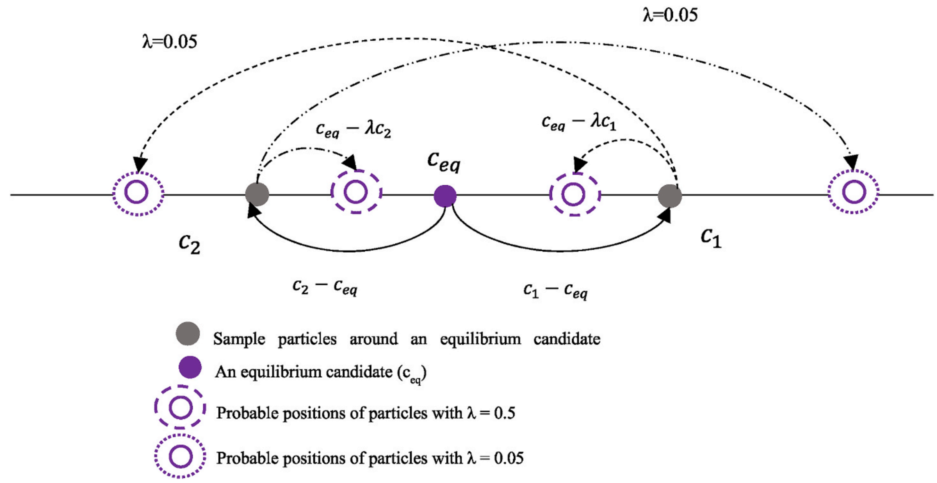

Fig. 1. 1-D presentation of concentrations updating in exploration and exploitation (Faramarzi, Heidarinejad, Stephens, & Mirjalili, 2020).

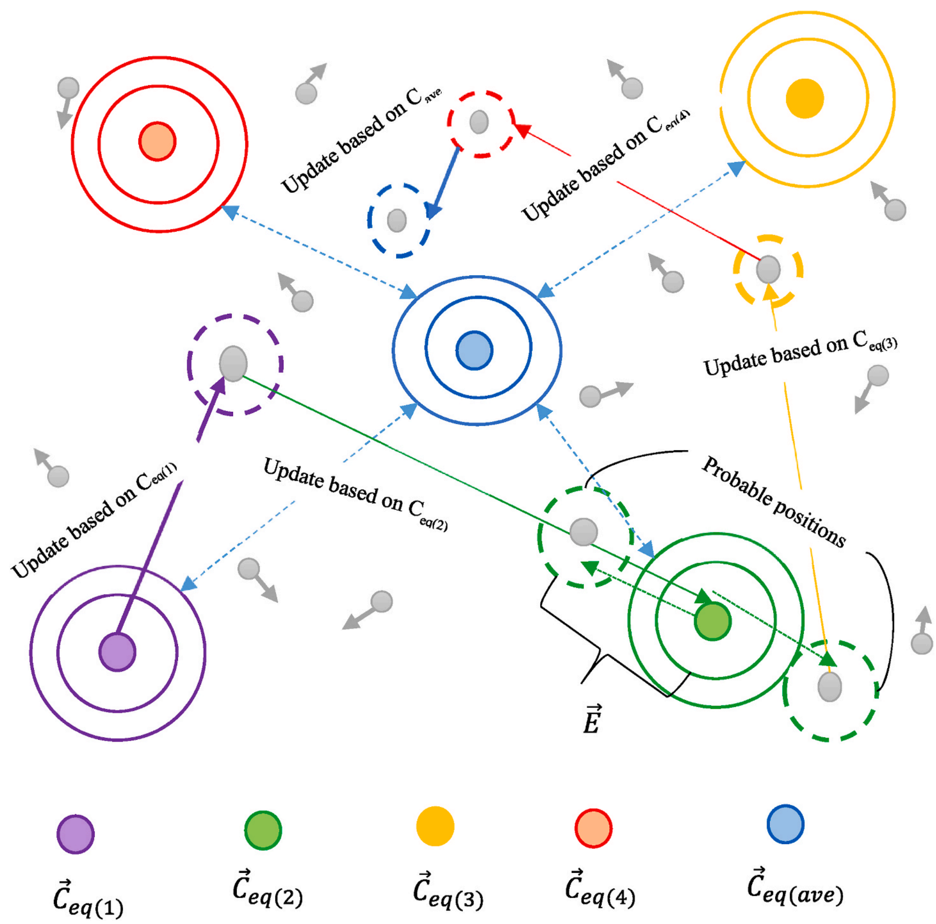

Fig. 2. Candidates' collaboration in updating a particles' concentration (Faramarzi, Heidarinejad, Stephens, & Mirjalili, 2020).

and frequently causes the creation of non-accurate methods. $GP = 0$ states that the generation rate term will participate in the procedure, elevating the local optima's stagnation possibility. $GP = 0.5$ supplies a good balance between exploitation and exploration phases based on experimental testing.

- *Equilibrium pool*: There are five particles in this vector. The choice of five particles is kind of random, but it was made after extensive research. In the first iterations, the candidates are all physically isolated from one another. The algorithm's capability to scan the space globally is improved by updating the concentrations dependent on these candidates. Since particles are further apart at the first iteration, the typical particle often assists in exploring hidden search spaces.

3.3.8. Exploitation ability of EO

The following are the key criteria and frameworks for executing exploitation and local search in EO:

Initialize the particles populations i=1, ..., n
Assign equilibrium candidates fitness a large number
Assign free parameters a1=2; a2=1; GP=0.5;
While Iter < Max_iter
  For i=1: number of particles (n)
  Calculate fitness of i^th particle
    If $fit\left(\vec{C}_i\right) < fit\left(\vec{C}_{eq1}\right)$
      Replace $\vec{C}_{eq1}$ *with* $\vec{C}_i$ *and* $fit\left(\vec{C}_{eq1}\right)$ *with* $fit\left(\vec{C}_i\right)$
    Elseif $fit\left(\vec{C}_i\right) > fit\left(\vec{C}_{eq1}\right)$ & $fit\left(\vec{C}_i\right) < fit\left(\vec{C}_{eq2}\right)$
      Replace $\vec{C}_{eq2}$ *with* $\vec{C}_i$ *and* $fit\left(\vec{C}_{eq2}\right)$ *with* $fit\left(\vec{C}_i\right)$
    Elseif $fit\left(\vec{C}_i\right) > fit\left(\vec{C}_{eq1}\right)$ & $fit\left(\vec{C}_i\right) > fit\left(\vec{C}_{eq2}\right)$ & $fit\left(\vec{C}_i\right) < fit\left(\vec{C}_{eq3}\right)$
      Replace $\vec{C}_{eq3}$ *with* $\vec{C}_i$ *and* $fit\left(\vec{c}_{eq3}\right)$ *with* $fit\left(\vec{c}_i\right)$
    Elseif $fit\left(\vec{c}_i\right) > fit\left(\vec{c}_{eq1}\right)$ & $fit\left(\vec{c}_i\right) > fit\left(\vec{c}_{eq2}\right)$ & $fit\left(\vec{c}_i\right) > fit\left(\vec{c}_{eq3}\right)$ & $fit\left(\vec{c}_i\right) < fit\left(\vec{C}_{eq4}\right)$
      Replace $\vec{C}_{eq4}$ *with* $\vec{C}_i$ *and* $fit\left(\vec{C}_{eq4}\right)$ *with* $fit\left(\vec{C}_i\right)$
    End (If)
  End (For)
$\vec{C}_{ave} = (\vec{C}_{eq1} + \vec{C}_{eq2} + \vec{C}_{eq3} + \vec{C}_{eq4})/4$
Construct the equilibrium pool $\vec{C}_{eq,pool} = \{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\}$
Accomplish memory saving (if Iter>1)
Assign $t = \left(1 - \frac{Iter}{Max\_iter}\right)^{\left(a_2 \frac{Iter}{Max\_iter}\right)}$ *Eq (9)*
  For i=1: number of particles (n)
    Randomly choose one candidate from the equilibrium pool (vector)
    Generate random vectors of $\vec{\lambda}$, $\vec{r}$ *from Eq. (11)*
    Construct $\vec{F} = a_1 sign\,(\vec{r} - 0.5)\left[e^{-\vec{\lambda}t} - 1\right]$ *Eq.(11)*
    Construct $\overrightarrow{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$ *Eq. (15)*
    Construct $\vec{G}_0 = \overrightarrow{GCP}(\overrightarrow{C_{eq}} - \vec{\lambda}\vec{C})$ *Eq. (14)*
    Construct $\vec{G} = \overrightarrow{G_0}.\vec{F}$ *Eq. (13)*
    Update concentrations $\vec{C} = \vec{C}_{eq} + \left(\vec{C} - \vec{C}_{eq}\right).\vec{F} + \frac{\vec{G}}{\vec{\lambda}V}(1 - \vec{F})$ *Eq. (16)*
  End (For)
  Iter=Iter+1
End while

Fig. 3. Detailed pseudo-code of EO.

- a_2: it controls the exploitation feature and determines the quantity of exploitation.
- *sign* $(r - 0.5)$: manages the quality of the exploitation (direction) as well. It determined the local search direction.
- *Memory saving*: it saves several best-so-far particles and replaces them with destroyed particles. This function specifically enhances the potential of the EO to be exploited.
- *Equilibrium pool*: Exploration fades out as iterations pass and exploitation emerges. As a result, when the equilibrium candidates are similar to each other in the final iterations, the focus updating procedure will assist in local search, contributing to manipulation.

3.3.9. Computational complexity analysis

It is represented by a function that relates the algorithm's execution time to the issue's input size. Big-O notation is for this reason. Complexity is related to the particles number (n), the dimension (d), and iterations. It is the expense devoted to the function assessment (Gupta, Deep, & Mirjalili, 2020).

$$\begin{aligned} O(EO) = {} & O(\textit{problem definition}) + O(\textit{initialization}) \\ & + O(t(\textit{function evaluations})) + O(t(\textit{Memory saving})) \\ & + O(t(\textit{Concentration Update})) \end{aligned} \tag{17}$$

So, the total computational complexity can be calculated as below:

$$O(EO) = O(1 + nd + tcn + tn + tnd) \cong O(tnd + tcn) \tag{18}$$

As can be demonstrated, the complexity is polynomial in nature. As a result, EO can be called a useful algorithm.

3.4. Fitness Function

The fitness function is as follows:

$$Fitness = W_1 * \ cost \ + W_2 * feasible \ travels \tag{19}$$

Where *cost* is the *total displacement time* and *feasible travels* is the number of feasible travels. Our goal is to minimize any *cost (total displacement time)* and maximize *feasible travels* using an EO algorithm and hidden Markov model.

In Eq. (19), $W_1 + W_2 = 1$. The elements of the objective function mentioned above should be normalized as follows:

$$N\,(V) = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{20}$$

Where N(V) is the normalized value of either *cost (total displacement time)* or *feasible travels.*

4. Simulation results

The model is used for public transportation bus service in Santander. The model is loaded using data from the evening rush hour, demanding about 4000 trips. The buses' size that is currently functioning in the city is 90 feasible urban travels/buses with an average travel time of 20 min. Six various bus models are utilized. Selected sizes were 30, 60, 90, 120,

Table 1
Unit costs as a function of the size.

Bus size	CK (€/km)	CF(€/bus/h)
30	0.30	14
60	0.40	23
90	0.60	32
120	0.80	35
150	0.85	37
180	0.90	41

150, and 180 feasible travels/vehicle. Table 1 summarizes the unit costs (CK) and fixed costs (CF)) utilized based on the size of the bus (dell'Olio et al., 2012).

Fig. 4 indicates the convergence diagrams of the EO algorithm and Markov model method. In the rest of this section, we have compared the proposed way with the modified Floyd-Warshall algorithm (Sakharov, Chernyi, Saburov, & Chertkov, 2021) PSO algorithm (Xu, Yang, & Gao, 2010, pp. 513–517) in terms of minimum cost (displacement time) and a maximum number of feasible urban travels.

❖ Cost (total displacement time)

The suggested technique outcomes are compared with the modified Floyd-Warshall algorithm and PSO algorithm route optimization algorithms. Floyd-warshall's algorithm does not just look for the shortest path between two particular nodes, but the shortest path table between them is created. Floyd-warshall algorithm is one type of algorithm that all pairs shortest path, which is to find the shortest route for all pairs of nodes on a graph. PSO is a population-based optimization technique that can be quickly adopted and used to address a variety of function optimization issues. In Fig. 5, the cost diagram (average cost) of the proposed method is improved compared to other algorithms.

❖ Maximum number of feasible travels

The capacity of the proposed method is then compared with the modified Floyd-Warshall algorithm and PSO algorithm. Floyd-warshall's algorithm is one of the dynamic programming variants, a method that performs problems with the solution generated as an interconnected decision. Solutions are formed from the previous, and there is a possibility of more than one solution, and the PSO algorithm is used to offer a solution to the transportation network design problem. The purpose of this method is to increase the items of feasible direct travels per unit of length. The results of comparing these algorithms are illustrated in Fig. 6. It is worth noting that the two mentioned variables were used to compare. The results show that enhancing the ways leads to the elevation of the feasible travels than other methods.

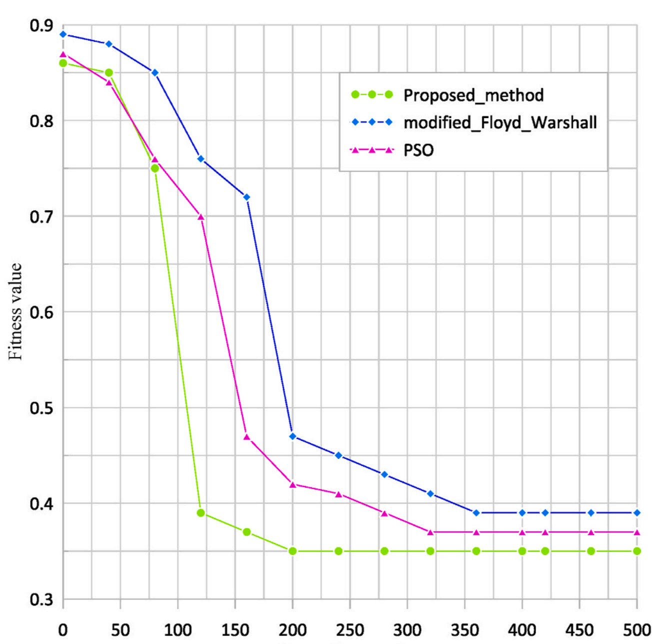

Fig. 4. Convergence diagram of the EO algorithm and Markov model.

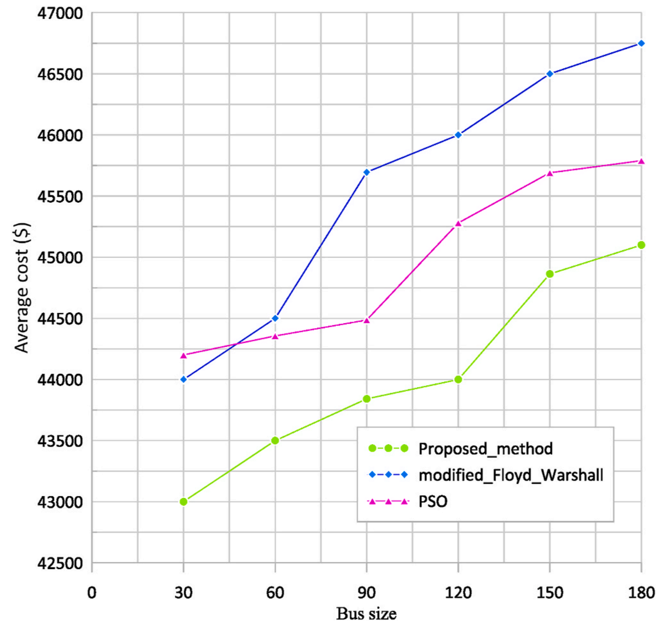

Fig. 5. The association among the average cost of routes and the bus size.

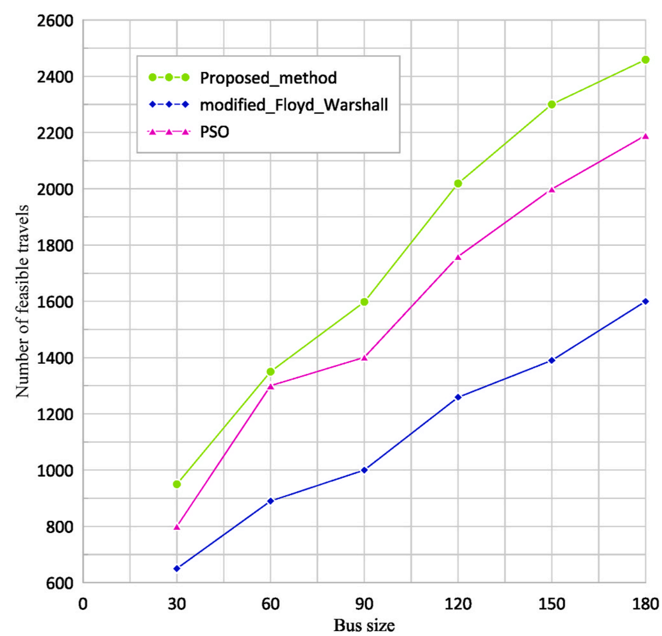

Fig. 6. Comparison of the outcomes for some feasible urban travels with the proposed method.

5. Conclusion

Because public transportation financing is under strain, the need to improve service quality is rising. Shorter and more convenient travel times are critical components of improved and more efficient public transportation. By eliminating bottlenecks in the processes, prices can be minimized while service efficiency increases. Consequently, revenue and ridership are increased. In this way, public transportation's cost efficiency is increased in two directions. The challenge of optimizing routes in this paper has been minimizing costs and maximizing the number of feasible travels using the Markov model and EO algorithm. The proposed method was compared to modified Floyd-Warshall algorithm methods. The results showed that applying the Markov model and EO algorithm significantly improved cost (reduction of displacement time) compared to some famous algorithms. The proposed method has also improved travel capacity.

In a future survey, we want to factor in station passenger demand to help assess the efficiency of public transportation networks. Our future big topic will also be determining the optimum number and location in a public bicycle-sharing system and ensuring the best efficiency at the lowest expense when installing a general plan in an urban transportation network.

CRediT authorship contribution statement

Yun Zhang: Conceptualization, Methodology, Simulation, Revision based on Comments of the Reviewers. **Weichu Xue:** Investigation, Validation, Simulation and Revision based on Comments of the Reviewers. **Wei Wei:** Supervision and Validation of the Results. **Habibeh Nazif:** Writing the Draft Version.

Acknowledgements

Thanks for the sponsorship of "University Young Teacher Training Funding Program of Shanghai".

References

Azhir, E., Navimipour, N. J., Hosseinzadeh, M., Sharifi, A., & Darwesh, A. (2021a). An efficient automated incremental density-based algorithm for clustering and classification. *Future Generation Computer Systems, 114*, 665–678.

Azhir, E., Navimipour, N. J., Hosseinzadeh, M., Sharifi, A., & Darwesh, A. (2021b). An automatic clustering technique for query plan recommendation. *Information Sciences, 545*, 620–632.

Bagal, H. A., Soltanabad, Y. N., Dadjuo, M., Wakil, K., & Ghadimi, N. (2018). Risk-assessment of photovoltaic-wind-battery-grid based large industrial consumer using information gap decision theory. *Solar Energy, 169*, 343–352.

Basnak, P., Giesen, R., & Muñoz, J. C. (2020). Technology choices in public transport planning: A classification framework. *Research in Transportation Economics, 83*, 100901.

Beielstein, T., Parsopoulos, K. E., & Vrahatis, M. N. (2002). *Tuning PSO parameters through sensitivity analysis*. Universitätsbibliothek Dortmund.

Cats, O., & Jenelius, E. (2015). Planning for the unexpected: The value of reserve capacity for public transport network robustness. *Transportation Research Part A: Policy and Practice, 81*, 47–61.

dell'Olio, l., Ibeas, A., & Ruisánchez, F. (2012). Optimizing bus-size and headway in transit networks. *Transportation, 39*, 449–464.

Faramarzi, A., Heidarinejad, M., Stephens, B., & Mirjalili, S. (2020). Equilibrium optimizer: A novel optimization algorithm. *Knowledge-Based Systems, 191*, 105190.

Guo, Z. (2002). Review of indoor emission source models. Part 1. Overview. *Environmental Pollution, 120*, 533–549.

Gupta, S., Deep, K., & Mirjalili, S. (2020). An efficient equilibrium optimizer with mutation strategy for numerical optimization. *Applied Soft Computing, 96*, 106542.

Heidari Gharehbolagh, H., Hafezalkotob, A., Makui, A., & Raissi, S. (2017). A cooperative game approach to uncertain decentralized logistic systems subject to network reliability considerations. *Kybernetes, 46*(8), 1452–1468.

Hosseini Firouz, M., & Ghadimi, N. (2016). Optimal preventive maintenance policy for electric power distribution systems based on the fuzzy AHP methods. *Complexity, 21*, 70–88.

Jiang, Y., Sun, B., Li, G., Lin, Z., Zheng, C., & Shen, X. (2017). Highway passenger transport based express parcel service network design: Model and algorithm. *Journal of Advanced Transportation, 2017*, Article 1712325.

Kim, J., Jeong, J., Kim, H., & Park, J.-S. (2020). Cloud-based battery replacement scheme for smart electric bus system. *IETE Journal of Research, 66*, 341–352.

Kuznetsov, A., Kuznetsov, S., & Ushkar, M. (2020). *Automation of the learning process of a neural network as classifier in the formation of a design basis of complex transport systems. 2020 systems of signals generating and processing in the field of on board communications* (pp. 1–4). IEEE.

Liu, J., Chen, C., Liu, Z., Jermsittiparsert, K., & Ghadimi, N. (2020). An IGDT-based risk-involved optimal bidding strategy for hydrogen storage-based intelligent parking lot of electric vehicles. *Journal of Energy Storage, 27*, 101057.

Lu, M., Abedinia, O., Bagheri, M., Ghadimi, N., Shafie-khah, M., & Catalão, J. P. S. (2020). Smart load scheduling strategy utilising optimal charging of electric vehicles in power grids based on an optimisation algorithm. *IET Smart Grid, 3*, 914–923.

Manser, P., Becker, H., Hörl, S., & Axhausen, K. W. (2020). Designing a large-scale public transport network using agent-based microsimulation. *Transportation Research Part A: Policy and Practice, 137*, 1–15.

Parouha, R. P., & Das, K. N. (2016). A memory based differential evolution algorithm for unconstrained optimization. *Applied Soft Computing, 38*, 501–517.

Sadrishojaei, M., Navimipour, N. J., Reshadi, M., & Hosseinzadeh, M. (2021). A new preventive routing method based on clustering and location prediction in the mobile Internet of Things. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3049631

Sakharov, V., Chernyi, S., Saburov, S., & Chertkov, A. (2021). Automatization search for the shortest routes in the transport network using the floyd-warshell algorithm. *Transportation Research Procedia, 54*, 1–11.

Sefati, S., & Navimipour, J. N. (2021). A QoS-aware service composition mechanism in the Internet of things using a hidden Markov model-based optimization algorithm. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2021.3074499

Sidi, M. M. O., Hammadi, S., Hayat, S., & Borne, P. (2008). Urban transport network regulation and evaluation: A fuzzy evolutionary approach. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 38*, 309–318.

Srivastava, S., & Sahana, S. K. (2019). *Application of bat algorithm for transport network design problem*. Applied Computational Intelligence and soft computing, 2019.

Sumpavakup, C., Suwannakijborihan, S., Ratniyomchai, T., & Kulworawanichpong, T. (2018). Peak demand cutting strategy with an on-board energy storage system in mass rapid transit. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 42*, 49–62.

Xu, M., Yang, J., & Gao, Z. (2010). Particle swarm optimization algorithm in transport continuous network design problems. In *2010 third international joint conference on computational science and optimization* (pp. 513–517). IEEE.

Zanbouri, K., & Jafari Navimipour, N. (2020). A cloud service composition method using a trust-based clustering algorithm and honeybee mating optimization algorithm. *International Journal of Communication Systems, 33*, e4259.

Zhu, X., Long, H., Li, Z., Diao, J., LLiu, H., Li, N., & Xu, H. (2020). Implication of unsafe writing on the MAGIC NOR gate. *Microelectronics Journal, 103*, 104866.