

Article

Development of a Convolution-Based Multi-Directional and Parallel Ant Colony Algorithm Considering a Network with Dynamic Topology Changes

Eunseo Oh  and Hyunsoo Lee * 

School of Industrial Engineering, Kumoh National Institute of Technology, P.O. 39177, Gumi, Korea

* Correspondence: hsl@kumoh.ac.kr; Tel.: +82-054-478-7661

Received: 5 August 2019; Accepted: 27 August 2019; Published: 4 September 2019



Abstract: While network path generation has been one of the representative Non-deterministic Polynomial-time (NP)-hard problems, changes of network topology invalidate the effectiveness of the existing metaheuristic algorithms. This research proposes a new and efficient path generation framework that considers dynamic topology changes in a complex network. In order to overcome this issue, Multi-directional and Parallel Ant Colony Optimization (MPACO) is proposed. Ant agents are divided into several groups and start at different positions in parallel. Then, Gaussian Process Regression (GPR)-based pheromone update method makes the algorithm more efficient. While the proposed MPACO algorithm is more efficient than the existing ACO algorithm, it is limited in a network with topological changes. In order to overcome the issue, the MPACO algorithm is modified to the Convolution MPACO (CMPACO) algorithm. The proposed algorithm uses the pheromone convolution method using a discrete Gaussian distribution. The proposed pheromone updating method enables the generation of a more efficient network path with comparatively less influence from topological network changes. In order to show the effectiveness of CMPACO, numerical networks considering static and dynamic conditions are tested and compared. The proposed CMPACO algorithm is considered a new and efficient parallel metaheuristic method to consider a complex network with topological changes.

Keywords: metaheuristics; Gaussian Process Regression (GPR); dynamic network topology; discrete pheromone convolution

1. Introduction

The route generation in a complicated network is a representative NP-Complete problem requiring a non-deterministic solution time with the polynomial increase of problem complexity. It has a number of applications, such as escape route production, and transport optimization. Specifically, in dynamic environment circumstances that cause traffic congestion, such as road construction or traffic accident, the driver's effective route is subsequently compounded with various dynamic situations. In order to overcome this complexity and to produce an efficient route, various existing studies [1–7], including Ant Colony Optimization (ACO) [8,9], have been presented, and meta-heuristic algorithms have been applied.

This research proposes an efficient route production algorithm for large-scale networks or mazes, which have a start point and a terminal point, by combining the ACO algorithm with a new distributed algorithm.

The study of Dorigo, Maniezzo, and Colonia [1] presented a meta-heuristic algorithm that produces routes by describing an agent's traces as a pheromone model using an ACO algorithm. Li and Xiao [2]

used an ACO algorithm to search parallel routes by introducing a parameter matrix design method, in order to produce fast escape routes from a maze. Yan and Yuan [3] developed faster routes using the modification of the pheromone renewal rule, by minimizing the pheromone concentration in a blind spot, whenever an ant agent in the maze reached the blind spot. Moreover, Ilie and Badica [4] generated a more efficient route using an exchange of messages among various ant agents. Yoshikawa and Nagura [5] conducted a study to limit ant agents that suddenly behave in an unusual way, minimizing deviations in route production. Moreover, Yoshikawa and Otani [6] presented a study that aimed to limit the possibility of an ant agent being trapped in a dead end. Kim and Lee [7] minimized the escape time using the cooperation of multiple agents to avoid consequent congestion. In the studies mentioned above, all agents used a unidirectional direction-based strategy [1–7] to produce routes from the start point to the terminal point, in the escape from a maze or large network. However, as the network becomes more complex, more time and computational loads are added to this strategy.

In order to overcome this issue, this research study uses a new and efficient strategy, the “Multi-directional and Parallel Ant Colony Optimization (MPACO)” algorithm, to start agents at any number of points simultaneously, not simply from the start point to the terminal point. The MPACO algorithm supports parallel searches of agents, and uses a Gaussian Processes Regression (GPR)-based pheromone updating method. Specifically, the proposed algorithm generates evacuation routes faster than the previous algorithms in a dynamic network where the topology of the network is transformed due to several reasons, such as traffic congestion or construction.

Section 2 of this research study examines the current ACO algorithms and relevant studies. Section 3 proposes the MPACO framework in a static environment. Section 4 proposes the Convolution-based Multi-directional and Parallel Ant Colony Algorithm (CMPACO) framework that is effective in a dynamic environment. Section 5 analyzes the effectiveness of the proposed framework with the experimental results of ACO, MPACO algorithm, and CMPACO algorithm.

2. Background Knowledge

2.1. Routing Generation Methods and Ant Colony Optimization

The purpose of routing generation is to search for the more effective paths among numerous alternative paths. A number of metaheuristic algorithms have been applied to find more efficient routes. Table 1 summarizes several existing meta-heuristics and the relevant applications to generate more efficient routes in complex networks.

Table 1. Existing research studies for route generation.

Route Generation Based Research Studies	Applications and Characteristics
Garcia, Tria and Talampas [10]	Particle Swarm Optimization-based Energy efficient route generation to maximize vehicle mileage
Quang, Sanner, Morin and Aoul [11]	Route generation of large-scale networks using genetic algorithm
Yu, Liu, Liu, Hu, Zhang and Xiao [12]	Clustering routing algorithm based on glowworm swarm optimization
Xu, Wang and Sun [13]	Generation of efficient distributed model to reduce communication load using intelligent routing algorithm
Qiu, Zhong, Luo, Liu, Luo and Jiang [14]	Power distribution network routing generation using ACO

Among a number of meta-heuristic algorithms, swarm-based optimization methods [10,12,14] have been widely used for more efficient route generations. One of the representative swarm-based optimization is ACO. ACO is an algorithm derived by Dorigo [15,16], based on the cooperative behavior

of ants in finding the shortest route from their colony to the food location. Table 2 shows that the algorithm has been utilized as one of the meta-heuristics in solving the route problem in complex and difficult networks.

Table 2. Several applications using Ant Colony Optimization.

Research Studies Using ACO	Application Fields
Dorigo, Maniezzo, Colorni and Trubian [17]; Arnaout, Musa and Rabadi [18]; Chen, Lo, Wu and Lin [19]	Transportation and scheduling issues
Dorigo and Gambardella [20]; Oh and Lee [21]	Efficient route production
Ahmadizar, Barzinpour and Arkat [22]	Permutation flow shop scheduling problem
Dorigo, Maniezzo and Colorni [1]	Traveling salesman problem
Reza, Mahfujur, Abdur, Wail and Abdulmotaleb [23]	Wireless sensor network design
Maniezzo and Colorni [24]	The quadratic assignment problem

In a general ACO algorithm, each ant acts as an artificial agent that moves to the target probabilistically. At each step, the agent moves based on the graph topology of the problem, deposits the pheromone, and updates the pheromone amount, using iterations to increase the probability of the optimal route production.

Figure 1a depicts how agents move to produce an efficient route using ACO in a multilayered graph. All agents start at the same position (e.g., the Ant Colony) at the beginning of each iteration. However, agents in Figure 1b are initially located at random places, and search for the shortest path more efficiently. Section 3 provides the detailed process and method.

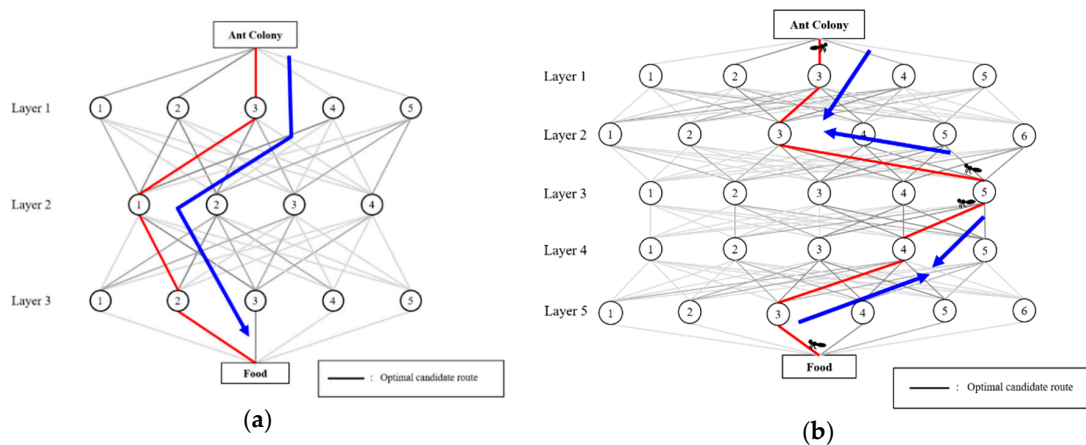


Figure 1. Conceptual comparisons between ACO algorithm and MPACO algorithm: (a) Optimal route search in ACO algorithm; (b) Optimal route search in MPACO algorithm.

The agents (the number of agents = AN) move from the first layer, the ant colony, up to the last layer, the food location. In each destination point, an agent can select one of the directly connected neighbor nodes, and diffuses its pheromone. The route is stronger as the concentration of pheromone is stronger.

$$P_{ij}^{(k)} = \frac{\tau_{ij}}{\sum_{j \in N_i^{(k)}} \tau_{ij}}, \quad \text{if } j \in N_i^{(k)} \tag{1}$$

In Equation (1), $P_{ij}^{(k)}$ denotes the probability that the k^{th} ant moves from node i to node j , $N_i^{(k)}$ denotes the set of all movable nodes directly connected to node i , and τ_{ij} is the pheromone deposited in the route from node i to node j .

$$f_k = \sum_{i \in S} \sum_{j \in S} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad k = 1, 2, \dots, N \tag{2}$$

Equation (2) indicates the cost function (f_k) when the k^{th} ant goes from node i to node j . S is the set of all nodes passed by the k^{th} ant. When the k^{th} ant is at node i , the coordinates are presented as (x_i, y_i) ; and when it is at node j , the coordinates are presented as (x_j, y_j) .

$$f_{best} = \min_{k=1,2,\dots,N} f_k \tag{3}$$

$$f_{worst} = \max_{k=1,2,\dots,N} f_k \tag{4}$$

The optimal route (Equation (3)) and the worst route (Equation (4)) between routes selected by the k^{th} ant are determined. When f_{best} is determined at one iteration count, the algorithm updates the deposited pheromone for the next iteration. The updated pheromone is used as a guide to search for the future route of other ants.

As shown in Equations (5) and (6), the ants deposit more pheromone on the selected route, using Equations (2) and (3). This enhanced route is used as a key indicator to the preferred route in future route selection by other ants. Longer routes that do not attract attention are reduced relative to pheromone evaporation as shown in Equation (7), and the pheromone in each route is updated as learning processing.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^N \Delta\tau_{ij} \tag{5}$$

$$\Delta\tau_{ij} = \begin{cases} \frac{C f_{best}}{f_{worst}}, & \text{if } (i, j) \in \text{global best tour} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \tag{7}$$

In Equation (6), C is a parameter to control the scale of the pheromone updated as an ant moves, while in Equation (7), ρ is a pheromone decay parameter. The pheromones of all the route evaporate, as shown in Equation (7). The pheromone of the route with the minimum objective function value is updated, as shown in Equation (5). The renewal amount of pheromone is determined using Equation (6) with f_{best} and f_{worst} . Closest nodes with fewer objective function values are selected using these pheromone updating processes. Finally, the algorithm generates the efficient solution route.

While a number of research studies have applied the ACO method to relevant application fields, it is limited in a complex network with dynamically changing topology. This research proposes the MPACO and CMPACO methods. The detailed algorithms are provided in Sections 3 and 4. The following subsection explains the Gaussian Process Regression (GPR) method, which the proposed method uses for estimating and updating pheromones.

2.2. Gaussian Processes Regression

GPR [25–27] is one of the representative nonparametric Bayesian algorithms. GPR is a technique of estimating and updating the frequency of occurrence, using a Gaussian nonlinear distribution function with a mean function and a covariance function. GPR algorithm has been used in many research fields, as provided in Table 3.

Table 3. Research applications using GPR.

Research Studies Using GPR	Application Fields
Ak, Ergonul, Sencan, Torunoglu and Gonen [28]	The time and space prediction of an infectious diseases
Luttinen and Ilin [29]	Sea level temperature reconstruction
Nguyen and Peters [30]	Kinetics model estimation
Nguyen, Hu and Spanos [31]	Efficient building field formation by estimating indoor environment fields
Chen, Qian, Meng and Nabney [32]	Wind prediction for energy efficiency

GPR has been used in many fields, as it has high prediction measurement for changing environment based on a Bayesian framework, and enables probabilistic analyses using Gaussian distribution.

Figure 2 illustrates the process for estimating the final output value Y_* using an f_* at the input value X_* after producing the intermediate value f_i by Bayesian renewal to convert the input value X_i to the observed value Y_i .

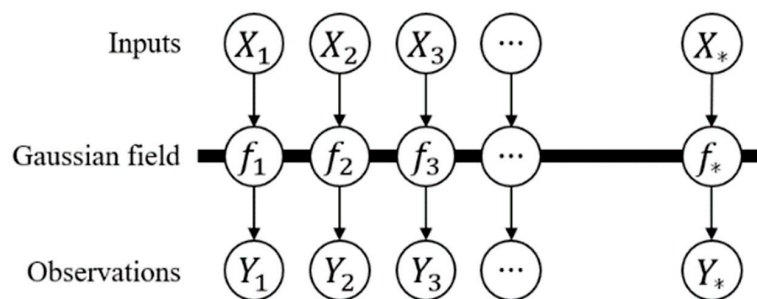


Figure 2. Schematic graph of a Gaussian processes regression.

Equations (8)–(11) summarize the general GPR model.

$$y = f + \epsilon, \text{ where, } \epsilon \sim N(0, \sigma_y^2 I) \tag{8}$$

$$D = \{(X_i, Y_i), i = 1, \dots, a\} \tag{9}$$

$$f \sim GP(m(X), k(X, X_*)) \text{ where } m(X) = E[f(X)] \text{ and } k(X, X_*) = E[(f(X) - m(X))(f(X_*) - m(X_*))]$$

In Equation (8), ϵ is a noise parameter with a mean of 0 and a variance of σ_y^2 subsequent to the Gaussian distribution, and I is an Identity matrix considering the data dimension. Equation (10) is the “distribution over functions” [33] for converting to the target value Y , if a training data is assigned with input vector X and output vector Y as shown in Equation (9), which is a Gaussian process distribution modeled with mean $m(X)$ and variance covariance $k(X, X_*)$. $k(X, X_*)$ is a covariance function ($cov(f(X), f(X_*)) =$), and indicates the similarity measure between both data. In general, a special kernel function is modeled as a radial basis kernel of Equation (11).

$$k(X, X_*) = \exp(-\frac{1}{2}|X - X_*|^2) \tag{11}$$

Equation (10) can be mapped to Equation (12) with the condition of the absence of noise, and then the pre-distribution is expressed as Equation (13).

$$f_* \sim N(0, k(X_*, X_*)) \tag{12}$$

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(X, X) & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix}\right) \tag{13}$$

However, the value obtained by the measurement of the actual data contains noise in general. In this manner, the Y value is expressed as Equation (14).

$$Y \sim N(0, k + \sigma_y^2 I) \tag{14}$$

This indicates that the estimated Y using a general GPR model follows the attributes of multivariate normal distribution with mean 0 and covariance $k + \sigma_y^2 I$. This derived GPR model is used as a prediction model using Bayesian framework.

When the function value $f_*(X_*)$ is predicted for the test data X_* , the pre-distribution for $f_*(X_*)$ considering the occurring noises is modeled using Equation (15).

$$\begin{bmatrix} Y \\ f_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(X, X) + \sigma_y^2 I & k(X, X_*) \\ k(X_*, X) & k(X_*, X_*) \end{bmatrix}\right) \tag{15}$$

Using the newly updated Gaussian process, the derived Maximum Likelihood Estimator (MLE) and its variance follow Equations (16) and (17), respectively.

$$\bar{f}_* = k(X_*, X)^T (k(X, X) + \sigma_y^2 I)^{-1} Y \tag{16}$$

$$V [f_*] = k(X_*, X_*) - k(X_*, X)^T (k(X, X) + \sigma_y^2 I)^{-1} k(X, X_*) \tag{17}$$

This research uses the introduced GPR framework for connecting and estimating the pheromones of paralleled agents. Section 3 provides the detailed framework and algorithms of MPACO algorithm using the GPR pheromone updating method.

3. Multi-Directional and Parallel Ant Colony Optimization (MPACO)

MPACO algorithm orients “multiple directional searches”, in which each group of ants start at any point, as opposed to the ACO algorithm, where all agents start from the same start point (Ant Colony). Moreover, MPACO algorithm has the characteristic of “parallelism”, in which each agent searches in multiple directions simultaneously.

This section explains how MPACO works in a static network environment. Figure 3 illustrates the detailed procedures of the MPACO algorithm.

Figure 3a depicts the initial situation of the MPACO algorithm. The agents of each group start at the start point and the terminal point as usual, not just the start point of the maze. If the number of groups (randomly selected starting points) is v , each group is composed of AN/v agents, where AN is the number of overall agents. When the agents from different groups often occur together with each other in a node, this node is defined as a collision node (q). The existence of the collision node implies that the node may have a higher probability to be used as one of the paths. For this reason, when the collision node is detected, the pheromone is diffused and calculated.

The pheromone value is estimated using a GPR-based updating method. As shown in Figure 3b, the collision node q is detected, and the amount of pheromone (Figure 3c) is adjusted using a GPR around the collision node. Equation (18) defines the GPR-type pheromone distribution D , where X denotes a node belonging to radius w in the detected collision node.

$$D(Y_*) \sim N(m(X), k(X, X_*)) \tag{18}$$

$$M^{(int)} = \arg \max_{\forall (i,j) \in A} \tau_{ij} \tag{19}$$

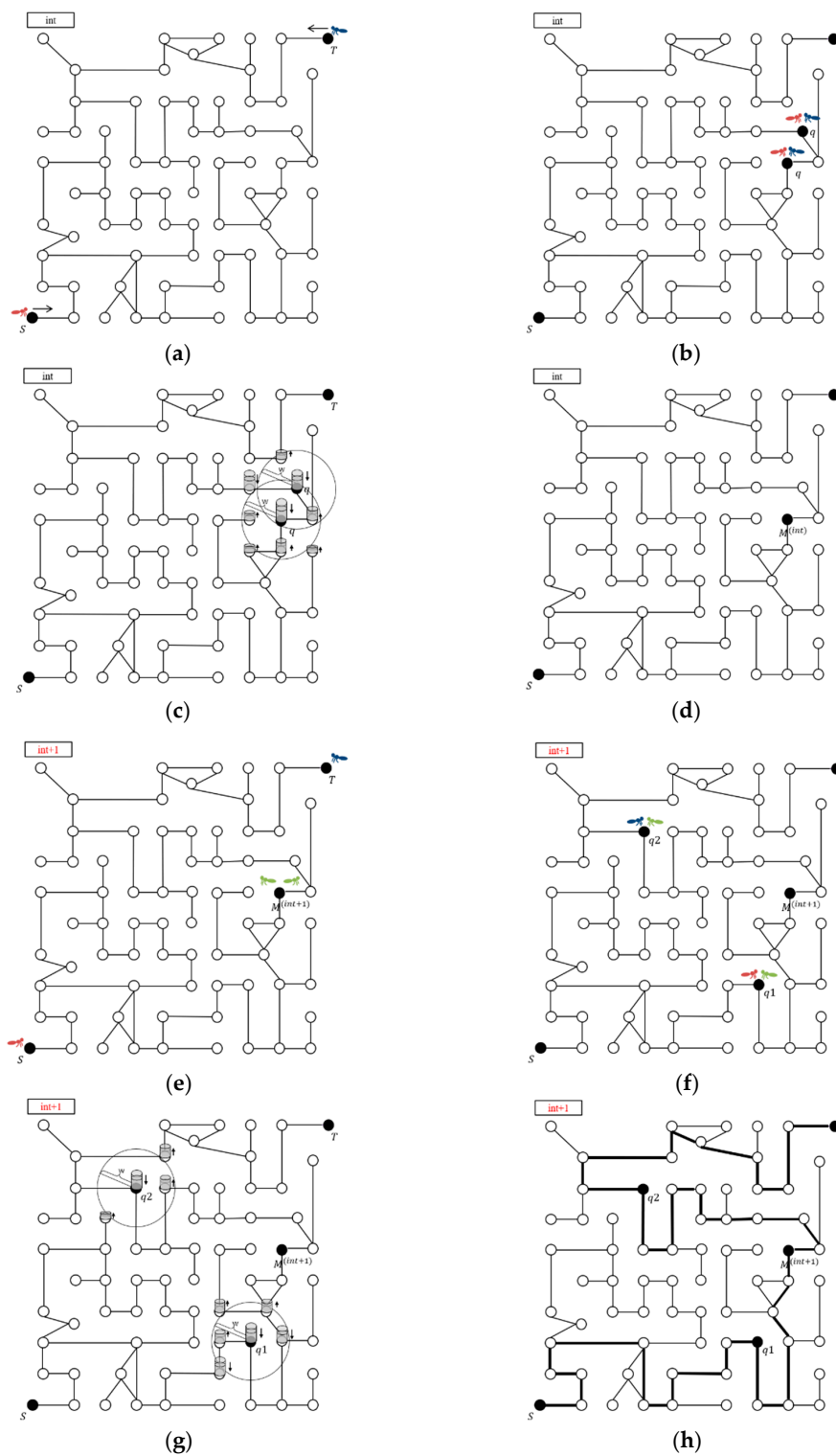


Figure 3. Conceptual framework of MPACO algorithm and procedures. (a) Initial search from two different nodes; (b) Discovery of the collision node q ; (c) GPR based pheromone renewal in Node q ; (d) Choice of the node with the highest pheromone at the iteration “int”; (e) Searches with multi directions at the nodes including $M^{(int+1)}$ at the following iteration “int + 1”; (f) Discovery of the collision nodes $q1$ & $q2$ at the following iteration; (g) GPR based pheromone updating in Node $q1$ & $q2$; (h) Search and connection of the divided routes.

Equation (18) is derived using the prediction procedure provided in Section 2.2. The pheromone value (Y_*) of node X_* is predicted based on the GPR, and used as the updated pheromone value. The parameter radius w depends on the characteristics of the problem. This research study uses 2 as the value for each experimental measurement.

After updating the pheromone, a new node (Equation (19)) with the highest pheromone in the network is selected. The node selected is called an intermediate node ($M^{(int)}$), as shown in Figure 3d. The reasoned $M^{(int)}$ node is added as a new starting node of the agents, as shown in Figure 3e, and additional collision nodes (Figure 3f) are detected in the following iteration. Then, the GPR-type pheromone is updated at the additional nodes, as shown in Figure 3g, using Equation (18). Finally, the strongest pheromone points are linked with each other, and determine a final effective route, as shown in Figure 3h.

When an agent diffuses pheromones in the shortened route using the cost function, MPACO algorithm with collision nodes and intermediate nodes generates the route to the large network faster and more efficiently than ACO algorithm. When these collision and intermediate nodes are not found, the MPACO algorithm follows the route search procedure of the ACO algorithm.

Figure 4 depicts a flowchart of MPACO algorithm that produces an optimal candidate route. As shown in Figure 4, two collision nodes are detected in the initial searches. Then, the intermediate node is reasoned using the GPR rule. The node is registered to starting node lists. Then, the final route is completed using the same iterations.

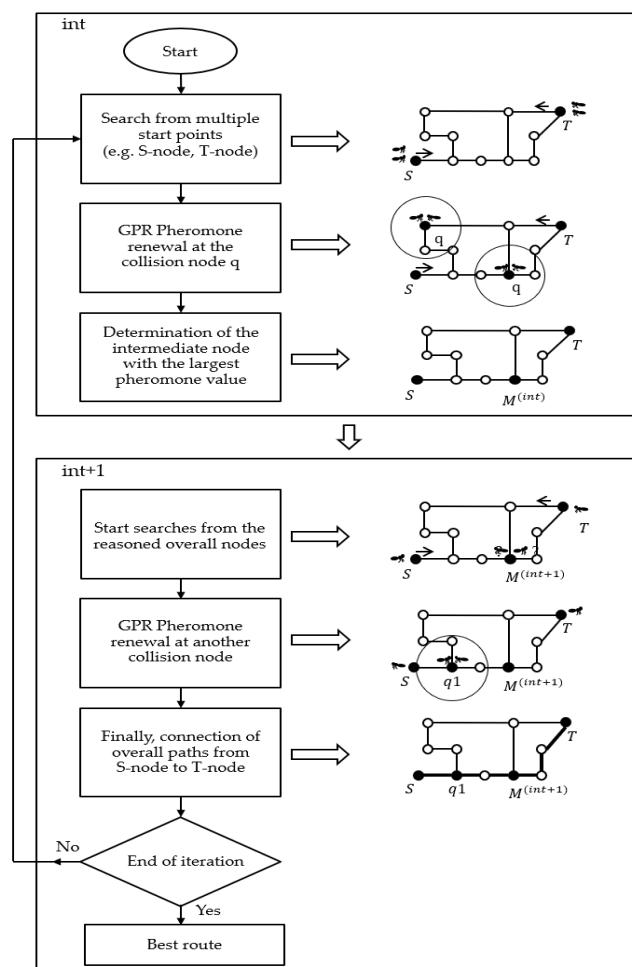


Figure 4. MPACO algorithm-based route generation framework.

Table 4 summarizes the comparison of the computation speed using ACO and the proposed MPACO. The network as shown in Figure 4 is used as a test network. Four ants are used in both algorithms, and the optimal paths are obtained in one iteration.

Table 4. Comparison of the computation speed between ACO and MPACO.

	ACO	MPACO
Iterations	1	1
Computation speed (Agents' Total Movement)	70.64	61.64

Table 4 shows that the overall agents' movement using ACO is longer than the movement using MPACO. This indicates that the computation speed of MPACO is faster than the speed of ACO.

Algorithm 1 explains the route search algorithm for detecting a collision node. As explained with the illustrations, initial agents are divided into two groups, which start at the S-node and T-node, respectively.

Algorithm 1. Route search algorithm for detecting a collision node.

Initialize:

Current iteration $\rightarrow int$

Overall number of agents $\rightarrow AN$

for $k=1$ to $AN/2$

Ants start from S node and T node (S node has $AN/2$ ants, and T node has $AN/2$ ants).

Choose the next node using a $P_{ij}^{(k)}$. (Equation (1))

end for

When ants in different groups meet in a node, it is classified as a collision node (q).

If (the existence of q == false)

Go to Algorithm 2.

else

Go to Algorithm 3.

end if

When a collision node is not detected, MPACO follows the procedures of ACO algorithm, as shown in Algorithm 2.

Algorithm 2. Route search in the case of the non-existence of a "collision node".

Initialize:

Evaporation rate $\rho \in (0, 1]$

The objective function is to minimize the distance of the route where the ants are moved (Equation (2)).

Combine the distances moved in each section.

Choose the optimal candidate route f_{best} (Equation (3)).

Evaporate the pheromones (Equation (5)).

Update the pheromones on f_{best} route (Equation (6)).

return Algorithm 1

When a collision node is detected, the GPR-type pheromone amount is estimated and updated with Algorithm 3.

Algorithm 3. GPR based Pheromone renewal Algorithm.

Initialize:Radius $w \in I^+$ In the q node,all nodes in the radius $w = X$ Y = the pheromone values of the X .Input: Training data point $\{ X, Y \}$ GPR is applied to predict test data X_* and Y_* (Equation (18)).Update pheromone with Y_* value on X_* node.Choose the $M^{(int)}$ node with the highest pheromone value (Equation (19)).Go to Algorithm 4.

Then, the subsequent process is to find an intermediate node (M^{int}) with the highest pheromone value near a collision node. The reasoned M^{int} is listed as another starting node, and agents start their search with an updated starting node list ($M^{(int+1)}$), as shown in Algorithm 4. The updated node list ($M^{(int+1)}$) at the following iteration ($int+1$) includes the initial starting point (S-node) and the target node (T-node). If the number of nodes in $M^{(int+1)}$ is greater than the number of overall agents (AN), additional detection processes for collision nodes and intermediate nodes are stopped. The numerical analyses using the proposed algorithm are provided in Section 5.1.

Algorithm 4. Route generation Algorithm with an intermediate node ($M^{(int+1)}$).

Initialize:Current iteration $\rightarrow int+1$ number of $M^{(int+1)} \rightarrow v$ **for** $k=1$ to AN/v Ants start from all $M^{(int+1)}$ nodes that are detected.Choose the next node using $P_{ij}^{(k)}$ (Equation (1)).**end for**

4. Dynamic Configurable Network and Convolution-Type Multi-Directional and Parallel Ant Colony Optimization (CMPACO)

While the previous section shows the effectiveness of the proposed network with networks with static topology, a number of real networks have dynamic changes in their topologies and configurations due to various reasons, such as route constructions and heavy traffic jams. An algorithm that constructs an efficient route in a dynamic network has been used in many research fields, as provided in Table 5. The changes of network topologies for learning processes have resulted in low performances of the existing algorithms. This section provides a variation of the MPACO algorithm that overcomes the issue, and explains how it works.

Although the MPACO algorithm is more successful than the ACO algorithm in static network environments, dynamic changes in network topologies might invalidate the pheromone-based route reasoning using the learning processes of MPACO. In order to overcome this issue, a new and efficient dynamic network framework (Convolution-type Multi-directional and Parallel Ant Colony Optimization (CMPACO) algorithm) is proposed. The new algorithm uses a discrete Gaussian convolution. Figure 5 illustrates how the pheromone is renewed when the collision nodes are detected using the proposed CMPACO algorithm.

As shown in Figure 5a, when a collision node is detected, the algorithm examines another collision node that shares the area within the given radius α, β, γ . The pheromone at each collision node is modeled as a step-based Gaussian distribution, as shown in Equation (20). Equation (20) is a discrete Gaussian distribution with mean 0 and standard deviation according to the discrete ranges

(α, β, γ) of the converted radius, while 't' is a parameter for controlling the concentration of diffused pheromone values.

$$f(x) = \frac{t}{\sqrt{(2\pi * \sigma)}} \exp^{-\frac{1}{2*\sigma^2}x^2}, \begin{cases} x = \alpha, & 0 < x \leq \alpha \\ x = \beta, & \alpha < x \leq \beta, \quad 0 < \alpha < \beta < \gamma \\ x = \gamma, & \beta < x \end{cases} \quad (20)$$

Then, pheromones in the shared regions are synthesized with the convolution of both the step-based Gaussian distributions, as shown in Figure 5b.

Table 5. Research applications on dynamic networks.

Research Studies on Dynamic Networks	Application Fields
Eklund, Kirkby and Pollitt [34]	Use dijkstra’s classic double bucket algorithm to find the network route
Giraldal, Rodriguez, Pernas, Higuera, Ortega and Zarzuela [35]	Intelligent system for transport fleet management
Noto and Sato [36]	Route search for car navigation systems using the extended dijkstra algorithm
Moss and Segall [37]	Optimal control theory to the problem of dynamic routing
Cauvery and Viswanatha [38]	Generate the shortest route applied ants algorithm and genetic algorithm

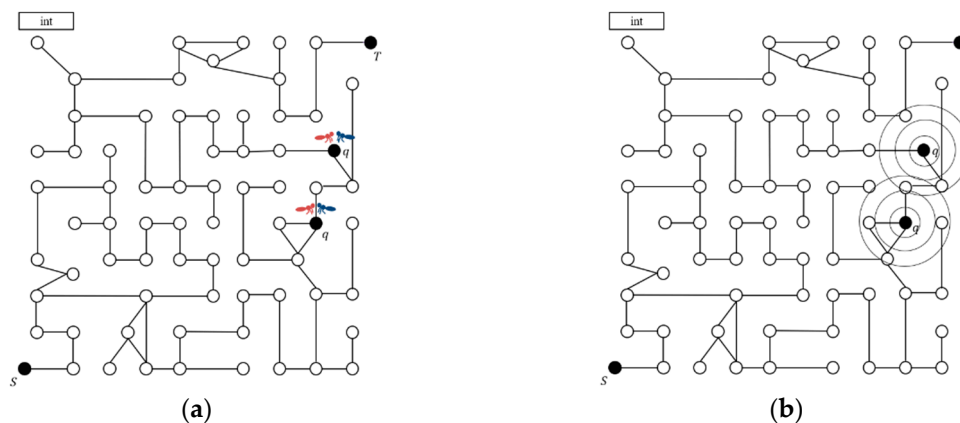


Figure 5. Discrete Gaussian convolution based pheromone renewal. (a) Detection of both Collision nodes; (b) Discrete Gaussian convolution based pheromone renewal.

As shown in Figure 6, the pheromone amount of A, B, and C in radii $\alpha, \beta,$ and γ at node q are modeled using Equation (21).

$$\begin{cases} \tau_{ij} \leftarrow A, & \text{if } (ij) \in \pi\alpha^2 \\ \tau_{ij} \leftarrow B, & \text{if } (ij) \in \pi\beta^2 - \pi\alpha^2 \\ \tau_{ij} \leftarrow C, & \text{if } (ij) \in \pi\gamma^2 - \pi\beta^2 \end{cases} \quad (21)$$

Then, the pheromone convolution between both collision nodes is executed using Equations (20) and (21). Figure 7 shows the pheromone convolution using both discrete Gaussian distributions.

The subsequent processes follow the processes of MPACO. While MPACO algorithm purses a node with the highest pheromone using the GPR process, topology changes in a target network might lead to mis-estimated nodes. However, the proposed CMPACO uses the discrete additions of the amount of diffusion of each pheromone when multiple collision nodes are located within a close radius,

as shown in Figure 7. The comparatively simple convolution-based node selection is influenced less by the dynamic changes in network topology. Then, it makes the finding of the effective route more robust against small network changes in the learning process. Section 5.2. provides the effectiveness of the proposed CMPACO algorithm with the examples of the dynamically changing networks

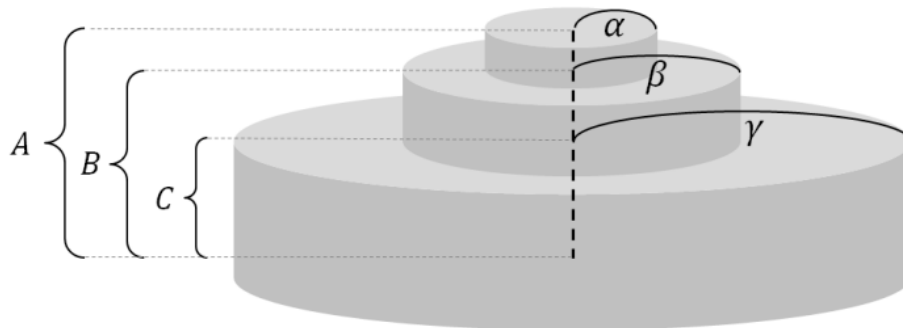


Figure 6. Estimation of pheromone using a discretized Gaussian distribution.

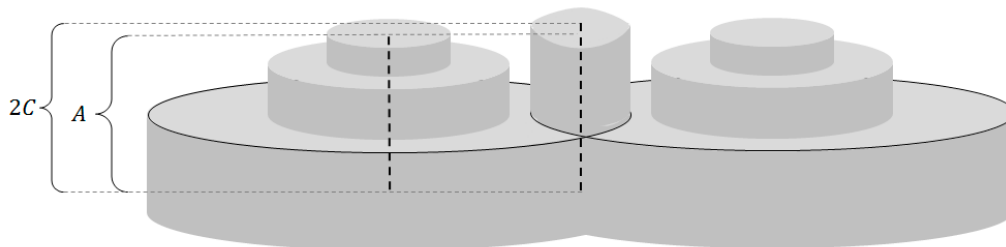


Figure 7. Calculation of pheromone in both collision nodes.

5. Numerical Studies and Performance Analysis of MPACO and CMPACO

5.1. Performance Comparisons of MPACO and ACO

This section provides the performance comparisons between ACO and the proposed algorithm; MPACO under networks with fixed topology. As explained in the previous section, the MPACO algorithm has the advantage of generating an efficient route faster than ACO algorithm in a large network, by simultaneously searching for routes in multiple directions, and updating pheromones based on GPR. In order to verify the validity of the proposed algorithm, numerical experiments and analyses under labyrinth type networks are provided and compared with ACO algorithm.

Tables 6 and 7 summarize the conditions and the assumptions for the numerical experiments conducted and the parameters used. Table 8 shows the test platform parameters.

Table 6. Experimental conditions and assumptions.

Type	Detailed Assumptions
Condition No.1	The same number of ants is used in both ACO and MPACO tests.
Condition No. 2	The locations for ant colony and food location are the same in each network for both cases.
Assumption No. 1	When ants in different groups meet each other in a node, the ants stop moving.
Assumption No. 2	When the ants reach the wall/block of the maze, they go back to find another route.

Table 7. Parameters and their values for experiments.

Variable	Value
Evaporation rate (ρ)	0.5
Initial pheromone (τ_{ij})	1
Number of ant (AN)	12

Table 8. The test platform.

Platform	Parameters
System Version	Windows 10 Pro. 64-bit
CPU	Intel(R) Core (TM) i7-4870HQ @ 2.50GHz
RAM	16GB

Figure 8 illustrates the tested network, and how the intensity of pheromone is strengthened during the iterations using the proposed MPACO algorithm. As shown in Figure 8a–d, the x and y axes are the coordinates of the node, and the z axis indicates the amount of pheromone. The explained GPR process is applied in every iteration, and the pheromone is deposited around the collision nodes. As a result, it can be seen that the route from each start node to the terminal node is searched effectively.

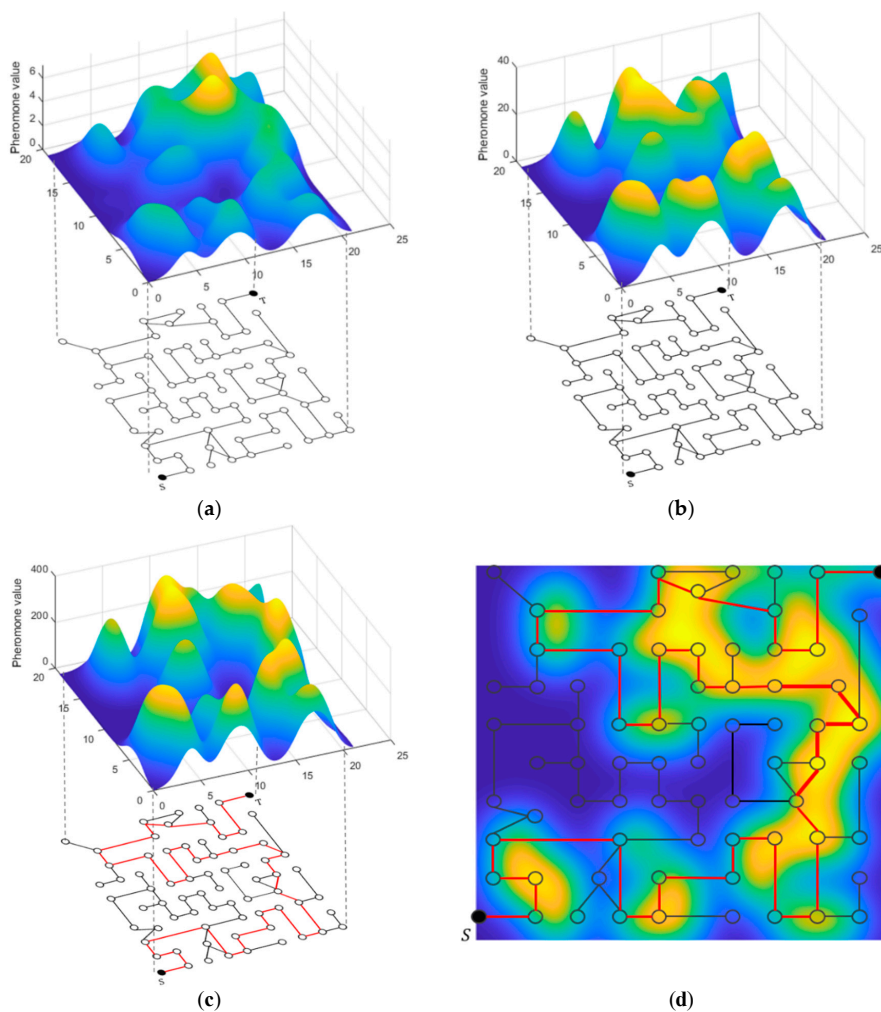


Figure 8. Pheromone distribution using the MPACO algorithm. (a) Pheromone distribution view in the maze (the 3rd iteration); (b) Pheromone distribution view (the 10th iteration); (c) Pheromone distribution view (the 100th iteration); (d) Final pheromone distribution and the optimal route using MPACO.

Figure 9 illustrates the change of the cost function value in each iteration using ACO and MPACO, where the x axis indicates the number of iterations, and the y axis the route distance from the start node to the terminal node.

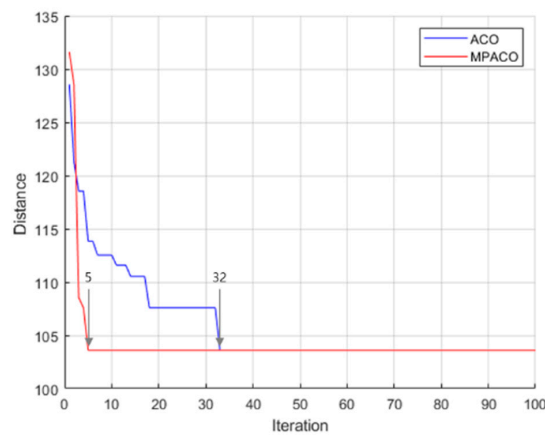


Figure 9. Route distance comparisons per each iteration between ACO and MPACO.

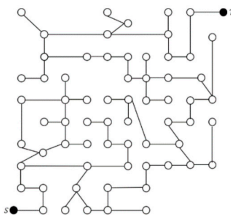

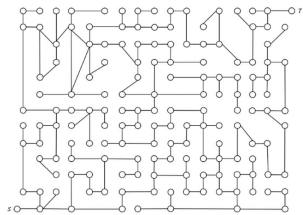
Figure 9 shows that the proposed MPACO algorithm produces the optimal routes much faster than ACO algorithm. Table 9 summarizes the computation time comparison between both algorithms.

Table 9. Computation time comparison between ACO and MPACO.

Test Framework	Iteration	Computation Time
ACO	32	19.16(s)
MPACO	5	4.92(s)

In order to show the effectiveness of the proposed algorithm, various networks are tested, as shown in Table 10. The tests using various networks provide the effectiveness of the proposed MAPCO framework compared to the existing ACO algorithm.

Table 10. Test comparison with various networks.

Map	Iteration	Route Search Time	
		ACO	MPACO
	100	18.96(s)	12.98(s)
	100	24.91(s)	16.41(s)
	100	27.63(s)	19.21(s)

5.2. Performance Analyses of CMPACO Considering Dynamic Network Topology

In order to verify the validity of the proposed CMPACO algorithm, numerical experiments are conducted on the dynamic changes of network topology, where during the learning process, a portion of the network changes. As shown in the previous section, the topology of the network often changes due to the sudden congestion that in real applications is often caused by road construction. The tests with CMPACO are analyzed and compared with tests with the ACO algorithm and MPACO algorithm. The test conditions and assumptions follow the descriptions in Tables 6 and 7 of Section 5.1.

For the comparison with the previous methods used, the route topology of the maze is set to change after every (30 and 60) iterations to describe a network with a dynamic topology change. σ and t in Equation (20) are modeled in this experiment with (1.4 and 5), respectively. As shown in Figure 10, a network has two topological changes during the learning processes.

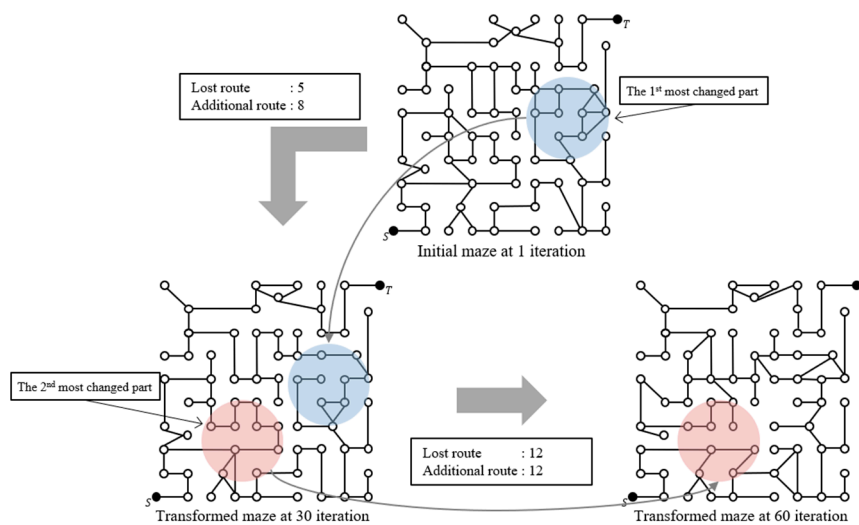


Figure 10. A network with two topological changes during the learning process.

Figure 11 depicts the amount of final pheromone using MPACO and CMPACO algorithm in the dynamically changing maze.

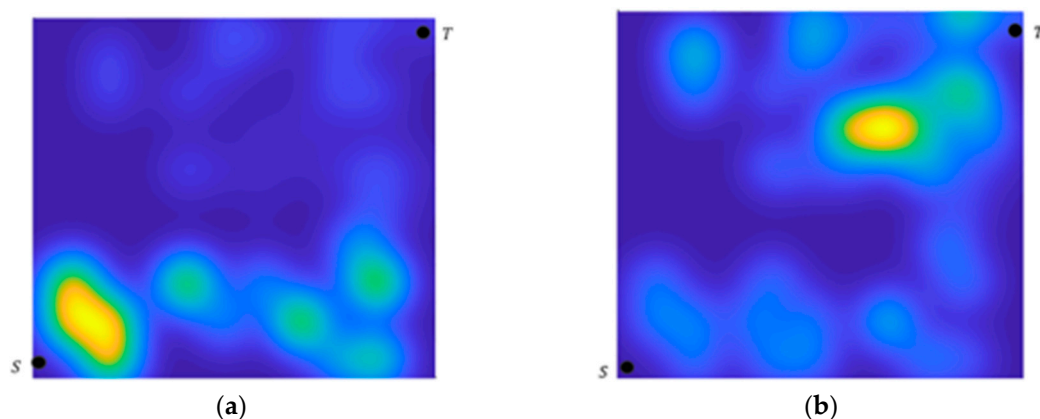


Figure 11. Pheromone distribution using MPACO and CMPACO in the maze. (a) Final pheromone distribution using MPACO in the maze; (b) Final pheromone distribution using CMPACO in the maze.

Figure 11 shows that the x and y axes are the node coordinates on the final network, and the level contour indicates the amount of finally formed pheromones. In the case of the MPACO algorithm (Figure 11a), the pheromone distribution is strong, primarily at the start point; however, the pheromone is low in the movement route. This observation indicates that MPACO is not effective in generating the

final route considering dynamical topology changes. On the other hand, in the case of the CMPACO algorithm (Figure 11b), a comparatively strong pheromone is formed around the movement route.

Table 11 summarizes the results of applying ACO, MPACO, and CMPACO. The results show that the proposed CMPACO has better performances than other algorithms in a network environment with variable topology. The test platform parameters follow Table 8.

Table 11. Computation time comparisons among ACO, MPACO and CMPACO.

Test Framework	Iteration	Computation Time (Unit: Seconds)
ACO	100	17.15
MPACO	100	25.81
CMPACO	100	9.88

Figure 12 illustrates comparisons of the cost function values using ACO, MPACO, and CMPACO algorithms in the network. Each dotted line indicates the time when the topology changes.

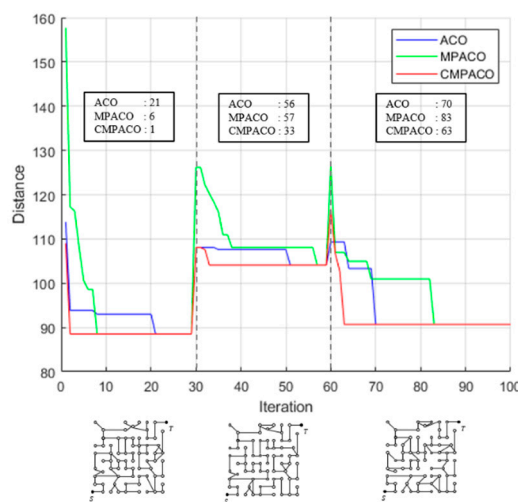


Figure 12. Comparisons of the cost function values using ACO, MPACO and CMPACO.

Taking into account the case of the ACO algorithm, the route search for the shortest distance takes the longest time in all dynamic environments. In the case of the MPACO algorithm, the performance of MPACO is more unstable than that of the proposed CMPACO algorithm.

The MPACO algorithm updates the pheromone around the collision node by applying the GPR whenever the route changes in a dynamic environment. Then, there is a high probability that the collision node will fail to predict the amount of pheromone in the neighboring node, due to the sudden change. It is impossible to predict a new collision node caused by a dynamic network topology change and the pheromone of a changed route. In addition, an effective route cannot be produced by predicting the pheromone distributed in the previous collision node.

However, the CMPACO algorithm updates the pheromone value discretely from the collision node to the radius node. As explained previously, CMPACO's discretized pheromone convolution is influenced less by the changes of network topology. The provided numerical example shows the effectiveness of the proposed algorithm, particularly when searching a complex maze with dynamic topology changes.

In order to compare algorithms' performances, a number of networks with dynamic topological changes are tested using ACO, MPACO and CMPACO. As shown in Figure 13a,b, two more networks are configured with two dynamical topologies during the learning processes.

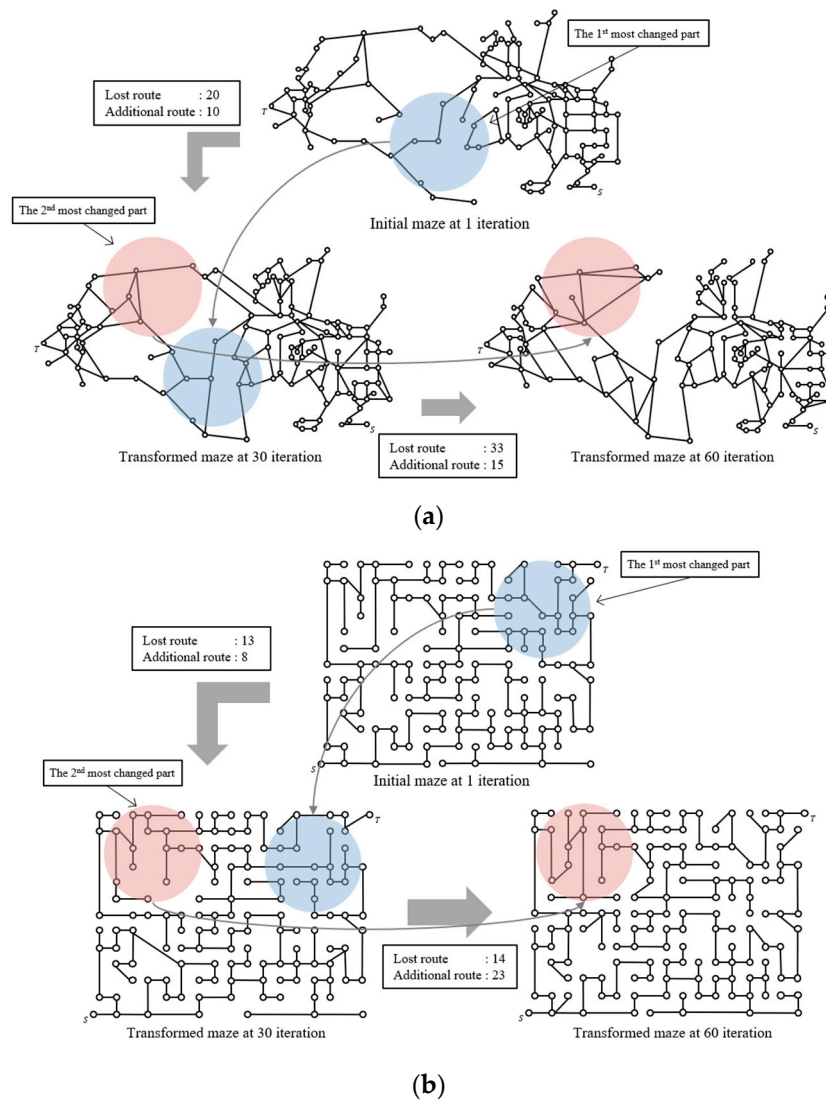


Figure 13. A second various network with two topological changes during the learning process. (a) Test network 1 with topological changes during the learning process; (b) Test network 2 with topological changes during the learning process.

Figure 14 shows the changes of the objective function value per iterations with the networks shown in Figure 13a.

As shown in Figure 14, CMPACO shows the best performances per each change of the network topology, compared to ACO and MPACO.

Table 12 compares each computation time for the final best routes using ACO, MPACO and CMPACO. As shown in Figure 14 and Table 12, these numerical studies show that the proposed CMPACO has better performances for network path generation considering dynamic network topology changes.

Table 12. Computation time comparisons among ACO, MPACO and CMPACO.

Test Networks	Computation Time for the Best Result		
	ACO	MPACO	CMPACO
Test network 1	15.82(s)	17.33(s)	10.45(s)
Test network 2	18.98(s)	20.55(s)	10.50(s)

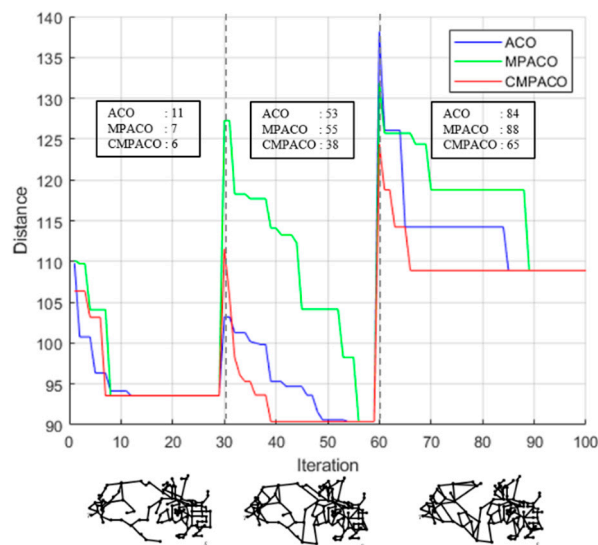


Figure 14. Comparisons of the cost function values using ACO, MPACO and CMPACO with the test network 1.

6. Conclusions and Further Studies

The path generation from the start node to the target node in a complex network is a representative problem that has been steadily investigated in the field of network optimization and machine learning. While most algorithms, including ACO, use a unidirectional search in a serial manner, the computation takes a long time as the complexity increases, with the drawback that it fails to search the route effectively in a dynamic environment.

In order to overcome these issues, the MPACO algorithm is proposed to search for an effective path by updating the pheromone through a Bayesian-type GPR process at the node where ant agents search multi-directionally. The numerical studies show the greater effectiveness of the MPACO algorithm than the existing ACO and the greater the complexity of the network, the more efficient it is. However, while the MPACO algorithm generates the route faster than the ACO algorithm in a static environment, the renewal process of the pheromone fails to recognize the neighboring nodes' topology changes in the network.

This issue is overcome using the proposed CMPACO framework, which updates the pheromone by interpolating discrete Gaussian distributions at the collision node. In order to verify the effectiveness of the proposed algorithm, it is compared to the ACO algorithm and the MPACO algorithm. The experimental results show that the presented CMPACO algorithm offers better performance than the ACO algorithm and MPACO algorithm, with respect to the computation speed and the optimized cost function.

The proposed framework can be applied to various network route problems with complex and dynamic topology changes and to real road networks. As further studies, more effective meta-heuristic algorithms and pheromone diffusion algorithms can be considered for much quicker path generation.

Author Contributions: E.O. and H.L. conceptualized the framework and developed the methodologies. E.O. implemented and validated the framework. H.L. supervised the overall research processes. E.O. wrote the manuscript. And H.L. reviewed and edited it.

Funding: This research was supported by The Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, S. Korea (grant number: NRF-2018R1D1A3B07047113).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dorigo, M.; Maniezzo, V.; Colorni, A. The ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41. [[CrossRef](#)] [[PubMed](#)]
2. Liu, Y.; Xiao, Y. Parallel solution of maze optimal path based on ant colony algorithm. In Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, Paris, France, 22–23 March 2013; pp. 1826–1829. [[CrossRef](#)]
3. Yan, Z.; Yuan, C.W. Ant Colony Optimization for Navigating Complex Labyrinths. In Proceedings of the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, Chongqing, China, 26–29 May 2003; pp. 445–448. [[CrossRef](#)]
4. Ilie, S.; Badica, C. Multi-agent approach to distributed ant colony optimization. *Sci. Comput. Program.* **2013**, *78*, 762–774. [[CrossRef](#)]
5. Yoshikawa, M.; Nagura, T. Adaptive Ant Colony Optimization Considering Intensification and Diversification. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 18–20 March 2009; pp. 200–203.
6. Yoshikawa, M.; Otani, K. Ant Colony Optimization Routing Algorithm with Tabu Search. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 17–19 March 2010; pp. 17–19.
7. Kim, J.; Lee, H. Multi-agent Reinforcement Learning based Evacuation Framework Considering Both Evacuation Time and Crowdedness. *J. Korean Inst. Intell. Syst.* **2016**, *26*, 335–342.
8. Dorigo, M.; Birattari, M. Ant Colony Optimization. In *Encyclopedia of Machine Learning*, 2010 ed.; Sammut, C., Webb, G.I., Eds.; Science+Business Media: Boston, MA, USA, 2010; pp. 37–40. [[CrossRef](#)]
9. Dorigo, M.; Caro, G.D. Ant colony optimization: A new metaheuristic. In Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 1999; pp. 1470–1477. [[CrossRef](#)]
10. Garcia, A.G.; Tria, L.A.R.; Talampas, M.C.R. Development of an Energy-Efficient Routing Algorithm for Electric Vehicles. In Proceedings of the IEEE Transportation Electrification Conference and Expo, Detroit, MI, USA, 19–21 June 2019; pp. 1–5. [[CrossRef](#)]
11. Quang, P.T.A.; Sanner, J.M.; Morin, C.; Aoul, Y.H. Multi-objective multi-constrained QoS Routing in large-scale networks: A genetic algorithm approach. In Proceedings of the International Conference on Smart Communications in Network Technologies, El Oued, Algeria, 27–31 October 2018; pp. 55–60. [[CrossRef](#)]
12. Yu, X.; Liu, Q.; Liu, Y.; Hu, M.; Zhang, K.; Xiao, R. Uneven clustering routing algorithm based on glowworm swarm optimization. *Ad Hoc Netw.* **2019**, *93*, 1–8. [[CrossRef](#)]
13. Xu, Y.; Wang, X.; Sun, T. Heuristic routing algorithm toward scalable distributed generalized assignment problem. *Soft Comput.* **2018**, *22*, 845–859. [[CrossRef](#)]
14. Qiu, S.; Zhong, Y.; Luo, X.; Liu, J.; Luo, Y.; Jiang, P. A Relay Routing Algorithm for Remote Concentrated Ammeter Reading Based on Ant Colony Optimization. In Proceedings of the International Conference on Systems Engineering, Sydney, Australia, 18–20 December 2018; pp. 1–8. [[CrossRef](#)]
15. Dorigo, M.; Caro, G.D.; Gambardella, L.M. Ant algorithms for discrete optimization. *J. Int. Soc. Artif. Life* **1999**, *5*, 137–172. [[CrossRef](#)]
16. Birattari, M.; Pellegrini, P.; Dorigo, M. On the Invariance of Ant Colony Optimization. *IEEE Trans. Evolut. Comput.* **2007**, *11*, 732–742. [[CrossRef](#)]
17. Dorigo, M.; Maniezzo, V.; Colorni, A.; Trubian, M. Ant systems for Job shop scheduling. *Belg. J. Oper. Res. Stat. Comput. Sci.* **1994**, *34*, 39–54.
18. Arnaout, J.; Musa, R.; Rabadi, G. Ant colony optimization algorithm to parallel machine scheduling problem with setups. In Proceedings of the IEEE Conference on Automation Science and Engineering, Arlington, VA, USA, 23–26 August 2008; pp. 578–582. [[CrossRef](#)]
19. Chen, R.; Lo, S.; Wu, C.; Lin, T. An effective ant colony optimization based algorithm for flow shop scheduling. In Proceedings of the IEEE Conference on Soft Computing in Industrial Applications, Muroran, Japan, 25–27 June 2008; pp. 25–27. [[CrossRef](#)]
20. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [[CrossRef](#)]
21. Oh, E.; Lee, H. Effective Routing Generation Framework using Multi-directional and Parallel Ant Colony Optimization. *J. Korean Inst. Intell. Syst.* **2018**, *28*, 523–530. [[CrossRef](#)]

22. Ahmadizar, F.; Barzinpour, F.; Arkat, J. Solving permutation flow shop sequencing using ant colony optimization. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 2–4 December 2007; pp. 753–757. [[CrossRef](#)]
23. Reza, G.A.; Mahfujur, R.; Abdur, R.; Wail, G.; Abdulmotaleb, E.S. Ant colony-based many-to-one sensory data routing in Wireless Sensor Networks. In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications, Doha, Qatar, 31 March–4 April 2008; pp. 1005–1010. [[CrossRef](#)]
24. Maniezzo, V.; Colorni, A. The ant system applied to the quadratic assignment problem. *IEEE Trans. Knowl. Data Eng.* **1999**, *11*, 769–778. [[CrossRef](#)]
25. Williams, C.K.I.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*; The MIT Press: London, UK, 2006; pp. 7–128.
26. Williams, C.K.I.; Rasmussen, C.E. Gaussian Processes for Regression. *Adv. Neural Process. Syst.* **1996**, *8*, 514–520.
27. Rasmussen, C.E. Gaussian Processes in Machine Learning. *Adv. Lect. Mach. Learn.* **2004**, *3176*, 63–71. [[CrossRef](#)]
28. Ak, C.; Ergonul, O.; Sencan, I.; Torunoglu, M.A.; Gonen, M. Spatiotemporal prediction of infectious diseases using structured Gaussian processes with application to Crimean-Congo hemorrhagic fever. *PLoS Negl. Trop. Dis.* **2018**, *12*, e0006737. [[CrossRef](#)] [[PubMed](#)]
29. Luttinen, J.; Ilin, A. Efficient Gaussian process inference for short-scale spatio-temporal modeling. In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 21–23 April 2012; pp. 741–750.
30. Nguyen, D.; Peters, J. Learning Robot Dynamics for Computed Torque Control using Local Gaussian Processes Regression. In Proceedings of the ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, Edinburgh, UK, 6–8 August 2008; pp. 59–64. [[CrossRef](#)]
31. Nguyen, L.; Hu, G.; Spanos, C.J. Spatio-temporal environmental monitoring for smart buildings. In Proceedings of the 13th IEEE International Conference on Control and Automation, Ohrid, Macedonia, 3–6 July 2017; pp. 277–282. [[CrossRef](#)]
32. Chen, N.; Qian, Z.; Meng, X.; Nabney, I.T. Short-term wind power forecasting using Gaussian processes. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; pp. 2790–2796.
33. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2017**, *85*, 1–16. [[CrossRef](#)]
34. Eklund, P.W.; Kirkby, S.; Pollitt, S. A Dynamic Multi-source Dijkstra’s Algorithm for Vehicle Routing. In Proceedings of the IEEE Australian and New Zealand Conference on Intelligent Information Systems, Adelaide, SA, Australia, 18–20 November 1996; pp. 329–333. [[CrossRef](#)]
35. Giralda, D.B.; Rodriguez, M.A.; Pernas, F.J.D.; Higuera, J.F.D.; Ortega, D.G.; Zarzuela, M.M. Intelligent system for dynamic transport fleet management. In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, Catania, Italy, 19–22 September 2005; pp. 773–776. [[CrossRef](#)]
36. Noto, M.; Sato, H. A Method for the Shortest Path Search by Extended Dijkstra Algorithm. In Proceedings of the International Conference on Systems, Man and Cybernetics, Nashville, TN, USA, 8–11 October 2000; pp. 2316–2320. [[CrossRef](#)]
37. Moss, F.H.; Segall, A. An Optimal Control Approach to Dynamic Routing in Network. *IEEE Trans. Autom. Control* **1982**, *27*, 329–339. [[CrossRef](#)]
38. Cauvery, N.K.; Viswanatha, K.V. Routing in Dynamic Network using Ants and Genetic Algorithm. *Int. J. Comput. Sci. Netw. Secur.* **2009**, *9*, 194–200.

