

IMPROVED ACO-BASED SWEEP COVERAGE SCHEME CONSIDERING DATA DELIVERY

Huang, P.*; Lin, F.**; Xu, L. J.*.#; Kang, Z. L.*; Zhou, J. L.** & Yu, J. S.***

* College of Mechanical and Electrical Engineering, Sichuan Agricultural University, Yaan,
P. R. China

** College of Computer Science, Sichuan University, Chengdu, P. R. China

*** Institute of Geological and Nuclear Sciences, 1 Fairway Drive, Lower Hutt, New Zealand

E-Mail: lijiaxu13@163.com (# Corresponding author)

Abstract

The newly introduced sweep coverage scheme uses mobile sensors to implement network coverage in wireless sensor networks (WSNs) and has attracted much attention from researchers. However, data buffer and moving speed of the mobile sensor are limited in sweep coverage. Thus, scheduling the minimum number of mobile sensors to efficiently implement dynamical network coverage while considering data delivery is still a challenging problem. To provide steady and efficient data gathering from sensors, an improved ant colony optimization-based sweep coverage (IACOSC) scheme supporting data delivery was proposed. In IACOSC, the artificial ants were used to create the initial coverage routes for points of interest. Then, a novel metric called route coverage efficiency was used to evaluate the routes. Finally, a local search algorithm based on route deletion and node insertion was employed to optimize these routes. Algorithm analysis shows that the time complexity of IACOSC is $O(n^3)$. Simulation results show that, compared with existing sweep coverage approaches considering data delivery, IACOSC significantly reduces the computational complexity and decreases the computation time by 50 % while reducing the mobile sensors by 16.73 % in the same network scenarios. The results obtained in this study can be applied to optimal deployment of WSN using the sweep coverage scheme.

(Received in November 2016, accepted in January 2017. This paper was with the authors 1 month for 1 revision.)

Key Words: Wireless Sensor Network (WSN), ACO, Mobile Sensor, Sweep Coverage

1. INTRODUCTION

In the last decade, wireless sensor networks (WSNs) have been adopted in various fields, such as tactical surveillance, intelligent environmental and structural monitoring, and target tracking [1]. Despite the broad utility of WSN, an important research problem in WSN is the coverage problem, which studies how well the field is monitored by sensors [2]. The traditional coverage problem in WSN can be generally divided into three categories: area coverage [3-6], barrier coverage [7-10], and point coverage [11-13]. The traditional coverage schemes require numerous stationary sensor nodes to provide coverage service. With the development of WSN, certain points of interest (POIs) need to be monitored periodically instead of continuously covering the entire area in some special applications, such as patrol inspection or forest fire prevention; the concept is called sweep coverage [14]. Unlike stationary deployed WSN, the sweep coverage scheme that uses mobile sensors to cover the POIs periodically in the network can improve the efficiency of the network coverage and significantly reduce the network deployment cost. This mobile sensor in sweep coverage can either be manned by a human or be an autonomous vehicle. Given that the POI is periodically covered, using the traditional coverage solution for sweep coverage scenario is not feasible. Li et al. [15] used the sweep coverage scheme to launch a project that aims at all-year-round ecological surveillance in a forest in Tianmu Mountain. They formally defined a min-sensor sweep-coverage problem that is proven to be NP-hard.

The sweep coverage problem is a difficult problem in combinatorial optimization and thus presents a great challenge. In practice, this problem contributes directly to reducing the cost and complexity of WSN. Most current studies focus on scheduling the mobile sensors to cover the POIs within a certain time but ignore data delivery. Meanwhile, finding a path from the mobile sensors to the sink nodes is difficult in some scenarios because of the limited node transmission range. The mobile sensors must visit the sink nodes to deliver the sensing data, and the data delivery can inevitably affect the performance of sweep coverage. Studying sweep coverage that considers data delivery has important practical value and application prospect.

2. STATE OF THE ART

The sweep coverage problem in WSN is still in its infancy stage. In the early 2000s, researchers have studied the sweep coverage problem of robotics in consideration of the metric of coverage frequency (i.e., the frequency of the coverage of each point) [16]. Since then, the sweep coverage problem has been a major issue in WSN. Li et al. [15] formally defined the sweep coverage problem of minimum number of required sensors in WSN (min-sensor sweep-coverage problem) and proved that the problem is NP-hard. This problem is equivalent to solving the traveling salesman problem (TSP). They proposed a centralized algorithm called centralized sweep (CSWEEP) algorithm and a distributed sweep (DSWEEP) algorithm. However, CSWEEP can only handle the POIs with the same coverage interval. DSWEEP may cause some mobile sensors to head for the same POI, thereby resulting in a waste of resource. CSWEEP and DSWEEP do not consider the problem of delivering data back to the sink. Many researchers conducted much work and improved the performance of sweep coverage on the basis of their research results [17-19]. Xi et al. [17] proposed a distributed algorithm based on a virtual 3D map of local gradient information to guide the movement of mobile nodes to achieve sweep coverage on dynamic POIs. However, this method cannot discover the relative direction for each neighbourhood and may guide all the mobile sensors to have a same path. A patrol point algorithm that can keep the patrol times of mobile node approximate to one another and can meet the detection requirements of a potential fire flashover situation was proposed in [18]. However, this algorithm fails to achieve sweep coverage of the given region with optimal operation time. Du et al. [19] proposed two algorithms, namely, minimal expand (MinExpand) algorithm and on sweep coverage with minimum mobile sensor (OSweep) algorithm, under two scenarios. When the mobile sensors were restricted to follow the same trajectory in different sweep periods, the centralized algorithm MinExpand was proposed to schedule the scan path. When the mobile sensors were not restricted to follow the same trajectory in different sweep periods, OSweep was proposed; all the mobile sensors in the algorithm are scheduled to move along a TSP ring composed of POIs. However, the abovementioned studies using centralized mobile sensor control only focused on controlling the sensor coverage of the POI at a given time while ignoring the data delivery that sends the data back to the sink node. The distributed scheme fails to achieve 100 % sweep coverage of a given region at a given time.

With the development of the sweep coverage technique, some scholars attempted to solve the sweep coverage problem while considering data delivery. Wang and Ma [20] focused on the problem of data collection in WSNs with minimum mobile nodes. They proposed a heuristics path planning algorithm called loop growth (LoopGrowth), which can reduce the number of required mobile nodes by as much as 55.6 %. However, LoopGrowth can only handle the POIs that have the same coverage interval. Lin et al. [21] studied sweep coverage while considering data delivery. They proposed a two-tier architecture called mobile-assisted sweep coverage by use of mobile-assisted carriers to help data delivery. However, the

structure of the network is complicated. Shu et al. [22] analysed the problem of minimum number of required sensors in sweep coverage as a variation of the vehicle routing problem (VRP) and formularize the said problem in sweep coverage by improving the Fisher and Jaikumar formula for VRP. They proposed a novel sweep coverage scheme called VRP-based sweep coverage (VRPSC), which supports dynamical POI coverage and data delivery simultaneously. Simulation results show that VRPSC achieves better performance than CSWEEP but lack comparison with other algorithms. Huang et al. [23] tested the feasibility of ant colony optimization (ACO) [24] for sweep coverage and proposed the ACO-based sweep coverage (ACOSC) algorithm to solve the sweep coverage problem while considering data delivery. ACOSC uses a two-point-crossover operation in the local search. The performance of ACOSC always outperforms VRPSC. However, ACOSC uses an approximate exhaustive local search algorithm and has high time complexity.

In summary, the current techniques have been used in exploring the sweep coverage problem in WSN. Researchers investigated the scheduling of mobile sensors to scan POIs within a given time. However, as discussed above, efficient scheduling of mobile sensor while considering data delivery is still a challenging problem.

This study aims to provide steady data collection and decreased computational complexity. For this purpose, an improved ACO-based sweep coverage (IACOSC) scheme for solving the sweep coverage problem considering dynamical POI coverage and data delivery simultaneously is proposed. The main task in designing IACOSC is to schedule the path of mobile sensors to satisfy the coverage requirement of POIs and deliver the data from POIs to sink while considering data delivery.

The rest of the study is organized as follows. Section 3 presents the mathematical model of the problem of minimum number of required sensors considering data delivery in sweep coverage and describes IACOSC. Then, Section 4 presents four simulations for evaluating the performance of IACOSC. Finally, Section 5 elaborates the conclusions of the study.

3. METHODOLOGY

3.1 Mathematical model of sweep coverage considering data delivery

In [22], the problem of minimum number of required sensors in sweep coverage is defined as shown in Fig. 1. In the figure, a fleet of mobile sensors of uniform buffer is scheduled to cover a given set of N POIs, p_i . Each POI is characterized by a data generation rate of q_i , a coverage requirement T_i , and a sense data T_{sense} . Assuming K mobile sensor nodes are assigned into M routes in the solution to access all the POIs. Each route originates and ends at the sink node with constant stay time interval T_{trans} to deliver data. Each coverage of POI can be done within the coverage requirement T_i . The total data collected in each route cannot exceed the mobile buffer Q . A solution ψ of sweep coverage is represented as $\psi = \{(k_m, R_m) \mid m = 1, 2, \dots, M\}$, where k_m ($m = 1, 2, \dots, M$) is the mobile sensor assigned to the m^{th} route in the solution and $R_m = (p_0, p_1, \dots, p_0)$ is the sequence of POIs in a route.

Mathematically, this problem with N POIs is defined over the weighted graph $G = (P, A, d)$, where the vertices are represented by $P = \{p_0, p_1, \dots, p_N\}$, and the arcs are represented by $A = \{(p_i, p_j), i \neq j\}$. A sink node where each mobile sensor starts its route is located at p_0 , and each of the other vertices represents the N POIs. The distances associated with each arc are represented by the variable d_{ij} , which is measured by Euclidean computations. Once a mobile node accesses certain POI, it collects q_i bytes of data. Let Φ_m denote the set of POIs on the m^{th} route and k_m denote the number of mobile sensor nodes that are assigned for the m^{th} route.

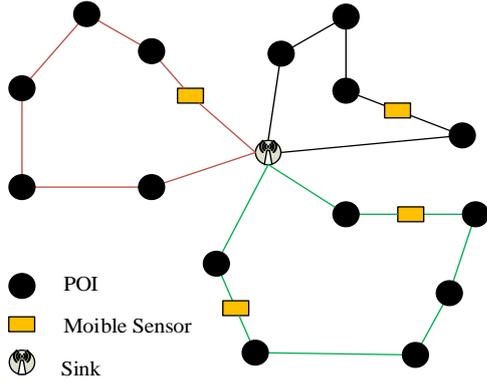


Figure 1: Node Trajectory in Sweep Coverage.

Defining $x_{i,j,k} = \begin{cases} 1, & \text{if mobile sensor } k \text{ moves from } p_i \text{ to } p_j \\ 0, & \text{otherwise} \end{cases}$, then the minimum number

of required sensors problem in sweep coverage considering data delivery can be formulated as

$$\text{Min } K = \sum_{m=1}^M k_m \quad (1)$$

$$\text{s.t. } \sum_{m=1}^M |\Phi_m| = N \quad (2)$$

$$\sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^K x_{i,j,k} = k_m \quad i, j \in \Phi_m \quad (3)$$

$$\sum_{j=0}^N \sum_{k=1}^K x_{0,j,k} = K \quad (4)$$

$$\sum_{i=0}^N \sum_{k=1}^K x_{i,0,k} = K \quad (5)$$

$$\sum_{i=0}^N \sum_{j=0}^N (x_{i,j,k} * q_i) \leq Q \quad (6)$$

$$l_m / (k_m V) + |\Phi_m| T_{\text{sense}} + T_{\text{trans}} \leq \text{Min}(T_i) \quad i \in \Phi_m \quad (7)$$

Eq. (1) represents the optimization objective. Eqs. (2) and (3) ensure that all the POIs are covered by route only once. Eqs. (4) and (5) ensure that all the routes start and terminate with the sink node. Eq. (6) represents the constraint on buffer size of mobile nodes. Eq. (7) represents the coverage requirement constraint.

3.2 IACOSC

In IACOSC, the computational ants are independent agents that communicate exclusively through pheromone trails similar to real ants. The procedure of IACOSC includes initialization, route construction, local heuristic search and pheromone updating. The remainder of the section describes the algorithm in detail.

1) Initialization

The algorithm starts with an initial solution, denoted by ψ_0 . The nearest neighbour [25] constructive heuristics is implemented to generate the initial solution. ψ_0 is also taken as the best solution generated by the heuristics initially.

2) Route construction

In IACOSC, an individual ant simulates an agent and its route is constructed by incrementally selecting POIs until all POIs are covered. Each ant starts at the sink node and the set of POIs included in its route is empty. The ant selects the next POI to cover from the list of feasible locations, and the storage capacity of the mobile is updated prior to selecting

another POI. The ant returns to the sink when the capacity constraint of the mobile sensor is met or the route cannot meet the coverage interval of the POIs in the route or when all customers are visited. For ant k , the probability of covering POI j after POI i (i.e., the last covered POI) uses the following probabilistic formula:

$$P_{ij}^k = \begin{cases} \arg \max_{j \in allowed_k} \{[\tau_{ij}^\alpha \cdot \eta_{ij}^\beta]\} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (8)$$

$$S = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{s \in allowed_k} \tau_{is}^\alpha \cdot \eta_{is}^\beta}, & s \in allowed_k, \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\eta_{ij} = K_1^{\Delta sensorium} \quad (10)$$

$$\Delta sensorium = newsensorium - oldsensornum \quad (11)$$

τ_{ij} is the amount of pheromone on the arc connecting POI i and POI j . τ_{ij} is the merit of arc (i, j) (route segment). The pheromones of the all arcs are initiated with a constant. K_1 is a constant. $oldsensornum$ and $newsensornum$ are mobile sensors before and after inserting POI j to the current route, respectively. Supposing (k_m, R_m) represents the m^{th} route of solution ψ that the ant is constructing, where $R_m = (p_0, p_k, \dots, p_i, p_0)$. If the ant inserts POI p_j into this route, then the route becomes (k_m', R_m') , where $R_m' = (p_0, p_k, \dots, p_i, p_j, p_0)$. The need for mobile sensor numbers in this route should be recalculated. The k_m represents the *oldsensornum* and the k_m' represents the *newsensornum*. Then, η_{ij} is a function of the changed sensor number and denotes the visibility on the arc (p_i, p_j) . The computation of η_{ij} considers the sweep time constraint and distance for the next chosen POI j . α and β are the weight parameters that denote the relative influence of the pheromone trails and the visibility values, respectively. $allowed_k$ contains feasible POIs to extend the current route of ant k . q_0 is a constant ($0 \leq q_0 \leq 1$). Each time, the ant selects a random number distributed in $[0, 1]$ for q . If $q \leq q_0$, then the pheromone of arc (p_i, p_j) is multiplied by the visibility η_{ij} . As a result, the ant chooses the edges that are shorter and with more pheromone than the others. Otherwise, the probability S is calculated. Then, the roulette wheel selection is used in deciding which candidate to select.

If the buffer size of the mobile sensor is full with collected data in constructing the route, then the mobile sensor must return to the sink and start to build a new route.

3) Local heuristic search algorithm

A local heuristic search algorithm is proposed in improving the performance of ACO. The local heuristic search procedure is considered a strong intensification process and can bring each solution computed by the artificial ants to its local optimal. The route metrics and the coverage efficiency are introduced first and then the local heuristic search algorithm is discussed.

(1) Route metrics

In sweep coverage, the optimization object is to minimize the number of mobile sensors and cover large number of POIs. If a route has a few POIs and many mobile sensors, then the route is a “low-quality” route. Such “low-quality” route should be eliminated or improved to enhance the solution built by ants. Therefore, coverage efficiency that is defined as the POI number divided by the sensor number in one route is introduced to indicate the quality of the routes. Route coverage efficiency is defined as

$$\alpha_i = \frac{P_i}{N_i} \quad (12)$$

where P_i is the POIs in the i^{th} route. N_i is the mobile sensors needed by the i^{th} route. α_i is the coverage efficiency of the i^{th} route.

(2) Local heuristic search algorithm

Once the ant has finished route construction and the solution S_0 is obtained, the local heuristic search algorithm is implemented. The local heuristic search algorithm is composed of three phases: computing, deleting, and reinsertion phases. In the computing phase, the coverage efficiency is computed for each route and the average coverage efficiency of all routes is calculated. In the deleting phase, the routes with coverage efficiency lower than the average coverage efficiency are placed into a set C and deleted from the original solution. In the reinsertion phase, a route c_i is randomly selected in C , the route is deleted from C , and a POI insertion algorithm is used to insert the POIs in c_i to the current solution. When the POI is inserted to the solution S_0 , two measurements are introduced to select a POI p_u in c_i to insert the adjacent POI p_i and p_j in the current solution.

$$\zeta_1 = \min(d_{i,u} + d_{u,j} - d_{i,j}) \tag{13}$$

$$\zeta_2 = \min(\zeta_1(u)) \tag{14}$$

The measurement ζ_1 denotes the minimum increment for inserting the POI p_u , which can be used to select the best insertion place. The measurement ζ_2 denotes the particular POI p_u , which costs the minimum increment for the current solution after insertion.

An example for the route improvement algorithm is shown in Fig. 2. The original solution and coverage efficiency of each route are shown in Fig. 2 a. The average coverage efficiency of four routes is calculated as 2. The route with coverage efficiency below the average coverage efficiency is deleted from the original solution, as shown in Fig. 2 b. In Fig. 2 c, the POIs in the deleted route are reinserted to the solution and a new solution S' is constructed. The new solution contains four mobile sensors and it is better than the original.

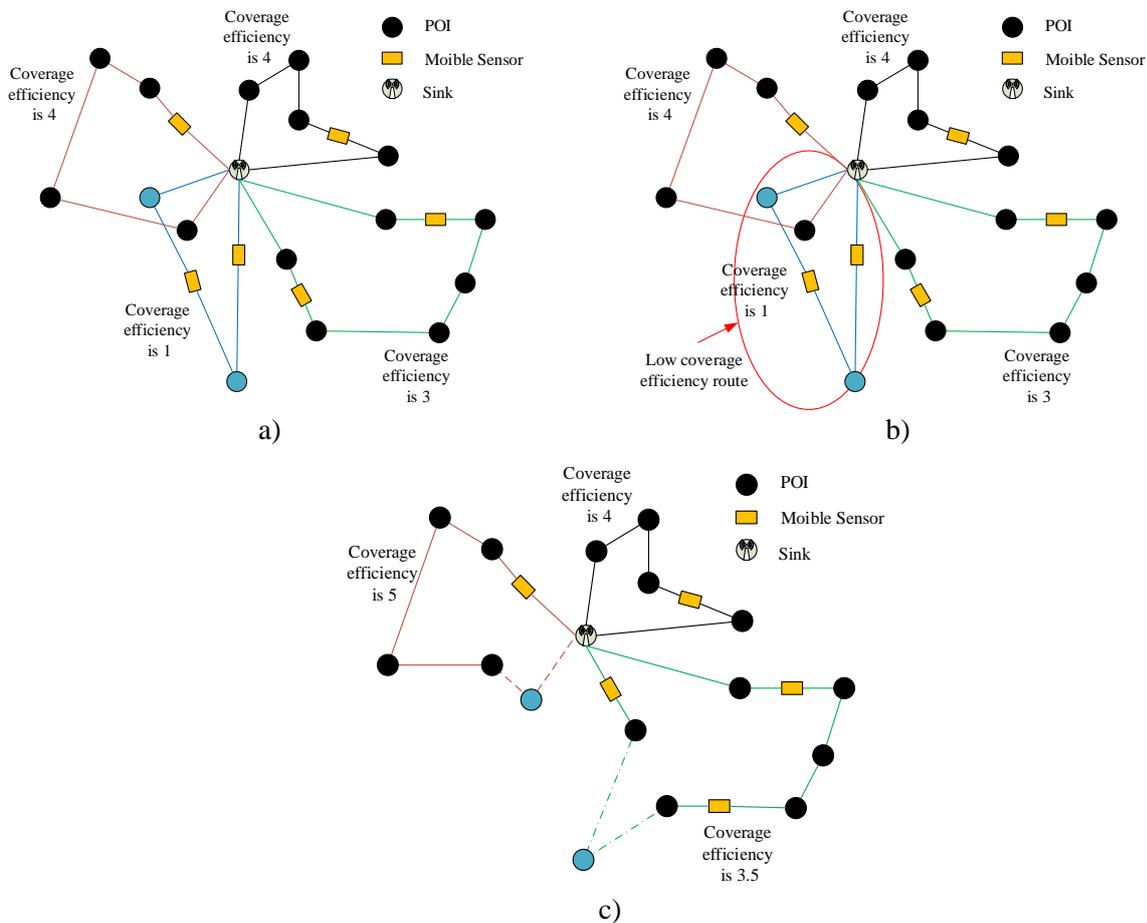


Figure 2: Route improvement algorithm.

4) Best solution updating strategy

After implementing the local heuristic search algorithm, a new solution is obtained. This new solution ψ_i in the i^{th} iteration may have the same mobile sensor number as the global optimal solution but shorter total route length. If a route has shorter length than that of other routes, then a large number of POIs are potentially inserted in this route. Therefore, if a solution has the same mobile sensor number as the global optimal solution but shorter total route length, then the possibility of decreasing the mobile sensor number in successive iterations is high. To increase the diversity of the solution, a best solution updating strategy is design as follows:

- (1) If the mobile sensor number in an iteration is smaller than that of the global optimal solution, then the iteration solution is substituted for the global optimal solution.
- (2) If the mobile sensor number in an iteration is equal to the global optimal solution, then the possibility of updating the global solution by the iteration solution is defined as

$$p_i = \exp(-K_2 \cdot len / \Delta T) \quad (15)$$

$$len = L_{\psi_g} - L_{\psi_i} \quad (16)$$

$$\Delta T = T_{total} - T_i \quad (17)$$

where L_{ψ_g} denotes the total length of the global optimal solution. L_{ψ_i} denotes the total length of the i^{th} iteration. T_{total} is the total iteration time and T_i is the i^{th} iteration time. K_2 is a constant. Eq. (15) shows that, when the difference in the total length between L_{ψ_g} and L_{ψ_i} is significant, the possibility of updating the global solution is low; when the difference in the total iteration time and the i^{th} iteration time is significant, the possibility of updating the global solution is low as well.

5) Pheromone information updating

(1) Local updating rule

Once an ant has gone through solution construction and local search, the local pheromone update procedure is applied. The local pheromone update rule can be written as

$$\tau_{ij}^{new} = (1 - \rho) \cdot \tau_{ij}^{old} + \rho \cdot \Delta \tau_{ij} \quad (18)$$

$$\Delta \tau_{ij} = \begin{cases} \frac{Q}{Num_{current\ solution}}, & \text{arc}(i, j) \text{ belongs to current solution} \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

$$Q = K_2 \cdot Num_{POI} \quad (20)$$

where τ_{ij} is the pheromone of arc (i, j) and $0 < \rho < 1$ is a parameter. Q is a parameter related to the number of POI numbers in sweep coverage problem and K_2 is a constant. $Num_{current\ solution}$ is the mobile sensor number in the current solution built by ants.

(2) Global updating rule

At the end of each iteration (i.e., once all ants have gone through solution construction and local search), the global pheromone update procedure is applied. The global pheromone update rule can be written as

$$\tau_{ij}^{new} = (1 - \gamma) \cdot \tau_{ij}^{old} + \gamma \cdot \frac{Q}{Num_{gb}} \quad (21)$$

where γ is the pheromone decay parameter $0 < \gamma < 1$, and Num_{gb} is the mobile sensor number of the globally best solution.

6) Stopping rules

The IACOSC procedure stops when no improvement is observed on the solution after several iterations or when the maximum number of iterations is reached.

4. SIMULATION RESULT

4.1 Algorithm analysis

Theorem 1. The time complexity of IACOSC is no more than $O(n^3)$, where n is the number of POIs.

Proof. Assuming m ants exist. The time complexity consists of five parts: the initial time, route construction, local search, local pheromone update, and global pheromone update.

(1) The initial time complexity is $O(n^2)$.

(2) The time complexity of route construction is $O(n^2 \cdot m)$.

(3) The time complexity of local search.

Assuming that M routes exist. The route T_i with the minimum route coverage efficiency is deleted. T_i has the POI number N_i . According to Eqs. (13) and (14), the time complexity of inserting a POI is $\sum_{k=1, k \neq i}^{M-1} (N_k + 1)$. The time complexity of inserting N_i nodes is

$N_i \cdot \sum_{k=1, k \neq i}^{M-1} (N_k + 1)$. In the worst case, the time complexity of the local search is $O(n^3)$.

(4) The time complexity of local pheromone updating is $O(m \cdot n)$.

(5) The time complexity of global pheromone updating is $O(n^2)$.

The time complexity of completing an iteration is:

$$O(n^2 + m) + O(n^2 \cdot m) + O(n^3) + O(m \cdot n) + O(n^2).$$

In the worst case, the time complexity of IACOSC is $O(n^3)$.

4.2 Simulation and performance evaluation

This section presents the simulation results for evaluating the proposed algorithm and compares the algorithm with VRPSC [22] and ACOSC [23]. The performance of these solutions is evaluated in different scenarios.

The POI number = 50, 60, 70, ..., 200, are randomly deployed in a 500 m × 500 m area. The required coverage time interval of each POI is randomly distributed in 100 s – 500 s. The mobile sensors stay $T_{sense} = 20$ s at any POI for data sensing and stay $T_{trans} = 20$ s at the sink for data transmission. The default moving velocity of mobile sensors is set to 1, 3, and 7 m/s, similar to those in literature [22].

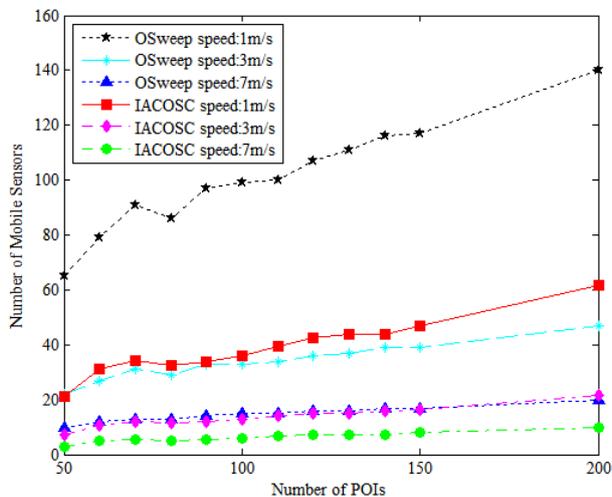


Figure 3: Comparison of OSweep and IACOSC.

Fig. 3 shows the comparison of OSweep and IACOSC. Given that OSweep does not consider the buffer size of mobile sensor and the data delivery, the buffer size of IACOSC is set sufficiently large to enable the comparison. When the velocity of mobile nodes is equal, OSweep requires a larger number of mobile sensors to perform coverage task than IACOSC. In addition, when the velocity of the mobile sensors is high, a small number of mobile sensors are required. The reason is that the sensors cannot cover large number of POIs in a certain route when the mobile sensors have low speed. Therefore, IACOSC has to create new routes with additional mobile sensors to complete the coverage.

In Figs. 4 a–f, the buffer size of the mobile sensors is set to 210, 840, and 3920 bytes. The velocity of the mobile sensors in each scenario is set to 1, 3, and 7 m/s.

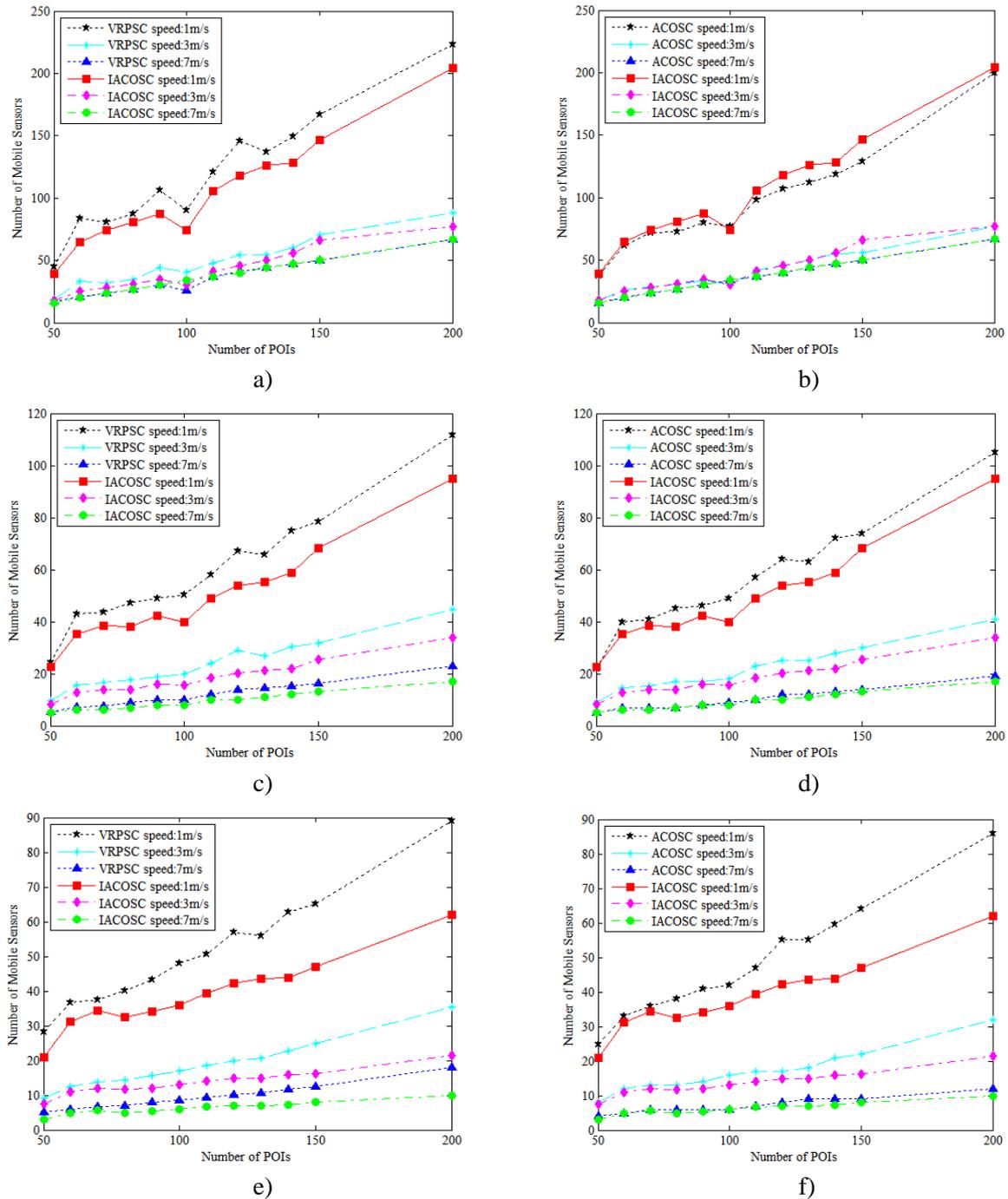


Figure 4: Performance comparison of VRPSC, ACOSC, and IACOSC.

Figs. 4 a and b show that the buffer size of the mobile sensors is fixed at 210 bytes and that the velocity of the mobile sensor is set to 1, 3, and 7 m/s. The figures also show that IACOSC has better performance than VRPSC and ACOSC. The number of required mobile sensor sensors with low speed is higher than that of the mobile sensors with high speed. This condition is due to that the sensor cannot cover large number of POIs in a certain route when the mobile sensors have low speed. When the speed is 7 m/s, the performance of the three algorithms is nearly the same. The reason is that the buffer constraint is the main one in route construction and each route can only cover a limited number of POIs when the buffer size is small.

In Figs. 4 c and d, the buffer size of the mobile sensors is fixed at 840 bytes, and the velocity of the mobile sensor is set to 1, 3 and 7 m/s. The figures also show that the number of required mobile sensors of VRPSC and ACOSC grows much faster than that of IACOSC.

Figs. 4 e and f show that the buffer size of the mobile sensors is fixed at 3920 bytes and that the velocity of the mobile sensor is set to 1, 3 and 7 m/s. The figures also show that the number of required mobile sensors of VRPSC and ACOSC grows much faster than that of IACOSC. The number of required mobile sensors with a velocity of 1 m/s is higher than that of the mobile sensors with a velocity of 7 m/s. This condition is attributed to that the sensors cannot cover large number of POIs in a certain route when the mobile sensors have less velocity but enough buffer size. Therefore, the velocity constraint is the main one in route construction. In summary, IACOSC reduces the mobile sensors by 16.73 % compared with the state-of-the-art methods.

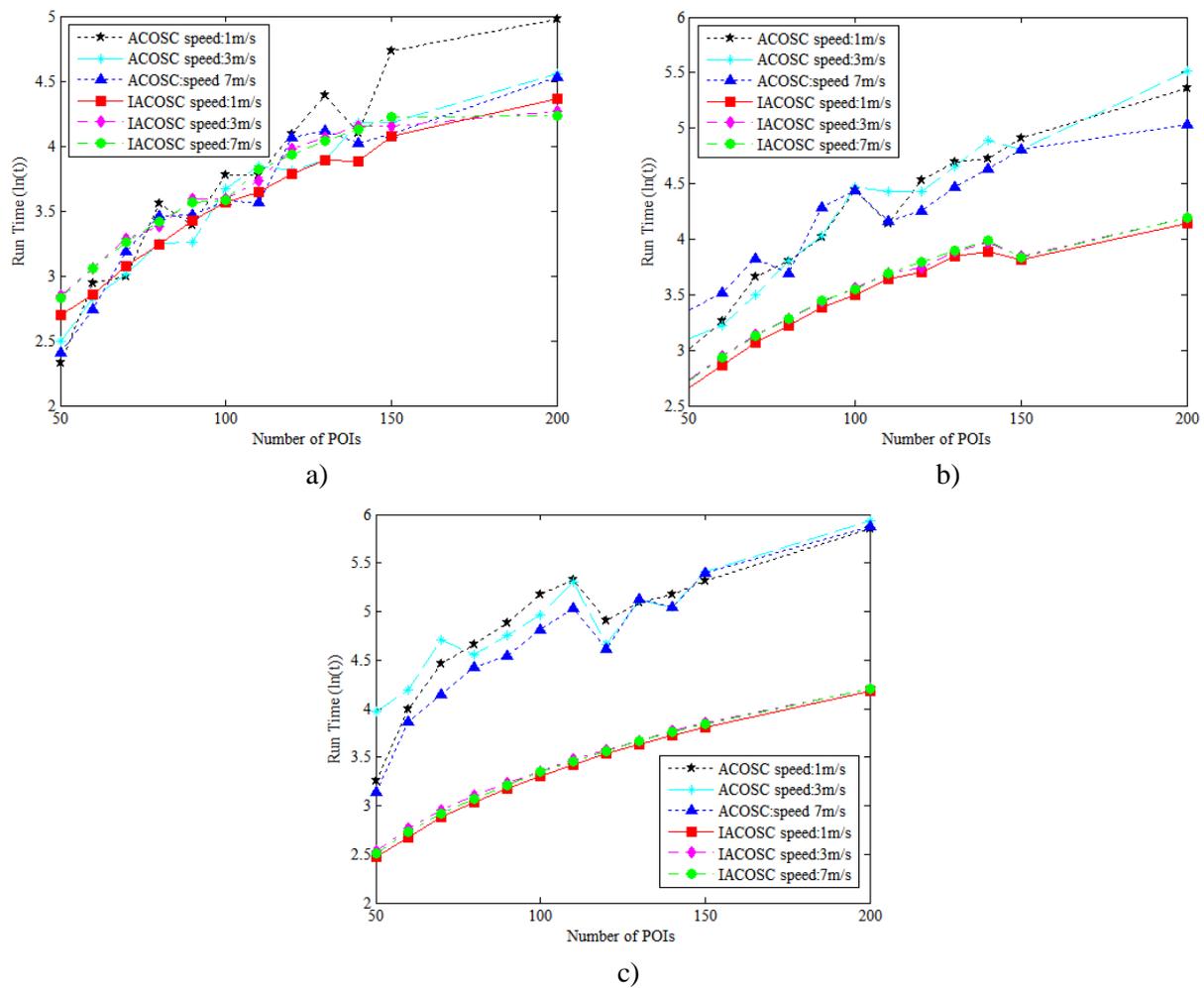


Figure 5: Calculation time of ACOSC and IACOSC.

Fig. 5 shows the comparison of calculation times between ACOSC and IACOSC. In Fig. 5 a, the two algorithms present a slight difference in their calculating times, which increase with the number of POIs. This result is due to that each route can only accommodate three POIs and the scale of a two-point-crossover in ACOSC is small in this scenario. When the number of POIs is larger than 150, ACOSC runs longer than IACOSC.

Fig. 5 b shows that the buffer constraint is moderate for moving sensor nodes. The change in the velocity of the moving sensor plays a major role in influencing the algorithm operation. ACOSC runs longer than IACOSC under three speed conditions because the time complexity of ACOSC $O(n^4)$ is higher than that of IACOSC $O(n^3)$. The experimental results confirm the theoretical analysis. As the moving speed increases, the running time of the algorithm increases because large number of POIs can be covered while the speed of moving the sensor nodes increases. In ACOSC, a large number of candidate POIs are covered when the ants build a route. When the solution is completely built, the number of POIs that can be covered by each route increases accordingly, thereby resulting in the increase in the route length of the two-point crossover. Similar to ACOSC, much computation is required to construct a route in IACOSC. In the local search algorithm, the location of route insertion increases when the routes with low coverage efficiency are deleted. This condition requires considerable computation time.

Fig. 5 c similarly shows that ACOSC runs longer than IACOSC under three speed conditions. The computation time of IACOSC changes slower than that of ACOSC. This difference is due to that the two-point crossover operation needs much computation time when a route has large number of POIs, and vice versa. The computation time of IACOSC increases linearly with the number of POIs.

5. CONCLUSION

Aiming at the sweep coverage problem that is integrated with data delivery, the sink node in WSN was treated as a special POI and a novel solution based on ACO called IACOSC was proposed. Through theoretical analysis and simulation, the following conclusions are obtained:

(1) The buffer size and speed of mobile sensors significantly influence the performance of IACOSC. If the buffer size of mobile sensors is set to infinity, then the number of mobile sensors is inversely proportional to speed. The number of mobile sensors reaches a minimal value when the buffer size is set to infinity. However, if the speed is determined, then simply increasing the buffer size of the mobile sensor cannot significantly reduce the number of mobile sensors.

(2) The local search algorithm plays an important role in reducing the computation time and improving the performance of ACO. The largest difference between ACOSC and IACOSC is the local search algorithm. The time complexity of the local search algorithm of ACOSC based on the two-point-crossover operation is higher than that of IACOSC. Thus, the results from the algorithm analysis show that the total time complexity of IACOSC is significantly lower than that of ACOSC.

In this study, the sweep coverage scheme considering data delivery is more approximate in practical application than the original one. However, the practical application of sweep coverage is influenced by many factors, and the proposed method is limited to considering dynamical network coverage and data delivery. Additional influencing factors, such as energy of mobile sensors, constraint of data transmission delay, and incomplete connection of POIs should be considered in future studies.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (no. 61272448), Doctoral Fund of Ministry of Education of China (no. 20110181130007), Science and Technology Supporting Plan of Sichuan (no. 2012GZ0005), Key Research Foundation of Education Bureau of Sichuan Province (no. 16ZA0032).

REFERENCES

- [1] Chowdhury, T. J. S.; Elkin, C.; Devabhaktuni, V.; Rawat, D. B.; Oluoch, J. (2016). Advances on localization techniques for wireless sensor networks: A survey, *Computer Networks*, Vol. 110, 284-305, doi:[10.1016/j.comnet.2016.10.006](https://doi.org/10.1016/j.comnet.2016.10.006)
- [2] Derr, K.; Manic, M. (2013). Wireless sensor network configuration – Part II: Adaptive coverage for decentralized algorithms, *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 3, 1728-1738, doi:[10.1109/TII.2013.2245907](https://doi.org/10.1109/TII.2013.2245907)
- [3] Ammari, H. M.; Das, S. K. (2010). A study of k-coverage and measures of connectivity in 3D wireless sensor networks, *IEEE Transactions on Computers*, Vol. 59, No. 2, 243-257, doi:[10.1109/TC.2009.166](https://doi.org/10.1109/TC.2009.166)
- [4] Li, J.-S.; Kao, H.-C. (2010). Distributed K-coverage self-location estimation scheme based on Voronoi diagram, *IET Communications*, Vol. 4, No. 2, 167-177, doi:[10.1049/iet-com.2009.0288](https://doi.org/10.1049/iet-com.2009.0288)
- [5] Ammari, H. M.; Das, S. K. (2012). Centralized and clustered k-coverage protocols for wireless sensor networks, *IEEE Transactions on Computers*, Vol. 61, No. 1, 118-133, doi:[10.1109/TC.2011.82](https://doi.org/10.1109/TC.2011.82)
- [6] Mao, X. F.; Liu, Y. H.; Tang, S. J.; Liu, H. F.; Han, J. K.; Li, X.-Y. (2013). Finding best and worst k-coverage paths in multihop wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 12, 2396-2406, doi:[10.1109/TPDS.2012.329](https://doi.org/10.1109/TPDS.2012.329)
- [7] Zhang, Y.; Sun, X.; Wang, B. W. (2016). Efficient algorithm for k-barrier coverage based on integer linear programming, *China Communications*, Vol. 13, No. 7, 16-23, doi:[10.1109/CC.2016.7559071](https://doi.org/10.1109/CC.2016.7559071)
- [8] Du, J. Z.; Wang, K.; Liu, H.; Guo, D. K. (2013). Maximizing the lifetime of k-discrete barrier coverage using mobile sensors, *IEEE Sensors Journal*, Vol. 13, No. 12, 4690-4701, doi:[10.1109/JSEN.2013.2270555](https://doi.org/10.1109/JSEN.2013.2270555)
- [9] He, S. B.; Gong, X. W.; Zhang, J. S.; Chen, J. M.; Sun, Y. X. (2014). Curve-based deployment for barrier coverage in wireless sensor networks, *IEEE Transactions on Wireless Communications*, Vol. 13, No. 2, 724-735, doi:[10.1109/TWC.2013.121813.130198](https://doi.org/10.1109/TWC.2013.121813.130198)
- [10] Wang, B.; Chen, J. Y.; Liu, W. Y.; Yang, L. T. (2016). Minimum cost placement of bistatic radar sensors for belt barrier coverage, *IEEE Transactions on Computers*, Vol. 65, No. 2, 577-588, doi:[10.1109/TC.2015.2423679](https://doi.org/10.1109/TC.2015.2423679)
- [11] Boudaren, M. E. Y.; Senouci, M. R.; Senouci, M. A.; Mellouk, A. (2014). New trends in sensor coverage modeling and related techniques: A brief synthesis, *Proceedings of the 2014 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, 6 pages, doi:[10.1109/SaCoNeT.2014.6867778](https://doi.org/10.1109/SaCoNeT.2014.6867778)
- [12] Yu, X. Y.; Huang, W. P.; Lan, J. J.; Qian, X. (2012). A novel virtual force approach for node deployment in wireless sensor network, *Proceedings of the 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, 359-363, doi:[10.1109/DCOSS.2012.32](https://doi.org/10.1109/DCOSS.2012.32)
- [13] Yang, S. H.; Dai, F.; Cardei, M.; Wu, J. (2005). On multiple point coverage in wireless sensor networks, *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, 757-764, doi:[10.1109/MAHSS.2005.1542868](https://doi.org/10.1109/MAHSS.2005.1542868)
- [14] Cheng, W. F.; Li, M.; Liu, K. B.; Liu, Y. H.; Li, X. Y.; Liao, X. K. (2008). Sweep coverage with mobile sensors, *Proceedings of the 22nd IEEE International Symposium on Parallel and Distributed Processing*, 9 pages, doi:[10.1109/IPDPS.2008.4536245](https://doi.org/10.1109/IPDPS.2008.4536245)
- [15] Li, M.; Cheng, W. F.; Liu, K. B.; He, Y.; Li, X. Y.; Liao, X. K. (2011). Sweep coverage with mobile sensors, *IEEE Transactions on Mobile Computing*, Vol. 10, No. 11, 1534-1545, doi:[10.1109/TMC.2010.237](https://doi.org/10.1109/TMC.2010.237)

- [16] Batalin, M. A.; Sukhatme, G. S. (2003). *Multi-robot dynamic coverage of a planar bounded environment*, Technical Report, Robotic Embedded Systems Laboratory, University of Southern California, Los Angeles
- [17] Xi, M.; Wu, K.; Qi, Y.; Zhao, J. Z.; Liu, Y. H.; Li, M. (2009). Run to potential: Sweep coverage in wireless sensor networks, *Proceedings of the 38th International Conference on Parallel Processing*, 50-57, doi:[10.1109/ICPP.2009.43](https://doi.org/10.1109/ICPP.2009.43)
- [18] Chu, H. C.; Wang, W.-K.; Lai, Y.-H. (2010). Sweep coverage mechanism for wireless sensor networks with approximate patrol times, *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing and 7th International Conference on Autonomic and Trusted Computing*, 82-87, doi:[10.1109/UIC-ATC.2010.60](https://doi.org/10.1109/UIC-ATC.2010.60)
- [19] Du, J. Z.; Li, Y. W.; Liu, H.; Sha, K. W. (2010). On sweep coverage with minimum mobile sensors, *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems*, 283-290, doi:[10.1109/ICPADS.2010.109](https://doi.org/10.1109/ICPADS.2010.109)
- [20] Wang, C.; Ma, H. D. (2011). Data collection in wireless sensor networks by utilizing multiple mobile nodes, *Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Networks*, 83-90, doi:[10.1109/MSN.2011.32](https://doi.org/10.1109/MSN.2011.32)
- [21] Lin, F.; Wang, W.; Zhou, J.-L. (2010). MASC: A sweep coverage approach with mobile-assisted carriers, *Journal of Sichuan University (Engineering Science Edition)*, Vol. 42, No. 6, 119-125+132, doi:[10.15961/j.jsuese.2010.06.029](https://doi.org/10.15961/j.jsuese.2010.06.029)
- [22] Shu, L.; Wang, W.; Lin, F.; Liu, Z. H.; Zhou, J. L. (2013). A sweep coverage scheme based on vehicle routing problem, *Telkomnika*, Vol. 11, No. 4, 2029-2036, doi:[10.11591/telkomnika.v11i4.2380](https://doi.org/10.11591/telkomnika.v11i4.2380)
- [23] Huang, P.; Lin, F.; Liu, C.; Gao, J.; Zhou, J.-L. (2015). ACO-based sweep coverage scheme in wireless sensor networks, *Journal of Sensors*, Vol. 2015, Paper ID 484902, 6 pages, doi:[10.1155/2015/484902](https://doi.org/10.1155/2015/484902)
- [24] Sicilia-Montalvo, J. A.; Royo-Agustin, B.; Quemada-Mayoral, C.; Oliveros Colay, M. J.; Larrode-Pellicer, E. (2015). A decision support system to long haul freight transportation by means of ant colony optimization, *DYNA*, Vol. 90, No. 1, 105-113, doi:[10.6036/7190](https://doi.org/10.6036/7190)
- [25] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations Research*, Vol. 35, No. 2, 254-265, doi:[10.1287/opre.35.2.254](https://doi.org/10.1287/opre.35.2.254)