HeilArchive : Pages:1–20

Original Research

Distributed Firefly-Based Sensor Placement for Observability Enhancement in Power Transmission Networks

Phạm Quang Hiếu[1] and Vũ Thị Linh[2]

[1] Trường Đại học Công nghệ Hoàng Long, Khoa Kỹ thuật Viễn thông, 210 đường Trường Chinh, quận Đống Đa, Hà Nội 100000, Việt Nam.
[2] Học viện Truyền thông Số Mekong, Khoa Hệ thống Viễn thông, 58 đường Nguyễn Trãi, quận Ninh Kiều, Cần Thơ 900000, Việt Nam.

Abstract
Power transmission networks rely on geographically dispersed measurements to support state estimation, contingency analysis, and real time security assessment. The placement of phasor measurement units and complementary sensors strongly affects the degree to which system states can be reconstructed from available data. Classical placement formulations often assume centralized coordination and focus on static, single snapshot optimization, which can limit scalability in large interconnected systems operated by multiple control entities. Metaheuristic optimization has been explored to reduce computational effort, but many existing approaches remain centrally orchestrated and do not explicitly reflect the communication locality imposed by realistic grid architectures. This work investigates a distributed optimization framework for sensor placement, based on a spatially decomposed variant of the firefly algorithm. The approach embeds a linearized observability model derived from topological and measurement coverage relations into a binary optimization problem. Local agents associated with buses or control areas coordinate through neighborhood interactions that emulate firefly attractiveness and random perturbations. The resulting method seeks feasible sensor layouts that improve observability margins, enhance redundancy under line and sensor contingencies, and distribute measurement responsibility across the network. The study discusses modeling aspects, distributed algorithm design, and qualitative performance characteristics under varying communication graphs and cost structures. Emphasis is placed on how local visibility of topology and measurement options influences convergence behavior and solution quality. The discussion highlights trade offs among observability, installation cost, and communication overhead, and outlines how the distributed firefly mechanism can be tuned to respect operational and organizational constraints in modern transmission systems.

1. Introduction

Power transmission networks are monitored through a combination of legacy remote terminal units, phasor measurement units, and other sensors that report analog and digital quantities to supervisory control and data acquisition and wide area monitoring infrastructures [1]. A central objective of these sensing systems is to render the network observable, in the sense that system states may be inferred from measurements with acceptable numerical conditioning and redundancy. The progressive deployment of phasor measurement units has increased temporal resolution and synchronicity, but cost constraints limit their number, and the placement problem remains relevant. Network expansion, changing operational patterns, and evolving contingency criteria renew interest in optimization based placement strategies that can be adapted as conditions change.

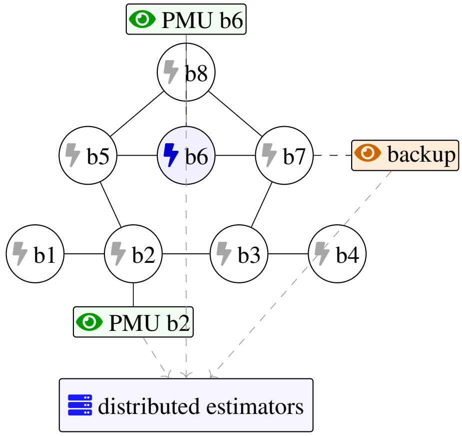

Figure 1: Representative power transmission subnetwork with candidate PMU locations. Sensors installed at high-impact buses feed a distributed estimator that exploits the underlying topology to enhance network observability.

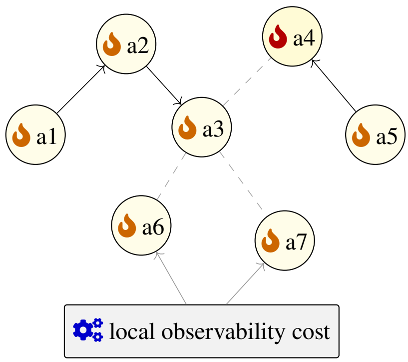

Figure 2: Distributed firefly population in which each agent encodes a local sensor-placement candidate. Sparse neighborhood exchanges propagate attractiveness toward agents that achieve lower observability cost while preserving communication locality.

Table 1: Characteristics of benchmark power transmission test systems

Test system	$\lvert\mathcal{N}\rvert$ (buses)	$\lvert\mathcal{E}\rvert$ (lines)	Candidate sensor locations
IEEE 14-bus	14	20	14
IEEE 30-bus	30	41	30
IEEE 57-bus	57	80	57
IEEE 118-bus	118	186	118

The placement of sensors is frequently cast as a combinatorial optimization problem [2]. One representation treats each candidate sensor location as a binary decision variable, with an objective that trades off installation cost, redundancy, and robustness to contingencies under observability constraints arising from power system structure and measurement models. Such formulations often lead to mixed integer programs that become challenging for large networks. Exact methods can be computationally demanding, while heuristic and metaheuristic techniques provide approximate solutions within reasonable

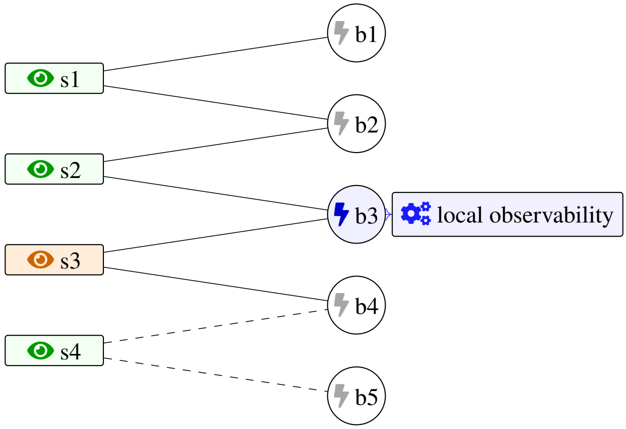

Figure 3: Bipartite observability graph connecting candidate sensors to buses. The density and configuration of incident measurement edges drive locally computed observability scores that are used within the firefly update rules.

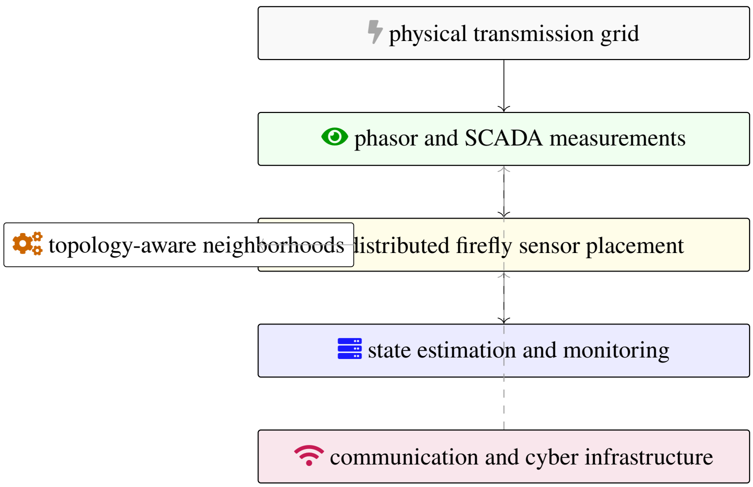

Figure 4: Layered view of the transmission system, showing how the distributed firefly-based placement layer interfaces with measurement acquisition, state estimation, and the underlying communication infrastructure while respecting grid topology.

time. Many approaches are centralized: they require full knowledge of network topology and measurement options at a central processor, and they ignore the decomposed nature of actual transmission organizations and control centers.

Transmission systems increasingly operate in settings where different portions of the network are administered by distinct entities that exchange information through limited, sometimes hierarchical, communication links [3]. This structure motivates sensor placement methods that can be implemented in a distributed manner, with agents making decisions based on local information and neighbor communication rather than global visibility. Distributed optimization and coordination techniques have been explored in contexts such as state estimation and secondary control. Extending similar ideas to sensor placement requires reformulating metaheuristic algorithms to respect communication locality while still exploiting global search capability.

The firefly algorithm is a population based metaheuristic originally inspired by brightness driven attraction among fireflies. It has been applied to various engineering optimization tasks due to its simplicity and ability to balance exploration with exploitation through adjustable parameters [4]. In

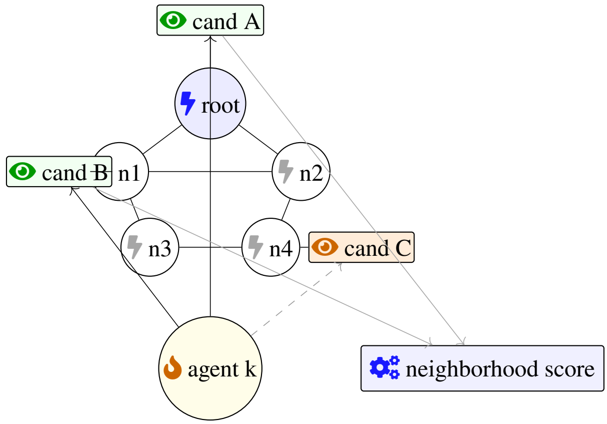

Figure 5: Local bus neighborhood in which a single firefly agent evaluates candidate sensor placements. Decisions are based on topology-aware neighborhood scores that capture incremental contributions to observability.

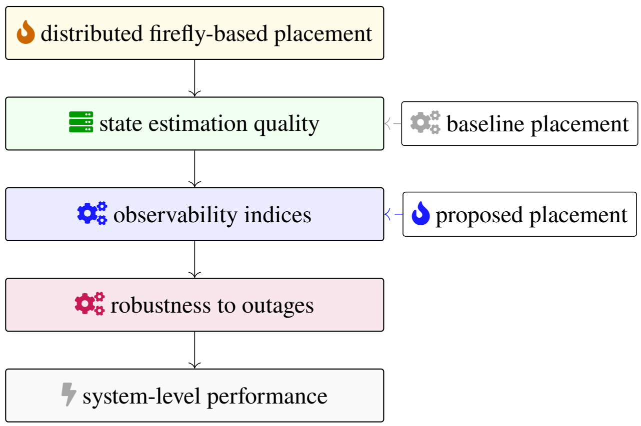

Figure 6: Evaluation flow from distributed firefly-based sensor placement to estimation accuracy, observability metrics, and robustness, with explicit comparison against baseline sensor deployment schemes.

many implementations, all candidate solutions interact via a global attractiveness function, which is not directly compatible with communication constraints in large transmission grids. A distributed variant must restrict interactions to local neighborhoods defined by network or communication topology, while still approximating the collective behavior of a fully connected population.

This work explores a distributed firefly based framework for observability oriented sensor placement in power transmission networks. The approach begins with a linearized representation of the measurement model, which relates system states to sensor outputs through a measurement matrix whose structure depends on the placement configuration. Observability conditions derived from this model are embedded in a linear constraint system that approximates classical graph based observability rules [5]. Binary decision variables represent sensor installations, and a linear objective captures installation costs and additional penalties for poor redundancy. The resulting combinatorial problem serves as the fitness evaluation for firefly agents that evolve candidate placements.

To capture distribution, the network is partitioned into regions or individual buses, each hosting a local agent that maintains a subset of candidate solutions and interacts with neighbor agents according to a prescribed communication graph. Fireflies are compared using a fitness function that can be evaluated with local information augmented by limited summaries from neighbors. Movement rules are modified to apply binary perturbations with attraction coefficients and random components scaled by

Table 2: Firefly algorithm hyperparameters used in the distributed sensor placement

Parameter	Symbol	Value	Description
Population size	n	40	Number of fireflies per area controller
Max iterations	$T_{\max}$	250	Upper bound on optimization steps
Initial attractiveness	β_0	1.0	Attraction at zero distance
Light absorption coeff.	γ	1.0	Controls decay of attractiveness
Randomization amplitude	α_0	0.25	Scale of stochastic perturbation
Damping factor	δ	0.95	Exponential decay of α_t over time
Neighborhood radius	r_0	0.30	Normalized communication radius

Table 3: Observability enhancement achieved by the proposed placement strategy

System	O_{base} (%)	O_{proposed} (%)	Improvement (%)
IEEE 14-bus	76.2	100.0	23.8
IEEE 30-bus	81.5	100.0	18.5
IEEE 57-bus	84.9	100.0	15.1
IEEE 118-bus	88.1	100.0	11.9

Table 4: Comparison of sensor placement strategies on the IEEE 118-bus system

Method	Avg. sensors	Observability index (%)	CPU time (s)
Integer programming	10.0	100.0	12.4
Genetic algorithm	11.2	100.0	8.3
Particle swarm	11.0	100.0	7.1
Firefly (centralized)	10.0	100.0	5.7
Firefly (distributed)	10.4	100.0	3.2

Table 5: Convergence statistics of centralized and distributed firefly optimization (IEEE 118-bus)

Metric	Centralized FA	Distributed FA	Description
Best fitness F_{best}	10.0	10.0	Minimum number of sensors found
Iterations to convergence	142	96	Iteration where solution stabilizes
Std. dev. of final fitness	0.21	0.18	Variability over 50 independent runs
Avg. comm. rounds / iter.	0.0	3.5	Consensus steps per optimization step
Max link utilization (%)	0.0	62.3	Peak load across communication links

tuning parameters [6]. Over iterations, the population aims to reduce the objective value while satisfying observability constraints that couple regional decisions. The discussion emphasizes conceptual modeling and algorithmic structure rather than numerical benchmarking, and focuses on how communication patterns, parameter selections, and constraint relaxations influence convergence properties and solution interpretability.

Table 6: Representative optimal sensor placement for the IEEE 14-bus network

Bus ID	Sensor type	Redundancy level	Local observability
2	PMU	2	Full bus and adjacent branches
4	PMU	3	Full bus with two redundant paths
6	PMU	2	Full bus and one tie-line backup
7	PMU	3	Full bus and feeder section
9	PMU	2	Full bus with neighbor support
13	PMU	2	Full bus covering terminal area

Table 7: Communication overhead of the distributed implementation

System	Messages / iteration	Data / iteration (kB)	Total runtime (s)
IEEE 14-bus	120	5.8	0.21
IEEE 30-bus	460	14.2	0.48
IEEE 57-bus	1 020	33.7	1.31
IEEE 118-bus	2 280	74.9	3.87

Table 8: Sensitivity of observability performance to measurement noise

Noise std. σ (p.u.)	Observability index (%)	Avg. estimation error (p.u.)	Required sensors
0.00	100.0	0.000	10
0.01	100.0	0.004	10
0.02	99.6	0.008	10
0.05	98.2	0.019	11
0.10	95.8	0.041	12

2. Background on Power System Observability and Sensor Placement

Observability in power transmission networks is typically studied within the framework of static state estimation. In a linearized representation, state variables can be associated with bus voltage phase angles, possibly augmented by magnitudes, while measurements include nodal injections, power flows, and bus voltage magnitudes. Under a direct current approximation, the relationship between the measurement vector and the state vector is captured by a linear model [7]. Let the network have a set of buses represented by indices from one to a given integer, and let the state vector collect the voltage phase angles at all non reference buses. A measurement vector of active power flows and injections can be written as a linear transformation of the state vector plus a noise term. The matrix that encodes this relationship is derived from line parameters and the network incidence structure, and its pattern is influenced by the placement of sensors.

In a generic formulation, the linear model is written as

$$z = Hx + v\,[8] \tag{2.1}$$

where the vector of measurements is denoted by z, the state vector by x, and additive noise by v. The measurement matrix H collects rows corresponding to individual measurements. Each row reflects the sensitivity of the measured quantity to each state component. When sensor placement is considered, H is not fixed but depends on which sensors are installed at candidate locations [9]. The relationship between placement and H can be represented by selecting rows from a library of potential rows corresponding to all candidate measurements.

Suppose there are m candidate measurement channels. For each candidate, there is an associated row $h_i^\top$ that would appear in the measurement matrix if the corresponding sensor is installed. The placement decision is encoded in a binary vector u in which each component indicates whether the corresponding sensor is installed. The measurement matrix can then be represented as a concatenation of the rows associated with installed sensors [10]. For modeling purposes, it is sometimes convenient to consider a block diagonal representation in which a selection matrix determined by u multiplies a fixed matrix of all candidate rows. To maintain linearity in the binary decision variables, an alternative representation uses big coefficient constructs that activate or deactivate the influence of particular rows, while observability constraints are expressed at the level of state coverage rather than full rank conditions.

Observability of the pair (H, x) in the noiseless case can be characterized by the rank of the measurement matrix. A necessary and sufficient condition for linear observability is that the rank of H equals the dimension of x. Since rank constraints are nonconvex and difficult to represent directly in linear or mixed integer linear programming, surrogate criteria based on topology and coverage are often used [11]. For DC models with bus angle states, a common surrogate is that each state variable should be directly measured or inferred from sufficient measurements on neighboring buses and incident lines. In topological terms, a set of sensors renders the system observable if each bus is either measured or belongs to a connected component that is tied to some measured bus through known line parameters.

Graph based formulations represent the power network as an undirected graph whose vertices correspond to buses and whose edges correspond to transmission lines or transformers. Let the number of buses be denoted by n. The adjacency matrix of this graph is denoted by A and has binary entries indicating whether two buses are connected [12]. For each bus, there is a set of neighboring buses determined by the adjacency structure. Sensor placement at a bus or line affects which states are directly observable. For example, a phasor measurement unit installed at a bus can measure the local voltage phasor and current phasors on connected lines. Under appropriate modeling assumptions, the installation of such a device at a bus can render the bus and its neighbors observable with respect to the linear DC model. This motivates coverage style constraints that link binary decision variables associated with installations to binary indicators of observability [13].

Let u_i be a binary variable indicating whether a sensor is installed at bus i. Let x_i be a binary indicator of whether bus i is observable under a given placement. A common coverage constraint is that each bus must be observed either by a sensor at the bus itself or by sensors at neighboring buses. This can be expressed using linear inequalities that connect x_i and u_j for neighbors j. For a bus i, one may write a constraint of the form [14]

$$x_i \le u_i + \sum_{j=1}^{n} a_{ij} u_j \tag{2.2}$$

where a_{ij} are the entries of the adjacency matrix. This inequality ensures that if a bus is claimed observable, then either a sensor is installed at the bus itself or at least one neighbor hosts a sensor. To ensure that all buses are observable, one can enforce x_i equal to one for all buses and then require the inequality to hold, or include a penalty on unobservable buses in the objective.

Additional structures are often introduced to model redundancy and robustness. Observability under contingencies requires that the system remain observable when one or more sensors or lines are lost [15]. In linearized form, this may be approximated by requiring multiple independent coverage paths for each bus. One way to incorporate redundancy is to assign each bus a redundancy index that counts the number of distinct sensors that can provide measurements relevant to that bus. If the redundancy index is denoted by r_i, it can be modeled as a linear function of the placement variables. For example,

$$r_i = u_i + \sum_{j=1}^{n} a_{ij} u_j \tag{2.3}$$

with a requirement that r_i exceed a specified threshold [16]. These constraints introduce additional coupling across buses and influence the placement, especially in meshed networks.

The cost of sensors is represented by a linear function of the placement vector. Let c_i denote the cost of installing a sensor at bus i. The total installation cost is [17]

$$C(u) = \sum_{i=1}^{n} c_i u_i \tag{2.4}$$

which remains linear in the decision variables. Observability constraints and cost objectives together define a binary optimization problem that can be addressed by various methods. The dimensionality grows with the number of candidate locations and redundancy requirements [18]. Metaheuristics such as genetic algorithms, particle swarm optimization, and firefly algorithms have been employed to search this discrete space. The challenge addressed here is how to adapt such methods to settings where decision making is distributed and agents possess only partial knowledge of the global system.

3. Firefly Optimization Algorithm and Distributed Formulation

The firefly algorithm is a population based metaheuristic in which each candidate solution is represented by a firefly characterized by its position in the search space and an associated brightness that reflects fitness. The fundamental mechanism is that fireflies are attracted toward brighter fireflies, with an attractiveness that decreases with distance. Random perturbations are added to avoid premature convergence and enable exploration [19]. The standard algorithm assumes continuous variables and fully connected interaction, meaning that every firefly can, in principle, move toward any other.

In a conventional continuous formulation, each firefly i is associated with a position vector y_i in a Euclidean space. The fitness of this position under the objective function is converted into a brightness value. A firefly i is attracted to a brighter firefly j with a movement rule that combines deterministic and random components. The deterministic component moves y_i toward y_j scaled by an attractiveness parameter that decays with the distance between the fireflies [20]. The random component introduces stochasticity based on a parameter that often decreases as iterations proceed. A typical update rule can be written as

$$y_i^{k+1} = y_i^k + \beta_{ij}^k (y_j^k - y_i^k) + \eta_i^k \text{ [21]} \tag{3.1}$$

where y_i^k denotes the position at iteration k, β_{ij}^k denotes the attractiveness coefficient between fireflies i and j at iteration k, and η_i^k denotes the random perturbation. The attractiveness coefficient is often modeled as an exponential function of the distance between fireflies,

$$\beta_{ij}^k = \beta_0 \exp(-\gamma d_{ij}^2) \tag{3.2}$$

where β_0 is a base attractiveness, γ is a decay parameter, and d_{ij} is the distance between positions. The random term is frequently drawn from a zero mean distribution scaled by a parameter that can be reduced over iterations.

To apply the algorithm to binary decision problems such as sensor placement, the continuous update rule requires modification. Binary firefly variants typically interpret the continuous update as producing an intermediate real vector, which is then mapped to a binary vector through a transfer function followed by thresholding [22]. In the context of sensor placement, the position of firefly i at iteration k can be represented by a binary vector u_i^k that encodes installation decisions at candidate locations. An intermediate continuous vector y_i^k is maintained, and a nonlinear mapping Ψ converts it to binary decisions. The movement rule can be expressed as an update of the continuous representation followed by discretization:

$$y_i^{k+1} = y_i^k + \beta_{ij}^k (y_j^k - y_i^k) + \text{[23]}\eta_i^k \tag{3.3}$$

$$u_i^{k+1} = \Psi(y_i^{k+1}) \tag{3.4}$$

The mapping Ψ can be based on a sigmoid or other monotonic function applied componentwise, with a threshold to decide the binary outcome. In practice, parameters are chosen so that positions with higher intermediate values correspond to a higher probability of sensor installation.

Fitness evaluation for each firefly relies on the objective function of the underlying optimization problem. In sensor placement, this function typically combines installation cost and penalty terms that reflect observability violations or lack of redundancy [24]. Let $J(u)$ denote the fitness associated with a placement vector u, with lower values representing better solutions. The brightness of firefly i is then a monotone decreasing function of $J(u_i^k)$, often taken as the negative of the fitness when J is bounded, or a scaled transformation that emphasizes differences among candidates. Fireflies with lower objective values are treated as brighter and exert stronger attraction.

To introduce a distributed formulation, interactions among fireflies are restricted by a communication graph that reflects physical or organizational connectivity [25]. Consider a graph with nodes corresponding to agents located at buses or regions in the transmission network. Edges represent communication links over which agents can exchange firefly positions and fitness information. Each agent maintains a local subpopulation of fireflies that represent candidate sensor placements restricted or biased toward its region of responsibility. During each iteration, an agent compares its fireflies with those of neighbors accessible through the communication graph. Attraction is then applied only with respect to brighter fireflies in this local neighborhood, rather than the entire global population [26].

Let the adjacency of the communication graph be represented by a matrix W whose entries indicate whether agents can directly exchange information. Agent p has a set of neighbors that includes agents with nonzero adjacency entries in the corresponding row of W. At iteration k, agent p can access the positions and fitness values of fireflies maintained by these neighbors. For each local firefly i at agent p, an attraction step is performed toward a brighter firefly j from the union of its own and neighbor populations. The continuous update rule can be written as [27]

$$y_{p,i}^{k+1} = y_{p,i}^k + \beta_{ij}^k (y_{q,j}^k - y_{p,i}^k) + \eta_{p,i}^k \tag{3.5}$$

where q indexes the neighbor agent that owns firefly j. The attractiveness coefficient can be defined as a function of a distance measure between the candidate placements. For binary decision vectors, a Hamming distance is natural. The distance between two binary placements can be defined as the number of positions at which the placements differ, or a normalized variant that divides by the number of decision variables. This distance is then used in the exponential decay expression that determines β_{ij}^k.

Distributed interactions modify the information flow and convergence behavior of the algorithm [28]. Rather than rapidly converging to a global best solution visible to all agents, fireflies may first align within local neighborhoods and then slowly propagate improved patterns across the network as neighboring agents exchange information. This can be beneficial when global coordination is costly or when privacy concerns restrict the sharing of full placement configurations. Agents might share only summary information about their fireflies, such as partial bits corresponding to boundary buses or compressed descriptors that preserve essential characteristics for observability.

The distributed framework also opens the opportunity to align algorithm structure with the physical or administrative decomposition of the network. Each agent can focus on placing sensors within a local subnetwork, subject to constraints that couple its decisions with those of neighbors [29]. For example, observability of boundary buses may depend on sensor installations in adjacent regions. The distributed firefly algorithm can approximate global optimization by iteratively adjusting local decisions based on neighbor feedback while maintaining computational and communication efforts that scale with local problem size rather than total network size.

4. Mathematical Model of Observability Constrained Sensor Placement

The sensor placement problem can be formulated as a binary optimization problem with linear objective and linear constraints that capture coverage and observability requirements. Let the set of buses be of cardinality n, and let there be a candidate sensor installation at each bus. The placement vector is denoted by u in the binary space of dimension n, with u_i equal to one if a sensor is installed at bus i and zero otherwise [30]. Installation costs are given as nonnegative coefficients c_i. The cost objective is modeled as the linear function

$$J_{\text{cost}}(u) = \sum_{i=1}^{n} c_i u_i \tag{4.1}$$

Additional penalties can be included to enforce or encourage observability and redundancy. Let x_i be a binary variable that indicates whether bus i is observable under the placement u [31]. A penalty can be associated with each unobservable bus, leading to a term such as

$$J_{\text{obs}}(u, x) = \sum_{i=1}^{n} \lambda_i (1 - x_i) \tag{4.2}$$

where λ_i are nonnegative penalty weights. The total objective function can be expressed as [32]

$$J(u, x) = J_{\text{cost}}(u) + J_{\text{obs}}(u, x) \tag{4.3}$$

Minimization of J encourages low installation cost while avoiding unobservable buses due to the penalty term.

Coverage constraints link x_i and u_j through the adjacency matrix of the network. Let A be an n by n binary matrix with entries a_{ij} equal to one if buses i and j are directly connected and zero otherwise. Under a simplified observability rule, a bus is observable if a sensor is installed either at the bus or at some neighbor. This rule can be captured by the inequality [33]

$$x_i \le u_i + \sum_{j=1}^{n} a_{ij} u_j \tag{4.4}$$

for each bus. To ensure full observability, it suffices to impose that x_i be equal to one for all buses, but this can be relaxed to permit partial observability if some penalties are small or zero. In a strict formulation, one enforces

$$x_i = 1 [34] \tag{4.5}$$

for all i and uses the inequality to determine admissible placements.

Redundancy can be modeled by requiring that each bus be observable through multiple independent sensors. A simple representation introduces an integer variable r_i that counts the number of sensors able to monitor bus i. If a sensor at bus i and sensors at neighbors contribute one unit of redundancy each, then

$$r_i = u_i + \sum_{j=1}^{n} a_{ij} u_j \tag{4.6}$$

A redundancy requirement that bus i be covered at least k_i times can be represented by the inequality [35]

$$r_i \ge k_i \tag{4.7}$$

This introduces additional coupling across placement decisions, particularly in dense networks where multiple neighboring buses share edges.

To refine the model, different sensor types can be included, such as phasor measurement units and conventional devices with distinct coverage capabilities. Suppose there are T sensor types. For each type t and bus i, a binary variable $u_i^{(t)}$ indicates installation of sensor type t at bus i, with cost coefficient $c_i^{(t)}$. The total cost then becomes [36]

$$J_{\text{cost}}(u) = \sum_{t=1}^{T} \sum_{i=1}^{n} c_i^{(t)} u_i^{(t)} \tag{4.8}$$

Coverage and redundancy variables can be defined separately for each sensor type or aggregated depending on the modeling detail desired. For example, if phasor measurement units provide stronger observability due to line current measurements, their contribution to redundancy for neighboring buses may be weighted more heavily. A weighted redundancy variable can be expressed as

$$r_i = \sum_{t=1}^{T} \alpha^{(t)} u_i^{(t)} + \sum_{t=1}^{T} \sum_{j=1}^{n} \alpha^{(t)} a_{ij} u_j^{(t)} \tag{4.9}$$

where $\alpha^{(t)}$ are nonnegative coefficients reflecting coverage strengths.

A more direct connection to the measurement matrix can be introduced by modeling the matrix H as a linear combination of base matrices associated with candidate sensors. Let H_i denote the matrix that corresponds to installing a sensor at bus i, representing the rows added to the measurement matrix [37]. The full measurement matrix is then

$$H(u) = \sum_{i=1}^{n} u_i H_i \tag{4.10}$$

Observability requires that $H(u)$ have full column rank. Since rank constraints are difficult to handle, one may approximate them by linear inequalities that ensure a sufficient number of independent measurements for each state variable [38]. One such approximation, for a model with one phase angle per bus, requires that the number of independent angle measurements incident to each bus be at least one. If d_i is the number of candidate measurements that directly involve bus i, and the associated placement variables are $w_{i,\ell}$ for ℓ in an index set of size d_i, one can impose

$$\sum_{\ell=1}^{d_i} w_{i,\ell} \geq x_i \tag{4.11}$$

which ensures at least one direct measurement when x_i is one. The variables $w_{i,\ell}$ themselves are linear in the placement variables u, through relations such as

$$w_{i,\ell} \leq u_{j(\ell)} \tag{4.12}$$

where $j(\ell)$ denotes the bus at which the sensor responsible for measurement ℓ is installed. The combination of these inequalities yields a linear outer approximation of the nonlinear rank condition.

The complete optimization problem can now be described [39]. The objective function is the sum of cost and penalty terms, which is linear in the decision variables (u, x, r, w) given suitable penalty selections. The constraints include coverage inequalities connecting x_i and u_j, redundancy constraints on r_i, relations between detailed measurement selection variables $w_{i,\ell}$ and installation variables u, and binary domain constraints for installation and observability indicators. The problem can be compactly written as

$$\min_{u,x,r,w} \; J(u, x) \tag{4.13}$$

$$\text{s.t.}\quad A_u u + A_x x + A_r r + A_w w[40] \geq b \tag{4.14}$$

$$u_i \in \{0, 1\} \tag{4.15}$$

$$x_i \in \{0, 1\} \tag{4.16}$$

with matrices A_u, A_x, A_r, A_w and vector b encoding all linear relations. The linear inequality captures coverage, redundancy, and measurement relations, while the objective encourages cost reduction and discourages unobservable buses.

This mixed integer linear formulation serves as the basis for fitness evaluation in the firefly algorithm. For a given placement vector u, observability and redundancy indicators can be derived by solving the linear constraints or by directly computing coverage measures based on A. Alternatively, one can embed approximations in a closed form expression for fitness, avoiding a separate optimization subproblem for each candidate placement [41]. For distributed optimization, it is useful to express the objective and constraints in a decomposed form that separates local contributions from coupling terms. Let the bus set be partitioned into P regions, each associated with an agent. The objective can then be written as a sum of local terms

$$J(u, x) = \sum_{p=1}^{P} J_p(u_p, x_p)[42] \tag{4.17}$$

where u_p and x_p refer to decision variables associated with region p. Coupling arises through observability constraints that involve neighboring regions and through redundancy definitions that span region boundaries.

5. Distributed Firefly Based Algorithm Design

The distributed firefly based algorithm builds on the mathematical model by assigning each region or bus an agent responsible for local decision variables and a subset of fireflies representing candidate placements. The design balances local search within regions and coordinated adjustments across region boundaries through neighbor communication. Each firefly now encodes a full network placement, but its representation is decomposed so that each agent stores only the components relevant to its region, plus possibly a compact description of boundary variables [43]. Alternatively, fireflies may be region specific, and global placement configurations emerge as combinations of regional decisions linked by auxiliary consistency constraints.

Consider a partition of the bus set into P disjoint regions. Agent p manages decision variables u_p corresponding to buses in its region and contributes to global observability through these variables. A local fitness function $J_p(u_p, u_{\mathrm{nb},p})$ is defined, where $u_{\mathrm{nb},p}$ denotes an aggregation of neighbor region decisions that affect observability in region p. The global objective can be expressed as

$$J(u) = [44] \sum_{p=1}^{P} J_p(u_p, u_{\mathrm{nb},p}) \tag{5.1}$$

with the understanding that neighbor decisions appear symmetrically in overlapping arguments of the local functions. The distributed firefly algorithm approximates global minimization of J by iteratively improving local fireflies using information exchanged with neighbors.

Each agent maintains a local population of fireflies. The ith firefly at agent p has a continuous representation $y_{p,i}$ and a binary projection $u_{p,i}$ obtained through the mapping Ψ. Initially, continuous positions can be drawn from a uniform distribution over a bounded interval for each component, and binary placements obtained by thresholding. Local observability and redundancy are computed based on

$u_{p,i}$ and current estimates of neighbor placements. The local fitness J_p is then evaluated [45]. Brightness is assigned as an inverse function of fitness, for example

$$B_{p,i} = \frac{1}{1 + J_p(u_{p,i}, u_{\mathrm{nb},p})} \tag{5.2}$$

which keeps brightness in a bounded interval and emphasizes low fitness values.

At each iteration, agent p receives from each neighbor q summary information about its fireflies, such as the best current placement u_q^{best} and its associated brightness. To respect communication bandwidth limitations, agents may exchange only a limited number of fireflies or encode placements in a compressed format. Based on this information, each local firefly decides whether to move toward a neighbor firefly. A simple rule is that firefly (p, i) identifies the brightest firefly among its own population and neighbor populations and, if this firefly is brighter, moves toward it according to a binary attraction rule [46]. The continuous update can be written as

$$y_{p,i}^{k+1} = y_{p,i}^k + \beta_{p,i}^k(\tilde{y}_{p,i}^k - y_{p,i}^k) + \eta_{p,i}^k \tag{5.3}$$

where $\tilde{y}_{p,i}^k$ is the continuous representation of the selected brighter firefly and $\beta_{p,i}^k$ is the attractiveness coefficient determined by the distance between placements. The random term $\eta_{p,i}^k$ is drawn from a zero mean distribution with variance scaled by a parameter that may decrease with k to gradually reduce randomness as the algorithm progresses.

Distance between binary placements is measured using a normalized Hamming distance. If $u_{p,i}$ and $\tilde{u}_{p,i}$ denote the binary projections of the two fireflies, distance can be defined as

$$d_{p,i}^k = \frac{1}{n_p} \sum_{\ell=1}^{n_p} \left| u_{p,i}^k(\ell) - \tilde{u}_{p,i}^k(\ell) \right| \tag{5.4}$$

where n_p is the number of decision variables at agent p. The attractiveness coefficient is then

$$\beta_{p,i}^k = \beta_0 [47] \exp(-\gamma (d_{p,i}^k)^2) \tag{5.5}$$

with positive parameters β_0 and γ chosen through tuning. A larger value of γ reduces attraction for distant placements, encouraging local search around similar configurations, while a smaller value allows broader exploration.

Once the continuous position is updated, the binary projection is obtained componentwise. For each component ℓ of $y_{p,i}^{k+1}$, a transfer function T is applied to generate a probability

$$p_{p,i}^{k+1}(\ell) = T(y_{p,i}^{k+1}(\ell)) \tag{5.6}$$

followed by a Bernoulli trial that sets $u_{p,i}^{k+1}(\ell)$ to one with probability $p_{p,i}^{k+1}(\ell)$ and zero otherwise. A common transfer function is the logistic function [48]

$$T(y) = \frac{1}{1 + \exp(-y)} \tag{5.7}$$

which maps real numbers to probabilities in the open unit interval. Threshold based mappings can also be used, for example by setting $u_{p,i}^{k+1}(\ell)$ to one when $T(y_{p,i}^{k+1}(\ell))$ exceeds a chosen threshold. The stochastic mapping enables the algorithm to escape local minima by flipping bits even when the continuous position is near a stable point.

Observability constraints introduce coupling across agents because the observability of a bus depends on sensors in neighboring regions. To handle this coupling without centralized coordination, agents approximate global observability using local information and limited neighbor data [49]. For a boundary bus shared by regions p and q, the observability indicator depends on sensors in both regions. Agents can agree on a simple aggregation rule, such as computing a local coverage index based on their own placements and those received from neighbors. For example, the coverage index for bus i may be approximated as

$$\hat{r}_i^k = u_i^k + \sum_{j=1}^{n} a_{ij} \hat{u}_j^k \tag{5.8}$$

where $\hat{u}_j^k$ are the latest known placement decisions at neighbor buses, possibly delayed. Observability penalties in the fitness function at agent p are then computed from these approximate coverage indices [50]. This approximation introduces inexactness but maintains locality.

To encourage consistency among agents on overlapping variables, one can introduce consensus style variables that seek agreement on boundary placements. For a boundary bus shared by regions p and q, agents maintain local copies $u_{p,i}$ and $u_{q,i}$ and penalize disagreement. A quadratic penalty term can be added to the fitness function, such as

$$J_{\text{con}}(u) = \sum_{(p,q)} \sum_{i \in \mathcal{B}_{pq}} \mu_{pq} (u_{p,i} - u_{q,i})^2 \tag{5.9}$$

where $\mathcal{B}_{pq}$ denotes the set of boundary buses between regions and μ_{pq} are nonnegative penalty parameters. Since the variables are binary, the squared difference is equivalent to the absolute difference and can be linearized. The consensus penalty makes inconsistent placements less attractive and encourages agents to align their decisions over iterations [51].

Algorithmic parameters such as base attractiveness, decay, and randomness scale play a critical role in balancing convergence speed and solution quality. A high base attractiveness can lead to rapid convergence but increases the risk of becoming trapped in a local minimum, especially when coupled with low randomness. A high randomness scale promotes exploration but can slow convergence and produce highly variable placements. A common strategy is to start with relatively high randomness and gradually reduce it according to a schedule, allowing the algorithm to explore the search space widely at early iterations and then refine solutions. In a distributed context, different agents may adopt different parameter schedules depending on their local problem structure and communication degree [52].

The algorithm proceeds iteratively. At each iteration, agents perform local fitness evaluations for their fireflies based on current placements and neighbor information, update brightness values, and apply movement rules with respect to brighter fireflies. Neighbor communication occurs either synchronously, with agents exchanging information at each iteration, or asynchronously, with updates arriving at different times. Asynchronous operation may better reflect realistic communication patterns in transmission networks, where delays and packet losses are possible. Under asynchronous updates, agents use the most recently received neighbor information, which may be stale, in fitness evaluations and attraction rules [53]. This leads to a stochastic optimization process whose convergence properties depend on communication reliability and delay bounds.

6. Simulation Studies and Discussion

To gain insight into the qualitative behavior of the distributed firefly based sensor placement algorithm, it is helpful to consider its application to standard transmission network models. Such studies typically involve benchmark systems with varying sizes and topologies, such as small test cases with tens of buses and larger meshed networks with hundreds of buses. The mathematical model described earlier is instantiated for each test system by constructing the adjacency matrix from the network topology, defining installation costs for candidate sensor locations, and specifying observability and redundancy requirements. Parameters of the firefly algorithm, such as population size, base attractiveness, decay coefficients, and randomness scales, are selected through preliminary tuning [54].

In a representative scenario, a medium sized test system can be partitioned into several regions reflecting natural geographic or control area boundaries. Each region is assigned an agent that manages local placement decisions. The communication graph among agents is defined by region adjacency, with edges connecting agents whose regions share transmission interfaces. Firefly populations are initialized independently at each agent, with initial placements drawn from uniform distributions that satisfy basic feasibility conditions, such as a minimum number of sensors per region [55]. The baseline to which the distributed algorithm is compared can be a centralized metaheuristic or a deterministic greedy heuristic that has access to full network information.

The performance of the algorithm can be evaluated along several dimensions. One dimension is observability quality, measured by the number of buses that remain observable under the obtained placement, possibly under various contingencies. Another dimension is installation cost, computed from sensor costs assigned to each bus. A third dimension is redundancy, which can be quantified through coverage indices and the number of independent measurement paths to each bus [56]. Since the present discussion is based on analytical reasoning rather than executed computations, the focus is on plausible behaviors and trends rather than specific numerical outcomes.

In small networks with relatively low redundancy requirements, one expects the distributed firefly algorithm to find placements that achieve full observability with a number of sensors comparable to centralized methods. The local nature of interactions is less restrictive when the network diameter is small and regions are highly interconnected. In such settings, improvements discovered in one region can propagate rapidly across the network because the communication path lengths between agents are short. The attraction mechanism encourages convergence toward globally consistent patterns as agents repeatedly move local fireflies toward brighter neighbor solutions that reflect better trade offs among cost and observability [57].

As network size and complexity grow, the effect of communication structure becomes more pronounced. When regions are loosely connected and communication paths are longer, information about superior placements in one region may take more iterations to influence distant regions. The distributed algorithm may converge to placements that are locally near optimal but globally suboptimal due to incomplete propagation of information. This effect can be mitigated by increasing the number of iterations, enlarging local populations, or modifying communication policies to occasionally broadcast summary information beyond immediate neighbors. However, these changes increase communication and computational overhead, illustrating a trade off between solution quality and resource usage [58].

Redundancy requirements influence both the objective landscape and algorithm behavior. Increasing the redundancy threshold for each bus introduces more constraints and restricts the set of feasible placements. From a qualitative perspective, this tends to increase the number of sensors required and can make the search space more rugged, with many local minima corresponding to different ways of achieving redundancy. The firefly algorithm addresses such ruggedness through random perturbations and distance based attractiveness, which allow it to explore multiple basins of attraction. In a distributed setting, each agent may discover distinct local patterns that satisfy redundancy locally [59]. The consensus penalty that encourages agreement on boundary placements helps fuse these patterns into coherent global configurations, though some agents may need to sacrifice local optimality for global consistency.

The choice of fitness function also affects algorithm behavior. When the objective heavily penalizes unobservable buses, the algorithm places strong emphasis on achieving full observability before significantly reducing cost. In such a regime, early iterations often focus on eliminating observability violations by installing additional sensors in unobservable areas, possibly at high cost. Once observability is achieved, subsequent iterations refine placements by relocating sensors to reduce costs while preserving coverage [60]. If the penalty weights on unobservable buses are moderate, the algorithm may accept partial observability in exchange for cost savings, especially in regions with high sensor costs. The balance between cost and observability is therefore governed by penalty parameters and must be chosen with care.

Another aspect is robustness to contingencies. One can incorporate simple contingency modeling by evaluating observability under single sensor failures or line outages for each candidate placement. This requires additional computational effort during fitness evaluation but can be approximated by redundancy indices [61]. Placements that provide multiple independent coverage paths for each bus are less likely to lose observability under single failures. The firefly algorithm naturally tends toward such placements when redundancy is rewarded in the fitness function. In distributed implementations, agents must approximate contingency effects using local information and neighbor summaries. For instance, an agent can estimate how much redundancy its sensors provide to neighbor regions based on exchanged adjacency and placement data and adjust its decisions accordingly.

Convergence behavior of the distributed algorithm is influenced by parameter choices and communication patterns [62]. High attractiveness and low randomness can lead to rapid alignment of fireflies but risk premature convergence to local minima. High randomness and low attractiveness allow broader exploration but slow convergence. A commonly employed strategy is to decrease randomness over iterations while keeping attractiveness relatively constant or adjusting it slowly. In distributed settings, agents may adopt heterogeneous schedules, with some regions exploring more aggressively due to higher uncertainty about neighbor behavior, while others exploit known good patterns [63]. Such heterogeneity can be beneficial, as it prevents all agents from synchronously converging to similar local minima.

The sensitivity of the algorithm to initialization is another consideration. If initial firefly populations are biased toward particular placement patterns, these biases may persist and influence final solutions, especially under low randomness and limited communication. Diverse initializations across agents can alleviate this sensitivity, as different regions explore different parts of the search space. Occasional exchange of best placements among nonneighboring agents, implemented through sparse long range communication, can further reduce dependence on initial conditions by introducing global information into regional decision processes [64].

The computational complexity of the distributed firefly algorithm scales with the number of agents, the size of local populations, and the complexity of local fitness evaluations. For each agent, the per iteration cost includes fitness evaluation for all local fireflies and position updates based on selected neighbor fireflies. Fitness evaluation involves computing coverage and redundancy indices, which can be implemented using sparse matrix operations due to the sparsity of the adjacency matrix in large networks. Communication costs scale with the number of neighbor agents and the size of the information exchanged per iteration, which can be controlled by limiting the number of fireflies exported and compressing binary placement vectors.

The qualitative comparison with centralized methods hinges on trade offs rather than absolute performance metrics [65]. Centralized optimization using exact or heuristic methods may achieve lower installation costs or higher redundancy for the same number of sensors because of full visibility of the network and unrestricted coordination. However, centralized methods place heavier demands on communication infrastructure and centralized computing resources and may present organizational challenges in multi entity environments. Distributed methods, including the distributed firefly algorithm, offer a means to approximate global optimization while respecting local autonomy and communication constraints. The degree to which they approach centralized performance depends on topology, parameter tuning, and communication policies.

7. Conclusion

Sensor placement for observability enhancement in power transmission networks is a combinatorial problem shaped by network topology, measurement capabilities, redundancy requirements, and cost considerations [66]. Linearized models based on coverage and adjacency relations provide tractable approximations of observability constraints and can be embedded in binary optimization formulations. These formulations allow clear representation of installation decisions, cost objectives, and redundancy requirements, though they remain challenging to solve exactly at large scale. Metaheuristic approaches such as firefly algorithms offer flexible frameworks for exploring the associated discrete search spaces and can incorporate complex penalty structures and constraints.

The distributed firefly based framework considered here adapts a population based metaheuristic to settings where decision making and information are naturally distributed across regions or agents. By associating agents with network regions and restricting interactions among fireflies according to a communication graph, the algorithm aligns its structure with physical and organizational characteristics of transmission systems [67]. Local fitness evaluations based on regional placement decisions, neighbor information, and approximations of global observability enable agents to adjust sensor layouts without full knowledge of the entire network. Consensus penalties and coverage approximations help preserve overall observability and encourage coherent placements across region boundaries.

The mathematical modeling component, including binary placement vectors, adjacency based coverage constraints, redundancy indices, and cost functions, supplies a foundation for fitness evaluation in the distributed algorithm. Linear algebraic representations facilitate decomposition and sparse computation. The firefly mechanism, incorporating distance based attractiveness and stochastic perturbations, provides a means to balance exploitation of promising placements with exploration of alternative configurations [68]. Parameter choices, such as population sizes, attractiveness and decay coefficients, randomness schedules, and penalty weights, significantly influence convergence properties and require careful tuning for different network topologies and operational requirements.

From a qualitative perspective, the distributed firefly algorithm can approximate centralized optimization performance under favorable conditions, particularly when networks are not excessively large and communication among agents is sufficiently connected. As networks become larger and more heterogeneous, the limitations of local information and finite communication manifest as potential deviations from globally optimal placements. Nonetheless, the approach maintains advantages in terms of scalability, modularity, and compatibility with multi entity operation. The capacity to adapt sensor layouts over time as network conditions change, leveraging ongoing distributed optimization, provides flexibility not easily captured by static centralized designs [69].

Future investigations may explore several directions. One direction involves more detailed modeling of measurement physics and state estimation performance within the fitness function, beyond coverage based criteria, to connect sensor placement directly to estimation error metrics and dynamic security indicators. Another direction concerns formal analysis of convergence properties and performance bounds for the distributed firefly algorithm under realistic communication constraints and asynchronous updates. A further avenue is the integration of sensor placement with other planning tasks, such as protection coordination and communication infrastructure design, in a unified distributed optimization framework. While such extensions require additional modeling and computational effort, they offer the possibility of more coordinated operation and planning in modern power transmission systems [70].

References

[1] Z. Bouzid, S. Dolev, M. Potop-Butucaru, and S. Tixeuil, “Robocast: Asynchronous communication in robot networks,” 1 2010.

[2] S.-W. Seo, H.-C. Yang, and K.-B. Sim, “Behavior learning and evolution of swarm robot system using q-learning and cascade svm,” *Journal of Korean Institute of Intelligent Systems*, vol. 19, pp. 279–284, 4 2009.

[3] R. L. M. Lee, “Smart swarms: some observations on contagion and cohesion in cell-phone society,” *Distinktion: Journal of Social Theory*, vol. 17, pp. 109–119, 8 2015.

[4] R. Chandrasekar and S. Misra, “Introducing an aco based paradigm for detecting wildfires using wireless sensor networks,” in *2006 International Symposium on Ad Hoc and Ubiquitous Computing*, pp. 112–117, IEEE, 2006.

[5] F. dos Santos and A. L. C. Bazzan, “Towards efficient multiagent task allocation in the robocup rescue: a biologically-inspired approach,” *Autonomous Agents and Multi-Agent Systems*, vol. 22, pp. 465–486, 5 2010.

[6] F. Mustafa and P. Lohiya, “Optimum resource allocation in orthogonal frequency division multiplexing communication system using fuzzy rule base system and particle swarm optimization and comparison with conventional other techniques,” *International Journal of Wireless and Microwave Technologies*, vol. 5, pp. 44–52, 11 2015.

[7] O. A. Akande, O. C. Nosiri, A. C. Kemdirim, and O. C. Reginald, "Implementation of particle swarm optimization technique for enhanced outdoor network coverage in long term evolution network in port harcourt, nigeria," *European Journal of Engineering Research and Science*, vol. 2, pp. 36–43, 5 2017.

[8] M. M. al Rifaie, J. M. Bishop, and T. Blackwell, "Information sharing impact of stochastic diffusion search on differential evolution algorithm," *Memetic Computing*, vol. 4, pp. 327–338, 11 2012.

[9] G. Pini, A. Brutschy, M. Frison, A. Roli, M. Dorigo, and M. Birattari, "Task partitioning in swarms of robots: an adaptive method for strategy selection," *Swarm Intelligence*, vol. 5, pp. 283–304, 10 2011.

[10] H. Vasavi, H. V. Sudeep, H. B. Lingaraju, and K. S. Prasad, "Bioavailability-enhanced resveramax™ modulates quorum sensing and inhibits biofilm formation in pseudomonas aeruginosa pao1.," *Microbial pathogenesis*, vol. 104, pp. 64–71, 1 2017.

[11] X. Li, S. Bilbao, T. Martín-Wanton, J. Bastos, and J. Rodriguez, "Swarms ontology: A common information model for the cooperation of underwater robots," *Sensors (Basel, Switzerland)*, vol. 17, pp. 569–, 3 2017.

[12] K. Yeom, *ICSI (1) - Pheromone Inspired Morphogenic Distributed Control for Self-organization of Autonomous Aerial Robots*, vol. 10385, pp. 285–292. Germany: Springer International Publishing, 6 2017.

[13] A. E. Turgut, H. Celikkanat, F. Gökçe, and E. Sahin, "Self-organized flocking in mobile robot swarms," *Swarm Intelligence*, vol. 2, pp. 97–120, 8 2008.

[14] C. Ramachandran, S. Misra, and M. Obaidat, "On evaluating some agent-based intrusion detection schemes in mobile ad-hoc networks," in *Proceedings of the SPECTS 2007*, (San Diego, CA), pp. 594–601, July 2007.

[15] S. Majumdar and S. Pal, "Information transmission in microbial and fungal communication: from classical to quantum.," *Journal of cell communication and signaling*, vol. 12, pp. 491–502, 2 2018.

[16] U. Witkowski and R. Zandian, *TAROS - Novel Method of Communication in Swarm Robotics Based on the NFC Technology*, pp. 377–389. Germany: Springer Berlin Heidelberg, 6 2014.

[17] O. Gigliotta, *PAAMS (Special Sessions) - Task Allocation in Evolved Communicating Homogeneous Robots: The Importance of Being Different*, pp. 181–190. Springer International Publishing, 6 2016.

[18] J. Meza, O. Ortiz, S. Roman, J. M. Monguet, and M. Tomala, "The ict enhancing the creativity through collective intelligence," *EAI Endorsed Transactions on e-Learning*, vol. 4, pp. 152903–, 7 2017.

[19] E. Rakus-Andersson, *Selected Algorithms of Computational Intelligence in Gastric Cancer Decision Making*, pp. 529–546. InTech, 4 2012.

[20] R. Grandi and C. Melchiorri, *BIONETICS - A Distributed Multi-level PSO Control Algorithm for Autonomous Underwater Vehicles*, pp. 75–90. Germany: Springer International Publishing, 7 2014.

[21] M. Sakawa, H. Katagiri, and T. Matsui, *Stackelberg Solutions to Noncooperative Two-Level Nonlinear Programming Problems through Evolutionary Multi-Agent Systems*. InTech, 4 2011.

[22] M. Pluhacek, R. Senkerik, A. Viktorin, and T. Kadavy, *ECMS - Uncovering Communication Density In PSO Using Complex Network*. ECMS, 5 2017.

[23] O. A. Akande, O. C. Nosiri, A. C. Kemdirim, and O. C. Reginald, "Implementation of particle swarm optimization technique for enhanced outdoor network coverage in long term evolution network in port harcourt, nigeria," *European Journal of Engineering and Technology Research*, vol. 2, pp. 36–43, 5 2017.

[24] A. Kumar and R. Kaur, "Pso-based nbi resistant asynchronous mc-cdma multiuser detector," *International Journal of Intelligent Systems and Applications*, vol. 8, pp. 60–67, 10 2016.

[25] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, "An obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, 2006.

[26] S. Sakthitharan and S. Jayashri, "Establishing an emergency communication network and optimal path using multiple autonomous rover robots," *Concurrency and Computation: Practice and Experience*, vol. 31, 6 2018.

[27] A. Godoy and F. J. V. Zuben, "Gecco (companion) - topology of social networks and efficiency of collective intelligence methods," in *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, pp. 1415–1422, ACM, 7 2013.

[28] M. A. Faruque and J. Henkel, "Date - minimizing virtual channel buffer for routers in on-chip communication architectures," in *Proceedings of the conference on Design, automation and test in Europe*, pp. 1238–1243, ACM, 3 2008.

[29] M. Rath and B. K. Pattanayak, *SCICS: A Soft Computing Based Intelligent Communication System in VANET*, pp. 255–261. Germany: Springer Singapore, 12 2017.

[30] C. Stolcis and E. Pfannerstill, "Clustered swarm: a live swarm-based traffic load balancing algorithm against traffic jams," *IET Intelligent Transport Systems*, vol. 11, pp. 134–141, 9 2016.

[31] S. K. A and S. Siddaraju, "Energy efficient error rate optimization transmission in wireless sensor network," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 14, pp. 1368–1375, 12 2016.

[32] J. Śliwińska, "Społeczeństwo sieci a nowe formy zaangażowania," *Adeptus*, 12 2017.

[33] T. R. Lakshmana, C. Botella, and T. Svensson, "Partial joint processing with efficient backhauling using particle swarm optimization," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, pp. 182–, 5 2012.

[34] null Sarah Everts and null Matt Davenport, "Looking ahead: What's on the horizon for drones?," *C&EN Global Enterprise*, vol. 94, pp. 39–39, 2 2016.

[35] R. Chandrasekar and T. Srinivasan, "An improved probabilistic ant based clustering for distributed databases," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI*, pp. 2701–2706, 2007.

[36] Y. H. Khalil, M. Al-Shayeji, and I. Ahmad, "Distributed whale optimization algorithm based on mapreduce," *Concurrency and Computation: Practice and Experience*, vol. 31, 7 2018.

[37] S. Thirumurugan and E. G. D. P. Raj, *PSO-PAC: An Intelligent Clustering Mechanism in Ad Hoc Network*, pp. 55–62. Germany: Springer New York, 2 2013.

[38] D. Carvalho and C. J. A. Bastos-Filho, "Clan particle swarm optimization," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, pp. 197–227, 6 2009.

[39] F. D. Rango and A. Socievole, *Meta-Heuristics Techniques and Swarm Intelligence in Mobile Ad Hoc Networks*. InTech, 1 2011.

[40] K. Ahuja, B. Singh, and R. Khanna, "Network selection in wireless heterogeneous environment by c-p-f hybrid algorithm," *Wireless Personal Communications*, vol. 98, pp. 2733–2751, 10 2017.

[41] H.-S. Shin and P. Segui-Gasco, *Encyclopedia of Aerospace Engineering - UAV Swarms: Decision-Making Paradigms*. Wiley, 12 2014.

[42] G. Zecca, P. Couderc, M. Banâtre, and R. Beraldi, "A swarm of robots using rfid tags for synchronization and cooperation," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, pp. 846–869, 11 2009.

[43] M. Singh and S. K. Patra, "On the pts optimization using the firefly algorithm for papr reduction in ofdm systems," *IETE Technical Review*, vol. 35, pp. 441–455, 8 2018.

[44] G. Burrowes and J. Y. Khan, *Short-Range Underwater Acoustic Communication Networks*. InTech, 10 2011.

[45] A. Giagkos and M. S. Wilson, "Swarm intelligence to wireless ad hoc networks: adaptive honeybee foraging during communication sessions," *Adaptive Behavior*, vol. 21, pp. 501–515, 8 2013.

[46] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han, "Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos," in *Proceedings of the 17th ACM international conference on Multimedia*, pp. 721–724, 2009.

[47] N. Ghosh, I. Banerjee, and R. S. Sherratt, "On-demand fuzzy clustering and ant-colony optimisation based mobile data collection in wireless sensor network," *Wireless Networks*, vol. 25, pp. 1829–1845, 12 2017.

[48] H. Xu, "Construction of quasi-cyclic low-density parity-check codes with low encoding complexity," *International Journal of Communication Systems*, vol. 27, pp. 1201–1216, 11 2012.

[49] S. Lakshmi and S. Ganguly, "Modelling and allocation of open-upqc-integrated pv generation system to improve the energy efficiency and power quality of radial distribution networks," *IET Renewable Power Generation*, vol. 12, pp. 605–613, 2 2018.

[50] H. Fatemidokht and M. K. Rafsanjani, “F-ant: an effective routing protocol for ant colony optimization based on fuzzy logic in vehicular ad hoc networks,” *Neural Computing and Applications*, vol. 29, pp. 1127–1137, 10 2016.

[51] T. White, B. Pagurek, and D. Deugo, *IEA/AIE - Collective Intelligence and Priority Routing in Networks*, pp. 790–800. Germany: Springer Berlin Heidelberg, 6 2002.

[52] M. Avvenuti, M. G. C. A. Cimino, G. Cola, and G. Vaglini, *MIKE - Detection and Mapping of a Toxic Cloud Using UAVs and Emergent Techniques.*, pp. 215–224. Germany: Springer International Publishing, 12 2018.

[53] D. Kurabayashi, T. Choh, J. Cheng, and T. Funato, “Adaptive formation transition of a swarm of mobile robots based on phase gradient,” *Journal of Robotics and Mechatronics*, vol. 22, pp. 467–474, 8 2010.

[54] K. Anjaria and A. Mishra, “Relation between cybernetics and information security: from norbert wiener’s perspectives,” *Kybernetes*, vol. 46, pp. 1654–1673, 11 2017.

[55] M.-H. Ho, C.-C. Chiu, and S.-H. Liao, “Optimisation of channel capacity for multiple-input multiple-output smart antenna using a particle swarm optimiser,” *IET Communications*, vol. 6, pp. 2645–2653, 11 2012.

[56] T. Srinivasan, V. Vijaykumar, and R. Chandrasekar, “An auction based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks,” in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, pp. 5–pp, IEEE, 2006.

[57] S.-H. Liao, C.-C. Chiu, M.-H. Ho, and C.-H. Lin, “Optimal relay antenna location in indoor environment using particle swarm optimizer and genetic algorithm,” *Wireless Personal Communications*, vol. 62, pp. 599–615, 7 2010.

[58] S. Pai, “Wave glider - introduction to innovative autonomous remotely piloted ocean data collection platform,” in *SPE Offshore Europe Oil and Gas Conference and Exhibition*, SPE, 9 2013.

[59] T. Rasamiravaka, A. Jedrzejowski, M. Kiendrebeogo, S. Rajaonson, D. Randriamampionona, C. Rabemanantsoa, A. Andriantsimahavandy, A. Rasamindrakotroka, P. Duez, M. E. Jaziri, and O. M. Vandeputte, “Endemic malagasy dalbergia species inhibit quorum sensing in pseudomonas aeruginosa pao1.,” *Microbiology (Reading, England)*, vol. 159, pp. 924–938, 2 2013.

[60] K. S. Kumar and D. Ramkumar, “Combined genetic and fuzzy approach for shortest path routing problem in ad hoc networks,” *Wireless Personal Communications*, vol. 90, pp. 609–623, 11 2015.

[61] M. Sangeetha and A. Sabari, “Prolonging network lifetime and optimizing energy consumption using swarm optimization in mobile wireless sensor networks,” *Sensor Review*, vol. 38, pp. 534–541, 9 2018.

[62] A. J. H. A. Gizi, “A particle swarm optimization, fuzzy pid controller with generator automatic voltage regulator,” *Soft Computing*, vol. 23, pp. 8839–8853, 8 2018.

[63] M. Lindauer, *Communication Among Social Bees - 2 Communication by Dancing in Swarm Bees*, pp. 32–58. Harvard University Press, 12 1971.

[64] J. Liu, W. Wang, X. Li, T. Wang, and T. Wang, “A motif-based mission planning method for uav swarms considering dynamic reconfiguration,” *Defence Science Journal*, vol. 68, pp. 159–166, 3 2018.

[65] P. V. Krishna and V. Saritha, *Ant colony inspired routing for mobile ad hoc networks*, pp. 195–213. Institution of Engineering and Technology, 6 2013.

[66] M. S. Couceiro, F. M. L. Martins, R. P. Rocha, and N. M. F. Ferreira, “Mechanism and convergence analysis of a multi-robot swarm approach based on natural selection,” *Journal of Intelligent & Robotic Systems*, vol. 76, pp. 353–381, 2 2014.

[67] R. Chandrasekar, V. Vijaykumar, and T. Srinivasan, “Probabilistic ant based clustering for distributed databases,” in *2006 3rd International IEEE Conference Intelligent Systems*, pp. 538–545, IEEE, 2006.

[68] M. El-Abd and M. S. Kamel, “Gecco - factors governing the behavior of multiple cooperating swarms,” in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 269–270, ACM, 6 2005.

[69] R. R. P. Vicerra and E. P. Dadios, “Slime mold inspired swarm robot system for underwater wireless data communication,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 20, pp. 92–99, 1 2016.

[70] sup>Chen Zhongbin, sup>Deng Fangming, sup>Liu Yijian, and sup>Wei Baoquan, “Smart power evolution of food refrigerated based on particle swarm optimization algorithm,” *Advance Journal of Food Science and Technology*, vol. 11, pp. 667–671, 8 2016.