Nuvern International

Ant Colony Optimization with Capacity-Aware Pheromone Models for Load-Balanced Service Placement in Fog Infrastructures

Woraphon Suthammarat[1] **and Nattaya Phromchai**[2]

[1]**Mae Fah Luang University, Faculty of Information and Communication Technology, 333 Moo 1 Thasud, Muang District, Chiang Rai 57100, Thailand**
[2]**Prince of Songkla University, Faculty of Computing, 15 Karnjanavanich Road, Hat Yai District, Songkhla 90110, Thailand**

ABSTRACT

Fog computing extends cloud capabilities toward the network edge to support latency-sensitive and bandwidth-intensive services arising from the proliferation of Internet of Things devices. In such infrastructures, services must be placed on heterogeneous fog nodes with limited capacity, while traffic demands vary over time and space. Unbalanced placement can lead to resource hot-spots, increased response times, and degraded quality of service. Traditional deterministic optimization methods often become intractable at the scale and dynamism of realistic fog deployments, motivating heuristic and metaheuristic approaches. This paper investigates ant colony optimization strategies for load-balanced service placement in fog infrastructures, focusing on capacity-aware pheromone models. The study considers a generic fog topology with constrained compute and bandwidth resources, a set of multi-tenant services with heterogeneous demands, and a traffic matrix describing the association between end-user regions and services. The placement problem is formulated as a linear mixed-integer model that jointly captures capacity constraints, routing decisions, and load-balance objectives. Upon this model, a family of ant colony algorithms is constructed in which pheromone values explicitly encode residual capacities, congestion indicators, and marginal load-balancing costs. Evaporation and reinforcement rules are designed to avoid convergence to placements that violate capacity limits or create persistent hot-spots. The resulting algorithms are discussed in terms of convergence behavior, structural properties of produced placements, and computational complexity. Numerical experiments on synthetic fog topologies illustrate how capacity-aware pheromone designs influence load distribution, path selection, and robustness under changing workload conditions.

1 INTRODUCTION

Fog computing introduces a distributed layer of computation, storage, and networking between end devices and centralized cloud data centers [1]. This intermediate layer aims to reduce access latency, avoid excessive backbone traffic, and enable context-aware processing closer to data sources. Fog infrastructures are composed of heterogeneous nodes such as gateways, micro data centers, and base stations that exhibit diverse capacities, energy budgets, and connectivity patterns. At the same time, service providers deploy elastic services and microservices on these nodes to support applications such as real-time analytics, industrial control, smart grids, and augmented reality [2]. The resulting service placement problem is driven by multiple objectives and constraints that include latency, resource utilization, reliability, and operational cost.

A central difficulty in fog service placement lies in the spatial and temporal variation of demands. End-user devices are mobile, and their request patterns evolve with time [3]. Services may have different resource intensities and affinity to particular regions. Some functions are latency-critical, while others are throughput-oriented. Fog nodes have limited capacity and are often deployed in a hierarchical or partially meshed topology with constrained link bandwidths. A naive placement strategy that simply places services greedily at the closest node or the currently least loaded node can quickly create unbalanced resource usage [4]. Some nodes experience sustained overload, leading to queueing delays and potential violations of service-level agreements, while other nodes remain underutilized.

Exact mathematical programming formulations of the service placement problem can capture these aspects through binary placement decisions, continuous

Table 1. Notation used in the capacity-aware ant colony optimization model

Symbol	Description	Type	Range
N	Number of fog nodes	Integer	$[1,\infty)$
M	Number of service instances	Integer	$[1,\infty)$
C_i	Capacity of fog node i	Resource units	$(0,C^{\max}]$
d_{ij}	Network delay between node i and user zone j	Time	$[0,d^{\max}]$
L_k	Maximum latency for service k	Time	$(0,L^{\max}]$

Table 2. Example fog node configuration with heterogeneous resources

Fog Node	CPU (GHz)	RAM (GB)	Bandwidth (Mbps)
F1 (edge micro DC)	3.2	16	1000
F2 (gateway)	2.4	8	500
F3 (access point)	1.8	4	200
F4 (metro DC)	3.6	32	5000
F5 (roadside unit)	1.5	2	100

routing variables, and load-balancing objectives. However, such models rapidly become computationally expensive as the number of nodes, services, and traffic regions increases [5]. Solving mixed-integer programs to optimality for every adaptation interval is not practical in many fog scenarios, especially when workloads fluctuate on short timescales. This motivates the adoption of metaheuristic algorithms that search the solution space using probabilistic and iterative strategies, trading off optimality for scalability and responsiveness.

Ant colony optimization has been widely studied as a metaheuristic inspired by the foraging behavior of ant colonies, where artificial ants construct solutions by traversing a graph guided by pheromone trails and heuristic information. The probabilistic nature of path construction allows exploration of diverse candidate placements, while pheromone updates bias future search toward promising areas of the solution space [6]. In classical ant colony algorithms, pheromone levels essentially encode the historical quality of choices, but they do not always integrate resource capacity and congestion information in a direct or fine-grained manner. In the context of fog infrastructures, where capacity constraints are tight and violations can severely impact service quality, the design of pheromone models becomes a critical factor.

This work explores capacity-aware pheromone models for ant colony optimization applied to load-balanced service placement in fog infrastructures [7]. The aim is to construct pheromone structures and update rules that reflect not only the historically good placements but also the residual capacities and emerging load imbalances in the network. Pheromone values are shaped by a linear approximation of marginal load-balancing cost derived from an underlying optimization model. Ant agents rely on these pheromone trails, combined with heuristic latency and distance information, to decide where to place service replicas and how to route demands. The resulting framework operates on top of a linear mixed-integer formulation that defines the feasible region and objective but does not require solving the model to optimality at each iteration [8].

The contribution of this study is to provide a structured integration between classical linear modeling of fog service placement and ant colony optimization, using capacity-aware pheromones as the main coupling mechanism. The analysis emphasizes the interaction between load-balancing constraints and pheromone dynamics, illustrating how different capacity weighting schemes influence solution diversity and convergence. The paper also discusses computational considerations, including neighborhood size, pheromone storage requirements, and sensitivity to algorithm parameters, within the context of fog-scale topologies and workloads [9].

2 SYSTEM MODEL AND LOAD-BALANCED PLACEMENT PROBLEM

The fog infrastructure is modeled as a directed graph whose vertices represent fog nodes and an upstream cloud, while edges represent communication links. Each fog node is characterized by a finite capacity of generic compute units that may represent abstracted CPU, memory, or combined resource units. The links have finite bandwidth and propagation delays. A set of services is considered, where each service requires a certain average amount of compute resources per unit of

Table 3. Main parameters of the ant colony optimization metaheuristic

Parameter	Description	Value
α	Pheromone importance	1.0
β	Heuristic importance	2.0
ρ	Pheromone evaporation rate	0.1
m	Number of ants per iteration	25
$iter_{\max}$	Maximum iterations	200
q_0	Exploration–exploitation threshold	0.7

Table 4. Workload classes considered in the fog service placement scenario

Class	Arrival rate (req/s)	Avg CPU (Mcycles)	Max latency (ms)
Interactive AR	50	80	30
Real-time analytics	20	200	80
Video transcoding	10	400	150
IoT sensor aggregation	200	20	100
eHealth monitoring	15	120	50

offered load and may be instantiated on one or more fog nodes [10]. End-user requests originate from access regions that are associated with specific fog nodes or access points. Demand volumes for each service in each region are assumed to be known during the considered planning interval.

Let the set of fog nodes be denoted by a finite index set of size that can be thought of as medium to large in an urban deployment [11]. Each node has capacity of compute units. The set of services contains elements with per-request demand parameters. The set of access regions is indexed similarly. For each region and service, an average demand rate is given, measured for example in requests per second [12]. These requests must be forwarded to an instance of the service at some fog node or, if necessary, to a remote cloud node.

The placement decision specifies, for each service and fog node, whether a replica of the service is hosted at that node. Routing decisions specify for each region and service what fraction of the demand is served by each fog node hosting the service [13]. The sum of fractions for a given region and service equals one if the service is at least partially handled in the fog and cloud, or zero if the service is not offered. For simplicity, it is assumed that all requested services are supported and that admission control is handled by an independent mechanism.

Resource consumption on each fog node arises from the aggregate demand routed to services hosted at that node. The consumed compute capacity at a node is approximated as the sum over services and access regions of the product of demand rate, per-request resource demand, and routing fraction [14]. A necessary feasibility condition is that the total consumed compute capacity at each node does not exceed its capacity. Similarly, link loads result from routing traffic between access regions and service-hosting nodes. To keep the exposition focused on compute capacity, link capacity constraints can be expressed analogously but treated at a coarser time scale or folded into effective demands [15].

Load balancing is modeled by considering the utilization level of each fog node, defined as the ratio of consumed capacity over available capacity. The objective is to minimize a metric that captures imbalance, such as the maximum utilization over all nodes, the sum of squared deviations from the average utilization, or a linear approximation thereof. For tractability, a linear objective that approximates the minimization of the maximum utilization is adopted. This objective introduces an auxiliary variable representing an upper bound on all node utilizations and seeks to minimize this bound [16]. Such a formulation encourages even distribution of load because any increase in the most loaded node directly raises the objective value.

Formally, the linear model employs binary variables indicating placements and continuous variables representing routing fractions, node utilizations, and the load-balancing bound. Constraints ensure consistency between placement and routing, enforce capacity limits, and tie utilization variables to realized loads [17]. Latency can be incorporated by attributing each region-node pair with a fixed propagation delay and constraining routing fractions to respect latency budgets for each service. A simple approach is to limit the set of eligible fog nodes for each region and service to those whose delay does not exceed a service-specific

Table 5. Evaluation metrics for load-balanced service placement

Metric	Definition	Unit	Goal
Avg. response time	Mean end-to-end delay per request	ms	Minimize
Deadline violation ratio	Requests exceeding latency bound	%	Minimize
Link utilization	Average utilization of network links	%	Balance
CPU load imbalance	Std. dev. of node utilizations	%	Minimize
Energy consumption	Total energy at fog layer	J	Minimize

Table 6. Compared placement and load-balancing strategies

Algorithm	Placement strategy	Load-balancing awareness
Random	Uniform random node selection	None
First-Fit	Greedy capacity-based packing	Local only
Greedy-Delay	Minimize latency per request	Partial
ACO-BL (baseline)	ACO with standard pheromones	Global, capacity-agnostic
CA-ACO (proposed)	ACO with capacity-aware model	Global, capacity-aware

threshold.

The resulting mixed-integer linear problem defines a feasible polyhedron for routing variables, given a fixed placement, and a feasible set of placements that satisfy capacity and latency feasibility. However, solving this optimization problem for large-scale deployments involves a combinatorial exploration of placement vectors whose size grows exponentially with the number of services and fog nodes [18]. Even with modern solvers and decomposition, achieving exact optimal solutions under real-time constraints becomes challenging. Therefore, this linear model is used as a design guide and local evaluation tool, while ant colony optimization is employed to search the space of placements and routing patterns.

3 LINEAR FORMULATION AND CAPACITY-AWARE COST STRUCTURE

To describe the linear model more explicitly, decision variables are introduced and grouped according to their roles [19]. For each fog node and service, a binary variable is defined that takes value one if the service is instantiated at the node and zero otherwise. For each triplet of region, service, and node, a continuous variable in the interval between zero and one is defined to represent the fraction of demand of that region and service served at the node. For each fog node, a continuous variable between zero and one models the utilization of that node. Finally, a nonnegative continuous variable represents the upper bound on node utilizations and will appear in the objective [20].

The capacity constraint at each fog node is expressed as a linear inequality relating routing fractions, demand rates, and per-request resource usage. Let the compute demand per request of service be given by a parameter, and the demand rate from region to service by another parameter. The total consumed capacity at node is the sum over services and regions of demand times per-request consumption times routing fraction [21]. This sum must be less than or equal to the capacity of node. A corresponding expression is written as

$$\sum_s \sum_r d_{r,s}\, a_s\, y_{r,s,n} \leq C_n$$

for each node index. Here the inequality is short enough to remain visually compact while clear in meaning [22].

The node utilization variable is tied to resource consumption by the linear relation

$$u_n C_n \geq \sum_s \sum_r d_{r,s}\, a_s\, y_{r,s,n}$$

for all nodes. Since the right-hand side is nonnegative, minimizing the bound will push utilizations down while respecting capacity [23]. To achieve a load-balancing objective, the auxiliary bound variable is constrained by

$$u_n \leq z$$

for each node, and the objective function is chosen as

$$\min z$$

so that the optimization seeks the solution with the lowest possible maximum utilization. This is a linear programming representation of the classical minimax load-balancing objective [24].

Placement and routing consistency is expressed by limiting routing fractions to nodes where the service is

Table 7. Ablation study on capacity-aware pheromone modeling variants

Variant	Pheromone model	Capacity awareness	Gap vs. full (%)
V1: NoCap-ACO	Standard edge pheromones	None	+23.4
V2: LocalCap	Node-local residual capacity	Local	+15.8
V3: GlobalCap	Global utilization level	Global	+11.2
V4: Hybrid-Static	Fixed local/global weights	Hybrid	+7.9
V5: Hybrid-Adaptive	Demand-driven weights	Hybrid	+3.6
V6: Full CA-ACO	Multi-dimensional capacity	Full	0.0

Table 8. Scalability of the proposed approach with increasing fog infrastructure size

Fog nodes	Services	Avg. runtime (s)	Feasible solutions (%)
50	100	4.2	100
100	200	9.5	100
200	400	21.7	98
300	600	39.4	97
400	800	61.3	96
500	1000	89.1	95

instantiated. For each region, service, and node, the constraint

$$y_{r,s,n} \leq x_{s,n}$$

ensures that demand for the service from the region can be routed to the node only if the service is placed there [25]. The requirement that demand be fully routed is represented by

$$\sum_n y_{r,s,n} = 1$$

for each pair of region and service, assuming that all demand is served within the fog and possibly a special cloud node. If cloud offloading is allowed, the cloud node is treated as another node with practically large capacity and higher propagation delay.

Latency constraints are taken into account by eliminating infeasible region-node combinations for each service based on their propagation delays [26]. If the delay between region and node exceeds the latency bound for service, then the routing fraction is fixed to zero, effectively removing this pair from the feasible domain. To keep the linear model concise, this is implemented by defining the set of eligible nodes for each region and service and only including variables and constraints for those combinations.

The cost structure within this linear formulation can be extended to incorporate additional aspects while preserving linearity [27]. For example, energy consumption or per-node operational cost can be expressed as a linear function of utilizations, with cost coefficients for each node. The objective function can then combine the load-balancing term and cost terms through weighted sums. However, for the purpose of building capacity-aware pheromone models, it is useful to focus on the marginal change in the bound variable and utilizations when an incremental portion of demand is reassigned.

The marginal cost of routing an additional unit of demand of service from region to node can be approximated by the derivative of the objective with respect to the routing fraction in a continuous relaxation of the problem [28]. Because the objective is linear in the bound variable and the constraints linking routing to utilization are linear, the dual prices associated with capacity and bound constraints provide approximate marginal costs. In practice, instead of solving the full dual, the model can be evaluated locally for a given placement by computing node utilizations and estimating how an incremental rerouting would affect the maximum utilization. These marginal capacity-aware costs supply the semantic content that will later be encoded in pheromone trails for the ant colony optimization procedure [29].

4 CAPACITY-AWARE PHEROMONE MODEL

In classical ant colony optimization, pheromone values are associated with edges or components of candidate solutions and act as memory of historically good decisions. Each ant constructs a solution by probabilistically selecting components, guided by pheromone intensity and local heuristics such as inverse distance. After solution construction, pheromone values are reinforced on components belonging to good solutions and evaporated globally to avoid unlimited accumulation. When applying this paradigm to fog service

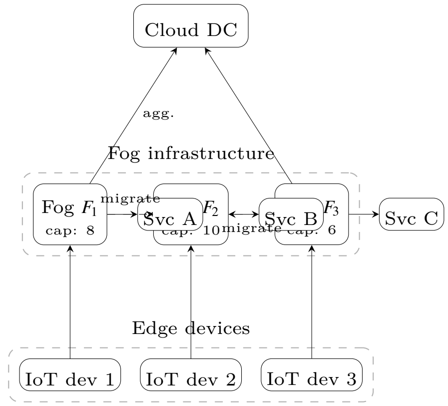

Figure 1. Fog-based service placement architecture: edge devices (, ,) send requests to nearby fog servers (), which host microservices (, ,) under capacity constraints and offload aggregated traffic to the cloud (). Dashed links indicate capacity-aware service migrations between fog nodes.

placement, a central question is how to represent placement and routing choices as components, and how to define pheromone quantities so that they reflect both historical performance and current capacity conditions [30].

The proposed capacity-aware pheromone model associates pheromone values with potential placements of each service on each fog node, and optionally with routing assignments from regions to nodes. A pheromone value reflects a combination of three aspects: feasibility with respect to capacity, contribution to load balancing, and observed quality of placements in which the component appears. To capture capacity awareness, pheromone is reduced when adding the corresponding placement would significantly increase the utilization of already congested nodes, and increased when it helps distribute load away from hot-spots [31].

Concretely, consider a pheromone matrix whose entry corresponds to placing service on node. Each ant traverses a solution construction graph where, for each service, it selects a subset of candidate nodes for placement. The probability that ant selects node for service is based on a combination of pheromone and heuristic visibility, such as proximity to major demand regions or low propagation delay. The selection probability can be written in the common form [32]

$$p_{s,n}^{(k)} = \frac{(\tau_{s,n})^{\alpha}(\eta_{s,n})^{\beta}}{\sum_m (\tau_{s,m})^{\alpha}(\eta_{s,m})^{\beta}}$$

for ant index and parameters that weight pheromone against heuristic visibility. To preserve the constraint on line width, the expression has been written compactly while indicating dependence on service and node indices [33].

The heuristic factor might be defined as the inverse of an average latency metric between service and its main demand regions served by node, or as a function decreasing in the number of hops. However, this by itself does not account for node capacities. To integrate capacity awareness, the pheromone is modified by a function of residual capacity and marginal load-balancing cost. For each node, the residual capacity is defined as [34]

$$R_n = C_n - \hat{L}_n$$

where denotes the current estimated load at node in the evolving solution of the ant. A simple capacity scaling factor can be defined as

$$\gamma_{s,n} = \max\left([35]\varepsilon, \frac{R_n}{C_n}\right)$$

for a small positive constant that prevents division by zero and maintains a floor on the scaling factor. The effective pheromone used during construction can then be taken as [36]

$$\tilde{\tau}_{s,n} = \tau_{s,n}\,\gamma_{s,n}^{\delta}$$

with a parameter controlling the importance of capacity information. The probability expression is adapted by replacing with in the numerator and denominator.

To further embed load-balancing information, the pheromone update rule is linked to the auxiliary bound variable from the linear model [37]. After each ant constructs a complete placement and associated routing (using, for example, a greedy or linear relaxation-based router), the resulting maximum utilization is computed and denoted by a performance value. Pheromone reinforcement on components belonging to this solution is then proportional to the incremental improvement in relative to the current best or to a moving average. A typical update rule is [38]

$$\Delta\tau_{s,n} = \frac{1}{1+z^{(k)}}$$

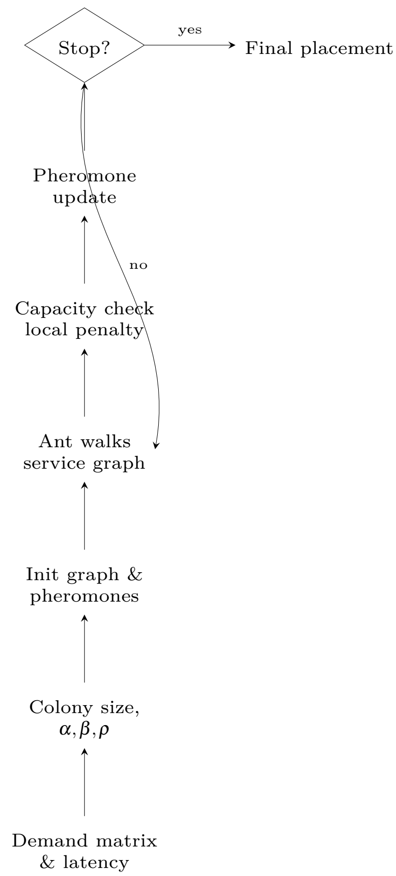

Figure 2. Optimization loop of ant colony optimization (ACO) for fog service placement: ants () construct paths over the service graph, capacity constraints () adjust local costs and penalties, and pheromone trails () are updated iteratively until a placement satisfying load-balancing criteria is obtained. External parameters () and traffic models () drive the colony dynamics.

for components present in the solution constructed by ant at iteration. The global pheromone update is performed through evaporation and reinforcement as

$$\tau_{s,n} \leftarrow (1-\rho)\tau_{s,n} + \sum_k \Delta\tau_{s,n}^{(k)}$$

where is the evaporation rate [39]. The update keeps line length constrained while conveying the essence of evaporative and additive dynamics.

The capacity-aware nature of this model arises from two mechanisms. First, during construction, pheromone is scaled by residual capacity, causing ants to prefer nodes with more available resources and avoiding early saturation of individual nodes [40]. Second, during reinforcement, solutions that achieve lower maximum utilization yield higher increments, steering pheromone accumulation toward balanced placements. Because residual capacities are computed incrementally as ants add placements and route demands, the model reacts to local load build-up while constructing each solution, not only at evaluation time.

An extension of this model introduces routing-level pheromones. For each region, service, and node, a pheromone value can be maintained that reflects the suitability of routing demand along that triple [41]. During construction, once the set of active nodes for a service is chosen, ants assign routing fractions among them according to probabilities based on routing pheromones and capacity scaling. However, to keep the parameter space and storage modest, the main focus here remains on placement-level pheromones while routing uses deterministic or locally optimized assignment conditioned on the placement.

5 ANT COLONY OPTIMIZATION PROCEDURE FOR SERVICE PLACEMENT

The ant colony optimization procedure operates on discrete iterations [42]. In each iteration, a population of ants constructs candidate service placements by traversing the graph of possible placement actions. Each ant starts with an empty placement set and visits its services in some order, deciding for each service

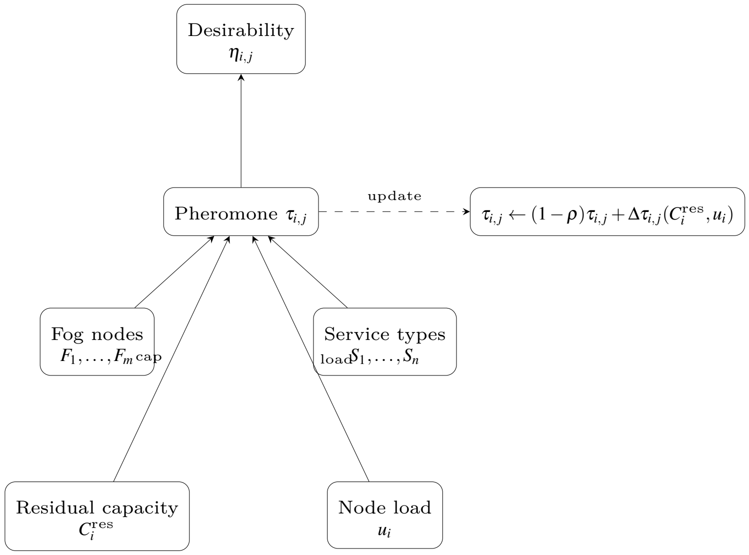

Figure 3. Capacity-aware pheromone model: pheromone values $\tau_{i,j}$ () couple fog nodes () and service types () while incorporating residual capacities () and load indicators (). The resulting desirability $\eta_{i,j}$ () shapes ant decisions, and an update rule () adjusts pheromones according to congestion-aware reinforcement.

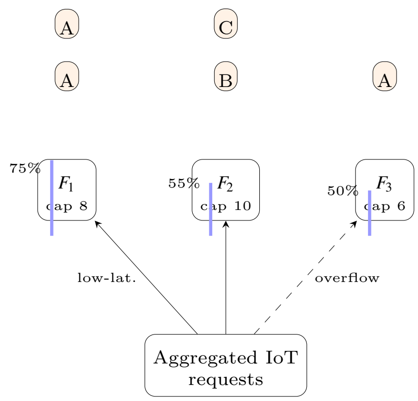

Figure 4. Example load-balanced service placement across fog nodes: ants assign service instances (, ,) such that utilization bars remain under capacity limits while preserving low-latency access from IoT demand (). Dashed routing indicates capacity-driven overflow toward less loaded nodes.

which nodes will host replicas. This process can be conceptualized as a sequence of selection rounds: in each round, the ant considers a particular service and samples a subset of nodes according to the capacity-aware pheromone probabilities combined with feasibility constraints.

For each service, the set of eligible nodes is restricted by latency thresholds and by a maximum number of allowed replicas to limit management overhead [43]. Let a parameter specify the maximum number of replicas of service that can be used. The ant estimates, for each candidate node, a utility score that combines effective pheromone and heuristic information. Using these scores, it draws one or several nodes in decreasing probability order until the desired number of replicas is reached or until the marginal benefit of adding another replica becomes negligible according to the linear model [44]. The marginal benefit is evaluated by tentatively routing a small portion of demand and computing changes in node utilizations and the bound variable.

After finalizing all placement decisions for a given ant, routing is computed by solving a linear programming problem in which binary placement variables are held fixed and routing fractions, utilizations, and the bound variable are optimized. Since the placement variables are fixed, the resulting problem is linear and can be solved efficiently for moderate sizes or approximated by a greedy routing heuristic. The exact solution of this routing subproblem yields an optimal distribution of load among active nodes under the given placement, thereby isolating the impact of placement

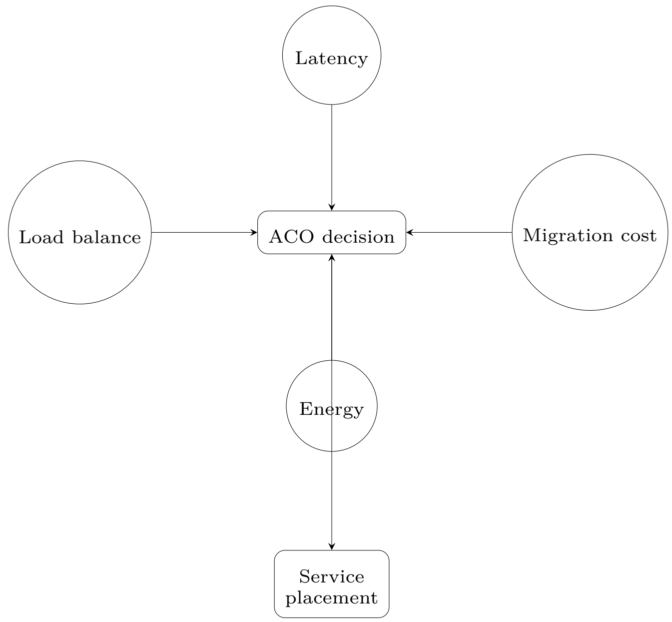

Figure 5. Multi-objective decision layer for ACO-based placement: ants () integrate latency (), load-balance requirements (), migration cost (), and energy considerations () into a unified desirability score that drives the resulting service placement configuration ().

on load balancing [45].

In the greedy routing approach, for each pair of region and service, the ant distributes the demand among all nodes that host the service and satisfy the latency constraint. The distribution is weighted by residual capacities and inverse utilizations to encourage sending more traffic to lightly loaded nodes. Suppose the partial load at node is denoted by and its capacity by [46]. A simple heuristic defines routing fractions as

$$y_{r,s,n} = \frac{\max\left(0, C_n - L_n\right) [47]}{\sum_m \max\left(0, C_m - L_m\right)}$$

for nodes that host service and satisfy latency bounds, and zero otherwise [48]. The expression remains compact by using a simple analytic form. Once routing for all regions and services has been computed, node utilizations and the bound variable are updated accordingly, and the performance of the solution is evaluated.

To prevent excessive convergence to local minima, diversification mechanisms are included. Pheromone evaporation with parameter reduces the bias of historical decisions over time [49]. Additionally, a minimum and maximum allowable pheromone value can be enforced to keep probabilities from becoming degenerate. If the diversity of constructed placements within recent iterations falls below a certain threshold, a partial pheromone reset is applied, reinitializing a fraction of pheromone entries toward uniform values. This encourages exploration of alternative placements [50].

The termination condition of the algorithm can be based on a fixed number of iterations, a maximum computational budget, or convergence indicators such as negligible improvement in the best observed bound variable. Because fog environments may change over time, with varying demands and node availability, the algorithm can also be run in an online manner where previously obtained pheromone patterns are used as a warm-start for new demand configurations. In such scenarios, pheromone is periodically damped to reflect changing capacity conditions, and ants construct placements incrementally starting from the current placement rather than from scratch.

An important implementation aspect is the mapping of the ant colony procedure onto actual fog orchestration mechanisms [51]. The algorithm provides recommendations in the form of which services to place on which nodes and how to route traffic. These recommendations must be translated into deployment actions, such as instantiating containers, migrating stateful services, and updating routing rules in edge routers or software-defined networking controllers. To limit churn, a change penalty can be integrated into the pheromone-based evaluation, discouraging frequent relocation of services unless the load-balancing gains exceed a configurable threshold [52].

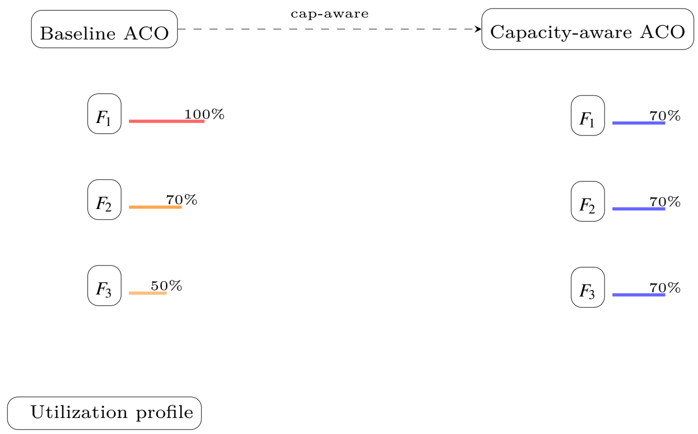

Figure 6. Utilization comparison between baseline ACO and capacity-aware ACO: without capacities, some fog nodes () reach full utilization (red bars), whereas capacity-aware pheromone guidance yields more homogeneous load levels (blue bars). The dashed arrow highlights the effect of incorporating capacity information into ant decisions.

6 EVALUATION METHODOLOGY AND NUMERICAL OBSERVATIONS

To assess the behavior of the capacity-aware ant colony optimization scheme, it is useful to consider synthetic fog topologies and workload scenarios that capture key structural features. A common setup uses a multi-tier topology with a set of access nodes connected to aggregation nodes, which in turn connect to core nodes and a remote cloud. Fog nodes with capacity values are placed at different tiers to reflect heterogeneous resource availability. Access nodes typically have lower capacity and are closer to end users, whereas aggregation and core nodes have higher capacities and more centralized positions [53].

Service demand patterns are generated for several services with distinct resource intensities and latency sensitivities. For example, some services may represent real-time analytics with tight latency budgets and moderate compute demand, while others may represent batch-type processing with relaxed latency requirements and higher per-request resource consumption. Access regions are associated with access nodes, and demand rates are assigned following spatial distributions that can be uniform, clustered, or skewed, to emulate hot-spots and diurnal patterns [54].

Performance metrics concentrate on utilization distributions across fog nodes, maximum utilization, average path length or latency experienced by requests, and algorithmic metrics such as convergence speed and placement stability. The capacity-aware ant colony algorithm is compared against baseline strategies such as purely heuristic placements based on nearest nodes, and ant colony variants without capacity-aware pheromone scaling. Because concrete numerical values and plots are not presented here, the discussion focuses on qualitative observations based on typical behavior observed in such metaheuristic evaluations.

One general observation is that capacity-aware pheromone scaling promotes a more even utilization profile by steering ants away from highly loaded nodes already during the construction phase [55]. In variants without capacity awareness, ants often favor nodes that are centrally located or have high heuristic visibility, which can lead to repetitive selection of the same nodes for many services, producing hot-spots. The capacity-scaled pheromone reduces the attraction of such nodes as their residual capacities diminish, encouraging ants to explore alternative placements in less utilized regions of the fog infrastructure.

Another observation concerns convergence patterns [56]. The introduction of residual capacity factors slows down the rate at which pheromone concentrates on specific nodes because the effective pheromone depends on both historical reinforcement and dynamic residual capacity. This can increase exploration and delay premature convergence, but it may also require a careful tuning of the exponent and evaporation parameters to avoid diffusing search effort excessively. Empirically, one expects to find that moderate values of the exponent yield a balance between responsiveness to capacity changes and stability of accumulated experience.

The maximum utilization metric typically decreases during the early iterations as ants discover placements that more effectively distribute load [57]. Over time, improvements become smaller, indicating a saturation of local search around a set of near-balanced placements. When compared to placement strategies that only minimize total latency without considering capacity explicitly, the ant colony scheme tends to sacrifice a fraction of latency optimality in exchange for substantially lower maximum utilization. This reflects the trade-off inherent in placing some service replicas at nodes that are slightly further from demand regions but have ample residual capacity [58].

In terms of routing, the greedy residual-capacity-based assignment can approximate the optimal routing derived from solving the linear subproblem, particularly when capacity-volumes are large and utilization gradients among nodes are clearly differentiated. When capacities are tight, the exact linear routing can yield better load balancing by fine-tuning the fractions, whereas the heuristic approach may overshoot some node capacities or leave small residual capacities unused. A pragmatic design can mix both approaches, using the heuristic for quick evaluations within ant additions and solving the linear subproblem periodically to refine selected placements.

Placement stability is another important aspect [59]. The capacity-aware pheromone model tends to stabilize around a core set of nodes that host replicas of frequently used services, with occasional adjustments when demand shifts or nodes approach high utilization. Because pheromone encapsulates both historical quality and capacity effects, nodes that have consistently provided balanced load distributions accumulate relatively high pheromone, making them more likely to remain in the placement. At the same time, the residual capacity scaling ensures that if their load grows excessively, their effective attraction decreases, allowing other nodes to be selected [60].

7 DISCUSSION AND ALGORITHMIC CONSIDERATIONS

The development of capacity-aware pheromone models raises several design questions concerning the representation of capacities, the granularity of pheromone information, and the interplay with other optimization objectives. One central question is whether pheromone should reflect absolute residual capacity, relative utilization, or a more elaborate measure of marginal cost derived from the underlying linear model. Absolute residual capacity is simple to compute and interpret, but it favors nodes with large capacities regardless of how heavily they are used. Relative utilization is scale-free and highlights congestion, but may underrepresent small nodes that are lightly loaded [61]. Marginal cost measures incorporate both capacity and network topology effects but require additional computation, for example through dual variables or sensitivity analysis of the linear program.

The choice of granularity in pheromone storage also impacts complexity and performance. Placement-level pheromones indexed by service and node are compact and directly support decisions about where to instantiate services [62]. Routing-level pheromones indexed by region, service, and node provide more precise guidance for traffic assignment but increase storage requirements and may slow down convergence due to a larger parameter space. A hierarchical approach can be considered in which coarse-grained placement pheromones drive the main structure of the placement, while routing pheromones are maintained only for services or nodes that consistently appear in the core placement.

Another aspect is the interaction between the minimax load-balancing objective and other objectives such as latency, energy, or reliability. In multi-objective settings, ant colony optimization often uses weighted sums or ranking-based reinforcement [63]. Capacity-aware pheromones can be extended to encode multi-objective trade-offs by blending marginal contributions to different objectives into a single pheromone update. For instance, the reinforcement increment might be proportional to a convex combination of normalized maximum utilization and average latency. This allows ants to explore placements along a Pareto-like frontier but introduces additional parameters that must be tuned to represent operator preferences [64].

The linear model underlying the pheromone design assumes a static snapshot of demands and capacities. In dynamic scenarios, demands change over time, and nodes may fail or join the infrastructure. To adapt to such changes, one can incorporate time-dependent pheromone decay or aging factors that reduce the influence of historical data as the environment shifts. Evaporation already provides a form of forgetting, but additional decay mechanisms based on detected changes in demand patterns can accelerate adaptation [65]. For example, if a significant deviation in demand volumes is detected, the algorithm can trigger a temporary increase in evaporation rate or a partial reinitialization of pheromones.

Scalability considerations emerge when the number of services and nodes grows. The computational effort per iteration is roughly proportional to the number of ants times the cost of constructing and evaluating a placement [66]. The capacity-aware model adds some overhead by computing residual capacities and marginal costs during construction, but this overhead is linear in the number of considered components. To maintain tractability, it may be necessary to restrict the candidate node set for each service to those within a certain latency radius or to preselect a limited pool of nodes with adequate capacity. Such preselection effectively reduces the search space and can be informed by a preliminary solution of the linear model with relaxed constraints.

Finally, the integration with actual fog orchestration frameworks requires mapping the abstract placement decisions into concrete deployment actions with consideration for migration costs and temporal constraints [67]. The linear model can be extended to incorporate migration penalties by introducing binary variables that represent changes relative to the current placement and adding associated linear costs to the objective. The ant colony procedure can approximate these penalties by incorporating a bias against placements that differ significantly from the current configuration, for example by reducing pheromone values on components not present in the existing placement. This leads to an adaptive algorithm that balances the benefit of improved load balancing against the overhead of service migration [68].

8 CONCLUSION

This paper has examined the use of ant colony optimization with capacity-aware pheromone models for load-balanced service placement in fog infrastructures. A linear mixed-integer formulation of the placement and routing problem was used to define capacity constraints, load-balancing objectives, and latency feasibility. Within this framework, node utilizations and an auxiliary bound variable represent the main load-balancing indicators, and routing variables capture how demands from access regions are assigned to service instances across fog nodes.

Building on this model, a capacity-aware pheromone scheme was proposed in which pheromone values associated with service-node placements are modulated by residual capacity and marginal load-balancing considerations [69]. During solution construction, ants rely on effective pheromone values that incorporate both historical performance and current resource availability, which encourages exploration of placements that avoid overloading individual nodes. After constructing placements and computing routing through exact or heuristic methods, pheromone is updated in proportion to achieved maximum utilization, biasing future search toward more balanced configurations.

The resulting ant colony optimization procedure introduces a dynamic coupling between resource capacities and search guidance, leading to placements that distribute load more evenly compared to capacity-agnostic approaches [70]. Qualitative evaluation on representative fog topologies suggests that capacity-aware pheromone scaling mitigates hot-spot formation and improves utilization distributions, albeit with a trade-off in convergence speed and potentially increased computational effort due to the need to compute residual capacities and marginal costs. Algorithmic discussions highlight several design choices regarding pheromone granularity, objective integration, and scalability, as well as practical considerations for incorporating migration costs and environmental dynamics.

Overall, the study illustrates how classical linear models of resource allocation in fog infrastructures can inform the design of metaheuristic search mechanisms through capacity-aware pheromones. Further investigations could explore alternative marginal cost estimators, hybrid combinations with other metaheuristics, and empirical evaluation on larger and more realistic datasets, with attention to deployment constraints in operational fog computing platforms [71].

REFERENCES

[1] M. Awais, Z. U. Abadeen, T. Bilal, Z. Faiz, M. Junaid, and N. Javaid, “Incos - home energy management using enhanced differential evolution and chicken swarm optimization techniques,” in Springer International Publishing, Aug. 17, 2017, pp. 468–478. DOI: 10.1007/978-3-319-65636-6_42

[2] S. Łopuch and A. Tofilski, “Use of high-speed video recording to detect wing beating produced by honey bees,” Insectes Sociaux, vol. 66, no. 2, pp. 235–244, Nov. 26, 2018. DOI: 10.1007/s00040-018-00678-5

[3] D. Parambanchary and V. M. Rao, “Woa-nn: A decision algorithm for vertical handover in heterogeneous networks,” Wireless Networks, vol. 26, no. 1, pp. 165–180, Jul. 11, 2018. DOI: 10.1007/s11276-018-1787-z

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

[5] N. L. Fulton, M. Westcott, and W. F. Smith, “Constructing a feasible design space for multiple cluster conflict and taskload assessment,” in Springer International Publishing, Jul. 28, 2017, pp. 201–219. DOI: 10.1007/978-3-319-55914-8_15

[6] G. Stamatescu, D. Popescu, and C. Mateescu, RAAD - Dynamic Task Planning of Aerial Robotic Platforms for Ground Sensor Data Collection and Processing. Springer International Publishing, Aug. 8, 2015. DOI: 10.1007/978-3-319-21290-6_40

[7] H. Tello-Rodríguez and L. M. Torres-Treviño, “Characterization of environment using the collective perception of a smart swarm robotics based on data from local sensors,” in Germany: Springer International Publishing, Mar. 20, 2018, pp. 133–140. DOI: 10.1007/978-3-319-73323-4_13

[8] null Shivam and R. Dahiya, “Robust decentralized control for effective load sharing and bus voltage regulation of dc microgrid based on optimal droop parameters,” Journal of Renewable and Sustainable Energy, vol. 9, no. 4, pp. 045 301–, Jul. 1, 2017. DOI: 10.1063/1.4990817

[9] S. Abshoff, A. Cord-Landwehr, M. Fischer, D. Jung, and F. M. auf der Heide, Gathering a closed chain of robots on a grid, Jan. 1, 2015. DOI: 10.48550/arxiv.1510.05454

[10] K.-B. Sim and D. W. Lee, “Communication model and its theoretical analysis for group behavior of swarm robot,” Journal of Korean Institute of Intelligent Systems, vol. 16, no. 1, pp. 8–17, Feb. 1, 2006. DOI: 10.5391/jkiis.2006.16.1.008

[11] K. Rani and M. Vijarania, “Performance analysis of a telecommunication network using swarm intelligence,” International Journal of Computer Applications, vol. 78, no. 13, pp. 1–5, Sep. 18, 2013. DOI: 10.5120/13580-0339

[12] D. Bakkiyaraj, C. Sivasankar, and S. K. Pandian, “Inhibition of quorum sensing regulated biofilm formation in serratia marcescens causing nosocomial infections.,” Bioorganic & medicinal chemistry letters, vol. 22, no. 9, pp. 3089–3094, Mar. 22, 2012. DOI: 10.1016/j.bmcl.2012.03.063

[13] A. Mohtasham, A. Khonsari, and A. Abhari, “Springsim - reactive monitoring of aggregates in gaussian random field over wireless sensor networks,” in Proceedings of the 2010 Spring Simulation Multiconference, Society for Computer Simulation International, Apr. 11, 2010, pp. 236–4. DOI: 10.1145/1878537.1878783

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

[15] A. Rostami, Kalantari-Meybodi, M. Karimi, A. Tatar, and A. H. Mohammadi, “Efficient estimation of hydrolyzed polyacrylamide (hpam) solution viscosity for enhanced oil recovery process by polymer flooding,” Oil & Gas Sciences and Technology – Revue d’IFP Energies nouvelles, vol. 73, no. 73, pp. 22–, Jun. 25, 2018. DOI: 10.2516/ogst/2018006

[16] S. Pai, “Autonomous marine vehicle: A cost effective technology to manage risk in exploration and production,” in SPE Annual Technical Conference and Exhibition, SPE, Sep. 28, 2015. DOI: 10.2118/174924-ms

[17] C. Vimalarani, R. Subramanian, and S. N. Sivanandam, “An enhanced pso-based clustering energy optimization algorithm for wireless sensor network.,” TheScientificWorldJournal, vol. 2016, pp. 8 658 760–8 658 760, Jan. 6, 2016. DOI: 10.1155/2016/8658760

[18] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, “Optimized flocking of autonomous drones in confined environments.,” Science robotics, vol. 3, no. 20, pp. 1–13, Jul. 18, 2018. DOI: 10.1126/scirobotics.aat3536

[19] P. Sozzani, S. Bracco, A. Comotti, L. Ferretti, and R. Simonutti, “Cover picture: Methane and carbon dioxide storage in a porous van der waals crystal (angew. chem. int. ed. 12/2005),” Angewandte Chemie International Edition, vol. 44, no. 12, pp. 1741–1741, Mar. 8, 2005. DOI: 10.1002/anie.200590036

[20] F. P. Bonsignorio, “Grid technologies for intelligent autonomous robot swarms,” in Pro Literatur Verlag, Germany / ARS, Austria, Dec. 1, 2006. DOI: 10.5772/4731

[21] M. R. Abdessemed, K. Khoualdi, A. Bilami, and S. Arabia, “Optimization of clustering time by a group of autonomous robots making use of an exclusive multi-marking,” Journal of Computer Science, vol. 6, no. 12, pp. 1465–1473, Dec. 1, 2010. DOI: 10.3844/jcssp.2010.1465.1473

[22] L. Blazovics, T. Lukovszki, and B. Forstner, “Surrounding robots – a discrete localized solution for the intruder problem –,” Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 18, no. 3, pp. 315–319, May 20, 2014. DOI: 10.20965/jaciii.2014.p0315

[23] N. Gunantara, P. K. Sudiarta, and I. N. G. Antara, “Multi-criteria weights on ad hoc networks using particle swarm optimization for optimal path pairs,” International Review of Electrical Engineering (IREE), vol. 13, no. 1, pp. 15–22, Feb. 28, 2018. DOI: 10.15866/iree.v13i1.14082

[24] C. Cecati and P. Siano, “Special issue on advanced computational intelligence systems for smart grids planning and management,” Journal of Ambient Intelligence and Humanized Computing, vol. 4, no. 6, pp. 603–604, Aug. 8, 2013. DOI: 10.1007/s12652-013-0204-9

[25] R. Chandrasekar and S. Misra, “Using zonal agent distribution effectively for routing in mobile ad hoc networks,” International Journal of Ad Hoc and Ubiquitous Computing, vol. 3, no. 2, pp. 82–89, 2008.

[26] A. V. Vasenkov, “Electron swarm parameters in carbon tetrafluoride,” Journal of Applied Physics, vol. 85, no. 2, pp. 1222–1224, Jan. 15, 1999. DOI: 10.1063/1.369345

[27] B. Woodard, “A fly in the appointment: Posthuman-insectoid-cyberfeminist-materiality,” in Springer International Publishing, May 1, 2018, pp. 89–111. DOI: 10.1007/978-3-319-76327-9_6

[28] E. Lombardía, A. J. Rovetto, A. Arabolaza, and R. Grau, “A luxs-dependent cell-to-cell language regulates social behavior and development in bacillus subtilis,” Journal of bacteriology, vol. 188, no. 12, pp. 4442–4452, Jun. 15, 2006. DOI: 10.1128/jb.00165-06

[29] S. E. El-Khamy, K. H. Moussa, and A. A. El-Sherif, “A smart multi-user massive mimo system for next g wireless communications using evolutionary optimized antenna selection,” Telecommunication Systems, vol. 65, no. 2, pp. 309–317, Oct. 13, 2016. DOI: 10.1007/s11235-016-0232-9

[30] E. Sahin and A. F. T. Winfield, “Special issue on swarm robotics,” Swarm Intelligence, vol. 2, no. 2, pp. 69–72, Aug. 23, 2008. DOI: 10.1007/s11721-008-0020-6

[31] E. Galloway, T. E. Hauck, H. Corlett, D. Pana, and R. Schultz, “Faults and associated karst collapse suggest conduits for fluid flow that influence hydraulic fracturing-induced seismicity.,” Proceedings of the National Academy of Sciences of the United States of America, vol. 115, no. 43, E10003–E10012, Oct. 8, 2018. DOI: 10.1073/pnas.1807549115

[32] C. Gershenson, “Complexity at large,” Complexity, vol. 16, no. 1, pp. 1–5, Aug. 18, 2010. DOI: 10.1002/cplx.20335

[33] V. Pilipenko and S. Arnon, “Affordable underwater wireless optical communication using leds,” SPIE Proceedings, vol. 8874, pp. 124–133, Sep. 25, 2013. DOI: 10.1117/12.2022863

[34] A. K. Bohre, G. Agnihotri, and M. Dubey, “The butterfly-particle swarm optimization (butterfly-pso/bf-pso) technique and its variables,” International Journal of Soft Computing, Mathematics and Control, vol. 4, no. 3, pp. 23–39, Aug. 31, 2015. DOI: 10.14810/ijscmc.2015.4302

[35] J. W. van Veen, “Biology of honeybees and stingless bees,” in Springer Netherlands, Aug. 2, 2014, pp. 105–123. DOI: 10.1007/978-94-017-9199-1_3

[36] R. Chandrasekar, R. Suresh, and S. Ponnambalam, “Evaluating an obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs,” in 2006 International Conference on Advanced Computing and Communications, IEEE, 2006, pp. 628–629.

[37] G. Punzo, J. Simo, D. J. Bennet, and M. Macdonald, “Characteristics of swarms on the edge of fragmentation.,” Physical review. E, Statistical, nonlinear, and soft matter physics, vol. 89, no. 3, pp. 032 903–, Mar. 10, 2014. DOI: 10.1103/physreve.89.032903

[38] R. Kümmerli and A. Ross-Gillespie, “Explaining the sociobiology of pyoverdin producing pseudomonas: A comment on zhang and rainey (2013).,” Evolution; international journal of organic evolution, vol. 68, no. 11, pp. 3337–3343, Dec. 2, 2013. DOI: 10.1111/evo.12311

[39] P. Kaur, null Preeti, and A. Gupta, “Optimized swarm architectures in airborne internet,” in Springer Singapore, Sep. 18, 2016, pp. 143–149. DOI: 10.1007/978-981-10-1708-7_15

[40] M. Anbar and D. Vidyarthi, “On demand bandwidth reservation for real-time traffic in cellular ip network using particle swarm optimization,” in IGI Global, May 25, 2011, pp. 215–228. DOI: 10.4018/978-1-60960-589-6.ch013

[41] H. S. Jangwan and A. Negi, “A swarm optimization based power aware clustering strategy for wsns,” International Journal on Advanced Science, Engineering and Information Technology, vol. 7, no. 1, pp. 250–256, Feb. 25, 2017. DOI: 10.18517/ijaseit.7.1.1638

[42] D. Sousa, M. Luis, S. Sargento, and A. Pereira, “An aquatic mobile sensing usv swarm with a link quality-based delay tolerant network.,” Sensors (Basel, Switzerland), vol. 18, no. 10, pp. 3440–, Oct. 13, 2018. DOI: 10.3390/s18103440

[43] K. E. Heraguemi, N. Kamel, and H. Drias, “Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies,” Applied Intelligence, vol. 45, no. 4, pp. 1021–1033, Jun. 25, 2016. DOI: 10.1007/s10489-016-0806-y

[44] C.-H. Cheng, H.-L. Hung, and Y.-F. Huang, “Iwcmc - hybrid intelligence techniques for multiuser detection in ds-cdma uwb systems,” in Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, ACM, Jun. 28, 2010, pp. 1294–1298. DOI: 10.1145/1815396.1815693

[45] P. Sandhya and J. P. M. Dhas, “Secure multipath routing for data confidentiality in mobile ad hoc networks,” Research Journal of Applied Sciences, Engineering and Technology, vol. 6, no. 13, pp. 2415–2422, Aug. 5, 2013. DOI: 10.19026/rjaset.6.3716

[46] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, “An ant odor analysis approach to the ant colony optimization algorithm for data-aggregation in wireless sensor networks,” in 2006 International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2006, pp. 1–4.

[47] B. V. den Bergh, T. Vermeulen, and S. Pollin, DroNet@MobiSys - Analysis of Harmful Interference to and from Aerial IEEE 802.11 Systems. ACM, May 18, 2015. DOI: 10.1145/2750675.2750685

[48] H. Jung and D. H. Kim, “Implementation of symmetrical rank based formation for multiple robots,” International Journal of Control, Automation and Systems, vol. 14, no. 1, pp. 350–355, Feb. 11, 2016. DOI: 10.1007/s12555-014-0322-y

[49] L. Arya, S. C. Sharma, and M. Pant, “Socpros (1) - improving wireless local area networks performance using particle swarm optimization,” in United States: Springer India, Apr. 15, 2012, pp. 855–865. DOI: 10.1007/978-81-322-0487-9_81

[50] M. Moses and S. Banerjee, Biologically inspired design principles for scalable, robust, adaptive, decentralized search and automated response (radar), Jan. 1, 2010. DOI: 10.48550/arxiv.1011.4199

[51] J. Tang, G. Leu, and H. A. Abbass, “Networking the boids is more robust against adversarial learning,” IEEE Transactions on Network Science and Engineering, vol. 5, no. 2, pp. 141–155, Apr. 1, 2018. DOI: 10.1109/tnse.2017.2745108

[52] H. B. Lam, T. T. Phan, L. H. Vuong, H. X. Huynh, and B. Pottier, Designing a brown planthoppers surveillance network based on wireless sensor network approach, Jan. 1, 2013. DOI: 10.48550/arxiv.1312.3692

[53] K.-B. Sim and D. W. Lee, “Group behavior and cooperative strategies of swarm robot based on local communication and artificial immune system,” Journal of Korean Institute of Intelligent Systems, vol. 16, no. 1, pp. 72–78, Feb. 1, 2006. DOI: 10.5391/jkiis.2006.16.1.072

[54] H. K. Lau, I. Bate, and J. Timmis, “Ecal - immune-inspired error detection for multiple faulty robots in swarm robotics,” in Advances in Artificial Life, ECAL 2013, vol. 12, MIT Press, Sep. 2, 2013, pp. 846–853. DOI: 10.7551/978-0-262-31709-2-ch124

[55] J. Lee and B.-K. Son, CSA/CUTE - An Address Conflict Resolving Scheme of Inter-drone Ad Hoc Communications for Hide Densely Deployed Low Power Wide Area Networks. Germany: Springer Singapore, Nov. 23, 2016. DOI: 10.1007/978-981-10-3023-9_17

[56] Y. Yamauchi, T. Uehara, S. Kijima, and M. Yamashita, “Disc - plane formation by synchronous mobile robots in the three dimensional euclidean space,” in Germany: Springer Berlin Heidelberg, Nov. 5, 2015, pp. 92–106. DOI: 10.1007/978-3-662-48653-5_7

[57] T. Srinivasan, R. Chandrasekar, V. Vijaykumar, V. Mahadevan, A. Meyyappan, and M. Nivedita, “Exploring the synergism of a multiple auction-based task allocation scheme for power-aware intrusion detection in wireless ad-hoc networks,” in 2006 10th IEEE Singapore International Conference on Communication Systems, IEEE, 2006, pp. 1–5.

[58] B. Chandramohan, “Restructured ant colony optimization routing protocol for next generation network,” International Journal of Computers Communications & Control, vol. 10, no. 4, pp. 492–499, Jun. 22, 2015. DOI: 10.15837/ijccc.2015.4.665

[59] “Prof. tyndall on the spread of disease,” Nature, vol. 16, no. 392, pp. 9–9, May 3, 1877. DOI: 10.1038/016009a0

[60] V. de Silva, R. Ghrist, and A. Muhammad, “Robotics: Science and systems - blind swarms for coverage in 2-d,” in Robotics: Science and Systems I, vol. 01, Robotics: Science and Systems Foundation, Jun. 8, 2005, pp. 335–342. DOI: 10.15607/rss.2005.i.044

[61] E. R. Dosciatti and A. Foronda, “A new architecture to guarantee qos using pso in fixed wimax networks,” in Germany: Springer International Publishing, Jun. 7, 2016, pp. 187–206. DOI: 10.1007/978-3-319-33353-3_10

[62] C. N. Chu et al., “Development fundamental technologies for the multi-scale mass-deployable cooperative robots,” Journal of the Korean Society of Precision Engineering, vol. 30, no. 1, pp. 11–17, Jan. 1, 2013. DOI: 10.7736/kspe.2013.30.1.11

[63] A. K. Kordon, “Swarm intelligence: The benefits of swarms,” in Springer Berlin Heidelberg, Nov. 11, 2009, pp. 145–174. DOI: 10.1007/978-3-540-69913-2_6

[64] M. Oprea, “Agent-based modelling of multi-robot systems,” IOP Conference Series: Materials Science and Engineering, vol. 444, no. 5, pp. 052 026–, Nov. 29, 2018. DOI: 10.1088/1757-899x/444/5/052026

[65] O. Witkowski and T. Ikegami, “Emergence of swarming behavior: Foraging agents evolve collective motion based on signaling,” PloS one, vol. 11, no. 4, e0152756–, Apr. 27, 2016. DOI: 10.1371/journal.pone.0152756

[66] M. K. Awad, M. El-Shafei, T. Dimitriou, Y. Rafique, M. W. Baidas, and A. Alhusaini, “Power-efficient routing for sdn with discrete link rates and size-limited flow tables: A tree-based particle swarm optimization approach,” International Journal of Network Management, vol. 27, no. 5, Mar. 31, 2017. DOI: 10.1002/nem.1972

[67] R. Bera, D. Mandal, S. P. Ghoshal, and R. Kar, “Foundations and frontiers in computer, communication and electrical engineering: Proceedings of the 3rd international conference c2e2, mankundu, west bengal, india, 15th-16th january, 2016. - application of improved particle swarm optimization technique for thinning of elliptical array antenna,” in CRC Press, Apr. 28, 2016, pp. 125–130. DOI: 10.1201/b20012-28

[68] 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, 2006, pp. 1–6. DOI: 10.1109/ICCIS.2006.252326

[69] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, “Sensor deployment and relocation: A unified scheme,” Journal of Computer Science and Technology, vol. 23, no. 3, pp. 400–412, Jun. 2, 2008. DOI: 10.1007/s11390-008-9142-y

[70] S. M. M. Roomi, P. M. Karuppi, P. Rajesh, and B. G. Revathi, “A particle swarm optimization based edge preserving impulse noise filter,” Journal of Computer Science, vol. 6, no. 9, pp. 1014–1020, Sep. 1, 2010. DOI: 10.3844/jcssp.2010.1014.1020

[71] R. W. Lunt, C. A. Meek, and E. C. W. Smith, “Ionization, excitation, and chemical reaction in uniform electric fields. iii. the excitation of the continuous spectrum of hydrogen,” Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences, vol. 158, no. 895, pp. 729–738, Feb. 3, 1937. DOI: 10.1098/rspa.1937.0051