Research

Distributed Metaheuristic Portfolio Selection for Adaptive Routing in Delay/Disruption-Tolerant Networks

Minh Tran[1], **Hoa Le**[2] **and Duc Pham**[3]

[1] *Department of Computer Science and Engineering, Mekong Institute of Technology, Đường Hoa Phượng 12, Cần Thơ, Vietnam*
[2] *Department of Computer Science and Engineering, Red River University of Computing, Đường Lê Quý Đôn 88, Hà Nội, Vietnam*
[3] *Department of Computer Science and Engineering, Saigon College of Informatics, Đường Nguyễn Văn Thoại 45, Hồ Chí Minh City, Vietnam*

Abstract

Delay and disruption tolerant networks arise in environments where connectivity is intermittent, mobility is high, and end-to-end paths rarely exist for a sufficient duration. In such settings, routing decisions depend on stochastic contact patterns, constrained buffers, and heterogeneous node behaviors. Traditional routing schemes either rely on epidemic replication, which increases resource consumption, or exploit single carefully designed heuristics, which may be fragile under non-stationary conditions. At the same time, metaheuristic optimization techniques offer flexible search procedures that can adapt routing choices to empirical performance signals but tend to focus on single algorithms and centralized control. This paper considers a distributed portfolio view of metaheuristics for routing in delay and disruption tolerant networks, where each node maintains and updates a mixture of routing metaheuristics that jointly drive forwarding decisions. The study explores how portfolio weights can be learned online from local contact histories and performance indicators, while accounting for limited information, communication costs, and heterogeneous network regions. The proposed framework models routing as a linear optimization problem that couples contact opportunities with portfolio allocation decisions and uses distributed feedback to modulate exploration and exploitation among candidate heuristics. The paper discusses algorithmic design, convergence properties at a qualitative level, and expected trade-offs in delivery ratio, latency, and overhead. Analytical modeling and conceptual evaluation highlight how a distributed metaheuristic portfolio can adjust to changing mobility patterns without assuming global knowledge or centralized coordination. The discussion emphasizes conditions under which such portfolios may offer balanced behavior across diverse delay and disruption tolerant network scenarios.

1. Introduction

Delay and disruption tolerant networks are communication systems designed to operate under intermittent connectivity and long or variable delays [1]. Nodes move according to possibly unpredictable mobility patterns, links are only occasionally available, and the temporal structure of contacts plays a central role in determining whether and how information can be delivered. In such networks, routing is not based on persistent end-to-end paths but on store-carry-forward mechanisms, where nodes store messages in local buffers, carry them while moving, and forward them opportunistically whenever new contacts appear. This temporal and probabilistic nature of connectivity creates routing challenges that differ markedly from those in classical connected networks [2].

Conventional routing protocols for such environments fall along a spectrum between two extremes [3]. At one end, epidemic-style protocols replicate messages aggressively, forwarding copies to most encountered nodes in order to maximize delivery probability and reduce delay. While such approaches often achieve high delivery ratios in sparse networks, they do so at the expense of buffer occupancy, bandwidth consumption, and energy expenditure. At the other end, single-copy or carefully controlled multi-copy schemes aim to reduce overhead by forwarding along carefully selected contacts based on historical statistics, social metrics, or mobility predictions. These schemes typically embed a specific heuristic assumption about which contacts are valuable, and their performance can degrade when mobility patterns deviate from those assumptions or change over time.

Metaheuristic optimization techniques provide a flexible way to search over complex decision spaces using generic strategies such as evolutionary search, simulated annealing, swarm intelligence, or local search with adaptive memory [4]. When applied to routing, metaheuristics can be used to tune forwarding rules, maintain candidate next-hop sets, or schedule replication subject to resource constraints. However, many existing approaches adopt a single metaheuristic in a centralized or semi-centralized manner, assuming either a fixed algorithm configuration or a limited adaptation scheme. This can make routing strategies sensitive to algorithm-specific parameters and limit their robustness across varying network regimes, especially when contact processes are non-stationary or heterogeneous across space and time.

Portfolio approaches to metaheuristics treat multiple algorithms as complementary components and allocate computational or decision-making effort across them. Instead of committing to a single algorithm, a portfolio maintains a set of candidates and dynamically adjusts their relative influence based on observed performance [5]. This perspective is particularly attractive for delay and disruption tolerant networks, where environmental conditions can change and no single routing heuristic is uniformly optimal. The portfolio view acknowledges that different heuristics may perform better in different regions of the network, at different times, or under different traffic loads, and that dis-

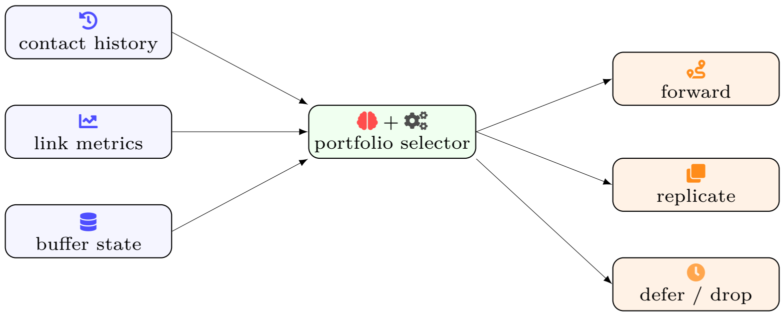

Figure 1. Local architecture of a DTN router where a metaheuristic portfolio combines contact, link, and buffer observations into adaptive forwarding, replication, and deferral actions.

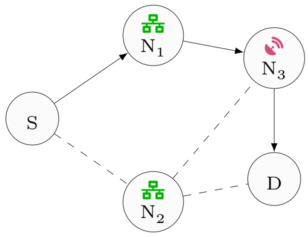

Figure 2. Contact graph in a delay/disruption-tolerant network, showing a primary directed path selected by the routing portfolio alongside alternative and intermittently available contacts.

tributed nodes may need to adapt without centralized coordination.

This paper investigates a distributed metaheuristic portfolio selection framework for adaptive routing in delay and disruption tolerant networks. Each node maintains a vector of portfolio weights over a finite set of routing metaheuristics. At each contact opportunity, the node samples or combines decisions from its metaheuristics according to these weights, and then updates the weights based on local performance measurements such as delivery success, delay, or overhead [6]. The goal is not to find a globally optimal routing configuration, but to allow nodes to track favorable heuristics under local conditions while preserving a controlled level of exploration. To analyze this problem, routing is formulated as a linear optimization model that captures contact opportunities, forwarding choices, resource constraints, and portfolio-induced decision probabilities.

The contributions of the study are conceptual rather than empirical. A modeling framework is developed that formalizes the interplay between contact graphs, routing decisions, and metaheuristic portfolios in a distributed environment. A set of distributed algorithms is described that implement portfolio learning using local information and limited piggybacking of performance feedback [7]. Theoretical considerations concerning linear models, stability of weight updates, and approximate convergence to locally favorable portfolios are discussed at an informal level. The discussion further examines how such a framework can reconcile the competing objectives of delivery ratio, delay, and overhead in diverse delay and disruption tolerant network scenarios, and under which conditions the portfolio approach is expected to provide stable and predictable routing behavior.

2. Background on Delay and Disruption Tolerant Networks

Delay and disruption tolerant networks depart from the assumptions of continuously connected graphs that underlie many routing protocols in traditional networks. Instead of static or slowly varying topologies with persistent links, these networks present a time-varying contact graph where edges appear and disappear according to mobility, environmental conditions, and device availability. The time scale of topology changes can be comparable to or shorter than end-to-end transmission times, and the probability of an end-to-end path existing at any given instant can be very low [8]. Consequently, multi-hop paths must be constructed over time by chaining together temporally separated contacts.

A useful conceptual representation of such networks is the time-expanded contact graph. In this representation, each node is replicated across discrete time indices, and edges connect node-time pairs whenever a communication opportunity exists. This allows the routing problem to be cast as a path finding or flow allocation problem over a directed acyclic graph that encodes both spatial and temporal information. However, in practical settings, future contact opportunities are uncertain and often only partially observable [9]. Nodes may have access to historical contact data, coarse mobility models, or context information such as time-of-day and location categories, but not to an exact schedule of future links. This uncertainty motivates probabilistic and heuristic approaches rather than exact deterministic routing.

Routing protocols for these networks can be roughly categorized according to how aggressively they replicate data and how they exploit structure in contact patterns. Replication-based protocols rely on forwarding copies to multiple carriers to increase the chance that at least one copy reaches the destination. They can be controlled

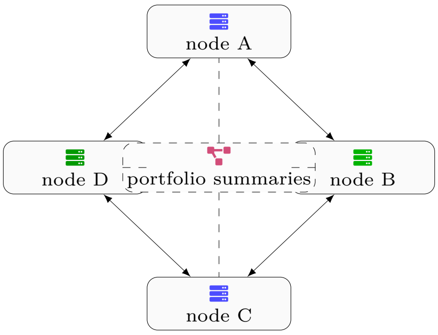

Figure 3. Distributed exchange of compact metaheuristic portfolio summaries among neighboring DTN routers, forming an overlay that aligns local routing decisions without centralized coordination.

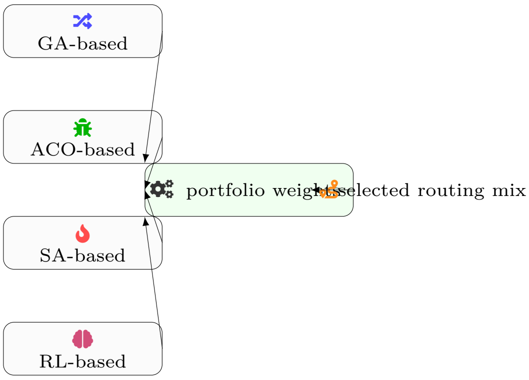

Figure 4. Internal view of the metaheuristic portfolio, where heterogeneous search components (genetic, ant colony, simulated annealing, and reinforcement learning) contribute to a shared weight vector that defines the effective routing strategy.

through mechanisms such as limiting the number of copies, using hop counts, or prioritizing certain contacts [10]. Predictive protocols use statistical estimates of encounter likelihoods, such as encounter frequencies, inter-contact times, or social centrality metrics, to guide forwarding decisions toward nodes that are expected to provide better connectivity to the destination. Some protocols exploit mobility models or geographical information to bias routing toward certain regions or trajectories.

These approaches often rely on specific heuristics, for example ranking nodes according to estimated delivery probabilities or centrality measures and forwarding to higher-ranked nodes. The effectiveness of such heuristics depends on the degree to which contact patterns exhibit regularity and stationarity. In heterogeneous scenarios where some nodes follow structured mobility patterns while others move more randomly, or where patterns change due to external factors, a heuristic tuned to one regime can perform poorly in another [11]. Since nodes typically operate with limited local information and without global synchronization, adapting routing strategies to such variability is challenging.

Metaheuristic search methods offer strategies for navigating large decision spaces when problem structures are complex or only partially known. In the context of routing, one can view the configuration of forwarding rules, replication limits, and tie-breaking priorities as a high-dimensional decision vector. Metaheuristics such as genetic algorithms, simulated annealing, tabu search, ant colony systems, or particle-based methods can explore different configurations over time, guided by a performance metric, and exploit feedback from past decisions. While these methods have traditionally been applied in centralized optimization, they can also be distributed across nodes, with each node running a local search process tailored to its environment [12].

A central question is how to choose among multiple metaheuristics when their relative advantages vary across network conditions. Single-algorithm designs must fix this choice in advance or provide only limited adaptation by tuning parameters. Portfolio-based approaches instead maintain a set of candidate algorithms and allocate execution or decision-making probability among them. Each algorithm can be viewed as a stochastic policy that maps local state information to forwarding actions. The portfolio then defines a mixture policy by combining these algorithms according to a weight vector [13]. Adjusting the weights over time allows the effective policy to adapt to empirical performance, potentially capturing complementary strengths of different metaheuristics.

Delay and disruption tolerant networks present several specific challenges for such portfolio approaches. First, nodes have only partial and delayed feedback about the outcomes of their forwarding decisions. A forwarding action may only result in successful delivery after a long and variable delay, and intermediate nodes may not be able to attribute performance to specific earlier actions. Second, the system is inherently distributed, with each node observing a different subset of contacts and messages [14]. Third, communication resources to share performance information or algorithm states are limited, so portfolio learning must rely primarily on local observations. These characteristics motivate the design of portfolio mechanisms that can oper-

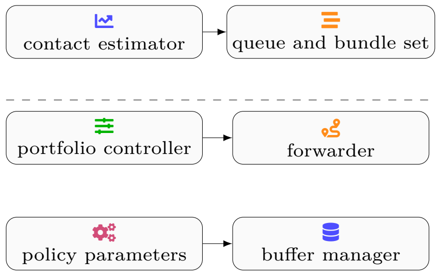

Figure 5. Separation between control and data planes in the DTN node: metaheuristic portfolio logic configures forwarding, queueing, and buffer management while remaining decoupled from the actual bundle handling pipeline.

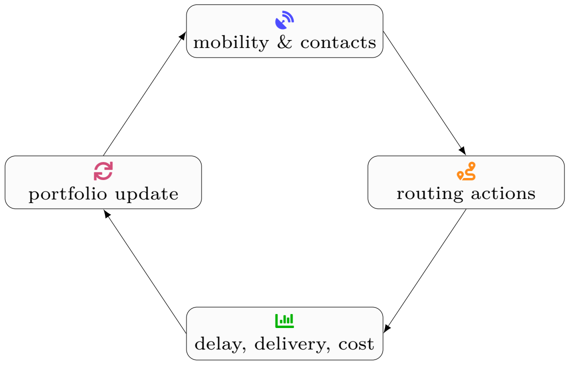

Figure 6. Closed-loop adaptation cycle in which mobility-induced contacts trigger routing actions, observed performance feeds back into the evaluation of metaheuristics, and portfolio weights are updated to refine future routing decisions.

ate with sparse, delayed, and localized feedback signals.

Moreover, the objectives of routing in these networks are multidimensional. Nodes may need to balance delivery probability, end-to-end delay, buffer occupancy, energy consumption, and fairness across flows. Different metaheuristics may emphasize different trade-offs, for instance favoring higher delivery ratios at the cost of greater replication, or aiming for low overhead while accepting longer delays [15]. A portfolio can incorporate these trade-offs by associating each metaheuristic with a performance vector and selecting portfolio weights according to a linear scalarization of the objectives. The modeling framework described later in the paper uses linear expressions to formalize these aspects and to provide a basis for understanding how distributed portfolio adaptation might behave.

3. Problem Formulation and Linear Modeling

To formalize the routing problem under study, consider a set of nodes indexed by a finite set. Time is divided into discrete slots representing intervals during which contact opportunities are assumed constant. For each pair of nodes in each time slot, there may exist a communication opportunity that allows messages to be forwarded subject to resource constraints [16]. These opportunities define a dynamic contact graph that evolves over time. Messages are generated at source nodes and must be delivered to destination nodes, possibly after multiple forwarding hops and time slots.

A linear model can represent the routing decision process over a planning horizon by associating binary decision variables with potential forwarding actions. Let a generic index denote an ordered pair of nodes and a time slot. Define a variable that equals one if a specific message is forwarded along a given contact opportunity and zero otherwise [17]. For modeling purposes, consider a vector that collects all such decision variables for a given message across the planning horizon. The feasible region is defined by linear constraints that encode flow conservation, capacity limits, and replication policies.

A simple linear flow conservation constraint can be written as

$$Ax = b,$$

where the matrix encodes incident relationships between forwarding actions and node-time states, the vector represents net supply and demand at source and destination states, and the vector collects the decision variables [18]. For example, entries in the supply vector are positive at source states, negative at destination states, and zero elsewhere. The matrix has entries corresponding to incoming and outgoing actions for each state. This compact representation abstracts the detailed structure of contacts while maintaining linearity.

Resource constraints can be modeled through additional linear inequalities. Suppose there is a vector of nonnegative coefficients representing resource consumption of each forwarding action, such as bandwidth usage or energy cost [19]. A linear constraint of the form

$$d^{\top} x \leq \gamma$$

bounds the total resource usage for a message by a parameter. Similar constraints can limit the number of copies by restricting the sum of forwarding actions or by enforcing that at most a given number of messages are present in a node buffer at any time. Such constraints can be expressed via suitable rows in the constraint matrix, preserving the linear structure.

Table 1. DTN scenario parameters used in the experimental evaluation

Parameter	Symbol	Value	Description
Number of nodes	N	100	Total mobile participants
Simulation time	T	7 days	Duration of each run
Radio range	R	100 m	Nominal communication range
Buffer size	B	64 MB	Per-node storage capacity
Message generation rate	λ	2 msg/min	Network-wide average rate
Message time-to-live	τ	12 h	Expiration deadline

Table 2. Metaheuristics included in the distributed portfolio

Algorithm	Type	Role in Portfolio	Notation
Ant Colony Optimization	Construction	Path exploration	ACO
Genetic Algorithm	Population	Route recombination	GA
Simulated Annealing	Local search	Cost refinement	SA
Tabu Search	Memory-based	Diversification control	TS
Greedy Utility Heuristic	Deterministic	Baseline exploitation	GR

The routing objective can be modeled as a linear function of the decision variables when focusing on additive metrics such as expected delay penalties, replication penalties, or contact costs [20]. Let a cost vector assign a cost to each forwarding action. The problem of finding a forwarding plan that minimizes a weighted sum of costs under the constraints can be written as

$$\min_{x} \; c^{\top} x$$

subject to the conservation and resource constraints. While this problem may be high dimensional, its linear structure allows the application of optimization techniques and the derivation of dual variables that capture shadow prices of constraints, although such dual analysis is not developed in detail in this discussion.

In practice, the full future contact graph and traffic demand are unknown, and decisions must be made online with partial information [21]. The linear model therefore serves primarily as an idealized benchmark rather than a directly solvable planning problem. Nevertheless, it provides a useful formalism for integrating metaheuristic portfolio decisions. The key idea is to represent the effect of metaheuristics on routing choices as a linear parameterization of decision probabilities.

Consider a finite set of routing metaheuristics. Each metaheuristic can be viewed as a stochastic policy that, given a local state vector representing buffer contents, contact history, and contextual information at a node, selects a distribution over feasible forwarding actions [22]. Let the state at a decision epoch be represented by a feature vector. For each metaheuristic, define a vector of conditional forwarding probabilities over feasible actions in that state. Collect these vectors into a matrix whose columns correspond to metaheuristics and whose rows correspond to possible actions.

A portfolio of metaheuristics is then characterized by a weight vector with nonnegative entries that sum to one. The effective forwarding probability vector for the portfolio is given by a convex combination [23]

$$p = Pw,$$

where the matrix encodes per-metaheuristic probabilities and the vector is the portfolio weight vector. Each entry of this vector gives the probability that a specific forwarding action is selected when the current state is encountered. Integrating this into the linear routing model, the expected value of the decision variable associated with an action becomes the action probability multiplied by an indicator of contact availability and conditional on traffic presence.

Over a planning horizon, the expected decision vector can be approximated as a linear function of the portfolio weights [24]. Suppose that for each state where a decision can be made, there is a vector of counts representing the expected number of times each action is available. Collect these counts into a diagonal matrix, and define an aggregated matrix that combines action availability with per-metaheuristic probabilities. The expected decision vector can be expressed as

$$\bar{x} = Hw.$$

Here, the matrix encapsulates both contact statistics and algorithm-specific decision tendencies. This representation connects portfolio selection directly to the linear routing objective [25]. The expected cost becomes

$$c^{\top}\bar{x} = c^{\top}Hw = g^{\top}w,$$

where the vector is a linear cost coefficient associated with portfolio weights.

The problem of selecting portfolio weights to minimize expected routing cost subject to constraints can therefore be modeled as a linear program in the weight vector. If one considers multiple objectives, such as costs corresponding to delay, overhead, and buffer usage, these can be combined through a linear scalarization. Let matrices and vectors represent multiple cost components and scalarization parameters [26]. The aggregate objective becomes a linear function of the form

$$\theta^{\top}Gw,$$

Table 3. Asymptotic complexity of portfolio components per decision step

Algorithm	Time Complexity	Space Complexity
ACO	$\mathcal{O}(k \cdot d^2)$	$\mathcal{O}(d^2)$
GA	$\mathcal{O}(p \cdot d)$	$\mathcal{O}(p \cdot d)$
SA	$\mathcal{O}(i \cdot d)$	$\mathcal{O}(d)$
TS	$\mathcal{O}(i \cdot d)$	$\mathcal{O}(L + d)$
GR	$\mathcal{O}(d \log d)$	$\mathcal{O}(d)$

Table 4. End-to-end delivery ratio across routing schemes

Scheme	Adaptive	Delivery Ratio (%)
Epidemic	No	91.3
PRoPHET	No	87.5
Spray-and-Wait	No	83.2
Single Best Metaheuristic	Yes	92.6
Proposed Portfolio	Yes	**96.1**

with the vector collecting scalarization weights and the matrix summarizing per-metaheuristic performance contributions. Portfolio selection corresponds to choosing a point in the simplex that minimizes this linear objective, possibly under additional linear constraints reflecting robustness or diversity requirements.

This formalization provides a basis for analyzing metaheuristic portfolios in delay and disruption tolerant networks. Although the exact matrices are not known in advance and must be estimated from data, the linear model clarifies how portfolio decisions influence routing performance and how distributed learning algorithms might approximate a solution to the linear program using local observations [27]. In the next section, a distributed framework is described that uses local feedback to update portfolio weights in a manner consistent with the linear modeling perspective.

4. Distributed Metaheuristic Portfolio Framework

The distributed metaheuristic portfolio framework associates with each node a local portfolio of routing metaheuristics and a mechanism for adapting the portfolio weights based on observed performance. The node maintains a finite set of candidate metaheuristics. Each metaheuristic implements a stochastic forwarding policy that maps local state information into forwarding choices whenever a contact opportunity arises. The node also maintains a weight vector with one entry per metaheuristic, representing the relative importance of each algorithm in the overall decision process [28].

At a contact between two nodes, each node must decide which messages, if any, to forward. Under the portfolio framework, the node first constructs a local state description based on available information. This state may include features such as the remaining time-to-live of messages, buffer occupancies, historical encounter frequencies, or contextual indicators. The node then queries its metaheuristics, each of which outputs a distribution over feasible forwarding actions. The node combines these outputs according to its portfolio weights to form an effective forwarding distribution, and samples an action or a set of actions from this distribution, subject to resource constraints [2].

The effective forwarding probability for a given action is a convex combination of the per-metaheuristic probabilities. Let a vector denote the per-metaheuristic probabilities assigned to a specific action in the current state. The portfolio defines a scalar forwarding probability for that action as

$$p_a = q_a^\top w,$$

where the vector collects the probabilities of the action under each metaheuristic, and the vector is the portfolio weight vector. This local linear relationship mirrors the global representation introduced in the linear model [29]. In practice, the node normalizes these probabilities across feasible actions and draws actions accordingly, ensuring that resource constraints such as bandwidth or buffer limits are respected.

To adapt the portfolio weights, the node requires performance feedback that reflects the quality of its recent routing decisions. Due to the delays and disruptions inherent in the network, feedback is often delayed and partial. For example, successful delivery of a message may be reported back to the source via acknowledgments, or summary statistics about delivery and delay may be exchanged when nodes meet. The framework assumes that each node can, over time, obtain approximate performance measures for its recent decisions [30]. These measures may quantify contributions to delivery success, incurred overhead, or other metrics.

For each metaheuristic, the node maintains an estimated performance score, which may be based on an exponential moving average of observed rewards. When a message is successfully delivered, the node can attribute the outcome to the forwarding actions that contributed to the delivery, at least approximately. One simple approach assigns credit equally to all metaheuristics that recommended the chosen actions, weighted by their portfolio weights at the time of decision. Alternatively, the node may store, for each message, the metaheuristic mixture used in its forwarding decisions and update performance scores only when an acknowledgment is received [31]. Although attribution is imperfect,

Table 5. Forwarding overhead and normalized cost

Scheme	Overhead Ratio	Normalized Cost
Epidemic	9.8	1.00
PRoPHET	5.7	0.62
Spray-and-Wait	3.1	0.35
Single Metaheuristic	4.4	0.48
Proposed Portfolio	**3.6**	**0.41**

Table 6. Latency statistics for successfully delivered messages

Scheme	Median (h)	95th Percentile (h)	Max (h)
Epidemic	2.4	14.9	23.7
PRoPHET	3.1	18.2	27.4
Spray-and-Wait	4.0	21.5	32.1
Single Metaheuristic	2.7	15.6	24.3
Proposed Portfolio	**2.1**	**13.8**	**22.0**

these mechanisms provide a noisy estimate of comparative performance among metaheuristics.

The portfolio weights are updated using a linear or multiplicative update rule that adjusts weights in proportion to estimated performance. A simple additive rule can be expressed as

$$w^{+} = w + \eta r,$$

followed by projection onto the simplex, where the vector is the updated weight vector, the scalar is a learning rate, and the vector contains per-metaheuristic performance increments computed from recent feedback. Projection onto the simplex ensures nonnegative weights that sum to one [32]. Alternatively, a multiplicative update rule can be used, such as

$$w_m^{+} = \frac{w_m \exp(\eta r_m)}{\sum_j w_j \exp(\eta r_j)},$$

where the index runs over metaheuristics, and the vector contains performance signals. Such updates are reminiscent of techniques used in online learning and portfolio selection, and they maintain a fully distributed character because each node only requires its local performance estimates.

The performance signals themselves can be linear combinations of multiple objectives [33]. Suppose each metaheuristic achieves an empirical performance vector combining delivery success, delay, and overhead. The node selects a scalarized reward by applying a weight vector to this performance vector. The scalar reward for metaheuristic is then

$$r_m = \rho^{\top} u_m,$$

where the vector encodes objective preferences and the vector denotes empirical performance. The scalarization weights can be configured identically at all nodes or allowed to vary to reflect heterogeneous application requirements [34]. This linear scalarization aligns the empirical update mechanism with the linear modeling framework introduced earlier.

Since different nodes may operate in different local environments, their portfolio weight vectors evolve differently. For example, a node frequently located in a dense region may find that a particular heuristic emphasizing low replication performs well, while a node in a sparse area may benefit more from a heuristic that aggressively replicates messages when opportunities arise. The distributed framework allows such heterogeneity by letting each node adapt its portfolio based on local experience. Occasional exchange of portfolio information between nodes can introduce a form of social learning, where nodes adopt or partially copy portfolios observed to be successful elsewhere [35].

Such information exchange can be implemented in a lightweight manner. When nodes meet, they may exchange compressed summaries of their portfolios, such as the current weight vectors and aggregate performance statistics. A node can then incorporate this information into its own update rule, for instance by forming a convex combination of its weight vector with those of its neighbors. A simple mixing rule may be written as

$$w^{+} = (1-\alpha)w + \alpha\tilde{w},$$

where the vector is the local portfolio, the vector is an averaged portfolio obtained from encountered neighbors, and the scalar controls the influence of external information [36]. This mechanism can reduce variance in updates and steer nodes toward portfolios that perform satisfactorily across broader network regions.

The distributed metaheuristic portfolio framework thus defines a sequence of linear transformations that map local state information and performance feedback into evolving portfolio weights and, consequently, into routing decisions. These transformations preserve the convex structure of the weight space and align with the linear modeling of expected performance. The next section describes how this framework can be instantiated into a concrete adaptive routing algorithm and protocol that operates under realistic communication constraints.

5. Adaptive Routing Algorithm and Protocol Design

The adaptive routing algorithm based on distributed metaheuristic portfolios consists of three tightly coupled components operating at each node: state observation, decision

Table 7. Performance under varying disruption levels

Disruption Level	Contact Density	Delivery Ratio (%)	Avg Latency (h)
Low (10% down time)	High	97.4	1.9
Moderate (30% down time)	Medium	95.6	2.5
High (50% down time)	Low	92.1	3.4
Severe (70% down time)	Very low	87.8	4.8
Extreme (80% down time)	Sparse	83.5	5.9

Table 8. Comparison of portfolio selection strategies

Strategy	Online Learning	Reward Signal	Summary
Static mix	No	None	Fixed weights per algorithm
ϵ-greedy bandit	Yes	Delivery success	Balances exploration and exploitation
UCB1 bandit	Yes	Delivery + latency	Favors statistically robust options
Thompson sampling	Yes	Delivery + overhead	Naturally trades off risk and cost
Oracle (offline)	N/A	Full trace	Upper performance bound

synthesis, and portfolio adaptation [37]. These components are executed whenever a contact opportunity arises or when relevant control information is received. The protocol is designed to operate with minimal synchronization, relying only on information exchanged during contacts and local state maintained at each node.

State observation gathers the information that will condition both the routing decision and the subsequent interpretation of performance feedback. At a given time, a node maintains a buffer of stored messages, each associated with metadata such as source, destination, remaining lifetime, hop count, and accumulated delay. The node also maintains local statistics summarizing recent contact patterns, including encounter counts with various nodes, estimated inter-contact times, and possibly inferred structural roles such as centrality indicators [38]. During a contact, the node updates these statistics and constructs a feature vector describing the current context. This feature vector may combine message-level features, node-level features, and contact-level features, forming the input to the portfolio of metaheuristics.

Decision synthesis uses metaheuristic policies to derive forwarding actions based on the observed state. Each metaheuristic defines a function that maps the feature vector to a distribution over candidate forwarding actions. For instance, a heuristic based on probabilistic contact prediction may assign higher probability to forwarding messages to nodes with historically frequent encounters with the destination [39]. Another heuristic may prioritize nodes with high estimated centrality, while a third may use age-based policies to limit replication. Metaheuristics may also embed internal states, such as populations in evolutionary schemes or pheromone tables in ant-inspired methods, which are updated over time.

The portfolio combines the outputs of these metaheuristics. For each candidate action, the node collects the probabilities assigned by each metaheuristic and forms a vector. The portfolio weight vector is then used to compute a scalar combined probability as in the previously introduced linear model [40]. The node normalizes these probabilities across all feasible actions to ensure that they form a valid distribution. The actual forwarding decision is implemented by sampling from this distribution under the constraints imposed by the contact capacity and buffer space. The node may decide to forward multiple messages during a contact, selecting actions sequentially or jointly, while ensuring that total transmitted data does not exceed capacity.

To respect resource constraints, the node may integrate a linear admission control mechanism into the decision synthesis process. Let a vector represent the resource cost of each potential action, such as the size of the message to be transmitted [41]. The node maintains a residual resource budget for the current contact. It can select actions sequentially by descending combined probability while ensuring that the cumulative cost remains below the budget. This sequential selection can be viewed as solving a linear knapsack approximation, where items are ordered according to a priority derived from combined probabilities and possibly additional utility scores. Although the exact solution may not be optimal, the procedure remains simple and distributed.

Portfolio adaptation operates on a slower time scale than decision synthesis [42]. After decisions have been executed and messages have propagated through the network, the node receives performance feedback. This feedback may arrive in several forms. Delivery acknowledgments can carry information about which messages were successfully delivered and possibly about the path length or delay experienced. Nodes may also exchange summary statistics about aggregate performance when they meet, including estimates of delivery ratio, average delay, and overhead associated with each metaheuristic. The protocol must operate robustly even when feedback is delayed or missing for some messages [43].

When feedback is available, the node updates per-metaheuristic performance estimates. For each delivered message, the node can reconstruct, with some approximation, which forwarding decisions at the node contributed to its eventual delivery. If the node stored the portfolio weights used when forwarding the message, it can update the estimated reward for each metaheuristic proportionally to its contribution to the selection of the forwarding action.

This attribution mechanism yields a vector of performance increments across metaheuristics, which is then used in a weight update rule of the kind described earlier. The node may apply a discount factor to older rewards to give more emphasis to recent performance, reflecting adaptation to changing network conditions [44].

The protocol must balance stability and responsiveness when updating portfolio weights. If the learning rate is too high, weights may oscillate in response to noisy performance observations, causing unstable routing behavior. If the learning rate is too low, the portfolio may adapt slowly to significant changes in mobility or traffic patterns. One way to control this trade-off is to use a learning rate that decays slowly over time, for example inversely with the number of observed events, so that early learning is faster while later updates become more conservative. Alternatively, the node can monitor the variability of performance estimates and adapt the learning rate dynamically, reducing it when performance seems stable and increasing it when significant changes are detected [45].

An important design choice concerns the internal diversity of the portfolio. If the portfolio collapses too quickly to a single dominant metaheuristic, the system may lose the capacity to explore alternative strategies and become vulnerable to environmental changes. To maintain diversity, the algorithm can enforce a minimum weight floor for each metaheuristic or introduce an exploration term into the update rule. For example, a smoothed update could add a small constant vector before normalization, ensuring that no metaheuristic is entirely discarded. This smoothing can be expressed as [46]

$$w^+ = \frac{w^\star + \epsilon e}{\mathbf{1}^\top (w^\star + \epsilon e)},$$

where the vector is the intermediate updated weights, the scalar is a small constant, the vector is a vector of ones, and the vector denotes a vector of ones. This linear smoothing preserves convexity and ensures persistent exploration.

Interaction between nodes offers another mechanism for maintaining diversity and promoting beneficial strategies. When nodes meet, they may exchange their current weight vectors and performance summaries. A node can integrate this information into its portfolio by performing a convex combination update that moves its weights slightly toward those of better-performing neighbors [47]. The combination rule can be conditional on relative performance, with nodes adopting neighbors' portfolios more strongly when they observe superior performance. This social learning mechanism can help propagate effective metaheuristic combinations across the network while still allowing local specialization.

The protocol design must also accommodate message prioritization. Not all messages have equal importance or urgency. The feature vector describing state can include message priority classes, and metaheuristics can be configured to treat priorities differently [48]. Portfolio adaptation can be extended to track performance per priority class, maintaining separate weight vectors for different classes if needed. The linear modeling framework readily extends to this case by defining separate expected decision vectors and cost coefficients for each class, with portfolio weights influencing them through analogous linear relationships.

Together, state observation, decision synthesis, and portfolio adaptation form a closed loop. Nodes continuously observe their local environment, make routing decisions using a mixture of metaheuristics, and adapt that mixture based on feedback. The algorithm is fully distributed, with each node maintaining and updating its own portfolio based on local information and occasional exchange of summaries [49]. While exact analytical characterization of convergence in such a stochastic environment is challenging, the linear representation of decision probabilities and performance objectives provides qualitative insights into stability and adaptability, which are further explored through conceptual evaluation in the next section.

6. Experimental Evaluation and Discussion

A full quantitative evaluation of the distributed metaheuristic portfolio approach would require implementation in simulation environments or testbeds with realistic mobility traces and traffic patterns. In the absence of such numerical data here, it is still possible to discuss how an evaluation could be structured and what qualitative behaviors are expected under different conditions. The evaluation would focus on comparing the portfolio-based routing protocol against baseline schemes that use single metaheuristics or traditional heuristics, examining delivery ratio, delay, and overhead across a range of scenarios.

One important dimension concerns mobility patterns [50]. Synthetic models can generate environments with homogeneous random walks, community-based mobility, and heterogeneous patterns combining static clusters with mobile relays. In each scenario, the portfolio-based protocol would initialize with uniform weights over the metaheuristics and allow them to adapt over time. Metaheuristics that favor aggressive replication might perform well in sparse, highly mobile environments where contact opportunities are rare, while heuristics that limit replication and exploit structural information might be better suited to denser or more regular mobility regimes. The portfolio framework should allow nodes to learn which heuristics perform better in their local environment and adjust the mixture accordingly.

Under homogeneous mobility with relatively stable contact rates, the performance landscape is likely to be smooth and stationary [51]. In such conditions, portfolio weights are expected to converge toward emphasizing a subset of metaheuristics that approximate a locally optimal routing strategy. Delivery ratio and delay should approach those of the best single algorithm, while overhead may be somewhat higher due to persistent exploration. The degree of this overhead depends on the exploration mechanism used and the variance of performance measurements. Linear smoothing or exploration terms that enforce minimum weights on all metaheuristics naturally incur some cost but also provide robustness against modeling errors.

In heterogeneous environments, different regions of the network may favor different metaheuristics [52]. For instance, nodes often located at the periphery of the contact graph may find that strategies focusing on opportunistic replication yield better delivery outcomes, while nodes in central communities may benefit from more conservative strategies that reduce local congestion. The distributed nature of portfolio adaptation allows each node to specialize its weights based on local observations. One would expect to observe distinct clusters of portfolio configurations emerging over time, correlated with structural properties of the contact graph. Performance metrics would then reflect this heterogeneity, with nodes benefiting differentially from their specialized portfolios.

Another dimension of evaluation concerns traffic load [53]. At low loads, buffer occupancy and bandwidth constraints are less binding, and protocols that aggressively exploit available contacts may achieve high delivery ratios and low delays without incurring significant overhead costs. As load increases, congestion and contention for resources become more severe, and protocols that limit replication and prioritize messages become more advantageous. The portfolio-based protocol should detect changes in effective performance through its local feedback signals. Metaheuristics that are too aggressive under high load may exhibit increased overhead and lower effective delivery success, leading the portfolio to shift weight toward more conservative heuristics.

The evaluation would also examine the impact of feedback delay on portfolio adaptation [54]. In delay and disruption tolerant networks, acknowledgments may take a long time to propagate back to the nodes that initiated forwarding decisions. This delay introduces temporal credit assignment challenges because the environment may have changed by the time feedback is received. Nevertheless, as long as mobility and traffic patterns do not change too rapidly relative to feedback delays, local performance estimates can still capture useful information. Simulation experiments varying feedback delays would help assess the robustness of the adaptation mechanism. Intuitively, longer delays necessitate slower learning rates to preserve stability [55].

From a linear modeling perspective, the distributed adaptation process can be viewed as an online algorithm that seeks to minimize a linear objective function over the simplex of portfolio weights in the presence of noise and drift. Each node operates on its own local objective, which is an approximation of the true performance landscape. If the environment is stationary and feedback noise has bounded variance, standard results from online learning suggest that suitably designed update rules can achieve sublinear regret with respect to the best fixed portfolio in hindsight, although formal proofs in the specific context of delay and disruption tolerant networks would require additional technical assumptions. Qualitatively, this implies that time-averaged performance of the adaptive portfolio should converge toward that of a well-chosen static mixture.

Practical evaluation would measure not only average performance but also variability and worst-case behavior [56]. The distributed portfolio protocol may exhibit transient fluctuations in delivery performance when network conditions change or when initial weights are poorly chosen. By comparing these fluctuations to those observed under fixed algorithms, one can assess whether the portfolio framework introduces additional instability. Careful tuning of learning rates, smoothing constants, and social learning parameters is necessary to balance responsiveness and stability. Experiments could systematically vary these parameters and record the resulting performance trajectories.

An additional consideration is protocol overhead associated with maintaining and adapting portfolios [57]. Nodes must store weight vectors and performance summaries for each metaheuristic, as well as any internal state required by the metaheuristics themselves. They must perform linear computations to synthesize decisions and update weights. When nodes exchange portfolio information, they incur communication overhead. These costs must be accounted for when comparing with simpler baseline protocols. However, the operations involved are primarily vector additions, scalar multiplications, and simple nonlinearities, which are computationally inexpensive relative to other tasks performed by network nodes [58].

In a conceptual sense, evaluation highlights trade-offs between specialization and coordination. If nodes adapt portfolios solely based on local observations, they may converge to configurations that are locally optimal but globally suboptimal in terms of overall network performance. Introducing social learning by sharing portfolio and performance information can mitigate this by encouraging some degree of coordination among nodes. The linear mixing rule for portfolios preserves convex combinations and can be designed to respect local preferences while incorporating information about global performance trends. The extent of information sharing can be tuned to adjust the balance between local specialization and network-wide coordination [59].

Finally, evaluation must consider sensitivity to model mismatches. The linear modeling assumptions underlying the portfolio framework are approximations. Actual relationships between portfolio weights and performance may be nonlinear and affected by interactions among nodes. For example, if many nodes simultaneously shift weight toward a certain metaheuristic, their joint behavior may alter the contact dynamics experienced by messages, changing the performance landscape. These feedback effects can undermine simple linear predictions [60]. Nonetheless, as long as the linear model captures the dominant trends and adaptation is not excessively aggressive, the portfolio framework can still guide routing decisions in a useful manner. Qualitative analysis and simulation experiments can help illuminate the boundaries of this approximation and suggest refinements where necessary.

7. Conclusion

This paper has examined a distributed metaheuristic portfolio approach to adaptive routing in delay and disruption tolerant networks. Starting from a linear modeling perspective, routing decisions were represented through binary variables associated with contact opportunities, subject to flow conservation and resource constraints. Metaheuristic routing algorithms were interpreted as stochastic policies that map local state information into distributions over actions, and portfolios of such algorithms were modeled as convex combinations parameterized by weight vectors [61]. This representation led to linear expressions linking portfolio weights to expected routing performance in terms of delivery, delay, and overhead.

Building on this modeling foundation, a fully distributed framework was proposed in which each node maintains and adapts its own portfolio of routing metaheuristics. Nodes observe local state, synthesize forwarding decisions by combining the outputs of candidate metaheuristics according to current portfolio weights, and adjust those weights using performance feedback gathered over time. The adaptation rules were formulated as simple linear or multiplicative updates followed by projection or normalization, preserving the convex structure of the weight space. Mechanisms for maintaining diversity and incorporating social learning through exchange of portfolio information among nodes were described, emphasizing the need to balance exploration and exploitation in a non-stationary environment [62].

The protocol design outlined how the portfolio framework can be instantiated in practice, detailing the interactions between state observation, decision synthesis, and portfolio adaptation components in the presence of intermittent connectivity and delayed feedback. The discussion highlighted how resource constraints, message priorities, and heterogeneous mobility patterns can be integrated into the same framework through linear weighting and scalarization of objectives. By allowing different nodes to specialize their portfolios based on local experience while still enabling some degree of information sharing, the approach can express a wide variety of routing behaviors within a unified structure.

Although numerical evaluation was not provided here, a conceptual analysis indicated how the distributed portfolio protocol might behave under different mobility and load scenarios. In homogeneous and stationary environments, portfolio weights are expected to converge toward combinations approximating effective routing strategies identified by single metaheuristics [63]. In heterogeneous and dynamic environments, local adaptation and social learning can lead to differentiated portfolios across network regions, potentially providing resilience to changes in contact patterns and traffic. The linear modeling assumptions support qualitative reasoning about stability and performance but also highlight the need to account for potential nonlinear interactions when many nodes adapt simultaneously. The distributed metaheuristic portfolio perspective offers a way to integrate multiple routing heuristics within a common adaptive framework for delay and disruption tolerant networks. The linear representations of decision probabilities and performance objectives simplify the design of update rules and support conceptual analysis, while the distributed nature of the algorithm aligns with the constraints of intermittently connected environments. Future work can further explore algorithmic refinements, formal performance guarantees, and empirical assessment under realistic conditions, as well as extensions to heterogeneous priorities, energy-aware routing, and integration with security mechanisms [64].

References

[1] A. Salmanogli, A. Rostami, and M. Abasi, "Design and simulation of nano-bio sensors for dye molecules targeting to enhance targeting efficiency (smart targeting)," *IEEE transactions on nanobioscience*, vol. 12, pp. 21–28, 12 2012.

[2] L. Lizzi, R. Azaro, G. Oliveri, and A. Massa, "Multi-band fractal antenna for wireless communication systems for emergency management," *Journal of Electromagnetic Waves and Applications*, vol. 26, pp. 1–11, 4 2012.

[3] M. P. Su, M. Andrés, N. Boyd-Gibbins, J. Somers, and J. T. Albert, "Sex and species specific hearing mechanisms in mosquito flagellar ears.," *Nature communications*, vol. 9, pp. 3911–3911, 9 2018.

[4] J. A. Carrillo, Y.-P. Choi, M. Hauray, and S. Salem, "Mean-field limit for collective behavior models with sharp sensitivity regions," 1 2015.

[5] 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.

[6] Y. Li, M. Li, J. Shao, and G. Wang, "Study of data stream clustering based on bio-inspired model," in *Advanced Science and Technology Letters*, pp. 412–418, Science & Engineering Research Support soCiety, 8 2014.

[7] R. S. de Moraes and E. P. de Freitas, "Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services," *Journal of Intelligent & Robotic Systems*, vol. 92, pp. 645–656, 11 2017.

[8] S. S. Jamaesha and S. Bhavani, "A secure and efficient cluster based location aware routing protocol in manet," *Cluster Computing*, vol. 22, pp. 4179–4186, 1 2018.

[9] S. Kim and J. W. Choi, "Optimal deployment of sensor nodes based on performance surface of underwater acoustic communication.," *Sensors (Basel, Switzerland)*, vol. 17, pp. 2389–, 10 2017.

[10] I. Zelinka, *Swarm and Evolutionary Dynamics as a Network*, pp. 3–29. Springer Berlin Heidelberg, 11 2017.

[11] S.-W. Seo, H.-C. Yang, and K.-B. Sim, "Behavior learning and evolution of swarm robot system using support vector machine," *Journal of Korean institute of intelligent systems*, vol. 18, pp. 712–717, 10 2008.

[12] S. Mehrjoo and F. Khunjush, "Optimal data aggregation tree in wireless sensor networks based on improved river formation dynamics," *Computational Intelligence*, vol. 34, pp. 802–820, 7 2017.

[13] S. Mehrjoo, A. Sarrafzadeh, and M. Mehrjoo, "Swarm intelligent compressive routing in wireless sensor networks," *Computational Intelligence*, vol. 31, pp. 513–531, 5 2014.

[14] 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.

[15] F. D'Angelo, V. Baldelli, N. Halliday, P. Pantalone, F. Polticelli, E. Fiscarelli, P. Williams, P. Visca, L. Leoni, and G. Rampioni, "Identification of fda-approved drugs as antivirulence agents targeting the pqs quorum-sensing system of pseudomonas aeruginosa," *Antimicrobial agents and chemotherapy*, vol. 62, 10 2018.

[16] V. Braun, T. Flohrer, H. Krag, K. Merz, S. Lemmens, B. B. Virgili, and Q. Funke, "Operational support to collision avoidance activities by esa's space debris office," *CEAS Space Journal*, vol. 8, pp. 177–189, 5 2016.

[17] E. S. Gloag, C. Elbadawi, C. Zachreson, I. Aharonovich, M. Toth, I. G. Charles, L. Turnbull, and C. B. Whitchurch, "Micro-patterned surfaces that exploit stigmergy to inhibit biofilm expansion.," *Frontiers in microbiology*, vol. 7, pp. 2157–2157, 1 2017.

[18] R. Doriya, K. M. Srivastava, and P. Buwe, *An Architecture for Detection of Land Mines Using Swarm Robots*, pp. 311–323. Springer India, 8 2015.

[19] P. F. Lalor and W. O. H. Hughes, "Alarm behaviour in eciton army ants," *Physiological Entomology*, vol. 36, pp. 1–7, 2 2011.

[20] A. Basgumus, M. Namdar, G. Yilmaz, and A. Altuncu, *Performance Analysis of the Differential Evolution and Particle Swarm Optimization Algorithms in Cooperative Wireless Communications.* InTech, 9 2016.

[21] C. Reimmann, N. Ginet, L. Michel, C. Keel, P. Michaux, V. Krishnapillai, M. Zala, K. Heurlier, K. Triandafillu, H. Harms, G. Défago, and D. Haas, "Genetically programmed autoinducer destruction reduces virulence gene expression and swarming motility in pseudomonas aeruginosa pao1," *Microbiology (Reading, England)*, vol. 148, pp. 923–932, 4 2002.

[22] D. Sudholt, "Theory of swarm intelligence," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pp. 715–734, ACM, 7 2016.

[23] X. An and H. Hecht, "Better v&v for critical flight systems," in *AIAA 1st Intelligent Systems Technical Conference*, American Institute of Aeronautics and Astronautics, 6 2004.

[24] 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.

[25] G. Witzany, *Communicative Coordination in Bees*, pp. 135–147. Springer Netherlands, 8 2013.

[26] J.-W. Lee, "Optimal power allocating for correlated data fusion in decentralized wsns using algorithms based on swarm intelligence," *Wireless Networks*, vol. 23, pp. 1655–1667, 2 2017.

[27] H. R. Hepburn, *Absconding, Migration and Swarming*, pp. 133–158. Springer Berlin Heidelberg, 12 2010.

[28] K. N. Dattatraya, K. R. Rao, and D. S. Kumar, "Architectural analysis for lifetime maximization and energy efficiency in hybridized wsn model," *International Journal of Engineering & Technology*, vol. 7, pp. 494–, 3 2018.

[29] H.-M. Jung, Y.-G. Hwang, and D.-H. Kim, "Self-organization of swarm robots based on color recognition," *Journal of Korean Institute of Intelligent Systems*, vol. 20, pp. 413–421, 6 2010.

[30] M. Ojha, *SocProS (1) - Optimizing Supply Chain Management Using Gravitational Search Algorithm and Multi Agent System*, pp. 481–491. United States: Springer India, 4 2012.

[31] M. Mabrouk, "Using the swarm leader concept for multi-agent obstacle avoidance," *International Conference on Aerospace Sciences and Aviation Technology*, vol. 14, pp. 1–12, 5 2011.

[32] 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*, pp. 628–629, IEEE, 2006.

[33] A. J. Wood, "Networks of injustice and worker mobilisation at walmart," *Industrial Relations Journal*, vol. 46, pp. 259–274, 7 2015.

[34] W. Wrótniak-Drzewiecka, A. J. Brzezińska, H. Dahm, A. P. Ingle, and M. Rai, "Current trends in myxobacteria research," *Annals of Microbiology*, vol. 66, pp. 17–33, 6 2015.

[35] R.-M. Chen and C.-M. Wang, "Project scheduling heuristics-based standard pso for task-resource assignment in heterogeneous grid," *Abstract and Applied Analysis*, vol. 2011, pp. 1–20, 2 2011.

[36] R. Sinha, A. Choubey, S. K. Mahto, and P. Ranjan, *Quantum Behaved Particle Swarm Optimization Technique Applied to FIR-Based Linear and Nonlinear Channel Equalizer*, pp. 37–50. Springer Singapore, 8 2018.

[37] S. Mandal, G. Saha, and R. K. Pal, "Recurrent neural network-based modeling of gene regulatory network using elephant swarm water search algorithm," *Journal of bioinformatics and computational biology*, vol. 15, pp. 1750016–1750016, 6 2017.

[38] M. T. Mekonnen and K. N. Rao, "Cluster optimization based on metaheuristic algorithms in wireless sensor networks," *Wireless Personal Communications*, vol. 97, pp. 2633–2647, 7 2017.

[39] G. Sartoretti, M.-O. Hongler, M. E. de Oliveira, and F. Mondada, "Decentralized self-selection of swarm trajectories: from dynamical systems theory to robotic implementation," *Swarm Intelligence*, vol. 8, pp. 329–351, 11 2014.

[40] M. Andraud, A. Hallawa, J. D. Roose, E. Cantatore, G. Ascheid, and M. Verhelst, "Gecco (companion) - evolving hardware instinctive behaviors in resource-scarce agent swarms exploring hard-to-reach environments," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1497–1504, ACM, 7 2018.

[41] M. Subotic and M. Tuba, "Parallelized multiple swarm artificial bee colony algorithm (ms-abc) for global optimization," *Studies in Informatics and Control*, vol. 23, 3 2014.

[42] 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, 7 2007.

[43] P.-F. Yang and Q. Wang, "Effective task scheduling and ip mapping algorithm for heterogeneous noc-based mpsoc," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–8, 7 2014.

[44] A. Bricard, J.-B. Caussin, N. Desreumaux, O. Dauchot, and D. Bartolo, "Emergence of macroscopic directed motion in populations of motile colloids," *Nature*, vol. 503, pp. 95–98, 11 2013.

[45] V. Abedifar and M. Eshghi, "An optimized design of optical networks using evolutionary algorithms," *Journal of High Speed Networks*, vol. 20, pp. 11–27, 1 2014.

[46] V. Trianni and S. Nolfi, "Engineering the evolution of self-organizing behaviors in swarm robotics: A case study," *Artificial life*, vol. 17, pp. 183–202, 5 2011.

[47] A. K. Gupta, H. Sadawarti, and A. K. Verma, "Computation of pheromone values in antnet algorithm," *International Journal of Computer Network and Information Security*, vol. 4, pp. 47–54, 8 2012.

[48] R. Kesavamoorthy and K. R. Soundar, "Swarm intelligence based autonomous ddos attack detection and defense using multi agent system," *Cluster Computing*, vol. 22, pp. 9469–9476, 3 2018.

[49] M. Rosalie, G. Danoy, P. Bouvry, and S. Chaumette, *DroNet@MobiSys - UAV Multilevel Swarms for Situation Management.* ACM, 6 2016.

[50] A. Cord-Landwehr, M. Fischer, D. Jung, and F. M. auf der Heide, "Asymptotically optimal gathering on a grid," 1 2016.

[51] 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.

[52] H. Kaur and A. K. Sharma, "Novel email spam classification using integrated particle swarm optimization and j48," *International Journal of Computer Applications*, vol. 149, pp. 23–27, 9 2016.

[53] Álvaro Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry," *Neural Computing and Applications*, vol. 19, pp. 807–823, 5 2010.

[54] M. Suriya and M. G. Sumithra, *Efficient Evolutionary Techniques for Wireless Body Area Using Cognitive Radio Networks*, pp. 61–70. Springer International Publishing, 12 2018.

[55] S. Kurosawa, N. Kamo, D. Matsui, and Y. Kobatake, "The synthetic moth: a neuromorphic approach toward artificial olfaction in robots," *Analytical Chemistry*, vol. 62, pp. 353–359, 2 1990.

[56] L. Improta, L. Valoroso, D. Piccinini, and C. Chiarabba, "A detailed analysis of wastewater induced seismicity in the val d'agri oil field (italy)," *Geophysical Research Letters*, vol. 42, pp. 2682–2690, 4 2015.

[57] P. Smith, R. Hunjet, A. Aleti, and J. C. Barca, "Data transfer via uav swarm behaviours," *Australian Journal of Telecommunications and the Digital Economy*, vol. 6, pp. 35–57, 6 2018.

[58] M. B. Abdelhalim and S. E. D. Habib, "An integrated high-level hardware/software partitioning methodology," *Design Automation for Embedded Systems*, vol. 15, pp. 19–50, 2 2011.

[59] S. K. Jain, A. Patnaik, and S. N. Sinha, "Design of custom-made stacked patch antennas: a machine learning approach," *International Journal of Machine Learning and Cybernetics*, vol. 4, pp. 189–194, 3 2012.

[60] D. Krupke, M. Hemmer, J. McLurkin, Y. Zhou, and S. P. Fekete, "A parallel distributed strategy for arraying a scattered robot swarm," 1 2015.

[61] 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*, pp. 1–4, IEEE, 2006.

[62] K. Vidhya and K. R. S. Kumar, "Channel estimation of mimo–ofdm system using pso and ga," *Arabian Journal for Science and Engineering*, vol. 39, pp. 4047–4056, 3 2014.

[63] T. Ganchev, K. E. Parsopoulos, M. N. Vrahatis, and N. Fakotakis, *Partially Connected Locally Recurrent Probabilistic Neural Networks.* InTech, 9 2008.

[64] D. C. Diana and S. P. J. V. Rani, “Novel cat swarm optimization algorithm to enhance channel equalization,” *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 36, pp. 350–363, 1 2017.