scisearch INTERNATIONAL PEER-REVIEWED JOURNALS

RESEARCH PAPER

Agent-Based Multi-Objective Optimization for Network Slicing and Admission Control in 5G/6G Softwarized Infrastructures

Le Hoang Kiet[1] and Pham Thi Lan Anh[2]

[1]Mekong Delta Institute of Technology, Department of Computer Science, Tran Hung Dao Street, Can Tho, Vietnam
[2]Da Nang Horizon University, Department of Information Systems, Nguyen Van Linh Street, Da Nang, Vietnam

Abstract

Fifth-generation and prospective sixth-generation mobile networks are based on highly softwarized infrastructures in which network functions and resources are dynamically instantiated and reconfigured across disaggregated computing and transport domains. Network slicing exposes these capabilities as logically isolated end-to-end tenants, each with distinct performance, reliability, and cost requirements. Admission control must decide whether new slice requests can be accepted without violating guarantees for existing slices, while resource allocation mechanisms must coordinate heterogeneous compute, storage, and radio resources. Multi-objective trade-offs arise naturally between spectral efficiency, latency, energy consumption, and operational expenditure. Centralized optimization approaches struggle with the dimensionality, dynamics, and partial observability of realistic 5G and 6G deployments. Agent-based coordination, where distributed software agents represent slices, infrastructure domains, and orchestration functions, provides a flexible means to adapt to local conditions, negotiate resource usage, and incorporate learning-based policies. This work examines an agent-based multi-objective optimization framework for network slicing and admission control in softwarized infrastructures. The study focuses on linear models of resource coupling and service-level constraints, explores scalarization and constraint-based formulations of multiple objectives, and considers both myopic and horizon-based decision processes. The discussion emphasizes how autonomous agents can approximate global optimization goals with limited information exchange and modest computational overhead, and how linear relaxations can support explainable admission decisions and capacity planning. The analysis covers system modeling, algorithmic design, and qualitative evaluation of trade-offs, aiming to clarify the conditions under which agent-based strategies achieve consistent and predictable behavior in the presence of rapidly varying traffic demands and slice heterogeneity.

1. Introduction

Softwarized mobile infrastructures are reshaping the way communication services are deployed and managed (1). Virtualization of network functions, programmable data planes, and cloud-native microservice architectures allow operators to instantiate end-to-end logical networks on demand. These logical networks, often called network slices, span radio access, transport, and core segments, while sharing the same physical infrastructure. Each slice is characterized by a service descriptor that specifies quality of service targets, reliability levels, security properties, and cost or revenue parameters (2). The orchestration platform must map these logical specifications into concrete allocations of radio spectrum, baseband processing, transport bandwidth, and core computing resources on a time-varying basis. This creates a complex resource management problem, where decisions must be taken at multiple time scales and under uncertainty about future traffic and environment conditions.

Admission control plays a central role in maintaining isolation and service guarantees among slices (3). When a new slice request arrives, the system must decide whether the current and projected resource capacities can accommodate its requirements without degrading already admitted slices beyond their specified contracts. This decision involves predicting traffic patterns, evaluating capacity margins, and assessing multi-domain dependencies across radio and transport segments. In 5G and anticipated 6G environments, the scale and diversity of slices, ranging from enhanced mobile broadband to massive machine-type communication and ultra-reliable low-latency communication, render simple threshold-based policies insufficient (4). More expressive optimization-based strategies are needed, yet fully centralized solvers are challenged by the combinatorial nature of embedding decisions, state space dimensionality, and stringent time constraints.

Agent-based models offer a way to decompose this decision process into smaller, interacting subsystems. In such models, software agents represent slices, infrastructure points of presence, or orchestration layers (5). Each agent observes a subset of the system state, applies local decision rules, and exchanges limited information with other agents. When properly designed, these local interactions can lead to emergent global behavior that approximates multi-objective optimality. Agents can adapt over time, either through learning mechanisms or through adaptive rule tuning, in order to track changes in traffic mix, user mobility, and hardware capacity (6). This form of distributed intelligence aligns with the architectural trends in

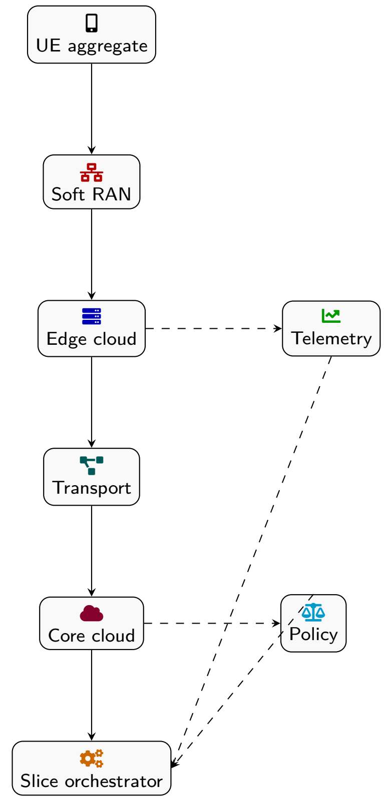

Figure 1. End-to-end softwarized slicing stack, with user equipment traffic traversing radio, edge, transport, and core segments under the control of a slice orchestrator that consumes telemetry and policy signals.

service-based 5G cores, open radio access networks, and edge computing, where control logic is naturally decomposed and deployed close to the resources it manages.

Multi-objective optimization is intrinsic to the management of network slices, because operators must balance revenue from admitted slices against performance metrics such as latency and packet loss, as well as operational expenditures expressed in terms of power consumption and resource fragmentation. Many of these objectives can be expressed or approximated using linear functions of decision variables, especially when focusing on long-term averages or aggregated flows (7). Linear models enable tractable formulations and provide interpretable sensitivity analysis, at the cost of abstracting away fine-grained protocol behavior. In contrast, fully detailed simulators and learning-based policies may capture more nuances but can be less transparent and harder to integrate into regulatory or contractual reasoning.

In this context, it is useful to examine how linear multi-objective formulations can be embedded into agent-based control architectures for network slicing and admission control (8). A key question is how to partition the optimization problem so that each agent solves a smaller linear program or mixed-integer linear program using locally available information, while still approximating desirable global trade-offs. Another question is how to encode inter-agent coordination mechanisms, such as pricing, bidding, or quota exchange, that implement implicit Lagrange multipliers associated with shared capacity constraints. These coordination mechanisms must remain lightweight enough to be executed within orchestration loops on time scales ranging from seconds to minutes (9).

The following sections explore these issues by first describing the characteristics of softwarized 5G and 6G network slicing infrastructures, then developing a linear system model and multi-objective formulation, and finally outlining an agent-based optimization architecture and discussing representative evaluation scenarios. The focus remains on neutral analysis of modeling choices and their implications, rather than on any particular implementation. Attention is given to how the linear structure of the model influences algorithm design and how agent-local decisions can be aligned with multi-objective criteria through suitable scalarization and constraint handling strategies (10).

2. Softwarized 5G/6G Network Slicing Background

The notion of network slicing emerged as a way to expose a shared infrastructure as a set of logically separate networks with predictable behavior. In 5G systems, a slice typically includes radio access functions such as distributed units and centralized units, transport segments across optical or packet-switched backbones, and core network functions implemented as virtual or containerized network functions. Each slice has an associated template that defines performance targets in terms of throughput, latency, reliability, and jitter, as well as life-cycle attributes such as elasticity and expected duration (11). These templates are translated by orchestration components

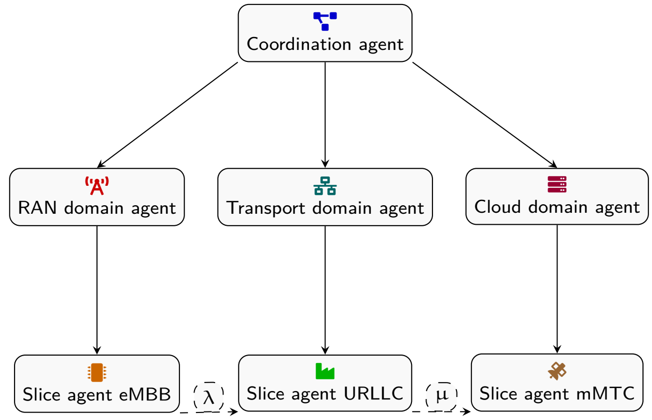

Figure 2. Agent hierarchy with a coordination agent interacting with domain-level agents and per-slice agents, supporting negotiation and exchange of aggregate control signals.

into placement and scaling decisions for network functions and bandwidth reservations across multiple domains.

Softwarization refers to the replacement of fixed-function hardware appliances with software functions running on general-purpose processing platforms, as well as the programmatic configuration of network devices through standardized APIs. In the radio access network, this includes the use of software-defined radios, flexible baseband processing chains, and split architectures that separate control and data flows (12). In the transport and core network, this includes software-defined networking controllers, virtual switches, and service meshes interconnecting microservices implementing network functions. The consequence is a high degree of configurability, which enables fine-grained resource allocation but also increases the complexity of control decisions.

In such environments, slices must compete for a mixture of resources (13). Radio access resources include spectrum segments and time-frequency blocks that can be assigned to users associated with a slice. Processing resources include central processing units, graphics processing units, and specialized accelerators located in edge clouds, central clouds, and possibly in baseband units. Transport resources include bandwidth along multiple paths in backhaul and fronthaul networks, with different latency and reliability characteristics (14). The softwarized infrastructure is subject to capacity constraints in all these dimensions, and these constraints are intertwined because a given slice instance may span several resource domains simultaneously.

Admission control decisions depend on assessing whether sufficient capacity exists in all relevant domains over a planning horizon. For example, an ultra-reliable low-latency communication slice for industrial control may require dedicated radio blocks, low-delay paths in the transport network, and reserved compute capacity in an edge cloud. The admission decision must consider not only current utilization but also the potential variations in traffic demand from existing slices (15). Traditional threshold-based policies often rely on static safety margins, which may be conservative or insufficient under varying conditions. More flexible policies rely on optimization frameworks that model capacity, demand, and service-level constraints explicitly.

Service descriptors for slices often contain linear or affine resource demand models, where required capacity is proportional to the expected traffic load, modulated by efficiency factors and overheads (16). For instance, the processing requirement for a core function can be expressed as a linear function of aggregate user throughput, with coefficients capturing protocol stack complexity. Similarly, bandwidth requirements on fronthaul links may be expressed as linear functions of modulation and coding schemes. These approximations are useful because they map naturally into linear constraints in optimization problems (17). While detailed protocol behavior and wireless channel dynamics are more complex, it is often sufficient for admission control purposes to operate at this level of abstraction.

Softwarized orchestration platforms provide several points of control suitable for agent-based implementations. Near-real-time control loops operate in radio access control units, making frequent decisions on scheduling and resource block assignment (18). Non-real-time control loops operate in higher-level controllers or management and orchestration entities, making decisions on slice instantiation, scaling, and placement. Agents can be associated with these control points, where they observe local metrics, maintain internal models, and interact with peer agents. The natural decomposition between radio, transport, and computing domains supports the definition of specialized agents with domain-specific expertise that collaborate to handle end-to-end slicing and admission control (19).

In 6G scenarios, the heterogeneity of services and infrastructures is expected to increase further. New use cases, such as integrated sensing and communication, digital twin synchronization, and extremely dense machine-type communication, introduce additional constraints on synchronization, spatial coverage, and energy efficiency. These constraints can be accommodated in linear models by introducing additional variables and constraints representing, for example, upper bounds on total energy consumption or lower bounds on coverage probabilities approximated by linear surrogates (20). At the same

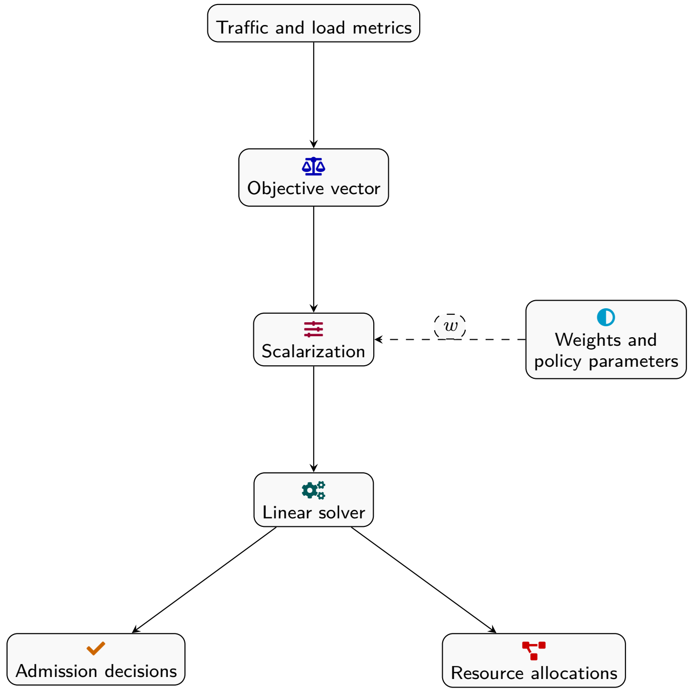

Figure 3. Conceptual flow from measured metrics to objective construction, scalarization, linear optimization, and resulting admission and allocation decisions with tunable weight parameters.

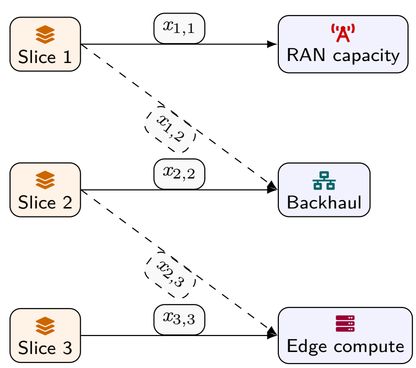

Figure 4. Bipartite view of slices and resource pools illustrating representative allocation variables that participate in linear capacity and demand constraints.

time, the number of slices and their dynamics may increase, making centralized optimization increasingly difficult to scale. Agent-based approaches become attractive as a way to handle this complexity, provided they can be structured to maintain consistent global behavior.

3. System Model and Multi-Objective Problem Formulation

The system model considers a set of slices and a softwarized infrastructure composed of multiple resource domains (21). Let the set of slices be denoted by a finite set, and let the set of infrastructure points of presence be another finite set. Each point of presence offers several resource types, such as processing, memory, and bandwidth. For each resource type, a nonnegative capacity is given (22). The model operates over a planning epoch during which admission decisions are made and resource allocations are assumed quasi-static at the selected time scale.

For each slice, a decision variable represents whether the slice is admitted during the considered epoch. This variable takes values in the interval between zero and one, with binary values representing strict admission or rejection and fractional values representing relaxed decisions in a linear programming context (23). For each slice and each point of presence, an allocation variable represents the amount of a particular resource type that is assigned to that slice on that point of presence. These allocation variables are nonnegative and are subject to capacity constraints on each resource and point of presence.

A typical capacity constraint for a single resource type and point of presence can be modeled as a linear inequality (24). For a given resource type and point of presence, let the allocation variable for slice index be denoted by a scalar. The sum of allocations over all slices cannot exceed the capacity at that

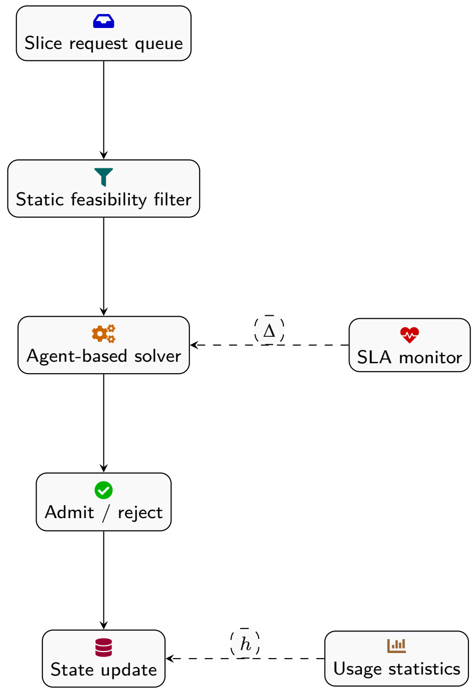

Figure 5. Admission control pipeline combining static pre-checks, agent-based optimization, and feedback from service monitoring and aggregate usage information.

point of presence and resource type. This yields a constraint of the form (25)

$$ {}_{s} x_{i,s} \leq C_i, $$

where the index runs over slices, the variable denotes the allocation to slice on point of presence, and the parameter denotes the capacity. The summation is taken over all slices that can potentially use that point of presence, and the inequality enforces that the allocated amount remains within the resource limit.

Each slice is associated with a demand profile that indicates the minimum amount of resources required in each domain if the slice is admitted (26). This minimum requirement can be expressed through constraints that link the admission variable and the allocation variables. For a given slice and resource type, a constraint of the form

$$ {}_{i} x_{i,s} \geq d_s a_s $$

can be used, where the index runs over points of presence, the variable denotes the allocation, the parameter denotes the demand, and the variable denotes the admission decision (27). When the admission variable is zero, the constraint does not impose any requirement on allocations. When the admission variable is one, the total allocated capacity across points of presence must be at least as large as the demanded amount. This structure preserves linearity and connects admission decisions with resource allocations in a consistent way.

The multi-objective nature of the problem arises from the need to balance different performance and cost metrics (28). A simple revenue objective can be formulated as a linear function of admission variables. For each slice, a scalar revenue coefficient can be associated, and the total revenue is

$$ f_R a = {}_{s} r_s a_s, $$

where the coefficient denotes the revenue associated with admitting slice and the decision variable denotes the admission state (29). An operator might also wish to penalize resource consumption to account for energy usage or capacity reservation. This can be modeled by a linear cost function of the allocation variables, such as

$$ f_C x = {}_{i,s} c_{i,s} x_{i,s}, $$

where the coefficients represent unit costs associated with allocating resources to slice at point of presence (30).

Additional objectives can capture quality of service aspects. While detailed latency distributions are not linear, aggregate latency or queueing surrogates can sometimes be approximated linearly in terms of allocated capacities and demand. For example, in a regime where latency decreases approximately linearly with over-provisioning, one may introduce a variable representing slack between provisioned capacity and demanded capacity for each slice, and weight these slacks in a linear objective (31). Similarly, reliability constraints can be modeled as linear constraints on redundant allocations when replicated functions or paths are used, with an associated penalty in the objective representing redundancy cost.

One common approach to handle multiple objectives within a linear framework is to form a scalar objective by taking a weighted sum of individual objectives. Let a vector of nonnegative weights be specified, and define the aggregate objective

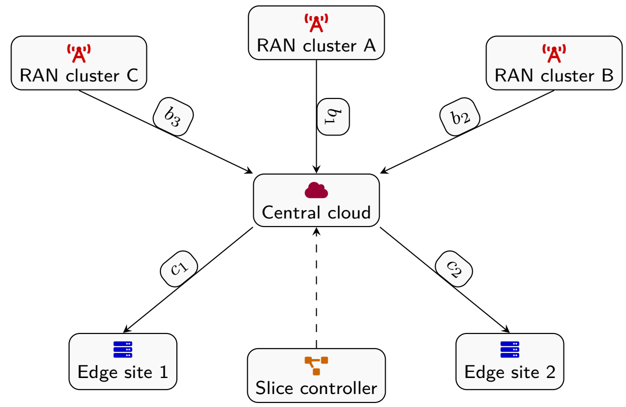

Figure 6. Multi-domain topology with radio clusters, central cloud, and edge sites, indicating representative bandwidth and compute flows coordinated by a slice controller.

as (32)

$$Fa, x = w_R f_R a - w_C f_C x,$$

where the weight represents the relative importance of revenue and the weight represents the importance of cost. Additional objectives can be included by extending this linear combination with further terms and associated weights. Alternatively, one can treat one of the objectives as a primary objective and the others as constraints, such as maximizing revenue subject to an upper bound on total cost or total energy consumption (33).

The overall linear optimization problem at the system level can be stated informally as finding decision variables that satisfy capacity constraints, service demand constraints, and possible additional policy constraints while optimizing a chosen scalarization of multiple objectives. In a compact form, one can write a linear program with the objective

$$\max_{a,x} \; Fa, x$$

subject to a set of linear inequalities and equalities that represent the constraints described above (34). In the strict admission control setting, one further constrains the admission variables to be binary. This yields a mixed-integer linear program, which is generally harder to solve globally but can often be approximated using relaxation and rounding strategies.

The agent-based view of this system model interprets the decision variables as being controlled by different agents (35). Slice agents control their admission proposals and preferred allocation patterns, infrastructure agents control capacity quotas for slices, and coordinating agents enforce global consistency. Each agent solves a local linear optimization problem, which is derived from the global model but restricted to the subset of variables and constraints relevant to that agent. Communication between agents conveys prices, dual variables, or quota updates that link these local problems together (36). From the perspective of the system model, this corresponds to distributed methods for solving the overall linear program or its multi-objective variants.

4. Agent-Based Optimization and Admission Control Algorithms

Agent-based optimization in softwarized network slicing systems relies on the decomposition of the global multi-objective problem into smaller parts that can be solved by individual agents using local information. Consider a collection of slice agents, each representing one slice or slice request, and a collection of infrastructure agents, each associated with a point of presence or resource domain (37). A central or federated coordination agent may also exist but is not assumed to maintain full information at all times. The goal is for these agents to collectively approximate the solution of the system-level linear or mixed-integer linear program, while respecting practical constraints on communication frequency and local processing capabilities.

A slice agent typically aims to maximize a local utility that reflects the expected revenue of being admitted minus potential costs paid for resources (38). Let a slice agent maintain an admission variable and a vector of desired allocations across points of presence. The local problem for a slice agent can be expressed as a linear program with objective

$$\max_{a_s, x_s} \; u_s a_s - {}_{i} p_{i,s} x_{i,s},$$

subject to a demand constraint of the form (39)

$${}_{i} x_{i,s} \geq d_s a_s,$$

and nonnegativity constraints on the allocations. Here the coefficient represents the local utility of being admitted, and the parameters represent prices or costs per unit resource advertised by infrastructure agents. The variable denotes the slice-specific allocation on point of presence, and the coefficient captures the willingness of the slice to pay for that allocation (40).

Infrastructure agents, in contrast, seek to keep their total allocations within capacity while possibly maximizing a function reflecting revenue collected from slices and penalties for congestion. For a given point of presence, an infrastructure agent sees a set of allocation requests and responds with prices. Its local problem can be written as (41)

$$\max_{x_i} \; {}_{s} p_{i,s} x_{i,s}$$

Table 1. Representative slice classes and qualitative latency and reliability regimes considered in agent-based admission control.

Slice class	Typical service	Latency regime	Reliability regime
eMBB	Video streaming, XR	Tens of ms	Medium to high
URLLC	Industrial control	Sub-ms to few ms	Very high
mMTC	Sensor networks	Seconds to minutes	Medium
Best-effort data	Web, background sync	Tens to hundreds of ms	Low to medium
Private network	Enterprise campus	Few ms to tens of ms	High

Table 2. Main resource domains with abstract capacities, monetary cost factors, and energy coefficients used in the linear optimization model.

Resource domain	Symbolic capacity	Unit cost factor	Energy coefficient
RAN processing	C^{ran}	c^{ran}	e^{ran}
Backhaul bandwidth	C^{bh}	c^{bh}	e^{bh}
Edge compute	C^{edge}	c^{edge}	e^{edge}
Core compute	C^{core}	c^{core}	e^{core}
Storage pool	C^{st}	c^{st}	e^{st}

subject to a capacity constraint

$${}_{s} x_{i,s} \leq C_i,$$

and nonnegativity of allocations. In a price-adjustment process, the prices may be updated as a function of congestion, for example by increasing prices when the capacity constraint is tight and decreasing prices when there is spare capacity. Slice agents then adjust their desired allocations according to these price signals, leading to an iterative process that can converge towards a resource allocation satisfying global capacity constraints and reflecting a balance between slice utilities and infrastructure limitations (42).

The multi-objective aspect enters when slice utilities and infrastructure pricing rules incorporate multiple criteria. For example, prices can be differentiated according to latency classes, reliability classes, or energy efficiency profiles. Linear models can represent these distinctions by assigning separate allocation variables and price coefficients to different resource classes (43). A slice that values low latency more than cost might be modeled with a higher utility coefficient for allocations on low-latency resources and a larger penalty for being rejected. In a multi-objective scalarization framework, these preferences are expressed as weights in the slice agent utility function and in the infrastructure agent cost function.

Admission control decisions emerge from the interaction between slice agents and infrastructure agents (44). A slice agent that cannot obtain sufficient resource allocations at acceptable prices to satisfy its demand constraint may set its admission variable effectively to zero, indicating that it does not enter the system. Conversely, if sufficient capacity is available at prices that still leave positive utility, the slice agent will choose an admission variable close to one and accept the associated resource allocations. In a strict admission setting with binary variables, this decision can be made by comparing the expected utility of being admitted with that of being rejected, given the current price and capacity conditions (45).

Agent-based algorithms can be implemented using iterative schemes where agents exchange messages at discrete time steps. At each iteration, infrastructure agents announce current prices, slice agents solve their local linear programs to update desired allocations and admission decisions, and infrastructure agents update prices based on aggregate demand. This type of process resembles primal-dual methods for linear programming, where prices approximate dual variables associated with capacity constraints (46). When step sizes and update rules are chosen suitably, the iteration can converge to a point close to an optimal solution of the global problem, within the limitations imposed by integer constraints and approximations.

Beyond simple price adjustment, agents may use learning mechanisms to adapt their strategies. For instance, a slice agent may treat the environment formed by infrastructure agents and other slices as a partially observable process and apply reinforcement learning to discover policies mapping demand and price observations to admission decisions and allocation requests (47). To maintain compatibility with linear modeling, the reward structure can be aligned with linear utility functions, and the policy search space can be restricted to strategies that effectively implement parametric linear decision rules. Infrastructure agents can similarly learn mappings from observed congestion patterns to price update rules that stabilize the system and maintain desired utilization levels.

The coordination of multiple objectives also benefits from agent-based decomposition (48). One approach is to assign different objectives to different agents and rely on negotiation to balance them. For example, a global energy management agent might adjust a vector of cost coefficients that influence infrastructure agents, which in turn adjust prices seen by slice agents.

Table 3. Structure of the scalar objective formed from admission decisions, allocations, and slack variables in the multi-objective formulation.

Objective component	Decision variables	Meaning	Weight
Revenue term	a_s	Admission utility per slice	w_R
Allocation cost	$x_{i,s}$	Marginal infrastructure cost	w_C
Energy term	$x_{i,s}$	Energy-proportional usage	w_E
Slack penalties	σ_s	Violations of QoS surrogates	w_S

Table 4. Agent types, internal state summaries, primary decisions, and characteristic operating time scales in the distributed control architecture.

Agent type	Local state	Primary decision	Time scale
Slice agent	Demand estimate, price view	a_s, $x_{i,s}$	Seconds to minutes
RAN agent	Cell load, interference	Local capacity shares	Sub-second to seconds
Transport agent	Link utilization	Per-slice bandwidth shares	Seconds
Cloud agent	CPU and memory load	VM or container quotas	Seconds to minutes
Coordination agent	Aggregate duals	Weight and price updates	Minutes

A service-level management agent might impose constraints on the minimum admission rate of certain slice categories by modifying utility coefficients (49). This layered approach preserves linearity at each layer while allowing the system to steer overall behavior in a multi-objective space.

From an implementation perspective, agent-based optimization must handle the fact that decisions are taken at different time scales. Fast control loops in the radio access network adjust scheduling decisions on time scales of milliseconds, whereas slice admission and placement decisions occur on time scales of seconds to minutes (50). The linear models described earlier are primarily suited to these slower time scales. Agents operating at this level can incorporate aggregated information from faster loops, such as average resource utilization and observed violations of service-level indicators, and adjust admission and allocation policies accordingly. This separation by time scale allows complex dynamics to be decomposed into layers, with agents at each layer using mathematical models appropriate for their horizon (51).

5. Evaluation Methodology and Results Discussion

Evaluating an agent-based multi-objective optimization framework for network slicing and admission control involves constructing scenarios that reflect realistic properties of softwarized 5G and 6G infrastructures while remaining manageable within a modeling and simulation environment. The evaluation can consider a multi-domain infrastructure composed of radio access clusters, edge clouds, and a central cloud, interconnected by a transport network with heterogeneous link characteristics. Each infrastructure element is modeled with capacity parameters for processing, storage, and bandwidth, along with latency and energy consumption profiles (52). Slice requests arrive according to traffic patterns representing enhanced mobile broadband, machine-type communication, and ultra-reliable low-latency communication, each with distinct demand models and service-level indicators.

In such an evaluation, a baseline centralized optimization approach can be compared with the agent-based framework. The centralized approach solves the system-level linear or mixed-integer linear program using full information about capacities, demands, and service descriptors (53). The agent-based approach, in contrast, relies on local agent decisions and limited information exchange, as described previously. Key performance metrics include admission ratio, defined as the fraction of slice requests that are admitted; resource utilization, measured as the ratio between allocated and available capacity across resource types; energy consumption, expressed as a linear function of allocation and activity variables; and violation rates of service-level indicators, such as latency bounds and throughput guarantees.

Simulation experiments can be configured to vary the intensity and composition of slice requests, the distribution of capacities between edge and central clouds, and the parameters of agent decision rules (54). For instance, one can consider a scenario with relatively abundant central cloud capacity and tight edge capacities, and another scenario with more balanced capacities. The arrival rate of ultra-reliable low-latency communication slices can be varied to stress latency and reliability constraints. For each scenario, the central optimizer and the agent-based framework can be applied to identical traffic realizations, and the resulting metrics can be compared. These comparisons help to identify regimes where the agent-based approach matches or deviates from the centralized benchmark (55).

In analyzing results, one may observe that the agent-based framework often achieves admission ratios close to those of the centralized optimizer when capacities are moderately loaded and traffic patterns are stable. In such situations, price-based coordination aligns slice decisions with available capacities without requiring precise global information. As the system ap-

Table 5. Simulation scenarios varying load, capacity distribution, traffic mix, and energy budget for comparing centralized and agent-based policies.

Scenario	Load level	Edge capacity bias	Traffic mix	Target energy budget
S1	Moderate	Balanced edge and core	eMBB heavy	Loose
S2	High	Edge constrained	URLLC heavy	Medium
S3	High	Edge rich	Mixed	Tight
S4	Low	Core dominated	mMTC heavy	Loose
S5	Bursty	Balanced	Alternating eMBB/URLLC	Medium

Table 6. Key metrics used to assess the behavior of the agent-based optimization scheme against a centralized baseline.

Metric	Symbol	Description	Aggregation
Admission ratio	η_{adm}	Admitted requests over total	Time-averaged
Resource utilization	ρ_k	Usage over capacity per domain	Time-averaged
Energy usage	E	Total consumed energy	Per scenario
SLA violation rate	ν	Fraction of requests with QoS slack > 0	Time-averaged
Convergence iterations	I	Iterations until stable prices	Per run

proaches high utilization, differences may emerge due to the myopic or decentralized nature of agent decisions (56). For example, agents may over-allocate resources to certain slice classes early in the simulation, limiting capacity for later arrivals that could have generated higher overall utility under the centralized policy. These observations highlight the influence of agent update rules and information structures on performance at high load.

Resource utilization patterns under agent-based control provide insights into how capacities are partitioned among slices and domains (57). In some configurations, agents may favor allocations on edge resources to meet latency objectives, leading to higher utilization at the edge and lower utilization in the central cloud. When edge capacities become tight, prices can rise, prompting slices with less stringent latency requirements to shift allocations toward central resources. This behavior can be captured in linear models by adjusting cost coefficients and observing the resulting allocation patterns (58). Comparing these patterns between agent-based and centralized solutions can reveal deviations that arise from approximate coordination and help identify options for refining agent update rules.

Energy consumption is another dimension of interest, particularly in 6G scenarios where energy efficiency is a prominent objective. Linear energy models can associate per-unit energy costs with different resource types and locations, with higher costs for less efficient hardware or for links that traverse longer distances (59). Under a multi-objective scalarization, an operator might assign a weight to energy consumption that competes with the revenue or admission objective. In simulations, varying this weight provides insight into the trade-off between resource utilization and energy savings. The agent-based framework may adopt different trade-off points than a centralized optimizer, depending on how energy costs are reflected in slice utilities and infrastructure prices (60). Observing how these trade-offs shift with parameter changes can inform the design of pricing and utility functions.

Service-level indicator violations, such as missed latency targets or shortfalls in throughput guarantees, can be monitored in the simulation environment. These violations can be linked to insufficient resource allocations or to misplacement of functions relative to user locations (61). In a linear model, such violations are typically approximated by slack variables in constraints, which can be penalized in the objective function. In an agent-based framework, agents may adjust their behavior in response to observed violations by modifying their demand estimates or by negotiating different allocation patterns. Analyzing the frequency and severity of violations under different control schemes helps assess robustness and adaptation capabilities (62).

While quantitative results from simulations depend on specific parameter choices and modeling assumptions, qualitative trends provide useful guidance. For example, when communication between agents is restricted to sparse or noisy signals, convergence to stable allocation patterns may be slower, and transient violations of capacity constraints may occur. Introducing damping factors or conservative safety margins in agent update rules can mitigate these effects at the cost of slightly lower utilization (63). Similarly, incorporating forecasting or history-dependent features into agent decision rules can improve adaptation to nonstationary traffic patterns. The linear structure of the underlying model facilitates such extensions, as agents can update coefficients in their local objective functions or constraints based on observed data without altering the fundamental optimization framework.

Table 7. Main control and model parameters for tuning the stability and trade-offs of the distributed linear optimization process.

Parameter	Symbol	Typical range	Interpretation
Revenue coefficient	r_s	$0.5, 2.0$	Relative importance of slice s
Capacity safety margin	γ	$0.85, 0.98$	Fraction of usable capacity
Step size (prices)	α	$10^{-4}, 10^{-2}$	Dual update rate
Scalarization weight	w_E	$0, 1$	Emphasis on energy term
Max iterations	$I_{\max}$	$50, 300$	Bound on agent updates

Table 8. Qualitative comparison of centralized optimization, heuristics, price-based agents, and learning-based agents in the context of softwarized slicing.

Method	Information scope	Optimality target	Computational burden
Central LP/MILP	Global	Near-global optimum	High
Local heuristics	Local	Feasible allocation	Very low
Price-based agents	Local plus prices	Approximate optimum	Moderate
Learning agents	Local plus history	Policy-level optimum	Moderate to high

6. Conclusion

Softwarized 5G and 6G infrastructures enable network slicing and fine-grained control over resource allocation, but they also introduce complex multi-objective optimization challenges (64). Admission control and resource allocation decisions must balance revenue, performance, energy consumption, and other criteria across heterogeneous slices and resource domains. Linear models offer a tractable and interpretable foundation for representing resource capacities, service demands, and high-level performance surrogates. Within this modeling framework, agent-based optimization presents a way to decompose global decision problems into local problems solved by interacting slice and infrastructure agents (65).

The analysis has outlined how admission variables and allocation variables can be coupled through linear constraints, and how multi-objective trade-offs can be represented as linear combinations of revenue, cost, and service-level metrics. It has described how agents can operate on local linear programs, using price signals or similar coordination mechanisms to respect global capacity constraints and align their decisions with system-level objectives. Different layers of control, operating on different time scales, can incorporate linear models tailored to their decision horizon, from near-real-time resource scheduling to slower slice admission and placement processes (66).

The discussion of evaluation methodology has highlighted the role of simulation environments in exploring the behavior of agent-based frameworks under varied traffic and capacity conditions. Comparing agent-based solutions to centralized optimizers helps quantify the impact of decentralization, limited information, and myopic decision rules on admission ratios, utilization, energy consumption, and service-level indicator violations. While the specific numerical outcomes depend on modeling choices, such comparisons can reveal characteristic patterns, such as performance convergence under moderate load and potential divergences near capacity limits (67).

Overall, the agent-based approach framed within linear multi-objective optimization represents one option for organizing control in softwarized network slicing systems. It supports modularity and scalability by distributing computation and decision-making, and it benefits from the transparency and structural properties of linear models. At the same time, it entails approximation and coordination challenges that warrant careful analysis. Future developments in 6G architecture, service diversity, and energy considerations will likely stimulate further refinement of both modeling techniques and agent-based algorithms, including enhanced learning capabilities and more nuanced representations of performance metrics, while maintaining compatibility with linear or piecewise linear frameworks where such compatibility remains advantageous (68).

References

[1] P. Visu, S. Koteeswaran, and J. P. Janet, "Artificial bee colony based energy aware and energy efficient routing protocol," *Journal of Computer Science*, vol. 8, pp. 227–231, 10 2012.

[2] K. Diwold, A. Aderhold, A. Scheidler, and M. Middendorf, "Performance evaluation of artificial bee colony optimization and new selection schemes," *Memetic Computing*, vol. 3, pp. 149–162, 7 2011.

[3] X. Wei, "Parameters analysis for basic ant colony optimization algorithm in tsp," *International Journal of u- and e-Service, Science and Technology*, vol. 7, pp. 159–170, 8 2014.

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

[5] M. Gardner, A. McNabb, and K. D. Seppi, "A speculative approach to parallelization in particle swarm optimization," *Swarm Intelligence*, vol. 6, pp. 77–116, 12 2011.

[6] C. Dong, L. Wang, and K. Zhao, "Embedded mobile crowd service systems based on opportunistic geological grid and dynamical segmentation," *EURASIP Journal on Embedded Systems*, vol. 2015, pp. 3–, 12 2015.

[7] A. M. Zungeru, L.-M. Ang, and K. P. Seng, "Termite-hill: From natural

Table 9. Qualitative outcome comparison between centralized and agent-based optimization across several performance dimensions.

Outcome dimension	Centralized model	Agent-based model	Typical deviation
Admission ratio	Baseline	Slightly lower at high load	Few percent
Utilization balance	Balanced across domains	Mild domain skew	Scenario dependent
Energy consumption	Reference	Higher or lower depending on w_E	Small to moderate
Violation rate	Reference	Slight increase at saturation	Scenario dependent

to artificial termites in sensor networks," *International Journal of Swarm Intelligence Research*, vol. 3, pp. 1–22, 10 2012.

[8] I. Susnea and C. Axenie, "Cognitive maps for indirect coordination of intelligent agents," *Studies in Informatics and Control*, vol. 24, pp. 111–118, 3 2015.

[9] M. M. al Rifaie, J. M. Bishop, and S. Caines, "Creativity and autonomy in swarm intelligence systems," *Cognitive Computation*, vol. 4, pp. 320–331, 3 2012.

[10] M. A. Tawfeek and G. F. Elhady, "Hybrid algorithm based on swarm intelligence techniques for dynamic tasks scheduling in cloud computing," *International Journal of Intelligent Systems and Applications*, vol. 8, pp. 61–69, 11 2016.

[11] K. Daberao, A. Kundgir, and S. S. Hatkar, "Peer to peer file sharing protocols, methods over manet: Survey.," *International Journal of Advanced Research*, vol. 5, pp. 1778–1786, 6 2017.

[12] M. H. Yar, V. Rahmati, and H. R. D. Oskouei, "A survey on evolutionary computation: Methods and their applications in engineering," *Modern Applied Science*, vol. 10, pp. 131–, 8 2016.

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

[14] 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.

[15] J. Prakash and P. K. Singh, "An effective multiobjective approach for hard partitional clustering," *Memetic Computing*, vol. 7, pp. 93–104, 1 2015.

[16] J. Sasikala and M. Ramaswamy, "Pso based economic emission dispatch for fixed head hydrothermal systems," *Electrical Engineering*, vol. 94, pp. 233–239, 1 2012.

[17] S. Y. Gang and F. Tong, "Network routing optimization based on mixed particle method research," *International Journal of Future Generation Communication and Networking*, vol. 8, pp. 239–250, 8 2015.

[18] M. Ismail and I. El-Henawy, "An efficient hybrid swarm intelligence technique for solving integer programming," *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 10, pp. 1531–1538, 8 2013.

[19] A.-H. Dediu and C. Martín-Vide, "Selected papers of the third international conference on the theory and practice of natural computing, tpnc 2014," *Soft Computing*, vol. 21, pp. 5527–5528, 9 2017.

[20] A. Sharma, H. Sharma, A. Bhargava, N. Sharma, and J. C. Bansal, "Optimal placement and sizing of capacitor using limaçon inspired spider monkey optimization algorithm," *Memetic Computing*, vol. 9, pp. 311–331, 8 2016.

[21] G. Pessin, D. O. Sales, M. A. Dias, R. L. Klaser, D. F. Wolf, J. Ueyama, F. S. Osório, and P. A. Vargas, "Swarm intelligence and the quest to solve a garbage and recycling collection problem," *Soft Computing*, vol. 17, pp. 2311–2325, 8 2013.

[22] P. S. Sajja, "Automatic generation of agents using reusable soft computing code libraries to develop multi agent system for healthcare," *International Journal of Information Technology and Computer Science*, vol. 7, pp. 48–54, 4 2015.

[23] S. A. Ludwig and A. Moallem, "Swarm intelligence approaches for grid load balancing," *Journal of Grid Computing*, vol. 9, pp. 279–301, 2 2011.

[24] 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*, pp. 1–5, IEEE, 2006.

[25] N. Dhanalakshmi and P. Alli, "An analysis and overview of wireless routing protocols," *Journal of Computer Science*, vol. 8, pp. 393–397, 10 2012.

[26] H. Singh and T. P. Singh, "Performance evaluation of token bucket based congestion control protocol for wireless sensor networks," *International Journal of Computer Applications*, vol. 122, pp. 32–37, 7 2015.

[27] P. Jiang, Y. Feng, and F. Wu, "Underwater sensor network redeployment algorithm based on wolf search," *Sensors (Basel, Switzerland)*, vol. 16, pp. 1754–, 10 2016.

[28] E.-S. M. El-kenawy, A. I. Eldesoky, and M. F. Al-rahamawy, "Distributing graphic rendering using grid computing with load balancing," *International Journal of Computer Applications*, vol. 47, pp. 1–6, 6 2012.

[29] G. A. Montoya, C. Velásquez-Villada, and Y. Donoso, "Energy optimization in mobile wireless sensor networks with mobile targets achieving efficient coverage for critical applications," *International Journal of Computers Communications & Control*, vol. 8, pp. 247–254, 2 2013.

[30] Y. Gheraibia, A. Moussaoui, Y. Djenouri, S. Kabir, and P.-Y. Yin, "Penguins search optimisation algorithm for association rules mining," *Journal of Computing and Information Technology*, vol. 24, pp. 165–179, 6 2016.

[31] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "Automatic circle detection on digital images with an adaptive bacterial foraging algorithm," *Soft Computing*, vol. 14, pp. 1151–1164, 10 2009.

[32] S. Rosenfeld, "Global consensus theorem and self-organized criticality: unifying principles for understanding self-organization, swarm intelligence and mechanisms of carcinogenesis.," *Gene regulation and systems biology*, vol. 7, pp. 23–39, 2 2013.

[33] A. A. Aldair, T. Y. Abdalla, and E. B. Alsaedee, "Design of1pid controller9using abc9for9full vehicle nonlinear active suspension system with passenger seat," *International Journal of Computer Applications*, vol. 167, pp. 18–27, 6 2017.

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

[35] L. Ma, X. Chen, and H. Zhang, "Passivity-based decentralized stabilization of multi-agent systems with uncertainty and mixed delays," *Journal of Control Theory and Applications*, vol. 11, pp. 579–585, 10 2013.

[36] C. Wehmeyer, G. F. von Rudorff, S. Wolf, G. Kabbe, D. Schärf, T. D. Kühne, and D. Sebastiani, "Foraging on the potential energy surface: a swarm intelligence-based optimizer for molecular geometry.," *The Journal of chemical physics*, vol. 137, pp. 194110–194110, 11 2012.

[37] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, and J. Hu, "Modified collective decision optimization algorithm with application in trajectory planning of uav," *Applied Intelligence*, vol. 48, pp. 2328–2354, 11 2017.

[38] B. Khaldi and F. Cherif, "An overview of swarm robotics: Swarm intelligence applied to multi-robotics," *International Journal of Computer Applications*, vol. 126, pp. 31–37, 9 2015.

[39] D. Spiller, F. Curti, and L. Ansalone, "Inverse-dynamics particle swarm optimization for spacecraft minimum-time slew maneuvers with constraints," *Aerotecnica Missili & Spazio*, vol. 96, pp. 111–123, 7 2017.

[40] L. Macura and M. Voznak, "Multi-criteria analysis and prediction of network incidents using monitoring system," *Journal of Advanced Engineering and Computation*, vol. 1, pp. 29–38, 6 2017.

[41] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, pp. 87–124, 12 2008.

[42] M. Alzaqebah, S. Abdullah, and S. Jawarneh, "Modified artificial bee colony for the vehicle routing problems with time windows," *SpringerPlus*, vol. 5, pp. 1298–1298, 8 2016.

[43] J. Wang, Y. Sun, and F. Liu, "An improved double-population artificial bee colony algorithm based on heterogeneous comprehensive learning," *Soft Computing*, vol. 22, pp. 6489–6514, 7 2017.

[44] 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.

[45] A. El-Zonkoly, "Optimal allocation and design of grid connected hybrid distributed generation systems," *ERJ. Engineering Research Journal*, vol. 35, pp. 349–358, 10 2012.

[46] D. gan Zhang, H. li Niu, and S. Liu, "Novel peecr-based clustering routing approach," *Soft Computing*, vol. 21, pp. 7313–7323, 7 2016.

[47] A. Maalla, "Performance enhancement of routing algorithm based on pso," *DEStech Transactions on Computer Science and Engineering*, 7 2017.

[48] M. I. Salman, C. K. Ng, N. K. Noordin, B. M. Ali, and A. Sali, "An adaptive threshold based cqi compression scheme for lte cellular networks," *Proceedings of the Asia-Pacific Advanced Network*, vol. 40, pp. 85–90, 10 2015.

[49] M. Sharawi, I. A. Saroit, H. N. Elmahdy, and E. Emary, "Routing wireless sensor networks based on soft computing paradigms: Survey," *International Journal on Soft Computing, Artificial Intelligence and Applications*, vol. 2, pp. 21–36, 8 2013.

[50] V. Selvi and R. Umarani, "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications*, vol. 5, pp. 1–6, 8 2010.

[51] K. Manochehri and A. Alizadegan, "Designing and comparing classic versus quantum artificial bee colony algorithm," *Journal of Mathematics and Computer Science*, vol. 14, pp. 183–192, 4 2015.

[52] R. H. Jacobsen, Q. Zhang, and T. S. Toftegaard, "Bioinspired principles for large-scale networked sensor systems: an overview.," *Sensors (Basel, Switzerland)*, vol. 11, pp. 4137–4151, 4 2011.

[53] J.-W. Lee and W. Kim, "Design of randomly deployed heterogeneous wireless sensor networks by algorithms based on swarm intelligence," *International Journal of Distributed Sensor Networks*, vol. 11, pp. 690235–, 8 2015.

[54] 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.

[55] C. W. Callaghan, "Crowdsourced 'r&d' and medical research.," *British medical bulletin*, vol. 115, pp. 67–76, 8 2015.

[56] X. Cai, H. Wang, Z. Cui, J. Cai, Y. Xue, and L. Wang, "Bat algorithm with triangle-flipping strategy for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 199–215, 11 2017.

[57] J.-C. Dunat, D. Grandblaise, and C. Bonnet, "Collaborative allocation of orthogonal frequency division multiplex sub-carriers using the swarm intelligence," *Journal of Communications*, vol. 1, pp. 68–76, 4 2006.

[58] G. Kaur and M. Kaur, "A study of transform domain based image enhancement techniques," *International Journal of Computer Applications*, vol. 152, pp. 25–29, 10 2016.

[59] S. Datta, "Emergence of digital twins - is this the march of reason?," *Journal of Innovation Management*, vol. 5, pp. 14–33, 11 2017.

[60] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, vol. 30, pp. 413–435, 11 2017.

[61] J.-J. Li, R. Zhang, and Y. Yang, "Multi-auv autonomous task planning based on the scroll time domain quantum bee colony optimization algorithm in uncertain environment.," *PloS one*, vol. 12, pp. e0188291–, 11 2017.

[62] B. Haberman and J. W. Sheppard, "Overlapping particle swarms for energy-efficient routing in sensor networks," *Wireless Networks*, vol. 18, pp. 351–363, 12 2011.

[63] D.-L. Zhang, Y. Tang, J.-H. Ma, and X. Guan, "Identification of wiener model with discontinuous nonlinearities using differential evolution," *International Journal of Control, Automation and Systems*, vol. 11, pp. 511–518, 6 2013.

[64] A. Groenewolt, T. Schwinn, L. Nguyen, and A. Menges, "An interactive agent-based framework for materialization-informed architectural design," *Swarm Intelligence*, vol. 12, pp. 155–186, 12 2017.

[65] 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.

[66] P.-W. Tsai, T. Hayat, B. Ahmad, and C.-W. Chen, "Structural system simulation and control via nn based fuzzy model," *Structural Engineering and Mechanics*, vol. 56, pp. 385–407, 11 2015.

[67] B. Rachid and H. Hafid, "Distributed monitoring for wireless sensor networks: a multi-agent approach," *International Journal of Computer Network and Information Security*, vol. 6, pp. 13–23, 9 2014.

[68] K. L. Teo, H.-F. Wang, and C. Wu, "Optimization in industrial systems," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–2, 12 2014.