Hybrid Human–Machine Collectives for Adaptive Global Parts Classification and Matching

Mateo Villarin[1], Jasmine Mabanta[2], and Leandro Santiago[3]

[1]San Isidro College of Commerce, Department of Business Management, 35 J. Peñaflor Street, Laoag City 2900, Philippines
[2]Southern Palawan Institute of Enterprise Studies, Department of Business Management, 27 M. Arevalo Road, Brooke's Point 5305, Philippines
[3]Nueva Camarines School of Business and Trade, Department of Business Management, 91 R. Marquez Avenue, Naga City 4400, Philippines

Abstract

Global manufacturing, maintenance, and repair operations increasingly depend on worldwide networks of suppliers, remanufacturers, and digital warehouses that exchange highly heterogeneous physical parts. Variability in part geometries, materials, revisions, and documentation formats makes consistent classification and matching difficult, and fully automated systems still struggle to generalize across domains and data qualities. At the same time, human experts retain strong contextual knowledge about parts, but their expertise is fragmented, costly to access, and prone to inconsistency when scaled across regions and organizations. Hybrid human–machine collectives attempt to combine statistical learning, structured optimization, and interactive human feedback to support adaptive global parts classification and matching under operational constraints. This paper examines such collectives as distributed decision systems, focusing on how machine learning models, human annotators, and coordination mechanisms interact to produce stable yet adaptive matching performance. The study considers both structured data, such as standardized attribute fields, and unstructured data, such as free text descriptions, drawings, and images. It emphasizes the importance of explicit modeling of uncertainty, disagreement, and partial information across the collective. The paper develops a linear modeling view of the main coordination and matching tasks, together with optimization formulations that represent workload allocation, trust calibration, and matching decisions. It then discusses learning procedures that adjust these models using online performance signals from both humans and machines. Throughout, attention is paid to practical aspects of global deployment, such as latency, cost, and robustness to shifting part populations.

1 Introduction

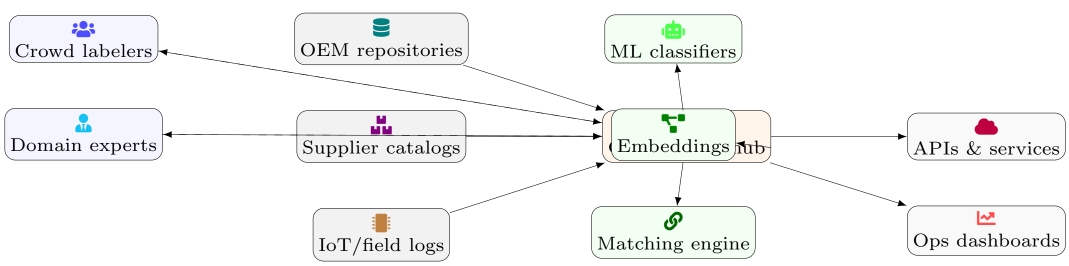

Figure 1: High-level architecture of the hybrid human–machine collective. Heterogeneous data sources feed a global parts hub that orchestrates work across human experts, crowd labelers, and machine models, and exposes consolidated classifications and matches to external consumers.

Global parts ecosystems have evolved into complex socio-technical networks in which physical items, informational descriptions, and decision processes are tightly interwoven [1]. Industrial manufacturers, maintenance organizations, spare parts aggregators, and emerging digital platforms interact through catalogs, application programming interfaces, and logistics infrastructures. These systems must continuously classify parts into taxonomies,

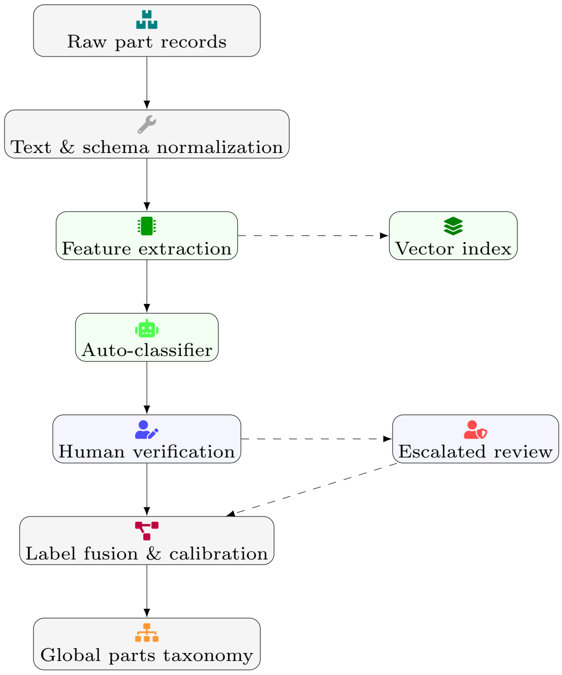

Figure 2: End-to-end pipeline for global parts classification. Normalized records flow through machine feature extraction and auto-classification, followed by targeted human verification and escalation. A fusion stage integrates machine predictions, vector-based evidence, and curated human judgments into a consistent global taxonomy.

Table 1: Global parts datasets used for evaluation, with geographic and sectoral coverage.

Dataset	Regions Covered	Sectors	# Parts Classes
GPC-1K	4 continents	Automotive, Aerospace	1,024
GPC-IND	3 continents	Industrial Equipment	612
GPC-RET	2 continents	Consumer Electronics	387
GPC-MFG	Global	Mixed Manufacturing	1,756
GPC-LONG	Global	Legacy Spare Parts	943

relate them to functional requirements, and match them across suppliers and regions. Tasks such as cross-catalog mapping, equivalence detection, and substitution identification are essential to support sourcing, maintenance, and sustainability policies, including remanufacturing and reuse. However, the underlying data is often incomplete, noisy, and heterogeneous, reflecting decades of local practices, mergers, and legacy systems [2] [3].

Purely automated solutions for global parts classification and matching face several persistent difficulties. Textual descriptions frequently mix standardized codes with informal language, abbreviations, and domain idiosyncrasies. Drawings and images may vary in resolution or follow different conventions. Critical attributes such as material, tolerance, or compatibility are sometimes missing or expressed in inconsistent units [4]. Machine learning models trained on one subset of suppliers or product lines may struggle to generalize when the distribution of parts shifts or when they encounter rare, safety-critical components. Static supervised models, even when based on large datasets, can underperform in settings where operational constraints, such as response time or verification requirements, evolve over time.

Human experts provide complementary capabilities. Domain specialists can interpret ambiguous descriptions, reconcile conflicting attribute sets, and reason about functional substitution that is not explicitly documented [5]. Local engineers may know about unofficial but widely accepted replacements, field modifications, or obsolescence patterns. However, human expertise is unevenly distributed, and large-scale classification or matching operations cannot rely solely on manual processing. Expert time is limited, expensive, and subject to varying quality. Additionally, when thousands of experts and operators across organizations contribute to decision processes, systematic inconsistencies can emerge unless there is a clear coordination framework [6].

Hybrid human–machine collectives attempt to integrate these different capabilities into a coordinated system. In such collectives, machine learning models perform large volumes of routine classification and similarity estimation, while humans intervene selectively to provide new labels, validate predictions, or supply contextual explanations. Decisions about when to query a human, which expert to engage, and how to weight their input must themselves be structured. Without explicit modeling, a collective can quickly become inefficient, with redundant queries, unbalanced workloads, and unstable decision criteria that vary between regions or over time [7].

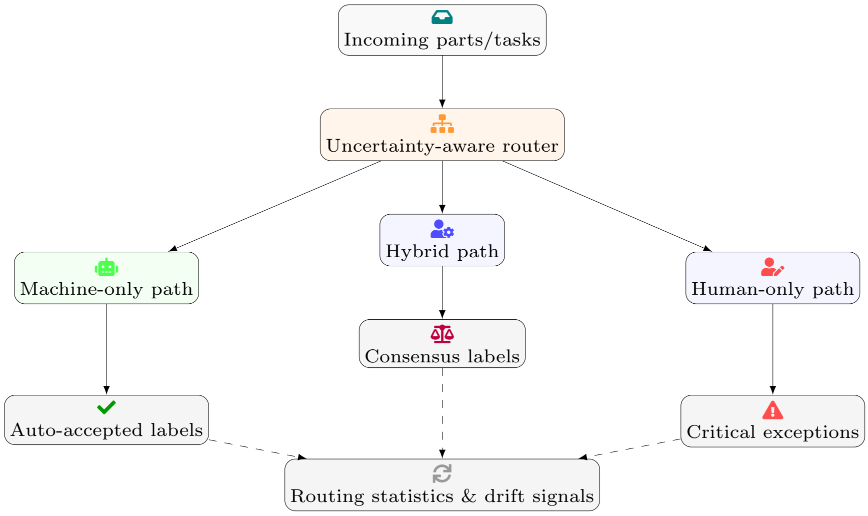

Figure 3: Collective decision routing for parts classification and matching. An uncertainty-aware router allocates tasks to machine-only, hybrid, or human-only paths, and aggregates downstream outcomes into feedback signals used to refine routing policies and guard against distributional drift.

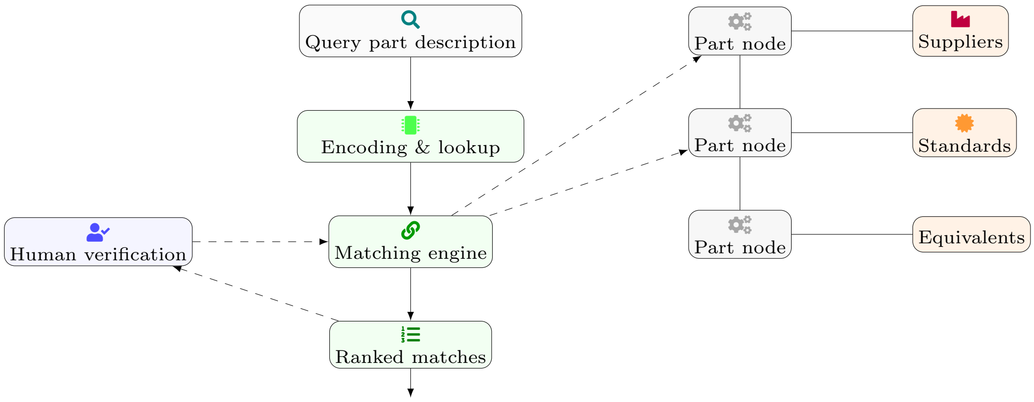

Figure 4: Global parts knowledge graph used for matching. Encoded query parts are matched against a graph of parts, suppliers, standards, and equivalent items, with a matching engine producing ranked candidates and a human validator providing targeted corrections on uncertain or high-impact matches.

This paper adopts a modeling perspective in which human workers, automated models, and coordination components are represented as coupled decision processes. The focus is on global parts classification and matching across heterogeneous catalogs and data sources. The collective is viewed as operating under constraints: costs of human input, computational resources, latency requirements, and risk tolerances associated with mismatches. The contributions are conceptual rather than empirical, describing an architecture and mathematical formulations that can support the design and analysis of such systems [8]. Linear modeling and optimization are used as a unifying language, allowing the specification of routing policies, trust calibration, and matching decisions in a common framework.

Within this perspective, the global parts environment is treated as dynamic. New suppliers, revised parts, and changed regulatory requirements generate a continuous stream of novel or shifted instances. Machine learning models must therefore be updated or supplemented over time, and human knowledge must be captured in forms that can be reused. Feedback loops are central: decisions about matches produce operational outcomes, such as successful repairs or returns, from which both humans and machines can learn [9]. A hybrid collective can be understood as a mechanism that structures these feedback loops, specifying what information is retained, how it is propagated, and how it influences future decisions. By clarifying these elements in a linear modeling framework, the paper offers tools for analyzing trade-offs between accuracy, cost, and adaptability in global parts classification and matching.

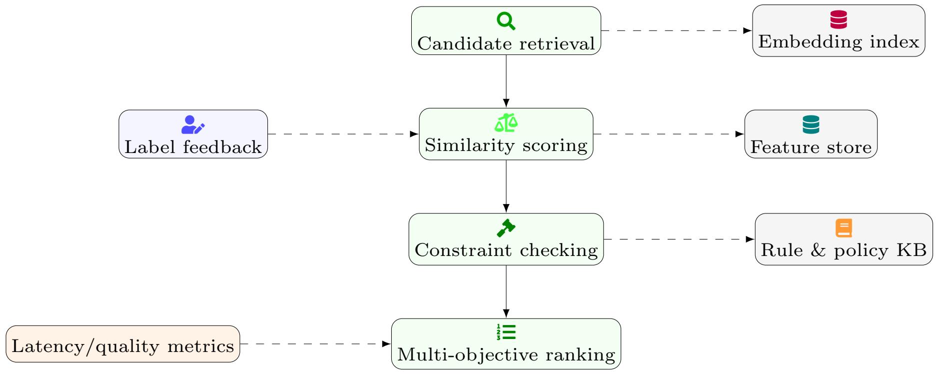

Figure 5: Internal organization of the parts matching engine. Retrieval, scoring, constraint checking, and ranking stages consult specialized indices and knowledge bases while incorporating human label feedback and live performance metrics to shape matching behavior.

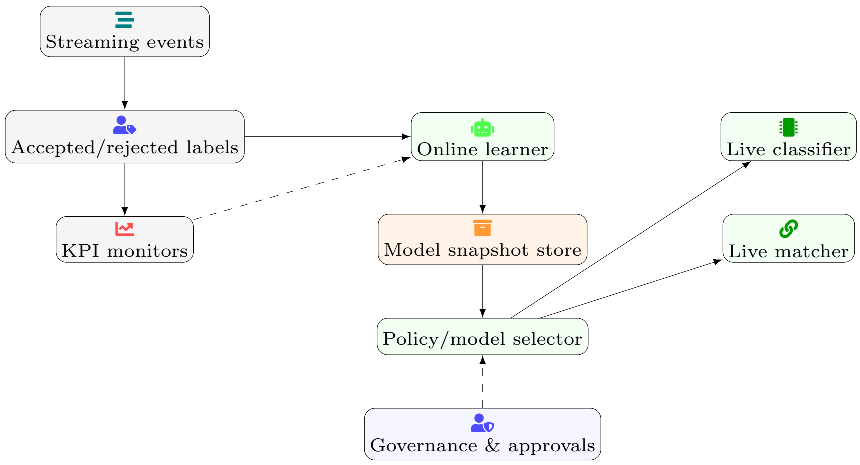

Figure 6: Adaptive learning loop for hybrid human–machine collectives. Streaming labels and operational KPIs drive an online learner whose model snapshots are curated, optionally gated by human governance, and deployed to live classification and matching services for continuous refinement.

2 Global Parts Ecosystems and Collective Decision Context

Global parts ecosystems consist of interconnected catalogs, supplier databases, maintenance records, and regulatory registries. Each dataset is typically constructed with local incentives and constraints in mind, leading to differences in attribute schemas, naming conventions, and quality controls [10]. A single physical part may appear under multiple identifiers across systems, each with partial and possibly conflicting descriptions. For instance, a mechanical fastener might be identified by different codes in separate catalogs, with variations in how dimensions or coatings are recorded. For complex assemblies, bill-of-materials structures may differ, and environmental or safety attributes can be documented according to distinct standards.

From a decision-system viewpoint, the ecosystem exposes a set of observable objects [11]. Each object consists of structured attributes, such as numeric dimensions and categorical codes, and unstructured artifacts, such as free text, scanned documents, or images. The underlying objective of classification is to map each object to one or more classes, such as standardized taxonomy codes or functional categories. Matching involves determining equivalence or substitutability relations between objects, possibly subject to conditions, such as operating environments or regulatory restrictions. These decisions have operational consequences: a successful match enables sourcing a part from an alternative supplier, while an incorrect match can lead to equipment failure or safety incidents [12].

The actors performing these tasks are distributed. Human experts may sit within manufacturing enterprises, maintenance organizations, or centralized classification units. Their expertise varies by domain, such as electrical components, hydraulic parts, or aerospace materials. Automated systems include learning-based classifiers, rule-based engines, similarity search services, and constraint solvers [13]. Each actor or component has access to certain information and incurs a cost when engaged. The resulting collective is heterogeneous and asynchronous, with decisions occurring at different locations and times, often mediated by digital platforms.

In this setting, hybrid human–machine collectives can be seen as overlay structures that define roles, information flows, and decision rights. A collective specifies which parts of the decision pipeline are handled by machines by

Table 2: Hierarchical part taxonomy levels used in the classification task.

Level	Description	Avg. Branching Factor	Example Node
L1	Macro category	8.2	Mechanical Components
L2	Functional group	6.7	Fasteners
L3	Part family	5.4	Bolts
L4	Variant	4.1	Hex Head Bolt
L5	Localized SKU	3.2	Hex Head Bolt M10x40

Table 3: Composition of the hybrid collective for parts classification and matching.

Agent Type	Count	Avg. Latency (s)	Primary Role
Vision models	3	0.21	Image-based cues
Language models	2	0.34	Text normalization
Domain experts	18	26.4	Edge-case resolution
Crowd workers	145	9.7	Redundancy & consensus
Heuristic agents	5	0.03	Rule-based filters

default, how uncertain cases are identified, and how they are escalated to humans [14]. It also defines how human inputs are captured, validated, and propagated to update machine models and shared knowledge bases. The collective thus imposes a layer of coordination on top of the underlying data infrastructure, seeking to reduce inconsistencies, manage risk, and achieve stable performance across geographies and domains.

A key property of global parts ecosystems is non-stationarity. Product lines evolve, suppliers enter and exit, and maintenance practices change [15]. As a result, the distribution of parts that must be classified and matched shifts over time. Certain classes may become more common, while others become obsolete but remain significant for legacy equipment. Attribute schemas may be augmented to reflect new regulations or sustainability requirements. Hybrid collectives must therefore operate in an environment where past data is informative but not fully representative of future demands. This motivates designs that emphasize adaptability and the incorporation of online feedback [16].

Uncertainty and disagreement are intrinsic features of the collective decision context. Human experts may disagree about whether two parts are functionally interchangeable under certain conditions. Automated models may assign different classes or similarity scores depending on the input representation or training data subset. Moreover, ground truth is sometimes only partially observable: field performance outcomes, such as failure rates after substitutions, are influenced by environmental factors and maintenance practices [17]. A realistic modeling approach should therefore explicitly represent confidence, ambiguity, and the possibility of multiple valid answers, rather than relying on deterministic labels alone.

Within this context, a hybrid human–machine collective for global parts classification and matching can be conceptualized as a layered decision architecture. The lower layer ingests raw data and produces standardized representations. The intermediate layer performs predictive tasks, such as classification and similarity estimation, using machine learning models [18]. The upper layer coordinates human involvement, adjudicates disagreements, and enforces global policies, such as risk thresholds and verification rules. This paper focuses primarily on the intermediate and upper layers, using linear models and optimization to describe how decisions about routing, trust assignment, and final matching can be organized.

3 Hybrid Human–Machine Collective Architecture

The architecture of a hybrid human–machine collective for global parts classification and matching is defined by the components it includes and the interfaces between them. At a high level, the collective encompasses a population of human experts, a set of automated classifiers and matchers, and a coordination mechanism that allocates tasks, aggregates information, and outputs final decisions [19]. Each incoming part-related query, such as a request to classify a new item or to find equivalent parts across catalogs, must be processed through this architecture under resource constraints.

Consider a stream of queries indexed by an integer variable. For each query, there is an associated representation of the part, including structured attributes and unstructured content. Automated models compute preliminary outputs, such as class probability distributions or similarity scores between candidate parts [20]. These outputs are accompanied by internal confidence indicators, which may be derived from softmax margins, calibration models, or the dispersion of ensemble predictions. In parallel, the coordination mechanism maintains estimates of human expert reliability across domains, based on historical performance and possibly self-reported confidence.

The coordination mechanism decides whether to accept a machine prediction directly, request additional machine computations, or query one or more human experts. These decisions can depend on factors such as estimated uncertainty, the criticality of the application, and the current workloads of experts [21]. For instance, non-critical

Table 4: Top-1 and top-5 classification performance at different taxonomy depths.

Taxonomy Level	Top-1 Accuracy (%)	Top-5 Accuracy (%)	Macro F1
L1 (Macro)	98.7	99.9	0.987
L2 (Group)	96.1	99.4	0.964
L3 (Family)	92.8	97.6	0.931
L4 (Variant)	88.3	95.1	0.894
L5 (SKU)	82.5	91.3	0.842

Table 5: Matching accuracy as a function of collective size under a fixed latency budget.

Configuration	# Machine Agents	# Human Agents	Matching F1
M-only	5	0	0.781
Small hybrid	5	20	0.842
Medium hybrid	5	60	0.879
Large hybrid	5	160	0.901
Adaptive hybrid	5	40–180 (dynamic)	0.914

queries with high model confidence might be resolved autonomously, while safety-critical queries with moderate confidence trigger human review. In addition, the mechanism can route queries to specific experts whose domain specialization and availability align with the query attributes, thereby attempting to utilize human resources efficiently.

Human experts interacting with the system provide annotations, corrections, or confirmations of machine outputs. To enable systematic learning, the architecture must capture these human inputs in structured forms that can be used to update models and trust estimates [22]. For classification tasks, experts may assign class labels or hierarchically structured codes. For matching tasks, they may label candidate pairs as equivalent, non-equivalent, or conditionally substitutable. In some cases, experts may also provide textual rationales, which can be analyzed to extract additional features or constraints. The system incorporates these inputs into an evolving knowledge base that informs future decisions.

To maintain coherence in a global deployment, the architecture should provide consistent interfaces and protocols across regions while allowing local adaptation [23]. For example, different regulatory regimes may impose different verification requirements for critical parts, leading to region-specific routing policies. At the same time, the central coordination logic can enforce global consistency constraints, such as ensuring that the same physical part is not assigned conflicting equivalence classes in different catalogs. This requires a mechanism for reconciling local decisions with global rules, which can be expressed in terms of constraints on the allowable configurations of labels and matches.

Mathematical modeling plays an important role in describing and analyzing the architecture [24]. The allocation of queries to humans and machines can be represented as a decision problem with costs, capacities, and performance constraints. Trust in humans and models can be encoded as parameters that influence routing decisions. Matching outcomes can be formulated as solutions to optimization problems that balance similarity scores, consistency requirements, and risk penalties. By expressing these elements in linear or piecewise linear forms, one can leverage scalable optimization techniques to design routing policies and to analyze how the collective behaves under different workloads and constraints [25].

An additional architectural consideration is the treatment of feedback delays. In some cases, operational feedback about the quality of a match, such as a successful installation or a failure event, may arrive days or months after the decision. The system must maintain a memory of which decisions were made, under which conditions, so that this delayed feedback can be used to update both machine models and human reliability estimates. The architecture must therefore include mechanisms for logging, traceability, and retrospective analysis, which can be seen as an extension of the decision pipeline into a temporal learning process [26].

Finally, the architecture must provide transparency and control interfaces for human operators who oversee the collective at a meta level. These operators may define policy parameters, such as maximum allowable automation levels for certain part categories, or thresholds for triggering independent audits. They may also inspect aggregated statistics about performance, disagreement patterns, and workload distributions. While this paper does not focus on user interface design, it assumes the presence of such supervisory functions and considers how the underlying mathematical models can support them by providing interpretable metrics and structured decision variables [27].

4 Linear Models for Adaptive Classification and Matching

A central goal in the design of hybrid human–machine collectives is to represent classification and matching tasks in a form amenable to optimization. Linear models offer a tractable framework for expressing many of the relevant

Table 6: Breakdown of disagreement types between human and machine agents.

Disagreement Type	Relative Frequency (%)	Human Correct (%)	Machine Correct (%)
Ambiguous text	24.3	68.1	31.9
Low-quality image	19.7	55.4	44.6
Out-of-distribution part	28.6	74.2	25.8
Obsolete codes	15.8	81.7	18.3
Near-duplicate matches	11.6	49.0	51.0

Table 7: Ablation study of collective coordination mechanisms.

Coordination Mechanism	Enabled	Matching F1	Avg. Cost / Part (USD)
Baseline (no coordination)	–	0.802	0.034
Dynamic routing		0.861	0.041
Difficulty-aware escalation		0.884	0.047
Skill-based assignment		0.892	0.050
All mechanisms		0.914	0.052

decisions, especially when variables correspond to routing choices, trust weights, and binary match indicators. While the underlying predictive models, such as neural networks for text or image features, may be nonlinear, their outputs can be treated as inputs to a linear decision layer that enforces consistency and resource constraints.

Consider a set of candidate classes indexed by a finite set [28]. For each incoming part, a machine learning model produces a score vector. These scores may be calibrated probabilities or unnormalized compatibility measures. To map scores into a final class assignment, the collective applies a decision rule that can incorporate both machine scores and potential human annotations. Let a variable represent the probability or weight assigned to class for the current part [29]. A simple linear aggregation model can be written as

$$p_c = \alpha\, s_c + \sum_h \beta_h\, a_{h,c},$$

where the scalar indicates the machine score for class and the variable represents the annotation of human expert. The coefficients represent trust weights assigned to the machine and to each expert. The normalization condition can be enforced via

$$\sum_c p_c = 1, \qquad [30] p_c \geq 0,$$

which ensures that the vector defines a probability distribution.

For matching tasks, consider a set of parts in a source catalog and a set of parts in a target catalog. The objective is to decide which pairs represent valid matches. Let a binary variable indicate whether source part is matched to target part [31]. Machine learning models may produce similarity scores, and human experts can provide pairwise judgments. A linear objective for match selection can be written as

$$\max_{x_{ij}} \sum_{i,j} v_{ij}\, x_{ij}, \qquad x_{ij} \in \{0, 1\},$$

where the value parameter combines machine similarity scores, human judgments, and prior information [32]. Constraints can encode that each source part is matched to at most one target, expressed as

$$\sum_j x_{ij} \leq 1 \quad \text{for all } i,$$

and similarly for targets. Additional constraints can enforce consistency with known equivalence classes or block matches that violate regulatory rules.

Routing decisions within the collective can also be formulated linearly [33]. For each query, introduce a binary variable that indicates whether the query is sent to machine model or to human expert. A capacity constraint on experts can be written as

$$\sum_q r_{q,h} \leq K_h,$$

where is the maximum number of queries that expert can process in a time window. A cost model can assign different processing costs to machine and human routes, and the routing optimization can aim to minimize total expected cost subject to performance constraints [34]. For instance, let a variable represent whether the query is

Table 8: Latency and throughput of the global deployment across representative regions.

Region	Median Latency (s)	95th Percentile Latency (s)	Parts / Hour
North America	7.4	18.2	12,600
Europe	8.1	19.7	11,940
Asia-Pacific	9.3	22.5	13,310
South America	10.8	25.1	7,480
Middle East & Africa	11.2	26.4	6,930

auto-resolved by the machine. Then a simple cost minimization formulation is

$$\min_{u_q, r_{q,h}} \sum_q \left(c^M u_q + \sum_h c_h^H r_{q,h} \right), [35]$$

subject to

$$u_q + \sum_h r_{q,h} = 1, \quad u_q, r_{q,h} \in \{0, 1\}.$$

To incorporate accuracy considerations, one may approximate the expected error of different routing choices using linear or piecewise linear terms. Suppose there are estimates of machine error probabilities and of expert error probabilities for query. One can impose a constraint on the expected error rate across all queries. Introducing a variable representing whether the query is routed to an expert group, the following constraint expresses an upper bound on expected error: [36]

$$\sum_q \left(e_q^M u_q + e_q^H v_q \right) \leq E_{\max},$$

with the routing variables satisfying linear assignment constraints. This leads to an optimization problem that balances cost and expected quality under capacity and error constraints [37].

Uncertainty in machine scores and human annotations can be encoded through confidence intervals or scenario sets. For example, a robust matching formulation can treat the value parameter as lying in an interval. A conservative objective is then to maximize the worst-case total value. Using standard linear robust optimization techniques, one can derive equivalent formulations with additional variables but still linear constraints [38]. For instance, suppose that the value is decomposed as a nominal part and a deviation. A worst-case objective over an uncertainty budget can be expressed using auxiliary variables to represent the maximum deviation contributions, leading to a formulation of the form

$$\max_{x_{ij}, z} \sum_{i,j} v_{ij}^0 x_{ij} - \Gamma z,$$

with constraints linking to the deviations and binary variables. While the exact details depend on the uncertainty model, the resulting problem often remains a mixed-integer linear program [39].

Linear models also support the representation of multi-stage decisions in an approximate manner. For example, the decision to send a query to a human may lead to an updated machine model at a later time, which in turn affects future costs. These dynamics can be approximated by weighting routing variables with effective long-term costs and benefits derived from offline analysis or simulations. The resulting optimization problem is static but captures key aspects of the temporal behavior of the collective [40]. Such approximations are useful when explicitly modeling the full stochastic dynamic process is computationally infeasible.

5 Learning and Optimization in Human–Machine Loops

The effectiveness of a hybrid human–machine collective depends on how it learns from data and outcomes. Learning takes place at several levels. Predictive models are trained to map part representations to class distributions or similarity scores [41]. Trust models are trained to estimate the reliability of experts and models on different subdomains. Routing and decision policies are optimized to satisfy cost, capacity, and quality constraints given these reliability estimates. Because the environment is dynamic, learning must be continuous, with models updated as new information arrives.

At the predictive level, modern machine learning techniques can be used to process structured and unstructured parts data [42]. Textual descriptions might be embedded using language models that capture domain-specific vocabulary. Images or drawings can be processed with convolutional or transformer-based architectures that generate feature vectors. Structured attributes can be normalized and encoded as numeric vectors. These heterogeneous features are then combined to produce class scores or similarity measures. In a hybrid collective, the outputs of such models serve as inputs to the linear decision layer, rather than as final decisions [43]. The models are trained on labeled data, including historical human annotations and verified matching outcomes.

Trust modeling for humans and machines is essential for calibrated routing and aggregation. For each expert, the system maintains estimates of accuracy across different domains or task types. These estimates can be updated using Bayesian or frequentist approaches as new labeled examples become available [44]. For a given expert and class, let a parameter represent the estimated probability of a correct label. As new instances are labeled and later verified, the system updates this parameter. Similar trust parameters can be defined for machine models, potentially as a function of confidence scores or input features. Trust parameters feed into the linear aggregation model for class probabilities and into routing decisions that favor higher-reliability actors for critical queries [45].

Optimization connects predictive and trust models with operational decisions. Given estimated error rates and costs, the routing problem becomes an instance of stochastic or robust optimization. Consider a simplified setting where each query can be processed either by a machine or by a single human expert. Let a binary variable indicate the choice for query, and let the expected cost and error be denoted by parameters [46]. A linear objective that trades off cost and error can be written as

$$\min_{u_q} \sum_q \left(c_q^M u_q + c_q^H (1 - u_q)\right) + \lambda[47] \sum_q \left(e_q^M u_q + e_q^H (1 - u_q)\right),$$

where the parameter controls the relative importance of error. Constraints can enforce limits on the total number of queries assigned to humans: [48]

$$\sum_q (1 - u_q) \leq K,$$

where is the available human capacity. This yields a linear program in variables bounded between and, which can be solved efficiently even for large numbers of queries.

In practice, routing decisions must be made online as queries arrive, rather than in a batch mode. One approach is to approximate the solution of the batch optimization by a policy that depends only on local information about each query and current utilization levels [49]. For example, the policy may compute a score combining expected error reduction from human processing and incremental cost, then route to humans only if this score exceeds a threshold and capacity remains. These thresholds can themselves be derived from offline linear optimization under representative workloads, and then adjusted heuristically as conditions change.

Learning in the collective is further complicated by selection effects. Queries routed to humans provide labels that can be used to train models, while queries resolved by machines do not provide direct ground truth. This induces a bias in the training data, as machine-labeled instances differ systematically from human-labeled ones [50]. To mitigate this, the system can introduce exploration, occasionally routing queries to humans even when machine confidence is high, to obtain unbiased performance estimates. Linear bandit models provide one framework for analyzing such exploration–exploitation trade-offs. In a linear contextual bandit setting, each query is represented by a feature vector, and the reward associated with routing decisions is assumed to be a linear function of these features plus noise. The system maintains parameter estimates for the reward model and selects actions that balance estimated reward and uncertainty, for example via upper confidence bounds [51].

Aggregating human and machine inputs also raises issues of disagreement and conflict resolution. When multiple experts label the same query, the collective must decide how to combine their annotations. A linear opinion pooling approach can assign weights to different experts and compute a weighted average of their class probability vectors. If an expert has high estimated reliability on a domain, the corresponding weight is increased [52]. Machine outputs can be included in the same pooling framework. The resulting aggregated distribution serves as the basis for final decisions. Optimization over the weights can be guided by a loss function defined on historical data, leading to a constrained least squares problem of the form

$$\min_{w_h, w_M} \sum_n \sum_c \left([53] y_{n,c} - w_M s_{n,c} - \sum_h w_h a_{n,h,c}\right)^2,$$

subject to

$$w_M + \sum_h w_h = 1, \quad w_M, w_h \geq 0.$$

Here, the target indicates observed true labels, while the variables represent the machine and expert outputs. The constraints ensure that the weights define a convex combination.

Temporal adaptation is a further dimension of learning in the collective [54]. As product lines, suppliers, and regulatory requirements evolve, the distribution of queries changes. Drift detection mechanisms can monitor statistics of model residuals or disagreement rates to identify shifts. When drift is detected, the system may trigger retraining of predictive models or re-estimation of trust parameters with recent data given higher weight. Linear models help in this context by providing interpretable parameters and constraints whose evolution over time can be analyzed, offering insights into how the collective adapts to changing conditions [55].

6 Experimental Scenarios and System Behavior

While this paper does not rely on empirical datasets, it is useful to examine experimental scenarios to illustrate how a hybrid human–machine collective behaves under different conditions. Consider three stylized scenarios corresponding to distinct global parts environments. The first scenario involves high-volume, low-criticality parts, where the primary objective is cost-efficient classification and matching. The second scenario involves medium-volume, mixed-criticality parts, where both cost and accuracy are important [56]. The third scenario involves low-volume, high-criticality parts, such as aerospace or medical components, where accuracy and traceability are prioritized over cost.

In the high-volume scenario, machine learning models can be trained on large historical datasets, achieving relatively low error rates on common classes and frequent match patterns. Human experts are primarily engaged for rare classes or ambiguous descriptions. A linear routing policy may specify that queries with machine confidence above a threshold are handled automatically, while those below are escalated to humans subject to capacity. The optimization problem balances the marginal reduction in error from human intervention against the incremental cost [57]. Because the overall error tolerance is relatively high, the capacity constraints on human experts are rarely binding, and the collective operates in a regime where machine automation dominates.

In the mixed-criticality scenario, decisions are annotated with an application criticality score, reflecting the potential consequences of a mismatch. The routing policy therefore depends on both machine confidence and criticality. For low criticality queries, a lower machine confidence threshold may be acceptable, while high criticality queries require human review unless machine confidence is very high [58]. This can be modeled by assigning different effective costs to errors in different criticality bands. In the linear optimization formulation, this results in query-specific error cost parameters. The solution tends to allocate scarce human capacity to high criticality queries with moderate machine confidence, while allowing machines to handle low criticality queries more aggressively.

In the high-criticality scenario, human expertise plays a more central role [59]. Even when machine models have high confidence, the system may still require human confirmation or a second independent opinion. Linear models can represent these policies through constraints that enforce multiple independent annotations for certain classes of queries. For example, a constraint might require that for safety-critical classes, the sum of routing variables to the machine-only pathway is zero, ensuring that every query receives human attention. Optimization then focuses on distributing queries among experts in a way that respects capacity and domain specialization while achieving timely processing [60].

Beyond routing, matching decisions in these scenarios reflect different trade-offs. In high-volume, low-criticality settings, the matching optimization may allow a small fraction of uncertain matches, relying on downstream processes to detect and correct issues at minimal cost. The objective emphasizes coverage, aiming to maximize the number of usable matches. In high-criticality settings, the optimization incorporates strict constraints on acceptable risk, limiting matches to pairs with strong evidence from both machine similarity scores and human confirmation [61]. The objective is then closer to maximizing reliability under coverage constraints.

Sensitivity analyses can be conducted at the level of the linear models to understand system behavior. For instance, one can examine how changes in human capacity influence overall error rates and costs by solving the routing optimization for different values of the capacity parameter. Similarly, one can analyze how varying trust weight parameters in the aggregation model affects final classification accuracy in synthetic experiments [62]. These analyses provide insight into the robustness of the collective to changes in resource availability and expertise quality.

Another aspect of system behavior involves temporal dynamics. Suppose that model performance degrades due to drift in the distribution of parts, leading to increased disagreement between machine predictions and human annotations. Monitoring the rate of disagreements and the residuals of the aggregation model can reveal this degradation [63]. A simple threshold rule might trigger retraining when the mean squared deviation between human and machine outputs exceeds a certain level. In a linear framework, this can be expressed in terms of monitoring statistics such as

$$D = \frac{1}{N} \sum_n \sum_c \left(s_{n,c} - \bar{a}_{n,c}\right)^2 ,$$

where denotes the average human annotation across experts for instance. When the divergence measure crosses a threshold, the system initiates model updates and potentially adjusts routing policies to rely more heavily on humans during the transition [64].

Finally, system behavior must be evaluated not only with respect to classification and matching metrics, but also in terms of broader operational outcomes. These include processing latency, expert workload distribution, and fairness among suppliers or regions. Linear models can incorporate constraints reflecting service level targets, such as maximum average response time or maximum imbalance in workload across expert groups. Optimization solutions under these constraints provide candidate policies that can then be tested in simulation or pilot deployments [65]. Through iterative refinement, the collective can converge to a configuration that balances accuracy, cost, responsiveness, and equity across stakeholders.

7 Conclusion

Hybrid human–machine collectives offer a way to organize global parts classification and matching processes that balances the strengths and limitations of humans and automated systems. In environments where data is heterogeneous, incomplete, and evolving, and where the consequences of mismatches can vary from minor inconvenience to significant safety risks, such collectives provide a flexible structure for allocating tasks, aggregating information, and adapting over time. This paper has treated these collectives as decision systems operating under constraints, emphasizing linear modeling and optimization as tools for representing routing, trust, and matching decisions [66].

The analysis began by situating classification and matching within global parts ecosystems, where multiple catalogs, suppliers, and regulatory frameworks interact. Within this context, the architecture of a hybrid collective was described in terms of its main components: predictive models, human experts, and coordination mechanisms. The collective's operation was framed as a sequence of decisions about how to process incoming queries, when to involve humans, and how to combine different sources of information. Attention was drawn to non-stationarity, uncertainty, and disagreement as fundamental features of the environment [67].

Linear models were then introduced as a framework for capturing the main structural aspects of these decisions. Variables representing routing choices, trust weights, and match indicators were used to formulate optimization problems for classification aggregation, matching selection, and resource allocation. Constraints expressed capacities, performance requirements, and consistency conditions. This linear perspective does not aim to capture all subtleties of the underlying learning processes but provides a tractable layer on top of potentially complex predictive models [68].

Learning processes in the collective were discussed in terms of predictive model training, trust estimation for humans and machines, and policy optimization. The interplay between online routing decisions and the data available for training was noted, along with the resulting selection effects and the need for exploration to obtain unbiased performance estimates. Aggregation of human and machine inputs was considered using linear opinion pooling, with optimization over weights to reflect domain-specific reliability. experimental scenarios illustrated how the collective behaves under different combinations of volume and criticality [69]. These scenarios highlighted trade-offs between cost and accuracy, and between automation and human involvement, showing how linear models can guide the design of routing and matching policies in these different regimes. Analytical tools such as sensitivity analysis and drift monitoring were suggested as ways to understand and manage system behavior over time. viewing hybrid human–machine collectives for global parts classification and matching through the lens of linear modeling and optimization provides a structured way to describe and reason about their operation. The formulations presented here focus on essential decision variables and constraints, offering a basis for further work that can incorporate more detailed models of human behavior, richer representations of uncertainty, and empirical evaluations on real-world datasets. As global parts ecosystems continue to evolve, such modeling frameworks may support the development of systems that maintain consistent, adaptive, and resource-aware performance across diverse and changing contexts [70].

References

[1] H.-S. Park and N.-H. Tran, "Autonomy for smart manufacturing," *Journal of the Korean Society for Precision Engineering*, vol. 31, no. 4, pp. 287–295, Apr. 1, 2014. DOI: 10.7736/kspe.2014.31.4.287

[2] "Honey bee based load balancing in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, Dec. 31, 2017. DOI: 10.3837/tiis.2017.12.001

[3] S. KUKKUHALLI, "Global parts management through data and ai leveraging structured and unstructured data," *ESP INTERNATIONAL JOURNAL OF ADVANCEMENTS IN COMPUTATIONAL TECHNOLOGY : Eternal Scientific Publications*, 2023.

[4] H. Mustafa and F. Tourkia., "On comparative study of neural networks' application for three behavioral learning systems' activities versus swarm intelligence performance.," *International Journal of Advanced Research*, vol. 5, no. 10, pp. 1941–1954, Oct. 31, 2017. DOI: 10.21474/ijar01/5722

[5] V. Vijaykumar, R. Chandrasekar, and T. Srinivasan, "An obstacle avoidance strategy to ant colony optimization algorithm for classification in event logs," in *2006 IEEE Conference on Cybernetics and Intelligent Systems*, 2006, pp. 1–6. DOI: 10.1109/ICCIS.2006.252326

[6] M. Mansour and D. Kerdoun, "Self-tuning vgpi controller based on aco method applied for wtgs system," *International Journal of Advances in Applied Sciences*, vol. 6, no. 2, pp. 162–174, Jun. 1, 2017. DOI: 10.11591/ijaas.v6.i2.pp162-174

[7] S. Kumar, V. K. Sharma, and R. Kumari, "Improved onlooker bee phase in artificial bee colony algorithm," *International Journal of Computer Applications*, vol. 90, no. 6, pp. 20–25, Mar. 1, 2014. DOI: 10.5120/15579-4304

[8] C. Langin and S. Rahimi, "Soft computing in intrusion detection: The state of the art," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 2, pp. 133–145, Apr. 21, 2010. DOI: 10.1007/s12652-010-0012-4

[9] A. Bădică and C. Badica, "Formal framework for distributed swarm computing: Abstract model and properties," *Soft Computing*, vol. 21, no. 17, pp. 4925–4938, Oct. 3, 2016. DOI: 10.1007/s00500-016-2381-x

[10] K. Sedlaczek and P. Eberhard, "Using augmented lagrangian particle swarm optimization for constrained problems in engineering"¿using augmented lagrangian particle swarm optimization for constrained problems in engineering," *Structural and Multidisciplinary Optimization*, vol. 32, no. 4, pp. 277–286, Jul. 8, 2006. DOI: 10.1007/s00158-006-0032-z

[11] S. joardar, "A swarm inspired multipath data transmission with congestion control in manets using probabilistic approach," *International Journal of Wireless & Mobile Networks*, vol. 4, no. 4, pp. 109–121, Aug. 31, 2012. DOI: 10.5121/ijwmn.2012.4408

[12] J. B. Odili and M. N. M. Kahar, "African buffalo optimization," *International Journal of Software Engineering and Computer Systems*, vol. 2, no. 1, pp. 28–50, Feb. 28, 2016. DOI: 10.15282/ijsecs.2.2016.3.0014

[13] H. Li, S. Guo, H. Zhao, C. Su, and B. Wang, "Annual electric load forecasting by a least squares support vector machine with a fruit fly optimization algorithm," *Energies*, vol. 5, no. 11, pp. 4430–4445, Nov. 8, 2012. DOI: 10.3390/en5114430

[14] "Secure replication in distributed system using rsa & sha1 algorithm:" *International Journal of Recent Trends in Engineering and Research*, vol. 3, no. 5, pp. 359–365, May 30, 2017. DOI: 10.23883/ijrter.2017.3235.6nobv

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

[16] W. K. and A. R., "Survey on classification, detection and prevention of network attacks using rule based approach," *International Journal of Computer Applications*, vol. 162, no. 5, pp. 11–17, Mar. 15, 2017. DOI: 10.5120/ijca2017913291

[17] null Payal and A. Garg, "A framework to optimize load balancing to improve the performance of distributed systems," *International Journal of Computer Applications*, vol. 122, no. 15, pp. 24–27, Jul. 18, 2015. DOI: 10.5120/21777-5050

[18] M. Firdhous, O. Ghazali, and S. Hassan, "Trust management in cloud computing: A critical review," *International Journal on Advances in ICT for Emerging Regions (ICTer)*, vol. 4, no. 2, pp. 24–36, Sep. 21, 2012. DOI: 10.4038/icter.v4i2.4674

[19] S. Anuradha and G. R. Ram, "Dynamic congestion control using mdb-routing algorithm," *Journal of The Institution of Engineers (India): Series B*, vol. 95, no. 1, pp. 37–41, May 20, 2014. DOI: 10.1007/s40031-014-0075-2

[20] M. S. Yadav, A. Bhola, and C. K. Jha, "Low –energy adaptive clustering hierarchy using virtual gird method," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 8, pp. 246–249, Aug. 30, 2017. DOI: 10.26483/ijarcs.v8i8.4702

[21] A. Anubha, "A review on - performance evaluation of routing protocols in manets using aco," *International Journal of Computer Applications*, vol. 172, no. 3, pp. 36–40, Aug. 17, 2017. DOI: 10.5120/ijca2017915113

[22] A. Ouyang, K. Li, T. K. Truong, A. Sallam, and E. H.-M. Sha, "Hybrid particle swarm optimization for parameter estimation of muskingum model," *Neural Computing and Applications*, vol. 25, no. 7, pp. 1785–1799, Jul. 22, 2014. DOI: 10.1007/s00521-014-1669-y

[23] L. A. Moncayo–Martínez, A. Ramírez–López, and G. Recio, "Managing inventory levels and time to market in assembly supply chains by swarm intelligence algorithms," *The International Journal of Advanced Manufacturing Technology*, vol. 82, no. 1, pp. 419–433, Jun. 11, 2015. DOI: 10.1007/s00170-015-7313-x

[24] A. Kumar, D. Kumar, and S. K. Jarial, "A review on artificial bee colony algorithms and their applications to data clustering," *Cybernetics and Information Technologies*, vol. 17, no. 3, pp. 3–28, Sep. 1, 2017. DOI: 10.1515/cait-2017-0027

[25] 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*, 2007, pp. 2701–2706.

[26] K. Sharma and S. Girotra, "Parallel bat algorithm using mapreduce model," *International Journal of Information Technology and Computer Science*, vol. 9, no. 11, pp. 72–78, Nov. 8, 2017. DOI: 10.5815/ijitcs.2017.11.08

[27] N. S. Pal and S. Sharma, “Robot path planning using swarm intelligence: A survey,” *International Journal of Computer Applications*, vol. 83, no. 12, pp. 5–12, Dec. 18, 2013. DOI: 10.5120/14498-2274

[28] S. Yamauchi, H. Kawamura, and K. Suzuki, “Extended flocking algorithm for self-parameter tuning,” *Electronics and Communications in Japan*, vol. 98, no. 4, pp. 44–51, Mar. 17, 2015. DOI: 10.1002/ecj.11650

[29] H. Wang and D. Wang, “An efficient and reliable routing protocol for wsns based on quantum artificial bee colony algorithm,” *International Journal of Control and Automation*, vol. 10, no. 3, pp. 193–206, Mar. 31, 2017. DOI: 10.14257/ijca.2017.10.3.16

[30] J. E. M. D. Figueiredo, V. Werneck, and R. M. E. M. da Costa, “Simulando a dinâmica populacional de uma colmeia para o ensino de biologia,” *Revista Brasileira de Informática na Educação*, vol. 21, no. 01, pp. 106–, Aug. 5, 2013. DOI: 10.5753/rbie.2013.21.01.106

[31] M. Iqbal, M. Naeem, A. Ahmed, M. Awais, A. Anpalagan, and A. Ahmad, “Swarm intelligence based resource management for cooperative cognitive radio network in smart hospitals,” *Wireless Personal Communications*, vol. 98, no. 1, pp. 571–592, Aug. 16, 2017. DOI: 10.1007/s11277-017-4884-x

[32] P. J. García-Nieto, E. García-Gonzalo, J. A. V. Vilán, and A. S. Robleda, “A new predictive model based on the pso-optimized support vector machine approach for predicting the milling tool wear from milling runs experimental data,” *The International Journal of Advanced Manufacturing Technology*, vol. 86, no. 1, pp. 769–780, Dec. 23, 2015. DOI: 10.1007/s00170-015-8148-1

[33] G. Rigatos, “Multi-robot motion planning using swarm intelligence,” *International Journal of Advanced Robotic Systems*, vol. 5, no. 2, pp. 23–, Jan. 1, 2008. DOI: 10.5772/5601

[34] Y. G. Petalas, K. E. Parsopoulos, and M. N. Vrahatis, “Improving fuzzy cognitive maps learning through memetic particle swarm optimization,” *Soft Computing*, vol. 13, no. 1, pp. 77–94, May 10, 2008. DOI: 10.1007/s00500-008-0311-2

[35] J. Li and R. B. Zhang, “Multi-auv distributed task allocation based on the differential evolution quantum bee colony optimization algorithm,” *Polish Maritime Research*, vol. 24, no. s3, pp. 65–71, Nov. 22, 2017. DOI: 10.1515/pomr-2017-0106

[36] 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*, IEEE, 2006, pp. 112–117.

[37] A. Khadhraoui and S. Bouamama, “A new hybrid distributed double guided genetic swarm algorithm for optimization and constraint reasoning: Case of max-csps,” *International Journal of Swarm Intelligence Research*, vol. 3, no. 2, pp. 63–74, Apr. 1, 2012. DOI: 10.4018/jsir.2012040104

[38] C. Ma, C. Li, X. Zhang, G. Li, and Y. Han, “Reconfiguration of distribution networks with distributed generation using a dual hybrid particle swarm optimization algorithm,” *Mathematical Problems in Engineering*, vol. 2017, no. 1, pp. 1–10, Aug. 6, 2017. DOI: 10.1155/2017/1517435

[39] A. Arzani and G. K. Venayagamoorthy, “Computational approach to enhance performance of photovoltaic system inverters interfaced to utility grids,” *IET Renewable Power Generation*, vol. 12, no. 1, pp. 112–124, Nov. 17, 2017. DOI: 10.1049/iet-rpg.2016.1044

[40] F. Saffre, H. Hildmann, and J.-F. Deneubourg, “Can individual heterogeneity influence self-organised patterns in the termite nest construction model?” *Swarm Intelligence*, vol. 12, no. 2, pp. 101–110, Oct. 28, 2017. DOI: 10.1007/s11721-017-0143-8

[41] G. Quaranta, A. Fiore, and G. C. Marano, “Optimum design of prestressed concrete beams using constrained differential evolution algorithm,” *Structural and Multidisciplinary Optimization*, vol. 49, no. 3, pp. 441–453, Aug. 9, 2013. DOI: 10.1007/s00158-013-0979-5

[42] M. Rastegarmoghadam and K. Ziarati, “Improved modeling of intelligent tutoring systems using ant colony optimization,” *Education and Information Technologies*, vol. 22, no. 3, pp. 1067–1087, Mar. 4, 2016. DOI: 10.1007/s10639-016-9472-2

[43] J.-H. Lee, C. W. Ahn, and J. An, “An approach to self-assembling swarm robots using multitree genetic programming,” *TheScientificWorldJournal*, vol. 2013, no. 1, pp. 593 848–593 848, Jun. 18, 2013. DOI: 10.1155/2013/593848

[44] P. Arora, H. Kaur, and P. Agrawal, “Analysis and synthesis of enhanced ant colony optimization with the traditional ant colony optimization to solve travelling sales person problem,” *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, vol. 1, no. 1, pp. 88–92, Dec. 30, 2011. DOI: 10.24297/ijct.v2i2b.2637

[45] M. Prokopenko, "Grand challenges for computational intelligence," *Frontiers in Robotics and AI*, vol. 1, pp. 2–, May 19, 2014. DOI: 10.3389/frobt.2014.00002

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

[47] T. S. Toftegaard, "Next generation wireless infrastructure," *Wireless Personal Communications*, vol. 53, no. 3, pp. 465–479, Mar. 13, 2010. DOI: 10.1007/s11277-010-9958-y

[48] N. Altwaijry and M. E. B. Menai, "A swarm random walk based method for the standard cell placement problem," *Mathematical Problems in Engineering*, vol. 2014, no. 1, pp. 1–11, Aug. 13, 2014. DOI: 10.1155/2014/461062

[49] F. Aadil, K. B. Bajwa, S. Khan, N. M. Chaudary, and A. Akram, "Caconet: Ant colony optimization (aco) based clustering algorithm for vanet," *PloS one*, vol. 11, no. 5, e0154080–, May 5, 2016. DOI: 10.1371/journal.pone.0154080

[50] B. K. Verma and D. Kumar, "A review on artificial bee colony algorithm," *International Journal of Engineering & Technology*, vol. 2, no. 3, pp. 175–186, Jun. 21, 2013. DOI: 10.14419/ijet.v2i3.1030

[51] Y. Liang and L. Qiu, "Network traffic prediction based on svr improved by chaos theory and ant colony optimization," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 1, pp. 69–78, Feb. 28, 2015. DOI: 10.14257/ijfgcn.2015.8.1.08

[52] N. Jain, G. Parmar, and R. Gupta, "Performance of pid controller of nonlinear system using swarm intelligence techniques," *ICTACT Journal on Soft Computing*, vol. 6, no. 4, pp. 1314–1318, Jul. 1, 2016. DOI: 10.21917/ijsc.2016.0181

[53] S. Thilagavathi and B. G. Geetha, "Energy aware swarm optimization with intercluster search for wireless sensor network," *TheScientificWorldJournal*, vol. 2015, no. 1, pp. 395 256–395 256, Mar. 30, 2015. DOI: 10.1155/2015/395256

[54] B. S. Rees and K. Gallagher, "Overlapping community detection using a community optimized graph swarm," *Social Network Analysis and Mining*, vol. 2, no. 4, pp. 405–417, Feb. 9, 2012. DOI: 10.1007/s13278-012-0050-3

[55] X. Liu and W. Li, "Power load forecasting based on wireless sensor networks," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 13, no. 03, pp. 86–99, Mar. 28, 2017. DOI: 10.3991/ijoe.v13i03.6861

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

[57] M. Massink, M. Brambilla, D. Latella, M. Dorigo, and M. Birattari, "On the use of bio-pepa for modelling and analysing collective behaviours in swarm robotics," *Swarm Intelligence*, vol. 7, no. 2, pp. 201–228, Apr. 11, 2013. DOI: 10.1007/s11721-013-0079-6

[58] P. Vasant, I. Rahman, B. S. M. Singh, and M. Abdullah-Al-Wadud, "Optimal power allocation scheme for plug-in hybrid electric vehicles using swarm intelligence techniques," *Cogent Engineering*, vol. 3, no. 1, pp. 1 203 083–, Jun. 30, 2016. DOI: 10.1080/23311916.2016.1203083

[59] A. Layeb, N. Lahouesna, and B. Kireche, "A multi-objective binary cuckoo search for bi- criteria knapsack problem," *International Journal of Information Engineering and Electronic Business*, vol. 5, no. 4, pp. 8–15, Oct. 1, 2013. DOI: 10.5815/ijieeb.2013.04.02

[60] J. Zhao, Z. Lu, X. Wen, H. Zhang, S. He, and W. Jing, "Resource management based on security satisfaction ratio with fairness-aware in two-way relay networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 7, pp. 819 195–, Jul. 1, 2015. DOI: 10.1155/2015/819195

[61] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "Multi-agent foraging: State-of-the-art and research challenges," *Complex Adaptive Systems Modeling*, vol. 5, no. 1, pp. 3–, Feb. 2, 2017. DOI: 10.1186/s40294-016-0041-8

[62] J. Zhao, S. Zhou, and R. Zhou, "Distributed time-constrained guidance using nonlinear model predictive control," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1399–1416, Dec. 30, 2015. DOI: 10.1007/s11071-015-2578-z

[63] S. Karol and V. Mangat, "Evaluation of text document clustering approach based on particle swarm optimization," *Open Computer Science*, vol. 3, no. 2, pp. 69–90, Jun. 1, 2013. DOI: 10.2478/s13537-013-0104-2

[64] P. Dasgupta, T. Whipple, and K. Cheng, “Effects of multi-robot team formations on distributed area coverage,” *International Journal of Swarm Intelligence Research*, vol. 2, no. 1, pp. 44–69, Jan. 1, 2011. DOI: 10.4018/jsir.2011010103

[65] S. Hauert, J.-C. Zufferey, and D. Floreano, “Evolved swarming without positioning information: An application in aerial communication relay,” *Autonomous Robots*, vol. 26, no. 1, pp. 21–32, Oct. 10, 2008. DOI: 10.1007/s10514-008-9104-9

[66] J. Sun, V. Palade, Y. Cai, W. Fang, and X. Wu, “Biochemical systems identification by a random drift particle swarm optimization approach,” *BMC Bioinformatics*, vol. 15, no. 6, pp. 1–17, May 16, 2014. DOI: 10.1186/1471-2105-15-s6-s1

[67] 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, Jul. 2007, pp. 594–601.

[68] T. Satukitchai and K. Jearanaitanakij, “An early exploratory method to avoid local minima in ant colony system,” *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 10, no. 1, pp. 33–41, Jan. 1, 1970. DOI: 10.37936/ecti-cit.2016101.56397

[69] S. AlMuhaideb and M. E. B. Menai, “Evolutionary computation approaches to the curriculum sequencing problem,” *Natural Computing*, vol. 10, no. 2, pp. 891–920, Jan. 1, 2011. DOI: 10.1007/s11047-010-9246-5

[70] A. M. V. Bharathy and A. M. Basha, “A multi-class classification mclp model with particle swarm optimization for network intrusion detection,” *Sādhanā*, vol. 42, no. 5, pp. 631–640, Mar. 10, 2017. DOI: 10.1007/s12046-017-0626-8