Orient Academies

Distributed Learning of Pricing and Discount Policies from Interacting Revenue Recovery Agents

Samir Boudjelal[1], Lina Merzougui[2]

Abstract

Revenue recovery processes for subscription, credit, and utility services increasingly rely on large collections of autonomous or semi-autonomous agents interacting with customers under diverse contractual, regulatory, and behavioral conditions. These agents offer payment plans, temporary discounts, and fee waivers in an attempt to restore revenue while satisfying operational and fairness constraints. The design of effective pricing and discount policies in this setting is complicated by limited observability, heterogeneous customer responses, and strong coupling among agents through shared budgets, risk limits, and regulatory caps on concessions. Centralized optimization approaches may be difficult to deploy when data are fragmented across business units or jurisdictions and when system operators require local autonomy of existing revenue recovery teams and tools. This paper investigates distributed learning mechanisms that infer pricing and discount policies from the behavior of interacting revenue recovery agents and from their long-run performance. The discussion formulates the problem as a multi-agent sequential decision process with linearly parameterized value functions and linear constraints capturing budget, exposure, and regulatory requirements. A distributed learning architecture is introduced in which agents update local policy parameters from their own interaction histories while exchanging low-dimensional aggregate statistics that gradually coordinate their policies. Analytical results describe the structure of the induced linear systems, properties of the distributed fixed points, and conditions under which the learning dynamics remain stable. Numerical experiments on stylized revenue recovery scenarios illustrate how local exploration, policy heterogeneity, and different coordination rates influence the evolution of pricing and discount policies.

[1]Université de Ksar El Boukhari, Department of Computer Engineering, Avenue Emir Abdelkader 17, Ksar El Boukhari 26400, Algeria

[2] Université de El Oued, Department of Computer Engineering, Route Nationale 16, El Oued 39000, Algeria

Contents

1. Introduction

Revenue recovery operations arise in several application domains where customers temporarily or persistently deviate from their contractual payment obligations [1]. Typical examples include consumer finance, utilities, telecommunications, insurance, and subscription services. In these settings, organizations deploy human or automated agents who interact with customers, negotiate revised repayment schedules, and select pricing or discount offers intended to restore some portion of the outstanding revenue. Each interaction must trade off immediate recovered cash, probability of future default, operational workload, and compliance requirements [2]. These trade-offs are sharpened by the fact that agents operate repeatedly over time, learning from customer responses while being constrained by global discount budgets, exposure limits, and evolving risk policies. The resulting system

Table 1. Notation for distributed pricing and discount learning

Symbol	Description	Domain / Value
N	Number of revenue recovery agents	$\{5, 10, 20\}$
T	Time horizon (episodes)	10^4
$\mathscr{A}$	Pricing & discount action space	$[0, p_{\max}] \times [0, d_{\max}]$
s_t^i	Local state of agent i at time t	$\mathbb{R}^{d_s}$
a_t^i	Action of agent i at time t	$\mathbb{R}^2$
r_t^i	Revenue of agent i at time t	$\mathbb{R}$
γ	Discount factor for returns	0.95

is inherently dynamic, stochastic, and multi-agent.

As digital channels expand and decision support systems become more pervasive, revenue recovery agents increasingly rely on algorithmic recommendations for pricing and discount policies. These recommendations are often derived from predictive models of customer propensity to pay, stratified by risk segment, product type, and behavior history [3]. While such predictive models can be accurate, translating them into effective decision policies still poses several challenges. Recovery agents interact with customers under partially observed conditions, where only fragments of behavior, income, and external obligation information are visible. Furthermore, recovery decisions are temporally coupled, since concessions granted today can influence future delinquency and recovery probabilities in complex ways. These coupled dynamics make it difficult to calibrate static pricing or discount rules that remain effective as portfolios and macroeconomic conditions evolve [4].

Traditional approaches to designing recovery policies often assume centralized access to historical data and a single decision maker optimizing a global objective. In practice, however, revenue recovery operations are frequently decentralized across business units, regions, or outsourcing partners, each maintaining local data stores, processes, and constraints. Local teams may already follow different heuristics and operational targets, and they may resist centralized policies that override existing practices [5]. At the same time, global constraints such as aggregate discount budgets, maximum percentages of accounts eligible for aggressive concessions, or caps on specific fee waivers require coordination across units. These considerations motivate distributed learning mechanisms that adjust local pricing and discount policies using locally observed outcomes, while aligning global statistics that indirectly enforce shared constraints.

This paper explores such mechanisms by framing revenue recovery operations as a multi-agent sequential decision problem, in which each agent controls pricing and discounts for its subset of accounts while being coupled to other agents through shared budgets and performance measures. The learning problem is cast in terms of estimating policies that map customer states and account histories to pricing and discount actions [6]. The objective is to approximate policies that balance recovered revenue, fairness, and cost over a long horizon. To render the problem analytically tractable, the discussion focuses on linear function approximation and linearly constrained formulations, which allow the interaction between agents to be expressed in terms of linear systems and distributed optimization schemes [7].

Distributed learning in this context must address several sources of heterogeneity. Agents may face different customer populations, with varying sensitivities to price and discount levels, and may operate under different local policies or regulatory regimes [8]. Their exploration behavior may differ, with some agents experimenting more aggressively than others. The communication topology connecting agents may be sparse due to privacy and organizational boundaries, which limits the degree of coordination achievable through direct parameter sharing. These features make naive centralized learning approaches difficult to implement and suggest the need for architectures in which agents retain local control while still converging, in some approximate sense, toward mutually compatible pricing and discount policies [9].

The distributed learning framework examined here builds on stochastic approximation and linear multi-agent optimization. Each revenue recovery agent maintains a parameterized representation of its pricing and discount policy, together with value-function or advantage-function approximations that quantify the impact of actions on long-run revenue and constraints. Local updates are computed from observed transitions and rewards, yielding incremental improvements to the estimated value of different actions in different customer states. Coupling among agents is induced through shared dual variables or consensus constraints that depend on aggregate statistics, such as the total discounted amount of concessions or the long-run fraction of customers receiving certain discount levels [10].

Within this formulation, pricing and discount policy design reduces to solving a family of linear systems and linear programs that represent the fixed points of the learning dynamics. At the same time, the multi-agent nature of the problem requires distributed algo-

Table 2. Summary of customer cohorts used in the experiments

Cohort	# Customers	Avg Debt (USD)	Baseline Recovery Rate
Synthetic–Homogeneous	10,000	320	0.41
Synthetic–Heterogeneous	15,000	540	0.37
Telco Arrears	8,200	210	0.33
Utility Arrears	6,500	460	0.29
Banking Delinquency	12,750	980	0.24

Table 3. Agent communication topologies and spectral properties

Topology	Avg Degree	$\lambda_2(L)$	Comment
Ring	2	0.39	Slow information mixing
Grid ($\sqrt{N}\times\sqrt{N}$)	4	0.62	Balanced locality
Erdős–Rényi ($p=0.2$)	3.8	0.71	Random sparsity
Barabási–Albert	3.1	0.53	Hub concentration
Complete	$N-1$	1.00	Centralized-equivalent

rithms that respect communication constraints and preserve some level of local autonomy. This combination of linear modeling, distributed optimization, and learning from interaction yields a framework that is analytically well structured while still capturing key features of operational revenue recovery environments. The subsequent sections detail the modeling assumptions, derive the associated linear systems, propose distributed algorithms for learning pricing and discount policies, and examine their properties through analysis and simulation [11].

2. Modeling Interacting Revenue Recovery Agents

The starting point is a stylized model of revenue recovery operations involving a population of agents indexed by an index set. Each agent manages a subset of accounts and interacts with them over discrete time periods. At each period, an agent observes a local state that summarizes the condition of its accounts, including features such as days past due, outstanding balance, prior concessions, payment history, and possibly exogenous covariates representing macroeconomic indicators or behavioral scores [12]. Based on this state, the agent selects a pricing and discount action, such as adjusting fees, offering a temporary reduction, or proposing an installment plan. The customer responds by making a payment, ignoring the offer, or transitioning to another status such as escalation, legal action, or write-off. The agent receives a reward proportional to recovered revenue net of any discount granted, and the system transitions to a new state.

Formally, the local state of an agent at time index can be represented as a vector in a state space [13]. The action includes the offered price level, the discount rate, and potentially non-monetary concessions coded as discrete levels. The action space can thus be modeled as a finite or compact set depending on whether prices and discounts are discretized or treated as continuous. Each local state-action pair induces a random next state according to a transition kernel that captures customer behavior and operational rules. This kernel may depend on exogenous factors, but from the perspective of the agent it is treated as a conditional distribution over next states [14].

To express these dynamics in a concise form, let a joint local configuration be denoted by a variable that aggregates the observable state and action. For each time step, the local evolution can be summarized as a sequence of pairs, where each pair generates a reward. The instantaneous reward of agent at time may be represented as a function, where the function encodes recovered payments minus costs of concessions and operational penalties [15]. In a simple linear approximation, one may write

$$r_{i,t} = \ell_i^{\top} x_{i,t},$$

where the vector collects features of the interaction such as realized payment, discount size, and follow-up workload, and the coefficient vector represents their monetary valuation. This linear representation simplifies the analysis of aggregated rewards and the formulation of optimization problems.

The agents are not independent because they share global constraints and operate under portfolio-level objectives [16]. For example, there may be an annual discount budget limiting the cumulative discounted amount across all agents, or a regulatory constraint bounding the fraction of accounts in specific high-risk segments receiving certain fee waivers. These couplings can be represented by linear constraints of the form

$$\sum_i g_i(x_{i,t}) \le b,$$

where the function maps local features to contributions toward the constrained quantity, and the vector rep-

Table 4. Comparison of pricing and discount learning methods

Method	Norm. Revenue	Episodes to Converge	Discount Cost Ratio
Myopic Heuristic	0.71	–	1.00
Centralized DQN	0.88	6,500	0.93
Centralized PPO	0.91	5,200	0.89
Fully Decentralized DQN	0.86	8,400	0.95
Proposed Distributed PPO	0.95	4,100	0.87

Table 5. Ablation over components of the distributed learner

Variant	Norm. Revenue	Avg Regret	Constraint Violations
Full Model	0.95	0.03	0.7%
w/o Consensus Updates	0.90	0.08	1.6%
w/o Discount Regularizer	0.93	0.06	4.3%
w/o Demand Forecasting	0.89	0.10	1.9%
Single-Agent Centralized	0.91	0.07	0.9%

resents the allowed limits [17]. Aggregating over time and discounting future contributions yields a global constraint on discounted streams of concessions and exposures, which can be approximated within a linear framework by tracking suitable occupancy measures or statistics.

To capture long-run objectives, one may define, for each agent, a discounted cumulative reward

$$J_i(\pi) = [18]\mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,t}\right],$$

where is a discount factor and denotes the joint policy profile across agents. The expectation is taken with respect to the stochastic dynamics induced by the policy and the environment. The global objective is often a linear combination of these agent-level returns, such as a weighted sum representing portfolio-level revenue, risk, and fairness metrics [19]. Under a linear parameterization of the reward in terms of features, these long-run returns can be expressed as inner products between feature expectations and valuation coefficients, which naturally leads to linear programming and linear system formulations.

The interaction among agents arises not only through explicit constraints but also through shared customers or correlated external conditions. For instance, customers may be jointly served by different product lines or regions, and their payment behavior may depend on aggregate collection intensity. Modeling these richer couplings would require game-theoretic constructs that go beyond simple linear constraints [20]. In the present discussion, the focus remains on scenarios where agents are coupled through shared linear constraints and global objective coefficients, while their state transitions are conditionally independent given exogenous factors. This simplifies the underlying process to a structured multi-agent Markov decision model with linearly coupled payoffs.

Finally, the modeling framework must accommodate partial observability and limited information sharing [21]. Agents may not observe the full state of the customer or the exact actions taken by others, and privacy or regulatory considerations may restrict the type of information that can be communicated across agents. To retain tractability, it is convenient to represent unobserved components as part of a stochastic disturbance term in the transition dynamics and reward function, and to assume that agent policies are functions of their observed local states. Information sharing is then limited to aggregated statistics computed from realized trajectories, such as empirical averages of feature vectors, approximate value-function gradients, or dual variables associated with shared constraints. These modeling choices support the design of distributed learning algorithms that align pricing and discount policies without requiring full centralization of data or detailed mutual observability [22].

3. Distributed Pricing and Discount Policy Learning

Given the multi-agent model of revenue recovery interactions, the goal is to design learning algorithms through which agents adjust their pricing and discount policies based on observed outcomes and limited communication. Each agent maintains a parameterized policy that maps local states to actions. For concreteness, consider a stochastic policy in which the action distribution is given by a parametric family indexed by a vector of parameters. The policy can be viewed as assigning probabilities to different price and discount combinations conditional on the observed state, and its parameters are updated as the agent gathers experience [23].

The learning problem is to adjust the parameters so as to improve a global objective subject to shared con-

Table 6. Hyperparameters used in distributed policy optimization

Hyperparameter	Symbol	Value	Notes
Discount factor	γ	0.95	Long-term revenue focus
Learning rate	α	3×10^{-4}	Adam optimizer
Batch size per agent	B	512	On-policy trajectories
Consensus steps per epoch	K	5	Gossip iterations
Clipping parameter	ε	0.2	PPO objective
Target KL	$\mathrm{KL}_{\mathrm{target}}$	0.03	Early stopping heuristic

Table 7. Sensitivity of recovered revenue to price and discount bounds

Price / Discount Range	Norm. Revenue	Avg Discount	Contact Attempts
Low price, low discount	0.78	4.2%	1.6
Low price, high discount	0.84	11.9%	1.9
Mid price, mid discount	0.91	7.8%	1.7
High price, low discount	0.87	3.5%	1.4
High price, high discount	0.90	10.7%	1.8

straints. One convenient approach is to combine local temporal-difference learning with distributed optimization. Each agent estimates a value function that approximates the expected discounted reward starting from a given state under the current policy [24]. With linear function approximation, the value function of agent can be written as

$$V_i(s) = \phi_i(s)^\top \theta_i,$$

where is a vector of features derived from the state and is a parameter vector. Temporal-difference updates attempt to adjust so that the approximate value function satisfies a Bellman-like relation [25]. For a transition from state to next state with reward, an incremental update of the form

$$\theta_i^{t+1} = \theta_i^t + \alpha_t \delta_{i,t} \phi_i(s_{i,t})$$

is applied, where is a step size and is a temporal-difference error computed as

$$\delta_{i,t} = r_{i,t} + \gamma \phi_i(s_{i,t+1})^\top \theta_i^t - \phi_i(s_{i,t})^\top \theta_i^t.$$

These recursions yield an approximate evaluation of the current policy for each agent based on local data [26].

To transform value estimates into improved pricing and discount policies, one can employ policy-gradient-style updates or actor-critic structures. An agent may maintain a separate policy parameter vector and update it in the direction of an estimated gradient of the global or local objective with respect to. For example, an actor-critic iteration of the form [27]

$$w_i^{t+1} = w_i^t + \eta_t g_i^t$$

can be used, where is a policy step size and is a stochastic estimate of the gradient based on the current trajectory, value-function parameters, and policy. Under suitable conditions, these coupled value and policy updates converge toward stationary points of an approximate performance surface. The distributed nature of the problem arises because the gradient estimates depend on quantities that are coupled across agents through shared constraints and global objective terms [28].

To handle shared budgets or exposure constraints, a useful technique is to introduce dual variables that penalize constraint violations. Let a vector of dual variables be associated with a set of linear constraints on discounted concession statistics or other aggregated quantities. The global objective can then be augmented with a term representing the inner product of the dual variables and the constraint residuals, yielding a Lagrangian that decomposes into local components. Each agent modifies its local reward signal by subtracting or adding a linear term involving the dual variables and local features [29]. This leads to an effective reward

$$\tilde{r}_{i,t} = r_{i,t} - \lambda^\top h_i(x_{i,t}),$$

where collects the features entering the constraint and is the current dual vector broadcast to agents. The learning updates then proceed as before but with replaced by the augmented reward [30].

The dual variables themselves can be updated in a distributed manner using subgradient or stochastic approximation rules based on observed constraint residuals. A typical dual update has the form

$$\lambda^{t+1} = \Big[\lambda^t + \beta_t [31] \sum_i \big(h_i(x_{i,t}) - b_i\big)\Big]_+,$$

where is a dual step size, represents local contributions to the constraint, and denotes projection onto the nonnegative orthant to maintain dual feasibility [32]. In decentralized settings with limited communication, each agent may maintain a local copy of the dual variables

Table 8. Group-wise effects of learned discount policies

Segment	Avg Discount	Recovery Rate	Escalation Rate
Overall	8.1%	0.54	2.3%
Low Risk	6.3%	0.62	1.1%
Medium Risk	8.7%	0.55	2.0%
High Risk	10.9%	0.47	4.6%
Recent Delinquency	9.4%	0.51	3.1%

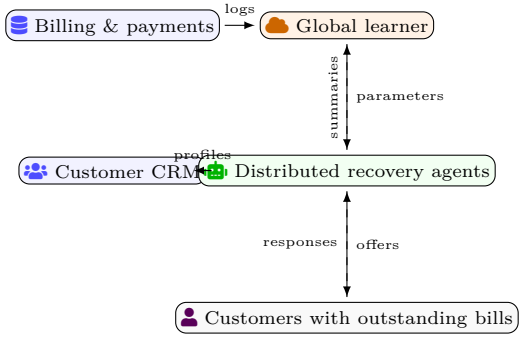

Figure 1. System-level architecture for distributed learning of pricing and discount policies: heterogeneous operational data feed a global learner that coordinates multiple revenue recovery agents interacting with delinquent customers.

and update it based on locally observed residuals and exchanged information with neighboring agents. Consensus schemes can be incorporated to ensure that these local copies remain close to one another over time.

In addition to dual-based coordination, agents may also employ direct consensus on value-function or policy parameters [33]. Suppose the agents are connected by a communication graph in which edges represent neighbor relationships. A consensus-based update for a parameter vector can be expressed as

$$\theta_i^{t+1} = \sum_j w_{ij}\theta_j^t - \alpha_t g_i^t,$$

where the coefficients form a row-stochastic mixing matrix reflecting the communication topology, and denotes a local gradient or temporal-difference direction. The first term averages parameter estimates among neighbors, gradually aligning them, while the second term incorporates local learning signals [34]. This mechanism allows agents to share information about pricing and discount policies and their value implications without central coordination.

Overall, the distributed learning architecture combines local temporal-difference evaluation, actor updates for pricing and discount parameters, dual-based handling of shared constraints, and consensus mechanisms that propagate information across the network. The interplay between these components gives rise to a set of coupled stochastic recursions that define the evolution of policy, value, and dual variables. The next section shows how these recursions can be understood as approximations to linear systems and optimization problems, which provides insight into their convergence properties and the structure of the learned policies [35].

4. Linear Approximation and Optimization Structure

The recursive learning rules can be studied by examining their fixed points and approximate deterministic limits. Under standard stochastic approximation assumptions, such as diminishing step sizes and sufficient exploration, the temporal-difference and policy-gradient updates track ordinary differential equations whose equilibria correspond to solutions of linear equations and linear inequalities derived from Bellman relations and constraints. This section outlines the linear structure underlying these equilibria and describes how it can be exploited for analysis and algorithm design [36].

For a fixed policy profile, the value-function parameters for each agent aim to approximate the true value function within the span of the feature vectors. The temporal-difference update can be interpreted as a stochastic approximation to the solution of a projected Bellman equation. In linear algebra terms, the limiting parameter vector satisfies a system

$$A_i\theta_i = [37]b_i,$$

where the matrix and vector depend on the transition dynamics under the policy, the feature map, and the reward function. Elements of are expectations of products of feature differences weighted by transition probabilities, while entries of reflect expected discounted rewards times features. When the feature covariance matrix is

Figure 2. Local control loop for a single revenue recovery agent: states are encoded, mapped to pricing and discount actions, and used to update the policy from observed revenue and risk-adjusted feedback.

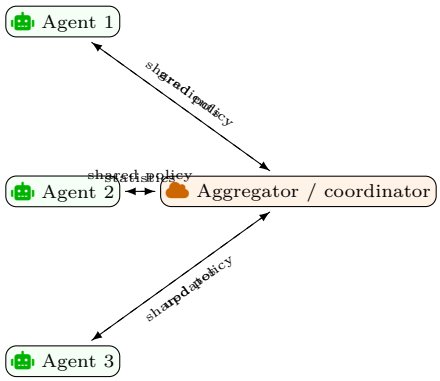

Figure 3. Distributed learning scheme: agents independently interact with customers while periodically exchanging compressed statistics and receiving shared pricing and discount policies from a lightweight coordinator.

nonsingular and the projected Bellman operator is a contraction, such a system has a unique solution.

Stacking the value-function parameters across agents yields a global linear system [38]

$$A\theta = b,$$

where is block-diagonal with blocks, and concatenates the vectors. Coupling among agents introduced through dual variables and augmented rewards enters the right-hand side via the reward components [39]. In the presence of consensus steps on, additional off-diagonal terms can appear in, reflecting the mixing of parameters. Under appropriate conditions on the mixing matrix, these off-diagonal components preserve the solvability and stability of the system, and the overall structure remains amenable to linear analysis.

Policy optimization under long-run average or discounted criteria can also be cast as a linear program over occupancy measures. For each agent and joint state-action configuration, an occupancy variable represents the expected discounted number of visits to that configuration under a policy [40]. The global objective of maximizing discounted recovered revenue, subject to linear constraints on discounted concessions and operational quantities, can then be written as the minimization or maximization of a linear function of these occupancy variables. For example, in a simplified discounted cost formulation one might write

$$\min_x c^\top x$$

subject to linear constraints that ensure consistency with the transition dynamics and constraints. The consistency constraints equate the discounted inflow and outflow of probability mass for each state, while the global constraints on concessions and exposures become linear inequalities in the occupancy variables [41].

When policies are parameterized and learning is incremental, the occupancy measures are not directly optimized; instead, the linear program serves as a conceptual representation of the fixed points toward which learning converges. The dual of this linear program yields value-function-like variables and Lagrange multipliers that re-

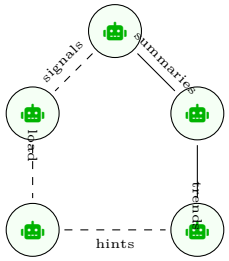

Figure 4. Interaction graph among revenue recovery agents: sparse peer-to-peer communication supports sharing of local load, behavioral trends, and recovery hints without centralized control.

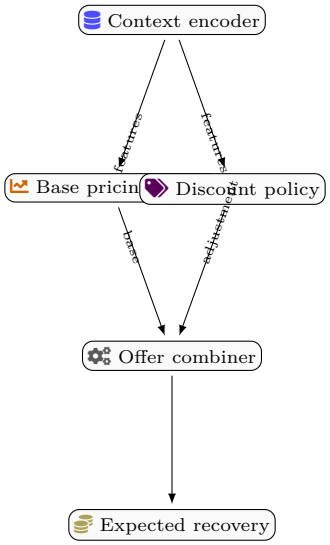

Figure 5. Decomposition of the learned control into a base price policy and a discount policy driven by shared context, jointly determining offers that maximize expected revenue recovery.

semble the dual variables used in the distributed learning architecture. In particular, the dual variables associated with the flow constraints correspond to value functions, while those associated with the global constraints correspond to dual prices on concessions or risk exposures [42]. This dual structure reinforces the idea that value-function estimation and dual-variable updates in the learning algorithm are approximating a primal-dual solution of an underlying linear program.

Distributed implementation of the linear program can be accomplished by decomposing the global problem into local subproblems coupled by shared constraints. Each agent controls local occupancy variables corresponding to its subset of states and actions. The local objective is a linear function of these variables, and the local constraints enforce consistency with agent-specific transition dynamics [43]. The coupling constraints across agents enter as linear combinations of local occupancy variables equated to global limits. A primal-dual decomposition then yields local linear programs augmented by dual variables that are updated based on global constraint residuals. This is consistent with the dual-based learning procedures described earlier, in which agents adjust their effective rewards using shared dual parameters.

The consensus updates on value-function and policy parameters can be interpreted as distributed methods for solving the global linear system for or approximating solutions to the dual or primal variables of the linear program [44]. For instance, an averaging scheme with gradient corrections resembles distributed gradient descent on a quadratic objective of the form

$$\min_{\theta} \frac{1}{2}\theta^{\top} H \theta - h^{\top}\theta,$$

where is a symmetric positive semidefinite matrix derived from and the communication topology, and is a vector formed from the reward-related terms. The presence of mixed value and policy parameters, together with dual variables, transforms this objective into a more general convex-concave saddle-point problem, but the core linear structure is preserved [45]. This connection enables the application of tools from distributed convex optimization to analyze convergence rates and error bounds for the learning algorithm.

In summary, linear function approximation, linearized Bellman equations, and linear programming formulations provide a common structural backbone for the distributed

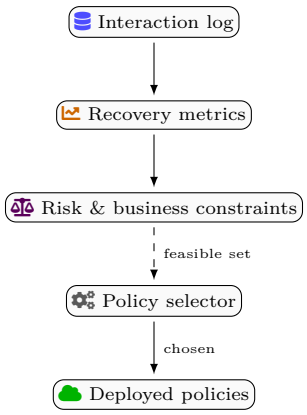

Figure 6. Offline evaluation and selection pipeline: interaction logs are summarized into recovery metrics, filtered through risk and business constraints, and used to choose deployable pricing and discount policies.

learning of pricing and discount policies. Values, policies, and dual variables are all related through systems of linear equations and inequalities whose solutions define candidate equilibrium behaviors for interacting revenue recovery agents. The next section uses this structure to study convergence and performance properties of the distributed learning dynamics [46].

5. Convergence and Performance Analysis

To assess the behavior of the distributed learning algorithms, it is important to analyze the stability and convergence of the coupled stochastic recursions governing value, policy, and dual variables. The analysis is conducted under assumptions that are typical in stochastic approximation and distributed optimization, adapted to the revenue recovery context. These assumptions include properties of the communication graph, conditions on step sizes, feature richness, and persistence of excitation in the observed trajectories.

Consider first the temporal-difference updates for value-function parameters with consensus steps [47]. Under a fixed policy and dual vector, the local temporal-difference recursion for each agent can be expressed as

$$\theta_i^{t+1} = \sum_j w_{ij}\theta_j^t + \alpha_t h_i^t,$$

where the term arises from the temporal-difference error and feature vector, and the mixing matrix reflects communication. Standard results for consensus-based stochastic approximation state that if the communication graph is connected, the mixing matrix is row-stochastic with a positive diagonal, and the step sizes satisfy usual summability and square-summability conditions, then the parameter iterates converge almost surely to a common limit [48]. This limit solves an averaged projected Bellman equation across agents, provided that the noise in the temporal-difference errors satisfies appropriate martingale difference conditions and that the projected Bellman operator is a contraction in the chosen norm.

In the presence of dual variables influencing rewards, the effective reward function and thus the value-function parameters depend on the evolving dual vector. If the dual variables are updated on a slower time scale than the value parameters, a two-time-scale stochastic approximation argument can be invoked. On the faster time scale, the value parameters track the solution of the projected Bellman equation for a quasi-static dual vector [49]. On the slower time scale, the dual variables observe approximate constraint residuals computed under the current value and policy estimates, and update accordingly. This structure allows one to show that the joint process of and converges to a set of stationary points of an associated ordinary differential equation whose equilibria satisfy complementary slackness and primal feasibility conditions for the underlying linear program.

The actor updates for policy parameters add another layer to this hierarchy. If policy parameters evolve on an even slower time scale than the dual variables, then they see value and dual variables that have essentially equilibrated to their quasi-static limits [50]. Under this assumption, policy updates approximate gradient ascent or descent on a performance function parameterized by the policy. Stochastic approximation theory on three time scales indicates that under regularity conditions, the combined process converges to a set of stationary points of a limiting system in which policies are locally optimal given the value-function approximation, dual variables enforce global constraints, and value functions solve projected Bellman equations for the limiting policies and dual variables.

Performance guarantees derived from this analysis are typically approximate due to function approximation and limited exploration [51]. However, the linear structure of the approximating equations yields interpretable error bounds. For instance, if the feature vectors span the space of true value functions across agents under the set of admissible policies, then the projected Bellman fixed points coincide with the true values, and the

learning algorithm can in principle converge to policies that are near-optimal for the underlying constrained decision problem. When the feature class is limited, the convergence points correspond to best approximations within the span of the features, and the performance loss relative to the optimal unconstrained policy can be bounded in terms of approximation error measures.

The communication topology also influences convergence speed and performance [52]. A denser communication graph and more frequent consensus steps generally accelerate the alignment of value and policy parameters across agents, which may reduce variability and improve constraint satisfaction at the portfolio level. Sparse graphs or low communication rates can slow down convergence and create persistent heterogeneity in policies and estimated values. Nevertheless, provided the graph remains connected over time and the mixing weights satisfy standard conditions, the asymptotic consensus properties continue to hold. The trade-off between communication overhead and coordination quality is particularly pertinent in revenue recovery operations where privacy or organizational boundaries restrict information sharing [53].

Another important aspect of performance analysis concerns constraint satisfaction. Because the dual variables are updated based on observed residuals computed from sample paths, constraint satisfaction can exhibit transient violations even if the limiting dual variables enforce feasibility. The magnitude and duration of these violations depend on the variability of the observed trajectories, step sizes, and the strength of the feedback induced by dual updates [54]. In practice, one may introduce conservative margins in the constraints or adjust step-size schedules to balance responsiveness against variability. The linear nature of the constraints and dual updates facilitates the derivation of probabilistic bounds on cumulative constraint violations over finite horizons, under assumptions about the boundedness of feature vectors and rewards.

Finally, the learning dynamics must remain stable under the continuous evolution of external conditions such as macroeconomic trends or regulatory changes. When environment dynamics change slowly relative to the adaptation rate of the learning algorithm, the system can track a moving target, with parameters staying close to the instantaneous equilibria associated with the current environment [55]. Linearization of the dynamics around these equilibria and analysis of the corresponding Jacobian matrices provide insights into the tracking behavior and robustness of the algorithm. The linear structure again simplifies this analysis, as stability reduces to spectral properties of matrices formed from the communication topology, feature covariance, and step-size parameters.

6. Experimental Evaluation with Stylized Revenue Recovery Scenarios

To illustrate the behavior of the distributed learning architecture, one can consider stylized experimental settings that capture essential features of revenue recovery while remaining analytically and computationally manageable. These scenarios involve synthetic customer populations, simplified state descriptions, and parameterized models of payment responses to pricing and discount offers [56]. While they do not claim to represent any specific operational environment, they serve to highlight qualitative effects of distributed learning, policy heterogeneity, and communication constraints.

A basic scenario may involve several agents each managing a segment of accounts characterized by different risk levels and responsiveness to discounts. Customer states can be represented by a small set of variables such as delinquency age, outstanding balance bucket, and a discrete behavior score [57]. Actions comprise a limited menu of discount levels and fee-waiver options. The transition dynamics specify probabilities of customer payment, continued delinquency, or default as functions of the current state and chosen action. These probabilities can be chosen to reflect plausible qualitative patterns, such as higher discounts increasing the likelihood of partial or full payment for some segments while having diminishing returns for others.

Agents start with randomly initialized policy and value parameters and interact with their synthetic customers over many periods [58]. Each agent observes local states, selects actions according to its current policy, collects rewards based on recovered payments and concession costs, and updates its parameters using the distributed learning rules. Simulation experiments can vary the communication topology among agents, the presence or absence of shared constraint enforcement through dual variables, and the parameterization of policies and value functions. For instance, policies may be modeled as softmax functions over action preferences that are linear in state features, while value functions use similar linear features.

One design of experiments explores the impact of dual-based coordination enforcing a global discount budget [59]. Without dual variables, agents learn policies solely based on local reward signals, which may lead them to offer high discounts whenever they appear locally profitable, regardless of the aggregate budget. The simulated trajectories may show that global concession levels drift above desired limits, even if local rewards improve. Introducing dual variables that penalize discounted concession statistics at the portfolio level modifies the reward signals [60]. Over time, the dual variables adapt to increase the effective cost of concessions when the budget is exceeded, leading agents to adjust

their policies toward offering fewer or smaller discounts. Simulation traces can display the evolution of dual variables, average discount levels, and recovered revenue for different segments, illustrating the balancing effect of the global constraint.

Another set of experiments studies the effect of communication sparsity on policy convergence. In a fully connected communication graph, consensus steps quickly align value and policy parameters, resulting in relatively homogeneous policies across agents facing similar customer segments [61]. In contrast, when the communication graph is a sparse structure, such as a ring or a collection of loosely coupled clusters, agents within the same cluster may converge to similar policies while remaining different from agents in other clusters. This heterogeneity can be beneficial when clusters correspond to different customer populations, as it allows for specialization of policies. However, it can also complicate the enforcement of global constraints and the interpretation of portfolio-level metrics. Simulation results can compare convergence rates and performance metrics across different communication graphs, providing insight into how network structure influences distributed revenue recovery learning [62].

A further dimension of evaluation involves the choice of feature representations for value functions and policies. Richer feature sets can capture more nuanced interactions between customer states and the profitability of different pricing and discount actions, potentially leading to improved policies. However, they also increase the dimensionality of parameter vectors and may slow convergence or exacerbate overfitting to idiosyncratic patterns in simulated data [63]. Experiments can compare linear feature sets of varying complexity, examining the trade-offs between model expressiveness, convergence behavior, and robustness to noise in the simulated environment. Since the learning algorithms operate within a linear approximation architecture, these experiments directly test the implications of the linear modeling assumptions on policy quality.

Finally, sensitivity analyses can be conducted by varying macro-level parameters such as the overall delinquency rate, the responsiveness of customers to concessions, and the cost structure associated with operational efforts. By adjusting these parameters, one can simulate changes in economic conditions or regulatory regimes and observe how the distributed learning system adapts [64]. For example, a scenario in which customer responsiveness to discounts decreases might prompt the algorithm to reduce discount levels and emphasize higher prices, as large concessions no longer yield proportional increases in recovered revenue. Another scenario where regulators impose stricter caps on aggressive collection practices can be modeled as tightening linear constraints, leading to adjustments in dual variables and a shift in policies toward less intensive strategies.

Through such stylized experiments, one can gain intuition about the qualitative dynamics of distributed pricing and discount policy learning in multi-agent revenue recovery systems. While these synthetic settings are not substitutes for real-world evaluations, they complement the analytical results by illustrating how linear modeling structures, consensus updates, and dual-based constraint enforcement interact in concrete scenarios [65].

7. Conclusion

Distributed learning of pricing and discount policies from interacting revenue recovery agents involves combining sequential decision modeling, linear approximation, and multi-agent optimization under constraints. In environments where agents manage different segments of a portfolio, face heterogeneous customer responses, and operate under shared budgets and regulations, centralized policy design can be difficult to implement. The framework considered here models each agent as learning from its own interaction history while exchanging limited information in the form of aggregate statistics or parameter estimates [66]. Linear function approximation and linearly constrained formulations provide a tractable way to capture long-run revenue and concession trade-offs, as well as the coupling induced by shared constraints.

By interpreting temporal-difference learning, policy-gradient updates, dual-variable adjustments, and consensus mechanisms as components of a single stochastic approximation system, one can analyze their joint behavior using linear systems and linear programming structures. The fixed points of this system correspond to approximate solutions of a constrained decision problem in which agents choose pricing and discount policies that balance local recovery performance against portfolio-level constraints. Stochastic approximation theory on multiple time scales offers conditions under which the learning dynamics are stable and converge to stationary points of the associated deterministic limit [67]. The linear nature of the value-function approximations and constraints simplifies the derivation of error bounds and facilitates reasoning about the impact of communication topology and feature representation.

Stylized simulation experiments on synthetic revenue recovery scenarios help elucidate how the distributed learning architecture behaves under different communication structures, constraint regimes, and feature choices. These simulations indicate that dual-based coordination can help enforce global discount budgets, that communication sparsity influences convergence speed and policy heterogeneity, and that feature richness affects both policy quality and convergence behavior. Although such experiments are simplified and do not capture all complexities of operational environments, they highlight key

qualitative trade-offs associated with distributed learning of pricing and discount policies [68].

The analysis presented here focuses on linear approximation and relatively simple models of customer behavior and interactions among agents. Extensions could consider richer models of partial observability, risk-sensitive objectives, or strategic responses from customers who anticipate future concessions. Further work could also examine privacy-preserving mechanisms for sharing information across agents, as well as adaptive communication strategies that allocate bandwidth and coordination effort where they are most beneficial. Overall, the study underscores that distributed learning with linear modeling offers a structured approach to aligning local revenue recovery decisions with portfolio-level objectives and constraints in multi-agent environments [69].

References

[1] E. Garcia, A. Giret, and V. Botti, ISD - Regulated Open Multi-Agent Systems Based on Contracts. Springer New York, 9 2011.

[2] Y. Zhang, Q. Yang, and W. Yan, "Network-based leader–following consensus for second-order multi-agent systems with nonlinear dynamics:," Transactions of the Institute of Measurement and Control, vol. 38, pp. 1165–1173, 7 2016.

[3] Y. Xu and Y.-P. Tian, "Design of a class of nonlinear consensus protocols for multi-agent systems," International Journal of Robust and Nonlinear Control, vol. 23, pp. 1524–1536, 5 2012.

[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] C. Kiourt, D. Kalles, and G. Pavlidis, EUMAS/AT - Human Rating Methods on Multi-agent Systems. Germany: Springer International Publishing, 4 2016.

[6] X.-L. Xie, S.-B. Huang, and X. juan Guo, "Collaborative plotting system based on web and multi-agent," International Journal of u- and e- Service, Science and Technology, vol. 9, pp. 157–164, 7 2016.

[7] K. S. HEGDE, "Recovering lost revenue by augmenting internal customer data with external data for accurate invoicing for large b2b enterprises," INTERNATIONAL JOURNAL, vol. 13, no. 11, pp. 613–615, 2024.

[8] Y. Su and J. Huang, "Cooperative output regulation with application to multi-agent consensus under switching network," IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, vol. 42, pp. 864–875, 1 2012.

[9] L. Cheng-Lin and L. Fei, "Dynamical consensus algorithm for second-order multi-agent systems subjected to communication delay," Communications in Theoretical Physics, vol. 59, pp. 773–781, 6 2013.

[10] A. Boubetra, N. Mouhoub, and N. E. H. Fares, Towards a Multi-Agents Simulation Meta-Model for Manufacturing Systems, pp. 913–920. ASME Press, 1 2009.

[11] H. yong Yang, G. deng Zong, and S. Zhang, "Movement consensus of delayed multi-agent systems with directed weighted networks," International Journal of Intelligent Computing and Cybernetics, vol. 4, pp. 265–277, 6 2011.

[12] K. Pireva and P. Kefalas, "The use of multi agent systems in cloud e-learning," 1 2016.

[13] T. N. Wong and F. Fang, "A multi-agent protocol for multilateral negotiations in supply chain management," International Journal of Production Research, vol. 48, pp. 271–299, 10 2008.

[14] M. Selecký and T. Meiser, "Integration of autonomous uavs into multi-agent simulation," Acta Polytechnica, vol. 52, 1 2012.

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

[16] J. L. Meliá, "A multi-agent safety response model in the construction industry.," Work (Reading, Mass.), vol. 51, pp. 549–556, 7 2015.

[17] G. Ren and Y. Yu, "Consensus of fractional multi-agent systems using distributed adaptive protocols," Asian Journal of Control, vol. 19, pp. 2076–2084, 8 2017.

[18] Y. Hu, P. Li, and J. Lam, "On the synthesis of h∞ consensus for multi-agent systems," IMA Journal of Mathematical Control and Information, vol. 32, pp. 591–607, 3 2014.

[19] S. Azaiez, M.-P. Huget, and F. Oquendo, "An approach for multi-agent metamodelling," Multiagent and Grid Systems, vol. 2, pp. 435–454, 12 2006.

[20] L. Gao, Y. Cui, W. Chen, and W. Chen, "Leader-following consensus for discrete-time descriptor multi-agent systems with observer-based protocols:," Transactions of the Institute of Measurement and Control, vol. 38, pp. 1353–1364, 7 2016.

[21] R. Beheshti, R. Barmaki, and N. Mozayani, ANAC@AAMAS - Negotiations in Holonic Multi-agent Systems. Germany: Springer International Publishing, 3 2016.

[22] M. Winikoff, "Challenges and directions for engineering multi-agent systems," 1 2012.

[23] W. Li, Z. Chen, and Z. Liu, "Output regulation distributed formation control for nonlinear multi-agent systems," Nonlinear Dynamics, vol. 78, pp. 1339–1348, 7 2014.

[24] X.-J. Zhang, B. Cui, and K. Lou, "Leaderless and leader-following consensus of linear multi-agent systems," Chinese Physics B, vol. 23, pp. 110205–, 11 2014.

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

[26] F. Wang and H. Yang, Dynamical Flocking of Multi-agent Systems with Multiple Leaders and Uncertain Parameters, pp. 13–20. Germany: Springer Singapore, 9 2016.

[27] K. Miyazaki and S. Kobayashi, "On the rationality of profit sharing in multi-agent reinforcement learning," in Proceedings Fourth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2001, pp. 421–425, IEEE, 11 2002.

[28] Y. L. Xu and L. L. Wang, "The application of multi-agent technology in the ascm model," Applied Mechanics and Materials, vol. 55-57, pp. 2080–2085, 5 2011.

[29] H. Igarashi, Y. Adachi, and K. Takahashi, "Adaptive cooperation for multi agent systems based on human social behavior," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 16, pp. 139–146, 1 2012.

[30] S. S. Wan, D. Wang, and Q. Cao, "Multi-agent based modeling simulation about vanet," Advanced Materials Research, vol. 760-762, pp. 680–684, 9 2013.

[31] Y. Jung, J. Lee, and M. Kim, "Aamas - multi-agent based community computing system development with the model driven architecture," in Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pp. 1329–1331, ACM, 5 2006.

[32] B. Dunin-Keplicz and R. Verbrugge, Teamwork in Multi-Agent Systems: A Formal Approach - Teamwork in Multi-Agent Systems: A Formal Approach. Wiley, 5 2010.

[33] M. Yang, D. Tang, H. Ding, W. Wang, T. Luo, and S. Luo, "Evaluating staggered working hours using a multi-agent-based q-learning model," TRANSPORT, vol. 29, pp. 296–306, 9 2014.

[34] Z. Yan and R. Wang, Consensus of Heterogeneous Multi-agent Systems Based on Event-Triggered, pp. 385–393. Germany: Springer Singapore, 10 2017.

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

[36] F. Xiao and L. Wang, "Consensus problems for high-dimensional multi-agent systems," IET Control Theory & Applications, vol. 1, pp. 830–837, 5 2007.

[37] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, and Z. Wang, "Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring," Expert Systems with Applications, vol. 37, pp. 2028–2036, 3 2010.

[38] R. Cimler, M. Husáková, and M. Kolackova, ICCCI (2) - Exploration of Autoimmune Diseases Using Multi-agent Systems, pp. 282–291. Germany: Springer International Publishing, 9 2016.

[39] N. O. Garanina, E. A. Sidorova, and E. V. Bodin, Ershov Memorial Conference - A Multi-agent Text Analysis Based on Ontology of Subject Domain, pp. 102–110. Germany: Springer Berlin Heidelberg, 4 2015.

[40] Z. Li, "Distributed robust consensus of linear multi-agent systems with switching topologies," The Journal of Engineering, vol. 2015, pp. 17–24, 1 2015.

[41] Y. S. Choi and S. I. Yoo, IDA - Multi-agent Web Information Retrieval: Neural Network Based Approach, pp. 499–512. Germany: Springer Berlin Heidelberg, 7 1999.

[42] F. A. C. A. Gonçalves, F. G. Guimarães, and M. J. F. Souza, "Gecco - an evolutionary multi-agent system for database query optimization," in Proceedings of the 15th annual conference on Genetic and evolutionary computation, pp. 535–542, ACM, 7 2013.

[43] Y. Feng and X. Tu, "Consensus analysis for a class of mixed-order multi-agent systems with nonlinear consensus protocols:," Transactions of the Institute of Measurement and Control, vol. 37, pp. 147–153, 6 2014.

[44] J. Coble, L. Roszman, and T. Frazier, "Dynamic control and formal models of multi-agent interactions and behaviors," 2 2003.

[45] 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.

[46] W. Alrawagfeh, E. Brown, and M. Mata-Montero, Norms of Behaviour and Their Identification and Verification in Open Multi-Agent Societies, pp. 129–145. IGI Global, 4 2012.

[47] H. Hu and Z. Lin, "Consensus of a class of discrete-time nonlinear multi-agent systems in the presence of communication delays.," ISA transactions, vol. 71, pp. 10–20, 2 2017.

[48] K. Zhang, Y. Maeda, and Y. Takahashi, "Group behavior learning in multi-agent systems based on social interaction among agents," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 15, pp. 896–903, 9 2011.

[49] W. L. Kang, X. M. Liu, and K. Wang, "A multi-agent management system for grid resource allocation," Advanced Materials Research, vol. 760-762, pp. 973–976, 9 2013.

[50] A. S. Gazafroudi, J. F. D. Paz, F. Prieto-Castrillo, G. Villarrubia, S. Talari, M. Shafie-khah, and J. P. S. Catalao, ISAmI - A Review of Multi-agent Based Energy Management Systems, pp. 203–209. Springer International Publishing, 6 2017.

[51] S. M. Avakaw, A. Doudkin, A. Inyutin, A. V. Otwagin, and V. A. Rusetsky, "Multi-agent parallel implementation of photomask simulation in photolithography," International Journal of Computing, vol. 11, pp. 45–54, 8 2014.

[52] R. R. Yager, "Multi-agent negotiation using linguistically expressed mediation rules," Group Decision and Negotiation, vol. 16, pp. 1–23, 6 2006.

[53] Y. Cheng and H. Yu, "Adaptive group consensus of multi-agent networks via pinning control," International Journal of Pattern Recognition and Artificial Intelligence, vol. 30, pp. 1659014–, 4 2016.

[54] P. Mathieu, J.-C. Routier, and Y. Secq, PRIMA - Principles for Dynamic Multi-agent Organizations, vol. 2413, pp. 109–122. Germany: Springer Berlin Heidelberg, 7 2002.

[55] 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.

[56] Y. Zheng, W. T. Li, L. Zeng, Y. Ge, X. Y. Cai, and X. Y. Meng, "Realization on the interactive remote video conference system based on multi-agent," MATEC Web of Conferences, vol. 63, pp. 04011–, 7 2016.

[57] L. Tong, "Research on path-planning of manipulator based on multi-agent reinforcement learning," Applied Mechanics and Materials, vol. 44-47, pp. 2116–2120, 12 2010.

[58] H. Bagheri, M. A. Torkamani, and Z. Ghaffari, "Multi-agent approach for facing challenges in ultra-large scale systems," International Journal of Electrical and Computer Engineering (IJECE), vol. 4, pp. 151–154, 4 2014.

[59] W. Zhang and Y. Liu, "Observer-based distributed consensus for general nonlinear multi-agent systems with interval control inputs," International Journal of Control, vol. 89, pp. 84–98, 7 2015.

[60] E. Majd, V. Balakrishnan, S. Gharib, K. Pourmahdi, L. Shakeri, and M. F. Abadi, "Calculating the reliability and reputation of agents in e-commerce multi-agent environments," Applied Mechanics and Materials, vol. 548-549, pp. 1478–1482, 4 2014.

[61] I. M. Ross and C. D'Souza, "Rapid trajectory optimization of multi-agent hybrid systems," in AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, 6 2004.

[62] C.-N. Bodea, R.-I. Mogos, and I. R. Badea, A Multi-Agent System for Acquiring Transport Services, pp. 105–129. Springer New York, 4 2013.

[63] H. C. Inyiama, I. Obiora-Dimson, and C. C. Okezie, "Designing multi-agent based linked state machine," International Journal of Research in Engineering and Technology, vol. 02, pp. 90–106, 7 2013.

[64] R. Verbrugge and B. Dunin-Keplicz, "Formal approaches to multi-agent sysems," Autonomous Agents and Multi-Agent Systems, vol. 19, pp. 1–3, 11 2008.

[65] W. Leong and M. Liu, "Sac - a multi-agent algorithm for vehicle routing problem with time window," in Proceedings of the 2006 ACM symposium on Applied computing, pp. 106–111, ACM, 4 2006.

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

[67] V. Mafik and M. Pechoucek, EASSS - Social knowledge in multi-agent systems, vol. 2, pp. 211–245. Germany: Springer Berlin Heidelberg, 6 2001.

[68] Z. Zhao, S. An, and X. Wang, "Congestion pricing revenue redistribution simulation based on multi-agent," in International Conference on Transporta-

tion Engineering 2009, pp. 3213–3218, American Society of Civil Engineers, 7 2009.

[69] V. Chiriacescu, L.-K. Soh, and D. F. Shell, "Understanding human learning using a multi-agent simulation of the unified learning model," International Journal of Cognitive Informatics and Natural Intelligence, vol. 7, pp. 1–25, 10 2013.