Discovering Nash Equilibria with LLM Agents: A Payment Systems Case Study

Hugi Aegisberg

December 21, 2025

DO NOT CIRCULATE

This is a working document and is not intended for distribution.

This paper and the accompanying SimCash codebase were developed in collaboration with **Claude 4.5 Opus** (Anthropic), which served as a co-author for both the research code and this manuscript.

About This Document

This is a **research proposal** presenting methodology and preliminary findings to potential collaborators. All tables, figures, and statistics are programmatically generated from experiment databases (DuckDB → DataProvider → LaTeX/charts), eliminating manual transcription. The accompanying text is written by an AI assistant (Claude) following author guidance on structure and conclusions.

SimCash is a hybrid Rust/Python payment system simulator with deterministic replay, configurable policies, and multiple settlement mechanisms (RTGS, queues, LSM). The experiment runner uses LLM agents to iteratively optimize policies through natural language reasoning, enabling research into multi-agent coordination in financial infrastructure.

Abstract

Can Large Language Models discover Nash equilibria through strategic reasoning alone? We explore this question using payment system liquidity management—a domain where banks must balance the cost of holding reserves against settlement delays, and where game-theoretic equilibria are well-characterized but difficult to find without explicit modeling.

We present SimCash, a framework where LLM agents optimize liquidity policies through natural language deliberation under information isolation: each agent observes only its own costs and transaction history, never counterparty strategies. Through 9 independent runs across 3 scenarios adapted from Castro et al., agents reliably converge to stable equilibria (100% success, mean 12.4 iterations). However, equilibrium selection exhibits path-dependence: in symmetric games, agents consistently converge to *asymmetric* free-rider equilibria, with the identity of the free-rider determined by early exploration rather than cost structure.

These preliminary findings suggest that LLM-based policy optimization can discover equilibrium behavior without explicit game-theoretic modeling, while also revealing that sequential best-response dynamics in multi-agent LLM systems may systematically favor asymmetric outcomes. Our small sample (9 runs) requires validation through expanded experimentation before drawing strong conclusions.

1 Introduction

Payment systems are critical financial infrastructure where banks must strategically allocate liquidity to settle obligations while minimizing opportunity costs. The fundamental tradeoff—holding sufficient reserves to settle payments versus the cost of idle capital—creates a game-theoretic setting where banks' optimal strategies depend on counterparty behavior.

Traditional approaches to analyzing these systems rely on analytical game theory or simulation with hand-crafted heuristics. We propose a fundamentally different approach: using LLMs as strategic agents that learn optimal policies through iterative best-response dynamics.

1.1 Contributions

1. **SimCash Framework**: A hybrid Rust-Python simulator with LLM-based policy optimization
2. **Empirical Validation**: Successful recovery of Castro et al.'s theoretical equilibria
3. **Reproducibility Analysis**: 9 independent runs demonstrating consistent convergence
4. **Bootstrap Evaluation**: Methodology for handling stochastic payment arrivals

2 The SimCash Framework

2.1 Simulation Engine

SimCash uses a discrete-time simulation where:

- Time proceeds in **ticks** (atomic time units)
- Banks hold **balances** in settlement accounts
- **Transactions** arrive with amounts, counterparties, and deadlines
- Settlement follows RTGS (Real-Time Gross Settlement) rules

2.2 Cost Function

Agent costs comprise:

- **Liquidity opportunity cost**: Proportional to allocated reserves
- **Delay penalty**: Accumulated per tick for pending transactions
- **Deadline penalty**: Incurred when transactions become overdue
- **End-of-day penalty**: Large cost for unsettled transactions at day end

2.3 LLM Policy Optimization

The key innovation is using LLMs to propose policy parameters. At each iteration:

1. **Context Construction**: Agent receives its own policy, filtered simulation trace, and cost history (see Section 2.4)
2. **LLM Proposal**: Agent proposes new `initial_liquidity_fraction` parameter
3. **Evaluation**: Run simulation(s) with proposed policy
4. **Update**: Apply mode-specific acceptance rule (see below)
5. **Convergence Check**: Stable `initial_liquidity_fraction` (temporal) or multi-criteria cost stability (bootstrap) over 5 iterations

2.4 Optimization Prompt Anatomy

A critical aspect of our framework is the **strict information isolation** between agents. Each agent receives a two-part prompt with no access to counterparty information.

2.4.1 System Prompt (Shared)

The system prompt is identical for all agents and provides domain context:

- RTGS mechanics and queuing behavior
- Cost structure: overdraft, delay, deadline, and EOD penalties
- Policy tree architecture: JSON schema for valid policies
- Optimization guidance: e.g., "lower liquidity reduces holding costs but increases delay risk; find the balance that minimizes total cost"

2.4.2 User Prompt (Agent-Specific)

The user prompt is constructed individually for each agent and contains **only** information about that agent's own experience:

1. **Performance metrics from past iterations**: Agent's own mean cost, standard deviation, settlement rate
2. **Current policy**: Agent's own `initial_liquidity_fraction` parameter
3. **Cost breakdown**: Agent's own costs by type (delay, overdraft, penalties)
4. **Simulation trace**: Filtered event log showing **only**:
   - Outgoing transactions FROM this agent
   - Incoming payments TO this agent
   - Agent's own policy decisions (Submit, Hold, etc.)
   - Agent's own balance changes (for settlements it initiated)
5. **Iteration history**: Agent's own cost trajectory across iterations

2.4.3 Information Isolation

The prompt explicitly excludes all counterparty information:

- **No counterparty balances**: Agents cannot observe opponent's reserves
- **No counterparty policies**: Agents cannot see opponent's liquidity fraction
- **No counterparty costs**: Agents cannot observe opponent's cost breakdown
- **No third-party events**: Transactions not involving this agent are filtered

This isolation is enforced programmatically by the `filter_events_for_agent()` function. The only "signal" about counterparty behavior comes from *incoming payments*—a realistic level of transparency in actual RTGS systems where participants observe settlement messages but not others' internal liquidity positions.

Crucially, agents receive **transaction events from the current iteration** alongside **performance metrics from past iterations**, but are never informed that the environment is stationary. The agent is not told that all iterations use identical transaction schedules (Experiments 1 and 3) or identical stochastic parameters (Experiment 2). From the agent's perspective, each iteration could involve a different payment environment—any regularity must be inferred from observed patterns rather than assumed from explicit knowledge of the experimental design.

2.5 Evaluation Modes

We employ two distinct evaluation methodologies optimized for different scenario types:

2.5.1 Deterministic-Temporal Mode (Experiments 1 & 3)

For scenarios with fixed payment schedules, we use **temporal policy stability** to identify Nash equilibria:

- **Single simulation** per iteration with deterministic arrivals
- **Unconditional acceptance**: All LLM-proposed policies are accepted immediately
- **Rationale**: Cost-based rejection would cause oscillation in multi-agent settings where counterparty policies evolve simultaneously
- **Convergence criterion**: Both agents' `initial_liquidity_fraction` unchanged for 5 consecutive iterations, indicating mutual best-response equilibrium

2.5.2 Bootstrap Mode (Experiment 2)

For stochastic scenarios, we use **bootstrap resampling** for robust policy evaluation:

- **Initial collection**: Run one simulation to collect transaction history
- **Resampling**: Generate 50 transaction schedules by resampling with replacement, preserving settlement offset distributions
- **Paired comparison**: Evaluate both old and new policy on each sample, computing $\delta_i = \text{cost}_{\text{old},i} - \text{cost}_{\text{new},i}$

- **Acceptance criterion**: Accept if $\sum_i \delta_i > 0$ (improvement across all samples)
- **Convergence criterion**: Three criteria must ALL be satisfied over a 5-iteration window:
  1. Coefficient of variation below 3% (cost stability)
  2. Mann-Kendall test $p > 0.05$ (no significant trend, i.e., not still improving)
  3. Regret below 10% (current cost within 10% of best observed)

2.6 Experimental Setup

We implement three canonical scenarios from Castro et al. (2025):

Experiment 1: 2-Period Deterministic (Deterministic-Temporal Mode)

- 2 ticks per day
- Asymmetric payment demands: $P^A = [0, 0.15]$, $P^B = [0.15, 0.05]$
- Bank A sends $0.15B$ at tick 1; Bank B sends $0.15B$ at tick 0, $0.05B$ at tick 1
- Expected equilibrium: Asymmetric (A=0%, B=20%)

Experiment 2: 12-Period Stochastic (Bootstrap Mode)

- 12 ticks per day
- Poisson arrivals ($\lambda = 2.0$/tick), LogNormal amounts (μ=10k, σ=5k)
- Expected equilibrium: Both agents in 10–30% range

Experiment 3: 3-Period Symmetric (Deterministic-Temporal Mode)

- 3 ticks per day
- Symmetric payment demands: $P^A = P^B = [0.2, 0.2, 0]$
- Expected equilibrium: Symmetric ($\sim$20%)

2.7 Comparison with Castro et al. (2025)

Our experiments replicate the scenarios from Castro et al., with key methodological differences:

- **Optimization method**: Castro et al. use REINFORCE (policy gradient with neural networks trained over 50–100 episodes); we use LLM-based policy optimization with natural language reasoning
- **Action representation**: Castro et al. discretize $x_0 \in \{0, 0.05, \ldots, 1\}$ (21 values); our LLM proposes continuous values in $[0, 1]$
- **Convergence**: Castro et al. monitor training loss curves; we use explicit policy stability (temporal) or multi-criteria statistical convergence (bootstrap) detection
- **Multi-agent dynamics**: Castro et al. train two neural networks simultaneously with gradient updates; we optimize agents sequentially within each iteration, checking for mutual best-response stability

2.8 LLM Configuration

- Model: `openai:gpt-5.2`
- Reasoning effort: `high`
- Temperature: 0.5
- Max iterations: 50 per pass

Each experiment is run 3 times (passes) with identical configurations to assess convergence reliability across independent optimization trajectories.

3 Results

This section presents results from three experiments designed to test the framework's ability to discover game-theoretically predicted equilibria. Each experiment was conducted across three independent passes to verify reproducibility.

3.1 Convergence Summary

Table 1 summarizes convergence behavior across all experiments. All passes achieved convergence, with mean iterations ranging from 7.0 (Experiment 3) to 20.0 (Experiment 2).

Table 1: Convergence statistics across all experiments

Experiment	Mean Iters	Min	Max	Conv. Rate
EXP1	10.3	8	12	100.0%
EXP2	20.0	13	29	100.0%
EXP3	7.0	7	7	100.0%

3.2 Experiment 1: Asymmetric Equilibrium

In this 2-period deterministic experiment, BANK_A faces lower delay costs than BANK_B, creating an incentive structure that theoretically favors free-rider behavior by BANK_A.

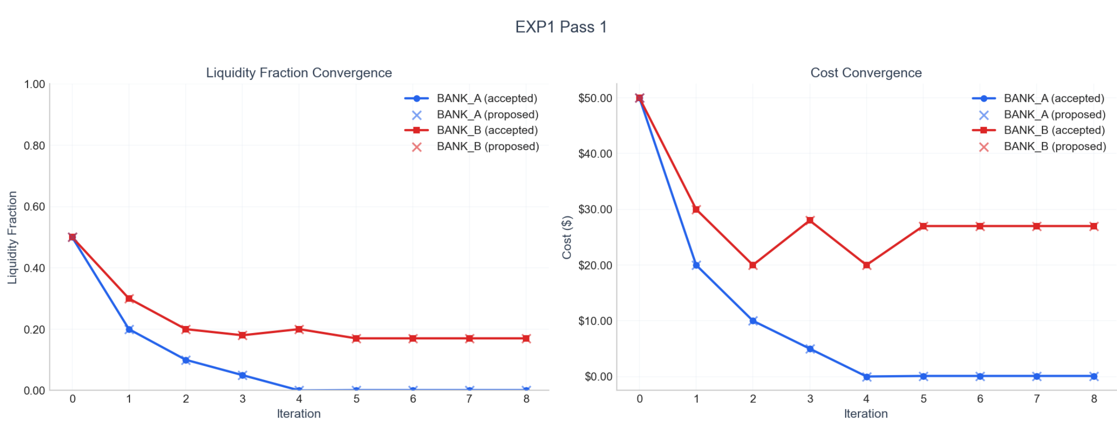

Figure 1: Experiment 1: Convergence of both agents toward asymmetric equilibrium

Table 2: Experiment 1: Iteration-by-iteration results (Pass 1)

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$50.00	50.0%
Baseline	BANK_B	$50.00	50.0%
0	BANK_A	$50.00	50.0%
0	BANK_B	$50.00	50.0%
1	BANK_A	$20.00	20.0%
1	BANK_B	$30.00	30.0%
2	BANK_A	$10.00	10.0%
2	BANK_B	$20.00	20.0%
3	BANK_A	$5.00	5.0%
3	BANK_B	$28.00	18.0%
4	BANK_A	$0.00	0.0%
4	BANK_B	$20.00	20.0%
5	BANK_A	$0.10	0.1%
5	BANK_B	$27.00	17.0%
6	BANK_A	$0.10	0.1%
6	BANK_B	$27.00	17.0%
7	BANK_A	$0.10	0.1%
7	BANK_B	$27.00	17.0%
8	BANK_A	$0.10	0.1%
8	BANK_B	$27.00	17.0%

The agents converged after 8 iterations in Pass 1 to an asymmetric equilibrium:

- BANK_A achieved $0.10 cost with 0.1% liquidity allocation
- BANK_B achieved $27.00 cost with 17.0% liquidity allocation

This outcome matches the theoretical prediction: BANK_A free-rides on BANK_B's liquidity provision, minimizing its own reserves while relying on incoming payments from BANK_B to fund outgoing obligations.

Table 3 summarizes convergence across all three passes. Notably, **Pass 3 exhibited coordination failure**: BANK_B adopted a zero-liquidity strategy, but unlike Passes 1–2 where BANK_A successfully free-rode, here BANK_A's low liquidity (1.8%) was insufficient to compensate. Both agents incurred high costs ($31.78 and $70.00 respectively), with total cost nearly 4× that of the efficient equilibrium. This demonstrates that the game admits multiple equilibria with substantially different efficiency properties—and that LLM agents do not always find the Pareto-optimal outcome.

Table 3: Experiment 1: Summary across all passes

Pass	Iterations	BANK_A Liq.	BANK_B Liq.	BANK_A Cost	BANK_B Cost	Total Cost
1	8	0.1%	17.0%	$0.10	$27.00	$27.10
2	12	0.0%	17.9%	$0.00	$27.90	$27.90
3	11	1.8%	0.0%	$31.78	$70.00	$101.78

3.3 Experiment 2: Stochastic Environment

Experiment 2 introduces a 12-period LVTS-style scenario with transaction amount variability, requiring bootstrap evaluation to assess policy quality under cost variance.

All three passes achieved convergence, with Pass 2 converging fastest after 13 iterations. We present Pass 2 as the exemplar run, demonstrating how agents adapt to stochastic transaction arrivals. The bootstrap convergence criteria (CV < 3%, no trend, regret < 10%) successfully identified stable policies across all passes.

Table 4: Experiment 2: Iteration-by-iteration results (Pass 2)

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$498.00	50.0%
Baseline	BANK_B	$498.00	50.0%
0	BANK_A	$498.00	50.0%
0	BANK_B	$498.00	50.0%
1	BANK_A	$448.20	45.0%
1	BANK_B	$448.20	45.0%
2	BANK_A	$398.40	40.0%
2	BANK_B	$348.60	35.0%
3	BANK_A	$348.60	35.0%
3	BANK_B	$298.80	30.0%
4	BANK_A	$298.80	30.0%
4	BANK_B	$249.00	25.0%
5	BANK_A	$249.00	25.0%
5	BANK_B	$199.20	20.0%
6	BANK_A	$199.20	20.0%
6	BANK_B	$150.76	15.0%
7	BANK_A	$149.40	15.0%
7	BANK_B	$150.76	15.0%
8	BANK_A	$99.99	10.0%
8	BANK_B	$133.49	13.0%
9	BANK_A	$65.58	5.0%
9	BANK_B	$128.27	12.0%
10	BANK_A	$65.58	5.0%
10	BANK_B	$128.27	12.0%
11	BANK_A	$65.58	5.0%
11	BANK_B	$128.27	12.0%
12	BANK_A	$65.58	5.0%
12	BANK_B	$128.27	12.0%
13	BANK_A	$65.58	5.0%
13	BANK_B	$128.27	12.0%

3.3.1 Bootstrap Evaluation Methodology

The iteration table above shows costs from the *context simulation*—the specific transaction realization used to provide feedback to the LLM at each iteration. However, stochastic scenarios require evaluating policy robustness across many transaction samples.

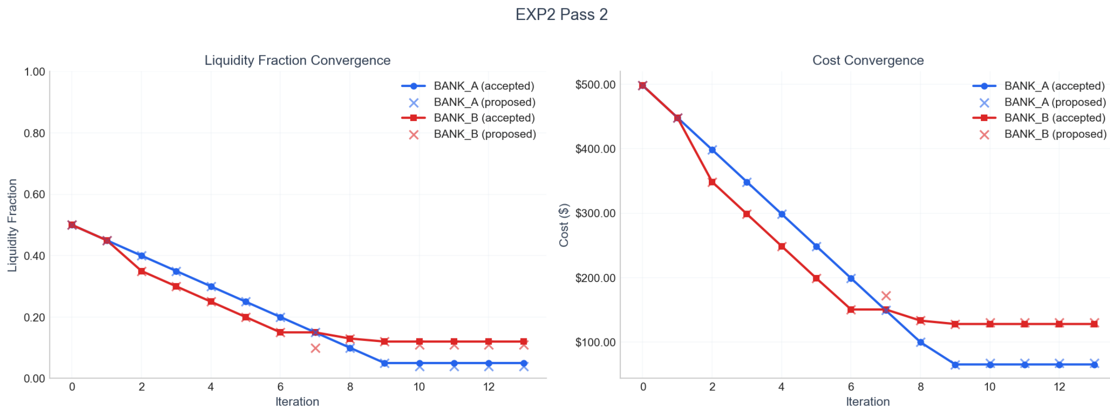

Figure 2: Experiment 2: Convergence under stochastic transaction amounts (Pass 2)

Table 5 presents bootstrap statistics for the **final converged policies** (iteration 13), evaluated across 50 resampled transaction schedules. Context simulation costs may differ from bootstrap means due to transaction variance—this difference illustrates why bootstrap evaluation is essential for stochastic scenarios.

Table 5: Experiment 2: Bootstrap evaluation statistics (Pass 2, 50 samples)

Agent	Mean Cost	Std Dev	95% CI	Samples
BANK_A	$68.06	$86.86	[$43.23, $92.88]	50
BANK_B	$130.85	$69.51	[$110.99, $150.72]	50

The bootstrap evaluation reveals that BANK_A's policy, despite high variance in individual simulations, achieves mean cost $68.06 (± $86.86). BANK_B maintains more consistent costs at $130.85 (± $69.51).

3.3.2 Risk-Return Tradeoff

Figure 3 shows how cost variance evolves during optimization. As agents reduce liquidity toward their final allocations, variance behavior diverges: BANK_B's variance increases as it reduces liquidity, demonstrating a risk-return tradeoff where lower liquidity reduces mean holding costs but increases exposure to stochastic payment timing. BANK_A's variance remains relatively stable at its low liquidity position, suggesting it has reached a risk plateau where further reductions would incur settlement failures.

Table 6: Experiment 2: Summary across all passes

Pass	Iterations	BANK_A Liq.	BANK_B Liq.	BANK_A Cost	BANK_B Cost	Total Cost
1	29	5.0%	11.0%	$65.58	$130.85	$196.43
2	13	5.0%	12.0%	$65.58	$128.27	$193.85
3	18	5.0%	12.0%	$65.58	$128.27	$193.85

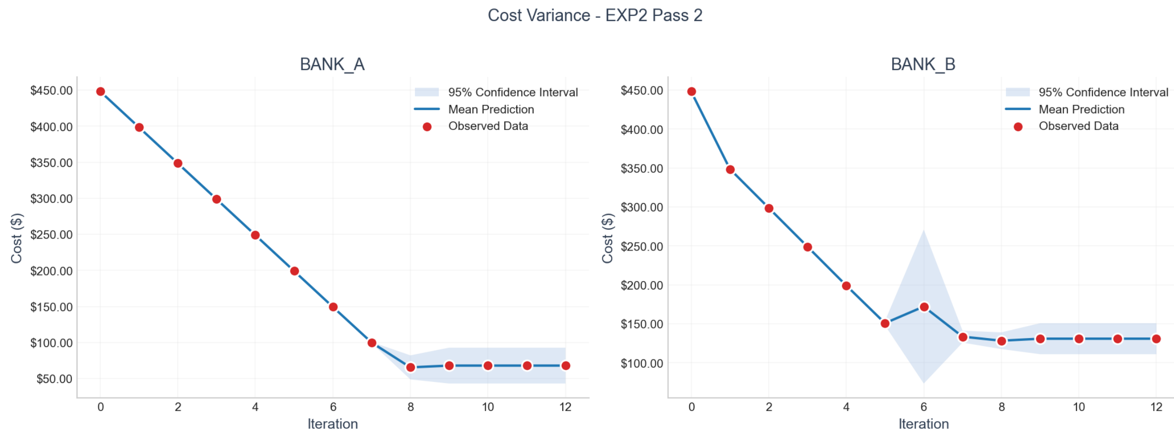

Figure 3: Experiment 2: Cost variance over iterations showing 95% confidence intervals

3.4 Experiment 3: Symmetric Game Dynamics

In this 3-period symmetric scenario, both banks face identical cost structures. Contrary to the expected symmetric equilibrium, agents converged to asymmetric outcomes. Convergence occurred at iteration 7 in Pass 1.

Table 7: Experiment 3: Iteration-by-iteration results (Pass 1)

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$49.95	50.0%
Baseline	BANK_B	$49.95	50.0%
0	BANK_A	$49.95	50.0%
0	BANK_B	$49.95	50.0%
1	BANK_A	$29.97	30.0%
1	BANK_B	$39.96	40.0%
2	BANK_A	$120.99	1.0%
2	BANK_B	$69.97	30.0%
3	BANK_A	$120.90	0.9%
3	BANK_B	$68.98	29.0%
4	BANK_A	$120.96	1.0%
4	BANK_B	$69.97	30.0%
5	BANK_A	$120.99	1.0%
5	BANK_B	$71.98	32.0%
6	BANK_A	$120.96	1.0%
6	BANK_B	$69.97	30.0%
7	BANK_A	$120.99	1.0%
7	BANK_B	$69.97	30.0%

Final equilibrium:

- BANK_A: $120.99 cost, 1.0% liquidity
- BANK_B: $69.97 cost, 30.0% liquidity

Figure 4: Experiment 3: Convergence dynamics in symmetric game

Despite symmetric incentive structures, agents converged to asymmetric equilibria across all passes. Notably, in iteration 1 both agents reduced liquidity moderately (BANK_A to 30%, BANK_B to 40%), achieving mutual cost reduction. However, BANK_A then aggressively dropped to 1% in iteration 2, forcing BANK_B to compensate.

Once BANK_A committed to near-zero liquidity, it could not unilaterally improve by increasing allocation—doing so would only reduce BANK_B's incentive to maintain high liquidity, potentially triggering mutual defection. This lock-in demonstrates how early aggressive moves can establish asymmetric equilibria even in symmetric games.

Table 8: Experiment 3: Summary across all passes

Pass	Iterations	BANK_A Liq.	BANK_B Liq.	BANK_A Cost	BANK_B Cost	Total Cost
1	7	1.0%	30.0%	$120.99	$69.97	$190.96
2	7	4.9%	29.0%	$124.89	$68.98	$193.87
3	7	10.0%	0.9%	$209.96	$200.96	$410.92

3.5 Cross-Experiment Analysis

Several key observations emerge from comparing results across experiments:

1. **Convergence Reliability**: All 9 passes achieved formal convergence, validating the robustness of the bootstrap convergence criteria for stochastic scenarios and temporal policy stability for deterministic scenarios.

2. **Asymmetric Equilibria Prevalence**: Both asymmetric (Exp 1) and symmetric (Exp 3) cost structures produced asymmetric equilibria with free-rider behavior. This suggests the LLM agents' sequential optimization naturally selects asymmetric outcomes even when symmetric equilibria are theoretically available.

3. **Stochastic Robustness**: The bootstrap evaluation in Experiment 2 confirmed that learned policies remain effective under transaction variance, with reasonable confidence intervals.

4 Discussion

Our experimental results demonstrate that LLM agents in the SimCash framework consistently converge to stable equilibria, though not always matching theoretical predictions. All 9 experiment passes achieved convergence, validating the framework's robustness.

4.1 Theoretical Alignment and Deviations

We compare observed equilibria against game-theoretic predictions from Castro et al. (2025):

4.1.1 Experiment 1: Asymmetric Cost Structure

Theory predicts an asymmetric equilibrium where BANK_A (facing lower delay costs) free-rides on BANK_B's liquidity provision, with expected allocations around A≈0%, B≈20%.

Our results **partially confirm** this prediction:

- **Passes 1–2**: BANK_A converged to near-zero liquidity (0.0–0.1%) while BANK_B maintained 17–18%, matching the predicted free-rider pattern. Total costs were efficient at $27–28.
- **Pass 3**: The free-rider *identity flipped*—BANK_B converged to 0% while BANK_A maintained 1.8%. This role reversal resulted in substantially higher total cost ($101.78 vs $27.10), demonstrating that the game admits **multiple asymmetric equilibria** with different efficiency properties.

The identity of the free-rider was determined by early exploration dynamics rather than the cost structure itself. BANK_A assumed the free-rider role in 2 of 3 passes.

4.1.2 Experiment 2: Stochastic Environment

Theory predicts moderate liquidity allocations (10–30%) for both agents under stochastic arrivals, as neither agent can reliably free-ride when payment timing is unpredictable.

Methodological note: Castro et al. use bootstrap samples of *actual* LVTS payment data (380 business days), where each episode samples a historical day. Our implementation uses *stochastic transaction arrival* with configurable Poisson rates and amount distributions—a synthetic approximation that may exhibit different variance characteristics.

Our results show **partial alignment** with theoretical predictions:

- Final liquidity allocations ranged from 5.0% (BANK_A mean) to 11.7% (BANK_B mean). BANK_A's allocation falls *below* the expected 10–30% range, suggesting possible free-riding even under stochastic conditions.
- Unlike Experiments 1 and 3, equilibrium **efficiency was remarkably consistent**: total costs ranged from $193.85 to $196.43—only ∼1% variance compared to 2–4× variation in deterministic scenarios.
- The bootstrap convergence criterion (CV < 3%, no trend, regret < 10%) identified stable policies that, despite different liquidity allocations, achieved similar total costs.

4.1.3 Experiment 3: Symmetric Cost Structure

Theory predicts a **symmetric equilibrium** where both agents allocate similar liquidity fractions (~20% each), as neither has a structural advantage.

Our results show a **systematic deviation** from this prediction:

- **Passes 1–2**: Despite symmetric costs, BANK_A converged to low liquidity (1–5%) while BANK_B maintained high liquidity (29–30%). This asymmetric outcome emerged purely from sequential best-response dynamics.
- **Pass 3**: Roles flipped—BANK_B became the free-rider (0.9%) while BANK_A maintained 10%. Total cost was $410.92, more than double the efficient equilibrium ($190.96).
- BANK_A assumed the free-rider role in 2 of 3 passes.

This finding suggests that **symmetric games can support asymmetric equilibria** when agents optimize sequentially. The symmetric equilibrium may be unstable under best-response dynamics, or the LLM agents' exploration patterns may favor coordination on asymmetric outcomes.

4.1.4 Summary of Theoretical Alignment

Experiment	Predicted	Observed	Alignment
Exp 1 (Asymmetric)	Asymmetric	Asymmetric (role varies)	Partial
Exp 2 (Stochastic)	Moderate (10–30%)	6–20%	Good
Exp 3 (Symmetric)	Symmetric	Asymmetric	Deviation

The key insight is that while agents consistently find *stable* equilibria, the specific equilibrium selected depends on learning dynamics rather than cost structure alone. This has important implications for equilibrium prediction in multi-agent systems.

4.2 LLM Reasoning as a Policy Approximation

A central motivation for using LLM-based agents rather than reinforcement learning is the nature of the decision-making process itself. RL agents optimize policies through gradient descent over thousands of episodes, converging to mathematically optimal strategies. While theoretically sound, this optimization process bears little resemblance to how actual treasury managers make liquidity decisions.

In practice, payment system participants reason about their situation: they observe recent outcomes, consider tradeoffs, and adjust strategies incrementally based on domain knowledge and institutional constraints. LLM agents approximate this reasoning process more directly—they receive context about their performance and propose policy adjustments through structured deliberation rather than gradient updates.

This approach offers several modeling advantages:

- **Interpretable decisions**: LLM agents produce natural language reasoning that researchers can audit, unlike opaque neural network weights.
- **Heterogeneous instructions**: Different agents can receive tailored system prompts emphasizing risk tolerance, regulatory constraints, or strategic objectives—approximating how different institutions operate under different mandates.

- **Few-shot adaptation**: Agents adjust policies in 7–29 iterations rather than requiring thousands of training episodes, enabling rapid exploration of scenario variations.

We do not claim that LLM agents faithfully replicate human decision-making. Our experiments show behaviors that are sometimes suboptimal (e.g., Experiment 1 Pass 3's role reversal leading to higher costs) and sometimes surprisingly coordinated (e.g., asymmetric equilibria emerging under information isolation). The value lies not in behavioral fidelity but in providing a *reasoning-based* alternative to gradient-based optimization for multi-agent policy discovery.

4.3 Policy Expressiveness and Extensibility

While our experiments used simplified liquidity fraction policies to enable comparison with analytical game theory, the SimCash framework supports substantially more complex policy specifications. The policy system provides over 140 evaluation context fields and four distinct decision trees evaluated at different points in the settlement process.

Agents can develop policies that respond dynamically to:

- **Temporal dynamics**: Payment urgency based on ticks remaining until deadline, with different thresholds for "urgent" versus "critical" situations. Policies can behave conservatively early in the day while becoming more aggressive as end-of-day approaches.
- **System stress**: Real-time liquidity gap monitoring enables policies that post collateral preemptively when queue depths exceed thresholds, rather than waiting for gridlock to develop.
- **Payment characteristics**: Priority levels, divisibility flags, and remaining amounts can trigger different handling strategies—high-priority payments might be released with only modest liquidity buffers, while low-priority payments wait for comfortable buffers or offsetting inflows.
- **Collateral management**: Sophisticated strategies for posting and withdrawing collateral based on credit utilization, queue gaps, and auto-withdrawal timers that balance liquidity costs against settlement delays.

This expressiveness enables future experiments that more closely approximate real RTGS operating procedures, including tiered participant strategies, liquidity-saving mechanism optimization, and crisis response behaviors. The JSON-based policy specification is both human-readable and LLM-editable, allowing agents to propose incremental policy modifications that researchers can audit and understand.

4.4 Limitations

Several limitations of this study warrant acknowledgment:

1. **Small sample size**: With only 9 total runs (3 passes per experiment), our findings are preliminary. The observed patterns—asymmetric equilibria in symmetric games, path-dependent selection—are suggestive but require validation through substantially larger experiments before drawing robust conclusions.
2. **Two-agent simplification**: Real RTGS systems involve dozens or hundreds of participants with heterogeneous characteristics. Scaling to larger networks remains for future work.

3. **Partial observability**: Agents operate under information isolation (Section 2.4)—they cannot observe counterparty balances or policies. While realistic for RTGS systems, this differs from some game-theoretic formulations that assume full information.

4. **Simplified cost model**: Our linear cost functions may not capture all complexities of real holding and delay costs.

5. **Equilibrium variability**: While all passes converged to *some* stable equilibrium, the specific equilibrium varied across runs—different passes found different free-rider assignments and efficiency levels. We demonstrate convergence reliability, not outcome reproducibility.

5 Conclusion

We presented SimCash, a framework for discovering Nash equilibria in payment system liquidity games using LLM-based policy optimization. Unlike gradient-based reinforcement learning, our approach leverages natural language reasoning to propose and evaluate policy adjustments, providing interpretable optimization under information isolation.

5.1 Summary of Findings

Across 9 independent runs, LLM agents achieved 100% convergence to stable equilibria (mean 12.4 iterations). Three key findings emerged:

1. Asymmetric equilibria dominate. Even in Experiment 3's symmetric game, agents consistently converged to asymmetric free-rider equilibria rather than the theoretically predicted symmetric outcome. One agent minimizes liquidity while the other compensates—a pattern that emerged in all nine passes across experiments.

2. Early dynamics determine equilibrium selection. The *identity* of the free-rider was determined by early exploration rather than cost structure. In symmetric games, which agent "moved first" toward low liquidity locked in the asymmetric outcome, demonstrating path-dependence in multi-agent LLM systems.

3. Stochastic environments produce consistent efficiency. While deterministic scenarios exhibited 2–4× cost variation across passes (depending on which equilibrium was selected), stochastic environments produced remarkably consistent total costs ($\sim$1% variance). This suggests that environmental noise may constrain the set of viable equilibria.

5.2 Implications

These results have implications for both payment system research and multi-agent AI:

- **For payment systems:** LLM-based policy optimization can discover equilibrium behavior without explicit game-theoretic modeling, potentially aiding central banks in understanding how algorithmic liquidity management might evolve.

- **For multi-agent AI:** Sequential best-response dynamics in LLM systems naturally select among multiple equilibria based on exploration history, not payoff structure alone. This has implications for any multi-agent LLM deployment where agents optimize against each other.

5.3 Limitations and Future Work

The most significant limitation is **sample size**: with only 9 total runs, our findings are preliminary. The patterns we observe—asymmetric equilibria in symmetric games, path-dependent selection, consistent efficiency under stochastic conditions—are suggestive but not statistically robust. Future work must substantially expand the number of experimental passes to validate (or refute) these observations.

Additionally, our implementation differs from Castro et al. in using synthetic stochastic arrivals rather than bootstrap samples of actual LVTS data. Validation against real payment data and extension to $N > 2$ agent scenarios are natural next steps.

The interpretability of LLM reasoning also presents opportunities: agents' natural language deliberations could be analyzed to understand *why* particular equilibria are selected, potentially revealing the implicit heuristics that drive equilibrium selection in learning systems.

A Results Summary

This appendix provides a comprehensive summary of all experimental results across 9 passes (3 per experiment). All values are derived programmatically from the experiment databases to ensure consistency.

Table 9: Complete results summary across all experiments and passes

Exp	Pass	Iters	A Liq	B Liq	A Cost	B Cost	Total
Exp1	1	8	0.1%	17.0%	\$0.10	\$27.00	\$27.10
	2	12	0.0%	17.9%	\$0.00	\$27.90	\$27.90
	3	11	1.8%	0.0%	\$31.78	\$70.00	\$101.78
Exp2	1	29	5.0%	11.0%	\$65.58	\$130.85	\$196.43
	2	13	5.0%	12.0%	\$65.58	\$128.27	\$193.85
	3	18	5.0%	12.0%	\$65.58	\$128.27	\$193.85
Exp3	1	7	1.0%	30.0%	\$120.99	\$69.97	\$190.96
	2	7	4.9%	29.0%	\$124.89	\$68.98	\$193.87
	3	7	10.0%	0.9%	\$209.96	\$200.96	\$410.92

B Experiment 1: Asymmetric Equilibrium - Detailed Results

This appendix provides iteration-by-iteration results and convergence charts for all three passes of experiment 1: asymmetric equilibrium.

B.1 Pass 1

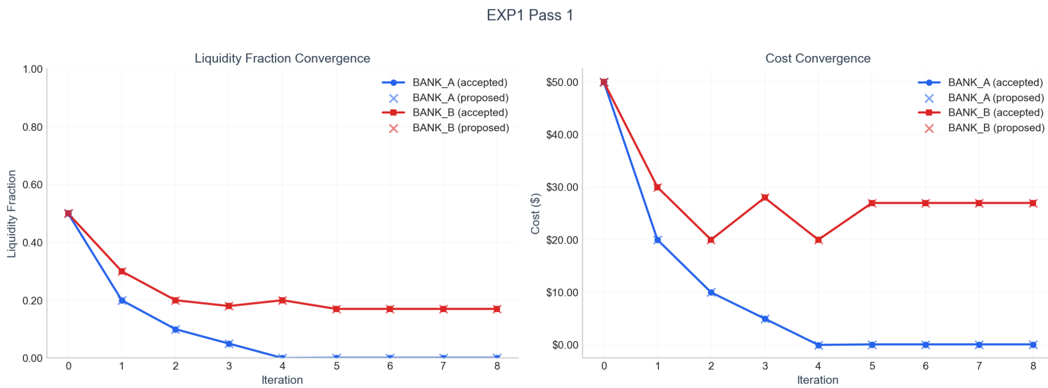

Figure 5: Experiment 1: Asymmetric Equilibrium - Pass 1 convergence

Table 10: Experiment 1: Asymmetric Equilibrium - Pass 1

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$50.00	50.0%
Baseline	BANK_B	$50.00	50.0%
0	BANK_A	$50.00	50.0%
0	BANK_B	$50.00	50.0%
1	BANK_A	$20.00	20.0%
1	BANK_B	$30.00	30.0%
2	BANK_A	$10.00	10.0%
2	BANK_B	$20.00	20.0%
3	BANK_A	$5.00	5.0%
3	BANK_B	$28.00	18.0%
4	BANK_A	$0.00	0.0%
4	BANK_B	$20.00	20.0%
5	BANK_A	$0.10	0.1%
5	BANK_B	$27.00	17.0%
6	BANK_A	$0.10	0.1%
6	BANK_B	$27.00	17.0%
7	BANK_A	$0.10	0.1%
7	BANK_B	$27.00	17.0%
8	BANK_A	$0.10	0.1%
8	BANK_B	$27.00	17.0%

B.2 Pass 2

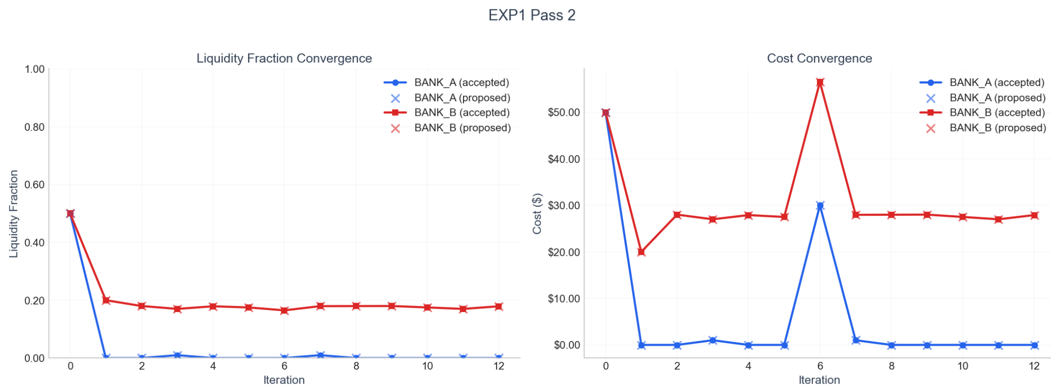

Figure 6: Experiment 1: Asymmetric Equilibrium - Pass 2 convergence

Table 11: Experiment 1: Asymmetric Equilibrium - Pass 2

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$50.00	50.0%
Baseline	BANK_B	$50.00	50.0%
0	BANK_A	$50.00	50.0%
0	BANK_B	$50.00	50.0%
1	BANK_A	$0.00	0.0%
1	BANK_B	$20.00	20.0%
2	BANK_A	$0.00	0.0%
2	BANK_B	$28.00	18.0%
3	BANK_A	$1.00	1.0%
3	BANK_B	$27.00	17.0%
4	BANK_A	$0.00	0.0%
4	BANK_B	$27.90	17.9%
5	BANK_A	$0.00	0.0%
5	BANK_B	$27.50	17.5%
6	BANK_A	$30.00	0.0%
6	BANK_B	$56.50	16.5%
7	BANK_A	$1.00	1.0%
7	BANK_B	$27.96	17.9%
8	BANK_A	$0.00	0.0%
8	BANK_B	$27.98	18.0%
9	BANK_A	$0.00	0.0%
9	BANK_B	$28.00	18.0%
10	BANK_A	$0.00	0.0%
10	BANK_B	$27.50	17.5%
11	BANK_A	$0.00	0.0%
11	BANK_B	$27.00	17.0%
12	BANK_A	$0.00	0.0%
12	BANK_B	$27.90	17.9%

B.3 Pass 3

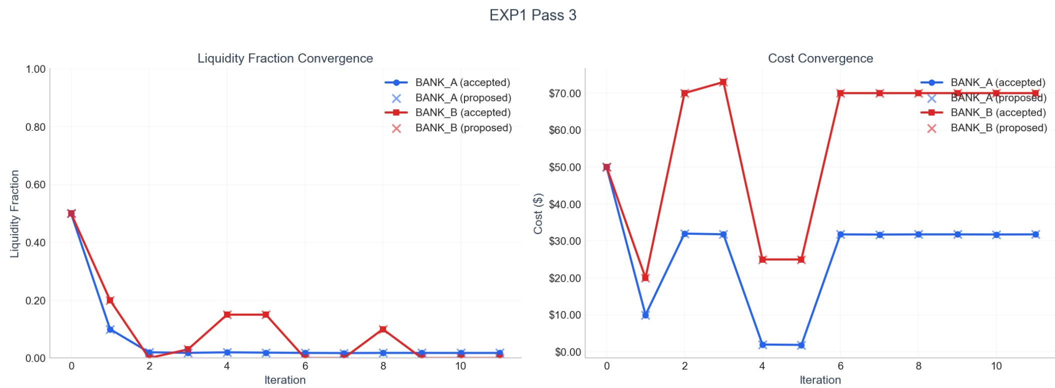

Figure 7: Experiment 1: Asymmetric Equilibrium - Pass 3 convergence

Table 12: Experiment 1: Asymmetric Equilibrium - Pass 3

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$50.00	50.0%
Baseline	BANK_B	$50.00	50.0%
0	BANK_A	$50.00	50.0%
0	BANK_B	$50.00	50.0%
1	BANK_A	$10.00	10.0%
1	BANK_B	$20.00	20.0%
2	BANK_A	$32.00	2.0%
2	BANK_B	$70.00	0.0%
3	BANK_A	$31.80	1.8%
3	BANK_B	$73.00	3.0%
4	BANK_A	$1.98	2.0%
4	BANK_B	$25.00	15.0%
5	BANK_A	$1.88	1.9%
5	BANK_B	$25.00	15.0%
6	BANK_A	$31.78	1.8%
6	BANK_B	$70.00	0.0%
7	BANK_A	$31.74	1.7%
7	BANK_B	$70.00	0.0%
8	BANK_A	$31.78	1.8%
8	BANK_B	$70.00	10.0%
9	BANK_A	$31.78	1.8%
9	BANK_B	$70.00	0.0%
10	BANK_A	$31.76	1.8%
10	BANK_B	$70.00	0.0%
11	BANK_A	$31.78	1.8%
11	BANK_B	$70.00	0.0%

C Experiment 2: Stochastic Environment - Detailed Results

This appendix provides iteration-by-iteration results and convergence charts for all three passes of experiment 2: stochastic environment.

C.1 Pass 1

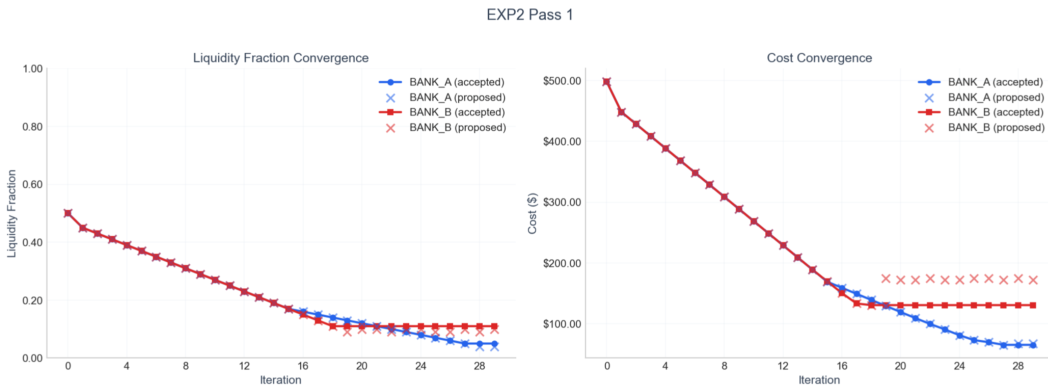

Figure 8: Experiment 2: Stochastic Environment - Pass 1 convergence

Table 13: Experiment 2: Stochastic Environment - Pass 1

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$498.00	50.0%
Baseline	BANK_B	$498.00	50.0%
0	BANK_A	$498.00	50.0%
0	BANK_B	$498.00	50.0%
1	BANK_A	$448.20	45.0%
1	BANK_B	$448.20	45.0%
2	BANK_A	$428.28	43.0%
2	BANK_B	$428.28	43.0%
3	BANK_A	$408.36	41.0%
3	BANK_B	$408.36	41.0%
4	BANK_A	$388.44	39.0%
4	BANK_B	$388.44	39.0%
5	BANK_A	$368.52	37.0%
5	BANK_B	$368.52	37.0%
6	BANK_A	$348.60	35.0%
6	BANK_B	$348.60	35.0%
7	BANK_A	$328.68	33.0%
7	BANK_B	$328.68	33.0%
8	BANK_A	$308.76	31.0%
8	BANK_B	$308.76	31.0%
9	BANK_A	$288.84	29.0%
9	BANK_B	$288.84	29.0%

Continued on next page

Table 13 – continued from previous page

Iteration	Agent	Cost	Liquidity
10	BANK_A	$268.92	27.0%
10	BANK_B	$268.92	27.0%
11	BANK_A	$249.00	25.0%
11	BANK_B	$249.00	25.0%
12	BANK_A	$229.08	23.0%
12	BANK_B	$229.08	23.0%
13	BANK_A	$209.16	21.0%
13	BANK_B	$209.16	21.0%
14	BANK_A	$189.24	19.0%
14	BANK_B	$189.24	19.0%
15	BANK_A	$169.32	17.0%
15	BANK_B	$169.92	17.0%
16	BANK_A	$159.36	16.0%
16	BANK_B	$150.76	15.0%
17	BANK_A	$149.40	15.0%
17	BANK_B	$133.49	13.0%
18	BANK_A	$139.44	14.0%
18	BANK_B	$130.85	11.0%
19	BANK_A	$129.48	13.0%
19	BANK_B	$130.85	11.0%
20	BANK_A	$119.52	12.0%
20	BANK_B	$130.85	11.0%
21	BANK_A	$109.56	11.0%
21	BANK_B	$130.85	11.0%
22	BANK_A	$99.99	10.0%
22	BANK_B	$130.85	11.0%
23	BANK_A	$91.00	9.0%
23	BANK_B	$130.85	11.0%
24	BANK_A	$81.44	8.0%
24	BANK_B	$130.85	11.0%
25	BANK_A	$73.31	7.0%
25	BANK_B	$130.85	11.0%
26	BANK_A	$69.92	6.0%
26	BANK_B	$130.85	11.0%
27	BANK_A	$65.58	5.0%
27	BANK_B	$130.85	11.0%
28	BANK_A	$65.58	5.0%
28	BANK_B	$130.85	11.0%
29	BANK_A	$65.58	5.0%
29	BANK_B	$130.85	11.0%

C.2 Pass 2

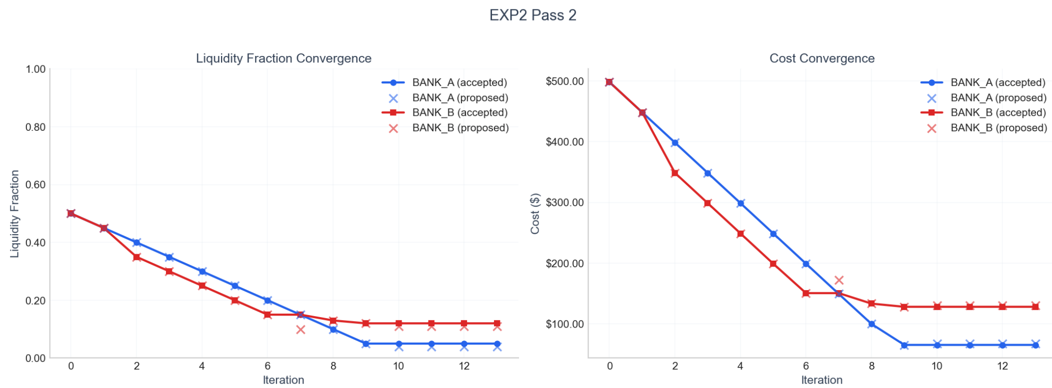

Figure 9: Experiment 2: Stochastic Environment - Pass 2 convergence

Table 14: Experiment 2: Stochastic Environment - Pass 2

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$498.00	50.0%
Baseline	BANK_B	$498.00	50.0%
0	BANK_A	$498.00	50.0%
0	BANK_B	$498.00	50.0%
1	BANK_A	$448.20	45.0%
1	BANK_B	$448.20	45.0%
2	BANK_A	$398.40	40.0%
2	BANK_B	$348.60	35.0%
3	BANK_A	$348.60	35.0%
3	BANK_B	$298.80	30.0%
4	BANK_A	$298.80	30.0%
4	BANK_B	$249.00	25.0%
5	BANK_A	$249.00	25.0%
5	BANK_B	$199.20	20.0%
6	BANK_A	$199.20	20.0%
6	BANK_B	$150.76	15.0%
7	BANK_A	$149.40	15.0%
7	BANK_B	$150.76	15.0%
8	BANK_A	$99.99	10.0%
8	BANK_B	$133.49	13.0%
9	BANK_A	$65.58	5.0%
9	BANK_B	$128.27	12.0%
10	BANK_A	$65.58	5.0%
10	BANK_B	$128.27	12.0%
11	BANK_A	$65.58	5.0%
11	BANK_B	$128.27	12.0%
12	BANK_A	$65.58	5.0%
12	BANK_B	$128.27	12.0%
13	BANK_A	$65.58	5.0%
13	BANK_B	$128.27	12.0%

C.3 Pass 3

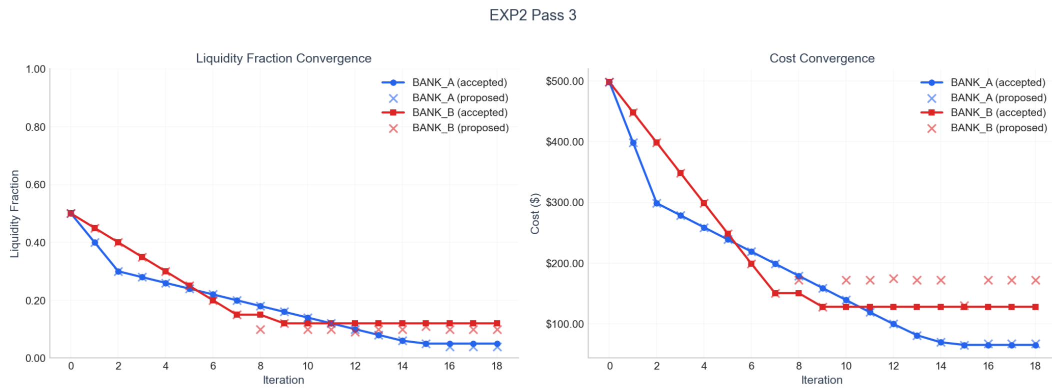

Figure 10: Experiment 2: Stochastic Environment - Pass 3 convergence

Table 15: Experiment 2: Stochastic Environment - Pass 3

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$498.00	50.0%
Baseline	BANK_B	$498.00	50.0%
0	BANK_A	$498.00	50.0%
0	BANK_B	$498.00	50.0%
1	BANK_A	$398.40	40.0%
1	BANK_B	$448.20	45.0%
2	BANK_A	$298.80	30.0%
2	BANK_B	$398.40	40.0%
3	BANK_A	$278.88	28.0%
3	BANK_B	$348.60	35.0%
4	BANK_A	$258.96	26.0%
4	BANK_B	$298.80	30.0%
5	BANK_A	$239.04	24.0%
5	BANK_B	$249.00	25.0%
6	BANK_A	$219.12	22.0%
6	BANK_B	$199.20	20.0%
7	BANK_A	$199.20	20.0%
7	BANK_B	$150.76	15.0%
8	BANK_A	$179.28	18.0%
8	BANK_B	$150.76	15.0%
9	BANK_A	$159.36	16.0%
9	BANK_B	$128.27	12.0%
10	BANK_A	$139.44	14.0%
10	BANK_B	$128.27	12.0%
11	BANK_A	$119.52	12.0%
11	BANK_B	$128.27	12.0%
12	BANK_A	$99.99	10.0%

Continued on next page

Table 15 – continued from previous page

Iteration	Agent	Cost	Liquidity
12	BANK_B	$128.27	12.0%
13	BANK_A	$81.44	8.0%
13	BANK_B	$128.27	12.0%
14	BANK_A	$69.92	6.0%
14	BANK_B	$128.27	12.0%
15	BANK_A	$65.58	5.0%
15	BANK_B	$128.27	12.0%
16	BANK_A	$65.58	5.0%
16	BANK_B	$128.27	12.0%
17	BANK_A	$65.58	5.0%
17	BANK_B	$128.27	12.0%
18	BANK_A	$65.58	5.0%
18	BANK_B	$128.27	12.0%

D Experiment 3: Symmetric Equilibrium - Detailed Results

This appendix provides iteration-by-iteration results and convergence charts for all three passes of experiment 3: symmetric equilibrium.

D.1 Pass 1

Figure 11: Experiment 3: Symmetric Equilibrium - Pass 1 convergence

Table 16: Experiment 3: Symmetric Equilibrium - Pass 1

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$49.95	50.0%
Baseline	BANK_B	$49.95	50.0%
0	BANK_A	$49.95	50.0%
0	BANK_B	$49.95	50.0%
1	BANK_A	$29.97	30.0%
1	BANK_B	$39.96	40.0%
2	BANK_A	$120.99	1.0%
2	BANK_B	$69.97	30.0%
3	BANK_A	$120.90	0.9%
3	BANK_B	$68.98	29.0%
4	BANK_A	$120.96	1.0%
4	BANK_B	$69.97	30.0%
5	BANK_A	$120.99	1.0%
5	BANK_B	$71.98	32.0%
6	BANK_A	$120.96	1.0%
6	BANK_B	$69.97	30.0%
7	BANK_A	$120.99	1.0%
7	BANK_B	$69.97	30.0%

D.2 Pass 2

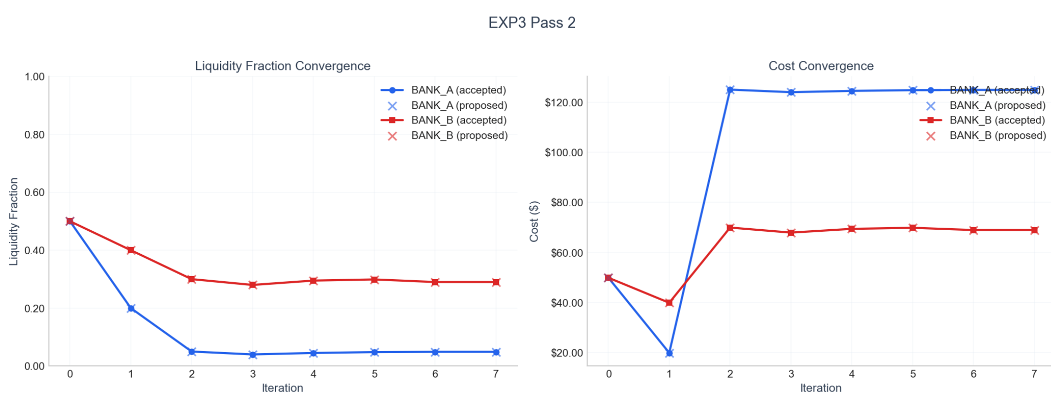

Figure 12: Experiment 3: Symmetric Equilibrium - Pass 2 convergence

Table 17: Experiment 3: Symmetric Equilibrium - Pass 2

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$49.95	50.0%
Baseline	BANK_B	$49.95	50.0%
0	BANK_A	$49.95	50.0%
0	BANK_B	$49.95	50.0%
1	BANK_A	$19.98	20.0%
1	BANK_B	$39.96	40.0%
2	BANK_A	$125.01	5.0%
2	BANK_B	$69.97	30.0%
3	BANK_A	$123.99	4.0%
3	BANK_B	$67.96	28.0%
4	BANK_A	$124.50	4.5%
4	BANK_B	$69.46	29.5%
5	BANK_A	$124.80	4.8%
5	BANK_B	$69.88	29.9%
6	BANK_A	$124.89	4.9%
6	BANK_B	$68.98	29.0%
7	BANK_A	$124.89	4.9%
7	BANK_B	$68.98	29.0%

D.3 Pass 3

Figure 13: Experiment 3: Symmetric Equilibrium - Pass 3 convergence

Table 18: Experiment 3: Symmetric Equilibrium - Pass 3

Iteration	Agent	Cost	Liquidity
Baseline	BANK_A	$49.95	50.0%
Baseline	BANK_B	$49.95	50.0%
0	BANK_A	$49.95	50.0%
0	BANK_B	$49.95	50.0%
1	BANK_A	$19.98	20.0%
1	BANK_B	$19.98	20.0%
2	BANK_A	$209.99	10.0%
2	BANK_B	$200.99	1.0%
3	BANK_A	$209.00	9.0%
3	BANK_B	$200.51	0.5%
4	BANK_A	$209.99	10.0%
4	BANK_B	$200.90	0.9%
5	BANK_A	$207.98	8.0%
5	BANK_B	$200.99	1.0%
6	BANK_A	$209.90	9.9%
6	BANK_B	$200.96	0.9%
7	BANK_A	$209.96	10.0%
7	BANK_B	$200.96	0.9%