GroupPay Whitepaper

Michael Genchev

September 2025

1 GroupPay Engine: Formal Specification

Abstract This specification defines a dual-modal expense distribution algorithm that optimizes both computational efficiency and financial equity within group payment systems.

2 Mathematical Framework

2.1 Definitions

Let

$$G = \{u_1, u_2, \ldots, u_n\}$$

represent the participant set, where $n \geq 2$.

Let

$$E = \{e_1, e_2, \ldots, e_m\}$$

denote the expense vector, where each $e_j \in \mathbb{R}^n$.

Total Expense:

$$E_t = \sum_{j=1}^{m} e_j$$

Payment Matrix:

$$P \in \mathbb{R}^{n \times m}$$

2.2 Algorithmic Specifications

Algorithm A: Equal Distribution

Balance Computation

$$balance_i = \sum_{j=1}^{m} P_{ij} - \frac{E_{total}}{n}$$

Computational Complexity: $O(n)$

Algorithm B: Item-Attribution

Consumption Matrix

$$C \in \{0,1\}^{n \times m}, \quad C_{ij} = 1 \text{ if participant } u_i \text{ consumed item } e_j$$

Individual Obligation

$$obligation_i = \sum_{j=1}^{m} C_{ij} \cdot e_j + sharedAllocation_i$$

Shared Cost Distribution

$$sharedAllocation_i = \frac{E_{shared}}{|participants|}$$

3 Settlement Optimization

Objective Function Minimize the transaction count T subject to balance equilibrium:

$$\min T \quad \text{s.t.} \quad \sum_{i=1}^{n} balance_i = 0$$

Debt Graph

$$D = (V, E) \quad \text{where} \quad V = G, \quad (u_i, u_j) \in E \text{ iff } balance_i < 0 \text{ and } balance_j > 0$$

Optimization Algorithm

1. Construct creditor set: $C = \{i : balance_i > 0\}$
2. Construct debtor set: $D = \{i : balance_i < 0\}$
3. Apply greedy matching: $\max(|balance_{creditor}|, |balance_{debtor}|)$
4. Generate minimal transaction sequence

4 Algorithm Selection Criterion

Price Variance Coefficient:

$$\sigma_{price} = \sqrt{\frac{\sum_{j=1}^{m}(e_j - \bar{e})^2}{m-1}}$$

Decision Function:

$$\text{Algorithm} = \begin{cases} \text{Equal Distribution}, & \text{if } \frac{\sigma_{price}}{\bar{e}} < \theta \\ \text{Item Attribution}, & \text{otherwise} \end{cases}$$

where $\theta \in [0.3, 0.5]$ represents the variance threshold parameter.

4.1 Convergence Properties

Theorem: Both algorithms guarantee financial equilibrium:

$$\sum_{i=1}^{n} obligation_i = E_{total}$$

Proof: Direct summation yields identity preservation under both computational paths.