CTU Journal of Innovation and Sustainable Development
ISSN 2588-1418 | e-ISSN 2815-6412

DOI:10.22144/ctujoisd.2026.054

Intelligent location-aware recommender system for urban micro-connections using general graph matching

Tran Huong Thao[1,2 *]

[1]Faculty of Mathematics and Computer Science, University of Science, Ho Chi Minh City, Viet Nam

[2]Vietnam National University, Ho Chi Minh City, Viet Nam

*Corresponding author (ththao@hcmus.edu.vn)

Article info.

Received 29 Jun 2026
Revised 3 Sep 2026
Accepted 15 Oct 2026

Keywords

General graph, intelligent systems, multi-objective matching, recommender systems, urban loneliness

ABSTRACT

The rise of conurbations exacerbates "urban loneliness," yet current connection platforms suffer from appearance bias, high cancellation rates, and inadequate safety constraints. To address these issues, our study proposes.SLOT, a real-time 1-on-1 social interaction system. It integrates anonymous recognition and a Location-Aware Recommender System to securely anchor meeting points in public spaces, optimizing bilateral converging distances. The core contribution is the multi-objective matching engine, solving the allocation problem on a generalized graph using Edmonds' Blossom algorithm. Combining time-window batching, elastic pruning, and symmetric credit scoring, our system empirically achieves a balanced multi-objective trade-off among liquidity, converging distance, and fairness. Compared to Weighted-sum Greedy at 3.8 and Distance-based Greedy at 13.0,.SLOT compresses credit deviation to 0.9 with a minor 1.8% trade-off in matching rate. Simulations on a synthetic dataset of 1,000 agents within a university campus micro-zone demonstrate that by spatially decomposing requests into micro-Anchor Stations, the system achieves sub-second latency, mitigating the $O(V^3)$ computational bottleneck of direct global batching.

1. INTRODUCTION

In rapidly expanding conurbations, "urban loneliness" (Hammoud et al., 2021), where high population density correlates with personal isolation, disrupts traditional friendship-building mechanisms and threatens young adults' mental health (Lee et al., 2022). A global Bumble BFF report reveals that 73% of Millennials long to make new friends, yet 52% admit they have not made any in the past year (Bumble, Survey conducted by Censuswide, 2023).

To identify specific behavioral requirements and formulate a technical solution for this widespread phenomenon, a system-design survey was conducted in a high-density university township. The study adopts a descriptive cross-sectional survey design combined with exploratory research among undergraduate students at VNUHCM-University of Science. Via non-probability sampling (combining convenience and online snowball sampling), a valid sample size of N = 268 was collected. The questionnaire comprises 9 closed-ended, single-choice items structured into two tiers: 4 frequency-measurement items adapted from the shortened UCLA Loneliness Scale and 5 behavioral and psychological items grounded in the Technology Acceptance Model (TAM; Davis, 1989) and UTAUT (Venkatesh et al., 2003), to

identify empirical needs and key user-reported friction addressed by .SLOT.

We collected data via Google Forms with mandatory single-option fields to eliminate missing values and logical errors, while adhering strictly to research ethics, complete anonymity, and informed consent. Response frequency distributions, rounded to one decimal place, directly parameterize the core technical constraints of the BTCU matching engine in .SLOT: 61.2% of students lacking companions for spontaneous activities informs the definition of Elastic Time-Window Batching; 51.5% desiring short-term micro-connections to avoid relationship maintenance pressure establishes an ephemeral anonymous 1-on-1 interaction protocol; 97.7% supporting equitable bill splitting informs the integration of Symmetric Credit Scoring to penalize cancellations, mitigate bad behaviors, and prioritize reputable users; and 93.5% demanding public spaces for safety guides anchoring meeting coordinates to verified public stations via Dual LARS.

Current friend-matching platforms exhibit major bottlenecks in user experience (UX) (Holtzhausen et al., 2020; Liu, 2024; Thompson, 2025) and algorithmic architecture (Shen et al., 2022). Regarding UX, profile- and visual-data matching easily engenders appearance bias, false expectations, and benefit-seeking burdens (Peng, 2020; Peng et al., 2022). Algorithmically, the overreliance on greedy matching causes choice overload (Šiša, 2024) or unequal interactions (Narr & Luong, 2023), alongside high "ghosting" and cancellation rates (Šiša, 2024). Finally, unconstrained establishment of meeting points can place users in unsafe situations when meeting strangers (Lauckner et al., 2019).

To address these challenges, this study contributes the .SLOT architecture, adapting the dispatch model of ride-hailing networks to address interpersonal connection challenges via the BTCU (Batching - Time - Credit - Utility) multi-objective matching engine. Core contributions include:

(1) Formulating a real-time 1-on-1 interaction protocol with anonymous identification to optimize meeting rates and eliminate appearance bias.

(2) Applying Dual LARS to anchor meeting coordinates at verified public locations and optimize bilateral converging travel distances.

(3) Optimizing peer allocations on a general graph using Edmonds' Blossom algorithm (Edmonds, 1965), integrated with an anti-starvation mechanism, to empirically achieve a balanced multi-objective trade-off among liquidity, converging distance, and credit fairness.

(4) Validating the model's feasibility through simulation on a synthetic dataset of 1,000 agents within a university campus micro-zone, demonstrating competitive performance in credit equity and showing that micro-anchor spatial request decomposition maintains sub-second operational latency for localized batches.

2. RELATED WORK

This study addresses technical gaps at the intersection of three domains:

2.1. Recommendation and matching algorithms on social platforms

Current connection platforms use three main approaches: asynchronous matching based on collaborative filtering (e.g., Bumble BFF), online scheduling (e.g., Lunchclub), and synchronous group matching with K-vertex constraints (e.g., Timeleft, Meetup, After5). However, these architectures frequently cause choice overload (Šiša, 2024), high risks of group collapse (Narr & Luong, 2023) due to operational latency, and a lack of physical spatiotemporal constraints (Lauckner et al., 2019). Furthermore, mainstream platforms fail to eliminate appearance bias (Peng, 2020; Peng et al., 2022), failing to integrate user reliability and travel costs to optimize the actual meeting execution rate.

2.2. Location aware recommender systems (LARS)

In platforms like Bumble BFF or Happn, geographical factors primarily act as passive macroscopic radius filters. These distance-based greedy heuristics evaluate spatial proximity alone while ignoring user reliability, resulting in high credit variance, severe mismatching, and frequent ghosting. Lunchclub eliminates physical space. Meanwhile, Meetup, Timeleft, and After5 assign a pre-selected restaurant without considering travel costs. These methods cannot intervene early in dispatch.

Conversely, ride-hailing platforms (Grab, Uber, Didi, be, XanhSM…) solve the spatio-temporal problem more effectively by requiring users to

specify their destinations upfront. However, their dispatch logic is inherently designed for asymmetrical movement (driver to passenger). Ultimately, current social platforms rely on passive or unilateral LARS mechanisms (Levandoski et al., 2012), whereas ride-hailing frameworks only optimize asymmetric movement. The absence of bilateral converging distance optimization in social matching contexts necessitates an active architecture that intervenes during the dispatch phase, anchoring meeting coordinates to verified public stations and optimizing bilateral converging distances to ensure both participant safety and equitable travel effort. We formulate this approach as DUAL LARS.

2.3. Multi-objective matching algorithms

To allocate resources in real-time, ride-hailing platforms apply time-window batching mechanisms and solve allocations on Bipartite Graphs (Jiang & Varakantham, 2022; Liang, 2024; Lyu et al., 2024) or via linear pairwise weighted-sum heuristics. However, these linear weighted-sum formulations suffer from "weight swallowing", where spatial distance metrics overwhelm rating scores, and fail to isolate low-credit violators from high-reputation users. Recent literature has attempted to integrate equity into these models via fairness-aware bipartite matching (e.g., Liang, 2024; Lyu et al., 2024).

However, this architecture still optimizes asymmetrically; from a graph-theoretic perspective, applying a bipartite matching model to a homogeneous user pool requires enforcing an arbitrary partition $V = A \cup B$ which eliminates valid intra-partition edges, restricts the feasible solution space, and degrades overall matching liquidity. Although exact general graph matching can be solved via Edmonds' Blossom algorithm (Edmonds, 1965), its mathcal $O(V^3)$ worst case complexity poses real-time scalability challenges for dynamic urban batches. Therefore, transitioning to symmetric, multi-objective optimization on a General Graph to encode bilateral credit fairness, while mitigating the cubic time bottleneck, represents the core technical gap this study aims to fill.

3. SYSTEM ARCHITECTURE AND MATHEMATICAL DISPATCH MODEL

3.1. Overall architecture

Synthesizing the empirical survey insights from Section 1 with technical strategies to overcome the limitations of existing platforms outlined in Section 2, .SLOT is structured into a 3-layer architecture (Figure 1).

The Application layer receives 1-on-1 requests by stripping demographic and personal profile data to enforce user anonymity, while capturing user-declared shirt colors, real-time GPS coordinates, and credit scores C_i, with no profile images or gender data. To reduce shirt-color collision risks at crowded public anchors, this layer executes a dynamic 2-way natural challenge-response passphrase protocol, where matched peers verify each other through an app-generated prompt and reply exchange contextualized to the physical environment. For example, at a coffee shop, an agent can approach a person wearing the matching shirt color and ask, “Do you like milk coffee?”; if the other person replies, “I prefer cappuccino”, they successfully authenticate as a matched .SLOT pair.

The Coordination layer serves as the core matching engine, integrating the BTCU engine to process multi-objective logic: managing credit C_i, elastic pruning, Dual LARS spatial routing & ETA broadcasting, and maximum weight matching on a general graph via Edmonds algorithm (Edmonds, 1965) with a worst-case complexity of $O(V^3)$. Crucially, this layer incorporates specific mechanisms to address credit-gaming risks:

(1) Collusion and fake check-ins: Credit updates are counted only when bilateral geofence validation confirms both peers are physically present at the designated SafeSpot.

(2) Ghosting & False reporting: To prevent an agent from no-showing while falsely accusing their peer to evade penalties, the system resolves disputes using geofenced arrival timestamps at the SafeSpot rather than manual self-reports. Upon log-based verification, the violator’s credit score C_i is penalized, which reduces their multiplicative weight $(C_i \times C_j)$ and deprioritizes them in subsequent matching cycles.

(3) Sybil Attacks: Newly re-registered accounts reset to the baseline score, yielding no competitive priority over reputable users. Achieving zero-cost Sybil resistance without invasive identity verification is acknowledged as an intrinsic trade-off of lightweight anonymous protocols.

The Data layer manages the safe anchor spot database, credit history, and the Tau τ waiting queue to store requests and prevent agent starvation.

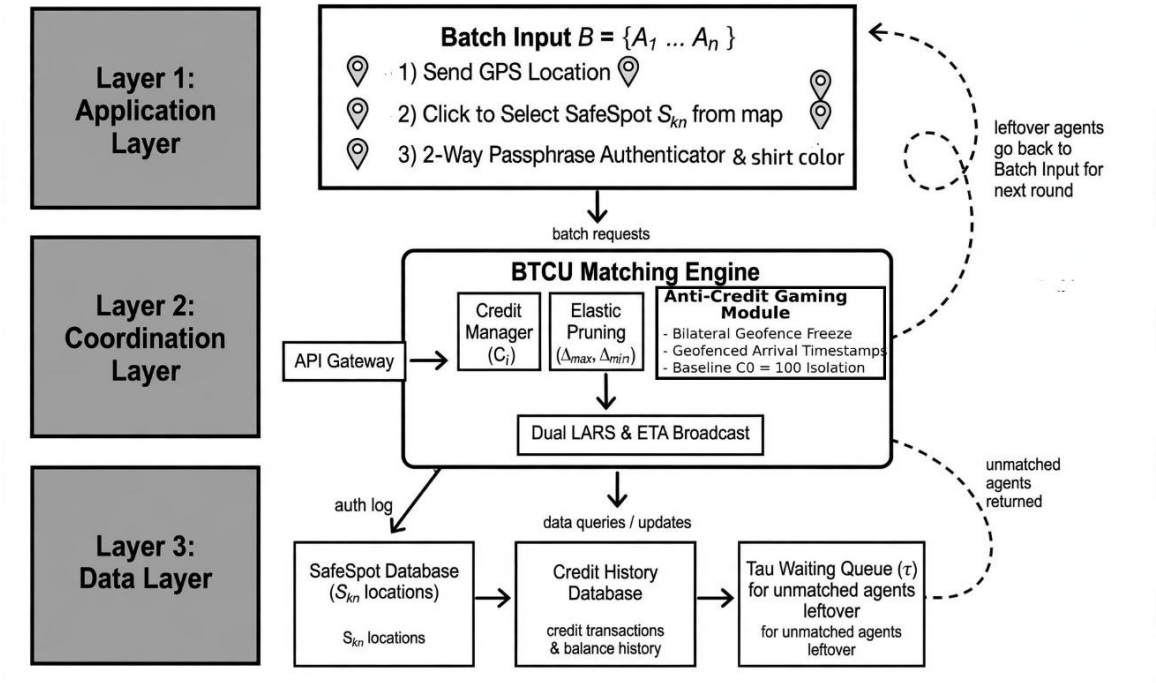


Figure 1. Overview of the 3-layer architecture of the .SLOT system

3.2. Modeling

The real-time micro-connection process operates across the Application, Coordination, and Data layers to connect symmetric 1-on-1 agents. Unlike traditional ride-hailing networks operating on naturally bipartite driver-passenger structures, peer-to-peer social connections involve symmetric agents where any two available individuals sharing a compatible domain can form a valid pair.

Building on the structural limitations identified in Section 2.SLOT formulates matching as a global maximum-weight matching problem on an edge-weighted general graph. An optimal match is formally defined as a global edge-disjoint matching set that maximizes collective system utility across an entire batching window, evaluating all feasible bilateral connections simultaneously.

The cross-layer execution sequence and state transition flow are illustrated in Figure 2 through four operational phases:

(1) Phase 1: Request ingestion and vector transmission. The Application layer strips demographic data, captures the user-declared shirt color alongside a dynamic challenge-response passphrase prompt for physical on-site identification, and transmits the anonymized state vector to the Coordination layer.

(2) Phase 2: BTCU engine optimization. The Coordination layer constructs an edge-weighted general graph and executes Edmonds' Blossom algorithm alongside Dual LARS spatial routing to determine the optimal match set.

(3) Phase 3: Field settlement and anti-gaming verification. Matched peers receive spatial ETAs and proceed to the designated SafeSpot. Upon bilateral geofence validation, shirt color and successful passphrase exchange, credit rewards are unlocked and committed to the Data layer. In cases of disputes or ghosting, the system audits geofenced arrival timestamps to penalize the violator, reducing their matching weight in subsequent cycles.

(4) Phase 4: Queue retention and anti starvation. Unmatched agents are not dropped; instead, they are retained in the τ queue within the Data layer across dynamic batching windows ($\Delta t \approx 30s - 3\ min$), incrementing their retry counter to elevate their priority in the subsequent batching cycle. To prevent indefinite waiting, retention is capped at τ_{max} = 3 cycles (total wait time ≈ 2 - 5min), after which a graceful timeout triggers alongside a 50 Mood Points compensation for the next request.

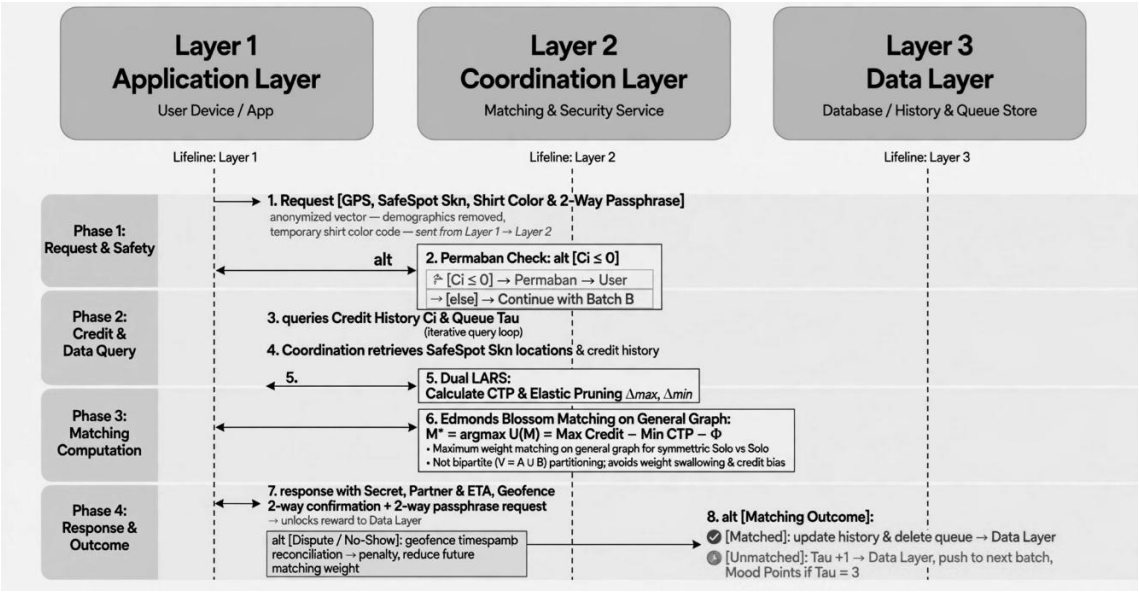


Figure 2. Sequence diagram of .SLOT system.

(Notes: $\boldsymbol{C_i}$: credit score for user ; $\boldsymbol{\theta}$: penalty term in utility; u τ: retry counter; U(M): matching utility).

3.3. Mathematical algorithms and conflict resolution mechanisms

Let S be a finite set of verified public meeting points. S is partitioned into 5 disjoint domains (Eat&Drink, Study, Chat, Walk, Sport): : $S = S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5$. Each connection request goes through 2 stages: selecting a domain S_k and a specific anchor point $S_{i,k_n} \in S_k$. After booking, the request is encoded into a state vector: $A_i = [loc_i, S_{i,k_n}, C_i]$. Where: loc_i is the anonymous coordinate of user i at the time of issuing the search command, C_i is the symmetric credit score (initialized at $C_i^{(0)} = 100$). The credit score is progressively updated upon cancellation with $C_i \leftarrow C_i - p_k$ where $p_k = \{20, 30, 50\}$, if $C_i = 0 \rightarrow Permanent\ ban$ to protect network liquidity.

The system does not process matching immediately but batches requests with the same anchor point into a batch (B) at start time (T):

$$B(S_{k_n}, T) * = \{ A_i \mid i \in \{1,2, \dots N\}, S_{i,k_n} = S_{k_n}, t(A_i) \in [T,\ T + \Delta t] \} \quad (1)$$

With the time window $\Delta t = \rho / |G(S_{k_n})| + \epsilon'$ being elastic according to regional density. Here, the DUAL LARS (Dual Location Aware Recommender System) extends conventional location-aware recommendation paradigms by shifting from passive, unilateral proximity filtering to an active, bilateral spatial routing framework. Rather than evaluating single-sided search radii or asymmetric provider-to-consumer movement, Dual LARS simultaneously evaluates the converging physical trajectories of both peers toward a verified public anchor station (S_{k_n}). Mathematically, Dual LARS operationalizes this spatial coordination via the Converging Travel Penalty (CTP), which combines aggregate physical displacement $\alpha\left(d(A_i, S_{k_n}) + d(A_j, S_{k_n})\right)$ and effort disparity $\beta|d(A_i, S_{k_n}) - d(A_j, S_{k_n})|$. This composite formulation is structurally necessary to balance macroscopic efficiency with microscopic equity. Minimizing aggregate distance alone $\alpha(d(A_i, S_{k_n}))$ is prone to severe effort asymmetry, such as pairing a nearby user with a distant counterpart, which causes high cancellation rates due to perceived unfairness. Conversely, relying solely on distance disparity $\beta|d(A_i, S_{k_n}) - d(A_j, S_{k_n})|$eliminates effort inequality but risks absurd dispatch decisions by favoring distant yet equally positioned pairs over significantly closer candidates. By jointly coupling aggregate displacement and effort disparity, Dual LARS forces the matching engine to identify candidate pairs that optimize total travel expenditure while enforcing balanced, converging movement toward the anchor station. This recommender system calculates Converging Travel Penalty (CTP) for both sides:

$$CTP(A_i, A_j, S_{k_n}) = \alpha(d(A_i, S_{k_n}) + d(A_j, S_{k_n})) + \beta|d(A_i, S_{k_n}) - d(A_j, S_{k_n})| \quad (2)$$

Where the positive hyperparameters established by the system are α to optimize the total distance traveled, and β to set the penalty for distance inequality. The Estimated Time of Arrival $ETA_i = d(A_i, S_{k_n})/v_{walk}$ is broadcast in real-time to the newly established optimal pair.

The total distance penalty and total credit (C) of any matching set M are defined as:

$$CTP_{total}(M) = \sum_{(A_i,A_j)\in M}[\alpha\left(d(A_i, S_{k_n}) + d(A_j, S_{k_n})\right) + \beta|d(A_i, S_{k_n}) - d(A_j, S_{k_n})|] \quad (3)$$

$$Credit_{total}(M) = \sum_{(A_i,A_j)\in M} \omega \cdot (C_i \times C_j) \quad (4)$$

To regulate peer reliability within the edge weight formulation, the credit interaction between candidate agents A_i and A_j is modeled as an explicit scalar product $C_i \times C_j$ rather than a linear additive sum $C_i + C_j$ or a vector dot product $C_i \cdot C_j$ across their full state vectors $A_i \cdot A_j$. Structurally, a linear additive scheme $C_i + C_j$ allows a high-reputation participant C_i= 100 paired with a low-credit violator C_j = 20 to yield a high composite score of 120, enabling unreliable actors to "piggyback" on reputable users and exposing high-credit individuals to frequent cancellations. Conversely, attempting a vector dot product across full state vectors $A_i = [loc_i, S_{i,k_n}, C_i]$ and $A_j = [loc_j, S_{j,k_n}, C_j]$ would introduce severe dimensional inconsistency by improperly multiplying physical spatial coordinates (measured in meters) with non-physical credit units. By employing an explicit scalar product $C_i \times C_j$, the formulation preserves dimensional purity while acting as a non-linear credit-isolation gate. This multiplicative penalty heavily suppresses matches involving low-credit agents ($100 \times 20 = 2{,}000$) compared to mutually reliable pairs ($80 \times 80 = 6{,}400$), forcing Edmonds' Blossom algorithm to segregate unreliable users into lower priority matching sub-graphs.

Here, ω serves as a scale normalization coefficient that bridges non-physical credit products with physical distance costs (measured in meters), preventing the "weight swallowing" phenomenon and enabling balanced multi-objective trade-offs.

To perform batch matching, candidate interactions are modeled as a general non-bipartite graph G = (V, E), where any arbitrary agent A_i can pair with A_j. The system then applies Edmonds' Blossom algorithm to identify the optimal matching set U that maximizes the global aggregate utility:

$$U = argmax_M \sum_{(A_i,A_j)\in M}\{\omega \cdot [C_i(1+\gamma)^{\tau_i} \times C_j(1+\gamma)^{\tau_j}] - CTP(A_i, A_j) - \theta(A_i, A_j)\} \quad (5)$$

Structurally, Edmonds' Blossom algorithm is selected because peer-to-peer micro-connection topologies form general non-bipartite graphs that frequently generate odd-length cycles, causing traditional bipartite solvers to fail. While exact general-graph matching carries an $O(V^3)$ computational complexity, .SLOT theoretically mitigates this scaling bottleneck by spatially decomposing requests into localized micro-anchor sub-graphs, effectively capping the vertex set V per matching instance.

Because Equation (5) explicitly unifies credit reputation, spatial friction, and queue priority into a composite objective, the resulting assignment U is inherently Pareto optimal for each localized batch: no single objective can be further improved for any pair without decreasing overall system utility.

To operationalize the multi-objective optimization engine, the hyperparameters (α, β, ω, $\Delta_{min}, \Delta_{max}, \gamma$) are calibrated based on domain constraints and dimensional alignment. Specifically, ω = 2.0 aligns credit scale with metric spatial costs. The spatial weights are configured with $\beta = 2.0 > \alpha = 1.0$ to prioritize bilateral effort equity over aggregate distance minimization. The priority amplification rate γ = 0.1 delivers a controlled 10% exponential priority boost per retry cycle, while the dynamic batching interval $[\Delta_{min} = 30s, \Delta_{max} = 180s]$ balances spatial request aggregation against maximum user latency tolerance. Empirical sensitivity analysis evaluating system behavior under varying parameter configurations is presented in Section 4.1.

To overcome spatial inertia faster than linear growth and prevent queue starvation, the waiting compensation coefficient γ amplifies the credit power based on the number of dropped batch cycles τ. If τ≥3, a safe disconnect is triggered and compensated with 50 Mood Points (which serve as in-app top-tier priority tokens for future matchings) capping weight inflation while preserving user retention.

Furthermore, as integrated in Equation (5), the infinity penalty function ($\theta(A_i, A_j$ automatically prunes violating edges as follows:

$$\theta((A_i, A_j) = \begin{cases} +\infty \ if \ \left|d(A_i, S_{k_n}) - d(A_j, S_{k_n})\right| \geq \Delta_{max} \\ \quad (Distance\ violation) \\ 0 \ if \ valid \end{cases}$$

3.4. Pseudo – code

Algorithm 1: BTCU Multi-Objective Matching via Edmonds' Blossom

Input: Batch B={A1...AN} at S_{k_n} , $d_i = d(A_i,S_{k_n})$, $d_j = d(A_j,S_{k_n})$ C_i, C_j (Credit, penalty $p_k \in \{20,30,50\}$) τ_i, τ_j, hyperparams $\omega,\alpha,\beta,\gamma,\Delta max,\Delta min, \tau_{max}$

Output: Matched pairs M, updated statuses, real-time ETA

1: $G=(V,E) \leftarrow \emptyset$; $M \leftarrow \emptyset$

2: foreach $A_i \in B$ do

3: if A_i canceled recently then $C_i \leftarrow C_i - p_k$, $p_k \in \{20, 30,50\}$

4: if $C_i = 0$ then Status(A_i) ← Permaban

5: else if $\tau_i \geq \tau_{max}$ then Status(A_i) ← Graceful Timeout, MPi ← MPi + 50

6: else $V \leftarrow V \cup \{A_i\}$

7: foreach $(A_i,A_j) \in V \times V$, $i \neq j$ do

8: if $|d_i-d_j| \geq \Delta max$ or ($|d_i-d_j| \geq \Delta min$ and $max(d_i,d_j) \geq 2 \cdot min(d_i,d_j)$) then continue // Elastic Pruning

9: $U_{i,j} \leftarrow \omega[C_i(1+\gamma)^{\tau_i} \times C_j(1+\gamma)^{\tau_j}] - [\alpha(d_i+d_j)+\beta|d_i-d_j|]$

10: if $U_{i,j} > 0$ then $E \leftarrow E \cup \{(A_i,A_j,U_{i,j})\}$

11: $M \leftarrow$ Edmonds_Maximum_Weight_Matching(G)

12: foreach $(A_u,A_v) \in M$ do

13: $\tau_u,\tau_v \leftarrow 0$; Broadcast ETA to A_u,A_v

14: foreach $A_i \in V \backslash V(M)$ do

15: $\tau_i \leftarrow \tau_i + 1$

16: return M, Status

4. SIMULATION TESTING BED

4.1. Experimental setup and dataset

The .SLOT system is simulated using Python, utilizing the NetworkX library to solve the Edmonds' Blossom maximum matching problem on a general graph. To ensure experimental reproducibility, all synthetic data generators and stochastic processes are bound to a fixed pseudo-random state (seed = 42). Based on user behavior from the survey in Section 1, a set of hyperparameters is established to balance multiple objectives, as summarized in Table 1:

Table 1. Configuration of hyperparameters for the BTCU matching engine

Parameter	Symbol	Value	Meaning
Credit weight	ω	2.0	Amplifies utility when matching reputable users
Converging distance penalty	α, β	1.0, 2.0	Controls total distance (α) and fairness deviation (β)
Activation threshold	Δ_{min}	400m	Minimum distance deviation threshold to trigger the 2x inequality filter
Elastic pruning	Δ_{max}	800m	Maximum physical distance deviation limit threshold
Starvation control	τ_{max}, γ	3, 0.1	Maximum waiting and credit amplification coefficient

The simulation is grounded in single-university pilot data: 4,926 freshmen per academic year (≈18,000 active students across 4 cohorts). We set N=1,000 concurrent agents to reflect a realistic 5.5% peak concurrency (1,000/18,000) during peak hours, based on our prior needs assessment. Distributed across 31 on-campus SafeSpot anchors, this yields an average batch density of $|V| \approx 32$ agents per anchor, divided into 6 time batches (T1-T6). The spatial distribution is strictly established: 60% of requests are in the core zone (<500m), 30% in the buffer zone (500-1000m) and 10% in the peripheral zone. Regarding credit, 85% of agents possess absolute reputation scores ($C_i = 100$), while 15% are intentionally configured as malicious accounts (e.g., frequent cancelers causing $C_i = 0$) to stress-test the system's robustness and its ability to purge bad actors.

Regarding parameter sensitivity, as designated in Section 3, an empirical sensitivity analysis was performed by varying core hyperparameters (α, β, ω, $\Delta_{min}, \Delta_{max}, \gamma$). Table 2 details system behavior across matching liquidity, distance equity, and credit fairness under peak load (N=1,000).

Table 2. Hyperparameter sensitivity evaluation of the .SLOT engine.

Hyperparameter	Evaluated Range	Calibrated Value	Matching Rate (%)	Distance Dev. (m)	Credit Dev.	Operational Impact & Optimal Trade-off
Spatial Weight (α)	Baseline	1.0	—	—	—	Normalized Unit: Serves as the fixed spatial unit reference.
Equity Weight (β)	0.0 → 4.0	2.0	98.2% → 94.2%	8.5 → 13.8	0.85 → 0.52	β = 2.0 eliminates walking disparity without causing over-pruning.
Credit Scale (ω)	0.5 → 5.0	2.0	97.3% → 96.7%	8.9 → 12.5	0.90 → 0.28	ω = 2.0 balances credit fairness against physical walking cost.
Equity Min (Δmin)	200 → 600m	400m	98.1% → 96.5%	9.1 → 11.2	0.62 → 0.49	Δmin = 400m defines the tolerance threshold for walking imbalance.
Pruning Bound (Δmax)	400 → 1200m	800m	88.2% → 98.5%	6.2 → 14.2	0.92 → 0.48	Δmax = 800m maintains liquidity (97.9%) within micro-walkability limits.
Starvation Boost (γ)	0.0 → 0.3	0.1	97.1% → 97.8%	9.8 → 10.9	0.88 → 0.61	γ = 0.1 delivers a 10% priority boost per cycle to prevent starvation.

As detailed in Table 2, α=1.0 serves as the normalized baseline spatial unit. The empirical evaluation confirms that the calibrated vector α=1.0, β=2.0, ω=2.0, $\Delta_{min}= 400m, \Delta_{max}= 800m, \gamma = 0.1$ achieves an optimal Pareto trade-off, maximizing matching liquidity (97.9%) while preserving strict physical walkability and credit fairness.

Theorem (Load invariant scalability): Let k=|V|=N/S be the local density. Under proportional scaling $S \propto N, k \approx 32$ is constant, hence per-anchor matching time $T_{anchor} = O(k^3) = O(1)$ Total latency $T_{total} = \max_{i \in S} T_{anchor}$ is therefore independent of global N, guaranteeing identical performance at N=1,000 and N=10,000.

4.2. Case studies

Instead of merely evaluating the surface-level successful matching rate, the simulation environment actively injects four edge-case agents into the system. The objective is to verify the robustness and accuracy of the internal conflict resolution mechanisms of the BTCU engine. As detailed in Section 3.1, physical-layer identification and location verification are strictly enforced via two-way passphrase matching and GPS geofencing. Table 3, which focuses on evaluating internal algorithmic edge-case resolutions within the matching engine, details the system's reaction to these algorithmic edge-case scenarios:

Table 3. Evaluation of the system's reaction to edge-case scenarios

Conflict Type	Simulated Agent	Input State	System's Resolution Mechanism
Malicious	*Frequent Canceler (User: Vua bùng kèo)*	$C_i = 0$	The Credit Manager module detects and triggers a permanent ban prior to batching.
Spatial	*Out-of-bounds User (User: Trọ ngoài Linh Xuân)*	d =2800m	The elastic filter prunes the edge due to the distance deviation exceeding the safe threshold Δ_{max} (800m).
Low credit	*Low-credit User (User: Chúa tể đi trễ)*	$C_i = 50$	Directly downweighted in the objective function, losing competitive advantage and continuously dropping from batches.
Starvation	*Starved User (User: Boy FA xui xẻo)*	Continuous waiting $\tau = 3$	Triggers a graceful disconnect and immediately compensates with 50 Mood Points.

4.3. Overall performance evaluation and baseline comparison

To evaluate system efficiency, matching quality, and scalability, the .SLOT matching engine was empirically benchmarked across dynamic real-world timeframes and compared against matching paradigms under operational load N = 1,000 agents.

4.3.1. Time frame performance and operational liquidity

Table 4 presents the operational performance of the .SLOT engine across six discrete daily time windows (T1 → T6). By integrating Micro-Anchor Spatial Decomposition with an elastic time margin Δt, the system maintains dynamic liquidity while keeping execution latency strictly within real-time limits.

Table 4. Overview of the BTCU system's matching performance across dynamic timeframes.

Time Window	Load	Acc Bans (per time)	Edge Pruning (θ)	Matched Pairs	Δt Variation (seconds)	Execution Time	Success Rate
T1 (06:00 - 08:30)	49	2	115	22	[90.6 - 182.7]	10.54 ms	89.8%
T2 (08:30 - 11:30)	96	5	523	47	[85.1 - 92.9]	71.55 ms	97.9%
T3 (11:30 - 13:30)	316	14	4099	**155**	**[31.3 - 34.4]**	987.03 ms	98.1%
T4 (13:30 - 17:00)	121	5	927	58	[35.2 - 90.6]	96.48 ms	95.9%
T5 (17:00 - 19:30)	241	10	3497	**118**	**[33.2 - 34.8]**	470.69 ms	97.9%
T6 (19:30 - 23:00)	141	10	1277	66	[32.7 - 89.9]	118.90 ms	93.6%
Total/ Overall	964	46	10,438	466	[31.3 – 182.7]	1,755.19 ms	96.7%

Across all timeframes, a total of 46 malicious accounts were purged prior to batching, leaving 964 valid requests (1,000 total initial agents). The engine achieved an overall success rate of 96.7% matched pairs. Even at peak midday load, T3 = 316 concurrent agents in batch; maximum single-batch execution time remained under 1 second (987.03 ms), validating the real-time feasibility of the system.

4.3.2. Comparative Baseline Evaluation

Traditional ride-hailing platforms utilize Bipartite Graphs, which enforce an asymmetric dispatch structure (e.g., driver vs. passenger). However, peer-to-peer micro-connections require a symmetric General Graph model. To justify our architectural design, .SLOT is benchmarked against three baseline algorithms in Table 5.

Distance-Based Greedy (DG): A general-graph spatial baseline prioritizing physical proximity by greedily selecting the closest agent pairs via $agr\min\{d_i + d_j\}\, subject\ to\ |d_i - d_j| < \Delta_{max}$, completely ignoring credit scores.

Weighted-Sum Greedy (WM): A general-graph heuristic maximizing a weighted sum of distance and credit score via $W_{ij} = \omega(C_i C_j) - [\alpha(d_i + d_j) + \beta\,|d_i - d_j|]$

Fairness-aware Bipartite Framework (F-BF): A dispatching model adapted from ride-hailing platforms, bisecting agents into disjoint subsets ($|V_1|=|V_2|=N/2$) and solving bipartite MCMF matching $\sum W_{ij}x_{ij}$, strictly pruning all intra-partition edges $|(V_1 \times V_1) \cup (V_2 \times V_2)$

BTCU marginal slight liquidity for near-absolute credit fairness (Table 5). Moreover, Edmonds' O(V3) complexity requires ~ 11.8s for 1,000 agents, risking delays at scale.

Table 5. Comparison of matching quality between the BTCU engine and baseline algorithms in the global batch

Algorithm	Matching Rate	Distance Deviation	Credit Deviation	Execution Time (ms)
Distance-Based Greedy (DG)	99.7%	9.0m	13.0	39.45 - 54.57 ms
Weighted-Sum Greedy (WM)	99.7%	9.3m	3.8	41.19 - 63.29 ms
Fairness Bipartite Framework (F-BF)	95.4%	12.8m	2.4	85.20 – 112.40 ms
BTCU system(.SLOT) monolithic Edmonds	97.9%	10.1m	0.9	11,447.91 - 11,899.03 ms
BTCU system(.SLOT) decomposed proposed	97.9%	10.1m	0.9	< 987.03 ms (peak T3)

To ensure scalability, BTCU spatially decomposes requests into micro-anchor stations (Table 4), maintaining execution time under 1 second even at peak loads. This helps the system maintain mathematical accuracy while fully meeting real-time scalability requirements, as evidenced in Table 4.

As illustrated in Figure 3, .SLOT operates directly on the non-dominated Pareto frontier. By setting $\omega = 2.0$, it intentionally trades a marginal 1.8% in matching liquidity (97.9% vs. 99.7%) to improve credit fairness from 15.0% to 96.1% (reducing credit deviation from 13 to 0.9), reducing reputation disparity by 93.1%.

Furthermore, .SLOT strictly Pareto-dominates F-BF (95.4%, 81.5%), while the decomposed implementation reduces execution latency from 11,899 ms to < 987 ms while preserving Pareto efficiency (Figure 3).

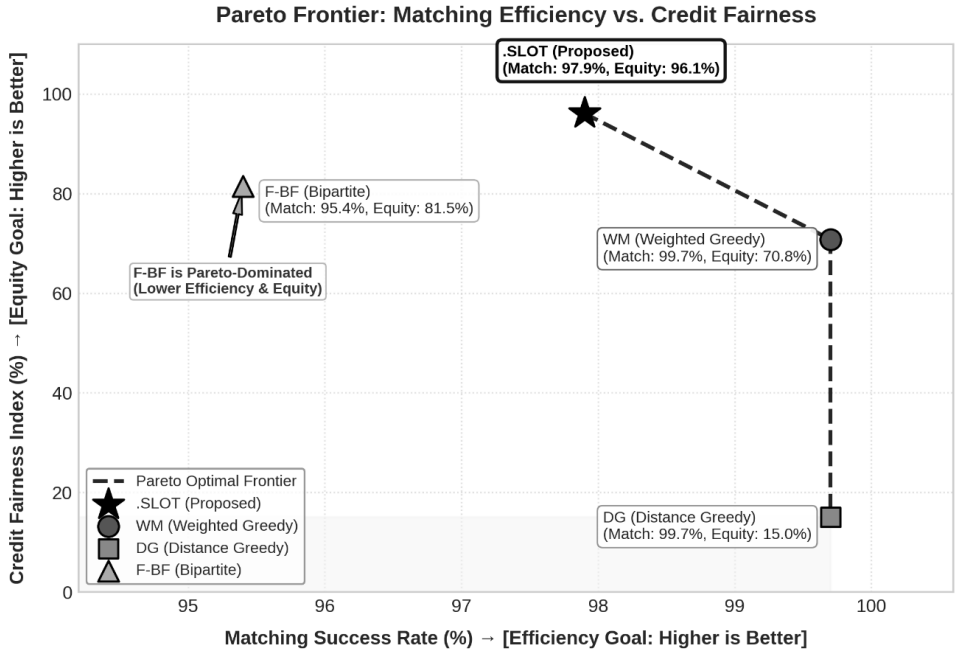


Figure 3. Pareto frontier analysis demonstrating the optimal multi-objective trade-off of .SLOT in balancing matching efficiency and credit equity against baseline models (WM, DG, and F-BF)

4.3.3. Reconciliation of execution latency and scalability

Running exact Edmonds' Blossom matching on a single monolithic global graph (N = 1,000) incurs $O(V^3)$ complexity, resulting in a processing latency of ~11.5 seconds ($\approx 11{,}447 - 11{,}899ms$). This would introduce prohibitive delays in real-time urban settings.

By deploying micro-anchor spatial decomposition (Table 4), the global graph is partitioned into localized subgraphs processed concurrently. As proven in Section 4.1 (Theorem: Load-Invariant Scalability), this bounds local subgraph density at k = |V| ≈ 32, successfully compressing peak execution time to under 1 second (< 987.03 ms) without degrading global optimization quality.

5. CONCLUSION

In conclusion, the .SLOT architecture and the BTCU multi-objective matching engine have demonstrated effectiveness by empirically achieving a Pareto trade-off among liquidity rate, converging distance, and credit score fairness. The core value of the system lies in thoroughly solving the human connection problem through face-to-face encounters (1-on-1 micro-connections). This fosters authentic social interactions, serving as a practical technological intervention against the "urban loneliness" syndrome.

Regarding the study's limitations and ethical considerations, this research currently relies on a synthetic dataset, generated within a single university campus micro-zone, highlighting the

need for real-world pilot testing across broader megacity environments in future work.

Furthermore, to strictly eliminate appearance bias, the system intentionally discards demographic data (e.g., gender, age), relying entirely on physical public spaces (Dual LARS) to enhance user safety and privacy; however, this lightweight anonymous protocol inherently trade-offs user preference filtering and absolute Sybil-attack immunity without invasive KYC verification. While potential physical shirt-color collisions at crowded anchors remain an edge case requiring further computer-vision interventions, the algorithm's primary mathematical constraint is that it operates solely on a symmetric 1-on-1 graph. Scaling to larger groups faces combinatorial explosion and bottlenecks from algorithmic complexity.

Therefore, the future map of this research will explore graph partitioning techniques or heuristic pruning prior to applying the Edmonds' algorithm to handle matching problems on hypergraphs for groups of 3 people, aiming to maintain sub-second scalability in denser scenarios.

CONFLICT OF INTEREST

The author declares that there are no financial or personal relationships that could be perceived as inappropriately biasing the work reported in this manuscript.

ACKNOWLEDGMENT

The study was conducted under the support of the University of Sciences of Ho Chi Minh City and Viet Nam National University of Ho Chi Minh City. I would like to extend our sincere gratitude to the colleagues and students who supported-me.

REFERENCES

Bumble. (2023). *Bumble for Friends data reveals friendship and loneliness trends online*. Bumble Buzz, Survey conducted by Censuswide. https://bumble.com/the-buzz/bumble-for-friends-bff-data-friendship-loneliness-online

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340. https://doi.org/10.2307/249008

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics, 17*, 449–467. https://doi.org/10.4153/CJM-1965-045-4

Hammoud, R., Tognin, S., Bakolis, I., Morgan, C., & Mechelli, A. (2021). Lonely in a crowd: Investigating the association between overcrowding and loneliness using smartphone technologies. *Scientific Reports*, *11*(1), Article 24134. https://doi.org/10.1038/s41598-021-03398-2

Holtzhausen, N., Fitzgerald, K., Thakur, I., Ashley, J., Symons, M., & Pit, S. W. (2020). Swipe-based dating applications use and its association with mental health outcomes: A cross-sectional study. *BMC Psychology, 8*(1), Article 22. https://doi.org/10.1186/s40359-020-0373-1

Jiang, H., & Varakantham, P. (2022). Hierarchical value decomposition for effective on-demand ride-pooling. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2022) (pp. 660–668). IFAAMAS.

Lauckner, C., Truszczynski, N., Lambert, D., Kottamasu, V., Meherally, S., Schipani-McLaughlin, A. M., & Hansen, N. (2019). "Catfishing," cyberbullying, and coercion: An exploration of the risks associated with dating app use among rural sexual minority males. *Journal of Gay & Lesbian Mental Health, 23*(3), 289–306. https://doi.org/10.1080/19359705.2019.1587729

Lee, S. B., Shin, Y. R., Jeon, Y. B., & Kim, S. H. (2022). Factors affecting social isolation among the young adults in South Korea: A cross-sectional analysis. *Frontiers in Public Health, 10*, Article 979136. https://doi.org/10.3389/fpubh.2022.979136

Levandoski, J. J., Sarwat, M., Eldawy, A., & Mokbel, M. F. (2012). *LARS: A location-aware recommender system. In Proceedings of the 20th IEEE International Conference on Advances in Geographic Information Systems* (SIGSPATIAL '12) (pp. 450–453). IEEE.

Liang, Y. (2024). Fairness-aware dynamic ride-hailing matching. *Electronics, 13*(4), Article 775. https://doi.org/10.3390/electronics13040775

Liu, L. (2024). Research on the relationship between virtual social interaction and the degree of loneliness based on algorithm matching technologies: A quantitative analysis on the SOUL APP—A virtual social software for strangers. *PLOS ONE, 19*(12), Article e0312522. https://doi.org/10.1371/journal.pone.0312522

Lyu, G., Cheung, W. C., Teo, C.-P., & Wang, H. (2024). Multiobjective stochastic optimization: A case of real-time matching in ride-sourcing markets. *Manufacturing & Service Operations Management, 26*(2), 500–518. https://doi.org/10.1287/msom.2020.0247

Narr, G., & Luong, A. (2023). Bored ghosts in the dating app assemblage: How dating app algorithms couple ghosting behaviors with a mood of boredom. *The*

Communication Review, 26(1), 1–23. https://doi.org/10.1080/10714421.2022.2129949

Peng, K. (2020). To be attractive or to be authentic? How two competing motivations influence self-presentation in online dating. *Internet Research, 30*(4), 1143–1165. doi: https://doi.org/10.1108/INTR-03-2019-0095

Peng, K., Lin, W.-Y., & Chen, H. (2022). Consequences of deceptive self-presentation in online dating. *Chinese Journal of Communication, 15*(4), 582–610. https://doi.org/10.1080/17544750.2022.2052130

Russell, D., Peplau, L. A., & Cutrona, C. E. (1980). The revised UCLA Loneliness Scale: Concurrent and discriminant validity evidence. *Journal of Personality and Social Psychology, 39*(3), 472–480. https://doi.org/10.1037/0022-3514.39.3.472

Shen, C. Y., Yang, D. N., Lee, W. C., & Chen, M. S. (2022). Activity Organization for Friend-Making Optimization in Online Social Networks. *IEEE Transactions on Knowledge and Data Engineering, 34*(1), 122-137. https://doi.org/10.1109/TKDE.2020.2980516

Šiša, A. (2024). Ghosting on Tinder: Examining disconnectivity in online dating. *Media and Communication, 12*, Article 8563. https://doi.org/10.17645/mac.8563

Thompson, R. (2025). Bumble's ticking clock: Dating app temporal design as neoliberal discipline. *Discourse, Context & Media, 66*, Article 100897. https://doi.org/10.1016/j.dcm.2025.100897

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly, 27*(3), 425–478. https://doi.org/10.2307/30036540