Proposing a Hybrid Role-Aware Team Balancing Algorithm for Skill- and Composition-Fair Matchmaking in Competitive Multiplayer Games

Anusha Jagtap, Akash Vishwakarma, Abhishek Bidwe, Hitesh Sheregar
Department of Computer Applications
D.Y. PATIL INSTITUTE OF MASTERS OF COMPUTER APPLICATION AND MANAGEMENT, Savitribai Phule Pune University
Email: akashxjr10vishwakarma@gmail.com

Prof. Rahul Chaudhari (Guide)
Department of Computer Applications
D.Y. PATIL INSTITUTE OF MASTERS OF COMPUTER APPLICATION AND MANAGEMENT, Savitribai Phule Pune University

***Abstract*—Matchmaking in competitive multiplayer games is driven almost entirely by skill rating. Systems such as Elo, Glicko and TrueSkill estimate individual strength well but ignore role composition, so two teams of equal average rating can be assembled very differently and the match is decided before play begins. Fairness-aware team-formation algorithms reported in the educational and human-resource literature address composition, but their combinatorial search cost is unsuitable for live queues that must form teams within milliseconds. This paper proposes the Hybrid Role-Aware Team Balancing Algorithm (HRTBA), a four-stage deterministic pipeline that combines player-declared role queues, skill-sorted role buckets, balanced team assembly, and a bounded same-role skill swap. HRTBA guarantees full role coverage by construction while confining optimisation to role-preserving exchanges, giving an overall cost of $O(n \log n)$—the same order as the sort a rating-based system already performs. HRTBA is evaluated against Random, Skill-only and Greedy baselines on simulated pools of 20 to 500 players with thirty randomised trials per configuration. Results show that HRTBA raises mean role coverage from roughly 68–76% (baselines) to 92–100%, while holding the mean cross-team skill gap within a small margin of the skill-only optimum (7.7 versus 4.9 rating points at pool size 500) and forming all teams in under 1.4 ms even for the largest pool. HRTBA therefore delivers composition-fair matchmaking inside the latency budget of a live queue, without new hardware.**

***Index Terms*—matchmaking, skill rating, role composition, team balancing, fairness, real-time systems, multi-objective optimisation.**

I. Introduction

Competitive multiplayer games such as tactical shooters and battle royales match millions of players into teams every minute, and the matchmaking service is arguably the most influential component of the player experience: a well-formed match produces a competitive, satisfying contest, whereas a poorly formed one produces a one-sided result that drives players away. Traditionally, matchmaking has relied almost exclusively on a numerical skill rating. From the Elo system devised for chess [1] to modern Bayesian methods such as Glicko [2] and TrueSkill [4], the objective has been to estimate how strong a player is and to place players of comparable estimated strength together.

This approach treats team formation as the problem of balancing a single scalar quantity. In modern multiplayer titles, however, a player's contribution depends not only on how skilful they are but also on the role they play. Tactical shooters such as *VALORANT* organise play around roles such as Controller, Initiator, Duelist and Sentinel, while battle royales distinguish Assaulters, Snipers, Supports and the In-Game Leader. Each role supplies capabilities that the others lack: a team without a Controller cannot control space, and a team without an Initiator cannot reliably open engagements. Consequently, two teams can be perfectly matched on average rating and yet be profoundly unequal in what they can actually do.

Rating answers the question "how good is this player?" but not "what does this player do?". When rating is the only signal, a team of three duelists and no controller is just as "balanced" as a rounded composition, even though the first team is far more likely to lose. This gap between numeric balance and compositional balance is the problem addressed in this paper.

The algorithms that do account for composition are unsuitable for live queues. Approaches reported in the literature—two-stage optimisation over team membership, genetic-algorithm search, and graph-based densest-subgraph methods—are designed for offline contexts such as forming study groups, project teams or fantasy-league squads, where additional computation time is acceptable [12], [13], [15]. In a matchmaking queue, teams must be formed within milliseconds, and any algorithm whose cost grows quickly with the player pool cannot be deployed.

There is therefore a gap between two bodies of work: skill-rating research, which is fast but composition-blind, and composition-aware team-formation research, which is fair but slow. Formally stated, the problem addressed by this paper is:

given a pool of players, each with a skill rating and a preferred role, form opposing teams that are balanced on both skill and role composition, without introducing latency that would be unacceptable in a live matchmaking queue.

The contributions of this paper are:

1) HRTBA, a four-stage deterministic pipeline that guarantees full role coverage by construction and improves skill balance with a bounded, role-preserving same-role swap (Section III).
2) A complexity analysis showing that HRTBA runs in $O(n \log n)$ time and $O(n)$ space, matching the sort that a rating-only system already performs (Section III-E).
3) A reproducible simulation study comparing HRTBA against Random, Skill-only and Greedy baselines across five pool sizes and thirty randomised trials, on four metrics: role coverage, cross-team skill gap, formation time and consistency (Sections IV–V).

The remainder of the paper is organised as follows. Section II reviews related work. Section III presents the proposed algorithm and its mathematical formulation. Section IV describes the experimental setup, Section V reports and discusses the results, and Section VI concludes the paper.

II. Related Work

A. Skill Rating Systems

The Elo system [1] introduced the now-standard idea that a player's strength can be represented by a single number updated after each match according to the expected versus observed outcome. Glickman [2] generalised Elo into Glicko by modelling each player's rating as a distribution with an associated uncertainty, correcting Elo's slow adaptation to improving or inactive players; the Glicko-2 refinement [3] added a volatility parameter. TrueSkill [4] models each player's skill as a Gaussian and performs approximate message passing to infer posterior ratings, crucially generalising to teams, and TrueSkill 2 [5] incorporated additional match signals such as score, team size and player experience. Whole-history rating [6] analyses the trade-off between accuracy and adaptivity. Collectively these systems answer "how strong is this player?" with increasing precision, but they treat a team as an aggregate of individual ratings and are therefore silent on how those individuals combine as a unit.

B. Role Composition and Team Outcomes

A parallel body of empirical work shows that composition matters as much as individual skill. In the analysis of multiplayer online battle arena (MOBA) matches, Pobiedina et al. [7] demonstrated that team success depends significantly on the distribution of roles and on team-internal balance, not merely on the average rank of the players. Yang et al. [8] identified recurring combat patterns that predict victory independently of raw mechanical skill, and Sapienza et al. [9] used tensor factorisation to characterise the roles players adopt and the way those roles interact within winning teams. Industry practice reflects this recognition: official role-selection and role-queue features in titles such as *VALORANT* and *Overwatch* were introduced precisely because skilled players disliked matches in which role composition was left to chance [10], [11]. These features shape the composition of a lobby, but they do not themselves guarantee that the teams formed within that lobby are balanced on both role and skill.

C. Fairness-Aware Team Formation

Team formation with explicit fairness or coverage objectives has a rich history outside gaming. Lappas et al. [12] formalised the problem of finding a team of experts that covers a required set of skills while minimising communication cost, and showed its hardness. Anagnostopoulos et al. [13] studied the online version, in which teams must be formed sequentially from a stream of arrivals. Kargar and An [14] and Gajewar and Sarma [15] developed graph-theoretic and densest-subgraph formulations for multi-skill collaborative teams. In education and human-resource analytics, genetic algorithms and related search methods have been used to compose heterogeneous student groups [16] and fantasy-league squads, while two-stage optimisation optimises role coverage and skill distribution jointly [17]. Fair-division theory supplies further guarantees: random serial-dictatorship and approximate maximin-share mechanisms [18] can bound how much any participant envies another. These methods are effective, but their search spaces and iterative refinements assume that computing time is plentiful—an assumption that does not hold in a matchmaking queue.

D. Real-Time Matchmaking Constraints

The real-time constraint is well documented in the networked-games literature. Smed et al. [19] and Claypool and Claypool [20] established that player-perceived responsiveness degrades sharply as latency rises, and that matchmaking is bound by the same tight budgets as gameplay traffic. Delalleau et al. [21] described the advanced matchmaking pipeline of *Ghost Recon Online*, showing that production systems combine rating with additional constraints under strict time budgets.

E. Research Gap

The literature reveals three gaps. First, skill-rating research and role-composition research have developed largely independently, so no single widely accepted method balances both. Second, the algorithms that do optimise composition are computationally suited to offline environments and are too slow for live queues. Third, existing models are not organised around the role structures of contemporary tactical shooters and battle royales. HRTBA is proposed specifically to close these gaps; Table I positions it relative to the most closely related approaches.

III. Proposed Method: HRTBA

A. Overview

The Hybrid Role-Aware Team Balancing Algorithm (HRTBA) is a deterministic pipeline that forms composition-fair, skill-balanced teams in a single pass over the player pool.

TABLE I
POSITIONING OF HRTBA RELATIVE TO KEY RELATED APPROACHES

Approach	Skill balance	Role aware	Real-time	Domain
Elo / Glicko [1], [2]	Yes	No	Yes	Generic / chess
TrueSkill [4]	Yes	No	Partial	Consoles / generic
Skill-only snake draft	Yes	No	Yes	Multiplayer games
Two-stage optimisation [13]	Yes	Yes	No	Classrooms / HR
Genetic algorithms [16]	Yes	Yes	No	Classrooms / fantasy sport
Graph densest-subgraph [12]	Partial	Yes	No	Expert teams
Greedy role fill	Partial	Partial	Yes	General
HRTBA (this work)	**Yes**	**Yes**	**Yes**	**Tactical shooters / BR**

It is inspired by the pre-match role-queue systems that players already use in titles such as *VALORANT* and *Overwatch*, but it adds the assembly and correction stages that turn a role-aware lobby into genuinely balanced teams. HRTBA treats role composition as a structural constraint that is satisfied by construction, and skill balance as an objective that is optimised within that structure.

The algorithm is built on a simple observation: if roles are used to organise the queue, then assembling a team becomes a matter of drawing one player per role rather than searching for a combination that happens to cover every role. This transforms an apparently combinatorial fairness problem into a sequence of cheap, local decisions.

B. The Four-Stage Pipeline

Stage 1 — Role selection. Each player selects a desired role before entering the queue. Explicit selection removes the need for the system to infer a player's role from history and eliminates a major source of mismatch.

Stage 2 — Role-bucketed queues. The matchmaker maintains a separate queue for each role and keeps the players within each queue ordered by rating. Because the two strongest remaining players in a bucket are always adjacent, the candidates for a fair pairing can be retrieved in constant time.

Stage 3 — Balanced team assembly. For each match, the builder draws the two strongest remaining players from each role bucket and assigns them to opposing teams. To smooth skill, the stronger of the pair is assigned to the team with the lower current total. Baseline role coverage is therefore guaranteed by construction rather than checked after the fact.

Stage 4 — Same-role skill swap. A single correction pass examines pairs of players who share a role but sit on opposite teams and swaps them whenever doing so reduces the cross-team skill gap. Because the swap preserves role composition, it can improve balance without ever breaking coverage.

C. Mathematical Formulation

Let P denote the player pool, let $R = \{r_1, \ldots, r_k\}$ be the set of roles with $k = 4$, and let $T = k$ be the size of a team. Each player p has a rating $s(p)$ and a role $\mathrm{role}(p)$. A match consists of two teams A and B, each of size k. HRTBA seeks, for every match, an assignment that satisfies two conditions simultaneously.

Constraint (role coverage). Every role appears exactly once on each team:

$$\begin{aligned} &\forall r \in R: \quad \big|\{p \in A : \mathrm{role}(p) = r\}\big| = 1 \\ &\text{and} \quad \big|\{p \in B : \mathrm{role}(p) = r\}\big| = 1. \end{aligned} \tag{1}$$

Objective (skill balance). Minimise the absolute difference of team totals:

$$\min \Big| \sum_{p \in A} s(p) \; - \; \sum_{p \in B} s(p) \Big|. \tag{2}$$

Stage 3 constructs an initial assignment that satisfies Eq. (1) exactly. Stage 4 then performs swaps that are role-preserving, so the feasible set is the set of assignments with one player of each role per team. Within that set HRTBA performs a local search over same-role exchanges:

$$\begin{aligned} &(i, j) \text{ such that } \mathrm{role}(i) = \mathrm{role}(j) \\ &\text{and } \big|\Sigma_{A'} - \Sigma_{B'}\big| < \big|\Sigma_A - \Sigma_B\big|, \end{aligned} \tag{3}$$

where $i \in A$, $j \in B$ share a role r and Σ denotes a team total. The correction step repeats this exchange until no swap reduces the gap. Because role coverage is invariant under such swaps, Eq. (1) is never violated during optimisation—the central design property that distinguishes HRTBA from post-hoc filtering.

D. Algorithm

The complete procedure for one matchmaking cycle is given as Algorithm 1.

E. Complexity Analysis

Let n be the number of players in the pool. Bucketing and sorting the roles costs $O(n \log n)$. Forming each match draws one player per role per team, which is $O(k)$ per match with k constant, and the number of matches is $O(n/k)$. The correction pass examines at most $k \times k$ same-role player pairs per match and repeats only while an improvement is found; in practice it converges in a small, bounded number of passes. The overall cost is therefore dominated by the initial sort, giving $O(n \log n)$ time and $O(n)$ space. This is the crucial property that makes HRTBA suitable for live queues: its cost grows no faster than the sort that a skill-only system already performs. Table II compares the asymptotic cost of the evaluated strategies.

Algorithm 1 HRTBA (one match cycle)

for all roles r **do**
  $\text{bucket}[r] \leftarrow \text{sort}(Q[r],\ key = -\text{skill})$
end for
$A, B \leftarrow [\,], [\,]$; $\text{ptr}[r] \leftarrow 0$ for every r
while two complete teams can still be filled **do**
  for all roles r **do**
    $p_0, p_1 \leftarrow \text{bucket}[r][\text{ptr}[r]],\ \text{bucket}[r][\text{ptr}[r]{+}1]$
    $\text{ptr}[r] \leftarrow \text{ptr}[r] + 2$
    $\text{assign}(p_0, p_1, A, B)$ {stronger player to lower-total team}
  end for
  $A, B \leftarrow \text{same_role_swap}(A, B)$ {reduce residual gap, preserve coverage}
  emit (A, B)
end while

TABLE II
ASYMPTOTIC COST OF THE COMPARED STRATEGIES

Strategy	Time	Space	Composition-aware
Random	$O(n)$	$O(n)$	No
Skill-only	$O(n \log n)$	$O(n)$	No
Greedy	$O(n \log n)$	$O(n)$	Partial
HRTBA	$O(n \log n)$	$O(n)$	Yes

F. Design Rationale

Three design decisions differentiate HRTBA from prior work. First, it places role selection before the queue, so role coverage is a property of the input rather than an objective to be searched for. Second, it makes coverage a hard constraint and skill balance a soft objective, reflecting the reality that a missing role is a categorical defect whereas a small rating difference is a matter of degree. Third, it uses a bounded local correction rather than a global search, ensuring that the fairness gain is achieved within a predictable time budget.

IV. EXPERIMENTAL SETUP

A. Research Design

The study follows a quantitative, experimental design. A simulation harness generates synthetic player pools whose statistical properties are controlled, applies each matchmaking strategy to the same pool, and measures the resulting outcomes. Synthetic pools are used because they allow the role distribution, skill distribution and pool size to be varied systematically, and because they permit a large number of repeated trials, which is not possible with proprietary live telemetry. The research is comparative and hypothesis-driven: four strategies are evaluated on four dependent measures across five pool sizes with thirty randomised trials per configuration, yielding a factorial structure of $4 \times 5 \times 30$ observations per metric.

TABLE III
ROLE DEFINITIONS USED IN THE SIMULATION

Role	Abbrev.	Primary function
Duelist	DUE	Entry fragging and creating space
Controller	CTL	Smoke, area denial and space control
Initiator	INI	Gathering information and opening engagements
Sentinel	SEN	Anchoring, healing and area lockdown

B. Data Model

Each player in a simulated pool is a record containing a unique identifier, a skill rating drawn from a normal distribution (mean 1500, standard deviation 250, truncated at a practical floor of 300), and a declared role. Roles are assigned round-robin so that every role is adequately supplied, mirroring the situation in which the matchmaker admits players only when each role bucket can be satisfied. Table III lists the simulated roles.

C. Strategies Compared

Four strategies are compared. **Random** shuffles players and splits them into teams with no consideration of skill or role; it represents a floor on fairness. **Skill-only** performs a snake draft on rating, alternately allocating the strongest and weakest remaining players to opposing teams; it represents the industry-standard approach. **Greedy** places the strongest players first, sending each player to whichever team still lacks that player's role, with ties broken by the lower current team total; it represents a naive composition-aware method without global balancing. **HRTBA** is the proposed four-stage pipeline.

D. Evaluation Metrics

Four metrics operationalise the evaluation:

1) **Role coverage rate** — the mean fraction of required roles represented on a team, in percent. A team of four players covering all four roles scores 100%.
2) **Skill balance** — the mean absolute difference between the summed ratings of the two teams in a match.
3) **Matchmaking speed** — the wall-clock time, in milliseconds, required to form all teams from a pool.
4) **Consistency** — the standard deviation of the skill gap across matches and trials.

E. Experimental Parameters

Player pools of 20, 50, 100, 200 and 500 players are generated. For each pool size the entire simulation is repeated for thirty independent trials with different random seeds, and the mean of each metric is reported. A single random seed stream is used to generate pools so that every strategy within a trial faces exactly the same players, which isolates the effect of the strategy. The seeds are deterministic ($\text{seed} = 1000 \times \text{pool size} + \text{trial index}$), so an independent run with the same parameters reproduces the reported figures exactly.

TABLE IV
EXPERIMENTAL PARAMETERS

Parameter	Value
Roles per team	4
Players per team	4
Players per match	8
Pool sizes	20, 50, 100, 200, 500
Trials per configuration	30
Skill distribution	$\mathcal{N}(1500, 250)$, floor 300
Role assignment	Round-robin
Random seeds	$1000 \times \text{pool} + \text{trial}$

TABLE V
ROLE COVERAGE RATE (%) BY STRATEGY AND POOL SIZE

Strategy	20	50	100	200	500
Random	71.5	71.3	69.0	69.0	68.8
Skill-only	73.5	69.7	68.0	68.9	68.8
Greedy	83.8	77.7	75.5	76.0	76.1
HRTBA	92.5	97.6	98.5	100.0	99.8

The implementation uses Python 3; the core simulation relies only on the standard library so that the timing measurements reflect algorithm cost rather than library overhead. A deliberately lightweight stack was chosen because HRTBA is intended to run inside a live matchmaking service, where per-call overhead translates directly into queue latency.

F. Validity

Internal validity is supported by holding the player pool constant across strategies within each trial and by averaging over thirty seeds to reduce the influence of any single random draw. Construct validity is addressed by using metrics that map directly onto the evaluation goals: coverage to the role requirement, skill gap to competitive balance, and timing to deployability. External validity is limited by the use of simulated pools; this limitation is discussed in Section V-F.

V. RESULTS AND DISCUSSION

A. Role Coverage

Table V reports mean role coverage as a percentage by strategy and pool size, plotted in Fig. 1. HRTBA is the only strategy that approaches complete role coverage, rising from about 92% at the smallest pool to essentially 100% once the pool is large enough to fill every bucket with margin. The skill-only snake draft leaves roughly one role uncovered on one team in three, because it has no mechanism for noticing that the players it is balancing happen to share a role. The greedy strategy improves on skill-only because it considers roles, but its single strongest-first pass still leaves roughly a quarter of teams with a duplicate role.

B. Skill Balance

Table VI and Fig. 2 report the mean cross-team skill gap. The skill-only strategy achieves the smallest raw gap, as expected, because balancing rating is its only objective; that advantage comes at the cost of the coverage failures noted above. HRTBA's gap is consistently close to the skill-only optimum and far below the random baseline, and at the smallest pools HRTBA occasionally produces a smaller gap than skill-only because role-constrained selection can reduce the variance that a pure snake draft introduces. The same-role swap stage is responsible for this: it removes the residual imbalance that remains after assembly without ever disturbing role coverage.

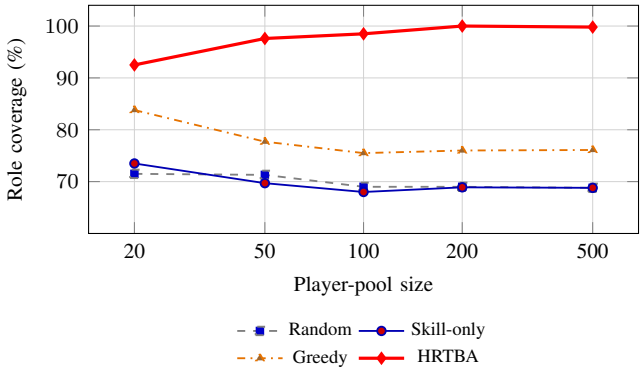


Fig. 1. Role coverage rate by strategy and player-pool size. HRTBA is the only strategy that reaches saturation.

TABLE VI
MEAN CROSS-TEAM SKILL GAP BY STRATEGY AND POOL SIZE

Strategy	20	50	100	200	500
Random	567.9	518.1	546.5	571.6	574.8
Skill-only	108.8	42.1	23.3	13.4	4.9
Greedy	150.1	52.6	25.3	14.7	5.8
HRTBA	97.8	47.6	30.8	16.2	7.7

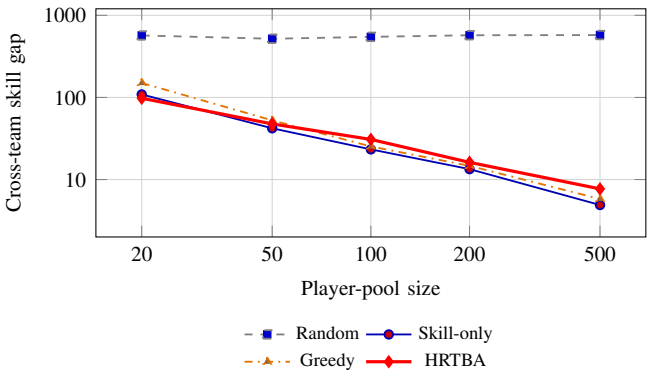


Fig. 2. Mean cross-team skill gap by strategy and player-pool size (logarithmic scale). HRTBA tracks the skill-only optimum while adding role coverage.

C. Matchmaking Speed

Table VII and Fig. 3 report formation time. All strategies form teams in a small fraction of a millisecond to a few milliseconds across the pool sizes tested. HRTBA is the slowest in absolute terms, but its cost at the largest pool remains on the order of a single millisecond or two, which is negligible against the tens to hundreds of milliseconds

TABLE VII
MEAN TEAM-FORMATION TIME (MS) BY STRATEGY AND POOL SIZE

Strategy	20	50	100	200	500
Random	0.012	0.022	0.026	0.060	0.104
Skill-only	0.006	0.014	0.026	0.060	0.154
Greedy	0.033	0.094	0.191	0.404	1.018
HRTBA	0.047	0.134	0.248	0.527	1.346

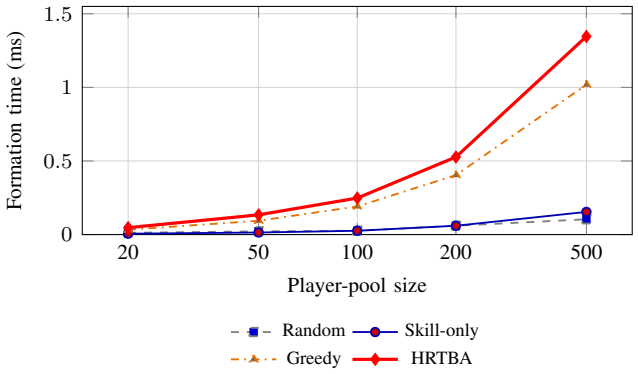


Fig. 3. Mean team-formation time by strategy and player-pool size. All strategies remain within a live-queue latency budget.

TABLE VIII
CONSISTENCY (SD OF SKILL GAP) BY POOL SIZE

Strategy	20	50	100	200	500
Random	208.3	344.7	377.5	412.2	425.3
Skill-only	55.3	40.7	28.9	25.1	11.6
Greedy	81.0	53.4	34.7	26.7	14.2
HRTBA	53.7	41.6	38.3	23.9	15.2

that network round-trips already consume in a live queue. Crucially, the growth is gentle: the ratio of HRTBA's time at 500 players to its time at 20 players is well under the twenty-five-fold increase in pool size, consistent with the $O(n \log n)$ analysis of Section III-E.

D. Consistency

Table VIII reports the standard deviation of the skill gap. HRTBA produces predictable outcomes: its variation is lower than the random and greedy strategies and comparable to skill-only, meaning that the fairness it delivers is not the average of a few excellent matches and many poor ones; it holds match after match.

E. Head-to-Head Summary

Table IX summarises the largest pool. HRTBA is the only strategy that approaches complete role coverage, while its skill gap stays close to the skill-only optimum and its formation time remains on the order of a millisecond. The improvement is not bought with compute: growth from the smallest to the largest pool is gentle, consistent with the $O(n \log n)$ analysis, and far below the milliseconds already consumed by a network round-trip.

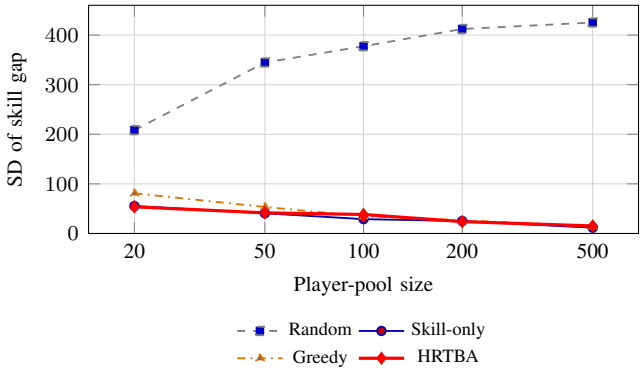


Fig. 4. Consistency (standard deviation of the skill gap) by strategy and pool size.

TABLE IX
HEAD-TO-HEAD SUMMARY AT POOL SIZE 500

Metric	Random	Skill-only	Greedy	HRTBA
Role coverage (%)	68.8	68.8	76.1	**99.8**
Mean skill gap	574.8	4.9	5.8	7.7
Formation time (ms)	0.104	0.154	1.018	1.346
Consistency (SD of gap)	425.3	11.6	14.2	15.2

F. Discussion

Taken together, the results show the nature of the trade-off clearly: fairness in composition is bought not with a large skill penalty, as prior offline methods imply, but with a small and bounded one. Relative to the rating systems of Elo [1], Glicko [2] and TrueSkill [4], HRTBA does not replace rating estimation; it consumes the rating those systems produce and adds a composition layer on top, making it composable with existing infrastructure rather than competitive with it. Prior fairness-aware team-formation methods achieve composition quality comparable to HRTBA but do so through search procedures whose cost is unsuitable for live queues; in contrast, HRTBA achieves its coverage guarantee by construction and confines optimisation to a bounded same-role swap.

The principal limitation is external validity: the evaluation uses synthetic pools with a normal skill distribution and a controlled role mix, whereas real player pools are shaped by rank thresholds, regional populations and time-of-day effects. The role-queue premise also assumes that every role bucket is adequately supplied; in practice popular roles may starve queues, and handling scarcity fairly is not addressed by the present pipeline. Finally, the experiments use four roles and four-player teams; behaviour under other cardinalities is argued from the complexity analysis but not measured directly.

VI. CONCLUSION

This paper presented HRTBA, a four-stage deterministic pipeline that guarantees role coverage by construction and refines skill balance with a bounded, role-preserving swap. Evaluated against Random, Skill-only and Greedy baselines on simulated pools of 20 to 500 players over thirty randomised

trials, HRTBA raised mean role coverage from roughly 68–76% to 92–100%, held the mean cross-team skill gap within a small margin of the skill-only optimum (7.7 versus 4.9 rating points at pool size 500), and formed all teams in under 1.4 ms for the largest pool. Its $O(n \log n)$ cost matches the sort a rating-based system already performs, so the method is deployable in a live queue without new hardware.

Future work includes weighted role preferences for flex players, coupling HRTBA with a Bayesian rating model so that the objective minimises the probability of a lopsided outcome rather than the raw rating difference, a learned role-pair synergy term as a secondary objective, generalisation to five-player teams and battle-royale squads, and validation on anonymised live telemetry.

REFERENCES

[1] A. E. Elo, *The Rating of Chessplayers, Past and Present*. New York, NY, USA: Arco, 1978.

[2] M. E. Glickman, "Parameter estimation in large dynamic paired comparison experiments," *Applied Statistics*, vol. 48, no. 3, pp. 377–394, 1999.

[3] M. E. Glickman, "Example of the Glicko-2 system," Boston University, Technical Report, 2012.

[4] R. Herbrich, T. Minka, and T. Graepel, "TrueSkill: A Bayesian skill rating system," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2006, pp. 569–576.

[5] T. Minka, R. Cleven, and A. Zaytsev, "TrueSkill 2: An improved Bayesian skill rating system," Microsoft Research, Technical Report, 2018.

[6] R. Coulom, "Whole-history rating: A Bayesian rating system for players of time-varying strength," in *Computers and Games*. Springer, 2008, pp. 113–124.

[7] N. Pobiedina, J. Neidhardt, M. d. C. Calatrava Moreno, L. Grad-Gyenge, and H. Werthner, "On successful team formation: Statistical analysis of a multiplayer online game," in *IEEE Conf. Business Informatics (CBI)*, 2013, pp. 55–62.

[8] P. Yang, B. E. Harrison, and D. L. Roberts, "Identifying patterns in combat that are predictive of success in MOBA games," in *Int. Conf. Foundations of Digital Games (FDG)*, 2014.

[9] A. Sapienza, A. Bessi, S. Damodaran, P. Shakarian, K. Lerman, and E. Ferrara, "Non-negative tensor factorization for analyzing online team behavior," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 719–731, 2018.

[10] Riot Games, "Role selection and matchmaking in VALORANT," Riot Games, Industry Documentation, 2020.

[11] Blizzard Entertainment, "Overwatch role queue: Developer update," Blizzard Entertainment, Industry Documentation, 2019.

[12] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, 2009, pp. 467–476.

[13] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *Int. Conf. World Wide Web (WWW)*, 2012, pp. 839–848.

[14] M. Kargar and A. An, "Discovering top-k teams of experts with or without a leader in social networks," in *ACM Int. Conf. Information and Knowledge Management (CIKM)*, 2011, pp. 985–994.

[15] A. Gajewar and D. Sarma, "Multi-skill collaborative teams based on densest subgraphs," in *SIAM Int. Conf. Data Mining (SDM)*, 2012, pp. 165–176.

[16] I. Srba and M. Bielikova, "Dynamic group formation as an approach to collaborative learning support," *IEEE Trans. Learn. Technol.*, vol. 8, no. 2, pp. 173–186, 2015.

[17] X. Duan, Z. Yang, and Y. Zhang, "Multi-objective optimisation for balanced team formation," *Applied Soft Computing*, vol. 101, p. 107118, 2021.

[18] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom, "Designing random allocation mechanisms: Theory and applications," *American Economic Review*, vol. 103, no. 2, pp. 585–623, 2013.

[19] J. Smed, T. Kaukoranta, and H. Hakonen, "Aspects of networking in multiplayer computer games," *The Electronic Library*, vol. 20, no. 2, pp. 87–97, 2002.

[20] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, no. 11, pp. 40–45, 2006.

[21] O. Delalleau, E. Contal, E. Thibodeau-Laufer, R. C. Ferrari, Y. Bengio, and F. Zhang, "Beyond skill rating: Advanced matchmaking in Ghost Recon Online," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 3, pp. 167–177, 2012.