DYNAMIC HETEROGENEOUS-STATE GUIDED EXACT SEARCH FOR MULTI-PAPER REVIEWER ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reviewer allocation requires assigning reviewer groups to papers while satisfying expertise and operational constraints. When multiple papers are assigned simultaneously, reviewer-load constraints couple their assignments, and optimizing aggregate reviewer–paper affinity may leave some papers with substantially lower expertise coverage. We therefore formulate the *Multi-Paper Reviewer Allocation* (MPRA) problem, which assigns a fixed-size reviewer group to each paper under keyword-relevance, reviewer-load, and group-level social-distance constraints, with the objective of maximizing the minimum keyword coverage over all papers. We prove that MPRA is NP-hard and propose *Dynamic Heterogeneous-State Guided Exact Search* (HGES), which combines rareness-aware greedy initialization with exact branch-and-bound search. HGES derives paper demand, reviewer availability, keyword rareness, and social feasibility from each partial allocation to guide paper expansion and reviewer-group enumeration, while admissible upper-bound pruning preserves optimality. We further develop an *Index-Enhanced Search Acceleration* (IESA) module that uses keyword inverted lists, paper-level candidate precomputation, and k-hop neighborhood indexing to reduce repeated search operations. On academic-network and attributed social-network instances, exact methods obtain matching average objectives on jointly solved instances. HGES+IESA reduces reported runtime relative to unguided branch-and-bound, while dynamic-ordering ablations show that fewer expanded nodes need not yield shorter runtime. General-purpose mixed-integer optimization remains faster in several settings.

1 INTRODUCTION

Reviewer assignment is a fundamental task in peer review and has been extensively studied from the perspectives of matching, recommendation, and constrained optimization (Wang et al., 2008; Zhao & Zhang, 2022; Jovanovic & Bagheri, 2023). Existing approaches commonly estimate reviewer–paper relevance from textual similarity, topic representations, or publication profiles, while incorporating operational constraints such as reviewer workload and conflict of interest (Dumais & Nielsen, 1992; Kobren et al., 2019; Stelmakh et al., 2019; Bouanane et al., 2024).

Most formulations characterize assignment quality primarily through reviewer–paper affinity (Zhao & Zhang, 2022; Cousins et al., 2023; Arabzadeh et al., 2024). Pairwise affinity alone, however, does not capture the complementary expertise of a reviewer group: a set of individually relevant reviewers may still overlap substantially in expertise, leaving a paper's research topics insufficiently covered (Fu et al., 2025; Kou et al., 2015). We therefore consider the *keyword coverage* achieved collectively by a reviewer group. The problem becomes harder when groups are assigned to multiple papers simultaneously, because reviewer capacities are shared and assignment decisions are globally coupled (Payan & Zick, 2021; Leyton-Brown et al., 2024): assigning a reviewer to one paper reduces the capacity available to the others. In addition, when reviewer relationship information is available, the composition of a group may be subject to a *group-level social-distance constraint*, requiring reviewers assigned to the same paper to be sufficiently separated in the reviewer graph (Li et al., 2017; Zhu et al., 2023). This structural constraint is considered separately from explicit paper–reviewer conflict-of-interest rules.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We therefore study the *Multi-Paper Reviewer Allocation* (MPRA) problem: given a keyword-attributed reviewer social graph and papers with required keyword sets, assign exactly r reviewers to each paper subject to keyword-relevance, reviewer-load, and group-level social-distance constraints, maximizing the minimum keyword coverage across papers. MPRA is challenging for three reasons: even for one paper, selecting r reviewers from a large candidate set is combinatorial; reviewer-load constraints couple the papers, since fixing one group changes the capacities available to the rest; and the social-distance constraint introduces pairwise feasibility dependencies within each group. With n reviewers, m papers, and group size r, naive enumeration may examine up to $\binom{n}{r}^m$ cross-paper combinations before feasibility filtering. We further prove that MPRA is NP-hard (Theorem 1), ruling out a polynomial-time exact algorithm for unrestricted instances unless $\mathrm{P} = \mathrm{NP}$.

Although the search space is large, a partial allocation reveals useful information about the remaining problem: the unassigned papers differ in keyword demand and candidate availability, reviewer capacities have changed, the supply of individual keywords may have decreased, and the social feasibility of a candidate depends on the group being constructed. These quantities evolve with the search and form the *dynamic heterogeneous state*.

To exploit this state while solving MPRA exactly, we propose the *Dynamic Heterogeneous-State Guided Exact Search* (HGES) framework. HGES first invokes GREEDYINIT, a rareness-aware greedy procedure that constructs a feasible allocation when possible, whose objective value initializes an incumbent lower bound. It then performs exact branch-and-bound search, where each node is a partial allocation A with size-r reviewer groups fixed for a subset of papers. From A, HGES builds a dynamic heterogeneous state $\mathcal{S}(A)$ characterizing paper demand, reviewer availability, keyword rareness, and social feasibility, and uses it to decide which remaining paper to expand next and in which order to examine candidate reviewers for that paper. The state is used only for ordering: it removes no feasible paper, reviewer, or group, and a branch is discarded only by explicit feasibility checking or when an admissible upper bound proves that no completion can improve the incumbent. This separation between state-guided ordering and correctness-preserving pruning lets HGES exploit the current state without sacrificing completeness or optimality. Because exact search repeatedly retrieves keyword-relevant reviewers and checks social feasibility, we further develop an *Index-Enhanced Search Acceleration* (IESA) module combining keyword inverted indexing, paper-level candidate precomputation, and k-hop neighborhood indexing; these structures preserve the MPRA feasibility conditions and objective, and only reduce online lookup and feasibility-checking costs at the expense of offline preprocessing and memory (Appendix G).

The main contributions are as follows:

- We formulate the *Multi-Paper Reviewer Allocation* (MPRA) problem, jointly considering reviewer-group keyword coverage, reviewer-load constraints, and group-level social-distance constraints under a max-min coverage objective, and prove that MPRA is NP-hard.
- We propose the Dynamic *Heterogeneous-State Guided Exact Search* (HGES) framework, which derives a dynamic heterogeneous state from each partial allocation to guide paper expansion and reviewer-group enumeration, while explicit feasibility checking and admissible upper-bound pruning preserve completeness and optimality.
- We develop an *Index-Enhanced Search Acceleration* (IESA) module that reduces repeated candidate retrieval and social-feasibility checking without changing the MPRA optimization problem.
- We conduct experiments on academic-network allocation and attributed social-network stress-test instances, comparing HGES with heuristic, sequential, and exact baselines on solution quality, search efficiency, parameter sensitivity, and the time–space overhead of IESA, and reporting both the regimes in which HGES reduces search effort and those where a general-purpose solver is faster.

2 RELATED WORK

We review *reviewer assignment with matching and fairness objectives* and *reviewer-group formation with coverage and social constraints*, then position MPRA relative to both.

Reviewer assignment with matching and fairness objectives. Reviewer assignment commonly models paper–reviewer compatibility using affinity scores and optimizes aggregate matching under operational constraints (Wang et al., 2008; Zhao & Zhang, 2022; Jovanovic & Bagheri, 2023). Affinity estimation has evolved from textual similarity (Dumais & Nielsen, 1992) to semantic and information-retrieval methods (Tan et al., 2021; Arabzadeh et al., 2024), with workload, conflict-of-interest, and related constraints incorporated into the assignment process (Aksoy et al., 2025). Fairness-aware methods further impose local fairness constraints (Kobren et al., 2019), protect the least well-served papers (Stelmakh et al., 2019), or balance assignment quality across papers (Bouanane et al., 2024), addressing disparities that aggregate-score maximization may overlook.

Coverage-based reviewer groups and social constraints. Coverage-based formulations consider the complementary expertise of a group rather than individual relevance alone: Kou et al. (2015) study weighted coverage-based assignment, review-team formation methods consider expertise and diversity (Mirzaei et al., 2019; Maleszka et al., 2020), and multi-subject coverage has recently been studied in expert group recommendation (Fu et al., 2025). Social-network information has also been incorporated: Li et al. (2017) consider academic relationships in fair assignment, while tenuous-group studies formulate graph constraints at the group level—the most tenuous group query (Li et al., 2023) seeks weakly related members, and the keyword-based socially tenuous group (KTG) query (Zhu et al., 2023) additionally requires members to cover query keywords.

Learning for combinatorial search. Recent work learns constructive policies (Kool et al., 2019), adapts solvers to test instances through active search (Hottung et al., 2022), or learns branching policies for branch-and-bound (Zhang et al., 2024; Kuang et al., 2024). HGES instead uses deterministic, problem-specific scores only to order exhaustive search; feasibility tests and admissible bounds determine pruning.

Positioning of MPRA. These studies motivate fairness-aware assignment, complementary expertise, and socially constrained groups. MPRA focuses on their joint treatment: it maximizes minimum keyword coverage over multiple reviewer groups subject to within-group social-distance constraints and shared reviewer capacities. This combination motivates exact search guided by the evolving partial allocation. Its distinction lies in the combined formulation and search procedure, rather than in introducing coverage or max-min fairness individually.

3 Problem Formalization

Let $G = (U, E, \mathcal{K})$ be a keyword-attributed reviewer social graph, where $U = \{u_1, \ldots, u_n\}$ is the reviewer set, $E \subseteq U \times U$ the set of undirected reviewer relationships, and $\mathcal{K}$ the global keyword universe; each reviewer $u \in U$ has an expertise keyword set $\mathcal{K}_u \subseteq \mathcal{K}$. Let $P = \{p_1, \ldots, p_m\}$, $m \geq 1$, be the papers, where each p_i has a nonempty keyword set $W_i \subseteq \mathcal{K}$ of required research topics. For $u, v \in U$, let $\mathrm{dist}(u, v)$ be their shortest-path distance in G, with $\mathrm{dist}(u, v) = \infty$ if they are disconnected.

Definition 1 (k-distance reviewer group). Given a nonnegative integer social-distance threshold k, a reviewer group $g \subseteq U$ is a k-distance reviewer group if

$$\mathrm{dist}(u, v) > k, \quad \forall u, v \in g,\ u \neq v. \tag{1}$$

This is a *group-level social-distance constraint*, restricting the relationships among reviewers assigned to the same paper.

Definition 2 (Keyword coverage). For paper p_i with keyword set W_i and reviewer group g, the keyword coverage of g for p_i is

$$\mathrm{cov}(g, p_i) = \frac{\left|\bigcup_{u \in g} (\mathcal{K}_u \cap W_i)\right|}{|W_i|}, \tag{2}$$

which lies in $[0, 1]$; repeated coverage of the same keyword by multiple reviewers does not increase it. We require every assigned reviewer to be keyword-relevant to its paper and define the *keyword-relevant candidate set* $C(p_i) = \{u \in U \mid \mathcal{K}_u \cap W_i \neq \emptyset\}$. With fixed eligibility or retrieval filters, $C(p_i)$ denotes the retained subset. Distances are still computed in the full underlying graph.

Definition 3 (Reviewer load). For any set of paper–group assignments A, the load of reviewer u is $\mathrm{load}_A(u) = |\{p_i \mid (p_i, g_i) \in A,\ u \in g_i\}|$. Given a reviewer-load limit L, A is load-feasible if $\mathrm{load}_A(u) \le L$ for every $u \in U$. This constraint couples the papers, since assigning u to one paper reduces the capacity of u available to the others.

Definition 4 (Multi-Paper Reviewer Allocation, MPRA). Given $G = (U, E, \mathcal{K})$, P, positive integer group size r, reviewer-load limit L, and threshold k, an *allocation* is $A = \{(p_i, g_i) \mid p_i \in P\}$, where g_i is the reviewer group assigned to p_i. The allocation A is *feasible* if

$$\begin{aligned} |g_i| &= r, && \forall p_i \in P, \\ g_i &\subseteq C(p_i), && \forall p_i \in P, \\ \mathrm{dist}(u, v) &> k, && \forall p_i \in P,\ \forall u, v \in g_i,\ u \neq v, \\ \mathrm{load}_A(u) &\le L, && \forall u \in U. \end{aligned} \tag{3}$$

Let $\mathcal{F}$ be the set of feasible allocations satisfying Eq. (3), and define $F(A) = \min_{p_i \in P} \mathrm{cov}(g_i, p_i)$ for $A \in \mathcal{F}$. MPRA seeks an *optimal allocation* $A^\star \in \arg\max_{A \in \mathcal{F}} F(A)$; hence, it maximizes the keyword coverage of the least-covered paper. If $\mathcal{F} = \emptyset$, the instance is infeasible. Optimality is relative to the fixed candidate sets $C(p_i)$. In Eq. (3), the first two conditions enforce group size and keyword relevance, the third enforces within-group social distance, and the fourth enforces reviewer load across all papers. An equivalent mixed-integer linear formulation is given in Appendix D.

Appendix E states and proves the NP-hardness result by reducing the decision version of INDEPENDENT SET to a single-paper MPRA instance. The result therefore holds even without cross-paper coupling, and hence also in the general multi-paper setting.

Exact search additionally uses a *partial allocation*, which is a search state rather than a solution to Definition 4. Specifically, $A = \{(p_i, g_i) \mid p_i \in P_{\mathrm{fix}}(A)\}$ assigns feasible size-r groups only to the papers in $P_{\mathrm{fix}}(A) \subseteq P$, while $P_{\mathrm{rem}}(A) = P \setminus P_{\mathrm{fix}}(A)$ contains the unassigned papers. Expanding $p^\star \in P_{\mathrm{rem}}(A)$ builds a temporary group g; only after $|g| = r$ and the group-level conditions hold does $(p^\star, g)$ generate a child partial allocation. Appendix C gives the full construction.

4 THE HGES FRAMEWORK

Because reviewer capacities are shared, fixing the group of one paper changes the resources available to the rest. Exhaustive enumeration preserves optimality but is combinatorial, whereas greedy allocation is inexpensive but need not attain the max-min optimum. We therefore propose the Dynamic *Heterogeneous-State Guided Exact Search* (HGES) framework, an exact branch-and-bound method for MPRA.

HGES first invokes GREEDYINIT to construct a feasible allocation when possible and then performs exact search over alternative group assignments. At each search node, it builds a *dynamic heterogeneous state* $\mathcal{S}(A)$ from the current partial allocation A, characterizing paper demand, reviewer availability, keyword rareness, and social feasibility. This state determines which remaining paper is expanded next and the order in which candidate reviewers are examined. It does not determine feasibility or pruning: a branch is discarded only when it violates an MPRA constraint or when the admissible upper bound of Section 4.3 proves that no completion can improve the incumbent. Figure 1 summarizes the resulting workflow; Appendix A gives the full pseudocode.

4.1 RARENESS-AWARE GREEDY INITIALIZATION

HGES maintains $A^\star$ as the best feasible allocation found so far and $LB^\star = F(A^\star)$ as its incumbent lower bound; before any feasible allocation is found, it sets $LB^\star = -\infty$. GREEDYINIT attempts to obtain this incumbent before exact search. A feasible allocation with a larger objective value gives a stronger lower bound and can improve subsequent pruning.

It uses a two-level ordering. At the paper level, papers are ordered by bottleneck keyword supply, candidate-set size, and aggregate initial rareness. At the reviewer level, each step selects the feasible reviewer with the largest rareness-weighted marginal gain until the group reaches size r. Appendix B gives the full definitions.

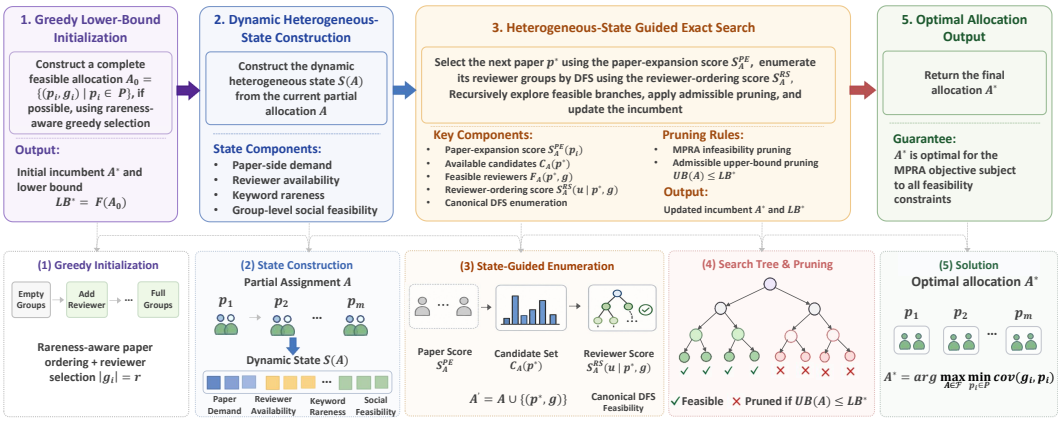

Figure 1: HGES overview. The dynamic state orders exact enumeration; only feasibility checks and the admissible bound prune branches.

If every paper receives a feasible group, HGES sets $A^\star \leftarrow A_0$ and $LB^\star \leftarrow F(A_0)$; this is a valid lower bound because A_0 is feasible. If GREEDYINIT cannot construct some group, it returns $\emptyset$. This does not establish infeasibility: a different sequence of earlier assignments may admit a feasible allocation, so HGES proceeds without an incumbent.

4.2 HETEROGENEOUS-STATE GUIDED ENUMERATION

Since GREEDYINIT is locally greedy, it cannot guarantee global optimality. HGES therefore performs exact branch-and-bound over partial allocations and builds a dynamic heterogeneous state from the current A to decide the traversal order: which unassigned paper to expand next, and in which order to examine candidate reviewers for it. Crucially, the state-guided scores only alter traversal order and eliminate no feasible paper, reviewer, or group.

4.2.1 HETEROGENEOUS STATE CONSTRUCTION

For each partial allocation A, HGES constructs

$$\mathcal{S}(A) = \langle \mathcal{D}_P(A), \mathcal{A}_U(A), \mathcal{R}_K(A), \mathcal{F}_G(A) \rangle , \tag{4}$$

where $\mathcal{D}_P(A)$, $\mathcal{A}_U(A)$, $\mathcal{R}_K(A)$, and $\mathcal{F}_G(A)$ represent paper-side demand, reviewer availability, keyword rareness, and group-level social feasibility, respectively; the branch-level quantities are derived from A, while group-dependent quantities are evaluated for the current temporary group g.

Paper-side demand. For paper p_i and temporary partial group g, the covered keyword set is $B(p_i, g) = \bigcup_{u \in g} (\mathcal{K}_u \cap W_i)$ and the residual demand is $D(p_i, g) = W_i \setminus B(p_i, g)$, so $D(p_i, \emptyset) = W_i$ for an unassigned paper. The temporary g is local to enumeration and does not modify A until it becomes a feasible size-r group.

Reviewer availability. Under A, the load-available candidate set of p_i is $C_A(p_i) = \{u \in C(p_i) \mid \mathrm{load}_A(u) < L\}$ and the residual capacity of u is $\mathrm{cap}_A(u) = L - \mathrm{load}_A(u)$. Thus $C(p_i)$ is static while $C_A(p_i)$ is its state-dependent subset.

Dynamic keyword rareness. For $w \in \mathcal{K}$, the remaining demand and load-available supply under A are $\mathrm{dem}_A(w) = |\{p_i \in P_{\mathrm{rem}}(A) \mid w \in W_i\}|$ and $\mathrm{sup}_A(w) = |\{u \in U \mid w \in \mathcal{K}_u, \mathrm{load}_A(u) < L\}|$, giving $\mathrm{rareness}_A(w) = \mathrm{dem}_A(w)/(\mathrm{sup}_A(w) + 1)$. For each remaining paper, the average and bottleneck rareness are

$$\rho_A(p_i) = \frac{1}{|W_i|} \sum_{w \in W_i} \mathrm{rareness}_A(w), \qquad \beta_A(p_i) = \max_{w \in W_i} \mathrm{rareness}_A(w), \tag{5}$$

summarizing the average and largest demand–supply imbalance, respectively. Supply counts available reviewers, not residual capacity units; rareness is an ordering heuristic.

Group-level social feasibility. For paper p_i, temporary group g, and partial allocation A,

$$F_A(p_i, g) = \{u \in C_A(p_i) \setminus g \mid \mathrm{dist}(u, v) > k,\ \forall v \in g\} \tag{6}$$

contains exactly the keyword-relevant, load-available reviewers that can be appended to g without violating the social-distance constraint. Because the ordering scores combine features on different scales, HGES applies min–max normalization within the set Ω of alternatives currently compared: $\mathcal{N}_\Omega[h(x)] = (h(x) - \min_{y\in\Omega} h(y))/(\max_{y\in\Omega} h(y) - \min_{y\in\Omega} h(y))$, and 0 when all values are equal. Normalization only constructs ordering scores and alters no feasibility condition.

4.2.2 Heterogeneous-State Guided Reviewer-Group Enumeration

Paper expansion score. With $\Omega_P = P_{\mathrm{rem}}(A)$, the *paper-expansion score* of $p_i \in \Omega_P$ is

$$\begin{aligned} S_A^{\mathrm{PE}}(p_i) = \mathcal{N}_{\Omega_P}[\beta_A(p_i)] + \mathcal{N}_{\Omega_P}[\rho_A(p_i)] \\ + \mathcal{N}_{\Omega_P}[1 - \mathrm{UB}(p_i \mid \emptyset, A)] + \mathcal{N}_{\Omega_P}\left[1 - |C_A(p_i)|/|U|\right]. \end{aligned} \tag{7}$$

The first two terms prioritize papers with larger demand–supply imbalance; the third prioritizes papers with more restricted optimistic achievable coverage, where the paper-level upper bound is defined in Section 4.3; the last prioritizes papers with fewer available candidates. HGES selects $p^\star \in \arg\max_{p_i \in P_{\mathrm{rem}}(A)} S_A^{\mathrm{PE}}(p_i)$. This score is not a feasibility criterion and removes no paper from the search space.

Reviewer-ordering score. Given $p^\star$, HGES enumerates its groups depth-first. Let g be the temporary group and $u \in F_A(p^\star, g)$. The newly covered keywords are $\Delta_A(u \mid p^\star, g) = (\mathcal{K}_u \cap W_{p^\star}) \setminus B(p^\star, g)$, with rareness-weighted contribution

$$G_A(u \mid p^\star, g) = \sum_{w \in \Delta_A(u|p^\star,g)} \mathrm{rareness}_A(w). \tag{8}$$

With $N_k(u) = \{v \in U \mid 1 \le \mathrm{dist}(u, v) \le k\}$, the social flexibility of u is $\phi(u) = 1/(|N_k(u)| + 1)$, larger when u has fewer potential conflicts. The rare-expertise consumption penalty is

$$\psi_A(u) = \frac{1}{\mathrm{cap}_A(u) + 1} \sum_{w \in \mathcal{K}_u} \mathrm{rareness}_A(w), \tag{9}$$

which is larger when a reviewer has limited residual capacity but covers expertise scarce relative to remaining demand. With $\Omega_R = F_A(p^\star, g)$, the *reviewer-ordering score* is

$$\begin{aligned} S_A^{\mathrm{RS}}(u \mid p^\star, g) = \mathcal{N}_{\Omega_R}[|\Delta_A(u \mid p^\star, g)|] + \mathcal{N}_{\Omega_R}[G_A(u \mid p^\star, g)] \\ + \mathcal{N}_{\Omega_R}[\mathrm{cap}_A(u)] + \mathcal{N}_{\Omega_R}[\phi(u)] - \mathcal{N}_{\Omega_R}[\psi_A(u)]. \end{aligned} \tag{10}$$

Candidates are examined in descending order of S_A^{RS}. Algorithm 1 (Appendix A) initializes the incumbent, evaluates feasible allocations at leaf nodes, applies the branch-level bound, and enumerates groups for the selected paper. Each feasible size-r group generates a child partial allocation. The canonical indices remain fixed when candidates are visited by score; recursing from index $t + 1$ gives each combination a unique increasing-index path. State guidance changes only visiting order and preserves enumeration of every feasible group not safely pruned.

4.3 HGES with Admissible Upper-Bound Pruning

The state determines only the traversal order. To reduce the explored branches, HGES derives an admissible upper bound on the best objective attainable by any completion of a partial allocation. For paper p_i, temporary group g, and partial allocation A with $|g| < r$, the optimistic coverable keyword set is

$$\widehat{W}_A(p_i, g) = B(p_i, g) \cup \bigcup_{u \in F_A(p_i,g)} (\mathcal{K}_u \cap W_i), \tag{11}$$

and $\widehat{W}_A(p_i, g) = B(p_i, g)$ when $|g| = r$. The paper-level upper bound is

$$\mathrm{UB}(p_i \mid g, A) = |\widehat{W}_A(p_i, g)|/|W_i|. \tag{12}$$

When $|g| < r$, Eq. (11) relaxes three restrictions: it ignores the remaining $r - |g|$ slot limit when combining coverable keywords, does not require individually feasible reviewers to be mutually socially feasible, and ignores future cross-paper competition for capacity. Each relaxation can only enlarge the coverable set, so the bound may be loose but never underestimates the coverage of a feasible extension.

For $P_{\text{fix}}(A)$ the coverage is already fixed, while each remaining paper is evaluated from an empty group, giving the branch-level bound

$$\mathrm{UB}(A) = \min \left\{ \min_{p_j \in P_{\text{fix}}(A)} \mathrm{cov}(g_j, p_j), \min_{p_i \in P_{\text{rem}}(A)} \mathrm{UB}(p_i \mid \emptyset, A) \right\}, \tag{13}$$

where the first term is omitted when $P_{\text{fix}}(A) = \emptyset$; when $P_{\text{rem}}(A) = \emptyset$, set $\mathrm{UB}(A) = F(A)$. Let $\mathrm{OPT}(A) = \max_{A' \succeq A} F(A')$ over feasible extensions of A, with $\mathrm{OPT}(A) = -\infty$ if none exists.

Given an incumbent $A^\star$, HGES prunes A whenever $\mathrm{UB}(A) \leq LB^\star$. The non-strict inequality is appropriate because MPRA requires one optimal allocation rather than all optima: equality may remove an alternative optimum, but cannot remove a strictly better solution. Appendix F formally proves both upper bounds admissible and establishes exactness. In particular, the ordering scores remove no branch, canonical enumeration preserves every feasible reviewer group, and every pruned branch is either infeasible or cannot improve the incumbent; thus exhaustive termination returns an optimal allocation for every feasible instance.

4.4 Index-Enhanced Search Acceleration

Exact search repeatedly retrieves keyword-relevant reviewers and checks group-level social feasibility; recomputing these from the original attributes and graph at every node is redundant. The *Index-Enhanced Search Acceleration* (IESA) module therefore precomputes keyword inverted indexes, paper-level candidate sets, and k-hop neighborhoods, and reuses them throughout the search. IESA changes only how candidate and social-feasibility information is accessed, leaving the objective, constraints, and pruning rule unchanged; traversal is also preserved when tie-breaking is identical. Construction and correctness are given in Appendix G, and the empirical evaluation in Section 5.5.

5 Experiments

We evaluate HGES from four aspects: comparison with baselines, ablation of the core components, parameter sensitivity, and the online and offline overhead of IESA.

5.1 Experimental Setup

We use four real-world networks. DBLP and AMiner are the primary academic reviewer-allocation datasets, where nodes are authors, edges are co-authorship relations, reviewer expertise is extracted from historical publications, and up to five title terms form each paper's required topics; the top 30 keyword-relevant reviewers per paper are retained and the reviewer universe of an instance is capped at 200. Gowalla and Brightkite contain friendship and check-in data rather than peer-review records and serve as attributed social-network stress tests, with check-in locations mapped to regions and treated as node keywords. Statistics and further construction details are in Appendix I. Unless stated otherwise, the default configuration uses group size $r = 3$, social distance $k = 2$, load limit $L = 3$, and $|P| = 16$ papers, with 30 independently generated instances per experiment.

We compare against **Sequential-KTG**, which adapts keyword-based socially tenuous group selection by processing papers sequentially without revisiting earlier groups; **MILP**, which solves the formulation of Appendix D with a general-purpose optimizer; **Unguided-B&B**, which shares the feasibility conditions, incumbent management, and admissible pruning of HGES but traverses in canonical order; and **Fixed-B&B**, which computes an informed order once from the initial state and never updates it. **HGES+IESA** is the full proposed framework. We report **MinCov** for allocation quality, **FeasibleRate** (fraction returning a feasible allocation), **SolvedRate** (fraction certifying an optimal allocation for a feasible instance), and wall-clock time together with the number of expanded search nodes; further metrics and the hardware configuration are in Appendix I.

Table 1: Comparison with baselines under the default configuration. Time is in seconds.

Method	DBLP		AMiner		Gowalla		Brightkite	
	MinCov	Time	MinCov	Time	MinCov	Time	MinCov	Time
Sequential-KTG	0.489	0.76	0.481	0.44	0.340	0.37	0.255	0.20
MILP	0.489	1.49	0.481	0.30	0.340	0.81	0.255	0.17
Unguided-B&B	0.489	14.61	0.481	240.72	0.340	887.58	0.255	729.10
Fixed-B&B	0.489	6.21	0.481	3.10	0.340	26.99	0.255	124.70
HGES+IESA	0.489	1.63	0.481	0.23	0.340	190.67	0.255	362.60

5.2 Comparison with Baselines

Table 1 shows that all exact methods obtain identical rounded average MinCov on the jointly solved instances—0.489, 0.481, 0.340, and 0.255 on DBLP, AMiner, Gowalla, and Brightkite—consistent with the design of HGES and IESA. Sequential-KTG matches these reported means on its feasible instances (FeasibleRate 90.0% on DBLP and AMiner, 66.7% on Gowalla, 73.3% on Brightkite) but permanently fixes each group before later papers are processed and provides no optimality certificate. Against Unguided-B&B, HGES+IESA reduces the reported average time from 14.61 s to 1.63 s on DBLP, from 887.58 s to 190.67 s on Gowalla, and from 729.10 s to 362.60 s on Brightkite. Fixed-B&B is faster than HGES+IESA on Gowalla and Brightkite. MILP is faster on DBLP, Gowalla, and Brightkite, whereas HGES+IESA is faster on AMiner; we therefore regard HGES as a problem-specific exact-search framework whose traversal behavior can be controlled and analyzed explicitly, rather than as a method expected to dominate a general-purpose optimizer on every instance.

5.3 Ablation Study

Unlike the baseline comparison, the variants here are derived from the proposed framework by retaining only part of its search mechanism or disabling selected state-guided ordering decisions.

5.3.1 Effect of Greedy Initialization and Exact Search

GreedyInit only runs the initialization of Section 4.1 and returns a feasible allocation when one can be built, without exact search; **HGES** uses it to obtain an incumbent and then performs exact search. The comparison tests whether the initialized allocation suffices or whether exact search is needed to revisit cross-paper decisions.

GreedyInit only is considerably faster, since it performs no exact search, but its ability to construct a feasible allocation varies: it succeeds on 86.7% of DBLP and 90.0% of AMiner instances, yet drops to 63.3% on Gowalla and 50.0% on Brightkite. Its average MinCov is also lower than that of HGES on DBLP (0.485 vs. 0.489), Gowalla (0.336 vs. 0.340), and Brightkite (0.240 vs. 0.255). These conditional means need not use identical successful subsets and do not establish paired quality gains. Initialization supplies a feasible incumbent, while exhaustive exact search establishes optimality and can revisit earlier decisions (Table 9 in Appendix J).

5.3.2 Effect of Heterogeneous-State Guided Enumeration

HGES w/o Dynamic Update computes informed paper and reviewer orders from the initial state and keeps them fixed; **HGES w/o PE** removes state-guided paper expansion, processing papers in canonical order while keeping reviewer ordering; **HGES w/o RS** keeps state-guided paper expansion but examines candidates in canonical order; **HGES** is the full mechanism. All variants share the same GreedyInit, feasibility conditions, incumbent-update rule, and upper-bound definition, so they differ only in traversal strategy.

Table 2 shows that the two dynamic components give complementary reductions on DBLP: w/o Dynamic Update expands 92,708 nodes, w/o RS 18,511, w/o PE 18,859, and full HGES 4,900—a reduction of about 94.7% over w/o Dynamic Update, and 73.5% and 74.0% over w/o RS and w/o PE. These results suggest that paper expansion and reviewer ordering can provide complementary

Table 2: Ablation of heterogeneous-state guided enumeration.

Dataset	Variant	Time (s)	Expanded	MinCov
DBLP	w/o Dynamic Update	121.90	92,708	0.496
	w/o RS	122.36	18,511	0.496
	w/o PE	242.86	18,859	0.496
	HGES	127.50	4,900	0.496
AMiner	w/o Dynamic Update	0.16	1	0.455
	w/o RS	126.99	12,421	0.455
	w/o PE	360.10	39,496	0.455
	HGES	149.24	4,086	0.455

search guidance on DBLP. The smaller node count does not yield a runtime improvement on DBLP: full HGES takes 127.50 s versus 121.90 s for w/o Dynamic Update. State updates and candidate sorting add overhead, although profiling is needed to quantify their contribution. On AMiner, the frozen variant reports one expanded node and 0.16 s, compared with 4,086 nodes and 149.24 s for full HGES; dynamic updating therefore does not uniformly reduce search effort or runtime. All variants return the same rounded mean MinCov on jointly solved instances.

5.4 Parameter Sensitivity Analysis

We vary the reviewer-group size r and the social-distance threshold k, one at a time, with ten independently generated instances per configuration. Because the settings are generated as separate instances rather than as paired modifications of a single instance, differences between settings should be interpreted descriptively rather than as isolated causal effects; we therefore focus on SolvedRate and observed search difficulty. Results for the load limit L and the number of papers $|P|$ are given in Appendix J.

Larger reviewer groups can increase the difficulty of exact optimization, since more selections must be coordinated per paper under all group-level constraints: Gowalla has no solved instance at $r = 5$ or $r = 7$, Brightkite has none at $r = 7$, and DBLP drops from 90% at $r = 3$ to 10% at $r = 6$, with the higher value at $r = 7$ (30%) reflecting the non-monotonic variation noted above (Table 5). The social-distance threshold shows the clearest trend: as k grows the group-level requirement becomes more restrictive. In these experiments, DBLP falls from 100% at $k = 1$ to 80% at $k = 3$, 20% at $k = 4$, and 10% at $k = 5$; AMiner follows the same pattern but stays tractable for more instances; and neither stress dataset has a solved instance for $k \geq 4$.

5.5 Effect and Overhead of IESA

Since IESA does not change the MPRA search space, we evaluate only its online acceleration and offline time–space overhead. It gives the clearest acceleration on the academic datasets, where the mean online time drops from 4.27 s to 1.63 s on DBLP (2.62×) and from 0.378 s to 0.228 s on AMiner, whereas the gain is small on Gowalla and Brightkite (1.06× and 1.01×), showing limited online acceleration; identifying the dominant cost requires profiling. The k-hop index dominates index memory, and total construction takes 0.32–5.34 minutes, confirming the intended trade-off: IESA is most attractive when one reviewer graph is reused across many instances, so that construction cost is amortized (Appendix J).

6 Conclusion

We formulated MPRA to maximize the minimum reviewer-group keyword coverage under keyword-relevance, reviewer-load, and group-level social-distance constraints. HGES combines rareness-aware initialization, state-guided enumeration, and admissible pruning; exhaustive termination returns an optimal allocation for the fixed candidate sets. IESA reduces repeated candidate retrieval and social-feasibility checking without changing the solution space. Experiments identify both settings where HGES reduces search effort and those where a general-purpose solver is faster. Future work will study tighter bounds, more memory-efficient indexing, and richer assignment constraints.

AI USE STATEMENT

Generative AI (ChatGPT) assisted with language editing, manuscript organization, LaTeX formatting, and review of mathematical exposition during revision. It did not generate new experimental measurements in this revision. The authors are responsible for verifying the final content and the completeness of this disclosure before submission.

ETHICS STATEMENT

This work studies reviewer assignment as a combinatorial optimization problem using bibliographic and location-based research datasets. Keyword coverage is a proxy for expertise, and automated assignment may inherit biases in publication and social-network data. The social-distance constraint is a structural modeling choice, not a normative judgment about collaboration or a substitute for explicit conflict-of-interest checks. Any deployment or release of derived data should respect privacy and applicable data-use permissions and include appropriate human oversight.

REPRODUCIBILITY STATEMENT

MPRA, HGES, and IESA are specified in Sections 3 and 4; full proofs are in Appendices E–F. Section 5 and Appendix I describe the reported setup, with additional results in Appendix J. The source code is available at: https://anonymous.4open.science/anonymize/mpra_-8EEB.

REFERENCES

Meltem Aksoy, Seda Yanik, and Mehmet Fatih Amasyali. A data-driven optimization approach for automated reviewer assignment using natural language processing. *Intelligent Systems with Applications*, pp. 200587, 2025.

Negar Arabzadeh, Sajad Ebrahimi, Sara Salamat, Mahdi Bashari, and Ebrahim Bagheri. Reviewerly: modeling the reviewer assignment task as an information retrieval problem. In *Proc. 33rd ACM International Conference on Information and Knowledge Management*, pp. 5554–5555, 2024.

Khadra Bouanane, Abdeldjaouad Nusayr Medakene, Abdellah Benbelghit, and Samir Brahim Belhaouari. FairColor: an efficient algorithm for the balanced and fair reviewer assignment problem. *Information Processing & Management*, 61(6):103865, 2024.

Cyrus Cousins, Justin Payan, and Yair Zick. Into the unknown: Assigning reviewers to papers with uncertain affinities. In *International Symposium on Algorithmic Game Theory*, pp. 179–197. Springer, 2023.

Susan T. Dumais and Jakob Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proc. 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 233–244, 1992.

Yongfan Fu, Jian Luo, Guofang Nan, and Dahui Li. Peer review expert group recommendation: a multi-subject coverage-based approach. *Expert Systems with Applications*, 264:125971, 2025.

André Hottung, Yeong-Dae Kwon, and Kevin Tierney. Efficient active search for combinatorial optimization problems. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nO5caZwFwYu.

Jelena Jovanovic and Ebrahim Bagheri. Reviewer assignment problem: a scoping review, 2023. URL https://arxiv.org/abs/2305.07887.

Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1247–1257, 2019.

Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByxBFsRqYm.

Ngai Meng Kou, Leong Hou U, Nikos Mamoulis, and Zhiguo Gong. Weighted coverage based reviewer assignment. In *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 2031–2046, 2015.

Yufei Kuang, Jie Wang, Haoyang Liu, Fangzhou Zhu, Xijun Li, Jia Zeng, Jianye Hao, Bin Li, and Feng Wu. Rethinking branching on exact combinatorial optimization solver: The first deep symbolic discovery framework. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jKhNBulNMh.

Kevin Leyton-Brown, Yatin Nandwani, Hedayat Zarkoob, Chris Cameron, Neil Newman, and Dinesh Raghu. Matching papers and reviewers at large conferences. *Artificial Intelligence*, 331: 104119, 2024.

Kaixia Li, Zhao Cao, and Dacheng Qu. Fair reviewer assignment considering academic social network. In *Proc. APWeb-WAIM Joint Conference on Web and Big Data*, pp. 362–376, 2017.

Na Li, Huaijie Zhu, Wenhao Lu, Ningning Cui, Wei Liu, Jian Yin, Jianliang Xu, and Wang-Chien Lee. The most tenuous group query. *Frontiers of Computer Science*, 17(2):172605, 2023.

Marcin Maleszka, Bernadetta Maleszka, Dariusz Król, Marcin Hernes, Denis Mayr Lima Martins, Leschek Homann, and Gottfried Vossen. A modular diversity-based reviewer recommendation system. In *Proc. Asian Conference on Intelligent Information and Database Systems*, pp. 550–561, 2020.

Maryam Mirzaei, Jörg Sander, and Eleni Stroulia. Multi-aspect review-team assignment using latent research areas. *Information Processing & Management*, 56(3):858–878, 2019.

Justin Payan and Yair Zick. I will have order! optimizing orders for fair reviewer assignment. *arXiv preprint arXiv:2108.02126*, 2021.

Ivan Stelmakh, Nihar B Shah, and Aarti Singh. Peerreview4all: Fair and accurate reviewer assignment in peer review. In *Algorithmic learning theory*, pp. 828–856, 2019.

Shicheng Tan, Zhen Duan, Shu Zhao, Jie Chen, and Yanping Zhang. Improved reviewer assignment based on both word and semantic features. *Information Retrieval Journal*, 24(3):175–204, 2021.

Fan Wang, Ben Chen, and Zhaowei Miao. A survey on reviewer assignment problem. In *Proc. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 718–727, 2008.

Changwen Zhang, Wenli Ouyang, Hao Yuan, Liming Gong, Yong Sun, Ziao Guo, Zhichen Dong, and Junchi Yan. Towards imitation learning to branch for MIP: A hybrid reinforcement learning based sample augmentation approach. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/850cc7c4e5619fdc2f30c4eaa5ce3c17-Abstract-Conference.html.

Xiquan Zhao and Yangsen Zhang. Reviewer assignment algorithms for peer review automation: a survey. *Information Processing & Management*, 59(5):103028, 2022.

Huaijie Zhu, Wei Liu, Jian Yin, Ningning Cui, Jianliang Xu, Xin Huang, and Wang-Chien Lee. Keyword-based socially tenuous group queries. In *Proc. IEEE 39th International Conference on Data Engineering*, pp. 965–977, 2023.

A HGES PSEUDOCODE AND ENUMERATION DETAILS

Figure 1 in the main paper summarizes the overall HGES workflow, and Algorithm 1 below gives the full pseudocode. HGES first invokes GREEDYINIT to obtain an incumbent allocation when possible. Each subsequent search node corresponds to a partial allocation A, which induces a dynamic heterogeneous state $\mathcal{S}(A)$. The state determines the paper-expansion and reviewer-ordering priorities, while reviewer groups are constructed within the branch-and-bound search. Each feasible size-r reviewer group generates a child partial allocation, and branches are removed only by feasibility checking or the admissible upper-bound rule.

Algorithm 1 HGES for MPRA

Require: Reviewer graph G, papers P, group size r, social-distance threshold k, load limit L
Ensure: An optimal allocation $A^\star$, or infeasible

$A_0 \leftarrow \text{GREEDYINIT}(G, P, r, k, L)$
if $A_0 \neq \emptyset$ **then**
  $A^\star \leftarrow A_0$; $LB^\star \leftarrow F(A_0)$
else
  $A^\star \leftarrow \emptyset$; $LB^\star \leftarrow -\infty$
end if
SEARCH($\emptyset$)
return $A^\star$

procedure SEARCH(A)
  if $P_{\text{rem}}(A) = \emptyset$ **then**
    if $A^\star = \emptyset$ or $F(A) > LB^\star$ **then**
      $A^\star \leftarrow A$; $LB^\star \leftarrow F(A)$
    end if
    return
  end if
  if $A^\star \neq \emptyset$ and $\text{UB}(A) \leq LB^\star$ **then**
    return
  end if
  Construct the dynamic heterogeneous state $\mathcal{S}(A)$
  Select $p^\star \leftarrow \arg\max_{p_i \in P_{\text{rem}}(A)} S_A^{\text{PE}}(p_i)$
  Fix a canonical ordering $\langle c_1, \ldots, c_q \rangle$ of $C_A(p^\star)$
  ENUMERATE($\emptyset, 1$)
end procedure

procedure ENUMERATE(g, s)
  if $|g| = r$ **then**
    $A' \leftarrow A \cup \{(p^\star, g)\}$
    SEARCH(A')
    return
  end if
  $\mathcal{R} \leftarrow \emptyset$
  for $t = s$ to q **do**
    $u \leftarrow c_t$
    if $u \in F_A(p^\star, g)$ **then**
      Compute $S_A^{\text{RS}}(u \mid p^\star, g)$
      Insert (u, t) into $\mathcal{R}$
    end if
  end for
  Sort $\mathcal{R}$ in descending order of S_A^{RS}
  for each $(u, t) \in \mathcal{R}$ **do**
    ENUMERATE($g \cup \{u\}, t + 1$)
  end for
end procedure

Enumeration context. Each ENUMERATE call retains the allocation A, selected paper $p^\star$, and canonical list of its enclosing SEARCH call. Child search calls must not overwrite this context. Loads are computed from the current allocation or updated and restored on recursion. Scores use

the comparison sets specified in the main text; ties use fixed identifiers. If exhaustive search ends with $A^\star = \emptyset$, the instance is infeasible.

B RARENESS-AWARE GREEDY INITIALIZATION: FULL DEFINITIONS

This appendix restates the GREEDYINIT procedure summarized in Section 4.1.

Initial keyword rareness. For each keyword $w \in \mathcal{K}$, the initial paper-side demand is $\mathrm{dem}_0(w) = |\{p_i \in P \mid w \in W_i\}|$ and the initial reviewer-side supply is $\mathrm{sup}_0(w) = |\{u \in U \mid w \in \mathcal{K}_u\}|$. The initial keyword rareness is

$$\mathrm{rareness}_0(w) = \frac{\mathrm{dem}_0(w)}{\mathrm{sup}_0(w) + 1}, \tag{14}$$

which is larger when more papers require w relative to the number of reviewers who can cover it.

Paper-level initialization order. Each paper p_i has bottleneck keyword supply $b_0(p_i) = \min_{w \in W_i} \mathrm{sup}_0(w)$ and aggregate initial rareness $R_0(p_i) = \sum_{w \in W_i} \mathrm{rareness}_0(w)$. GREEDYINIT processes papers in ascending order of $(b_0(p_i), |C(p_i)|, -R_0(p_i))$, so papers with smaller bottleneck supply are handled earlier, with ties broken by smaller candidate sets and then by larger aggregate rareness.

Reviewer-level greedy selection. Let A^G be the partial allocation under construction and g_i the group being built for p_i. A reviewer $u \in C(p_i) \setminus g_i$ is feasible for the next position only if $\mathrm{load}_{A^G}(u) < L$ and $\mathrm{dist}(u, v) > k$ for all $v \in g_i$. Its newly covered keyword set is $\Delta_0(u \mid p_i, g_i) = (\mathcal{K}_u \cap W_i) \setminus \bigcup_{v \in g_i} (\mathcal{K}_v \cap W_i)$, and the rareness-weighted marginal gain is

$$G_0(u \mid p_i, g_i) = \sum_{w \in \Delta_0(u \mid p_i, g_i)} \mathrm{rareness}_0(w). \tag{15}$$

GREEDYINIT repeatedly selects a feasible reviewer with maximum G_0 until $|g_i| = r$, then inserts (p_i, g_i) into A^G and updates the reviewer loads.

C PARTIAL ALLOCATIONS AND SEARCH NODES

Each search node of HGES corresponds to exactly one partial allocation $A = \{(p_i, g_i) \mid p_i \in P_{\mathrm{fix}}(A)\}$, where $P_{\mathrm{fix}}(A) \subseteq P$ is the set of papers whose feasible size-r reviewer groups have already been fixed and $P_{\mathrm{rem}}(A) = P \setminus P_{\mathrm{fix}}(A)$ is the remaining paper set. When HGES expands a paper $p^\star \in P_{\mathrm{rem}}(A)$, it temporarily constructs a *partial reviewer group* g for $p^\star$. The temporary group g is not yet part of the partial allocation A; only when $|g| = r$ and g satisfies all group-level feasibility conditions does it generate the child partial allocation $A' = A \cup \{(p^\star, g)\}$. Hence A represents assignments already fixed at a search node, whereas g represents the reviewer group currently being constructed for the next paper.

D MIXED-INTEGER LINEAR FORMULATION OF MPRA

To support comparison with general-purpose exact solvers, we give an equivalent mixed-integer linear formulation of MPRA. Let $x_{i,u} \in \{0, 1\}$ indicate whether reviewer u is assigned to paper p_i. For each $w \in W_i$, let $y_{i,w} \in \{0, 1\}$ indicate whether keyword w is covered by at least one reviewer assigned to p_i. Let $z \in [0, 1]$ denote the minimum keyword coverage over all papers. MPRA can be

formulated as

$$
\begin{aligned}
\max \quad & z \\
\text{s.t.} \quad & \sum_{u \in U} x_{i,u} = r, && \forall p_i \in P, \\
& \sum_{p_i \in P} x_{i,u} \le L, && \forall u \in U, \\
& x_{i,u} + x_{i,v} \le 1, && \forall p_i,\ u, v \in U,\ u \neq v,\ \mathrm{dist}(u,v) \le k, \\
& x_{i,u} = 0, && \forall p_i,\ u \notin C(p_i), \\
& y_{i,w} \le \sum_{\substack{u \in U \\ w \in \mathcal{K}_u}} x_{i,u}, && \forall p_i,\ w \in W_i, \\
& \sum_{\substack{u \in U \\ w \in \mathcal{K}_u}} x_{i,u} \le r\, y_{i,w}, && \forall p_i,\ w \in W_i, \\
& z \le \frac{1}{|W_i|} \sum_{w \in W_i} y_{i,w}, && \forall p_i \in P, \\
& x_{i,u}, y_{i,w} \in \{0,1\}, \qquad 0 \le z \le 1.
\end{aligned}
\tag{16}
$$

The third constraint enforces the group-level social-distance condition by forbidding the joint selection of any pair of distinct reviewers at distance at most k. The two constraints involving $y_{i,w}$ ensure that $y_{i,w} = 1$ if and only if at least one selected reviewer covers keyword w. Therefore,

$$\frac{1}{|W_i|} \sum_{w \in W_i} y_{i,w} = \mathrm{cov}(g_i, p_i),$$

and maximizing z is equivalent to maximizing the minimum keyword coverage.

E Proof of Theorem 1

Theorem 1. *The MPRA problem is NP-hard.*

Proof. We give a polynomial-time reduction from the decision version of Independent Set. Given an undirected graph $H = (V_H, E_H)$ and an integer $1 \le r \le |V_H|$ on a nonempty graph, the Independent Set problem asks whether H contains a set of r pairwise non-adjacent vertices.

From (H, r), construct an MPRA instance containing a single paper p_1. For every vertex $v \in V_H$, create a reviewer u_v and associate u_v with a unique keyword w_v. For every edge $(v_a, v_b) \in E_H$, create a social edge (u_{v_a}, u_{v_b}) between the corresponding reviewers. Let $W_1 = \{w_v \mid v \in V_H\}$, and set the reviewer-group size to r, the social distance threshold to $k = 1$, and the reviewer load limit to $L = 1$. Since $w_v \in W_1$ for every $v \in V_H$, all constructed reviewers are keyword-relevant to p_1. Consider the decision version of MPRA that asks whether there exists a feasible allocation $\mathcal{A}$ satisfying $F(\mathcal{A}) \ge \tau$, where $\tau = r/|W_1|$.

Suppose first that H contains an independent set $S \subseteq V_H$ of size r. Assign to p_1 the reviewer group $g_1 = \{u_v \mid v \in S\}$. Because no two vertices in S are adjacent in H, no two reviewers in g_1 are connected by a social edge. Hence $\mathrm{dist}(u, v) > 1$ for all distinct $u, v \in g_1$, and g_1 satisfies the $k = 1$ social-distance constraint. Moreover, the r selected reviewers cover r distinct keywords, giving $\mathrm{cov}(g_1, p_1) = r/|W_1| = \tau$. Thus, the constructed MPRA decision instance is a yes-instance.

Conversely, suppose that the constructed MPRA instance admits a feasible allocation A with $F(A) \ge \tau$. Since the instance contains only one paper p_1, let g_1 denote its assigned reviewer group. The group-size constraint gives $|g_1| = r$. Because $k = 1$, the social-distance constraint requires $\mathrm{dist}(u, v) > 1$ for every pair of distinct reviewers $u, v \in g_1$. Hence, no two selected reviewers are connected by an edge in the constructed reviewer graph. By construction, such an edge exists if and only if the corresponding vertices are adjacent in H. Therefore, the r vertices corresponding to the reviewers in g_1 are pairwise non-adjacent and form an independent set of size r in H.

Every feasible group in the constructed instance covers exactly r distinct keywords and therefore attains coverage τ. Therefore, the constructed MPRA instance admits a feasible allocation with

objective value at least $\tau = r/|W_1|$ if and only if H contains an independent set of size r. The reduction is polynomial, and hence MPRA is NP-hard. □

F Proofs of Admissibility and Exactness

Lemma 1 (Paper-level upper-bound admissibility). *For any partial allocation A, paper p_i, and temporary partial reviewer group g, every size-r reviewer group g'_i extending g and feasible under A satisfies*

$$\mathrm{cov}(g'_i, p_i) \le \mathrm{UB}(p_i \mid g, A). \tag{17}$$

Proof. Every keyword already covered by g belongs to $B(p_i, g) \subseteq \widehat{W}_A(p_i, g)$. Consider any additional keyword covered by some reviewer $u \in g'_i \setminus g$. Since g'_i is a feasible extension under A, u is keyword-relevant to p_i, satisfies $\mathrm{load}_A(u) < L$, and satisfies the social-distance constraint with every reviewer in g. Thus $u \in F_A(p_i, g)$, and every keyword contributed by u is contained in $\widehat{W}_A(p_i, g)$. Hence, all keywords covered by g'_i are contained in $\widehat{W}_A(p_i, g)$, which yields Eq. (17). □

Lemma 2 (Branch-level upper-bound admissibility). *For every partial allocation A,*

$$\mathrm{OPT}(A) \le \mathrm{UB}(A). \tag{18}$$

Proof. For a fixed paper $p_j \in P_{\mathrm{fix}}(A)$, every completion of A retains the same reviewer group g_j and therefore the same coverage. For a remaining paper $p_i \in P_{\mathrm{rem}}(A)$, Lemma 1 gives $\mathrm{cov}(g_i, p_i) \le \mathrm{UB}(p_i \mid \emptyset, A)$ for every feasible completion. Since the MPRA objective is the minimum coverage over all papers, every feasible completion has objective value at most Eq. (13). Taking the maximum over all feasible completions proves Eq. (18). If there is no feasible completion, $\mathrm{OPT}(A) = -\infty$ and the inequality holds trivially. □

Theorem 2 (Exactness of HGES). *Upon exhaustive termination, HGES returns an optimal allocation for every feasible MPRA instance.*

Proof. The paper-expansion score and reviewer-ordering score change only the order in which search branches are visited. By construction of the canonical DFS enumeration, every feasible size-r reviewer group remains reachable unless the corresponding branch is pruned. For any selected paper, every feasible completion contains a group for that paper; hence choosing one paper by priority does not require enumerating paper permutations.

Branches violating an MPRA constraint cannot contain a feasible allocation. For a branch pruned by the rule $\mathrm{UB}(A) \le LB^\star$, Lemma 2 gives

$$\mathrm{OPT}(A) \le \mathrm{UB}(A) \le LB^\star.$$

Since $LB^\star$ is the objective value of an already discovered feasible allocation, removing the branch cannot eliminate a strictly better solution. If equality holds for a branch containing a globally optimal allocation, the incumbent already attains the same optimal objective value.

Therefore, every branch removed by HGES is either infeasible or cannot improve the current incumbent. When the search terminates, $A^\star$ attains the optimal MPRA objective. □

G Index-Enhanced Search Acceleration

HGES repeatedly performs two basic operations during exact search: retrieving reviewers relevant to paper keywords and checking whether a candidate reviewer satisfies the group-level social-distance constraint. Recomputing these operations from the original keyword attributes and reviewer graph at every search node introduces redundant cost. IESA consists of the following three structures, which are constructed before exact search and reused throughout the branch-and-bound procedure.

Keyword inverted index. For each keyword $w \in \mathcal{K}$, IESA constructs

$$I_{\mathrm{kw}}(w) = \{u \in U \mid w \in \mathcal{K}_u\}. \tag{19}$$

Thus $I_{\mathrm{kw}}(w)$ contains exactly the reviewers whose expertise includes w, and the index supports keyword-specific reviewer retrieval without repeatedly scanning the complete reviewer set.

Paper-level candidate precomputation. Using the keyword inverted index, the static keyword-relevant candidate set defined in Section 3, before any optional fixed filters, can be obtained as

$$C(p_i) = \bigcup_{w \in W_i} I_{\mathrm{kw}}(w). \tag{20}$$

If eligibility or retrieval filters are used, they are applied to the union in Eq. (20) before materialization. IESA then materializes the resulting $C(p_i)$ once before exact search. At a search node corresponding to partial allocation A, the dynamic candidate set is then obtained by

$$C_A(p_i) = \{u \in C(p_i) \mid \mathrm{load}_A(u) < L\}. \tag{21}$$

Hence, static keyword relevance is handled offline, while state-dependent reviewer availability remains evaluated under the current partial allocation.

k-hop neighborhood index. For each reviewer u, IESA precomputes

$$I_{\mathrm{soc}}(u) = N_k(u) = \{v \in U \mid 1 \le \mathrm{dist}(u, v) \le k\}. \tag{22}$$

A pair of distinct reviewers violates the social-distance constraint exactly when $v \in I_{\mathrm{soc}}(u)$. For $u \in C_A(p_i) \setminus g$, social compatibility with g holds if and only if $I_{\mathrm{soc}}(u) \cap g = \emptyset$. The same index is reused during reviewer-group enumeration and social-feasibility evaluation.

Proposition 1. IESA preserves the feasible solution space and the optimal objective value of MPRA.

Proof. Equation (20), followed by the same fixed filters when applicable, produces the candidate set $C(p_i)$ used by MPRA, and Eq. (22) represents exactly the predicate $\mathrm{dist}(u, v) \le k$ for distinct reviewers. Therefore, replacing repeated scans and distance computations with these index lookups does not add or remove any feasible reviewer assignment. All objective evaluations remain unchanged, and hence the feasible solution space and optimal objective value are preserved. □

IESA introduces an offline time–space trade-off: materializing candidate sets and k-hop neighborhoods reduces repeated online operations but requires additional preprocessing time and memory.

H Illustrative Examples

Example 1: motivation of joint allocation. Fig. 2 shows a keyword-attributed reviewer social graph with reviewers $u_0, \ldots, u_6$, and Table 3 lists three papers and their required keywords. Reviewer u_5, for example, covers keywords required by both p_1 and p_2. Assigning u_5 to one paper therefore consumes capacity that may also be useful to another paper. Meanwhile, reviewers assigned to the same paper must satisfy the prescribed social-distance constraint. Hence, reviewer groups for different papers need to be determined jointly by considering keyword coverage, reviewer capacity, and group-level social feasibility.

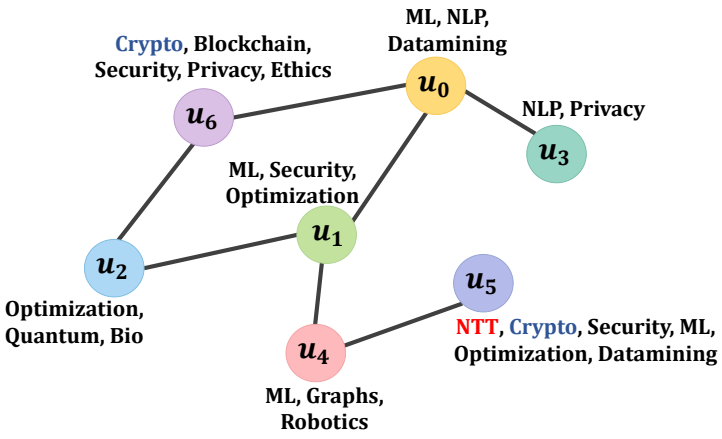

Figure 2: A keyword-attributed reviewer social network.

Table 3: Papers and their required research topics.

Paper	Research topics
p_0	ML, NLP, Optimization
p_1	NTT, Crypto, Optimization
p_2	Blockchain, Crypto, ML

Example 2: GREEDYINIT on a toy instance. Consider the MPRA instance in Fig. 2 and Table 3, with social distance threshold $k = 1$, reviewer load limit $L = 2$, and reviewer-group size $r = 3$. For this instance, GREEDYINIT obtains $b_0(p_0) = 2$, $b_0(p_1) = 1$, and $b_0(p_2) = 1$. After applying the remaining ordering criteria of Section 4.1, the papers are processed in the order $p_1 \to p_2 \to p_0$. For p_1, reviewer u_5 is considered first because its expertise contributes the required keywords with limited supply, including NTT. The group is subsequently completed with reviewers that satisfy the load and social-distance constraints, resulting in $g_1 = \{u_5, u_1, u_6\}$. For p_2, GREEDYINIT obtains $g_2 = \{u_6, u_5, u_1\}$. After these two assignments, reviewers u_1, u_5, and u_6 have each reached the load limit $L = 2$ and therefore cannot be used for p_0; the remaining feasible selections yield $g_0 = \{u_0, u_4, u_2\}$. The resulting allocation $A_0 = \{(p_1, g_1), (p_2, g_2), (p_0, g_0)\}$ satisfies all MPRA constraints. Since all paper keywords are covered, $F(A_0) = 1$, and HGES initializes $A^\star \leftarrow A_0$ and $LB^\star \leftarrow 1$. This toy instance is used only to illustrate the GREEDYINIT procedure.

I ADDITIONAL EXPERIMENTAL SETUP DETAILS

Datasets. Table 4 summarizes the statistics and experimental roles of the four datasets.

Table 4: Dataset statistics and experimental roles.

Dataset	#Nodes	#Edges	Prepared Nodes	Role
DBLP	3,802,717	26,029,344	51,464	Academic allocation
AMiner	1,476,093	4,143,559	46,438	Academic allocation
Gowalla	196,591	950,327	38,054	Graph stress test
Brightkite	58,228	214,078	25,756	Graph stress test

Instance construction. For DBLP and AMiner, nodes represent authors and edges represent historical co-authorship relations. Reviewer expertise is extracted from historical publication records preceding the corresponding target year. After tokenization, stopword removal, and TF–IDF weighting, high-weight terms are retained as reviewer expertise keywords. Up to five valid terms from each target-paper title are used as the required research topics of that paper. A reviewer is considered keyword-relevant to a paper if the reviewer shares at least one expertise keyword with the paper. Candidate reviewers are additionally required to satisfy the publication-history condition and not be an author of the target paper. For each paper, the top 30 keyword-relevant reviewers ranked by TF–IDF similarity are retained. The reviewer universe of a multi-paper instance is obtained from the union of the paper-level candidate sets and is capped at 200 reviewers. Social distances are evaluated on the complete underlying network rather than on the retrieved candidate subgraph. Consequently, the exactness results reported in the paper refer to the MPRA instances induced by the retrieved candidate universe.

Gowalla and Brightkite contain friendship and check-in information rather than peer-review records. We therefore use them only as graph-based stress tests. Check-in locations are mapped to geographic regions and treated as node keywords, while friendship relations define the social graph. The generated tasks retain the keyword-coverage, reviewer-load, and group-level social-distance structure of MPRA.

The prepared DBLP and AMiner collections each contain 1,200 target papers, while the prepared Gowalla and Brightkite collections each contain 1,200 allocation tasks.

Metrics. For computational efficiency, we report wall-clock running time and the number of expanded search nodes. Upper-bound pruning statistics are also recorded when applicable. For IESA, index construction time and memory consumption are measured separately from online search time.

Hardware and software. All experiments were conducted using Python 3.10 on a Windows platform with an Intel processor providing 16 logical CPUs and 15.72 GB of main memory.

J Additional Experimental Results

Sensitivity to the reviewer-group size r. Table 5 reports the SolvedRate under different reviewer-group sizes r, complementing the social-distance results of Table 6 in this appendix. Larger groups require more reviewer selections to be coordinated per paper under all group-level feasibility conditions, which generally increases the difficulty of exact optimization. The effect is particularly evident on the stress-test datasets: Gowalla has no solved instance at $r = 5$ or $r = 7$, and Brightkite has no solved instance at $r = 7$. DBLP shows a substantial decline from 90% at $r = 3$ to 10% at $r = 6$, while the higher value at $r = 7$ (30%) reflects the non-monotonic variation that arises because the settings are generated as separate instances rather than as paired modifications of a single instance.

Table 5: Solved rate (%) under different reviewer-group sizes r.

Dataset	$r = 3$	$r = 4$	$r = 5$	$r = 6$	$r = 7$
DBLP	90	80	60	10	30
AMiner	90	90	40	70	50
Gowalla	80	70	0	10	0
Brightkite	50	70	30	10	0

Sensitivity to the reviewer-load limit L and the number of papers $|P|$. Tables 7 and 8 report the SolvedRate under different reviewer-load limits L and different numbers of papers $|P|$. The observed behavior with respect to L is less monotonic. For a fixed MPRA instance, a larger reviewer-load limit relaxes the cross-paper capacity constraint; however, since the reported settings are based on separately generated instances, the measured SolvedRate also reflects differences in keyword demand, candidate overlap, and social structure. The results therefore show that practical search difficulty is jointly determined by reviewer capacity and the structure of the generated allocation instance. Increasing the number of papers introduces more coupled reviewer-group decisions under the shared reviewer-load constraint. The trend is most apparent on AMiner and Brightkite: AMiner decreases from 100% SolvedRate at $|P| = 12$ and $|P| = 16$ to 60% at $|P| = 28$, while Brightkite decreases from 90% at $|P| = 12$ to 30% at both $|P| = 24$ and $|P| = 28$. These results indicate that cross-paper coupling becomes an increasingly important source of exact-search difficulty as the allocation instance grows.

Effect of Rareness-Aware Greedy Initialization and exact search. Table 9 reports the full comparison behind Section 5.3.1.

Effect and overhead of IESA. Tables 10 and 11 report the online running time of HGES and HGES+IESA under the default configuration and the offline memory and construction overhead of IESA.

Table 6: Solved rate (%) under different social-distance thresholds k; results for group size r are in Table 5.

Dataset	k=1	k=2	k=3	k=4	k=5
DBLP	100	90	80	20	10
AMiner	100	100	80	70	20
Gowalla	90	50	50	0	0
Brightkite	100	40	30	0	0

Table 7: Solved rate (%) under different reviewer-load limits L.

Dataset	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$
DBLP	90	80	90	90	80
AMiner	100	100	90	90	100
Gowalla	60	100	90	60	60
Brightkite	70	60	80	80	70

Table 8: Solved rate (%) under different numbers of papers $|P|$.

Dataset	$\|P\| = 12$	$\|P\| = 16$	$\|P\| = 20$	$\|P\| = 24$	$\|P\| = 28$
DBLP	90	80	80	90	70
AMiner	100	100	80	80	60
Gowalla	80	30	50	20	28
Brightkite	90	70	70	30	30

Table 9: Effect of Rareness-Aware Greedy Initialization and subsequent exact search. FR is FeasibleRate for GREEDYINIT only; SR is SolvedRate for HGES.

	DBLP			AMiner			Gowalla			Brightkite		
Method	MinCov	Time	FR/SR	MinCov	Time	FR/SR	MinCov	Time	FR/SR	MinCov	Time	FR/SR
GREEDYINIT only	0.485	0.76	86.7%	0.481	0.10	90.0%	0.336	0.30	63.3%	0.240	0.05	50.0%
HGES	0.489	4.27	90.0%	0.481	0.38	90.0%	0.340	202.66	66.7%	0.255	366.67	73.3%

Table 10: Online running time of HGES and HGES+IESA under the default configuration.

Dataset	HGES (s)	HGES+IESA (s)	Speedup
DBLP	4.27	1.63	2.62×
AMiner	0.378	0.228	1.66×
Gowalla	202.66	190.67	1.06×
Brightkite	366.67	362.60	1.01×

Table 11: Offline memory and construction overhead of IESA.

Dataset	Total index (GB)	k-hop index (GB)	Build time (min)
DBLP	0.89	0.75	5.34
AMiner	0.34	0.22	0.57
Gowalla	2.88	2.83	3.15
Brightkite	0.56	0.53	0.32