A Survey of Subgraph Optimization for Expert Team Formation*†

MAHDIS SAEEDI, School of Computer Science, University of Windsor, Windsor, Canada
HAWRE HOSSEINI, Thomson Reuters, Toronto, Canada
CHRISTINE WONG, School of Computer Science, University of Windsor, Windsor, Canada
HOSSEIN FANI✉, School of Computer Science, University of Windsor, Windsor, Canada

Expert Team Formation is the search for gathering a team of experts who are expected to collaboratively work towards accomplishing a given project, a problem that has historically been solved in a variety of ways, including manually in a time-consuming and bias-filled manner, and algorithmically within disciplines like social sciences and management. In the present effort, while providing a taxonomy to distinguish between search-based versus learning-based approaches, we survey graph-based studies from the search-based category, motivated as they comprise the mainstream. We present a unifying and vetted overview of the various definitions in this realm, scrutinize assumptions, and identify shortfalls. We start by reviewing initial approaches to the Expert Team Formation problem to lay the conceptual foundations and set forth the necessary notions for a more grounded view of this realm. Next, we provide a detailed view of graph-based Expert Team Formation approaches based on the objective functions they optimize. We lay out who builds on whom and how algorithms have evolved to solve the drawbacks of previous works. Further, we categorize evaluation schemas and elaborate on metrics and insights that can be drawn from each. Referring to the evaluation schemas and metrics, we compare works and propose future directions.

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Social recommendation**; • **Human-centered computing** → **Social recommendation**; • **General and reference** → Surveys and overviews.

Additional Key Words and Phrases: Subgraph Optimization, Expert Team Formation, Social Information Retrieval

1 Introduction

Algorithmic search for expert teams, also known as Expert Team Formation, aims to automate forming teams of experts whose combined skills, applied in coordinated ways, can successfully solve difficult tasks [5, 108]. Some examples include a research team whose success can be measured by publications and citations in the scientific community, or the next blockbuster *'sci-fi'* movie with a touch of *'drama'* in the entertainment industry. Expert Team Formation can also be seen as social information retrieval (Social IR) where the right group of experts are searched for and hired to solve the task at hand [65, 66]. Successful teams have firsthand effects on creating an organizational performance in academia [84], manufacturing [7], law [99], freelancing [4] and the healthcare sector [13], among others. Forming a successful team whose members can effectively collaborate and deliver

*We publicly released the artifacts in preparing this survey at `https://github.com/fani-lab/OpeNTF/tree/main/docs/survey/graph`.
†This survey includes appendices for further details.

Authors' Contact Information: Mahdis Saeedi, School of Computer Science, University of Windsor, Windsor, Ontario, Canada; e-mail: msaeedi@uwindsor.ca; Hawre Hosseini, Thomson Reuters, Toronto, Ontario, Canada; e-mail: hawre.hosseini@thomsonreuters.com; Christine Wong, School of Computer Science, University of Windsor, Windsor, Ontario, Canada; e-mail: wong93@uwindsor.ca; Hossein Fani✉, School of Computer Science, University of Windsor, Windsor, Ontario, Canada; e-mail: hfani@uwindsor.ca.

ACM 1557-7341/2025/5-ART
https://doi.org/10.1145/3737455

the outcomes within the constraints such as a planned budget and timeline is challenging due to the immense number of candidates with various backgrounds, personality traits, and skills, as well as unknown synergistic balance among them; not all teams with *best* experts are successful [103]. Further, the team members need to be *adaptive* to adjust to the environment and facilitate coordination [5].

Traditionally, teams were formed manually relying on human experience and instinct, which is a tedious, error-prone, and suboptimal process due to *i*) hidden personal and societal biases, *ii*) a multitude of criteria to optimize, and *iii*) an overwhelming number of candidates, among other reasons. On the one hand, the formation was heavily influenced by the team formers' subjective opinions that already inherit hidden unfair societal biases, largely ignoring the *diversity* in recommended experts and resulting in discrimination and reduced visibility for an already disadvantaged group [62], disproportionate selection of popular experts [51], and over/under-representation and racial/gender disparities [76]. Additionally, since this process involves a multitude of criteria including project importance, budget, time constraints and team size limitations, the decision-making is all the more difficult. On top of these, candidates for the formation of an expert team should ideally be examined based on several individual and relative factors such as technical abilities, availability, individual cost, productivity, behavior, personality traits, knowledge, negotiation skills, proactivity, and communicability, among others, which makes manual Expert Team Formation on a large scale almost impossible. It has long been well-established in the fields of psychology and cognitive science that human individuals have limited capacity to store and process information, making large-scale computation beyond their capacity [9, 87]. Also, research in industrial engineering [82] has shown that the *size* and *variety* of components within a task, like the task of forming a team, are crucial factors in determining its complexity, which are limited for human performance. The limitations of human intuition and cognitive biases have been further shown in complex decision-making tasks [14, 61]. Such studies necessitate an algorithmic approach, at least as a supplement, when dealing with tasks of large-scale size and with diverse components like Expert Team Formation [39]. For instance, major global corporations with thousands of employees invest in developing expert recommender systems, such as IBM's SmallBlue [21], which exploits corporate email correspondences to assist in locating experts among the company's broad geographically distributed presence in various countries for a given project team.

In an effort to automate forming expert teams, researchers in different disciplines, such as operations research (OR) [11, 23, 34, 36, 108], psychology [56, 60, 75], management [3], engineering [10, 11], social network analysis [59, 70, 79, 101], and recently, artificial intelligence (AI) [15, 16, 44–46, 97] have long been endeavouring to find algorithmic solutions. This has resulted in a rich body of various approaches grounded in computational and social science theoretical and conceptual frameworks wherein the problem definition of Expert Team Formation remains the same essentially, while it has been referred to by such other names as team allocation, team selection, team composition, and team configuration. Despite the substantial number of algorithmic approaches, there is, however, yet to be a comprehensive survey with comparative analysis and critical reviews on approaches' applicability in real-world scenarios, especially when each comes with a domain-specific method and benchmark dataset with no standard implementation, incapable of accommodating different use-cases, not to mention the codebases and details are scarcely publicly available.

1.1 Existing Surveys

To the best of our knowledge, Wang et al. [106] and Juárez et al. [30] are the only surveys on *algorithmic* Expert Team Formation. Wang et al. is a pioneering survey that presents a comparative study of Expert Team Formation algorithms on a reproducible, publicly available platform. They re-implemented at-the-time (then) state-of-the-art algorithms and evaluated them on four real-world datasets under the same experimental settings. However, Wang et al.'s work has the following limitations: *i*) lacks an analysis of Expert Team Formation from the perspective of problem setting and the challenges in each setting like optimizing multiple objectives or the fact that experts can

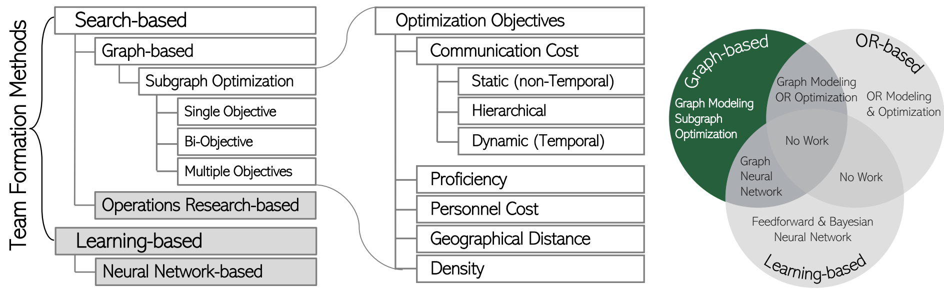


Fig. 1. (Left) A taxonomy of the computational Expert Team Formation methods. (Right) A Venn diagram to highlight overlapping areas where operations research and learning-based methods leverage an expert graph for their optimizations. Gray areas are outside the scope of the current survey.

hold attributes that impact Expert Team Formation like workload capacity; *ii*) only a limited number of methods using merely one optimization objective (communication cost), are covered, while other optimization objectives such as density or geographical proximity or methods that consider multiple optimization objectives in tandem (e.g., bi-objective optimizations) are overlooked; *iii*) it has been almost a decade since its publication, missing prominent state-of-the-art approaches to date. More recently, Juárez et al. [30] has introduced a taxonomy with a special focus on the modeling aspects of the problem at a conceptual level where teams are formed either through a match-making process between individual candidates and skills or via a search for a coherent community in the collaborative networks of candidates. However, they forgo comparative technical study of the methods, especially *i*) the optimization objectives when forming an optimum team, *ii*) strategies to evaluate the performance of the methods, and *iii*) experimentation setups.

Expert Team Formation has also been researched in social ecology, a field that studies the relationships between people and their environment and the interdependence of people and institutions [91, 102], and Science of Team Science (SciTS), which aims at understanding and enhancing the outcomes of large-scale collaborative *science* teams. Such studies are, however, *non*-computational where no algorithmic realization of such theoretical foundations has been explored. In this survey, we consider such studies out of scope and only focus on algorithmic graph-based methods to form teams of experts applicable not only in science, as in SciTS, but also in any collaborative domain like the entertainment industry [2, 73] or crowdsourcing [28, 50].

1.2 Our Contributions and Scope Disclaimer

In this survey, we present a novel taxonomy from a computational perspective, as shown in Figure 1. Expert Team Formation approaches can be distinguished based on the way optimizations are performed: *i*) search-based, where the search for an *almost surely* successful team (optimum team) is carried out over the subgraphs of an expert graph using subgraph optimization methods, or it is performed on subsets of experts as variants of the set cover problem [18, 56, 88] using operations research (OR) techniques including integer linear/nonlinear programming, and *ii*) learning-based, where machine learning approaches are used to learn the distributions of experts and skills in the context of previous (un)successful teams in order to draw future successful teams.

Within the search-based category, operations research, by and large, optimizes the mutually independent selection of experts, overlooking the organizational and collaborative ties among individuals [23, 34, 69, 88]. However, graph-based methods rely on the premise that a team is inherently relational and is a property of the interaction among the experts and how effectively they can collaborate. Following Cheatham et al. [8] and the seminal work by Lappas et al. [79], graph-based methods took the stage and became canonical in the literature. Recently, learning-based methods have been proposed to bring efficiency while enhancing efficacy due to their iterative and online learning procedure [15, 45]. Nonetheless, since effective collaboration among experts is a crucial aspect of Expert Team Formation, graphs provide a useful way to represent the connections between experts, making them valuable tools for studying Expert Team Formation. Therefore, we can observe the synergistic integration of expert graph in operations research [20, 36] and learning-based methods [44, 46, 97]. As seen in Figure 1 (right), there is an overlap between graph-based and operations research-based methods, where optimization functions have been defined based on linear or nonlinear equation of Boolean variables representing edges on the expert graph [36]. Additionally, learning-based methods utilize the expert graph to learn vector representations of skills using graph neural networks (GNNs), which helps reduce the complexity of neural models at the input layer [27]. Specifically, graph neural networks (GNN) [43, 54] provide an effective and efficient general framework for solving graph analytics problems by converting a graph into a low-dimensional vector space while preserving its graph-structured information. Having demonstrated strong performances across a wide range of problems, including natural language processing. [35], knowledge graphs [55], recommender systems [48], graph neural networks are gradually finding their application in Expert Team Formation [27]. Successful as they are, learning-based literature is in its early stages with little work, as shown in Figure 1.

This survey pertains to the graph-based Expert Team Formation algorithms, that is, those that employ graphs to model the experts' collaboration ties followed by subgraph optimization algorithms, as they comprise the mainstream body of research. While some OR-based works like Camelo et al. [36] model the dataset as a graph structure, they opt for linear/non-linear integer/real programming methods as opposed to subgraph optimization. In this survey, we include works that not only model the data as graphs but also apply subgraph optimization methods. We exclude works that are based on operations research and learning-based methods, as they differ fundamentally from subgraph optimization algorithms. We recognize the importance of these areas and the wealth of work they include, but a thorough analysis of them is beyond the scope of this work and merits separate surveys. We identify any existing or ongoing comparative studies within this scope, explain how expert graphs are formed, discuss the objectives for an optimum team, subgraph optimization algorithms and their complexities followed by workarounds for efficiency, and finally, elaborate on evaluation metrics used. These concepts have not been extensively explored in previous surveys, making our research original and innovative. A comprehensive *technical* review of the proposed methods in terms of their formalization, evaluation methodology, and benchmark test beds was needed to foster future research in the field.

After screening 126 papers addressing Team Formation problems using computational and non-computational models, we present a comprehensive overview of 17 seminal graph-based research papers on the Expert Team Formation problem within the scope of our survey, including 18 unique objectives, to be optimized via 63 subgraph optimization algorithms considering variations of their exact algorithms and the heuristics they used to address the efficiency of their algorithms. Our survey brings forth a unifying and vetted methodology to review the various definitions of the notions, criticizes assumptions and comparative benchmarks, and points out shortfalls to smooth the path for future research directions. It targets the information retrieval (IR) and recommender systems (RecSys) research communities to propose new Expert Team Formation solutions and evaluate their effectiveness compared to existing methods and on datasets from various domains. Further, having regard to the unified comparative analysis, organizations and practitioners can compare different models and readily pick the most suitable one for their application to form teams of experts whose success is almost surely guaranteed. Below, we summarize our contributions in this survey:

(1) We study graph-based approaches to the problem of Expert Team Formation utilizing a taxonomy of computational methods. To foster standards and conventions, which literature of this field lacks, we set forth a unified set of notations to formalize the problem and its different aspects conveniently. Our notations are designed to be a *minimal superset* of varied notations in papers while respecting their specifications. Our unified

notations help with finding overlaps or correlations among formulas, which can improve understanding, simplify analysis, validate models, and avoid redundancy, while allowing seamless integration of novel ideas by reusing prior notations.

(2) We provide a detailed study of the graph-based optimization techniques used for tackling the Expert Team Formation problem. In doing so, we adopt a chronological view that reflects the advancements over time, drawing upon the drawbacks of each method and how the subsequent works have sought to overcome them.
(3) While introducing the available datasets in this field and their drawbacks, we perform a comparative analysis of the performance of the various techniques in the field of graph-based Expert Team Formation. Further, we show how the field suffers from the void of proper benchmarking, baselining and performance comparisons.
(4) Based on various open issues and challenges in the literature, we categorically set forth four major lines of research as future directions, namely *i*) fairness-aware, *ii*) time-sensitive, *iii*) multi-factorial and *iv*) open-source implementation for Expert Team Formation. For each line of work, we thoroughly outline the shortcomings in the literature and value solutions. We believe that these future directions deserve more attention from the academic sphere.

2 Problem Formalization

The graph-based modeling approaches tackle the Expert Team Formation problem through defining subgraph optimization problems on a graph where the different aspects of real-world teams and the Expert Team Formation process are explored. In this section, we formalize subgraph optimization problems in a unified framework with integrated notations for better readability and fostering conventions in this realm. Table 1 shows the summary of the notations.

2.1 Notations and Definitions

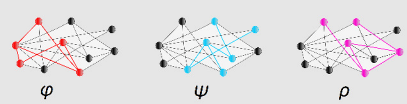
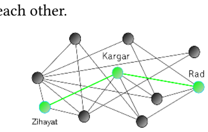
Definition 2.1. **Expert Graph**. In the Expert Team Formation problem, a skillful individual who can be a candidate for a team is referred to as an *expert*. In graph-based approaches, an expert graph, also known as an expert collaboration network, is an undirected weighted and attributed graph denoted by $G = (V, A, E, w)$, where V represents a set of experts as nodes and A is the set of attributes associated to each expert as node attributes.

For instance, the nodes V could be researchers in an academic network (e.g., DBLP), software developers in an open-source network (e.g., GitHub), inventors in patents (e.g., USPTO) or cast and crew in moving picture industry (e.g., IMDB). The node's attributes generally express the capabilities of an expert from different perspectives, including:

- Expert's skillset S_v, which is a subset of the entire set of skills S, i.e., $S_v \subseteq S$, that represents the expertise of the expert v such as research domains for a researcher like {*'graph theory'*, *'social network analysis'*, ...}, programming languages for a developer like {*'python'*, *'java'*, ...}, patent classes for an inventor like {*'textile'*, *'engineering'*, ...}, or movie genres for an actress like {*'sci-fi'*, *'comedy'*,...};
- Expert's level of expertise per skill (also referred to as *authority* per skill) $x_{v_s} \in \mathcal{R}^+$ given a skill $s \in S$, that indicates the level of competence or excellence of the expert v in the skill s among her peers like *'the number of citations'* to the publications related to the skill s;
- Expert's level of authority a_v that indicates the *overall* competence or excellence of the expert among her peers, like the *'h-index'* of a researcher in an academic network. This is different from level of expertise, which refers to competence per skill;
- Expert's hiring budget b_v, that can be the requested or allocated salary for an expert position in a project; and,
- Expert's capacity c_v, the maximum number of responsibilities that an expert can handle.

The subset of skills an expert v has along with her level of expertise in the skills is denoted by $S_v \times \mathcal{R}^+$. Inversely, for a given skill $s \in S$, $V_s = \{v \in V : (s, x_{v_s}) \in S_v \times \mathcal{R}^+\}$ denotes the set of all experts owning it. A rare

Table 1. Summary of the notations used in this survey. Examples are fictional.

Notation	Description	Example
$G = (V, A, E, w)$	Expert Graph, an undirected weighted attributed graph of experts (nodes) V, attributes A, edges E, and weight function w.	A co-authorship graph where two authors are connected if they publish at least one paper together.
$V = \{v_i\}$	The set of all experts (nodes) enumerated by i.	Computer science researchers with at least one paper.
v	An expert (node) of the expert graph.	*'Lappas'*.
$A_v = (S_v, a_v, b_v, c_v)$	An attribute for an expert (node) v in the expert graph that represents the expert's skills along with the level of expertise in each skill ($S_v \times \mathcal{R}^+$), the level of authority a_v, budget b_v and capacity c_v.	$A_{v:\text{'Lappas'}} = (S_v : \{(s$: *'team formation'*, $x : 10)$, $(s$: *'recommender systems'*, x:5$)\}$, a_v:24, b_v:\$150, 000, c_v:5).
b_v	Required budget to hire expert v or her requested salary.	$b_{v:\text{'Lappas'}}$ = \$150, 000.
c_v	Maximum capacity in terms of the number of responsibilities or tasks that could be handled by expert v.	$c_{v:\text{'Lappas'}} = 5$.
a_v	The level of authority of expert v.	Researcher's h-index.
$S = \{s_j\}$	The set of all skills enumerated by j.	Research domains in computer science as in the ACM/IEEE computing classification system like {*'graph theory'*, *'social network analysis'*, *'operations research'*, ...}.
$S_v \times \mathcal{R}^+ = \{(s, x_{v_s}) : s \in S_v, x_{v_s} \in \mathcal{R}^+\}$	The subset of skills the expert v has along with the level of expertise for each skill (x_{v_s}).	$S_{v:\text{'Lappas'}}$={(s: *'team formation'*, x:10), (s: *'recommender systems'*, x: 5)}.
x_{v_s}	The level of expertise in skill s for an expert.	Number of citations to the publications related to the skill s.
$V_s \subseteq V$	The subset of experts that are skillful for s.	$V_{s:\text{'team formation'}}$ = {*'Lappas'*, *'Kargar'*}.
$s^\dagger$	The rarest skill with the fewest number of skillful experts, i.e., $\lvert V_{s^\dagger}\rvert = min_{s\in S}\lvert V_s\rvert$.	$s^\dagger$: *'theory of fairness'*.
E	The set of all edges of the expert graph.	The set of all collaborations.
$e_{v,v'}$	A weighted edge of a graph that represents a connection between a pair of experts v and v'.	Co-authorship in a publication.
$w_\uparrow(e_{v,v'})$	The weight of the edge e, desired to be high.	The number of successful joint publications between experts v and v'.
$w_\downarrow(e_{v,v'})$	The weight of the edge e, desired to be low.	Geographical distance between experts v and v'.
p	A project yet to be successfully accomplished.	Research project on *'fairness-aware team formation'*.
$S_p = \{(s, x_{v_s}, l_s, u_s)\} \subseteq S \times \mathcal{R}^+ \times \mathcal{N} \times \mathcal{N}$	The predefined subset of skills required by the project p. For each skill $s \in S_p$, l_s and u_s denote the lower and upper bound on the number of required experts, respectively, and x_{v_s} is its required level of expertise.	{(s: *'graph theory'*, x_{v_s}:100, l_s:1, u_s : 2), (s : *'theory of fairness'*, x_{v_s} : 1, l_s : 2, u_s : 3)}.
l_s	The lower bound on the number of required experts for skill s.	$l_s = 3$ for a research project that needs at least 3 postdoctoral fellows.
u_s	The upper bound on the number of required experts for skill s.	$u_s = 6$ for a research project that needs at most 6 postdoctoral fellows.
$V_p \subseteq V$	The subset of experts in the expert graph whose attributes include at least one of required skills S_p for project p. It represents the subset of experts that collectively cover required skills for p.	$V_{p:\text{'fairness-aware team formation'}}$ = {*'Lappas'*, *'Schwiegelshohn'*}.
$G[V_p] = (V_p, A_p, E_p, w)$	An induced subgraph with the experts that represents the prediction for an optimal subset of experts who will almost surely accomplish the project p successfully.	
$\mathcal{P}(G)$	Power set of a graph including all subgraphs of a graph	All possible teams that can be formed.
$\varphi, \phi, \rho, \chi, \psi(G[V_p])$	Optimization functions that seek to optimize overall communication cost (φ), proficiency (ϕ), personnel cost (ρ), geographical distance (χ) or density (ψ) within a subgraph $G[V_p]$ of the expert graph G.	
$d(v, v')$	Distance between experts (nodes) v and v' in the expert graph G defined as sum of edge weights of the shortest path between v and v'.	d(*'Zihayat'*, *'Rad'*) = 2, both of them have joint papers with *'Kargar'*, but never with each other. 

skill $s^\dagger$ is the one possessed by the fewest number of experts, i.e., $|V_{s^\dagger}| = min_{s \in S}|V_s|$. In total, given an expert v, her attribute is a tuple $A_v = (S_v, a_v, b_v, c_v)$. For example, the attribute for the expert $v =$ *'Lappas'* would be $A_v = (S_v : \{(s :$ *'team formation'*$, x_{v_s} : 10), (s :$ *'recommender systems'*$, x_{v_s} : 5)\}, a_v : 24, b_v : \$150,000, c_v : 5)$ whose levels of expertise in *'team formation'* and *'recommender systems'* are 10 and 5, respectively, based on *'the number of citations'* to the expert's publications in each of the research domains, and 24 being the expert's level of authority based on *'h-index'*. As will be explained in Section 3.1 (Expert Graph Construction) and Section 4.1 (Datasets), an expert's skills S_v may not be self-declared and should be inferred via an assumption based on an underlying benchmark dataset. For instance, in the academic network, an expert is an author of one or more publications and her skills can be set to the words in the titles of her publications after removing the stopwords [17, 70, 79, 80].

Next, E represents the connection between the experts as graph edges and $w(e_{v,v'}) : E \rightarrow \mathcal{R}$, is the weight function that maps an edge $e_{v,v'} \in E$ between two experts v and v' to its weight. Edges can represent experts' previous collaborations, followership in a social network, or spatial proximity, and their weights $w(e_{v,v'})$ can be based on the number of joint projects, number of recent communications, or the geographical distances in kilometres between experts v and v'. In graph-based methods, the weight function is the primary element of the optimization process based on which an optimization function is to be maximized or minimized. For instance, experts who have collaborated the most in the past can communicate the best (technically, have the lowest communication cost) [1, 12, 17, 24, 40–42, 70–72, 79]; therefore, they are the optimum choice for an *almost surely* successful team. In this case, higher edge weights between experts are desired and indicated by $w_\uparrow(e_{v,v'})$. In contrast, or rather meanwhile, the less geographical distance, the better/easier communication can be observed [33] such that lower edge weights are desired and shown as $w_\downarrow(e_{v,v'})$.

Definition 2.2. **Experts Distance.** The distance between a pair of experts v and v' in the expert graph G, $d(v, v')$, is defined as the sum of edge weights of the shortest path between v and v'. Given the weight function $w(e_{v,v'})$, the shortest path from v to v' is the path $(v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_n)$ including n intermediary experts, where $v_1 = v$ and $v_n = v'$, that minimizes the sum in $\sum_1^{n-1} w(e_{v_i,v_{i+1}})$ over all possible n. While Floyd-Warshall algorithm [57] solves all pairs shortest paths in $\Theta(V^3)$, a shortest path, for efficiency, can be calculated stochastically based on a random walk [31], that is, random selection of experts (nodes) on a path (single steps of a walk) from v to v' based on a probability distribution, e.g., uniformly where the adjacent experts are selected with the same probability, or non-uniformly such as in relation to the edge weight $\mathrm{P}(v_{i+1}|v_i) \propto w(e_{v_i,v_{i+1}})$ assuming Markov property. Further, the distance between an expert v and a subgraph $G[V_p], V_p \subseteq V$ is defined as:

$$d(v, V_p) = \min_{v' \in V_p} d(v, v') \tag{1}$$

Definition 2.3. **Project Team.** From an organizational perspective, the search for an expert team is determined for a given project p with a predefined required subset of skills $S_p \subseteq S$. In graph-based methods, a team is estimated by an induced subgraph of the expert graph G, denoted by $G[V_p]$, with nodes $V_p \subseteq V$ representing the team's members such that for each required skill $s \in S_p$ there should be *at least one* expert v whose skill subset S_v includes s. Therefore, the team's members are required to *collectively* cover the project's required subset of skills S_p, that is, $\forall s \in S_p, \exists v \in V_p : s \in S_v$ and so $S_p \subseteq \bigcup_{v \in V_p} S_v$. Further, an organization may limit the number of experts in a team per each required skill of a project along with a predefined level of expertise. For skill $s \in S_p$, we denote l_s and u_s for lower and upper bound on the number of required experts, respectively, and x_{v_s} is its required level of expertise. Hence, the predefined required subset of skills $S_p \subseteq S$ for a project p becomes a set of tuples $(s, x_{v_s}, l_s, u_s) \in S_p$ where $S_p \subseteq S \times \mathcal{R}^+ \times \mathcal{N} \times \mathcal{N}$.

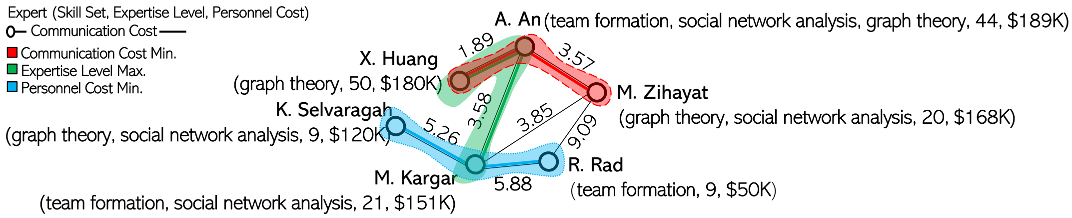


Fig. 2. An example of subgraph optimization using different objectives for the Expert Team Formation problem. Best viewed in color.

For example, to form a research team for a project on '*fairness-aware team formation*' with a required subset of skills (research domains) $S_p = \{$'*graph theory*', '*theory of fairness*'$\}$ with one or two senior researchers having more than 100 publications in '*graph theory*', and 2 to 3 junior researchers with at least one publication in '*social network analysis*', S_p would become $S_p = \{(s$:'*graph theory*', x_{v_s}: $100, l_s$: $1, u_s$: $2)$, $(s$:'*social network analysis*', x_{v_s}: $1, l_s$: $2, u_s$: $3)\}$.

Once the expert graph G is created and the project p is defined, the next step involves the search for a subgraph $G[V_p]$ whose experts (nodes) V_p would form a team. This step faces two important challenges: *i*) there may be more than one solution, that is, two or more subgraphs whose experts collectively cover the required skill subset of the project p, and *ii*) the search over all possible subgraphs of G is computationally prohibitive in medium to large-scale expert graphs. A common greedy method would involve selecting those subgraphs whose experts have had many previous successful collaborations. However, this is not always possible due to, for example, the unavailability of such experts. Also, there are many instances of successful collaborations in teams whose members meet for the first time (no prior collaborations). For instance, '*Barolli*' and '*Xhafa*''s first collaboration in 2006 [107] ignited their long-lasting successful collaborations *thereafter* on 220+ research publications until 2019. To find an optimum subgraph (team) of experts efficiently (fast), optimization objectives have been proposed based on varied assumptions to reduce the search space. In optimization, subgraphs that maximize or minimize a selected objective or a combination of objectives in tandem are of interest. In the following, we formalize such optimization objectives and lay out their details.

2.2 Subgraph Optimization Objectives

Given a graph G and a project p, the task is to find an *optimal* subgraph $G[V_p]$ as the recommended team, given an objective function with a set of constraints. Examples of such objectives are a team of the highest expertise, minimum geographical distance or communication cost, or all in tandem summing over *all* its members. As seen in Figure 2, given an organization aims to hire a team of experts for a project p requiring a subset of skills S_p=$\{$'*team formation*', '*social network analysis*', '*graph theory*'$\}$, with candidate experts connected in an expert graph based on their joint project teams in the past and the edges being weighted based on '*the more joint project teams the less communication cost*' (Equations 20 or 19), different optimum subgraphs are selected depending on an objective we seek to optimize; (red) communication cost (Equation 3): $\{$'*Huang*', '*An*', '*Zihayat*'$\}$, (cyan) personnel cost (Equation 14): $\{$'*Rad*', '*Kargar*', '*Selvarajah*'$\}$, or (green) expertise level (Equation 10): $\{$'*Huang*', '*An*', '*Kargar*'$\}$. Personnel costs are included for illustrative purposes only. Table 2 shows an overview of the objectives studied in the literature, which will be elaborated in the following.

2.2.1 *Minimizing Communication Cost* (φ)

A key team performance indicator is how effectively experts communicate, and is measured by a metric well-known as communication cost; a lower communication cost in a team indicates easier communication, better

Table 2. An overview of objectives employed in subgraph optimizations for Expert Team Formation problem.

			Communication Cost						Proficiency						
	Year	Hybrid	Diameter	MST	Leader Distance	Sum of Edge Weights	Random Walk	Sum of Distance	Trust Score	Expertise Level	Connector Authority	Skill Holder Authority	Personnel Cost	Geographical Distance	Density
Lappas et al. [79]	2009		✓	✓											
Li et al. [80]	2010			✓											
Farhadi et al. [24]	2011	■	✓							✓					
Kargar et al. [70]	2011				✓			✓							
Datta et al. [17]	2012		✓	✓											
Kargar et al. [71]	2012	■						✓					✓		
Gajewar et al. [58]	2012														✓
Kargar et al. [72]	2013	■	✓					✓					✓		
Kargar et al. [1]	2013	■	✓					✓					✓		
Rangapuram et al. [93]	2013	■												✓	✓
Zihayat et al. [40]	2014	■						✓		✓			✓		
Huang et al. [29]	2016				✓			✓							
Zihayat et al. [41]	2017	■				✓					✓	✓			
Zihayat et al. [42]	2018	■				✓					✓	✓	✓		
Juarez et al. [68]	2018	■								✓					✓
Nemec et al. [31]	2021						✓								
Selvarajah et al. [33]	2021	■						✓	✓					✓	

understanding and collaboration among team members, and hence, more likely for that team to succeed. For instance, a friend or a long-life colleague can be reached out to quickly and effortlessly through various media like messaging services or direct phone calls, whereas a more formal request or an appointment through administrative steps is needed to meet a manager at a higher level of an organization's hierarchical network structure like a dean or chair of a department. Hence, the communication cost of the first case is much less than the second one. Communication cost has been the prominent objective to be minimized among the graph-based methods and is quantified based on *i*) previous joint (successful) collaborations [1, 17, 29, 40–42, 70, 79], *ii*) the time of the last collaboration [33], as well as *iii*) the similarity between cultural backgrounds and languages [33]. Though, communication cost has primarily been estimated based on the previous joint collaborations assuming an equality condition (if and only if) between communication cost and team performance (success) where low (minimum) communication cost is the *necessary* and *sufficient* condition for team performance. In other words, when low communication costs in the past lead to past successful collaborations, past successful collaborations would also indicate low communication costs in a yet-to-be-formed team in the future and would yield the team's success. Looking closely at this assumption, communication cost objective becomes a mediator from past teams' successes to signal new successful ones.

Once the edges of the expert graph G are weighed based on communication costs, the optimum team is a subgraph within which the overall communication cost is minimized based on, e.g., the summation of edge weights in the subgraph's diameter [1, 17, 79], its spanning tree [79, 80], or its entirety [41, 42], among others, as detailed below. We identify three types of search algorithms based on communication cost minimization in the literature: non-temporal (static), hierarchical, and temporal (dynamic).

Non-Temporal (Static): Several *static* optimization functions $\varphi : \mathcal{P}(G) \rightarrow \mathcal{R}^{+}; \varphi(G[V_p])$ have been defined to measure the overall communication cost within a subgraph of G among all possible ones $\mathcal{P}(G)$, overlooking the time dimension and the fact that experts' skills and social attributes are constantly changing over time:

- *Sum of distances* (φ_D) [31, 33, 40, 70, 71]:

$$\varphi_D(G[V_p]) = \sum_{v,v' \in V_p} d(v, v') \tag{2}$$

  where $d(v, v')$ is the distance between two experts and should be calculated between all pairs of experts, each contributing to the subgraph's overall communication cost. Using random walks, should $\varphi_D(G[V_p])$ be the lowest, it is expected that a *random* communication between a random pair of v and v' in V_p is also the lowest, and hence, V_p would be an optimum team for the project p.
- *Sum of edge weights* (φ_E) [41, 42] which is defined as:

$$\varphi_E(G[V_p]) = \sum_{e \in E_p} w_{\downarrow}(e) \tag{3}$$

  where $w_{\downarrow}(e)$ is the communication cost based on a pair of experts. Herein, the summation is over all edges of a subgraph, in contrast to Equation 2 where only edges in shortest paths are considered.
- *Diameter* (φ_R) is the largest path among shortest paths between pairs of experts in the subgraph [1, 17, 79]

$$\varphi_R(G[V_p]) = \max_{v,v' \in V_P} d(v, v') \tag{4}$$

  Intuitively, as shown in Figure 3 (a), the diameter signifies the longest distance, and hence, the highest communication cost possible between a pair of experts as a worst-case scenario in a team. Equation 4 is used to search for an optimum team whose worst communication cost is minimal.
- *Cost of the spanning tree* (φ_{MST}) is the sum of the weights of the minimum spanning tree (MST) on the subgraph ($G[V_p]$) [79, 80]:

$$\varphi_{MST}(G[V_p]) = \sum_{e \in MST(G[V_p])} w_{\downarrow}(e) \tag{5}$$

  A minimum spanning tree (MST) is a tree subgraph including all the nodes of a graph (spans all nodes) in such a way that the total edge weights is minimum among all possible tree subgraphs of the graph, as shown in Figure 3 (b). Herein, the subgraph's MST represents the best-case scenario where a team of connected experts has the lowest possible communication cost overall. Equation 5 is used to find the optimal team with the minimum overall communication cost across all subgraphs, represented by their MSTs in the best-case scenarios. Sometimes, as seen in Figure 3 (right), adding extra nodes to a subgraph from outside can lead to forming a tree with a smaller cost. The resulting tree is referred to as the Steiner tree of a subgraph. While a minimum spanning tree spans all nodes of a given subgraph, a Steiner tree spans extra nodes in an effort to reduce the cost.

Hierarchical: On some occasions, a team pertaining to a hierarchical structure is desired, like having a leader who is responsible for intra- and inter-team communications. Leader distance $\varphi_L(G[V_p])$ in such teams is defined as the sum of all distances between the leader L and other members as follows:

$$\varphi_L(G[V_p]) = \sum_{v \in V_p} d(v, L) \tag{6}$$

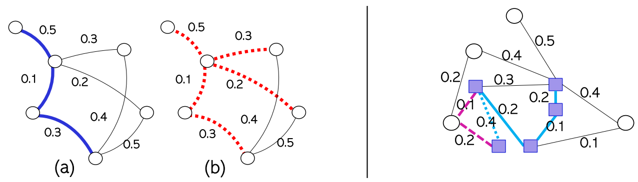


Fig. 3. Left: Minimum subgraph based on (a) diameter (solid blue edges) vs. (b) spanning tree (dashed red edges). Right: Minimum spanning tree vs. Steiner tree. Given a subgraph with square blue nodes, a minimum spanning tree has the minimum cost 0.9. However, adding extra nodes from outside (the circle white nodes) forms a Steiner tree and reduces the minimum cost to 0.8.

In a more general case, levels of hierarchy can be increased to have a team of *sub*teams, in which the team's leader L is in contact with each subteam's leader L_i. Accordingly, hierarchical communication cost [29, 33, 70] is defined as:

$$\varphi_H(G[V_p]) = \sum_{v \in V_i} d(v, L_i) + \sum_i d(L_i, L) \tag{7}$$

where V_i is the set of experts in the i-th subteam and $V_p = \bigcup_i V_i$.

In a cross-functional team, where experts from various seemingly unrelated disciplines are needed and having experts directly work together on a joint task could be challenging, subteams are formed to include experts with related skill sets. For example, a city council may have the mayor as the leader of a team with a lead urban architect, a chief environmental planner, and a chief police officer, each of whom is a leader of subteams of architects, environmental planners, and police officers, respectively. As another example, a research laboratory can have the principal investigator as the lead for a team of postdocs, each leading a subteam of graduate students.
Temporal (Dynamic): Temporal (dynamic) communication cost is based on the fact that the *least* communication cost exists between experts who could maintain many successful collaborations over time until recently or currently; the factor that might have caused the termination of collaboration in the *past* may be an obstacle to re-cooperation *now* or, the experts' interests toward skills might have changed. Further, recent few successful collaborations should outweigh successful ones in the far past. Accordingly, temporal communication cost is defined as the sum of distances with temporal regularization [33]:

$$\varphi_{DT}(G[V_p]) = \sum_{v, v' \in V_p} d(v, v') + \alpha(t - t') \tag{8}$$

where $d(v, v')$ is based on Definition 2.2, $(t - t')$ denotes the time gap since the last (most recent) collaboration for a pair of experts; the longer the time gap the more costly the communication, and α is an attenuation factor.

2.2.2 Maximizing Proficiency (ϕ)

Proficiency of experts indicates the level of expertise in a particular profession or a skill. *Mastery level* [33], *expertise level* [68], *authority* [41, 42] and *trust score* are proposed to measure proficiency based on some criteria, for example, *'h-index'* or *'number of citations'* in academic networks. Given a graph G, the optimization task aims to find the optimum subgraph $G[V_p]$ that maximizes the proficiency objective function ϕ, defined based on the sum of node weights:

- The *mastery level* [33] of a team $G[V_p]$ is the sum over the team members' expertise among all of their skills:

$$\phi_M(G[V_p]) = \sum_{v \in V_p, s \in S_p} x_{v_s} \tag{9}$$

Note that x_{v_s} can be dynamic [33]; an expert becomes more professional in a skill as she gains more experience.

- The *total expertise level* [68] of a team $G[V_p]$ is the value of mastery level normalized by the size of the team:

$$\phi_E(G[V_p]) = \frac{\phi_M(G[V_p])}{|V_p|} \tag{10}$$

where $|V_p|$ is the size of the subgraph $G[V_p]$ that collectively covers the required skills for p.

- The *skill holder authority* is defined in conjunction with *connector authority* where experts holding the required skills may not be directly connected. Therefore, a subgraph $G[V_p]$ representing a team has two types of nodes: skill holders, H, who have the required skills, and connectors, C, who lack the skills, yet connect skill holders. For instance, in real-life scenarios, a reputable project manager (e.g., a professor) may lack the required skills (e.g., a programming language) but can facilitate communication between experts who hold the skills (e.g., programmers). Accordingly, the proficiency of a team is defined through the help of skill holders and connectors separately as [41, 42]:

$$\phi_A(G[V_p]) = \sum_{v \in A} \frac{1}{a_v}; A \in \{H, C\} \tag{11}$$

where the sum of the inverse of authority is minimized instead of maximizing the sum of authorities since it can be jointly employed with other objectives whose minimizations are required, like with communication cost objective under a single optimization function, as explained further in Section 2.3 (Hybrid (Multi-Objective) Optimization).

- The trust score measures the trust between a pair of experts v and v', and is of special importance when two experts have never interacted with one another in the past. It can be calculated based on [33]:
i) *explicit trust score* $\varkappa$, valued between −1 to +1, based on the number of collaborations (edges) between a pair of experts. The trust score is 0 if there has been no collaboration between two experts in the past.
ii) *profile similarity score* F based on the similarity of skills or number of common skills between a pair of experts. This score is rooted in social psychology studies [86], which have shown that people with similar interests tend to trust each other more.
iii) *emotional intelligence index* ϖ, which refers to the ability of an expert to perceive, control and evaluate emotions in oneself and others to make decisions, solve problems, and communicate.
The final trust score between a pair of experts is denoted by $\tau(v, v')$ and is defined as the weighted sum of explicit trust, profile similarity and emotional intelligence index:

$$\tau(v, v') = \alpha_1 \varkappa(v, v') + \alpha_2 F(v, v') + \alpha_3 \varpi(v) \tag{12}$$

where $\sum \alpha_i = 1$. The *collective* trust score of a subgraph $G[V_p]$ is the sum of trust scores over all team members:

$$\phi_\tau(G[V_p]) = \sum_{v, v' \in V_p} (\tau(v, v') + \tau(v', v)) \tag{13}$$

Note that the trust score is not commutative [33]; the trust score of a team is the sum of the trust scores of expert v to v' *and* v' to v for all pairs of experts v and v' in the team.

2.2.3 *Minimizing Personnel Cost* (ρ)

In the real world, experts are compensated financially for their efforts on a given project. The amount of compensation is relative to their expertise level as well as the available budget for the project. Thus, it is crucial to form a team with a reasonable amount of financial compensation based on the project funds. To this end, the personnel cost of a team, denoted by $\rho(G[V_p])$, is the sum of the required budgets to hire team members working on project p [42, 71]:

$$\rho(G[V_p]) = \sum_{v \in V_p} b_v \tag{14}$$

which is to be minimized as the less money spent on a project, the better, given the project is accomplished successfully.

2.2.4 *Minimizing Geographical Distance* (χ)

In the search for an optimal team, companies also look for experts in the geographical region where the organization is based. This leads to new challenges as it requires drilling down on the skills of experts while maintaining the condition of a given geolocation, which is not necessarily met under a skill-driven selection of candidates. Although remote work via online platforms has facilitated today's globalized work environment, geographical proximity remains important for different reasons, including enabling face-to-face interactions, promoting cultural understanding, time zone differences, and the necessity of access to such local resources as certain region-locked services by companies [33], and so forth. These factors can impact team dynamics, coordination, and effectiveness [108]. For example, forming a team of experts from different time zones, e.g., from GMT and EDT time zones, where the business hours/days of one expert are non-working/resting periods for another, heavily discounts the efficiency of communication and accrue more costs associated with time, effort, and resources. Out of the few works that have considered geographical proximity, Selvarajah et al. [33] defined an objective function as the sum of geographical distances between members as follows:

$$\chi(G[V_p]) = \sum_{v,v' \in V_p} d_{geo}(v, v') \tag{15}$$

where $d_{geo}(v, v')$ is the geographical distance between v and v' in G.

2.2.5 *Maximizing Density*(ψ)

Assuming that a greater number of collaborations between team members in the past increases the likelihood of success for the team, a *dense* subgraph with many internal edges has been sought to represent an optimum team. It is expected that increasing the density of a subgraph yields a team with stronger collaboration and less communication cost [58, 93]. The density of a subgraph is defined as the ratio of the total number of edges within the subgraph to the number of its nodes or, in weighted graphs, the sum of edge weights in a subgraph to the sum of the node weights; intuitively, the greater the number of edges compared to nodes in a graph, the denser the subgraph. Formally, the density of a subgraph is defined as [93]:

$$\psi(G[V_p]) = \frac{\sum_{v,v' \in V_p} w_\uparrow(e_{v,v'})}{\sum_{v \in V_p} w(v)} \tag{16}$$

where $w(v)$ is the weight of the node v. In the case of unweighted nodes, $w(v) = 1.0$ and Equation 16 is simplified to:

$$\psi(G[V_p]) = \frac{\sum_{v,v' \in V_p} w_\uparrow(e_{v,v'})}{|V_p|} \tag{17}$$

Alternatively, the density of a subgraph is defined by Juarez et al. [68] as:

$$\psi(G[V_p]) = 2 \times \frac{\sum_{v,v' \in V_p} w_\uparrow(e_{v,v'})}{|V_p|(|V_p| - 1)} \tag{18}$$

where density decreases quadratically by increasing the number of team members, favoring teams of small sizes, as opposed to Equation 16 and 17. Figure 4 compares the density formulas in sample graphs. As seen, while the

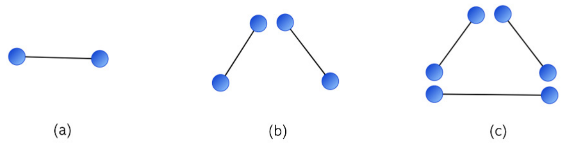


Fig. 4. Based on Equation 18, the density of graph (a), (b), and (c) are 1.0, $\frac{1}{3}$, and $\frac{1}{5}$, respectively, but by Equation 17 and 16, the density of all these graphs are equal to $\frac{1}{2}$. Here, all edge weights and node weights are assumed to be 1.0.

density based on Equation 16 or 17 is equal to $\frac{1}{2}$ for all graphs, the graph in (a) has the highest density based on Equation 18.

2.3 Hybrid (Multi-Objective) Optimization

The previous sections introduced and formalized various objectives crucial in Expert Team Formation literature. The interplay between different objectives in forming a team is also important to consider so as to model real-world circumstances more accurately. For example, companies may demand experts possessing the skills required to accomplish the tasks, and yet such experts must be in a region where the organization is geographically located while limiting the total personnel cost to the project budget. In the literature, the optimization of the following combinations of objectives is studied in a *multi-objective* optimization framework:

- Communication cost (§2.2.1) and proficiency (§2.2.2) where teams with minimum communication costs and maximum expert proficiencies are desired. This is motivated by social science studies showing that collecting a group of the best experts in each required skill does *not* necessarily ensure success as they may lack proper communication, among other reasons [75]. Zihayat et al. [41, 42] studied variations of proficiency in terms of authority metrics for skill holder experts and connector experts (Equation 11) in conjunction with communication cost.
- Communication cost and personnel cost (§2.2.3) [1, 42, 71, 72] where finding a team of skillful experts with minimum communication cost and financial compensation (salary) is desired. This is motivated by real-world scenarios where a project is constrained by a limited amount of money while different experts incur different compensations for performing the tasks of the project.
- Communication cost, personnel cost and proficiency [40] where team members can smoothly communicate with each other *and* hold the highest level of expertise (Equation 9) in the required skills while the total amount of money paid to them is minimized. Therefore, communication and personnel costs are to be minimized, yet proficiency is to be maximized jointly.
- Dynamic communication cost, geographical proximity (§2.2.4), and proficiency where communication cost and geographical distances between experts of a team are to be minimized while experts' proficiency is to be maximized. Selvarajah et al. [33] proposed to maximize mastery level and trust score (§2.2.2) as representations for experts' proficiencies, and to minimize communication cost and geographic distances to find an optimum team.
- Density (§2.2.5) and proficiency [68] where the mostly connected experts with the highest proficiencies within a subgraph are desired as an optimum team. As explained earlier, this is motivated by the fact that highly skillful experts who have collaborated frequently in the past are likely to succeed if they group together in a new team. Maximizing density and proficiency disfavour collaboration with unknown, new or early career experts.

2.4 Optimization Constraints

As explained, subgraph optimization objectives on the expert graph G aim to identify an *optimal* subgraph $G[V_p]$ as the recommended team for a project p. This process is typically performed within a set of *predefined* constraints, also referred to as boundaries, restrictions or criteria, that any feasible subgraph $G[V_p]$ for an optimum team must satisfy. The most common constraint has been skill coverage [1, 24, 41, 58, 70, 71, 79, 80, 93], that is, the expert nodes of the optimum team must collectively cover the required skills for the project p, i.e, $\forall s \in S_p, \exists v \in V_p : s \in S_v$ and $S_p \subseteq \bigcup_{v \in V_p} S_v$. Other constraints include team size (cardinality) [29, 93], availability of experts [33], capacity c_v of experts [17], required budget for the project [93], i.e, $\sum_{v \in V_p} b_v$, and a minimum number of experts with specific skills per team [58], i.e., l_s. Table 9 in Appendix D provides a summary of the constraints studied in this domain along with the approaches that incorporate them. While constraints are domain-dependent and vary based on the underlying organizational requirements, they have been addressed during the optimization process by the optimization algorithms. For example, in the case of skill coverage, an algorithm foremost identifies expert nodes that possess at least one of the required skills, disregarding the rest of expert nodes [29, 41]. Similarly, when constraints such as an upper limit on team size are present, an algorithm can check the number of experts with the required skills at each iteration and continue searching neighboring nodes until the constraint is satisfied [58].

Thus far, we have formalized Expert Team Formation problems and their optimization objectives for finding optimal teams, mostly related to the question of *'what are the optimization objectives?'*. Next, we lay out details of the proposed techniques developed to provide solutions to the optimization problem, i.e., *'how are the objectives optimized?'*.

3 Optimization Techniques

Subgraph optimization problems are proven to be NP-hard [74], and so finding an optimum team via subgraphs of the expert graph optimizing an objective function is computationally prohibitive. Therefore, different heuristics have been developed to solve this problem in polynomial time using greedy and approximation algorithms. In this section, we first describe proposed methods for modelling teams with collaborative ties within their members as an expert graph (Definition 2.1). Next, we explain seminal works followed by the majority of researchers. As will be explained, most of the graph-based works have either targeted minimizing the communication cost only, or have considered hybrid optimization of several other objectives on top of the communication cost. In light of that, we categorize graph-based Expert Team Formation studies into three groups, those that consider: *i*) minimizing communication cost only [17, 24, 29, 70, 79, 80]; *ii*) additional objectives like personnel cost, expertise level and geographical proximity jointly with communication cost [1, 33, 40–42, 71, 72]; and, *iii*) maximizing the teams' density only [58, 68].

3.1 Expert Graph Construction

Central to all the works in graph-based Expert Team Formation is building the expert graph G (Definition 2.2) that represents the experts and their past collaborations. All works in this realm consider experts as nodes, and assign experts' skills as node attributes. As we will explain in Section 4.1 (Datasets), skills might not be predefined by nature and should be inferred based on what makes intuitive sense depending on the underlying benchmark dataset. For instance, authors of a research paper can be the experts and the skills of all of the experts can be determined by the keywords in the paper title [17, 70, 79, 80], even though the authors of the paper may be skillful in a few, not all, of the skills. As to the edges of the expert graph, two experts can be connected if they have collaborated in a given minimum number of teams. In the same example, co-authors are supposed to have collaborated as part of a team. Edges can be weighted by a number between zero and one according to pairwise communication costs between experts (nodes). In academic networks of the scientific research community, for instance, the *power* of collaboration between a pair of experts v and v' is defined based on the Jaccard Index as the number of joint publications over the sum of publications by each, and the communication cost will be its complement (one minus the power) [1, 17, 24, 40–42, 70–72, 79]:

$$w_{\downarrow}(e) = w(u,v) = 1 - |\frac{P_u \cap P_v}{P_u \cup P_v}| \in \mathcal{R}^{[0,1]} \tag{19}$$

where $w_{\downarrow}$ shows the desired less value of the weights, and $P_{.}$ is the set of teams that an expert has participated in. However, there are concerns about this equation:

- The expert graph G becomes a fully dense graph with edge weights in $\mathcal{R}^{[0,1]}$, including edges of weight 1.0 for all pairs of experts with no previous collaboration, which is infeasible in large-scale graphs. To relieve this, a threshold is usually used to filter out edges with a weight above that threshold, making the graph a sparse graph [17, 24, 41, 68, 70];
- The equation is invariant to the number of past collaborations. A pair of experts with only one past joint collaboration receive zero (best) communication cost;
- The expert graph G has edges of weight 0.0 for the minimum communication cost, which is counterintuitive in graph theory where non-existent edges are commonly represented with this value. Further, in terms of implementing the expert graph with an adjacency matrix, it hinders the usage of efficient sparse matrix data structures.

To alleviate the problem of zero edge weights, Li et al. [80] have proposed :

$$w_{\downarrow}(e) = |\frac{1}{P_u \cap P_v}| \in \mathcal{R}^{(0,1]\cup\{\infty\}} \tag{20}$$

Equation 20, however, overlooks the total number of collaborations in teams for each of the experts, as a result of which early career experts and long-standing well-experienced experts are treated equally, which may render it unfair for either of the groups in different scenarios. For instance, Equation 19 favours early career experts as their total number of participation in teams would be low, hence, an instance of joint collaboration would weigh more toward minimum communication cost compared to experienced experts (e.g., $1 - \frac{1}{10} = 0.900$ better communication/lower cost vs. $1 - \frac{1}{1000} = 0.999$ worse communication/higher cost).

Nonetheless, Equations 19 and 20 have been employed in scenarios where lower weights signal the communication cost based on the number of joint collaborations between two experts and subgraph minimization has been desired. In some works, however, a maximization optimization is involved, where the edge weights are desired to be maximized. Hence, the edge weight between two experts can be directly set by the number of joint collaborations as in [58, 68, 93]:

$$w_{\uparrow}(e) = |P_u \cap P_v| \in \mathcal{N}^{[0,\infty)} \tag{21}$$

3.2 Communication Cost Minimization Algorithms

The work of Lappas et al. [79] is the first attempt to form a team based on the subgraph optimization on the expert graph. A major component of Lappas et al.'s approach is minimizing the communication cost among experts based on two alternatives over possible subgraphs: *i*) diameter, and *ii*) spanning trees. For the former alternative, they proposed *RarestFirst* wherein a subgraph with the minimum diameter is to be found. For each required skill s, it finds a subset of experts that have the skill V_s, where the skill $s^{\dagger}$ with the minimum number of experts is called a *rare* skill (§2.1). To form a team, an expert who holds the rare skill is selected as the seed (rarest first), and other experts who own the other required skills and have the minimum distance from the seed expert are selected as the team members. Since there may exist more than one expert who holds the rare skill (more than one seed expert), there can be more than one possible team. The optimum team is the one with the minimum diameter in its respective subgraph.

For the latter alternative, i.e., spanning tree-based optimization of communication cost, Lappas et al. [79] proposed *CoverSteiner* and *EnhancedSteiner* algorithms. *CoverSteiner* is a heuristic greedy two-step algorithm: in

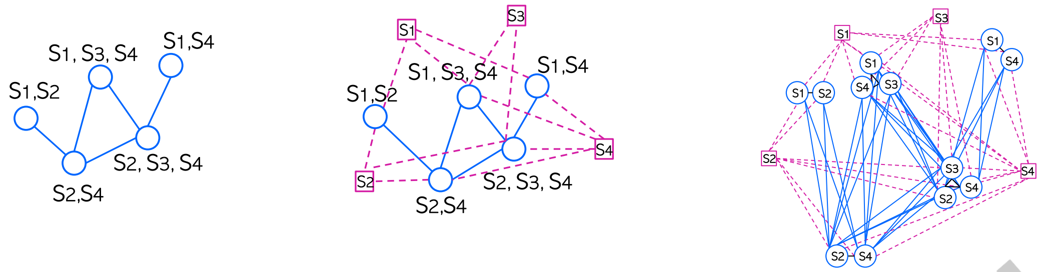


Fig. 5. Graph extension of the Steiner tree algorithm where, in the first step, the graph extends by adding additional skill nodes and connecting them to all of the experts associated with that skill, and in the second step, original nodes are replaced with a complete graph of the size $|S_v|$ in such a way that each node in the complete graph is assigned a specific skill.

the first step, the set of candidate experts covering all of the project's required skills are selected by solving a classic *set cover* problem [105]. The set of chosen experts is then expanded to include other experts in order to build a minimum spanning tree via the Steiner tree algorithm [104]. *CoverSteiner* algorithm, however, overlooks the underlying graph structure of experts in the set cover algorithm in the first step, which may lead to possible teams with high communication costs, like when a pair of selected experts is disconnected in the expert graph. To avoid such expert selections, Lappas et al. [79] proposed *EnhancedSteiner* algorithm. *EnhancedSteiner* is also a two-step algorithm. In the first, as shown in Figure 5, it forms an extended *heterogeneous* graph, H, of the expert graph as follows: for each required skill s, an additional node (an expert with that skill, which they refer to as skill node) is added and connected to all of the expert nodes that have the skill s. Then, each node v, excluding the added nodes, is replaced by a complete graph of the size $|S_v|$ in a way that each node of the mentioned complete graph is equipped with one of the skills of S_v. The weights of edges between skill nodes and expert nodes are set to a large number, and the edge weights of the mentioned complete graphs are set to zero. In the second step, like *CoverSteiner*, the same Steiner tree algorithm is applied to the newly formed extended graph where the set of required skills is the input subset for *CoverSteiner* algorithm. Finally, skill nodes are omitted from the output set to find the optimum subgraph.

To improve the time complexity of the proposed diameter-based and spanning tree-based algorithms, Lappas et al. [79] further proposed greedy variations, referred to as *GreedyDiameter* and *GreedyMST*. Lappas et al. [79] evaluated their proposed algorithms based on minimizing the communication cost, team size and connectivity of the respective subgraphs for the output recommended teams on DBLP dataset. In terms of communication cost, the greedy versions of the proposed algorithms performed poorer than the exact algorithms, as expected. Surprisingly, no comparative analysis has been performed between the diameter-based algorithm, i.e., *RarestFirst*, and the spanning tree-based algorithms, i.e., *CoverSteiner* and *EnhancedSteiner*. Therefore, it is unclear which one is a better estimator for communication cost. In terms of team size, the *RarestFirst* algorithm results in larger teams, which is usually undesirable due to other factors like personnel cost, while the *EnhancedSteiner* algorithm generally forms smaller teams, owing to their different minimization methods where they consider the diameter and spanning tree of the subgraphs, respectively. In terms of team connectivity, *CoverSteiner* often fails to find connected subgraphs as an optimum team, whereas *RarestFirst* (diameter-based) and *EnhancedSteiner* (spanning tree-based) can recommend more connected subgraphs, as they consider the graph structure of the expert graph.

Following Lappas et al. [79], a considerable number of studies propose a wide variety of algorithms that minimize the communication cost incurred by the subgraphs in an expert graph through minimization of the diameter, Steiner tree, or sum of distances in the subgraphs [17, 24, 29, 31, 70, 79, 80]. Given an additional constraint on expert capacity, Datta et al. [17] propose *MinDiamSol* aiming to minimize the diameter and *MinAggrSol* aiming to minimize the spanning tree of the team's subgraph (§2.2.1) in their definition of the Expert Team Formation problem. Both algorithms first transform the expert graph G to a simpler graph g by excluding expert nodes whose distance from the seed node is more than a desired (arbitrary) number. Then, *MinDiamSol* attempts to minimize the distance between the seed node and subgraph's nodes, while *MinAggrSol* greedily picks those nodes of g whose addition to the subgraph leads to maximizing the skill coverage. They also extended Lappas et al.'s *EnhancedSteiner* and *RarestFirst* algorithms to consider the capacity of experts as an additional constraint, referred to as *GreedySteiner* and *GreedyDiam*, respectively. Regarding efficacy, Datta et al.'s algorithms empirically outperform Lappas et al.'s counterparts on GitHub and DBLP datasets. Specifically, on GitHub, *MinDiamSol* and *MinAggrSol* can reduce the communication costs by up to 40%. In terms of efficiency, the diameter-based algorithms, that is, *GreedyDiam* and *MinDiamSol*, were faster than others in both datasets.

Another extension to Lappas et al.'s algorithms has been made by Li et al. [80] who proposed *GeneralizedEnhancedSteiner* to consider an arbitrary number of experts for each required skill ($l_s = u_s > 1$) to form a team; Lappas et al.'s algorithms were limited to *one* expert per skill ($l_s = u_s = 1$). Similar to Lappas et al.'s *EnhancedSteiner*, Li et al.'s *GeneralizedEnhancedSteiner* extends the graph with nodes for skills foremost (see Figure 5). Next, the algorithm selects an expert with a higher density in the neighborhood as a seed node. Li et al. presumed that the higher the neighborhood density of a skill node is, the more likely it is for that skill node to have a smaller distance from other nodes. Given a pair of expert nodes v and v', the density of the expert v is more than v' should more experts be closer to v, and intuitively, the density increases as more experts come closer to the node. This approach is, however, computationally intractable when many skills are required, or the expert graph is large. To reduce the time complexity, Li et al. proposed *GroupingDensity* and *GroupingRandom* algorithms. These algorithms firstly form a modified expert graph, referred to as a *'group graph'*, by merging expert nodes who have a required skill into a single *super* node followed by modified edge weights based on communication cost between such super nodes, each of which representing a subset of skill holder experts. Next, the *GeneralizedEnhancedSteiner* algorithm is applied but only on the *'group graph'* to find an optimum Steiner tree whose super nodes are expanded to the original expert nodes as the optimum team.

Farhadi et al. [24] build upon Li et al.'s algorithms by considering those experts who have at least $x_{v_s} \in \mathcal{R}^+$ level of expertise for each required skill $s \in S$ only (§2.1). They consider the level of expertise of each expert in each skill as a node attribute, and experts whose expertise are higher than or equal to the minimum required expertise level are selected to cover a required skill. Farhadi et al. compare their algorithm with Lappas et al. [79]'s *RarestFirst* and Li et al. [80]'s *GeneralizedEnhanceSteiner* on DBLP dataset and show improvements in both optimum team size and communication cost via taking expertise levels into account.

Diameter-based or minimum spanning tree-based optimization methods are, however, *unstable* or *sensitive* to changes, even minor, in the expert graph; adding or removing an edge leads to major changes in the solution [58, 70, 80]. Towards more stable or robust optimization methods, Kargar et al. [70] introduce the *sum of distances* (Equation 2) in a subgraph for minimizing the communication cost, yet proving that finding a team that minimizes this distance is NP-hard. They proposed an approximation algorithm, *BestSumDistance*, that finds the optimum team with 2−approximation. This algorithm works as follows: given a required skill s and each of the experts v that own it, i.e., $v \in V_s$ (§2.1), the algorithm computes the distance of v to each of the experts v' in the complement set of V_s, i.e, $v' \in V \setminus V_s$. The expert v will be added to the subgraph if it has the least sum of distance over all such v'. In the end, *BestSumDistance* returns a subgraph with the minimum sum of distances as the optimum team.

To reduce the time complexity, even more yet address a real-world scenario, Kargar et al. [70] introduced forming a team with a leader (Equation 6). They propose an exact algorithm, *BestLeader*, that tries to minimize the sum of distances between the team's leader and all team members. Initially, this algorithm assumes each expert L as a leader. Then, for each required skill s, the expert v of V_s who has the least distance to L is added to the subgraph. *BestLeader* computes the distance of each expert node to the leader to determine its addition to the subgraph, as opposed to the *BestSumDistance* that computes the distance of each expert to all other experts. In *BestLeader*, hence, the algorithm forms a subgraph per each assumed leader L. In the end, a subgraph with the least leader distance and its leader are returned by the algorithm as the optimum team. They evaluate their algorithms by experimenting on DBLP and show that *BestLeader* and *BestSumDistance* outperform the baselines, *RarestFirst* and *EnhancedSteiner* of Lappas et al. [79].

Like Kargar et al. [70], to form a team with a leader, Huang et al. [29] argue that in real-life scenarios, especially when teams' sizes are large, a team should be considered as a collection of subteams each of which is equipped with a leader. The effectiveness of the team as a whole, thus, depends on inter-subteam collaborations through subteams' leaders; there is no need for all experts in different subteams to collaborate with each other. Therefore, Huang et al. aim to minimize the communication cost between subteam leaders with subteam members and the main leader. To minimize leaders' communication cost (Equation 7), they propose the *Assignment and Pruning (AP)* algorithm grounded on the Assignment Problem [6] and a pruning framework. The Assignment Problem involves finding the optimal way to match a set of tasks or jobs (herein, required skills of a project) to a set of workers or resources (herein, experts), while optimizing an objective function related to cost, time, or efficiency. In the Assignment Problem, however, no graph structure has been utilized. Huang et al. use *AP* algorithm to find all possible teams that cover all of the required skills foremost, followed by pruning all but those that have a minimum communication cost in their respective subgraphs. According to their experiments on the GitHub dataset and a small-scale synthetic dataset, Huang et al.'s method based on leaders' distances yields competitive performance in comparison with solutions based on diameter minimization, minimum spanning trees, and the sum of distances between team members.

More recently, Nemec et al. [31] have proposed to employ random walks between a pair of experts in the expert graph as opposed to the shortest path when minimizing communication cost. They argue that the shortest path overlooks the structure of the expert graph and falls short when the majority of experts share the same or similar shortest paths. They present *Random Walk with Restart (RWR)* algorithm where the communication cost between a pair of expert nodes v and v' has been defined in inverse relation to the estimated probability (likelihood) of reaching from v' to v, via a uniform random selection of edges in a walk; connected sparse expert nodes with few edges would have more probabilities as opposed to connected expert nodes that are located in a dense subgraph. Higher probability indicates stronger connections, hence, less communication cost between pairs of expert nodes. Nemec et al. [31] show that the Expert Team Formation problem still remains NP-hard when the distances between expert nodes are computed by the *RWR* algorithm. To relieve this, they propose a greedy algorithm to reduce the search space by choosing an expert v with the rarest skill $s^{\dagger}$ as a seed node. In each step, an expert that covers one of the remaining skills while having the minimum communication cost with the seed node based on *RWR* is added to the subgraph. In the end, among all of the formed subgraphs around the seed node, the algorithm returns the one with the minimum communication cost.

3.3 Hybrid Optimization Algorithms

Forming a team based on a single objective like minimizing communication cost only, forgo other human and non-human factors affecting the team's success, such as project budget, personnel cost, proficiency and geographical proximity. While there may be no solution that optimizes all criteria simultaneously, many works [1, 33, 40–42, 68, 71, 72] target multi-objective optimization in the Expert Team Formation problem considering a combination of crucial factors such as the ones listed above. Approaches to tackling the multi-objective optimization problems in graph-based Expert Team Formation studies can be categorized as follows, also summarized in Table 3:

(1) Initially, integration of objectives was proposed where multiple objectives are merged into a modified expert graph and proxied by a single synthetic objective whose optimum is also optimum for other constituent

Table 3. Summary of multi-objective algorithms and the method that has been used for optimizing objectives in tandem.

		Multi-Objective Optimization Method		
	Algorithms	Integration of objectives	Trade-off	Pareto search
Kargar et al. (2012) [71]	Approx	✓	✓	
	Replace	✓	✓	
	MCC	✓	✓	
	MCC-Rare	✓	✓	
Kargar et al. (2013) [1, 72]	Approx{diameter, PCost}		✓	
	Approx{sumDistance, PCost}		✓	
	Approx{PCost, diameter}		✓	
	Approx-Pareto			✓
Zihayat et al. (2014) [40]	TwoPhase			✓
	FirstPhase			✓
	PLS			✓
Zihayat et al. (2017) [41]	CA-CC	✓	✓	
	SA-CA-CC	✓	✓	
Zihayat et al. (2018) [42]	PC-CC	✓	✓	
Selvarajah et al. (2021) [33]	MOCA			✓

objectives [41, 42, 71]. However, such methods depend heavily on the merging protocol to build the modified expert graph.

(2) Next, linear interpolation of conflicting objectives with trade-off parameters (coefficient) was proposed, mainly when optimizing multiple objectives simultaneously is impossible, and a balance should be considered among them [1, 72] like communication cost and geographical distance joint minimization.

(3) Finding Pareto solutions has been the state of the art in hybrid subgraph optimization for Expert Team Formation wherein multiple objectives reach a win-win state such that any attempt to favour one objective makes at least one other objective worse [1, 40, 72]. In other words, a subgraph (team) $G[V_p]$ dominates another subgraph $G[V'_p]$ with respect to two objectives if $G[V_p]$ is better than $G[V'_p]$ in one objective and not worse than $G[V'_p]$ (i.e., equal or better) in the other objective. A subgraph is a Pareto optimum team if it is *not* dominated by any other subgraphs.

Kargar et al. [71] is the pioneering graph-based approach that considers two objectives jointly for minimizing communication cost and personnel cost, a *bi*-objective optimization function that has been shown to be NP-hard. Following (1) integration of an objective approach to solve the problem, they present an approximation algorithm called *Approx*. In an expert graph, *Approx* initially spreads node weights, representing experts' personnel cost, and interpolates them with the existing edge weights, representing communication cost, such that minimizing the sum of distances in the modified graph is equal to minimizing the *bi*-objective function in the original graph. *Approx* then selects an expert holding at least one required skill as the seed node and incrementally adds the nearest expert nodes to the seed expert node to cover all required skills for a team. Hence, a candidate subgraph is formed based on the nearest distances of experts to the seed expert node only and inter-distances between selected experts are ignored. After forming all such candidate subgraphs, a subgraph with the smallest sum of distances to the seed is selected as the optimum team. To improve the efficiency of *Approx*, Kargar et al. [71] further propose three heuristic algorithms, *Replace*, *Minimal Cost Contribution (MCC)* and *MCC-Rare*. *Replace* selects an expert with the lowest personnel cost holding at least one of the required skills and replaces her with another expert holding the same skills but with a lower communication cost. *MCC* is similar to *Approx*, but it examines the communication cost of the yet-to-be-selected expert with not only the seed expert but also with other selected experts. In *MCC*, new team members are added incrementally, and each new member is chosen by comparing its communication cost with that of all the current members (not only the seed member) of the team in addition to the personnel cost of the new member. Finally, to reduce the run time of the *MCC* algorithm, they propose *MCC-Rare*, which is a variation of *MCC* where an expert with the rarest required skill $s^{\dagger}$ is used as a seed node. Kargar et al.'s experiments on DBLP and IMDB datasets show that *MCC* outperforms other algorithms regarding combined communication and personnel costs, and *MCC-Rare* and *Approx* are the runners-up. However, in terms of time complexity, *Replace* was the fastest, yet the poorest algorithm with the highest combined communication and personnel costs, *MCC-Rare* and *Approx* are the second and third fastest algorithms, and *MCC* is the slowest one. Further, Kargar et al. [71] benchmark their algorithms on each objective separately. To minimize communication cost, *Approx* yields the best results, while *MCC-Rare* could form a team with the lowest personnel cost.

Following the (2) integration of objectives with trade-off parameters, Kargar et al. [72] propose an $(\alpha, \beta)-$approximation algorithm to find teams where the value of the first objective (e.g., personnel cost) is at most α times a given fixed amount (e.g., overall allocated budget) while the value of the second objective (e.g., communication cost) is at most β times the minimum value of any solution that meets the first objective. Their first algorithm *Approx{diameter, PCost}* computes the diameter-based communication cost (Equation 4) considering a threshold on diameter, then attempts to minimize the personnel cost. It is a $(2, log|S_p|)-$approximation algorithm where S_p is the set of required skills (§2.1). This algorithm works as follows: for each node v equipped with the rarest skill $s^{\dagger}$, it computes the distance of v from all nodes of $V \setminus V_{s^{\dagger}}$. Then, for each required skill, the node v' will be added to the subgraph if its distance from v is less than the mentioned threshold on diameter. If all of the expert nodes' distances from v exceed this threshold, then there is no possible solution. In the end, among all the formed subgraphs, the algorithm returns the subgraph with the minimum personnel cost. *Approx{sumDistance, PCost}* is the second algorithm by Kargar et al. [1, 72] which is similar to the previous one, only it uses the sum of distances based communication cost (Equation 2) instead of diameter. This is a $(|S_p|, log\ |S_p|)-$approximation algorithm and considers each expert node with one of the required skills as a seed node, unlike the previous algorithm that only uses rare skill holders as seed.

The third algorithm by Kargar et al. [1, 72] is *Approx{PCost, diameter}*. This algorithm is a $(log\ |S_p|, 2)-$approximation; it aims to minimize diameter-based communication cost given a desired threshold for personnel cost. It first starts by computing the diameter of the input graph $\varphi_R(G[V])$ and stores it. Then, *Approx{PCost, diameter}* calls *Approx{diameter, PCost}* and considers $\varphi_R(G[V])$ as a threshold for diameter to find a team with the minimum personnel cost and a diameter less than $\varphi_R(G[V])$. If the personnel cost of the formed team is more than the required threshold on personnel cost, then there is no possible team since personnel cost and communication cost are conflicting objectives. However, if the personnel cost of the formed team is less than the required threshold for personnel cost, the algorithm decreases the diameter threshold $\varphi_R(G[V])$ to a smaller value (e.g., $\frac{\varphi_R(G[V])}{2}$) and calls the *Approx{diameter, PCost}* again to find a team with a diameter less than the new threshold and the minimum personnel cost. This process ends when the diameter becomes less than a threshold. Last, Kargar et al. [1, 72] propose the *Approx-Pareto* algorithm to find (3) Pareto optimal teams for the personnel and communication costs. The *Approx-Pareto* algorithm repeatedly calls *Approx{diameter, PCost}* and for several different diameter thresholds forms a team with minimum personnel cost. Thus, a set of teams is formed, each with the minimum personnel cost under the given diameter threshold. In the end, the algorithm checks if each generated team is dominated by other teams or not, and then ignores the dominated ones and returns non-dominated teams as Pareto optimal teams. In DBLP and IMDB, *Approx-Pareto* could achieve state-of-the-art performance in terms of accuracy and speed.

In another line of work, Zihayat et al. [41, 42] optimize communication cost and authority of experts (proficiency) in a team. As explained in Section 2.2.2 (Maximizing Proficiency), skill holders are team members with required skills, which may be connected via experts who cover *none* of the required skills, referred to as connectors. Zihayat et al. [41, 42] argue that the authorities of connectors have an impact on the team performance besides their connector role. They define two separate objective functions to incorporate the connectors' authorities and skill holders' authorities. Following (1) the integration of objectives, Zihayat et al. merge the expert node weights, representing the authority level of experts, with edge weights, representing communication costs. In doing so, they first transfer each node weight to all the edges that come to the node and then define a new edge weight for the graph. They show that minimizing the communication cost based on the sum of edge weights (Equation 3) for the modified expert graph results in an optimized communication cost and connector authorities. Zihayat et al. [41, 42] show that minimizing the final synthetic objective is NP-hard and propose the *Communication Cost-Connector Authority (CC-CA)* algorithm for finding the optimum team. This algorithm tries to build a tree around each expert node v as a root so that it has the minimum sum of edge weights. For each root v, the algorithm finds the nearest skill holder v' to the root and adds v' as well as all of the expert nodes between v and v' to the team. In the end, among all the formed tree subgraphs, the algorithm returns the subgraph with the lowest sum of edge weights based on Equation 3 as the optimum team. Zihayat et al. [41, 42] also propose *Skill holder Authority-Connector Authority-Communication Cost (SA-CA-CC)* algorithm to optimize communication cost, connector authority and skill holder authority by merging skill holder authority, similar to *CC-CA*, in the modified graph.

With respect to more than two objectives, Zihayat et al. [40] show that the optimization of three objectives is NP-hard and, based on (2) Pareto solutions, they propose *TwoPhase (FirstPhase and PLS)*, an approximation algorithm to optimize communication cost, personnel cost and expert proficiency. The first step of this algorithm is responsible for finding a subset of all Pareto teams. This subset is the output provided by the *lexicographic* minimization solution of a trade-off relation between communication cost, personnel cost and proficiency. In a lexicographic optimization problem, candidate optimal outputs are reached by initially optimizing the first objective. Then, if two candidate solutions are the same as per the first objective, the optimum one is selected based on the second objective. The same procedure is pursued for the subsequent objectives. It is proven by Paquete et al. [90] that the output of the mentioned optimization problem always consists of Pareto teams; however, the reverse does not hold in this first phase of the algorithm, i.e., there might be Pareto teams that are missed by the algorithm.

Zihayat et al.'s *FirstPhase* works as follows (herein, we assume the communication cost is the first objective to be optimized without loss of generality to any of the three criteria mentioned): an expert with a required skill is selected as the seed node randomly. Then, for each uncovered skill, a skill holder closest to the seed node is added to the subgraph. This continues until all of the required skills are covered. By the end of this step, the algorithm has formed subgraphs with minimum communication cost as experts are selected in a way that they have a minimum distance from the seed node. In the next step, the algorithm tries to consider other objectives as well, i.e., personnel cost and proficiency. In the final step of the first phase, the algorithm compares all the formed subgraphs based on a second objective (e.g., personnel cost), and the subgraph with the desired rate is kept as the optimum team. If they tie on the second objective, the third objective (e.g., personnel costs) is considered, and the one with the desired rate is kept as the optimum team. The aforementioned procedure is done for all experts with the required skills. Since not all Pareto teams can be found, as explained earlier, another subset of Pareto optimal teams is discovered in the second phase of *TwoPhase* algorithm via a *Pareto Local Search (PLS)* algorithm.

More recently, Selvarajah et al. [33] optimize communication cost, level of expertise, trust score and geographical proximity within the context of a multi-objective Expert Team Formation problem. They propose a unified framework for multi-objective Expert Team Formation problems and present a *Multi-Objective Cultural Algorithm (MOCA)*, which is an extended version of a class of evolutionary algorithms [67], that returns the highest ranked Pareto optimal teams. The crucial difference between their work and previous works is considering the dynamic nature of the mentioned objectives. For instance, an expert's proficiency can change over time through consistent practicing and improving their skills. Therefore, they attempt to include the time-related aspects of objectives in the Expert Team Formation problem. Selvarajah et al. [33] convert static formulas to dynamic versions by adding a term which accounts for the difference between the current time and the last time that two experts have collaborated on a task.

While Pareto optimization has remained the state of the art in graph-based Expert Team Formation, modern multi-objective optimization techniques have been developed beyond Pareto [98] such as multi-factorial [81] and multi-task optimization [25, 49] to support parallel optimization of multiple functions simultaneously, each of which can be single- or multi-objectives, and can be employed for multiple subgraph optimization to form multiple optimum teams that are interrelated. We will provide a more in-depth explanation in Section 5.3. Modern Hybrid Optimization Algorithms.

3.4 Community-based Optimization Algorithms

A few works propose detecting communities as a proxy for optimum teams [58]. In communities, edges, which represent previous collaborations between the experts, are highly densely connected and indicate internal cohesion, hence minimum communication cost, while edges between experts in different communities are sparse and indicate externally loose connections, therefore, maximum communication cost.

Gajewar et al. [58] is the first to present community-based algorithms. They target forming a team with a maximal subgraph density and a desired, arbitrary number of experts required for each skill. They prove that finding a team with maximal density is NP-hard and propose two approximation algorithms, *s-DensestAlk* and *m-DensestAlk*, to solve this problem. Gajewar et al. [58] also present three heuristics called *Enhanced-Dense*, *PartialTrimmedDense* and *CompleteTrimmedDense* which are extensions of previous approximation algorithms. These heuristics trade off between the size of the returned solution and the density while covering all required skills. Gajewar et al. [58] also extend *RarestFirst* of Lappas [79] to *MinDiameter* algorithm for projects wherein more than one expert may be needed for each skill, as happens in real-world situations. In evaluation, the density-based algorithms *s-DensestAlk* and *m-DensestAlk* outperform the diameter-based algorithm *MinDiameter* as per different metrics such as *team cardinality*, *teamPubs*, *partialTeamPubs* and *teamPubRati*, which will be explained later in the Section 4 (Evaluation Methodology). Also, formed teams will be connected if the social network is connected; however, there is no upper bound for team size.

Juarez et al. [68] also present a multi-objective formulation to maximize the density along with expert proficiency among team members. Using a different formula to compute the density of a subgraph (Equation 18), followed by a genetic algorithm for optimization, Juarez et al.'s algorithm yields smaller and more connected teams compared to Gajewar et al.'s work. In comparison with the *RarestFirst* algorithm of Lappas et al. [79] and *m-DensestAlk* of Gajewar et al. [58], Juarez et al.'s algorithm outperforms the baselines on DBLP dataset in terms of optimal team size, minimum communication cost and maximum density.

3.5 Efficiency and Scalability Enhancement

Mid- to large-scale datasets yielding large graphs pose high computational complexity on graph-based optimization techniques. For scalability, researchers have tried to reduce the computational cost, including time and space, while maintaining accuracy in two directions: *i*) pruning the dataset or expert graph during preprocessing, and *ii*) utilizing heuristics during subgraph optimization. Pruning techniques aim to reduce the search space on the expert graph by filtering the number of teams, the skills S, expert nodes V, and/or joint collaboration among experts for edge set E. To reduce the number of teams, which also may lead to a further decrease in the expert nodes and skills, researchers filter out teams based on the team size, e.g., removing the papers or movies with a single author or cast and crew in DBLP and IMDB, respectively. This filtering has been implied by their definition of a team, as it assumes that collaboration is necessary, although may be not explicitly mentioned. Alternatively, they focus merely on a subset of projects based on the areas of specialty, e.g., only *database* and *data mining*-related papers in DBLP [17, 24, 40, 41, 70–72, 79, 80]. There have been works that select a subset of teams based on a timestamp associated with a team, e.g., movies with release dates from 2000 to 2002 [70, 71]. For skill set reduction, a minimum distribution threshold is set. For instance, in DBLP dataset, the set of terms that appear in a minimum number of paper titles are considered as the skill set [41, 72, 80]. Likewise, to reduce the expert set, a minimum threshold on the number of participations in teams has been used, e.g., minimum number of papers in DBLP [24, 72] and movies in IMDB [40, 70]. Some works apply the pruning on edge set E based on a minimum threshold for collaborations among experts like minimum number of co-authorship in papers in DBLP or co-staring in movies in IMDB [17, 24, 40, 70–72]. Rarely, researchers apply graph sampling techniques to address scalability of their proposed algorithms like Datta et al. [29]'s work where subgraphs of varying sizes were generated from the expert graph in GitHub by randomly selecting a node and performing breadth-first search around it. In Table 4 of Appendix F, we summarize the filtering methods in each of the graph-based algorithms.

4 Evaluation Methodology

This section outlines the evaluation methods for Expert Team Formation approaches.

4.1 Datasets

The major datasets used in Expert Team Formation studies vary in terms of domain, including computer science publications (DBLP), moving pictures (IMDB), and open-source software (GitHub). As seen, employed datasets can be from the same domain but with different graph sizes, from small to large scales.

4.1.1 DBLP

Following Lappas et al. [79], the computer science bibliography dataset, known as DBLP, has become a benchmark dataset not only in the graph-based studies but also in the entire field [1, 17, 24, 40–42, 58, 68, 70–72, 80, 93]. This dataset has been generated from the DBLP website, established in 1993, resulting in four million publications authored by about two million computer scientists and researchers. This dataset contains extensive information on publications, including titles, authors' names and fields of research, references, venues, publishers, publishing dates, DOI, number of citations, and so forth, as well as information pertinent to authors such as full names and affiliations. An advantage of DBLP is the availability of open-source libraries that can efficiently load the data, which is essential for researchers. To form the expert graph, each publication is mapped to a team whose expert members are the publication's authors and the required skills S_p are set either as the publication's fields of studies [58] or keywords in the paper's title [1, 24, 41, 70–72, 79, 80]. In this setting, thus, the graph nodes V and edges E represent authors and co-authorship in publications, respectively. Node attributes A could include the *h-index* of the authors as the proficiency level a_v, the number of citations to the expert's publications in each of the research domains as their expertise x_{v_s} in each domain, and the maximum number of projects that the experts can handle in a year as their capacity c_v. DBLP dataset can be obtained from its official online repository or via third-party bibliographic databases such as AMiner [32][1].

4.1.2 IMDB

Another popular dataset in this domain, the Internet Movie Database known as IMDB [1, 70–72] is publicly available and includes information about moving pictures such as movies, television shows, and video games, as well as information on actors, actresses, and other members of cast and crew. The choice of IMDB in Expert Team Formation literature is not to be confused with its use cases in recommender systems or review analysis research; herein, the goal is to form a team of cast and crew for a moving picture's production, as opposed to a moving picture recommendation, for instance. To form the expert graph based on the IMDB dataset, each moving picture is considered a team whose expert members V are the cast and crew and the moving picture's genre/s are the

[1] https://www.aminer.org/citation

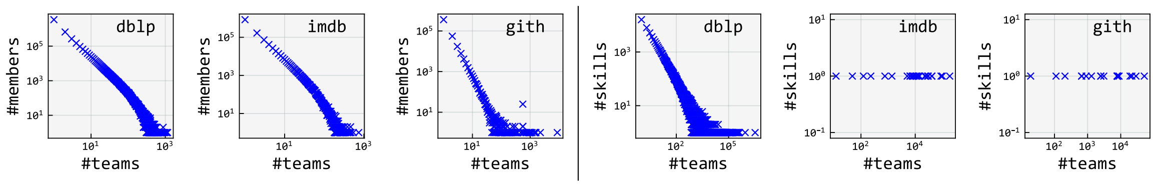


Fig. 6. Distribution of teams over members (left) and skills (right) for DBLP, IMDB, and GitHub datasets.

team's required skills S_p. Therefore, nodes can be members of cast, edges E can represent co-plays in moving pictures, and node attributes A could include the personnel cost ρ as the day/week rate of the member of cast and crew acting or working in a moving picture, and the number of moving pictures in which the members of cast and crew worked in as their proficiency level a_v. As it will be described later in Section 5 (Challenges and Future Direction), as opposed to DBLP, opting for movie titles' keywords to represent skills can be naive since movies' titles rarely correspond to the movie making skills. The IMDB dataset is easily accessible for research purposes via its official repository at the mentioned link in Table 4 of Appendix E.

4.1.3 GitHub

The other dataset is GitHub, which hosts a large number of open-source software projects (repositories) and software developer profiles [17, 29, 31]. Unlike DBLP and IMDB, GitHub provides explicit edges E through followership connections. In GitHub, two experts are connected in the expert graph if they have contributed to the same repository. Attributes for an expert A_v can include the list of followers and the repositories created and contributed to. The skill set S_p is either derived from the repositories' titles or the programming languages employed in the repositories. The GitHub dataset can be accessed through the official APIs to stream the information on a per-repository basis[2]. It is worth noting that there are GitHub datasets that are streamed and publicly available in a single compressed file, but mostly for software engineering research purposes [52] and may miss the information required to form an expert graph for the Expert Team Formation problem.

Figure 6 has been rendered for the latest versions of DBLP, IMDB, and GitHub datasets. As seen, in left the long tail problem occurs in the distributions of teams over experts in these datasets; this is because many experts (researchers in DBLP, cast and crew in IMDB, and developers in GitHub) have participated in very few teams (papers in DBLP, movies in IMDB, and repositories in GitHub). For instance, 10^6 researchers have participated in 1 team only, while few researchers have co-authored more than 10^3 papers in DBLP. With respect to the set of skills, DBLP follows different distributions compared to IMDB and GitHub. While DBLP suffers further from the long-tailed distribution of skills in teams, IMDB and GitHub follow a more fair distribution, as shown in Figure 6 right. Specifically, IMDB and GitHub have a limited variety of skills (genres and programming languages), which are, by and large, employed by many movies and repositories, respectively. Such distinct distributions in varying domains open benchmark challenges for reproducibility and generalizability of Expert Team Formation algorithms, which have received little to no attention in the literature, as discussed further in Section 5.4 (Open-Source Implementation).

4.1.4 Synthetic

Last, when obtaining real-world datasets has become challenging or restricted, researchers adopt or generate synthetic datasets to benchmark their proposed algorithm [8, 29, 33]. For instance, Selvarajah et al. [33] generated synthetic expert graphs following the Lancichinetti et al. [78]'s graph sampling method, designed to benchmark community detection methods in graphs with heterogeneity (power law) in the distributions of node degrees and

[2]`https://api.github.com/repos`

Table 4. Evaluation methodologies adopted by proposed methods in Expert Team Formation problem.

		Lappas et al. (2009) [79]	Li et al. (2010) [80]	Farhadi et al. (2011) [24]	Kargar et al. (2011) [70]	Datta et al. (2012) [17]	Kargar et al. (2012) [71]	Gajewar et al. (2012) [58]	Kargar et al. (2013) [72]	Kargar et al. (2013) [1]	Rangapuram et al. (2013) [93]	Zihayat et al. (2014) [40]	Huang et al. (2017) [29]	Zihayat et al. (2017) [41]	Zihayat et al. (2018) [42]	Juarez et al. (2018) [68]	Nemec et al. (2021) [31]	Selvarajah et al. (2021) [33]
Intrinsic	Precision								✓	✓		✓						
	Recall								✓	✓		✓						
	Top-cited Papers	✓																
	Highly-rated Venues													✓				
	Exact Algorithm				✓		✓		✓	✓		✓		✓				✓
Quantitative		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Qualitative	Team Size (Cardinality)	✓	✓	✓	✓													
	Team Connectivity	✓																
	Number of Liaisons		✓															
	Number of Common projects				✓													
	Skill Count		✓		✓													
	Hypervolume								✓	✓		✓						
	Average Distance								✓	✓		✓						
	Maximal Distance								✓	✓		✓						
	Number of Non-dominated Solutions																	✓
	TeamPubs							✓										
	PartialTeamPubs							✓										
	Number of Top-cited Authors							✓										

community sizes. There have been other works benchmarked on synthetic datasets, yet the underlying graph sampling methods to generate the datasets have mostly remained unexplained [29, 80].

4.2 Evaluations

The proposed approaches surveyed in this paper provide different experimental analyses of their performances based on both effectiveness (accuracy) and efficiency (speed).

4.2.1 Effectiveness (Accuracy)

Measuring the effectiveness of Expert Team Formation approaches faces a fundamental challenge since the basic question of *'what it takes for a team to be considered successful'* has gone under examination and has remained controversial in the literature. Finding experts who collectively cover the required skills and have optimum communication cost, personnel cost, or any other objective individually or jointly for a team might be insufficient for labelling the *truth* about its success or failure, as these criteria do not necessarily guarantee the team's success. Nonetheless, almost all graph-based works have assumed existing teams in a dataset as successful teams (positive samples). For instance, in DBLP, a publication is considered a tangible indication of a team's success. However, the rejected manuscripts, which by comparison represent teams' failures, are missing in the dataset. Similarly, in IMDB, all instances of moving pictures are considered successful since they have at least completed the production procedure and have been released for public viewing. In the following, we categorize the evaluation metrics and means used in the Expert Team Formation realm into three types: intrinsic, quantitative, and qualitative, as summarized in Table 4.

Intrinsic vs. Project Generation Evaluation. An intrinsic evaluation schema, as we denote it, hinges on the idea that we can have a standalone gold standard (ground truth) against which we can measure an approach's effectiveness. Gold data is commonly created through the selection of a subsample of a given dataset, a procedure that is based on the assumption that the existing instances of the teams in a dataset are of type *successful*. In this schema, one can measure their approach's effectiveness based on the existence of their output teams in this gold set. As opposed to the intrinsic evaluation, an extrinsic evaluation schema evaluates a given method's output based on its impact on the performance of other applications [63]. In the DBLP dataset of publications, some works put a stricter constraint on a team to be a gold team, like if the team is cited more than a certain number [79] or published in a highly-rated venue [41]. A third evaluation schema, utilized by some graph-based works [17, 24, 79, 93], is performed differently than intrinsic and extrinsic schemas. In such works, the dataset is entirely used to build the expert graph, and for evaluation purposes, *synthetic* projects are generated, each of which is followed by an exhaustive search over *all* possible subgraphs of the expert graph. A subgraph that obtains the optimum value for a project is labelled as a gold team. Such evaluation schema has been also referred to as project generation [17, 24, 29, 40–42, 70, 72, 79, 93]. Based on these evaluation schemas, metrics have been employed to measure the effectiveness of an algorithm, as explained hereafter.

Quantitative Metrics. Evaluation metrics can be utilized where we have access to a gold standard, either in the context of an intrinsic evaluation methodology or a gold set acquired via an exhaustive search for a task (synthetic subset of skills). Intrinsic evaluation has been primarily measured via such classification metrics as precision and recall [1, 40, 72]. However, in project generation evaluation, the results of the methods are evaluated in comparison with the optimum subgraphs; a method that outputs a more similar subgraph or a subgraph whose objective value, e.g., communication cost, is closer to the optimum value is considered more effective [1, 33, 40, 41, 70–72]. Since an exhaustive search for optimum subgraphs might be computationally prohibitive, some works [24, 79, 93] simply omit the exhaustive search for an optimum subgraph per projects and proceed with the direct comparison between different methods based on how well they optimize objective functions; a method that yields the lowest (highest) value for an objective or a combination of objectives that are to be minimized (maximized) is considered as the most effective one on average over all of the projects. Some of the main objectives include minimizing the communication cost by searching for subgraphs with minimum diameter, sum of the edge weights, sum of distances between experts, or spanning tree. Other objectives include personnel cost and geographical distance (to be minimized), or trust score and expertise level (to be maximized), as explained in Section 2.2.

Qualitative Metrics. Last, some works have adopted qualitative evaluation methodologies to demonstrate the effectiveness of the predicted teams. Qualitative metrics are a go-to choice that make intuitive sense in the absence of a gold standard. This is similar to the case of project generation-based evaluation without exhaustive search. Even with the existence of gold standard data, this schema has been utilized in some works as complementary confidence in the methods' performance. Such qualitative measures are described below:

- *Team Size (Cardinality)* is the number of team members; effective teams are usually desired to be small and more cost-effective [70, 79, 80].
- *Team Connectivity* indicates the degree of connectedness of the team's members in their respective subgraph; densely connected team members show many previous collaborations, directly or via other team members, hence have lower communication costs [79].
- *The Number of Liaisons*, also referred to as intermediators [80] or connectors [41], are the number of team members that do not cover any of the required skills for the project but keep other skill-holder members connected in the respective subgraph of the team. It is desired to minimize the number of liaisons for a team.

- *Skill Count* shows how many times an expert member of a team has participated in projects in the past that shared the same required skill. This value is averaged over all required skills and all members of the team. A higher skill count indicates a greater level of experience, and hence better expertise, within the team concerning the set of skills required for the project [70].
- *Hypervolume* evaluates the quality of solutions in multi-objective optimization problems and measures how well a set of solutions covers the possible optimum outcomes across multiple objectives. A higher hypervolume value suggests a better distribution of non-dominated solutions and a more comprehensive representation of the trade-offs between different objectives. This can help in comparing and selecting solutions that provide a balanced and diverse set of outcomes in multi-objective optimization [1, 40, 72].
- *Average Distance* is a qualitative metric that measures the distance between teams formed by a definite algorithm (e.g., Pareto optimum teams) and a set of gold truth teams (e.g., outputs of a Pareto-exact algorithm). The distance is computed via the mean distance from each member of the first set and the closest member in the second set. The Maximal distance has also been used as a qualitative measure in several works [1, 40, 72], which computes the greatest distance between two mentioned sets.
- *TeamPubs* and *PartialTeamPubs* are metrics that have been used for the DBLP dataset yet can be generalized for other domains. For a recommended team, the TeamPubs is equal to the number of publications (collaboration) in the past whose authors (members) are exactly the same as in the recommended team, [58] and *PartialTeamPubs* is the number of publications where at least half of their authors belong to the recommended team [58]. A larger value of these metrics shows that the team is more likely to be successful due to its more successful collaborations in the past.
- *Number of Common Projects* (e.g., publications) is the number of joint projects shared by at *least two* experts of the recommended team [70].
- *Number of Top-cited Authors* is obtained by comparing the experts of the formed teams against the list of top-cited authors and is equal to the number of experts in the recommended team who are among the top cited authors [58].
- *Number of Non-dominated Solutions*, also referred to as the number of Pareto solutions, is the number of solutions that cannot be improved in one objective without compromising another objective.

4.2.2 Efficiency (Speed)

It is crucial to have an algorithm that not only forms an effective team, but also gets the answer in an acceptable time. Lappas et al. [79] is the first work to address the difficulty class of Expert Team Formation problems. They prove that finding a team that covers all the required skills and minimizes communication costs via the diameter or spanning tree is NP-complete. In the following, the difficulty class of other Expert Team Formation problems are also studied, such as, minimizing sum of distance [40, 58, 70, 71], minimizing sum of edge weight [41, 42] and maximizing density [58, 80]. They prove that all of these problems are NP-hard. Out of all Expert Team Formation optimization problems, forming a team with a leader is proved to be solved in polynomial time [70]. Researchers have proposed different approximation and heuristic workarounds to address the NP time complexities of such algorithms. For instance, to form a team by minimizing the diameter, some works [1, 58, 72, 79] propose algorithms that grow quadratically with the number of required skills. Table 5 is the summary of algorithms attempting to decrease time complexity categorized based on their optimization method as well as their time complexities.

As an empirical view on real data, several works [1, 17, 29, 33, 40, 70–72, 80] also studied time complexity as the elapsed run time of the algorithm (wall clock). Some works compare the run time of their algorithms with their own proposed algorithms, e.g., a greedy version or heuristic algorithms [1, 17, 29, 40, 71, 72, 80]. Another approach involves addressing the scalability of algorithms by studying their run time by increasing nodes of the graph or the number of required skills [1, 33, 70, 72]. While there is no work that compares the efficiency of their algorithms with other works, a joint analysis of efficiency (time complexity) vs. efficacy has been studied

Table 5. Time complexity of graph-based Expert Team Formation methods. (×) shows that the time complexity is overlooked.

Complexity Class	Optimization Method	Algorithm	Time Complexity	
NP-Hard / NP-Complete	Diameter	RarestFirst [79]	$O(\|V\|^2)$	
		MinDiamSol [17]	$O(\|S_p\|\|V\|^2(\log\|V\| + \log\|S_p\|)\log\|V\|)$	
		Approx{diameter, PCost} [1, 58]	$O(\|V\|^2\|S_p\|)$	
		Approx{PCost, diameter} [1, 58]	$O(\|V\|^2\|S_p\|(\log_2 \frac{MaxDiameter}{\epsilon} + 1))$	ϵ is an input precision
		Generalized Diameter [24]	×	
		MinDiameter [58]	$O(\|V\|^2)$	
		Generalized Diameter with Skill Grading [24]	×	
	MST	CoverSteiner [79]	$O(\|V\|^3)$	
		EnhancedSteiner [79]	$O(k\|E\|)$	k is the number of nodes added to form the extended graph.
		MinAggrSol [17]	$O(\|S_p\|\log\|S_p\|)$	
		GeneralizedEnhancedSteiner-Random [80]	×	
	Density	m-DensestAlk [58]	$O(\alpha\|V\|^3)$	$\alpha = \sum_{s\in S_p} l_s$
		GeneralizedEnhancedSteiner-Density [80]	×	
		GroupingDensity [80]	×	
	Sum of Distance	Best-SumDistance [70]	$O(\|S_p\|^2\|V\|^2)$	
		FirstPhase [40]	$O(\|S_p\|^2\|V\|^2)$	
		Assignment and Pruning (AP) [29]	×	
		Approx [71]	$O(\beta^2\|S_p\|^2)$	$\beta = \max_{s\in S_p}\|V_s\|$
		MCC-Rare [71]	$O(\beta^2\|S_p\|^3)$	$\beta = \max_{s\in S_p}\|V_s\|$
		MCC [71]	$O(\beta\gamma\|S_p\|^2)$	$\beta = \max_{s\in S_p}\|V_s\|$, $\gamma = \min_{s\in S_p}\|V_s\|$
		Replace [71]	$O(\beta\log\beta)$	$\beta = \max_{s\in S_p}\|V_s\|$
		MOCA [33]	×	
	Sum of Edge Weights	PC-CC [42]	×	
		CC [41]	$O(\beta\|S_p\|\|V\|)$	$\beta = \max_{s\in S_p}\|V_s\|$
Polynomial	Leader Distance	Best-Leader [70]	$O(\beta\|S_p\|\|V\|)$	$\beta = \max_{s\in S_p}\|V_s\|$

rarely [17, 29, 33, 71, 80]; the general findings in most of these works show that the more effective an algorithm, the lower its runtime.

4.3 Baselines

Surprisingly, almost all the works in graph-base Expert Team Formation forgo benchmarking their algorithms against other baselines or at-the-time state of the art, most presumably due to: *i*) lack of a common benchmark dataset, *ii*) unavailability of source codes, and *iii*) improvements have been mostly made for an *additional* objective, as opposed to the objective optimization method, which makes the comparison inevitably unfair. For example, Zihayat et al. [40] propose considering personnel cost and expertise level in addition to communication cost and seek multi-objective optimization through minimizing the sum of distance method. A fair comparison would be with other existing methods that optimize the same objectives; this has been rare and has led to no available baseline. To fill the gap and further support the efficiency or efficacy of their proposed algorithms, researchers compare their proposed algorithms with their variations. For example, Huang et al. [29] compare their method with its greedy versions, or Datta et al. [17] benchmark their method by comparison with Lappas et al.'s [79] algorithms, which have not considered Capacity.

Almost all proposed works compare their algorithms with their variations and only a few have been compared with the pioneering works by Lappas et al. [79] and Kargar et al. [71]. In Appendix A, we illustrate the cross-comparing of all proposed algorithms with each other in a table where the main diameter is mostly selected. Lappas et al.'s *RarestFirst* and *EnhanceSteiner* have been the main baselines for works that propose better optimization for communication cost. Also, a few works [80] that propose an extra node attribute on top of communication cost compare their methods with *modified* versions of Lappas et al.'s methods to have a fair comparison. Last, a common baseline to almost all proposed works has been a *random* algorithm, which randomly selects a subset of experts in the expert graph as a recommended team to provide a minimum comparison basis upon which the proposed methods desire to improve efficacy.

5 Challenges and Future Directions

With a bird's-eye view of the chronological evolution of approaches to Expert Team Formation, we observe that graph-based algorithms represent a significant leap forward by modeling collaboration and social ties utilizing expert graphs [79]. This breakthrough not only surpassed earlier methods that relied solely on operations research [23, 34, 69] but also established graph-based approaches as an indispensable consideration for both operations research [20, 36] and the emerging learning-based paradigm [44, 46, 97] in the field. While there are some aspects of graph-based methods that remain consistent, including expert graph construction and subgraph optimization for an optimum project team, one fundamental aspect has been variable over time, the optimization objectives. Initially, communication cost was the only constraint consistently optimized by early works and treated as a universal constraint [17, 24, 29, 31, 70, 79, 80]. Gradually, other critical constraints such as personnel cost [71, 72], proficiency [40], geographical distance [93] and evolving expert collaborations over time [33] have been included to enhance the effectiveness of the formed project team for the real-world applications. Moreover, early approaches focused on single-objective optimization [17, 24, 29, 31, 70, 79, 80]. However, multi-objective optimization methods began to emerge, initially optimizing both communication cost and personnel cost simultaneously [71]. As multi-objective optimization methods have expanded, different approaches have introduced new objectives to the model, such as trust score and authority, which are followed by others in different methods [33, 40, 72]. Meanwhile, graph-based methods faced significant challenges in efficiency and robustness. They are computationally intractable because subgraph optimizations are NP-hard [74]. Therefore, techniques for subgraph search space reduction and polynomial heuristics have been proposed. Also, as the expert networks are dynamic and collaborations occur in real-time, few methods have been proposed to accommodate regular changes to the expert graph while preventing a complete re-optimization computation [58, 71]. In the following, we go more in depth in some future research directions that show promise in further unleashing the potential of subgraph optimization in forming effective teams, prioritized based on their societal impact and immediate applications to real-world scenarios.

5.1 Fairness and Diversity

The primary focus of existing Expert Team Formation methods is to optimize objectives toward successful teams while largely ignoring fairness and diversity when recommending experts. It has been well-explored that data-driven methods that produce recommendations suffer from unfair biases. They result in discrimination and reduced visibility for an already disadvantaged group [62], disproportionate selection of popular candidates [51], and over/under-representation and racial/gender disparities [76] since they are trained on real-world datasets that already inherit hidden societal biases. Meanwhile, social science research provides compelling evidence about the synergistic effects of diversity on team performance [64]; diversity breeds innovation and increases teams' success

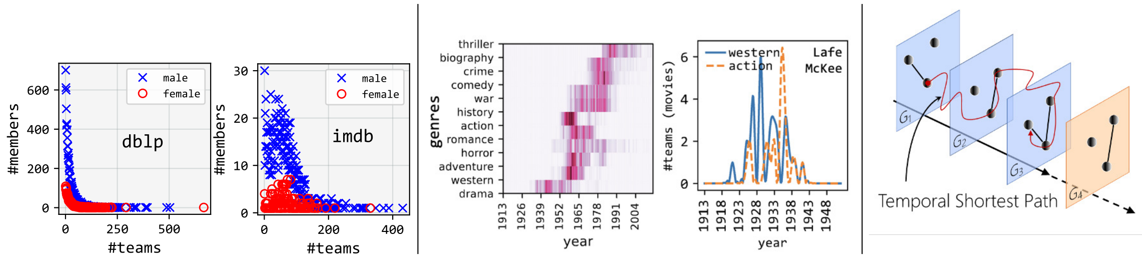


Fig. 7. Left: distribution of genders in DBLP and IMDB. Middle: temporal distribution of movies over genres, and temporal activity of an actor in two genres. Right: temporal expert graph as a stream of graph snapshots in each time interval from the past to the future.

by enabling a stronger sense of community and support, reducing conflict, and stimulating more creative thinking. As explained in Section 4 (Evaluation Methodology), having an expert who has participated in many teams previously, i.e., densely connected with other experts, is an advantage for a team. In spite of this or rather because of it, a method that pursues optimization of communication cost based on, e.g., spanning tree or sum of distances, would select/recommend such experts more often, leading to *popularity* bias and overlooking early career experts. Furthermore, no study has been performed on the datasets from the real world in terms of the distribution of demographic attributes like gender, race, or age over teams to uncover the hidden societal biases. As shown in Figure 7 (left), DBLP and IMDB datasets are suffering from gender bias; that is, teams are dominated by the majority males while females are heavily under-represented, which is overlooked by the proposed methods. To the best of our search, there is no fairness-aware algorithmic method that mitigates societal biases in this domain except that of the recent work by Barnabò et al. [26] that proves *fair* Expert Team Formation is NP-complete; therefore, computationally prohibitive for practical use. Therefore, the front-most future direction is to develop fairness-aware objectives and efficient (fast) optimization techniques in Expert Team Formation methods for recommending a *diverse* list of experts in terms of *i*) popularity and *ii*) demographic attributes (e.g., gender, race, or age), given required skills while almost surely promising the success of the recommended experts considering objectives such as communication cost. It is worth noting that a pure fairness-centric Expert Team Formation algorithm that solely overfits to satisfy fairness, neglecting the success of the team, is also undesirable for and unfair to the organizations, e.g., a team of experts from an under-represented group (e.g., all team members are early career experts or all team members are from the same cultural background) who are unable to accomplish the tasks.

In summary, existing Expert Team Formation methods need urgent further development to counter unfairness [47, 83]. The concrete research path would include: *i*) quantifying notions of fairness for teams, *ii*) automating forming fair yet successful teams, *iii*) mitigating existing unfair biases in real-world datasets by training strategies, *iv*) explore the synergistic balance of fusing bias mitigation methods based on notions of fairness into the Expert Team Formation.

5.2 Temporality

Experts' interests, skills, and levels of expertise change due to society's demands, novel technologies, and working experience. For instance, with the growth of automation, more and more experts are acquiring skills related to computer science, as seen in social science, biology, and linguistics, among other sciences [85]. Figure 7 (middle) demonstrates the *non*-uniform and temporal distribution of movies over genres (skills) and casts and crews (experts) within *yearly* time intervals in IMDB dataset. Although the set of genres remains the same over 100 years, the number of movies that adopt each genre varies over time (e.g., *'drama'* vs. *'musical'*). Further, we observe that actors who were active in 1930s are not active in years after 2000 (presumably due to aging). Therefore, should a method recommend actors in a yet-to-be-made movie in the genre of *'drama'*, it should learn the actors' temporal experiences in this genre from the *past* and avoid recommending *in*active or *in*experienced ones for the *future*.

Despite a large body of graph-based methods, the positive impacts of considering temporality have been considered in no work but in the Selvarajah et al. [33]'s where communication cost has been further weighted based on recency (Section 2.2.1). Also, there has been little work in operations research that used time as a *constraint* to model experts' availability or predefined start and due dates of projects. Durfee et al. [23] considered scheduling constraints or preferences in a two-step Expert Team Formation process. First, teams are built in the matchmaking optimization stage, taking into account the ability to be more readily (re)scheduled with respect to the timing requirement. Next, in the scheduling optimization stage, time slots are allotted to the team such that they minimize the starting times of all the members. Rahmanniyay et al. [92] studied the impact of various factors like weather conditions that can change the duration of a project or delay the delivery of material to a manufacturing company. Yang et al. [22] apply integer programming to determine the optimum team of experts available at a certain point in time. Contrary to considering time as an optimization constraint, consuming time as an *aspect* through which experts' skills and collaboration ties evolve remains an open question and is a promising research direction in graph-based Expert Team Formation methods.

As opposed to graph-based methods that are trained on a stationary expert graph, temporal methods need to consider an evolving expert graph whose expert node attributes (e.g., experts' level of expertise or authority), and edges (collaboration within time) change over time. A commonly proposed solution is to model a temporal expert graph as a stream of snapshots of expert graphs at each time interval, i.e., $\mathcal{G} = [G_0, G_1, \cdots, G_t, \cdots, G_T], 0 \leq t \leq T$, and generalize the definitions for the distance, shortest path, or random walk between a pair of experts but in graphs of different time intervals [37, 89], as shown in Figure 7 (right). Temporal graph has been successfully applied in route planning in a road network [53] and nervous system modeling [100]. Temporal study of Expert Team Formation problem finds immediate application for online, on-demand, and large-scale crowd work in crowdsourcing platforms [28, 50], where experts hold portfolios and timelines, showcasing their successes and additional evidence of their expertise through previously established credentials within time. In a crowdsourcing platform, candidate experts advertise their certified skills and bid prices for their participation while continuously adapting to changing task demands by upskilling or shifting their skills to remain relevant. While there has been Expert Team Formation research on crowdsourcing platforms using social networks [28, 50], no work has considered the temporal aspect of skill set and experts' skills.

5.3 Modern Hybrid Optimization Algorithms

While Pareto optimum is based on trade-offs among multiple, mostly conflicting, objectives, the search space is a single vector space, each dimension of which is one objective, and the target is to find a *single* optimum team among the set of feasible solutions. However, organizations, more often than not, aim to form *multiple* project teams from a shared pool of experts based on different sets of objectives, which may be interrelated. For instance, forming two project teams, one based on maximizing proficiency and minimizing personnel cost, and the other based only on communication cost. Available experts may have a limited capacity of one project team at a time, and hence, forming an optimum project in the first one influences the available pool of experts for the second one, and vice versa. As such, we face two optimization tasks: one bi-objective and one single objective, each of which has its own vector spaces that can be searched *simultaneously*, yet solutions from one space impact the solutions in the other one. To fill the gap, multi-task optimization techniques like multi-factorial optimization can be leveraged [25, 49, 81]. Particularly, multi-factorial optimization employs the implicit parallelism between multiple orthogonal or inter-related search spaces, each of which can be a single- or multi-objective search space individually, for simultaneous concurrent search while seamlessly transferring knowledge between different inter-related search spaces. Multi-factorial or other multi-task optimization algorithms to form inter-related project teams have remained unexplored in subgraph optimization algorithms despite their widespread success in operations research for other problems like engineering structural design [77], supply chain and logistics [94], and algorithmic trading [19], to name a few.

5.4 Open-Source Implementation

As shown in Appendix A (Baselines), the cross-comparison of the results among proposed algorithms is limited mainly because the proposed algorithms suffer from the lack of public availability in terms of implementation and experimental setup, limiting them being a comparative baseline. We investigated online public web pages associated with the authors of the graph-based papers; our search yielded *no* public code repositories, except that of Wang et al. [106] who *re*-implemented a limited number of Expert Team Formation algorithms, signifying that accessibility and reproducibility of proposed algorithms were *not* a concern among researchers in this area. An interesting avenue of research is to solicit benchmark libraries that repeat, reproduce, generalize, and analyze prior work with a focus on generating new findings from the re-application of established approaches akin to a test of time and investigating the extent to which assumptions of the original work hold up. Particularly, *reproducibility* efforts by different teams under different experimental setups, rather than *replicability* efforts, i.e., with different teams but the same experimental setup, could be of more interest for generating new research insights with the existing approaches.

Tutorials and workshops at conferences or forums, like the Society for Industrial and Applied Mathematics (SIAM)'s Symposium on Discrete Algorithms, the annual meeting by the Institute for Operations Research and the Management Sciences (INFORMS), and Special Interest Group on Information Retrieval (SIGIR), can introduce Expert Team Formation-specific challenges [38, 95, 96]. These challenges could feature realistic constraints from different domains and would encourage researchers to push the boundaries of current algorithms using cross-disciplinary methodologies, helping to foster hybrid models that take advantage of each paradigm's strengths. Leaderboards could drive ongoing improvements while creating a reference point for comparison across different methods, fostering fair comparative analysis. Ethical issues must be prioritized and promoted at conferences like Fairness, Accountability, and Transparency in AI (FAccT), to ensure benchmarks promote fairness and inclusivity while avoiding biases or harmful practices that arise from poorly designed datasets or biased assumptions in the Expert Team Formation process.

6 Concluding Remarks

We presented a survey of the graph-based category of algorithmic approaches to the Expert Team Formation problem, as the mainstream body of research in this domain. A comprehensive overview of seminal graph-based solutions, including proposed optimization objectives has been presented. We have identified a shortfall in this field to be the lack of standards and conventions, which we addressed in this survey by proposing and leveraging a unified set of notations to formalize the problem and its different aspects and sub-tasks. Adopting a chronological approach that reflects the advancements over time, we provided a detailed study of the graph-based optimization techniques in the Expert Team Formation. We introduced the benchmark datasets in this field along with their unique characteristics as well as methodologies and metrics for intrinsic, quantitative, and qualitative evaluations of proposed graph-based algorithms in terms of efficacy and efficiency. We identified five major lines of research as future directions based on the pinpointed open issues and challenges in the literature, including fairness concerns, temporality aspects of Expert Team Formation, modern hybrid optimization algorithms, and pitfalls to avoid in evaluation (Appendix C). We believe that these directions considerably help improve state-of-the-art not only in the graph-based within the search-based category but also can act as guidelines to consider in other novel learning-based works.

References

[1] A. An, M. Kargar, and M. Zihayat. 2013. Finding Affordable and Collaborative Teams from a Network of Experts. In *Proceedings of the 13th SIAM International Conference on Data Mining,*. 587–595. doi:10.1137/1.9781611972832.65

[2] A. Anagnostopoulos et al. 2012. Online team formation in social networks. In *Proceedings of the 21st international conference on World Wide Web*. 839–848.

[3] B. K. Baiden and A. D. Price. 2011. The effect of integration on project delivery team effectiveness. *International Journal of Project Management* 29, 2 (2011), 129–136.

[4] R. B. Bernabé, I. Navia, and F. J. García-Peñalvo. 2015. Faat: freelance as a team. In *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM*. 687–694. doi:10.1145/2808580.2808685

[5] M. T. Brannick et al. 1997. *Team performance assessment and measurement: Theory, methods, and applications.* Psychology Press.

[6] R. Burkard, M. Dell'Amico, and S. Martello. 2012. *Assignment problems: revised reprint.* doi:10.1137/1.9781611972238

[7] K. M. Bursic. 1992. Strategies and benefits of the successful use of teams in manufacturing organizations. *IEEE transactions on engineering management* 39, 3 (1992), 277–289.

[8] Mi. Cheatham and K. Cleereman. 2006. Application of Social Network Analysis to Collaborative Team Formation. In *2006 International Symposium on Collaborative Technologies and Systems, CTS*. 306–311. doi:10.1109/CTS.2006.18

[9] O. Chen, F. Paas, and J. Sweller. 2023. A cognitive load theory approach to defining and measuring task complexity through element interactivity. *Educational Psychology Review* 35, 2 (2023), 63.

[10] S. Chen. 2005. An Integrated Methodological Framework for Project Task Coordination and Team Organization in Concurrent Engineering. *Concurr. Eng. Res. Appl.* 13, 3 (2005), 185–197. doi:10.1177/1063293X05056462

[11] S. Chen and L. Lin. 2004. Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *IEEE Trans. Engineering Management* 51, 2 (2004), 111–124. doi:10.1109/TEM.2004.826011

[12] P. R. Cohen, H. J. Levesque, and I. A. Smith. 1997. On team formation. *Synthese Library* (1997), 87–114.

[13] M. Craig and D. McKeown. 2015. How to build effective teams in healthcare. *Nursing times* 111, 14 (2015), 16–18.

[14] K. Daniel. 2011. *Thinking, fast and slow.* macmillan.

[15] A. Dashti, S. Samet, and H. Fani. 2022. Effective Neural Team Formation via Negative Samples. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3908–3912. doi:10.1145/3511808.3557590

[16] A. Dashti, K. Saxena, D. Patel, and H. Fani. 2022. OpeNTF: A Benchmark Library for Neural Team Formation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3913–3917. doi:10.1145/3511808.3557526

[17] S. Datta, A. Majumder, and K. V. M. Naidu. 2012. Capacitated team formation problem on social networks. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*. 1005–1013. doi:10.1145/2339530.2339690

[18] A. Anagnostopoulos et al. 2010. Power in unity: forming teams in large-scale community systems. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM*. 599–608. doi:10.1145/1871437.1871515

[19] A. Gupta et al. 2017. Multiobjective Multifactorial Optimization in Evolutionary Multitasking. *IEEE Trans. Cybern.* 47, 7 (2017), 1652–1665. doi:10.1109/TCYB.2016.2554622

[20] C. Ding et al. 2017. TeamGen: An Interactive Team Formation System Based on Professional Social Network. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 195–199. doi:10.1145/3041021.3054725

[21] C. Lin et al. 2009. SmallBlue: Social Network Analysis for Expertise Search and Collective Intelligence. In *Proceedings of the 25th International Conference on Data Engineering, ICDE*. 1483–1486. doi:10.1109/ICDE.2009.140

[22] D. Yang et al. 2011. On Social-Temporal Group Query with Acquaintance Constraint. *Proc. VLDB Endow.* 4, 6 (2011), 397–408. doi:10.14778/1978665.1978671

[23] E. H. Durfee et al. 2014. Using hybrid scheduling for the semi-autonomous formation of expert teams. *Future Gener. Comput. Syst.* 31 (2014), 200–212. doi:10.1016/j.future.2013.04.008

[24] F. Farhadi et al. 2011. An Effective Expert Team Formation in Social Networks Based on Skill Grading. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference*. 366–372. doi:10.1109/ICDMW.2011.28

[25] F. Ming et al. 2024. Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer. *IEEE Trans. Evol. Comput.* 28, 1 (2024), 77–89. doi:10.1109/TEVC.2022.3230822

[26] G. Barnabò et al. 2019. Algorithms for Fair Team Formation in Online Labour Marketplaces. In *Companion of The World Wide Web Conference*. 484–490. doi:10.1145/3308560.3317587

[27] H. Nguyen et al. 2023. Learning heterogeneous subgraph representations for team discovery. *Inf. Retr. J.* 26, 1 (2023), 8. doi:10.1007/S10791-023-09421-6

[28] I. Lykourentzou et al. 2016. Team Dating: A Self-Organized Team Formation Strategy for Collaborative Crowdsourcing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1243–1249. doi:10.1145/2851581.2892421

[29] J. Huang et al. 2017. Forming Grouped Teams with Efficient Collaboration in Social Networks. *Comput. J.* 60, 11 (2017), 1545–1560. doi:10.1093/comjnl/bxw088

[30] J. Juárez et al. 2022. A Comprehensive Review and a Taxonomy Proposal of Team Formation Problems. *ACM Comput. Surv.* 54, 7 (2022), 153:1–153:33. doi:10.1145/3465399

[31] J. Nemec et al. 2021. RW-Team: Robust Team Formation using Random Walk. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management.* 4759–4763. doi:10.1145/3459637.3481969

[32] J. Tang et al. 2008. ArnetMiner: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 990–998. doi:10.1145/1401890.1402008

[33] K. Selvarajah et al. 2021. A unified framework for effective team formation in social networks. *Expert Syst. Appl.* 177 (2021), 114886. doi:10.1016/j.eswa.2021.114886

[34] L. Wang et al. 2020. Team Recommendation Using Order-Based Fuzzy Integral and NSGA-II in StarCraft. *IEEE Access* 8 (2020), 59559–59570. doi:10.1109/ACCESS.2020.2982647

[35] L. Wu et al. 2023. Graph Neural Networks for Natural Language Processing: A Survey. *Found. Trends Mach. Learn.* 16, 2 (2023), 119–328. doi:10.1561/2200000096

[36] M. B. Campêlo et al. 2020. The sociotechnical teams formation problem: a mathematical optimization approach. *Ann. Oper. Res.* 286, 1 (2020), 201–216. doi:10.1007/s10479-018-2759-5

[37] M. Latapy et al. 2018. Stream graphs and link streams for the modeling of interactions over time. *Soc. Netw. Anal. Min.* 8, 1 (2018), 61:1–61:29. doi:10.1007/S13278-018-0537-7

[38] M. Saeedi et al. 2024. Paradigm Shifts in Team Recommendation: From Historical Subgraph Optimization to Emerging Graph Neural Network. In *Proceedings of the International Conference on Research and Development in Information Retrieval in the Asia Pacific Region, SIGIR-AP*.

[39] M. Twyman et al. 2022. Teammate invitation networks: The roles of recommender systems and prior collaboration in team assembly. *Soc. Networks* 68 (2022), 84–96. doi:10.1016/J.SOCNET.2021.04.008

[40] M. Zihayat et al. 2014. Two-Phase Pareto Set Discovery for Team Formation in Social Networks. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).* 304–311. doi:10.1109/WI-IAT.2014.112

[41] M. Zihayat et al. 2017. Authority-based Team Discovery in Social Networks. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT*. 498–501. doi:10.5441/002/edbt.2017.54

[42] M. Zihayat et al. 2018. Effective Team Formation in Expert Networks. In *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management (CEUR Workshop Proceedings, Vol. 2100).* http://ceur-ws.org/Vol-2100/paper4.pdf

[43] R. Bing et al. 2023. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artif. Intell. Rev.* 56, 8 (2023), 8003–8042. doi:10.1007/s10462-022-10375-2

[44] R. Hamidi Rad et al. 2020. Learning to Form Skill-based Teams of Experts. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management.* 2049–2052. doi:10.1145/3340531.3412140

[45] R. Hamidi Rad et al. 2021. PyTFL: A Python-based Neural Team Formation Toolkit. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management.* 4716–4720. doi:10.1145/3459637.3481992

[46] R. Hamidi Rad et al. 2024. A Variational Neural Architecture for Skill-based Team Formation. *ACM Trans. Inf. Syst.* 42, 1 (2024), 7:1–7:28. doi:10.1145/3589762

[47] R. Moasses et al. 2024. vivaFemme: Mitigating Gender Bias in Neural Team Recommendation via Female-Advocate Loss Regularization. In *Advances in Bias and Fairness in Information Retrieval - 5th International Workshop, BIAS*, Vol. 2227. Springer, 78–90.

[48] S. Wu et al. 2023. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput. Surv.* 55, 5 (2023), 97:1–97:37. doi:10.1145/3535101

[49] W. Han et al. 2023. Multi-regularization sparse reconstruction based on multifactorial multiobjective optimization. *Appl. Soft Comput.* 136 (2023), 110122.

[50] W. Wang et al. 2017. Toward Efficient Team Formation for Crowdsourcing in Noncooperative Social Networks. *IEEE Trans. Cybern.* 47, 12 (2017), 4208–4222. doi:10.1109/TCYB.2016.2602498

[51] Y. Zhang et al. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 11–20. doi:10.1145/3404835.3462875

[52] Z. Alshara et al. 2023. PI-Link: A Ground-Truth Dataset of Links Between Pull-Requests and Issues in GitHub. *IEEE Access* 11 (2023), 697–710. doi:10.1109/ACCESS.2022.3232982

[53] Z. Luo et al. 2022. Diversified Top-k Route Planning in Road Network. *Proc. VLDB Endow.* 15, 11 (2022), 3199–3212. doi:10.14778/3551793.3551863

[54] Z. Wu et al. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24. doi:10.1109/TNNLS.2020.2978386

[55] Z. Ye et al. 2022. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. *IEEE Access* 10 (2022), 75729–75741. doi:10.1109/ACCESS.2022.3191784

[56] E. Fitzpatrick and R. G. Askin. 2005. Forming effective worker teams with multi-functional skill requirements. *Comput. Ind. Eng.* 48, 3 (2005), 593–608. doi:10.1016/j.cie.2004.12.014

[57] R. W. Floyd. 1962. Algorithm 97: Shortest path. *Commun. ACM* 5, 6 (1962), 345. doi:10.1145/367766.368168
[58] A. Gajewar and A. Das Sarma. 2012. Multi-skill Collaborative Teams based on Densest Subgraphs. In *Proceedings of the Twelfth SIAM International Conference on Data Mining*. 165–176. doi:10.1137/1.9781611972825.15
[59] M. E. Gaston, J. Simmons, and M. desJardins. 2004. Adapting Network Structure for Efficient Team Formation. In *Artificial Multiagent Learning, Papers from the 2004 AAAI*, Vol. FS-04-02. 1–8. https://www.aaai.org/Library/Symposia/Fall/2004/fs04-02-001.php
[60] R. Gyanchandani, B. Nathani, and D. Jaroliya. 2019. Factors Affecting Team Performance in IT Sector: An Exploratory Analysis. *J-GIBS* 11, 1 (2019).
[61] J. R Hackman. 2002. *Leading teams: Setting the stage for great performances.* Harvard Business School Press.
[62] S. Hajian, F. Bonchi, and C. Castillo. 2016. Algorithmic Bias: From Discrimination Discovery to Fairness-aware Data Mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2125–2126. doi:10.1145/2939672.2945386
[63] G. Hirst. 2011. The handbook of computational linguistics and natural language processing. *Language* 87, 4 (2011), 897–899.
[64] B. Hofstra et al. 2020. The diversity–innovation paradox in science. *Proceedings of the National Academy of Sciences* 117, 17 (2020), 9284–9291.
[65] D. Horowitz and S. D. Kamvar. 2010. The anatomy of a large-scale social search engine. In *Proceedings of the 19th International Conference on World Wide Web, WWW*. 431–440. doi:10.1145/1772690.1772735
[66] D. Horowitz and S. D. Kamvar. 2012. Searching the village: models and methods for social search. *Commun. ACM* 55, 4 (2012), 111–118. doi:10.1145/2133806.2133830
[67] T. Jansen. 2013. *Analyzing Evolutionary Algorithms - The Computer Science Perspective.* doi:10.1007/978-3-642-17339-4
[68] J. Juárez and C. A. Brizuela. 2018. A multi-objective formulation of the team formation problem in social networks: preliminary results. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO*. 261–268. doi:10.1145/3205455.3205634
[69] S. J. Kalayathankal et al. 2021. A modified fuzzy approach to project team selection. *Soft Computing Letters* 3 (2021), 100012.
[70] M. Kargar and A. An. 2011. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM*. 985–994. doi:10.1145/2063576.2063718
[71] M. Kargar, A. An, and M. Zihayat. 2012. Efficient Bi-objective Team Formation in Social Networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD (Lecture Notes in Computer Science, Vol. 7524)*. 483–498. doi:10.1007/978-3-642-33486-3_31
[72] M. Kargar, M. Zihayat, and A. An. 2013. *Affordable and collaborative team formation in an expert network.* Technical Report. Department of Computer Science and Engineering, York University.
[73] M. Kargaret al. 2020. Effective keyword search over weighted graphs. *IEEE Transactions on Knowledge and Data Engineering* 34, 2 (2020), 601–616.
[74] R. M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proceedings of a symposium on the Complexity of Computer Computations (The IBM Research Symposia Series)*. 85–103. doi:10.1007/978-1-4684-2001-2_9
[75] J. R. Katzenbach and D. K. Smith. 2015. *The wisdom of teams: Creating the high-performance organization.* Harvard Business Review Press.
[76] M. Kay, C. Matuszek, and S. A. Munson. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for OccupationsCHI. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3819–3828. doi:10.1145/2702123.2702520
[77] B. Kodess et al. 2022. Reference data and multi-factorial methods for development of modern materials. *Key Engineering Materials* 910 (2022), 966–975.
[78] A. Lancichinetti, S. Fortunato, and F. Radicchi. 2008. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* 78 (Oct 2008), 046110. Issue 4. doi:10.1103/PhysRevE.78.046110
[79] T. Lappas, K. Liu, and E. Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 467–476. doi:10.1145/1557019.1557074
[80] C. Li and M. Shan. 2010. Team Formation for Generalized Tasks in Expertise Social Networks. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT*. 9–16. doi:10.1109/SocialCom.2010.12
[81] G. Li, Q. Lin, and W. Gao. 2020. Multifactorial optimization via explicit multipopulation evolutionary framework. *Inf. Sci.* 512 (2020), 1555–1570. doi:10.1016/J.INS.2019.10.066
[82] P. Liu and Z. Li. 2012. Task complexity: A review and conceptualization framework. *International Journal of Industrial Ergonomics* 42, 6 (2012), 553–568.
[83] H. Loghmani and H. Fani. 2023. Bootless Application of Greedy Re-ranking Algorithms in Fair Neural Team Formation. In *Advances in Bias and Fairness in Information Retrieval - 4th International Workshop, BIAS*, Vol. 1840. Springer, 108–118.
[84] J. Magill-Evans, M. Hodge, and J. Darrah. 2002. Establishing a transdisciplinary research team in academia. *Journal of allied health* 31, 4 (2002), 222–226.

[85] T. H. McCormick et al. 2017. Using Twitter for demographic and social science research: Tools for data collection and processing. *Sociological methods & research* 46, 3 (2017), 390–421.
[86] D. H. McKnight, V. Choudhury, and C. J. Kacmar. 2002. The impact of initial consumer trust on intentions to transact with a web site: a trust building model. *J. Strateg. Inf. Syst.* 11, 3-4 (2002), 297–323. doi:10.1016/S0963-8687(02)00020-3
[87] G. A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review* 63, 2 (1956), 81.
[88] Megan Muniz and Tülay Flamand. 2024. A column generation approach for the team formation problem. *Comput. Oper. Res.* 161 (2024), 106406.
[89] L. Oettershagen, P. Mutzel, and N. M. Kriege. 2022. Temporal Walk Centrality: Ranking Nodes in Evolving Networks. In *WWW '22: The ACM Web Conference 2022*. 1640–1650. doi:10.1145/3485447.3512210
[90] L. Paquete, M. Chiarandini, and T. Stützle. 2004. Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study. In *Metaheuristics for Multiobjective Optimisation*. Vol. 535. 177–199. doi:10.1007/978-3-642-17144-4_7
[91] C. R. Paris, E. Salas, and J. A. Cannon-Bowers. 2000. Teamwork in multi-person systems: a review and analysis. *Ergonomics* 43, 8 (2000), 1052–1075.
[92] F. Rahmanniyay, A. J. Yu, and J. Seif. 2019. A multi-objective multi-stage stochastic model for project team formation under uncertainty in time requirements. *Comput. Ind. Eng.* 132 (2019), 153–165. doi:10.1016/j.cie.2019.04.015
[93] S. S. Rangapuram, T. Bühler, and M.s Hein. 2013. Towards realistic team formation in social networks based on densest subgraphs. In *22nd International World Wide Web Conference, WWW*. 1077–1088. doi:10.1145/2488388.2488482
[94] A. Rauniyar, R. Nath, and P. K. Muhuri. 2019. Multi-factorial evolutionary algorithm based novel solution approach for multi-objective pollution-routing problem. *Comput. Ind. Eng.* 130 (2019), 757–771. doi:10.1016/J.CIE.2019.02.031
[95] M. Saeedi, Ch. Wong, and H. Fani. 2024. Collaborative Team Recommendation for Skilled Users: Objectives, Techniques, and New Perspectives. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, UMAP*.
[96] M. Saeedi, Ch. Wong, and H. Fani. 2025. Bridging Historical Subgraph Optimization and Modern Graph Neural Network Approaches in Team Recommendations. In *18th International Conference on Web Search and Data Mining (WSDM)*.
[97] A. Sapienza, P. Goyal, and E. Ferrara. 2019. Deep Neural Networks for Optimal Team Composition. *Frontiers Big Data* 2 (2019), 14. doi:10.3389/fdata.2019.00014
[98] S. Sharma and V. Kumar. 2022. A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering* 29, 7 (2022), 5605–5633.
[99] P. D. Sherer. 1995. Leveraging Human Assets in Law Firms: Human Capital Structures and Organizational Capabilities. *ILR Review* 48, 4 (1995), 671–691. doi:10.1177/001979399504800405
[100] A. Sizemore and D. S. Bassett. 2018. Dynamic graph metrics: Tutorial, toolbox, and tale. *NeuroImage* 180, Part (2018), 417–427. doi:10.1016/J.NEUROIMAGE.2017.06.081
[101] M. Sozio and A. Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 939–948. doi:10.1145/1835804.1835923
[102] D. Stokols et al. 2008. The science of team science: overview of the field and introduction to the supplement. *American journal of preventive medicine* 35, 2 (2008), S77–S89.
[103] R. I. Swaab et al. 2014. The too-much-talent effect: Team interdependence determines when more talent is too much or not enough. *Psychological Science* 25, 8 (2014), 1581–1591.
[104] H. Takahashi. 1980. An approximate solution for the Steiner problem in graphs. *Math. Japonica* (1980), 573–577.
[105] V. Vazirani. 2001. *Approximation algorithms* (1st ed.). Springer.
[106] X. Wang, Z. Zhao, and W. Ng. 2015. A Comparative Study of Team Formation in Social Networks. In *Database Systems for Advanced Applications - 20th International Conference, DASFAA (Lecture Notes in Computer Science, Vol. 9049)*. 389–404. doi:10.1007/978-3-319-18120-2_23
[107] F. Xhafa, L. Barolli, and A. Durresi. 2007. Batch mode scheduling in grid systems. *International Journal of Web and Grid Services* 3, 1 (2007), 19–37.
[108] A. Zzkarian and A. Kusiak. 1999. Forming teams: an analytical approach. *IIE transactions* 31, 1 (1999), 85–97.

Received 29 November 2023; revised 3 March 2025; accepted 8 May 2025