Parameterized Complexity of Team Formation in Social Networks

Robert Bredereck[1], Jiehua Chen[1], Falk Hüffner[1], and Stefan Kratsch[2]

[1] Institut für Softwaretechnik und Theoretische Informatik,
Technische Universität Berlin, Germany
[2] Institut für Informatik I, Universität Bonn, Germany
{robert.bredereck, jiehua.chen, falk.hueffner}@tu-berlin.de
kratsch@cs.uni-bonn.de

Abstract. Given a task that requires some skills and a social network of individuals with different skills, the Team Formation problem asks to find a team of individuals that together can perform the task, while minimizing communication costs. Since the problem is NP-hard, we identify the source of intractability by analyzing its parameterized complexity with respect to parameters such as the total number of skills k, the team size l, the communication cost budget b, and the maximum vertex degree Δ. We show that the computational complexity strongly depends on the communication cost measure: when using the weight of a minimum spanning tree of the subgraph formed by the selected team, we obtain fixed-parameter tractability for example with respect to the parameter k. In contrast, when using the diameter as measure, the problem is intractable with respect to any single parameter; however, combining Δ with either b or l yields fixed-parameter tractability.

1 Introduction

Assembling teams based on required skills is a classic management task. Recently, it has been suggested to take into account not only the covering of the required skills, but also the expected communication costs (see Lappas et al. [11] for a survey). This cost can be estimated based on a given edge-weighted social network, where a low weight value on an edge between two individuals indicates a low communication cost. For example, edge weights can reflect distance in an organizational chart or the number of joint projects completed.

Lappas et al. [10] formalized the setting as the optimization problem of minimizing the communication cost and studied two cost measures: the diameter (Diam) and the weight of a minimum spanning tree (Mst). For our complexity analysis, we formulate it as a decision problem by fixing the maximum team size.

Diam-Team Formation
Input: An undirected graph $G = (V, E)$ with edge-weight function $w\colon E \to \mathbb{N}$, a set T of k skills, a skill function $S\colon V \to 2^T$, a team size $l \in \mathbb{N}$, and a budget $b \in \mathbb{N}$.
Question: Is there a subset $V' \subseteq V$ with $|V| \leq l$ such that $\bigcup_{v\in V'} S(v) = T$ and the w-weighted diameter of the induced subgraph $G[V']$ is at most b?

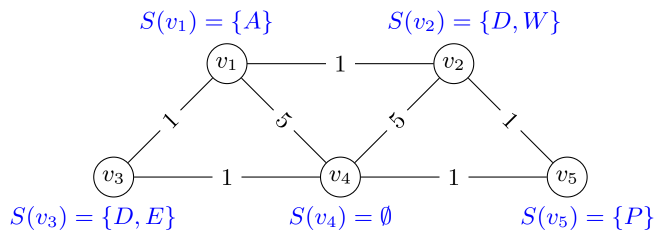

Fig. 1. A TEAM FORMATION example: a social network of five potential team members and five skills, "algorithms" (A), "data bases" (D), "software engineering" (E), "programming" (P), and "web programming" (W). When minimizing the weight of a minimum spanning tree of the subgraph induced by a team (MST), the team with members v_1, v_2, v_3, v_5 has the lowest cost, 3, (one can build a path with weight one at each edge). However, when minimizing the diameter of the subgraph induced by a team (DIAM), it is worthwhile to add the individual v_4—who has no specific skill—to reduce the diameter of 3 in $G[\{v_1, v_2, v_3, v_5\}]$ to the diameter of 2 in $G[\{v_1, v_2, v_3, v_4, v_5\}]$.

Here, the *diameter* of an edge-weighted graph G, denoted as DIAM(G), is the maximum distance between any two vertices in the input graph and the *distance* between two vertices is the minimum sum of the weights of the edges along any path between these two vertices. Our formulation of the team formation problem allows to choose individuals (vertices) that do not contribute any skills, but serve as intermediate vertices to lower overall communication costs. We further assume w.l.o.g. that no individual has a skill that is not in the request set T.

The *weight of a minimum spanning tree* of graph G, MST(G), is the smallest sum of the weights of the edges in a spanning tree of G. We define the corresponding MST-TEAM FORMATION problem by replacing "diameter" in the definition of DIAM-TEAM FORMATION with "weight of a minimum spanning tree".

Figure 1 illustrates an example for the DIAM-TEAM FORMATION and MST-TEAM FORMATION problems. Lappas et al. [10] showed that both problems are NP-complete. Experiments on DIAM-TEAM FORMATION, MST-TEAM FORMATION and similar team formation problems so far use heuristic algorithms [1, 4, 8, 10, 12]. However, it might be that instances encountered in practice are actually easier than a one-dimensional complexity analysis suggests, and can be solved optimally. For example, it might be reasonable to assume that only a small number of skills is required. Thus, we try to identify the sources of intractability by a parameterized complexity analysis.

Optimization variant. There are two natural ways to define approximate solutions for our team formation problem. First, one allows to find solutions with larger communication costs. This leads to the MINCOST-ζ-TEAM FORMATION problem, ζ being either DIAM or MST, which asks for a vertex subset $V' \subseteq V$ with $|V'| \leq l$ such that $\bigcup_{v \in V'} S(v) = T$ and the communication cost $\zeta(G[V'])$ is minimized. Second, one allows to find solutions with larger teams. This leads to the MIN-

TEAMSIZE-ζ-TEAM FORMATION problem, which asks for a minimum vertex subset $V' \subseteq V$ such that $\bigcup_{v \in V'} S(v) = T$ and $\zeta(G[V']) \leq b$.

Cost measure "diameter". Arkin and Hassin [2] studied MINCOST-DIAM-TEAM FORMATION with unlimited team size l under the name MULTIPLE-CHOICE COVER. They showed that even when no skill is allowed to be covered by more than three team members the problem cannot be approximated with a constant-factor error guarantee, unless P = NP. However, when the weights satisfy the triangle inequality, a 2-approximation is possible; this bound is sharp [2].

Cost measure "minimum spanning tree". As already mentioned by Lappas et al. [10], the MINCOST-MST-TEAM FORMATION problem with an unlimited team size l is equivalent to the GROUP STEINER TREE problem: given an undirected edge-weighted graph $G = (V, E)$ and vertex subsets (*groups*) $g_i \subseteq V$, $1 \leq i \leq k$, find a subtree $T = (V_T, E_T)$ of G such that $V_T \cap g_i \neq \emptyset$ for all $1 \leq i \leq k$ and the cost $\sum_{e \in E_T} w(e)$ is minimized. Clearly, each group of GROUP STEINER TREE corresponds to a subset of vertices in MST-TEAM FORMATION that have a particular skill. From this relation to GROUP STEINER TREE and an inapproximability result of Halperin and Krauthgamer [9], we obtain that it is unlikely that MINCOST-MST-TEAM FORMATION can be approximated to a factor of $O(\log^{2-\epsilon} k)$ for any $\epsilon > 0$, where k is the number of skills to be covered.

Despite the polylogarithmic inapproximability result, we can obtain fixed-parameter tractability for the parameter "number k of skills to be covered". First, we reduce the MST-TEAM FORMATION problem with limited team size l to the MST-TEAM FORMATION problem with an unlimited team size by adding a large weight W (for example the sum over all edge weights) to each edge weight and adding $l \cdot W$ to the budget. Then, by the relation between MST-TEAM FORMATION and GROUP STEINER TREE, we can think of the resulting instance as a GROUP STEINER TREE instance, which can be solved by reducing it to STEINER TREE: introduce a new vertex for each group and connect it to each vertex contained in this group by an edge with very high weight. The resulting STEINER TREE instance can be solved using inclusion–exclusion in $O^*(2^k)$ time and polynomial space when the edge weights are integers [13] (the O^* notation omits factors polynomial in the input size); for arbitrary weights, it can be solved in $O^*(3^k)$ time and exponential space by dynamic programming [6].

Further related work. Our team formation problem can be generalized in different ways. First, we can require each skill to be covered by a given number of team members instead of once. Li and Shan [12] proposed three heuristics for this problem; Gajewar and Sarma [8] studied it with the objective of maximizing the *collaborative compatibility*, an alternative to DIAM and MST. Here, the collaborative compatibility is the sum of the weights of all edges in the subgraph induced by the team divided by the team size (the number of vertices in the subgraph). They showed that this version is also NP-hard and provided a 1/3-approximation algorithm. Second, we can additionally require the workload to be balanced within the team [1, 4].

A number of experimental studies examine the validity of these models, using data for example from bibliography databases [1, 8, 10, 12] or the GitHub programming collaboration platform [4].

Diam-Team Formation and Mst-Team Formation have also applications in keyword search in relational databases [15]: the vertices in the graph correspond to tables, edges represent foreign key relationships, and skills model keywords that match the table. A subgraph covering all keywords with small communication costs helps to create efficient SQL queries.

Parameterized complexity. Parameterized algorithmics analyzes problem difficulty not only in terms of the input size, but also for an additional parameter, typically an integer p. Thus, formally, an instance of a parameterized problem is a tuple of the unparameterized instance I and the parameter p. A parameterized problem with parameter p is *fixed-parameter tractable* (FPT) if there is an algorithm that decides each instance (I, p) in $f(p) \cdot |I|^{O(1)}$ time, where f is a computable function depending only on p; we call this algorithm a *fixed-parameter algorithm*. In such case, we say that our problem can be solved in FPT-time for the parameter p. Clearly, if the problem is NP-hard, we must expect f to grow superpolynomially.

There are parameterized problems for which there is good evidence that no fixed-parameter algorithms exist. Analogously to the concept of NP-hardness, the concept of W[1]-*hardness* was developed. It is widely assumed that a W[1]-hard problem cannot have a fixed-parameter algorithm (hardness for the classes W[t], $t \geq 2$ has the same implication). To show that a problem is W[t]-hard, a *parameterized reduction* from a known W[t]-hard problem can be used. This is a reduction that runs in FPT-time and maps the parameter p to a new parameter p' that is upper-bounded by some function $g(p)$. We refer to recent text books [3, 5, 7, 14] for details on parameterized complexity theory and W[t]-complete problems.

Contributions. We focus on the parameterized complexity of Diam-Team Formation, which has to the best of our knowledge not been considered before. We consider parameters that are related to the communication cost and to the input graph: the number k of skills to be covered, the cost budget b, the maximum vertex degree Δ, and the team size l.

For the parameter l, Diam-Team Formation is W[2]-hard even with either constant budget b or constant maximum degree Δ (Proposition 1). For the parameter k, while Mst-Team Formation is fixed-parameter tractable, Diam-Team Formation is W[1]-hard even on graphs of maximum vertex degree three and with unrestricted team size l (Theorem 1). For the combined parameter $l + k$, Diam-Team Formation is W[1]-hard even if the cost budget b is two (Theorem 2). Concerning the parameter maximum degree Δ, we find that the problem is NP-hard even if the graph is a caterpillar with maximum degree $\Delta = 3$ (Proposition 1), where a caterpillar is a tree in which all the vertices are within distance one of a central path. Our results rule out fixed-parameter tractability for all considered single parameters and several parameter combinations.

By our parameterized hardness reductions, we can obtain that MinCost-Diam-Team Formation is inapproximable even when we allow for a superpolynomial running time factor in the team size l, even on complete graphs or on stars (Corollary 1). MinTeamsize-Diam-Team Formation is inapproximable even when we allow for a superpolynomial running time factor in the number k of skills, even on graphs with maximum degree $\Delta = 3$ (Corollary 2).

Geared towards robustness, we also consider the situation where the subgraph induced by the team is two-connected (that is, between each two team members, there are at least two edge-disjoint paths). We find that unless $\mathsf{FPT} = \mathsf{W}[1]$, it is unlikely that there is an algorithm that forms a team of size at most l, covering all k skills and inducing a two-connected subgraph, in $f(k+l) \cdot |I|^{O(1)}$ time, where $|I|$ denotes the size of our input instance (Theorem 3).

On the positive side, we provide some tractability results: Diam-Team Formation can be solved in $O^*(\Delta^{\Delta^b} \cdot \text{dcheck})$ time and in $O^*(\Delta^l \cdot \text{dcheck})$ time (Theorem 4), where dcheck denotes the running time of checking whether a subgraph has diameter most b, which can for example be solved in $O(\Delta \cdot n^2 \cdot \log(n))$ time by Dijkstra's algorithm. Finally, if the input graph is a tree, then we obtain that Diam-Team Formation is fixed-parameter tractable for parameter k (Theorem 5).

2 Hardness results

Throughout this section, we assume each edge in the input graph to have weight one; thus, we omit the introduction of the edge weight function w. We will see that our Diam-Team Formation problem is already hard in this setting. First, to get a feeling for the computational hardness of our Team Formation model we start with a simple observation which basically says that Diam-Team Formation with an unbounded number k of skills is basically at least as hard as the Set Cover problem, even on simple graph classes.

Set Cover
Input: A set family $\mathcal{F} = \{F_1, \ldots, F_\alpha\}$ over a universe $U = \{u_1, \ldots, u_\beta\}$ and a non-negative integer h.
Question: Is there a *set cover* of size at most h, that is, a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \leq h$ such that $\bigcup_{F \in \mathcal{F}'} F = U$?

Observation 1 *For edge weight one,* Diam-Team Formation *parameterized by the team size l generalizes* Set Cover *parameterized by the set cover size h, even on simple graph classes such as (1) complete graphs, (2) stars, and (3) caterpillars with maximum vertex degree three.*

Proof. Given a Set Cover instance $(\mathcal{F}, U, h)$, we construct a Diam-Team Formation instance $(G = (V, E), T, S, l, b)$ for each of the settings as follows.

(1) Define the skill set $T := U$, and for each set $F_i \in \mathcal{F}$, create one vertex v_i and define $S(v_i) := F_i$. Add an edge between each pair of vertices to obtain a complete graph. Finally, define the team size $l := h$ and let the cost budget b be an arbitrary integer at least one.

(2) Define the skill set $T \coloneqq U$, and for each set $F_i \in \mathcal{F}$, create one vertex v_i and define $S(v_i) \coloneqq F_i$. Add a special skill 0 to T, add a center vertex v_r to V, and define $S(v_r) \coloneqq \{0\}$. Construct a star graph with center v_r by adding an edge between each vertex v_i and v_r, $1 \leq i \leq \alpha$. Finally, define the team size $l \coloneqq h$, and let the cost budget b be an arbitrary integer at least two.
(3) Define the skill set $T \coloneqq U \uplus \{0, 1, 2\}$, and for each set $F_i \in \mathcal{F}$, create two vertices u_i and v_i. Define $S(v_i) \coloneqq F_i$ for all $1 \leq i \leq \alpha$, $S(u_1) \coloneqq \{0\}$ and $S(u_\alpha) \coloneqq \{2\}$, and $S(u_i) \coloneqq \{1\}$ for all $1 < i < \alpha$. Add an edge between u_i and u_{i+1} for all $1 \leq i < \alpha$ and and edge between u_i and v_i for all $1 \leq i \leq \alpha$. Finally, define the team size l to be at least $h + \alpha$, and the cost budget $b \coloneqq \alpha + 2$.

It is easy to verify that the constructed instances are yes-instances if and only if $(\mathcal{F}, U, h)$ is a yes-instance. □

In terms of parameterized and classical complexity analysis, Observation 1 yields the following hardness results.

Proposition 1. *Even when each edge has weight one, the following holds. (1)* Diam-Team Formation *parameterized by the team size l is* W[2]*-hard even if the budget b is one and the graph is complete. (2)* Diam-Team Formation *parameterized by the team size l is* W[2]*-hard even if the budget b is two and the graph is a star. (3)* Diam-Team Formation *is* NP*-hard even on caterpillar graphs with maximum degree three.*

We note that the budget in the proof of Statements (1)–(2) in Observation 1 as well as the team size in the proof of Statements (3) may have extremely large values that effectively do not upper-bound the communication costs or the team size. Since Set Cover is NP-complete and W[2]-complete when parameterized by h, in terms of minimizing the communication cost or team size, we have the following inapproximability result.

Corollary 1. *Unless all problems in* W[2] *are fixed-parameter tractable,* Min-Cost-Diam-Team Formation *is inapproximable even in* FPT*-time for the parameter team size l, even on complete graphs or on stars. Unless* P = NP*,* MinTeamsize-Diam-Team Formation *is inapproximable even in polynomial time, even on caterpillar graphs with maximum degree three.*

Consequently, to identify tractable cases one should start with cases where Set Cover is tractable. A very well-motivated restriction for Diam-Team Formation is to assume that there are not too many skills to cover, that is, the number k of skill is (part of) the parameter. Our next result, however, shows that this assumption (alone) does not make the problem fixed-parameter tractable.

Theorem 1. Diam-Team Formation *parameterized by the number k of skills is* W[1]*-hard, even on graphs with maximum degree three and with each edge weight one, and when the team size is unrestricted.*

Proof. We give a parameterized reduction from the W[1]-complete problem MULTICOLORED CLIQUE parameterized by the clique size h to TEAM FORMATION parameterized by k on graphs of maximum degree three.

> MULTICOLORED CLIQUE
> **Input:** An undirected graph $G = (V, E)$, a non-negative integer $h \in \mathbb{N}$, and a vertex coloring $\phi\colon V \to \{1, 2, \ldots, h\}$.
> **Question:** Does G admit a colorful h-clique, that is, a size-h vertex subset $Q \subseteq V$ such that the vertices in Q are pairwise adjacent and have pairwise distinct colors?

Let (G, ϕ, h) be a MULTICOLORED CLIQUE instance. We construct in FPT-time an equivalent DIAM-TEAM FORMATION instance $(G' = (V', E'), T, S, l, b)$. Without loss of generality, we assume that in G all edges are between vertices of different colors (according to ϕ) since they could be deleted without changing presence of a colorful h-clique. Let $n := |V|$, let $V = \{v_1, \ldots, v_n\}$ be an arbitrary ordering of the vertices, and let y be the smallest integer with $n \leq 2^y$.

Construction. We construct the graph G' by adding to V' all vertices in V (as an independent set). We connect the vertices in V by the following three steps:

1. Attach to each v_i a path of length s (to be determined later) whose other endpoint is denoted w_i; i.e., the distance between v_i and w_i shall be s. We call this path, including v_i and w_i the *path of* v_i.
2. Make each w_i the root of a newly added complete binary tree of height y, i.e., with 2^y leaves. Arbitrarily pick any n of its leaves and assign them names $x_{i,1}, \ldots, x_{i,n}$. Thus, $x_{i,j}$ will be the jth leaf in the binary tree attached by a path of length s (via w_i) to vertex v_i. We call the binary tree with root w_i and leaves $x_{i,1}, \ldots, x_{i,n}$ the *binary tree of* v_i.
3. Finally, we encode the adjacency from G as follows: If v_i and v_j are vertices that are adjacent in G, which implies by our assumption that they have different colors, i.e., $\phi(v_i) \neq \phi(v_j)$, then add an edge between $x_{i,j}$ and $x_{j,i}$. Thus, a leaf in the binary tree of v_i (namely $x_{i,j}$) is now adjacent to a leaf in the binary tree of v_j (namely $x_{j,i}$). Note that the naming convention of leaves prevents using each leaf for more than one adjacency.

This completes the construction of the graph G'. The graph can be constructed in FPT-time; indeed, it can be computed in polynomial time. Its maximum degree is three. Observe that if v_i and v_j are adjacent in G, then they have distance at most $2s + 2y + 1$ in G'. We set $s := 4n$ (with the intention of having vertices in two binary trees with adjacent leaves be at distance at most $4n$).

To complete the construction define the skill set $T := \{1, 2, \ldots, h\}$ and the skill function $S\colon V \to 2^T$ such that $S(v) := \{\phi(v)\}$ for all vertex $v \in V \subseteq V'$, i.e., for all vertices of the input graph G the skill equals the color according to ϕ, and $S(v) = \emptyset$ for all further vertices of $V' \setminus V$. The budget b (the diameter) is set to $b := 2s + 2y + 1$. Finally, the team size l is set to $|V'|$, i.e., the team size is effectively unbounded. We return instance $(G' = (V', E'), T, S, l, b)$.

The equivalence of (G, ϕ, h) and (G', T, S, l, b) is omitted due to space. □

We know from Proposition 1 that Diam-Team Formation is W[2]-hard for the parameter team size l and from Theorem 1 that Diam-Team Formation is W[1]-hard for the parameter number k of skills (in both cases even if the maximum degree Δ is a small constant). This invokes the question whether our problem becomes tractable for the combined parameter $l + k$, that is, for cases where both the team size and the number of skills are small. In the following, we obtain W[1]-hardness for $l + k$ even for a constant cost budget b. We will see later (Theorem 4 in Section 3) that our problem becomes tractable when both values, the maximum vertex degree Δ and the cost budget b, are small.

Theorem 2. Diam-Team Formation *parameterized by the combined parameter $l + k$ is* W[1]*-hard, even if the cost budget is two.*

Proof. We provide a parameterized reduction from Multicolored Clique parameterized by the clique size h. Let (G, ϕ, h) be an instance of Multicolored Clique; w.l.o.g. there are no edges $\{u, v\}$ with $\phi(u) = \phi(v)$. For $i \in \{1, \ldots, h\}$ define $V_i := \phi^{-1}(i)$, i.e., the set of all vertices of color i in G.

We create a graph G' from G as follows. First, subdivide all edges of G using new vertices. We use $V_{i,j}$ with $1 \leq i < j \leq k$ for the set of all newly introduced vertices that subdivide an edge between V_i and V_j. Now, we turn all subdividing vertices into a single large clique by adding edges. We define the skill set $T := \{1, 2, \ldots, h\}$. We assign each vertex v in a set V_i the skill set: $S(v) := \{i\}$, and each vertex v in a set $V_{i,j}$ an empty skill set: $S(v) := \emptyset$. We set the team size l to be $h + \binom{h}{2}$ and set the budget b (the diameter) to be two. This completes the construction which can clearly be done in FPT-time; indeed it can be computed in polynomial time. The correctness proof is omitted due to space. □

Observe that in the proofs of Theorem 1 and Theorem 2 we have made no use of the upper bound on the team size. Any team with all k skills and diameter at most b was proved to lead directly to a k-clique. Thus, the minimum team size is strongly inapproximable in the sense that even finding any feasible team respecting the cost budget is W[1]-hard with respect to k.

Corollary 2. *Unless all problems in* W[1] *are fixed-parameter tractable,* Min-Teamsize-Diam-Team Formation *is inapproximable even in* FPT*-time for the parameter number k of skills, even either on graphs with maximum degree three or with cost budget two.*

Finally, we show that the W[1]-hardness for the combined parameter $l + k$ still holds if we require that the graph induced by the team is only two-vertex-connected (resp. two-edge-connected) instead of requiring a small diameter, that is, requiring robustness instead of low communication costs.

Theorem 3. *Finding a team of size at most l, covering all k skills, such that the subgraph induced by the team is two-connected is* W[1]*-hard with respect to the combined parameter parameters $l + k$.*

Proof. We provide a parameterized reduction from MULTICOLORED CLIQUE parameterized by the clique size h to our problem parameterized by $l + k$ where l denotes the team size and k the number of skills. Given a MULTICOLORED CLIQUE instance (G, ϕ, h) construct a graph G' by again subdividing all edges. Just as in the proof of Theorem 2, use vertex sets $V_1, \ldots, V_h$ and let $V_{(i,j)}$ contain the subdividing vertices of (former) edges between V_i and V_j, $i < j$. Define the skill set $T := \{1, \ldots, h\} \cup \{(i, j) \mid 1 \leq i < j \leq h\}$. Assign the skills according to membership in sets V_i and $V_{(i,j)}$, i.e., a vertex v has skill $x \in T$ if and only if it is contained in set V_x. Finally, set the team size l to $h + \binom{h}{2}$. Note that the parameter value $l + k$ is upper-bounded by $O(h^2)$, which is sufficient. This completes the construction which can clearly be done in FPT-time; indeed it can also be computed in polynomial time. Equivalence of (G, ϕ, h) and the constructed instance is omitted due to space. □

3 Tractability results

In contrast to our hardness results, which all hold even for unit weights, we identify tractable cases with arbitrary positive integer weights. The first case models the situation where each potential team member is connected only to few others in the social network (that is, the maximum vertex degree Δ) and either the budget (that is, the diameter b) or the desired team size l are small.

Theorem 4. DIAM-TEAM FORMATION *can be solved in* $O^*(\Delta^l \cdot dcheck)$ *time and in* $O^*(\Delta^{\Delta^b} \cdot dcheck)$ *time where* Δ *is the maximum vertex degree of the input graph,* b *is the communication cost budget (the diameter),* l *is the team size, and checking whether the diameter of a subgraph is at most* b *takes dcheck time.*

Proof. Let $(G = (V, E), w, T, S, l, b)$ be our DIAM-TEAM FORMATION instance. Without loss of generality, we assume that the team $V' \subseteq V$ which we search for induces a connected subgraph. Given that each vertex has at most Δ neighbors, we build a search tree algorithm that branches into selecting one of the Δ neighbors of a potential team member (vertex) adding it to our partial solution (team). Since the team can have at most l members, the depth of our search tree is upper-bounded by l. In each node of the search tree we need to check whether the subgraph induced by the partial solution has diameter at most b (regarding the edge weight function w); this check runs in polynomial time. We omit the details due to space. □

Our second tractable case models situations where the team members are organized in a hierarchical tree structure.

Theorem 5. *If the input social network is a tree, then* DIAM-TEAM FORMATION *can be solved in* $O(2^k \cdot n \cdot b^2 \cdot B_k)$ *time, where* k *denotes the number of skills,* n *denotes the number of individuals in the network,* b *denotes the target diameter, and* B_k *denotes the* k*th Bell number.*

Proof. We describe a dynamic programming algorithm to solve Diam-Team Formation on trees. Let $I = (G = (V, E), w, T, S, l, b)$ be an instance of Diam-Team Formation with the input graph G being a tree. The basic idea is to store for each vertex $v \in V$ of the tree and each subset $T' \subseteq T$ of skills whether T' can be covered within the subtree rooted at v. To this end, we assume that $G = (V, E)$ is an arbitrarily rooted tree and denote the subtree rooted at $v \in V$ by subtree(v). We denote the set of children of each vertex $v \in V$ by children(v).

We define the dynamic programming table A as follows. For each subset $T' \subseteq T$ of skills, each vertex $v \in V$, each cost budget b', $b' \in \{0, 1, \ldots, b\}$, and each depth bound z, $z \in \{0, 1, \ldots, b\}$, the entry $A(T', v, b', z)$ stores the size of a smallest team $V' \subseteq V$ which fulfills the following requirements:

(a) V' covers T'.
(b) V' consists of vertex v and vertices only from subtree(v).
(c) The subgraph induced by V' is a tree with diameter at most b' and depth at most z. (That is, the largest weight of a shortest path between two arbitrary vertices of the tree $G[V']$ is at most b' and the largest weight of a shortest path between v and an arbitrary vertex $v \in V'$ is at most z.)

It is easy to see that there is a yes-instance if and only if $\min_{v \in V} A(T, v, b, b) \leq l$.

We fill the table entries following the tree from the leaves to the root. We initialize the entries concerning the set $V_L \subseteq V$ of leaves of the tree as follows.

$$\forall T' \subseteq T; v \in V_L; b' \in \{0, \ldots, b\}; z \in \{0, \ldots, b\}:$$
$$A(T', v, b', z) = \begin{cases} 1 & \text{if } T' \subseteq S_v \\ \infty & \text{otherwise} \end{cases}$$

Now, we consider some non-leaf v of the tree. The key question is which subtree rooted at some child of v contributes to the team. Clearly, in a *smallest* team,, each of such subtrees must cover at least some skill uniquely. Observe that there are at most B_k partitions of T' where B_k is the kth Bell number. That is, there are at most B_k possibilities of having at most k subtrees, each of which is rooted at some child of v and contributes to some smallest team covering T'. The idea is to consider for each part of the partition only the "cheapest" subtree covering it while fulfilling the diameter and depth requirements. Another crucial observation is that the diameter of the subtree rooted at $v \in V$ is the maximum of

(1) the largest diameter of all subtree(v'), $v' \in$ children(v), and
(2) the length of a longest path in subtree(v) containing v.

To calculate the value in (2), we need to know the length z_1 of a longest path from v to a leaf of the subtree(v') rooted at a child v' of v, and the length z_2 of a longest path from v to a leaf of the subtree(v'') rooted at a child v'' of v with $v'' \neq v'$.

Using these ideas, updating the entries bottom up works as follows. To handle the diameter costs that come from two different subtrees in subtree(v), we fix a partition of T' and the part of skills to be covered by the child v' of v such that subtree(v') has the largest depth.

$$\forall T' \subseteq T; v \in V; b', z \in \{0, \dots, b\} \colon A(T', v, b', z) = \min_{\substack{1 \leq i' \leq k' \leq k \\ T' = S_v \uplus T'_1 \uplus T'_2 \uplus \dots \uplus T'_{k'}}} \text{cheapestCover}(T'_1, \dots, T'_{k'}, i', v, b', z),$$

where "cheapestCover()" denotes the size of a smallest team covering T' in the following way.

(i) Each disjoint subset T'_i of skills is covered by the vertices of the subtree rooted at one child of v.
(ii) The team that covers $T'_{i'}$ induces a subtree with the largest depth.
(iii) The overall team (including v and covering T') has diameter at most b', and depth at most z.

It can be computed as follows.

$$\begin{aligned} &\text{cheapestCover}(T'_1, \dots, T'_{k'}, i', v, b', z) = \\ &1 + \min_{\substack{z_2 \leq z_1 \leq z \\ z_1 + z_2 \leq b'}} \Big\{ \min_{\substack{v' \in \text{children}(v) \\ z_1 \geq w(\{v, v'\})}} A(T'_{i'}, v', b', z_1 - w(\{v, v'\})) + \\ &\qquad \sum_{\substack{1 \leq i \leq k' \\ i \neq i'}} \min_{\substack{v'' \in \text{children}(v) \\ z_2 \geq w(\{v, v''\})}} A(T'_i, v'', b', z_2 - w(v, v'')) \Big\} \end{aligned}$$

For the correctness of our algorithm, if $\min_{v \in V} A(T, v, b, b) \leq l$, then there is indeed a set $V' \subseteq V$ with at most l vertices such that $\bigcup_{v \in V'} S(v) = T$ and $\textsc{Diam}(G[V']) \leq b$, which can be constructed by standard backtracking of our dynamic programming algorithm. However, it is not obvious that our algorithm considers all possible solutions, since it assumes that each part $T'_i, 1 \leq i \leq k'$ of the partition $T' = S_v \uplus T'_1 \uplus T'_2 \uplus \dots \uplus T'_{k'}$ is covered by a distinct cheapest subtree. To see that this is no real restriction, consider some fixed partition with $T' = S_v \uplus T'_1 \uplus T'_2 \uplus \dots \uplus T'_{i_1} \uplus T'_{i_2} \uplus \dots \uplus T'_{k'}$. Of course, it may happen that the cheapest subtrees for T'_{i_1} and T'_{i_2} are identical. In this case, the value of "cheapestCover()" might be higher than the size of the corresponding team and one might think that a smaller team may not be identified. However, as this also means that $T'_{i''} := T'_{i_1} \uplus T'_{i_2}$ can also be covered within the same subtree, the team will be found using some partition with $T' = S_v \uplus T'_1 \uplus T'_2 \uplus \dots \uplus T'_{i''} \uplus \dots \uplus T'_{k'}$, that is, replacing the two parts T'_{i_1} and T'_{i_2} with $T'_{i''}$. Summarizing, for every team there is always a partition of T' such that no two parts are covered within the same cheapest subtree.

Finally, the size of the two tables is upper-bounded by some function in $O(2^k \cdot n \cdot b^2)$. The initialization phase takes $O(2^k \cdot n \cdot b^2)$ time. The update phase takes $O(2^k \cdot n \cdot b^2 \cdot B_k)$ time. □

Finally we conjecture that the fixed-parameter tractability result from Theorem 5 can be extended to hold even for the combined parameter "number k

of skill" and "treewidth t". However, showing this certainly requires extensive technical details going beyond the scope of this work.

Acknowledgement. This work was started at the research retreat of the TU Berlin Algorithms and Computational Complexity group held in April 2014.

References

[1] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi. Online team formation in social networks. In *Proc. 21st International Conference on World Wide Web (WWW '12)*, pages 839–848. ACM, 2012.

[2] E. M. Arkin and R. Hassin. Minimum-diameter covering problems. *Networks*, 36(3):147–155, 2000.

[3] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer, 2015.

[4] S. Datta, A. Majumder, and K. Naidu. Capacitated team formation problem on social networks. In *Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 1005–1013. ACM, 2012.

[5] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Springer, 2013.

[6] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1(3):195–207, 1972.

[7] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.

[8] A. Gajewar and A. D. Sarma. Multi-skill collaborative teams based on densest subgraphs. In *Proc. 12th SIAM International Conference on Data Mining (SDM '12)*, pages 165–176. SIAM/Omnipress, 2012.

[9] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proc. 35th Annual ACM Symposium on Theory of Computing (STOC '03)*, pages 585–594. ACM, 2003.

[10] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pages 467–476. ACM, 2009.

[11] T. Lappas, K. Liu, and E. Terzi. A survey of algorithms and systems for expert location in social networks. In *Social Network Data Analytics*, pages 215–241. Springer, 2011.

[12] C. Li and M. Shan. Team formation for generalized tasks in expertise social networks. In *Proc. 2nd IEEE International Conference on Social Computing (SocialCom '10)*, pages 9–16. IEEE, 2010.

[13] J. Nederlof. Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems. In *Proc. 36th International Colloquium on Automata, Languages and Programming (ICALP '09)*, volume 5555 of *LNCS*, pages 713–725. Springer, 2009.

[14] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

[15] J. Park and S. Lee. Keyword search in relational databases. *Knowledge and Information Systems*, 26(2):175–193, 2010.