Interaction Mining and Skill-dependent Recommendations for Multi-objective Team Composition

Vienna University of Technology
Information Systems Institute
Distributed Systems Group

Christoph Dorn, Florian Skopik,
Daniel Schall, Schahram Dustdar
lastname@infosys.tuwien.ac.at

TUV-1841-2011-03 April 29, 2011

Web-based collaboration and virtual environments supported by various Web 2.0 concepts enable the application of numerous monitoring, mining and analysis tools to study human interactions and team formation processes. The composition of an effective team requires a balance between adequate skill fulfillment and sufficient team connectivity. The underlying interaction structure reflects social behavior and relations of individuals and determines to a large degree how well people can be expected to collaborate. In this paper we address an extended team formation problem that does not only require direct interactions to determine team connectivity but additionally uses implicit recommendations of collaboration partners to support even sparsely connected networks. We provide two heuristics based on Genetic Algorithms and Simulated Annealing for discovering efficient team configurations that yield the best trade-off between skill coverage and team connectivity. Our self-adjusting mechanism aims to discover the best combination of direct interactions and recommendations when deriving connectivity. We evaluate our approach based on multiple configurations of a simulated collaboration network that features close resemblance to real world expert networks. We demonstrate that our algorithm successfully identifies efficient team configurations even when removing up to 40% of experts from various social network configurations.

Keywords: team formation, social network, composition heuristic, recommendation trade-off model

Argentinierstr. 8/184-1
A-1040 Vienna, Austria
phone: +43 1 58801-18402
fax: +43 1 58801-18491
URL: http://www.infosys.tuwien.ac.at/

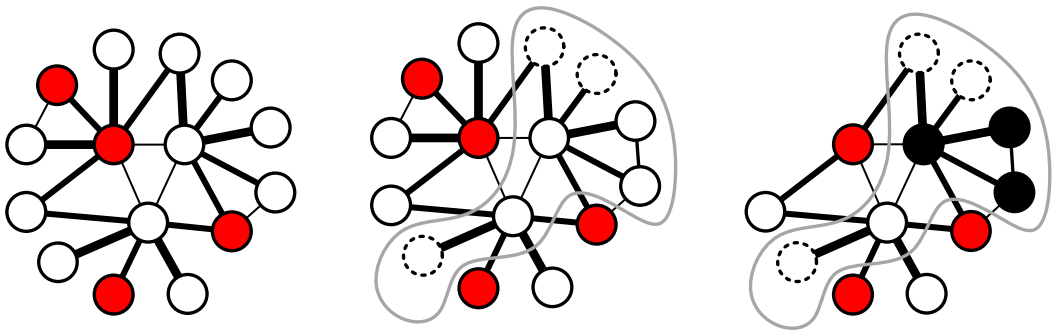
set {A, D, E, G}. The *connectivity* of this team is, however, rather low. The metric *interaction distance* is an indicator for the structural connectivity in collaboration networks. One of the novel contributions in this work is to use *recommendations* to identify suitable experts for replacing unavailable members. As shown in Figure 1b, a match yielding tighter relations (based on interaction weights) can be obtained when considering {B, E, F}. Recommendations are based on skills (e.g., {ml} associated with the profiles of, for example, C and F), thereby introducing new links between nodes. For simplicity, we augment the visualization of the interaction network by showing these links as dashed lines. Recommendations help establish new collaborations between members of a network. In Figure 1b, C recommends F to B, which subsequently results in forming a new link from B to F. The strength (weight) of the recommendation depends on the interaction-based distance of B and C as well as C and F.

While finding a suitable team in the given example is rather simple, the team formation problem becomes much more challenging in a large social network. With a linear increase in experts, the number of combinations to check grows exponentially. At the same time, the best connected experts — which are usually also the best skilled ones — display an increasing number of relations. Consequently, checking for recommendations becomes computationally expensive.

3.3. Approach Outline

Three major stages describe the progress toward team composition:

1. *Network Establishment.* First, we establish a group of experts and corresponding social network (Figure 2a) that provides the weighted interaction links among them, the respective skill profiles, and information about expert availability.
2. *Candidate Selection.* From this network, we select the set of team candidates which need to be available and provide at least one required skill (Figure 2b). However, the top ranked experts provide the highest expertise but usually come with low connectivity.
3. *Heuristic Optimization.* Finally, team composition aims to find a better connected team based on direct interactions and recommendations while maintaining high skill coverage (Figure 2c).



a) Expert Network b) Candidates c) Team Composition

Figure 2: Lines represent weighted edges based on previous interactions; unavailable experts are marked in red (a). Dashed circles are the top experts for each skill amongst the team candidates (bordered area). The final expert configuration (black, filled circles) provides the best combination of skills and distance (c). Colors online.

4. Multi-Objective Team Composition

4.1. Preliminaries

A team composition consists of experts, a weighted social network structure, and a set of skills $\mathcal{S}$. The social network is modeled as an undirected, weighted graph $\mathcal{G}_{SN}(\mathcal{U}, \mathcal{E})$ where the vertices are experts $u \in \mathcal{U}$, and the edges represent collaborations in joint activities. The edge weight w_e describes the distance between two experts.

A small edge weight represents frequent collaboration between two experts while a high edge weight describes rare collaboration. The proximity (Eq. 1) of an expert to itself is zero ($prox(u_i, u_i) = 0$). The proximity $prox(u_i, u_j)$ between two experts is defined by the shortest hop path (SHP) with minimum edge weights. We sum across all traversed edges and take the hop count (h) to account for the number of intermediary experts.

$$prox(u_i, u_j) = (h * (h-1) + 1) * \sum_k w_h \qquad \forall w_h \in max[SHP(u_i, u_j)] \tag{1}$$

An expert's skill profile $\mathcal{P}_i$ contains for each skill a corresponding expertise level $q_i(s)$. The expertise level is measured on a scale from 0 to 1, where 1 describes the maximum achievable expertise. Set of required skills for a team is denoted as $\mathcal{S}_R \subseteq \mathcal{S}$. A valid team configuration $\mathcal{T}(\mathcal{U}_\mathcal{T}, \mathcal{S}_R)$ consists of experts $\mathcal{U}_\mathcal{T} \subseteq \mathcal{U}$, such that for each $s_i \in \mathcal{S}_R$ there exists at least one expert $u_j \in \mathcal{U}_\mathcal{T}$ providing that skill ($q_j(s_i) > q(s)_{min}$ with $q(s)_{min} \geq 0$). We identify the expert u assigned to provide skill s within team $\mathcal{T}$ as $u_\mathcal{T}(s)$.

We establish the collaboration network and skill profile through observation of user interactions. Each interaction takes place in the scope of an activity which in turn is associated with exactly one skill. The sum of all involvements in activities produces the expert's skill profile. Specifically for each skill, there exists a mapping function $f_s(\mathcal{E}_u \mapsto q_u(s))$ that determines the expertise level from the set of an expert's edges $\mathcal{E}_u$.

Figure 3 (top left) displays an excerpt from the interaction log of users Alice, Bob, Carol, and Dave.[3] Alice and Bob have interacted together three times in a *p2p* related activity and once in a *dm* related activity. In total, Bob applied his *p2p* skills 6 times, his *dm* skills once, and his *ml* skill also only once. In this example, the mapping function f from edges to expertise level is a linear transformation. For each skill, we take the maximum skill occurrence (e.g., $p2p = 6$), thus deriving Alice's expertise level of $q_{Alice}(p2p) = 0.5$.

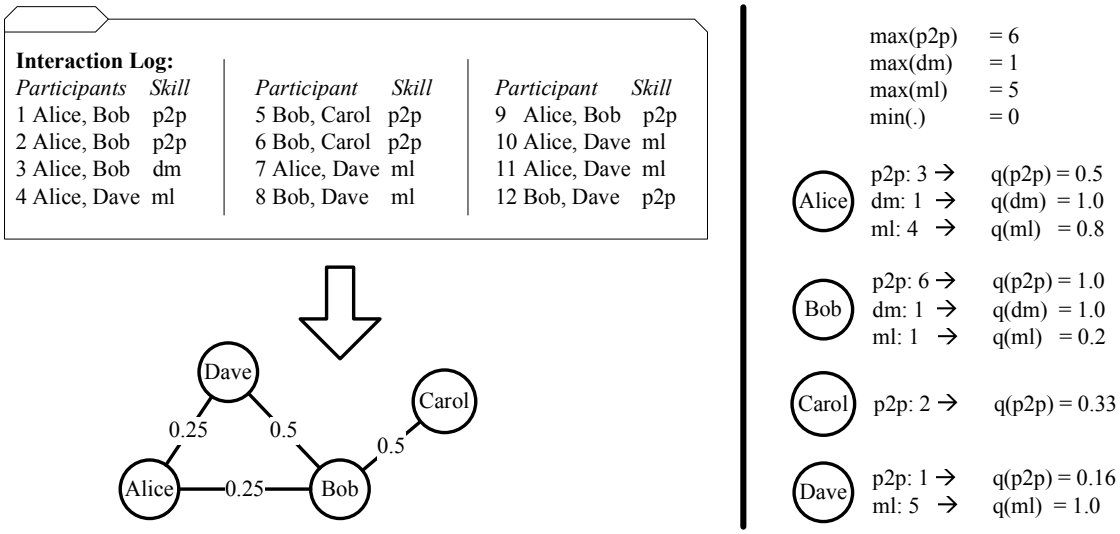


Figure 3: Deriving skill profiles and weighted social network from interaction logs.

In addition to the skill profile, each expert provides load information to indicate his ability to participate in a new team. Load is modeled as a boolean value $l_u = true|false$ where *true* denotes an available expert, while *false* denotes an overloaded expert.

4.2. Problem Definition

Given a set of experts $\mathcal{U}$, a social network $\mathcal{G}_{SN}$, and required skill set $\mathcal{S}_R$, find the team configuration $\mathcal{T}$ that provides the best match of user skills to required skills while reducing team distance as much as possible. Having a

[3]We omitted detailed log information such as timestamps for sake of clarity.

multi-objective function, we aim at maximizing the skill coverage function $C_{\mathcal{T}}$, while minimizing the team density function $\mathcal{D}_{\mathcal{T}}$. Additionally we need to satisfy following constraints: for each skill there exists a lower threshold $q(s)_{min}$ that defines the minimum expertise level within the team, and a set of team configuration constraints $\mathcal{G}$. Example constraints restrict the maximal number of skills a single expert may provide, or the minimum size of a team. How such constraints are formulated in detail is irrelevant at this point. Formally we can describe the team formation problem as follows:

$$\begin{aligned} \texttt{maximize} \quad & C(\mathcal{T}) \\ \texttt{minimize} \quad & \mathcal{D}(\mathcal{T}) \end{aligned}$$

such that:

$$\begin{aligned} u_{\mathcal{T}}(s) > q(s)_{min} \quad & \forall s \in \mathcal{S}_R \\ g(\mathcal{T}) = \texttt{true} \quad & \forall g \in \mathcal{G} \end{aligned}$$

Brute-force testing of every possible combination quickly becomes unfeasible. Testing m experts for $|\mathcal{S}_{\mathcal{R}}|$ skills has $O(m^{|\mathcal{S}_{\mathcal{R}}|})$ computational complexity (i.e., already for 10 experts and 10 skills, we would need to analyze 10 billion combinations). In the next subsection, we show that the described team formation problem is NP-complete and thus requires a heuristic to derive good solutions. We further analyze the computational complexity for calculating a team's quality when discussing the heuristics in Section 6.7.

4.3. NP Completeness

We demonstrate that the multi-objective team formation problem is NP complete by reducing it first to a single objective problem. We assume that for a particular skill s all experts with an expertise level above a certain threshold are equally well suited ($q(s) > req(s)$). Thus, the problem is reduced to finding the team structure with the strongest ties in-between all members.

This problem is related to determining a clique in a weighted graph. At this moment, however, it is unclear which and how many experts are part of the best team. Also it is unlikely that this team actually exhibits a fully connected subgraph. Consequently we cannot directly search for the best clique yet. We model skills and interaction structure as a weighted complete k-partite graph. A k-partite graph consists of k distinct sets of nodes so that there are no edges between any two nodes in the same set. Edges exist between nodes of different sets. In a *complete* k-partite graph, every node is connected to (i.e., is adjacent to) every other node outside its set. For our problem, the set of required skills corresponds to the k sets in the k-partite graph. The nodes within each set are 2-tuples of expert and skill $< u, s >$. An expert providing multiple skills above the threshold produces a tuple in each corresponding set. Ultimately, each set in the k-partite graph consists of tuples that contain the same skill s and all experts providing that skill. The edge weights between two tuples in different sets is given by the proximity measurement (see above) between the two corresponding experts.

Ultimately, a selection of exactly one tuple from each set constitutes a valid team where each tuple is connected to every other tuple in the team. The edges between the tuples determine the team's density. The tuples in the selection determine which experts is to provide what skill. When flattening the k-partite graph into a regular graph following properties hold: (i) any valid team will be a clique and (ii) the maximum clique size is k. The best team is then a matter of finding the minimum-weighted clique. This is trivially transformed into a maximum clique problem by inverting the edge weights. It is generally known that the maximum clique problem is NP-complete. We, thus, can infer that also the observed team formation problem is NP-complete.

4.4. Modeling the Objective Functions

To obtain a team's skill coverage $C(\mathcal{T})$ (Eq. 2), we take the experience level of each expert assigned to provide a particular skill s within team $\mathcal{T}$. Ignoring any team constraints, the best possible composition consists of the best skilled experts (i.e., an expert having maximum experience level $q(s) = 1$) for every skill $s \in \mathcal{S}_{\mathcal{R}}$. One user potentially covers multiple skills.

$$C(\mathcal{T}) = \frac{\sum_{\mathcal{S}_{\mathcal{R}}} u_{\mathcal{T}}(s)}{|\mathcal{S}_{\mathcal{R}}|} \qquad \textit{where } u \in \mathcal{T} \qquad \forall s \in \mathcal{S}_{\mathcal{R}} \tag{2}$$

We denote the best possible composition ($C(\mathcal{T}) = 1$) as $Top(\mathcal{S}_\mathcal{R})$. This top expert composition, however, usually does not yield tight relations between the experts. Hence, we try to reduce a team's distance $\mathcal{D}(\mathcal{T})$.

The distance function reflects our assumption that a-priori acquaintance of any two team members is crucial to successful collaboration. The direct team distance (Eq. 3) is thus defined as the sum of link weights between members plus a penalty distance for non-existing links. The number of non-existing links are determined by calculate the maximum possible number of links between members, and subtracting the number of existing intra team links ($|\mathcal{E}_t|$).

$$\mathcal{W}_e(\mathcal{T}) = \sum_{\mathcal{T}} w_e(u_i, u_j) + (|\mathcal{T}| * (|\mathcal{T}| - 1) * 0.5 - |\mathcal{E}_t|) * \beta * max(w_e) \qquad \forall u_i, u_j \in \mathcal{T} \tag{3}$$

The penalty parameter β determines the impact on distance when we drop an edge with maximum weight $max(w_e)$. For $\beta = 1$, we treat two experts yielding link of $max(w_e)$ as if they were not connected at all. If we set β too low, non-existing links are not penalized and we will not be able to find a better connected team than $Top(\mathcal{S}_\mathcal{R})$. If we set β too high, only fully connected teams will yield low distance and thus be considered. A sensible value derived from our experiments is $\beta = 4$ which we will use throughout this paper.

We inverse the distance objective function and instead of minimizing distance we aim to maximize the distance improvement (Eq. 4) compared to the Top Team.

$$\mathcal{D}(\mathcal{T}) = \frac{\mathcal{W}_e(\mathcal{T})}{W_e(Top(\mathcal{S}_\mathcal{R}))} \tag{4}$$

where $W_e(Top(\mathcal{S}_\mathcal{R}))$ is the team distance of the Top Team.

We aggregated the objective functions in order to determine the best team. The overall composition quality $\mathcal{Q}(\mathcal{T})$ (Eq. 5) is given as a linear combination of skill coverage and distance improvement. The trade-off parameter α encodes a preference towards best coverage ($\alpha \dashv 1$) or towards minimum distance ($\alpha \dashv 0$).[4] The optimum team configuration is then given by the composition with maximum quality.

$$\mathcal{Q}(\mathcal{T}) = \alpha * C(\mathcal{T}) + (1 - \alpha) * (1 - \mathcal{D}(\mathcal{T})) \qquad with\ \alpha = [0, 1] \tag{5}$$

For a required skill set $\mathcal{S}_\mathcal{R} = \{p2p, ml\}$ and $\alpha = 0.5$, the best workforce of experts in Figure 3 comprises users Alice and Bob. For $\mathcal{T}_{A,B} = \{Alice, Bob\}$, the skill coverage is $C(A, B) = 0.93$, the distance amounts to $\mathcal{W}_e(A, B) = 0.50$ (already applying the multi-skill per expert aware distance function as outlined in the next subsection). The overall quality yields $\mathcal{Q}(A, B) = 0.72$. There is an alternative set $\mathcal{T}_{B,D} = \{Bob, Dave\}$ that exhibits better coverage ($C(B, D) = 1$) but higher distance ($\mathcal{W}_e(B, D) = 0.5$), which overall yields only $\mathcal{Q}(B, D) = 0.5$.

4.5. Multiple Skill Provisioning

So far we have assumed that each expert in a composition provides exactly one skill. In some situations, it is beneficial to have an expert provide multiple skills. For a set of required skills $\mathcal{S}_\mathcal{R} = \{p2p, ml, dm\}$, Bob can provide *p2p* and *dm* skills, while Alice provides *ml*. We need to create a new composition view to correctly calculate recommendations and ultimately also distance. We simply split an expert profile that provides multiple skills and create a virtual profile for every skill. In Figure 4, Bob's profile is separated into Bob1 and Bob2, one for *p2p* and *dm* respectively. Bob's social network edge set is replicated for each virtual profile.

Distance needs to reflect the fact that an expert potentially provides multiple skills. So far teams with experts that provide multiple skills would yield lower distance. For the above listed example skill set $\mathcal{S}_\mathcal{R}$ we have three composition candidates: $\mathcal{T}_{B1,D,A} = \{Bob1, Dave, Alice\}$ happens to be the Top team, $\mathcal{T}_{B1,D,B2} = \{Bob1, Dave, Bob2\}$, and $\mathcal{T}_{B1,A,B2} = \{Bob1, Alice, Bob2\}$. Table 1 left part provides the coverage, interaction distance, and overall quality for the multi-skill unaware distance calculation. Apparently, a composition of Alice and Bob is preferable over a composition comprising Alice, Bob, and Dave; which in turn outperforms a combination of Bob and Dave. When we compare $\mathcal{T}_{B1,D,A}$ and $\mathcal{T}_{B1,D,B2}$ more closely, we notice that both provide the same skill coverage. $\mathcal{T}_{B1,D,B2}$, however, yields a better distance ratio as the distance measurement is unaware that Bob provides two skills.

[4] Throughout this paper we interpret $x \dashv y$ as variable x being close to or identical to value y.

	$C(\mathcal{T})$	$W_e(\mathcal{T})$	$Q(\mathcal{T})$	$W_e^v(\mathcal{T})$	$Q^v(\mathcal{T})$
$Top(\mathcal{S}_\mathcal{R}) = T_{B1,D,A}$	1.00	1.00	0.50	1.00	0.50
$T_{B1,D,B2}$	1.00	0.50	0.75	1.00	0.50
$T_{B1,A,B2}$	0.93	0.25	0.82	0.50	0.72

Table 1: Effect of multi-skill aware distance calculation on overall team composition quality.

The distance is more accurately represented when the distance calculation is based on the network including the virtual experts. This requires introducing an interaction link between the virtual expert profiles. In Figure 4 the additional edge is displayed as a red, double line. As we assume that provisioning of two skills by the same expert does not raise any expert internal conflicts we apply the minimum possible interaction edge weight $w_e(u_i, u_i) = 0$.

Applying this new distance calculation we determine new distance measurements. The superscript v is used only in this section to distinguish between calculation with and without virtual profiles. Later calculations are virtual profile aware by default. The right part of Table 1 provides the multi-skill aware distance and quality measurements. We still observe $\mathcal{T}_{B1,A,B2}$ as the best composition, however, $\mathcal{T}_{B1,D,B2}$ is now considered of equal quality as $\mathcal{T}_{B1,D,A}$ as its distance is the same. Note that the distance calculation does not include recommendations yet. We outline in the following section how to integrate interaction weights and recommendations.

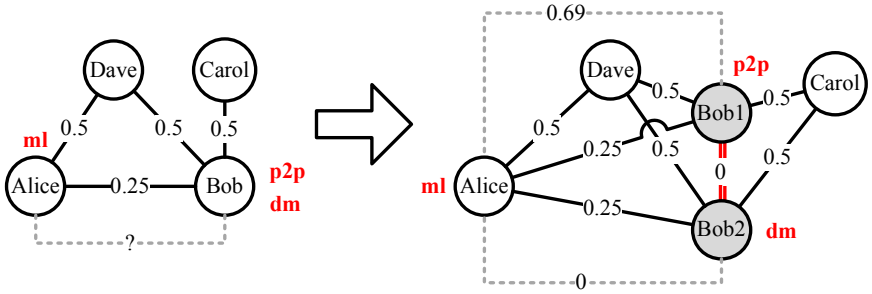


Figure 4: Splitting an expert profile for Bob who provides multiple skills to correctly calculate recommendations and distance. Edges labels display the interaction weights (w_e). The weight of the interaction edge (red/double line) connecting profiles Bob1 and Bob2 obtains the minimum possible interaction edge weight ($w_e = 0$).

5. Expert Recommendation

The goal of recommendation is to increase skill coverage and/or density by considering experts that are not directly connected to the existing team composition of experts $\mathcal{T}$. When Alice has frequently worked with Bob, and Bob in turn has frequently interacted with Carol, then Carol might be a suitable candidate in a composition with Alice when Bob is not available. Adding a non-connected expert to a composition, however, will greatly increase the team distance. In this section, we discuss how to use recommendations in the distance calculation to mitigate the effect of missing links.

Recommendations are skill dependent. For a given single recommendation in Figure 5 between Alice and Carol (the recommendees) via Bob (the recommender), we observe which skills the two (disconnected) experts provide within the team. Here, Alice contributes skill *dm* while Carol is contributing skill *p2p* — Bob's particular role in the composition is irrelevant in this recommendation. The recommendation reflects then how well Bob can evaluate the expertise of both recommendees.

The recommendation's strength depends on two factors. First, both the expertise level of recommending expert and the expertise level of the recommended experts have impact. Two experts can increasingly well evaluate each other the higher their corresponding expertise levels are. Thus, the recommendation strength will be equal for a low-skilled expert recommending a high-skilled expert or whether the recommendation occurs in the opposite direction.

In the former case, one cannot expect the low-skilled expert to give an accurate recommendation, while in the latter case the high-skilled expert will not fully endorse the low-skilled neighbor.

Second, the amount of past collaborations determines how well the two experts were in contact to obtain a correct view on their neighbors' skills. So far we did not consider which skills where involved in the common activities that formed the direct edge weight w_e. When calculating direct distance, we primarily want to answer who are able to work together successfully. For the purpose of giving recommendations, on the other hand, we must base our calculations on skill-centric edge weights w_s. It reflects how often two experts have collaborated within the scope of a particular skill s. When an (unavailable) expert u_z gives a recommendation (Eq. 6) between expert u_x (providing skill s_k) and expert u_y (providing skill s_l)s/he can only do so if s/he is qualified in the two involved skills (s_k, s_l). It reflects the confidence u_x has in the recommending expert u_z that s/he can correctly judge the expertise of u_y. Therefore the link strength of $w_s(u_z, u_y)$ is limited to interactions in scope of s_l) and vice versa the confidence of u_y in u_z to judge u_x correctly. When considering all skills, that (unavailable) expert u_z might have no personal experience in how well any of the other two experts provide their respective skill and therefore cannot give reliable recommendations. For example, the edge between Alice and Bob for skill *p2p* has $w_s = 3$. A single recommendation rec_1 between two experts (u_x, u_y) via a connecting expert (u_z) is subsequently defined as:

$$rec_1(u_x, s_k, u_z, u_y, s_l) = 0.5 * (q_x(s_k) * q_z(s_k) * w_s(u_x, u_z, s_k) + q_y(s_l) * q_z(s_l) * w_s(u_z, u_y, s_l)) \; \forall q_i \neq 0 \tag{6}$$

where s_k and s_l are the skill contributed by u_x and u_y, respectively. When at least one expertise level $q = 0$, then the recommendation chain is considered broken and $rec_1 = 0$. Note also that the recommendation is not reciprocal as the recommending expert would have to apply different edge weights, i.e., $rec_1(u_x, s_k, u_z, u_y, s_l) \neq rec_1(u_y, s_k, u_z, u_x, s_l)$.

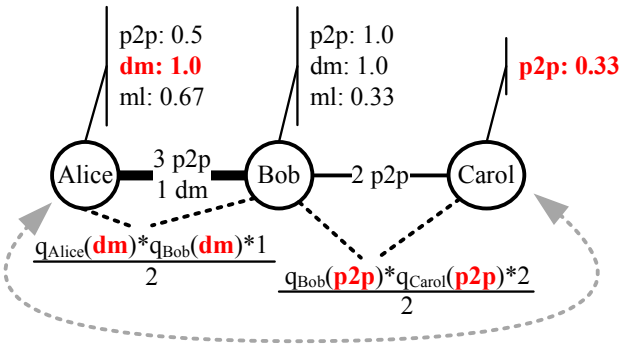


Figure 5: Single, skill dependent recommendation via expert Bob. Alice provides skill *dm*, while Carol provides skill *p2p*.

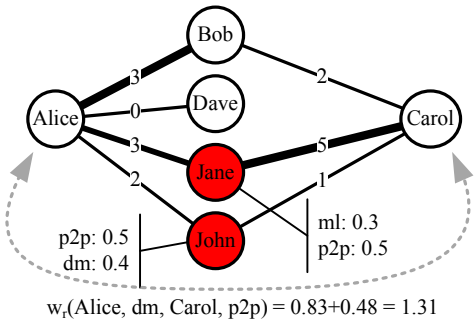


Figure 6: Aggregated skill dependent recommendations via experts Bob and John. Unavailable experts are red/shaded (colors online). Link labels provide the skill-centric edge weights specific to *dm*, respectively *p2p*.

A recommendation between two members is not limited to a single common neighbor. Instead we aggregate the recommendation of all joint neighbors (Eq. 7). This has the advantageous side effect of making use of the social network structure of overloaded experts. The links of all members—regardless of the member's availability—provide the input to recommendations. Figure 6 introduces two additional experts—John and Jane—and their respective expertise levels.[5] When calculating the aggregated recommendation weight between Alice and Carol, we consider only Bob, Jane, and John—the three joint neighbors. Although Jane has strong links to both Alice and Carol, she does not contribute to the recommendation as she does not share the *dm* skill with Alice. In case, however, Alice would contribute the *ml* skill, Jane's recommendation would weigh in heavily at $rec_1(Alice, ml, Jane, Carol, p2p) = 0.75$. Ultimately, only Bob and John produce the aggregated recommendation $w_r(u_x, s_k, u_y, s_l)$:

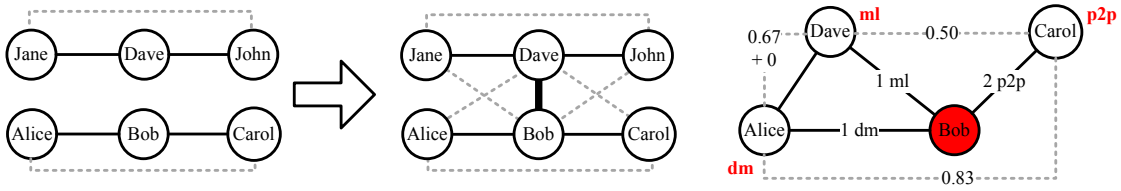
$$w_r(u_x, s_k, u_y, s_l) = \sum_i rec_1(u_x, s_k, u_i, u_y, s_l) \qquad \{\forall \; u_i \mid \exists \; e(u_x, u_i) \wedge \exists \; e(u_i, u_y)\} \tag{7}$$

[5] We have not recalculated the expertise values of Figure 3 for sake of simplicity.

This definition ensures that a recommendation is skill dependent. Only links from common neighbors determine the recommendation's strength. A particular expert might exhibit a large number of strong interaction links. If those links, however, do not end at common neighbors, they will not yield a strong recommendation.

A single strong recommendation between two experts is insufficient to decide whether they should form a composition with others. Instead, we need to calculate recommendations between the non-connected expert and all other experts in an existing composition. Recommendations exist also between directly connected experts. Take the segmented team in Figure 7a as an example. Initially recommendations exists only between Alice and Carol as well as between John and Jane. An additional single link that combines the two segments (Figure 7b) causes a substantial rise in recommendations. In order to avoid a distortion by a single new link, we also calculate the recommendations between directly connected experts (u_x, u_y) by aggregating $rec_1(u_x, s_k, u_y, u_y, s_l)$ and $rec_1(u_x, s_k, u_x, u_y, s_l)$. The overall team recommendation weight $\mathcal{W}_r(\mathcal{T})$ is then defined as:

$$\mathcal{W}_r(\mathcal{T}) = \sum w_r(u_i, s_k, u_j, s_l) \qquad \forall u_i, u_j \in \mathcal{T} \wedge i \neq j \tag{8}$$



a) Segmented Team b) Connected Team c) Regular and Direct Recommendations

Figure 7: Recommendation links: a) a segmented team has limited pairwise recommendations. b) Introducing a single connecting link increases recommendations substantially. c) Pairwise recommendations between Alice, Carol, and Dave when Bob is unavailable. Full lines are interaction edges (edge labels are skill-centric weights), dotted lines represent recommendations. The recommendation between Alice and Dave includes also the direct recommendation measurement which is 0, however, as Alice and Dave have only interacted applying skill *ml*.

5.1. Aggregating Interaction and Recommendation Weights

For testing various composition candidates, we need to be able to compare the effect of including a non-connected expert to including a weakly connected expert in terms of interaction edge weight. Our approach is to derive also recommendations for already connected experts (as outlined above), and evaluate the distance based on a linear combination of interactions and recommendations.

We can avoid comparing interaction distance and recommendations directly. The expert team formation problem is rather focused on how much interaction distance decreases, respectively the recommendation weight increases, compared to the initial expert configuration $Top(\mathcal{S}_\mathcal{R})$.

We replace the pure interaction-based sum as the sole distance measurement as introduced in Section 4. In particular, we apply a linear combination of interactions $\mathcal{W}_e(\mathcal{T})$ and recommendations $\mathcal{W}_r(\mathcal{T})$ according to the trade-off factor $\gamma = [0, 1]$ (Eq. 9). Note that we aim to minimize distance, respectively maximize recommendations. The overall quality function $\mathcal{Q}(\mathcal{T})$ assumes distance to decrease with better team configurations. Thus, we minimize $1/\mathcal{W}_r$ to obtain a correct overall distance function $\mathcal{D}(\mathcal{T})$.

$$\mathcal{D}(\mathcal{T}) = \gamma * \frac{\mathcal{W}_e(\mathcal{T})'}{\mathcal{W}_e(Top(\mathcal{S}_\mathcal{R}))} + (1 - \gamma) * \frac{\mathcal{W}_r(Top(\mathcal{S}_\mathcal{R}))}{\mathcal{W}_r(\mathcal{T})} \tag{9}$$

The interaction distance W_e penalizes missing team edges. By choosing γ close to or equal to 0, we no longer analyze those direct edges and instead rely on recommendations only.

5.2. Interaction and Recommendation Tradeoff

In highly connected networks, we risk having the recommendations overpower the direct interaction links. Especially social networks that lack a rich-club structure (see [9]) are prone to produce compositions of non-connected

experts. The rich-club phenomenon describes a type of network topology where the highest-degree nodes are well connected among themselves—they form a 'club'. This club would provide a team of top experts which result in low distance and also high recommendations. However, investigations of the rich-club phenomenon in scientific collaboration networks (e.g., [10]) have shown that such tight collaborative groups exist only within particular research domains but not beyond. Without a rich-club structure, the highest degree experts exhibit a large number of less skilled neighbors. These have tight links to multiple top-ranked experts (see Figure 8) and thus produce strong recommendations. In such a setting, these recommendations are likely to dominate over direct interaction links. Consequently, we need a careful balance between recommendations and direct interaction links (i.e., a suitable value for γ).

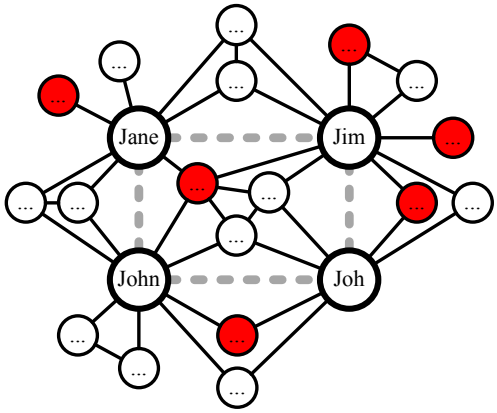
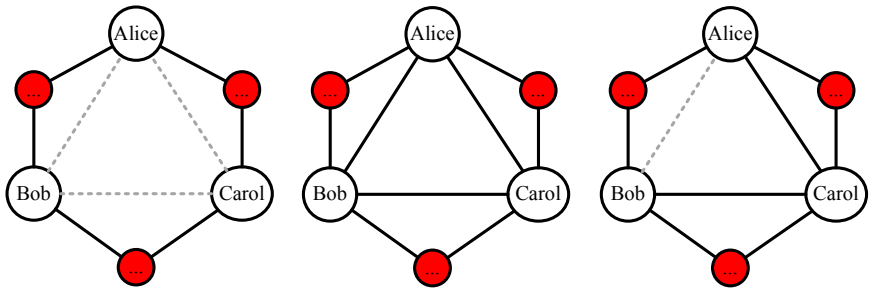


Figure 8: Example for a social network without rich-club structure: experts composition of Jane, John, Jim, and Joh comes with strong recommendation links (dashed lines) while they experts have not interacted before (solid lines).

We analyze the number of links between experts to determine the best value for γ. Figure 9 displays three simple social network examples. In each case, the experts Alice, Bob, and Carol are the available the experts from which we aim to find a optimal subset. The set of available experts for a given formation problem is defined as the candidate graph $\mathcal{G}_{Cand} \subseteq \mathcal{G}_{SN}$.



a) Unconnected b) Fully connected c) Semi-connected

Figure 9: Candidate expert graphs $\mathcal{G}_{Cand} = \{Alice, Bob, Carol\}$ for unconnected $\mathcal{G}_{Cand}$, fully connected $\mathcal{G}_{Cand}$, and semi-connected $\mathcal{G}_{Cand}$. All interaction edges (full lines) and recommendation edges (dashed lines) are assumed to yield weight $w = 1$.

Recommendations are most important when there are no or hardly any direct links between experts. In the extreme case (see Figure 9a) no direct links can be applied to derive the distance. Here we have to rely solely on recommendations and consequently set $\gamma = 0$. In case of the other extreme (Figure 9b), all candidate experts are connected to each other. Here, we do not need recommendations. Hence, we set $\gamma = 1$ and apply only interaction edge weights. Most of the time, however, the candidate graph's structure will remain within those extremes (e.g., Figure 9c).

We apply the concept of graph density (Eq. 10) to determine a suitable value for γ. The graph's density describes

the ratio of existing edges to maximum number of possible edges (i.e., a full graph):

$$dens_{\mathcal{G}} = \frac{2 * |\mathcal{E}|}{|\mathcal{V}| * |\mathcal{V} - 1|} \tag{10}$$

Particularly, we compare the candidate graph's density with the social network's density (Eq. 11). When the two values are identical, then $\gamma = 0.5$. In the interval $[dens(\mathcal{G}_{SN}); 2 * dens(\mathcal{G}_{SN})]$ γ increases towards 1, whereas it drops to zero when moving from $dens(\mathcal{G}_{SN})$ to 0:

$$\gamma = \begin{cases} 1 & if\ dens(\mathcal{G}_{Cand}) \geq min[2 * dens(\mathcal{G}_{SN}); 1] \\ dens(\mathcal{G}_{Cand}) * min[2 * dens(\mathcal{G}_{SN}); 1]^{-1} & if\ 0 < dens(\mathcal{G}_{Cand}) < min[2 * dens(\mathcal{G}_{SN}); 1] \\ 0 & otherwise \end{cases} \tag{11}$$

The example social network in Figure 9c has $dens(\mathcal{G}_{SN}) = 8/15$ and candidate graph density $dens(\mathcal{G}_{Cand}) = 2/3$. Hence with $\gamma = 2/3$, there is more focus on direct links rather than recommendations. In the evaluation we apply also fixed values of γ to analyze the effect of applying only direct interactions (i.e., $\gamma_1 = 1$), respectively only recommendations (i.e., $\gamma_0 = 0$), to determine distance.

6. Heuristics for Multi-Objective Team Composition

Our goal is to find a better connected team than the aggregation of the top experts for each skill but not necessarily the best possible solution. As demonstrated earlier, the multi-objective team formation problem is NP complete, thus we require a heuristic to find good solutions. Whether a given solutions is also optimal, however, cannot be determined. Simulated Annealing [27] (SA) and Genetic Algorithms [28] (GA) are two common heuristics suitable for the underlying problem type. We outline in the following subsection how to solve the team formation problem with these two heuristics and how they differ in finding a solution.

Simulated Annealing and Genetic Algorithms are similar as both test candidate teams, evaluate their quality (for SA denoted *energy*, for GA denoted $fitness$). Both algorithms continue from good teams to improve on the quality until the improvements become too small or the maximum number of iterations is reached. GA and SA differ in their techniques to determine subsequent team candidates, and which candidates to maintain for further exploration. The candidate team with the highest utility at the end is the team formation solution. Figures 10 and 11 outline a schematic procedure of the genetic algorithm, respectively simulated annealing.

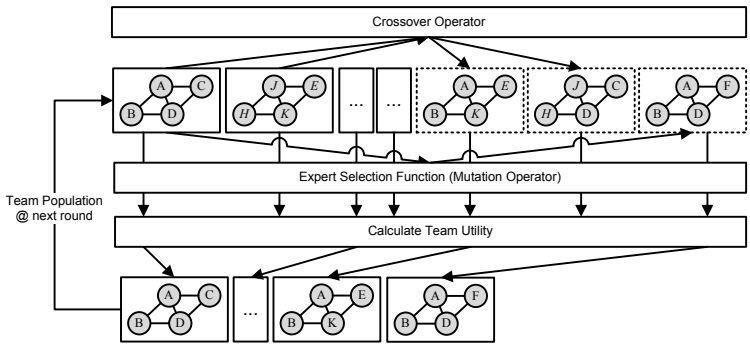


Figure 10: Genetic Algorithm overview: Crossover and mutation increase the current population of teams (new teams in dashed boxes). The team utility function restores the the population size through selection of the fittest teams.

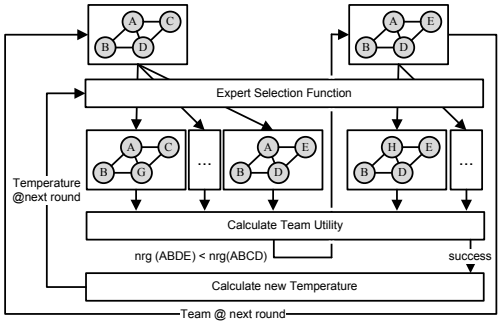


Figure 11: Simulated Annealing overview: new team configurations are derived from the current best team, which gets replaced by any improved team configuration. The number of better teams (success) determines the temperature in the next round.

6.1. Genetic Algorithms

A genetic algorithm treats the team formation problem as a population of individuals with different chromosomes. Individuals (i.e., various teams) of each generation mate and generate offspring: the new generation. Individuals with higher fitness as defined by their chromosomes (i.e., team configuration) are more likely to survive and multiply. The dominating individual of the last generation is deemed the best team configuration. An entity's chromosomes consist of a set of genes, where each gene represents one skill. The gene value is then the expert that provides that skill.

In each generation, GA utilizes two genetic operators to search for better team configurations:

Crossover generates new individuals by combining genes from two individuals. The two individuals are chosen randomly. Thus each individual has equal chance to be selected for mating, however, only fit offspring will survive into the next generation. During crossover, a randomly chosen gene position divides the chromosomes of two individuals (A and B) in two parts. One new individual obtains the first section of genes from A and the second section of genes from B. The second new individual obtains the inverse genes. In our case, two new team configurations arise each receiving part of their new experts from the other team. We check the next team configurations for constraint violations before these are added to the population pool.

Mutation takes the chromosomes of an individual and randomly changes one or multiple gene values. The individual itself remains unchanged, rather a copy including the mutation effect is added to the population pool. The amount of change is defined by the mutation rate. This corresponds to a random exchange of one expert for another. Here we ensure upfront, that only experts can be chosen that do not violate any team formation constraints. The expert selection function, as also applied by Simulated Annealing, is detailed further below in subsection 6.4.

Crossover and Mutation increase the population size through *mating* and *mutation* (Algorithm 1 lines 5+6). Next, GA evaluates the fitness (see subsection 6.3) of all individuals (line 7-9) and then selects the best ones to keep the population size constant (line 10). These best individuals are the new generation of team configurations (line 11). In the last generation, the team with highest fitness is selected as winner (line 14).

Algorithm 1 Genetic Algorithm GA($maxIt$, $popSize$, $mutRate$, $xoverCount$).

```
1:  function GA(maxIt, popSize, mutRate, xoverCount)
2:      Pop ← initRandomPopulation(popSize)
3:      generation ← 0
4:      while generation < maxIt do
5:          Pop ← Pop ∪ addSolutionsByCrossover(Pop, xoverCount)
6:          Pop ← Pop ∪ addSolutionsByMutation(Pop, mutRate)
7:          for T ∈ Pop do
8:              fitness_T ← calcFitness(T)
9:          end for
10:         Pop_new ← selectBestSolutions(Pop, popSize)
11:         Pop ← Pop_new
12:         generation ← generation + 1
13:     end while
14:     return T ← getBestSolution(Pop)
15: end function
```

6.2. Simulated Annealing

Simulated Annealing treats the team formation problem as a dynamic/hot system (representing a team configuration) that undergoes sequential transitions (changing the team) until it settles in a desirable stable/cold state (best team configuration).

In Algorithm 2 lines 2+3, SA takes an initial team configuration and derives the corresponding energy (see subsection 6.3). Simulated Annealing continues to evaluate team configurations (line 9+10) as long as the temperature

has not reached zero and there are more available iterations (line 6). A new team configuration is always accepted when it comes with lower energy (lines 11-16). Worse teams are accepted with probability p_{SA} defined as:

$$p_{SA} = e^{-\frac{\delta_{energy}}{temp}} \tag{12}$$

where δ_{energy} is the energy difference between the current and new team configuration and *temp* is the current annealing temperature. Accepting teams with higher energy is possible as long as the temperature remains high, and/or the energy difference is very small.

The freezing process describes the system's progress towards settling in a stable state. Freezing thus depends on the cooling rate and current iteration state. As long as the number of successful transitions is high (i.e., *success* close to *count*) the system remains in a search space region that still provides many solutions with lower energy. Thus while hot, the SA searches for configurations that are comparatively different to the current team. To obtain a new configuration, SA applies the same expert selection function (see further below in Section 6.4) as the GA mutation operator: one expert from the team is replaced by another available expert who doesn't violate the team formation constraints. The system cools down when compared solutions are worse than the currently team configuration. In this case, SA assumes the system to be near a global optimum. It then tests team configuration that are comparatively similar to the current configuration to get closer to the optimum. The function for determining the temperature for the next iteration (line 18) is defined as:

$$temp_n = r_{cooling}^{(limit_{accept} - \frac{success}{count}) * count} * temp \tag{13}$$

where *count*, $r_{cooling}$, and $limit_{accept}$ are configuration parameters.

Algorithm 2 Simulated Annealing SA($maxIt$, $startTemp$, $count$).

```
1:  function SA(maxIt, startTemp, count)
2:      T ← Top(S_R)
3:      nrg ← calcEnergy(T)
4:      temp ← startTemp
5:      iteration ← 0
6:      while temp > 0 and iteration < maxIt do
7:          success ← 0
8:          for 1 .. count do                ▷ Expert selection function provides a new team configuration
9:              newTeam ← calcNewExpert(T, temp)
10:             nrg_new ← calcEnergy(newTeam)
11:             δ_energy = nrg − nrg_new
12:             if doTransition(δ_energy, newTeam, temp) then
13:                 T ← newTeam
14:                 nrg ← nrg_new
15:                 success ← success + 1
16:             end if
17:         end for
18:         temp ← calcTemperature(temp, success)
19:         iteration ← iteration + 1
20:     end while
21:     return T
22: end function
```

6.3. Team Utility Function

GA and SA share the team formation objective function but differ in how they interpret the actual quality value. Using their respective terminology, GA aims at maximizing a team's fitness (Eq. 14), while SA aims at minimizing a

team's energy (Eq. 15). For GA, the fitness function for a team configuration $\mathcal{T}$ is defined as:

$$fit(\mathcal{T}) = \frac{1-\alpha}{\alpha * (1 - C(\mathcal{T})) + (1-\alpha) * \mathcal{D}(\mathcal{T})} \tag{14}$$

The corresponding SA energy function is the inverse of the GA fitness function:

$$nrg(\mathcal{T}) = \frac{\alpha * (1 - C(\mathcal{T})) + (1-\alpha) * \mathcal{D}(\mathcal{T})}{1-\alpha} \tag{15}$$

where dividing the aggregation of coverage and distance by $1-\alpha$ ensures that regardless of α the $Top(\mathcal{S}_{\mathcal{R}})$ team and other proportional tradeoffs will yield $nrg = 1$. $Top(\mathcal{S}_{\mathcal{R}})$ provides an upper boundary for the possible coverage. As no team configuration can yield higher coverage than the top experts, any team with higher distance than $\mathcal{D}'_{top}$ will yield $nrg > 1$ (respectively $fit < 1$) and thus can safely be ignored. Consequently, any better configuration must exhibit lower energy/higher fitness by reducing the distance $\mathcal{D}(\mathcal{T})$. Expert compositions that additionally come with lower coverage need to yield proportionally even lower distance. The improvement ratio is determined by the tradeoff factor α. Thus, any configuration that reduces coverage and distance to similar extent (as defined by α) also yields $nrg = fit = 1$.

6.4. Expert Selection Function

The expert selection function generates a new team configuration given a current configuration. The function needs to be able to (a) traverse the search space in short time and (b) find neighboring configuration with similar quality. The first requirement guarantees that both heuristics are able to reach all states in a timely manner, thus potentially identifying the optimum solution. The second requirement ensures convergence of Simulated Annealing as a random solution is more likely to be worse (rather than better) than the current solution.

Our expert selection function addresses both concerns. We randomly select a required skill s and exchange the current expert u_{old} with another expert u_{new} with probability p_{nh} (introduced below). Depending on the trade-off parameter γ we have to apply a different ranking criteria to identify suitable candidates. We apply the interaction proximity ($prox$) when $\gamma \dashv 1$ and direct interactions determine the team's overall distance. In contrast, we apply expert candidate degree ($degree_{cand}$) when $\gamma \dashv 0$ and recommendations determine the distance. We calculate the degree in the candidate network $\mathcal{G}_{cand}$ as the candidates provide a minimum level of expertise and thus can give and receive significant recommendations. The complete social network based degree, in contrast, is an unsuitable indicator as it would promote also experts that primarily link to non-qualified neighbors—which therefore are not in the candidate set and subsequently can give only weak recommendations if at all.

We rank each expert (Eq. 16) according to proximity ($prox$) and degree ($degree_{cand}$), with γ defining the weight distribution between the two metrics:

$$r(u_i) = \frac{\Big(\gamma * r_{prox}(u_i) + (1-\gamma) * (1 - r_{degree}(u_i))\Big) - min(u_i)}{max(u_i) - min(u_i)} \tag{16}$$

where r is the rank of expert u_i in the interval $[0, 1]$ such that the highest proximity, respectively largest degree results in $r_{prox|degree} = 1$ and the lowest value results in $r_{prox|degree} = 0$. The overall rank $r(u_i)$ is normalized to the interval $[0, 1]$.

The selection probability $p_{sd} = [0, 1]$ (Eq. 17) depends on the chosen heuristic. The probability is determined by the current temperature in SA $p_{nh} = temp/maxTemp$ and by the mutation rate in GA $p_{nh} = mutRate$.

$$p_{nh}(u_i) = \begin{cases} m^{-1} & if\ r(u_i) \leq s_{nh} \\ (1 - r(u_i)) * (1 - p_{nh})^{-1} & otherwise \end{cases} \tag{17}$$

where m is the number of candidate experts, $u_i \in \mathcal{G}_{Cand}$ are all candidates that provide a minimum expertise level of the selected skill ($q_i(s) > q(s)_{min}$). This prevents the selection of experts that are in close proximity, respectively yield high degree, but who do not provide the required skill. The expert selection function ensures that experts *similar* to the current solution are more likely to be selected when either the mutation rate is low or the temperature is low. Here *similarity* is determined by the prevailing distance measurement (i.e., direct interactions and/or recommendations).

6.5. Heuristic Differences

GA and SA differ in following three main aspects:

- Number of team configurations available in the next round: GA keeps a population of team configurations while SA keeps only the best team. Hence, GA can simultaneously develop multiple team configurations to avoid getting stuck in local optima.
- Team configuration change operators: GA applies crossover to generate radically different teams (where potentially more than one member is new) and mutation to introduce small, random changes. SA in contrast relies only on exchanging a single expert.
- Search progress: SA analyzes the number of improved versus worse team configurations to determine the system's temperature. It then utilizes that temperature to determine how far a new expert member can be from the current team. GA on the other hand remains unaware of the quality of teams within a population.

6.6. Selecting Suitable Heuristic Parameters

Selecting parameters for SA and GA is not straight forward as there are no general rule applicable to every problem domain. The three main GA parameters are population size, crossover, and mutation probability. Previous work suggests dynamic adaptation for crossover and mutation probabilities [29], whereas [30] applies clustering techniques to determine suitable values. The correlation of population size and cross over is investigated in [31]. These three exemplary works, however, address very different problem domains. In the case of simulated annealing, work on optimizing parameters is similarly problem specific: [32] addressing a graph partitioning problem, [33] focusing on the longest common subsequence problem, and [34] dealing with distributing workload across multiple processors.

As these efforts demonstrate, suitable parameters depend greatly on the the underlying problem type. We thus argue that more research effort is required to completely understand the impact of parameter values on SA and GA in the team formation domain. A rigorous analysis of suitable parameters, however, is definitely outside the scope of this paper.

Here, we discuss our settings based on our experience. We applied the simulated annealing algorithm as provided in the JUNG 1.7.6 framework[6], setting $tries = 200$, $count = 100$, $r_{cooling} = 0.99$, and $limit_{accept} = 0.97$. Example JUNG code provided initial parameter values which where then minimally adjusted through experiments. It took about 10 experiment iterations to fine tune those parameters, so we expect similarly low effort when applying our approach to other social network domains. For GA, we utilized the Java Genetic Algorithms Package (JGAP) [7]. We set the population size to 200, the same value as SA $tries$ to obtain comparable runtime behavior. Crossover applies to 35% of the population and the mutation rate is $8.\dot{3}\%$, both the default values used by the JGAP framework. We expect that values for SA and GA are suitable for social networks with similar link structure and skill distribution. Our experiments demonstrate that these parameters work for networks of various size.

As for tuning of the tradeoff parameter α and skill threshold values $q(s)_{min}$ we propose following mechanisms. Parameterless multi-objective algorithms such as NSGA-II [35] provide multiple pareto-optimal solutions to the team formation problem without setting α to any particular value. The user then inspects the results of a test run and derives from the most 'usable' solutions the corresponding α applicable in SA and GA. The skill threshold reflects the number of evaluated experts and thus has an impact on execution time. For high thresholds, only the very few top experts make it into the candidate set. Consequently, the heuristics potentially converge soon but might not find the best tradeoff. For very common skills and low threshold values, on the other hand, every expert in the social network becomes a candidate and the search space increases dramatically. In practise, an upfront analysis of the distribution of each skill is necessary to find suitable thresholds. We assume domain dependent analysis results, but expect the analysis to be of limited effort to conduct. We introduced a single, general limit for each skill as in our simulation all skills are equally common. That limit was 20% of the experts in the overall social network and $q(s)_{min} = 0.2$, whichever applied first.

[6] `http://jung.sourceforge.net/`
[7] `http://jgap.sourceforge.net/`

Team Coverage	$C_{\mathcal{T}}$	$O(\lvert\mathcal{S}_{\mathcal{R}}\rvert)$
Team Interactions	$\mathcal{W}_e(\mathcal{T})$	$O(\lvert\mathcal{S}_{\mathcal{R}}\rvert^2)$
Candidate Proximity	$prox(Cand)$	$O((\lvert E_{Cand}\rvert + \lvert Cand\rvert\ log\ \lvert Cand\rvert) * \lvert Cand\rvert)$
Single Recommendation	$w_r(u_x, s_k, u_y, s_l)$	$O(avgCommonNeighbors)$
Team Recommendations	$\mathcal{W}_r(\mathcal{T})$	$O(\lvert\mathcal{S}_{\mathcal{R}}\rvert^2 * avgCommonNeighbors)$
Expert Selection	$p_{nh}(Cand)$	$O(\lvert Cand\rvert)$
Team Quality	$Q(C_{\mathcal{T}}, \mathcal{W}_e(\mathcal{T}), \mathcal{W}_r(\mathcal{T}))$	$O(\lvert\mathcal{S}_{\mathcal{R}}\rvert^2 * avgCommonNeighbors)$
Genetic Algorithms	GA	$O(maxIt * popSize * Q(\mathcal{T}))$
Simulated Annealing	SA	$O(maxIt * maxRounds * Q(\mathcal{T}))$

Table 2: Runtime complexity of the heuristics and functions applied in determining a team's quality.

6.7. Scalability Aspects

We listed the computational complexity for the main functions in Table 2. The number of skills $|\mathcal{S}_{\mathcal{R}}|$ seem to have a significant effect on team interaction distance and recommendations due to the quadratic runtime complexity. However, we expect no negative performance impact as the number of independent skills within a team is usually low. We propose clustering of commonly collocated skills when a larger skill set is required.

The set of candidate experts has a similarly low impact on performance. Here, we propose precomputing the interaction distance between all candidates. Finding a single shortest path is in $O(|E_{Cand}| + |Cand|\ log\ |Cand|)$ [36]. Both heuristics subsequently apply those distance values during every team transformation (i.e., expert selection) and team quality evaluation. The precomputation is also feasible even for larger candidate sets when distance values remain stable over a larger period of time, e.g., a month. Optimization strategies such as constraining the distance calculation (e.g., assigning a default distance beyond 3 hops) brings an additional performance boost.

A-priori calculation of recommendations, however, is too computationally expensive. There are potentially $|\mathcal{S}_{\mathcal{R}}| * (|\mathcal{S}_{\mathcal{R}}| - 1)$ different recommendations a single expert u_z can give between two experts u_x and u_y. There exist $O(|Cand|^3)$ of those triples in the form $< u_x, u_y, u_z >$ set when the underlying the candidate set is a fully connected graph. While this assumption is unrealistic in most cases, it still prohibits the precalculation of recommendations. Thus the heuristics calculate the recommendations on the fly as needed. Several performance measures are applicable: for small team transformation, such as when only a single member of the team is exchanged, only the recommendations involving the leaving member(s) and the new member(s) need to be determined. Further more, we propose the caching of recommendations for even quicker access.

There are additional measures that go beyond tuning the computation of distance and recommendations. Incremental search, for example, starts with a small candidate set that quickly returns results. If the user is not satisfied with the proposed team configurations, s/he can retry with a larger candidate set. Alternatively, the search in a larger candidate set can be carried out in parallel to the initial request. Starting with a small core team that needs to be as well connected as possible is another strategy. Once a core team is established, experts for less important skills that need not yield so good connectivity join the team. The detailed discussion on the implications of these strategies on team quality and algorithm performance is beyond the scope of this paper.

7. Evaluation

In this section, we focus on three aspects of the team formation heuristics: (i) we observe the improvement of team quality to demonstrate the heuristics' ability to find better team configurations than the initial Top Team ($Top(\mathcal{S}_{\mathcal{R}})$); (ii) we analyze the impact of the dynamic trade-off factor γ as compared to pure interaction and pure recommendation driven composition to motivate the need for dynamic γ; and finally (iii) we compare the performance of the genetic algorithm and simulated annealing.

We evaluate the performance of our workforce composition algorithm based on an synthetic data. Thus, we first outline the generation of the social network structure, distribution of skills, and the strategy for selecting overloaded experts. Second, we present the results for various network and skill configurations and display an example of initial and final team configuration. The analysis also includes a comparison to a real world data set which demonstrates the viability of our simulation model.

The experiment results demonstrate the effectiveness of dynamically balancing interaction-based and recommendation-based distance calculation. The dynamic adjustment of trade-off factor γ achieves best results in social networks lacking a rich-club phenomenon and which rapidly become very sparse when experts are overloaded. Both heuristics are able to identify team configurations with significantly improved team distance and while maintaining high skill coverage. The Genetic Algorithm, however, consistently outperforms Simulated Annealing.

7.1. Experiment Model

Most social networks yield a power-law degree distribution [37, 38]. In such networks most nodes exhibit only a few neighbors while a few nodes are extremely well connected. For simulation purposes such a degree structure emerges from preferential attachment of edges [39]. For our experiments preferential attachment generates an undirected graph yet without edge weights. This mechanisms has new nodes (e.g., node N in Figure 12a) link preferably to well connected nodes (eventually to node A in Figure 12b).

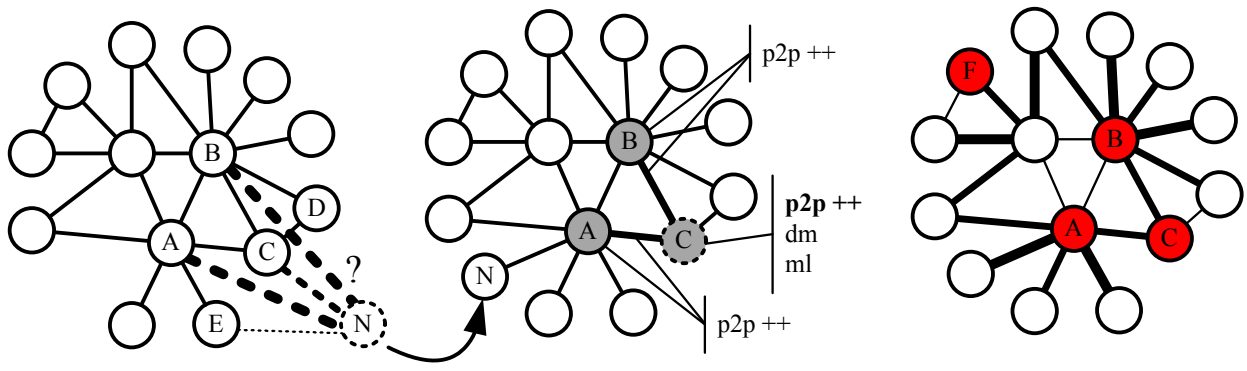
The plain network graph serves as foundation for defining edge weights and distributing skills. We mimic collaborative behavior by selecting a node and a subset of its neighbors. For this node we now determine a skill depending on the set of distinct, already acquired skill and the maximum allowed amount. The skill counter is then increased on all edges connecting the chosen node and neighborhood subset (e.g., skill *p2p* for nodes A, B, and C in Figure 12b). At the same time, also the skill counter of the involved nodes increases by 1. We repeat this step $degree * nodeCount * 10$ times.

The underlying power-law distribution causes the node and skill selection process to produce a rich-club free edge-weight structure. Whenever we select a low-degree node, the randomly selected neighbor subset is more likely to be of higher degree than equal or lower degree (Figure 12b). Analogous, a selected high-degree node will exhibit more low-degree neighbors. Consequently, edges in-between high degree nodes and in-between low-degree nodes have significantly lower weights than edges connecting high-degree with low-degree nodes (represented by line thickness in Figure 12c). The second side effect is that high-degree nodes have higher expertise scores than low-degree nodes as they are more likely to participate in collaborations (i.e., neighborhood subsets).

Finally, we introduce the strategy for selecting overloaded experts. We apply degree-based preferential selection of experts rather than a random set. Well-connected experts are more likely to become part of a composition and thus are more likely to be overloaded, respectively unavailable. Hence, high-degree nodes have a higher probability for being marked as unavailable than low-degree nodes (see also Figure 12c).

7.2. Experiment Configuration

We simulated two expert networks with 200 (Experiment 1.1, 1.2), respectively 1000 experts (Experiment 2). In the experiments, the overall skill set $\mathcal{S}$ comprises 30 and 100 skills, where each expert can acquire a maximum of 8



a) Preferential attachment b) Skill distribution c) Availability Selection

Figure 12: Simulation process: (a) creating a power-law graph via preferential attachment (dashed line thickness represents link probability), (b) selecting random node $\{C\}$ and neighborhood subset $\{A, B\}$ for skill distribution, and (c) marking nodes $\{A, B, C, F\}$ as unavailable based on preferential selection.

and 15 skills. The skill distribution mechanism selects for each expert a subset of 5 and 15 neighbors. The required skill set $\mathcal{S}_{\mathcal{R}}$ consists of 8, respectively 10 randomly selected skills. In addition we apply composition constraints during the team formation process to replicate real world settings: in experiment 1.1 each expert could provide only a single skill, while for experiment 1.2 an acceptable team solution is required to have a minimum team size of 6. In experiment 2, teams must consist of a minimum of 7 experts.

We tested for the effect of expert overload by marking 0%, 10%, 20%, 30%, and 40% as unavailable based on degree-based preferential selection. The benefit of trading off interactions-based and recommendation-based distance becomes apparent when comparing fixed values of γ (γ_0, γ_1) with dynamically calculating γ. Results for each of these configurations derive from the average of 10 iterations with different initialization of the underlying network structure and skill distribution. During all experiments we set the trade-off parameter $\alpha = 0.1$. This value reflects the fact that coverage can drop sharply when low-quality members join the team, while distance can never reach zero. Thus, we put more weight on reducing distance than aiming for members of high expertise. Experts had to yield $q(s)_{min} \geq 0.2$ to be considered for a particular skill s to ensure a minimum level of coverage. Experts that did not provide a single required skills were temporarily removed from the social network to improve processing speed. The candidate network $\mathcal{G}_{Cand}$ for the two experiments exhibits on average around 120 and 470 experts, respectively.

7.3. Effect of the Dynamic Interaction-Recommendation Tradeoff

Figure 13 depicts the impact of fixed and dynamic γ when applying the Genetic Algorithm for increasing levels of unavailable experts. Figure 14 provides the same visualizations for the Simulated Annealing heuristic. In each subfigure the green boxes provide the values for dynamically calculated γ. Blue circles depict the results for $\gamma = 1$, respectively the red triangles for $\gamma = 0$. Where applicable, the turquoise crosses on dotted lines provide the values of the starting Top team ($Top(\mathcal{S}_{\mathcal{R}})$). We compare both heuristics in Figure 17. Higher values are preferred for coverage C and recommendation distance W_r. In contrast, lower values for interaction distance W_e and overall team distance improvement $\mathcal{D}$ signify better performance. The error bars illustrate the standard deviation of 10 experiment runs. Note that we shifted some results +.5% along the x-axis and occasionally printed only the upper or the lower error bar for sake of visibility.

7.3.1. Impact on the Genetic Algorithm

Inherent to our multi-objective team quality function, any team configuration that comes with merely slightly reduced coverage, needs to yield only equally small improvements on distance (and vice versa). Consequently, with fixed $gamma_0$ focusing only on interactions and thus ignoring direct interactions results in consistently higher coverage, higher interaction distance[8], and higher recommendations (Fig. 13). We obtain the inverse result with fixed γ_1. This nicely demonstrates that the team formation problem consists indeed of conflicting optimization functions: maximum coverage and minimum direct interaction distance.

As the trade-off suggests, the dynamically calculated γ has direct interactions W_e and recommendations W_r always between the fixed γ_0 and γ_1 settings. Values for dynamic γ tend to be closer to the results for γ_1 rather than γ_0 as the network never experiences extreme fragmentation. Thus the fact that $W_r(\gamma)$ never reaches the same level as pure recommendation (γ_0) is negligible as recommendations serve merely as substitute for direct interactions and thus are not the main objective. However, dynamic γ does converge towards γ_0 as more experts become unavailable.

Most importantly, with dynamic γ, the genetic algorithm simultaneously achieves higher recommendations (W_r) and lower interaction distance W_e than the top team. In the case of coverage, dynamic γ even outperforms both fixed configurations (γ_0, γ_1): occasionally in experiment 1.1 (Fig. 13a) and 1.2, and consistently in experiment 2 (Fig. 13d).

Comparing coverage and direct distance with γ_1 highlights the effect of increasingly unavailable experts. In experiment 1.1, coverage remain stable, but distance increases (Fig. 13b), while in experiment 2 low distance (Fig. 13e) can only be maintained through reduction in coverage. The dynamic threshold mitigates expert unavailability by successfully exploiting recommendations. This strategy thus maintains high coverage while still improving on distance and recommendation when compared to the Top Team. Experiment 1.2 provides similar results as Experiment 1.1, thus respective figures are excluded here but reported in Annex A (20a-c).

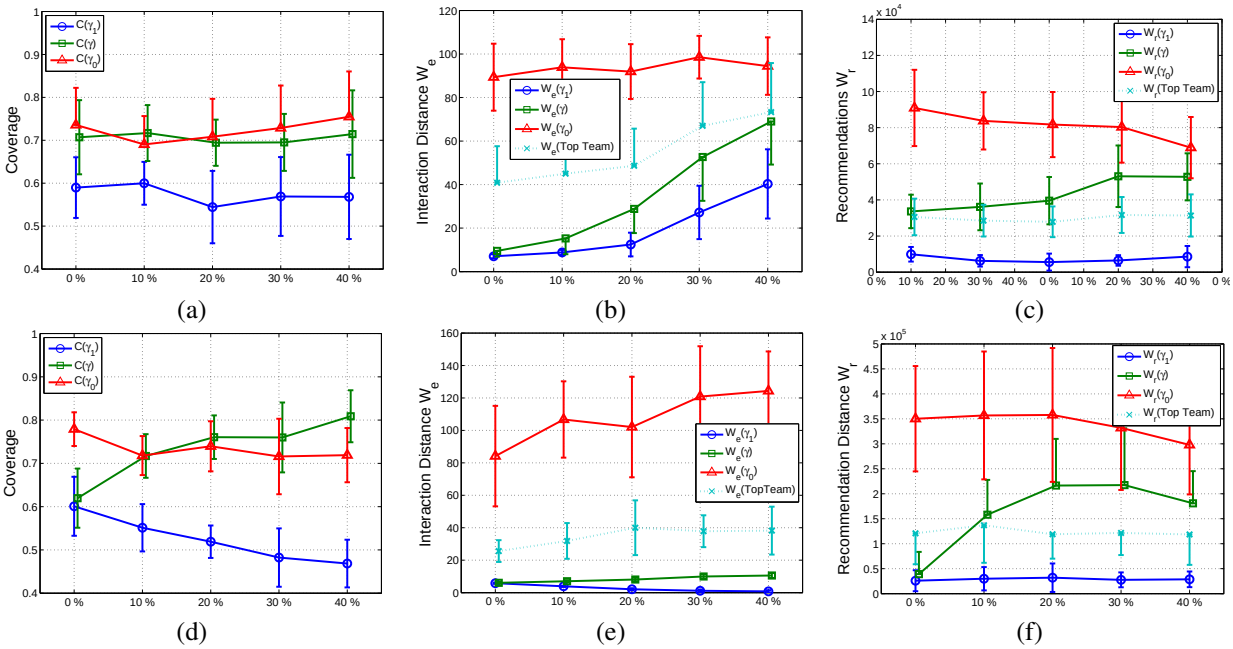


Figure 13: Genetic Algorithm results for: Coverage C (a,d), Interaction Distance W_e (b,e), and Recommendation Distance W_r (c,f) for experiment 1.1 (a-c), and experiment 2 (d-f).

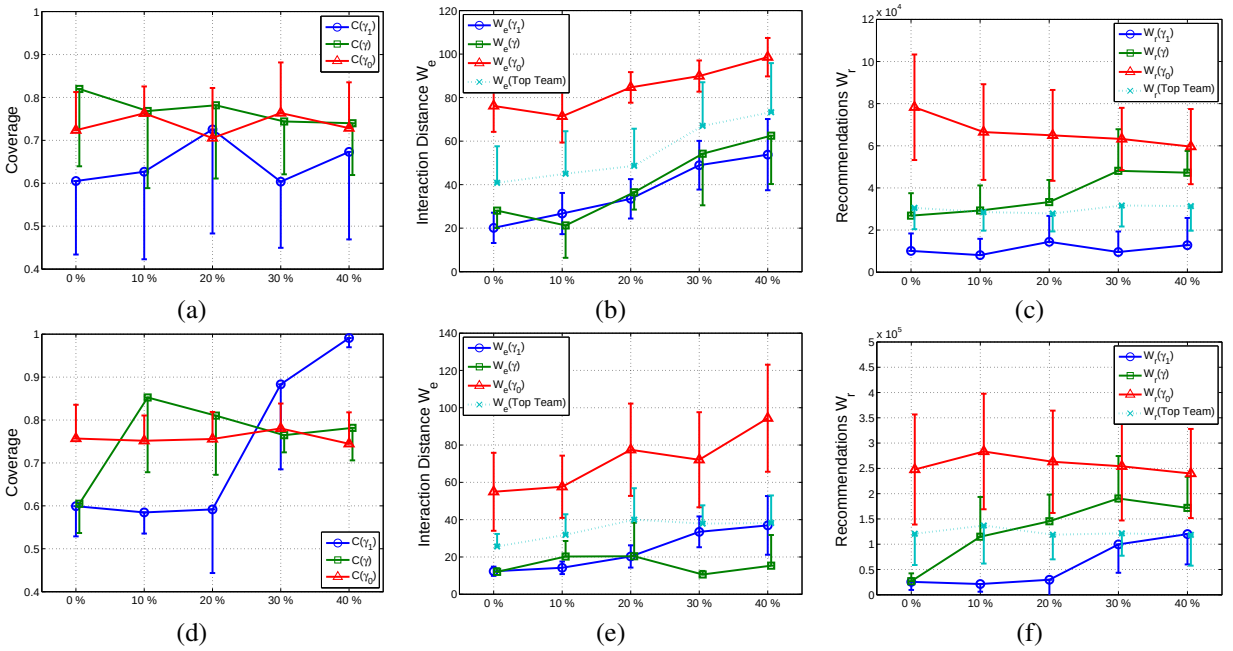


Figure 14: Simulated Annealing results for: Coverage C (a,d), Interaction Distance W_e (b,e), and Recommendation Distance W_r (c,f) for experiment 1.1 (a-c) and experiment 2 (d-f).

7.3.2. Impact on Simulated Annealing

Overall, SA has similar result characteristics (Fig. 14) as the GA. For experiment 1.1 and 1.2, dynamic γ has direct interaction distance (W_e), and recommendations (W_r) between both fixed strategies (γ_1, γ_0). When comparing

[8]Remember that we aim to reduce interaction distance.

coverage, we notice large deviations for γ_1. In experiment 2, SA with γ_1 cannot find significantly better teams than Top Team for more then 30% unavailable experts.

Notice the proximity of W_e and W_r to the Top Team values (dotted line in Fig. 14e+f) combined with high coverage. SA with dynamic γ, however, provides similar improvements over the Top Team across all expert availability levels and all experiments. Again, experiment 1.2 provides similar results as Experiment 1.1, thus respective figures are excluded here but reported in Annex A (20e-f). Overall, the data demonstrates that both heuristics provide better solutions than either composition strategy by itself when applying the dynamic combination of interactions and recommendations.

7.4. Example Team Formation

An excerpt from experiment 1.2 demonstrates the improvement in team distance as achieved by the genetic algorithm (Fig. 15a) and simulated annealing (Fig. 15b). Table 3 provides the corresponding skill, team, and metric information. The initial team (Top Team) consists of seven members ($[U7, U21, U23, U109, U83, U76, U143]$ from a network of 200 experts) provide the eight required skills ($S0 \rightarrow S7$). The Top Team is weakly connected, with $U76$ and $U83$ having no previous interactions with any of the remaining members. The normalized team density[9] of 0.321 is rather low.

Both heuristic exploit the composition restriction (minimum 6 experts) to the full extent and reduce the initial number of involved experts. The genetic algorithm preserves three members from the Top Team ($[U21, U23, U143]$) but assigns $U21$ a different skill. All three additional members come with high skills ($q(s) \geq 0.7$) and considerably reduce the team distance. The normalized team density is comparatively high with 0.893 due to $U23$ and $U22$ linking to every other member, and only $U143$ featuring less than three intra-team relations. Simulated annealing improves similarly on distance, but not as successfully. Two members from the Top Team $U21$ and $U143$ join 4 new members all having skills higher than 0.45. The lower normalized team density of 0.607 is largely due to two experts ($U143$ and $U161$) having only a single link, and only a single expert ($U22$) yielding previous interactions with all other members.

In direct comparison, the genetic algorithm manages to maintain higher coverage while achieving lower distance than simulated annealing. Inverting the energy metric we obtain the team quality: GA more than doubles the team quality, whereas GA provides only a 58% increase (compared to the Top Team that has quality and energy always equal to 1). Teams produced by GA and SA highlight a general property of the formation process: at least some members of the Top Team will always be part of the best found team. In this example, GA and SA produced rather similar final teams (overlap of 5 members). Later in Section 7.5.1 we test for correlation of GA and SA to determine if this is usually the case.

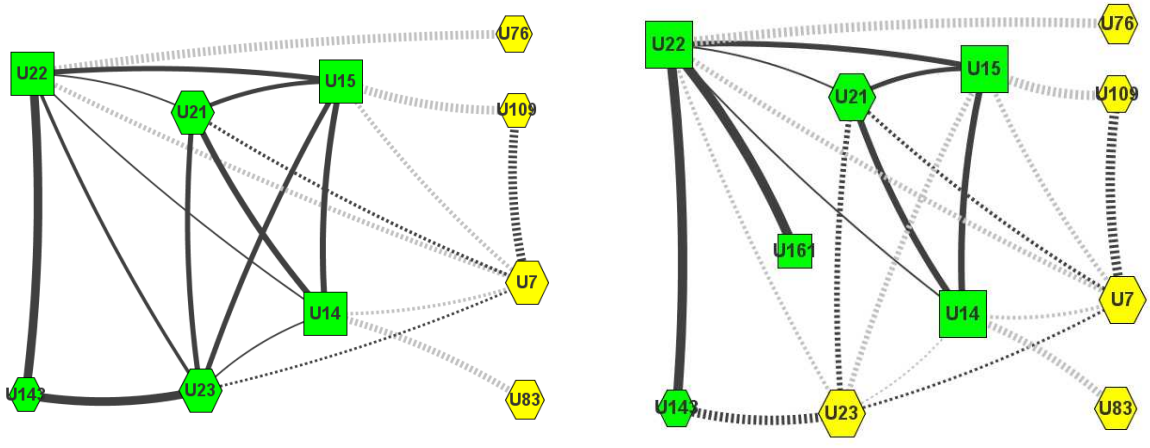
7.5. Comparing GA and SA for dynamic γ

In our experiments, the GA consistently outperform SA. Figure 16 compares the average fitness values for the best final teams. For full expert availability the genetic algorithm delivers a 2.3x to 3.5x fitness improvement compared to the initial Top Team. SA in contrast increases fitness only by a factor 1.1x to 1.9x. The difference becomes smaller with declining γ. For 40% unavailable experts, GA improves fitness by 38% to 92%, whereas SA ranges between 37% and 68% respectively. Looking in more detail at coverage and distance, we note that the GA performs better at finding densely connected teams while maintaining consistently high coverage.

For both heuristics, coverage C remains comparatively high regardless of the amount of available expert, even though we have put more focus on achieving lower distance than high coverage ($\alpha = 0.1$). Across all experiments and both heuristics, average coverage resides above 0.6 even though we have set the lower limit for expertise to $q(s)_{min} = 0.2$. SA produces teams with higher coverage than GA but simultaneously delivers smaller distance improvements.

When applying genetic algorithms, distance $\mathcal{D}$ decreases linearly and converges to roughly the same level as simulated annealing as an increasing number of experts become unavailable. GA thus clearly outperforms SA below 30% unavailable experts. Given the large standard deviation in distance improvements and also coverage, the results are closer together for 30% and 40% missing experts. While SA failed to produce a better final team than $Top(\mathcal{S}_{\mathcal{R}})$ 25x across 150 experiment rounds (50 rounds each in Ex1.1, 1.2, and 2), GA only failed to do so 6x. This also explains simulated annealing's spike in distance improvement for 10% and 20% in Experiment 2 when 3 out of 10 team configurations could not be improved.

[9] I.e., the number of actual links divided by the maximum possible number of links in a graph



(a) (b)

Figure 15: Team formation result with Genetic Algorithms (a) and Simulated Annealing (b) from Experiment 1.2, 40% unavailable experts, using dynamic γ for trade-off between W_e and W_r. Full edges between best team members (green squares, and hexagons), dashed edges between Top Team members (yellow hexagons). Thicker edges represent tighter links.

Skill	Top Team	T(GA)	q(s)	T(SA)	q(s)
S0	U7	U21	0.71	U161	0.53
S1	U21	U15	0.86	U15	0.86
S2	U23	U23	1.00	U15	0.67
S3	U23	U23	1.00	U14	0.47
S4	U109	U14	0.79	U21	0.79
S5	U83	U23	0.87	U14	0.91
S6	U76	U22	0.70	U22	0.70
S7	U143	U143	1.00	U143	1.00
Coverage		0.87		0.74	
Distance		0.45		0.60	
Energy		0.46		0.63	
Quality		2.17		1.58	

Table 3: Top Team and best teams generated by Genetic Algorithm and Simulated Annealing for an example simulated team formation from experiment 1.2.

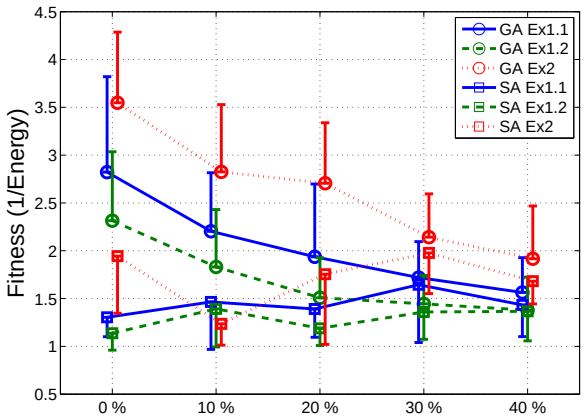
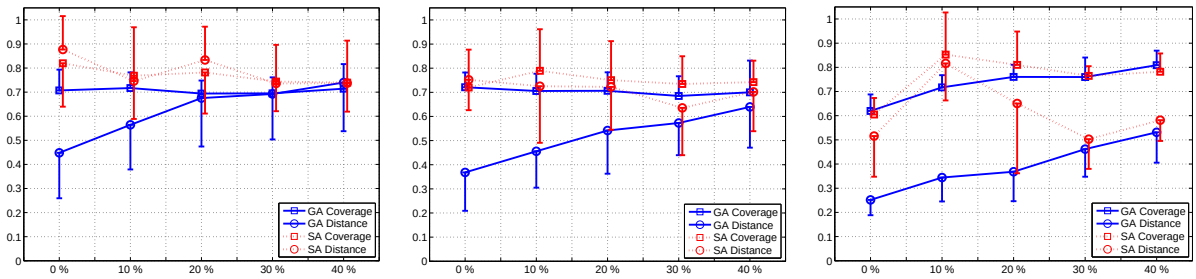


Figure 16: Average fitness of the best final teams produced by Simulated Annealing (boxes) and Genetic Algorithm (circles).

7.5.1. Heuristic Correlation

As indicated by the error bars in Figure 17, both heuristics yield a significant amount of deviation in distance improvements and also coverage. We therefore test these two metrics for correlation to determine if one heuristic performs better in cases where the other heuristic performs poorly and vice versa and thus complement each other.

Comparing SA and GA with the pearson correlation coefficient, we specify the null hypothesis $\rho = 0$ for all experiment sets. We obtain following values for coverage ($\rho_{Ex1.1} = 0.27$; $\rho_{Ex1.2} = 0.14$; $\rho_{Ex2} = 0.41$), and distance improvement ($\rho_{Ex1.1} = 0.28$; $\rho_{Ex1.2} = 0.43$; $\rho_{Ex2} = 0.11$). With sampling size 50 and a significance level for two-tailed test of 0.01, ρ needs to be greater than 0.354 to be significantly different from zero, respectively ρ needs to be smaller than -0.354 for inverse correlation. Hence, we accept the null hypothesis for coverage (Ex1.1, Ex2) and distance improvement (Exp1.1, Exp2) that there exist no correlation between GA and SA. There is even a small, but significant relationship between SA and GA for distance improvement in experiment 1.2 and coverage in experiment 2. Given these correlation measurements there is no evidence that the two heuristics complement each other.



(a) (b) (c)

Figure 17: Coverage and Distance for experiment 1.1 (a), 1.2 (b), and experiment 2 (c) for SA and GA. Error bars are given in one direction only for sake of clarity.

In addition we inspect the few instance where GA or SA were unable to find better teams than the initial Top Team. As mentioned above, GA failed to so 6x, while SA failed 25x. Out of the 6 unsuccessful GA attempts, SA similarly did not provide a better solution and provided only little improvement for the remaining 2 attempts. In contrast, GA found significantly better teams for the remaining 19 SA misses. This supports the correlation measurement above: simulated annealing and the genetic algorithm are not complementary.

7.5.2. Comparison to Real-world data

We compare the data generated by our experiment model and a real-world data set to validate our results. Slashdot is an IT-centric news site where each posted story receives extensive discussions. A reply posting between two comment authors becomes an interaction link. The skills are extracted from a news story's subdomain. We derived an expert network from the Slashdot postings from January 1st 2008 to June 30th 2008, covering the subdomains *apple*, *ask*, *entertainment*, *mobile*, *linux*, *developers*, *games*, *news*, *slashdot*, and *it*. We didn't, however, run detailed team formation experiments on the Slashdot data set as the number of available skills is limited and thus does not allow for sufficient experiment variations necessary to generate reliable results.

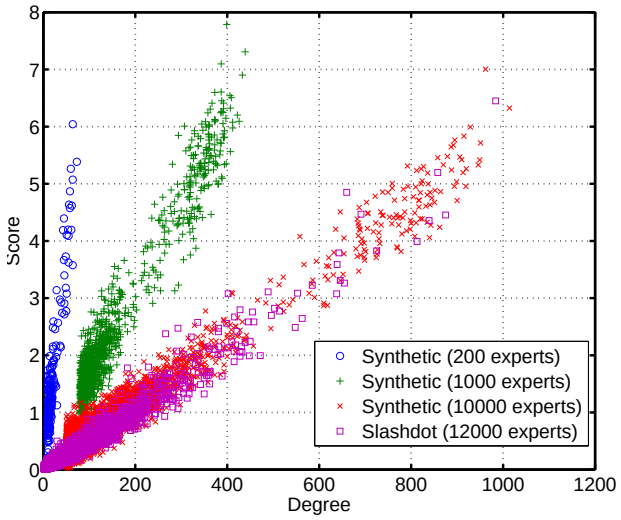


Figure 18: Degree vs. Score for synthetic network of 200, 1000, and 10000 nodes and Slashdot with 12000 nodes.

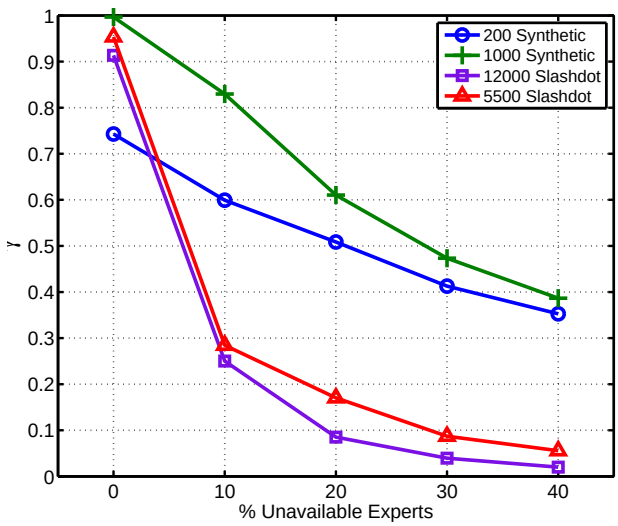


Figure 19: Effect of increasingly unavailable experts on γ for synthetic and Slashdot networks.

General social network characteristics of the Slashdot postings are discussed in [11]. In this paper, we focus on the correlation of degree and expertise to demonstrate that our simulation model correctly reproduces real-world

settings. In addition we compare the effect of unavailable experts on the candidate network density. Figure 18 depicts the degree of an expert printed against his/her total expertise score $q(\cdot)$. The compared data sets comprise the two networks used in the evaluation (blue circle, green plus), a larger synthetic set (10000 experts, red crosses) and the Slashdot data set of 12000 experts (purple squares). All four data sets derive from a different *overall skill*, *required skill*, and *expert count* configuration[10], yet we observe in every case a linear correlation between degree and score. The slope is defined by the amount of required skills (theoretical maximum score is the amount of skills) and the maximum degree.

The power-law network topology has a profound effect on the results of our team formation algorithm. The strategy for marking experts as unavailable directly affects the connectivity of the observed candidate network. Degree-based availability reduces the connectivity rapidly as predominantly well connected experts are marked as overloaded. Here γ drops from 0.75, respectively from 1, to 0.39. Already with 10% (experiment 1), respectively 20% (experiment 2), of experts unavailable, γ is below 0.6. The effect of degree-based node removal in power-law networks is well analyzed (e.g., [40]) but briefly repeated here to highlight its significance on the trade-off between interaction-based and recommendation-based team formation.

We visualize the effect of expert unavailability on γ in Figure 19. The complete Slashdot network (purple squares) comprises 8 out of 10 skills from 12000 experts. A second network (a subset of the complete Slashdot expert network (red triangles)) contains 6 skills, from which we select 5 to generate the candidate network. Degree preferential selection of Slashdot experts shows an even more swift decrease in density than our model: γ drops in both Slashdot data sets below 0.4 when removing 10% of experts and then converges to 0.05. We have to assume an inverse rich-club effect, stronger than in our model. Thus we can safely assume that a trade-off between interaction and recommendation-based distance is even more significant in real world expert crowds.

8. Conclusion

Team composition requires a balance between maintaining sufficiently high skill coverage and adequately connected team members. Finding a suitable team for a given trade-off between skills and connectivity is NP complete. The problem is further aggravated in crowds as the top experts are most popular and thus become easily overloaded. The social network structure of crowds usually lacks a rich-club structure. When a small fraction of top experts ($\approx$ 10%) becomes unavailable, the social network becomes too sparse to compose a team based on direct interactions alone. We thus propose to apply skill-dependent recommendations to substitute direct interactions. In addition, the introduced team-centric recommendation model considers only the recommendations of the team members' common neighbors. Our approach then analyzes network density to dynamically adjust the importance of direct interactions compared to recommendations.

Genetic Algorithms and Simulated Annealing are two suitable heuristics to address the multi-objective team composition problem. Genetic Algorithms apply cross over and mutation of team configurations to approach the best team while Simulated Annealing relies on a cooling schedule. Experiments have successfully shown that both heuristics are able to find significantly better solutions than the simple set of top ranked experts. The Genetic Algorithm, however, produced better and more stable results, especially for smaller networks. Analysis of increasingly unavailable experts demonstrated that our trade-off model accurately guides the transition from from direct interaction links to recommendations. Consequently the composition mechanism focuses on direct links as long as possible before experts are integrated into the team based on recommendations. Such dynamic adaptation enables to obtain teams that yield both higher recommendations and lower distance than the initial top team. Ultimately we were able to obtain team configurations that exhibit desirably low team distance, while maintaining consistently high skill coverage even when removing up to 40% experts from the social network.

We intend to continue our research in multiple directions. First, we plan to evaluate our approach with other real-world data sets which exhibit a larger skill set. The DBLP dataset is one possible candidate. This effort however has to go in hand with a rigorous analysis of skill distribution and therefore was not attempted in the scope of this work. Second, we intend to investigate how to exploit interaction patterns [41] to more accurately interpret the link strength

[10] Experiment 1: 200 experts, 8 out of 30 skills; Experiment 2: 1000 experts, 10 out of 100 skills; Experiment 3: 10000 experts, 8 out of 200 skills; Slashdot: 12000 experts, 8 out of 10 skills.

between experts. Finally, we expect exciting results from exploring how our approach can enhance existing work on team-aware process management [42] in terms of composing an optimal team for executing a process.

Acknowledgment

This work has been partially supported by the EU STREP project Commius (FP7-213876), the EU IP project COIN (FP7-216256), and Austrian Science Fund (FWF) J3068-N23.

References

[1] D. Brabham, Crowdsourcing as a model for problem solving: An introduction and cases, Convergence 14 (1) (2008) 75.

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

[3] H. Wi, S. Oh, J. Mun, M. Jung, A team formation model based on knowledge and collaboration, Expert Syst. Appl. 36 (5) (2009) 9121–9134. doi:http://dx.doi.org/10.1016/j.eswa.2008.12.031.

[4] A. Baykasoglu, T. Dereli, S. Das, Project team selection using fuzzy optimization approach, Cybern. Syst. 38 (2) (2007) 155–185. doi:http://dx.doi.org/10.1080/01969720601139041.

[5] M. Cheatham, K. Cleereman, Application of social network analysis to collaborative team formation, in: CTS '06: Proceedings of the International Symposium on Collaborative Technologies and Systems, IEEE Computer Society, Washington, DC, USA, 2006, pp. 306–311. doi:http://dx.doi.org/10.1109/CTS.2006.18.

[6] T. Lappas, K. Liu, E. Terzi, Finding a team of experts in social networks, in: KDD '09: Proceedings of the 15th ACM SIGKDD Int. Conference on Knowledge discovery and data mining, ACM, New York, NY, USA, 2009, pp. 467–476. doi:http://doi.acm.org/10.1145/1557019.1557074.

[7] M. E. Gaston, J. Simmons, M. desJardins, Adapting network structure for efficient team formation, in: AAMAS-04 Workshop on Learning and Evolution in Agent Based Systems, 2004.

[8] L. Backstrom, D. Huttenlocher, J. Kleinberg, X. Lan, Group formation in large social networks: membership, growth, and evolution, in: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, 2006, pp. 44–54. doi:http://doi.acm.org/10.1145/1150402.1150412.

[9] J. J. McAuley, L. da Fontoura Costa, T. S. Caetano, Rich-club phenomenon across complex network hierarchies, Applied Physics Letters 91 (8) (2007) 084103. doi:10.1063/1.2773951.

[10] V. Colizza, A. Flammini, M. A. Serrano, A. Vespignani, Detecting rich-club ordering in complex networks, Nature Physics 2 (2006) 110–115.

[11] V. Gómez, A. Kaltenbrunner, V. López, Statistical analysis of the social network and discussion threads in slashdot, in: WWW '08: Proceedings of the 17th Int. Conference on World Wide Web, ACM, New York, NY, USA, 2008, pp. 645–654. doi:http://doi.acm.org/10.1145/1367497.1367585.

[12] L. A. Adamic, J. Zhang, E. Bakshy, M. S. Ackerman, Knowledge sharing and yahoo answers: everyone knows something, in: WWW '08: Proceeding of the 17th Int. Conference on World Wide Web, ACM, New York, NY, USA, 2008, pp. 665–674. doi:http://doi.acm.org/10.1145/1367497.1367587.

[13] C. Bird, A. Gourley, P. Devanbu, M. Gertz, A. Swaminathan, Mining email social networks, in: MSR '06: Proceedings of the 2006 Int. Workshop on Mining software repositories, ACM Press, New York, NY, USA, 2006, pp. 137–143. doi:http://doi.acm.org/10.1145/1137983.1138016.

[14] P. Chundi, M. Subramaniam, D. K. Vasireddy, An approach for temporal analysis of email data based on segmentation, Data & Knowledge Engineering 68 (11) (2009) 1253 – 1270, including Special Section: Conference on Privacy in Statistical Databases (PSD 2008) - Six selected and extended papers on Database Privacy. doi:DOI: 10.1016/j.datak.2009.04.011.
URL `http://www.sciencedirect.com/science/article/B6TYX-4WB3N78-1/2/ca0fcdcf7cccd593416a6c32092cad23`

[15] C. Bird, D. Pattison, R. D'Souza, V. Filkov, P. Devanbu, Latent social structure in open source projects, in: SIGSOFT '08/FSE-16: Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, ACM, New York, NY, USA, 2008, pp. 24–35. doi:http://doi.acm.org/10.1145/1453101.1453107.

[16] S. Dustdar, W. Schreiner, A survey on web services composition, Int. J. Web Grid Serv. 1 (1) (2005) 1–30. doi:http://dx.doi.org/10.1504/IJWGS.2005.007545.

[17] Y. Yang, F. Mahon, M. H. Williams, T. Pfeifer, Context-aware dynamic personalised service re-composition in a pervasive service environment, in: UIC, 2006, pp. 724–735.

[18] R. Quitadamo, F. Zambonelli, G. Cabri, The service ecosystem: Dynamic self-aggregation of pervasive communication services, in: Software Engineering for Pervasive Computing Applications, Systems, and Environments, 2007. SEPCASE '07. First International Workshop on, 2007, pp. 1–10. doi:10.1109/SEPCASE.2007.11.

[19] Z. Maamar, D. Benslimane, P. Thiran, C. Ghedira, S. Dustdar, S. Sattanathan, Towards a context-based multi-type policy approach for web services composition, Data Knowl. Eng. 62 (2) (2007) 327–351. doi:http://dx.doi.org/10.1016/j.datak.2006.08.007.

[20] L. Baresi, D. Bianchini, V. D. Antonellis, M. G. Fugini, B. Pernici, P. Plebani, Context-aware composition of e-services., in: TES, 2003, pp. 28–41.

[21] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Computer Networks and ISDN Systems 30 (1-7) (1998) 107–117, proceedings of the Seventh International World Wide Web Conference. doi:DOI: 10.1016/S0169-7552(98)00110-X.

[22] T. Haveliwala, Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 784–796. doi:10.1109/TKDE.2003.1208999.

[23] D. Schall, Human interactions in mixed systems - architecture, protocols, and algorithms, PhD Thesis, Vienna University of Technology, Karlsplatz 13, 1040 Vienna, Austria (2009).
[24] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, in: International Conference on Information and Knowledge Management, ACM, 2003, pp. 556–559.
[25] C.-W. Hang, M. P. Singh, Trust-based recommendation based on graph similarity, in: 13th AAMAS Workshop on Trust in Agent Societies (Trust), 2010.
[26] F. Skopik, D. Schall, S. Dustdar, Modeling and mining of dynamic trust in complex service-oriented systems, Information Systems 35 (7) (2010) 735–757. doi:http://dx.doi.org/10.1016/j.is.2010.03.001.
[27] B. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, Journal of the Operational Research Society 57 (10) (2005) 1143–1160.
URL http://www.palgrave-journals.com/doifinder/10.1057/palgrave.jors.2602068
[28] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, 1st Edition, Wiley, 2001.
[29] M. Srinivas, L. Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, Systems, Man and Cybernetics, IEEE Transactions on 24 (4) (1994) 656 –667. doi:10.1109/21.286385.
[30] J. Zhang, H. S.-H. Chung, W.-L. Lo, Clustering-based adaptive crossover and mutation probabilities for genetic algorithms, Evolutionary Computation, IEEE Transactions on 11 (3) (2007) 326 –335. doi:10.1109/TEVC.2006.880727.
[31] K. A. D. Jong, W. M. Spears, An analysis of the interacting roles of population size and crossover in genetic algorithms, in: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature, PPSN I, Springer-Verlag, London, UK, 1991, pp. 38–47.
URL http://portal.acm.org/citation.cfm?id=645821.670188
[32] D. Weyland, Simulated annealing, its parameter settings and the longest common subsequence problem, in: Proceedings of the 10th annual conference on Genetic and evolutionary computation, GECCO '08, ACM, New York, NY, USA, 2008, pp. 803–810. doi:http://doi.acm.org/10.1145/1389095.1389253.
URL http://doi.acm.org/10.1145/1389095.1389253
[33] M.-W. Park, Y.-D. Kim, A systematic procedure for setting parameters in simulated annealing algorithms, Computers & Operations Research 25 (3) (1998) 207 – 217. doi:DOI: 10.1016/S0305-0548(97)00054-3.
URL http://www.sciencedirect.com/science/article/B6VC5-3SX6XDK-5/2/59aefca9874d9f2a4458c44aba791670
[34] H. Orsila, E. Salminen, T. D. Hämäläinen, Parameterizing simulated annealing for distributing kahn process networks on multiprocessor socs, in: Proceedings of the 11th international conference on System-on-chip, SOC'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 19–26.
URL http://portal.acm.org/citation.cfm?id=1736530.1736534
[35] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, Evolutionary Computation, IEEE Transactions on 6 (2) (2002) 182 –197. doi:10.1109/4235.996017.
[36] M. Fredman, R. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, Foundations of Computer Science, Annual IEEE Symposium on 0 (1984) 338–346. doi:http://doi.ieeecomputersociety.org/10.1109/SFCS.1984.715934.
[37] R. Albert, H. Jeong, A.-L. Barabási, The diameter of the world wide web, CoRR cond-mat/9907038.
[38] R. Albert, A.-L. Barabasi, Statistical mechanics of complex networks, Reviews of Modern Physics 74 (2002) 47.
URL http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0106096
[39] A. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.
[40] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, D. J. Watts, Network robustness and fragility: Percolation on random graphs, Phys. Rev. Lett. 85 (25) (2000) 5468–5471. doi:10.1103/PhysRevLett.85.5468.
[41] S. Dustdar, T. Hoffmann, Interaction pattern detection in process oriented information systems, Data & Knowledge Engineering 62 (1) (2007) 138 – 155. doi:DOI: 10.1016/j.datak.2006.07.010.
URL http://www.sciencedirect.com/science/article/B6TYX-4KRY749-1/2/10ace51a52186445e4c1048f911d4f35
[42] W. M. P. van der Aalst, A. Kumar, A reference model for team-enabled workflow management systems, Data & Knowledge Engineering 38 (3) (2001) 335 – 363. doi:DOI: 10.1016/S0169-023X(01)00034-9.
URL http://www.sciencedirect.com/science/article/B6TYX-4447T69-4/2/7fc8ebe39be4d6a51301e9ff1f294e76

Annex A - Extended Experiment Results

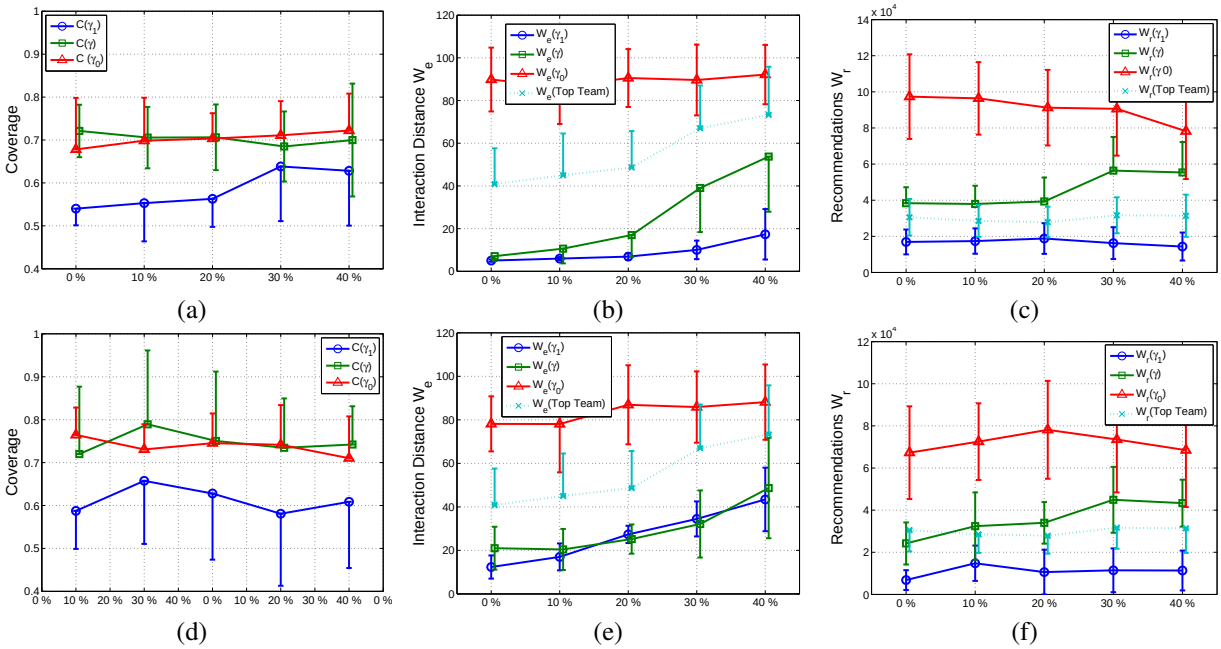


Figure 20: Experiment 1.2: Coverage C (a,d), Interaction Distance W_e (b,e), and Recommendation Distance W_r (c,f) obtained by the genetic algorithm (a-c) and simulated annealing (d-f).