Multi-Objective Group Discovery on the Social Web (Technical Report)

Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, Pierre-Francois Dutot, Denis Trystram

▶ **To cite this version:**

Behrooz Omidvar-Tehrani, Sihem Amer-Yahia, Pierre-Francois Dutot, Denis Trystram. Multi-Objective Group Discovery on the Social Web (Technical Report). [Research Report] RR-LIG-052, LIG. 2016. hal-01297763

HAL Id: hal-01297763
https://hal.science/hal-01297763v1

Submitted on 4 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Objective Group Discovery on the Social Web *(Technical Report)*

Behrooz Omidvar-Tehrani[†], Sihem Amer-Yahia[‡], Pierre-Francois Dutot[‡], and Denis Trystram[‡]

[†]The Ohio State University, USA, [‡]Univ. Grenoble Alps, CNRS, France
[†]omidvartehrani.1@osu.edu, [‡]{firstname.lastname}@imag.fr

Abstract. We are interested in discovering user groups from collaborative rating datasets of the form $\langle i, u, s \rangle$, where $i \in \mathcal{I}$, $u \in \mathcal{U}$, and s is the integer rating that user u has assigned to item i. Each user has a set of attributes that help find *labeled groups* such as *young computer scientists in France* and *American female designers*. We formalize the problem of finding user groups whose quality is optimized in multiple dimensions and show that it is NP-Complete. We develop α-MOMRI, an α-approximation algorithm, and h-MOMRI, a heuristic-based algorithm, for multi-objective optimization to find high quality groups. Our extensive experiments on real datasets from the social Web examine the performance of our algorithms and report cases where α-MOMRI and h-MOMRI are useful.

1 Introduction

Today's data scientists are faced with large volumes of data to explore. In particular, collaborative rating sites have become essential data resources to make decisions about mundane tasks such as purchasing a book, renting a movie or going to a restaurant. The availability of a number of datasets on the social Web, such as MovieLens, a movie rating site, LastFM, a music rating site and BookCrossing, a book rating site, appeals to scientists today who design algorithms that help analysts make better decisions on complex tasks such as crowd data sourcing (which users to ask ratings from), advertisers in determining which items to recommend to which users, and social scientists in validating hypotheses such as *young professionals are more inclined to buying self-help books*, on large datasets.

In practice, however, there does not exist analytics tools that enable the scalable, on-demand discovery of user groups. In this paper, we are given a dataset of rating records in the form $\langle i, u, s \rangle$, where $i \in \mathcal{I}$ (set of items), $u \in \mathcal{U}$ (set of users), and s is the integer rating that user u has assigned to item i. We define the notion of *user group* as a conjunction of demographic attributes over rating records, such as *rich young professionals* or *teachers who live in the countryside*. Given a dataset, e.g., ratings of Woody Allen movies, we formalize

the problem of discovering *high quality* user groups. Quality is formulated as the optimization of two dimensions: *coverage* and *diversity*. Optimizing coverage ensures that most input records $\langle i, u, s \rangle$ will belong to at least one group in the output. Optimizing diversity ensures that found groups are as different as possible from each other, e.g., *males and females* or *young and old*, and unveils ratings by different users. User groups with high coverage and high diversity, can help analysts make a variety of decisions such as audience targeting in advertising or hypothesis validation in social science. The following two examples illustrate two common cases in practice.

Example 1. (Audience Targeting) Julia who works in an advertising company, is responsible for finding the best target audience for a 20% reduction on the new book of *John Grisham*, the American author known for his popular thrillers. To find a target group, Julia goes to BookCrossing website[1], a database of book ratings, and finds 6,913 rating records for all Grisham's books. A group-centric examination of those records would reveal that over 89% of users who rated Grisham's book are either *i. young reviewers who live in Connecticut*, *ii. middle-age reviewers in France* or *iii. old females*. Such groups are diverse, i.e., they do not overlap because their reviewers belong to different age-categories. Julia finds the first two groups promising, as their average rating scores are 4.6 and 4.0 out of 5, respectively, while it is only 2.3 for the third group. Julia exploits these two groups for audience targeting, as they capture the attention of many readers (becasue of coverage) and address different sub-populations (because of diversity).

Example 2. (Hypothesis Validation) It is generally believed that romantic movies (e.g., *American Beauty*, 1999) are mostly watched by females. This observation is based on *demographic breakdown* reports on IMDb.[2] Anna, who is a social scientist, wants to validate this hypothesis by exploring diverse user groups that cover most ratings for *romance* genre movies. Such a group-centric examination would provide the following 3 user groups: *i. female reviewers from DC* (District of Columbia), *ii. young female reviewers*, and *iii. male teenager reviewers* with average ratings of 4.6, 3.7 and 3.1 (out of 5), respectively. By observing those groups, Anna finds that the hypothesis holds only for a sub-population of female reviewers, *middle-age* or *residents of DC*. Also the results show another group of *romance* genre lovers, *male teenagers*, which contradicts the hypothesis. Anna is confident in her observation (as the results has high coverage) and she can notice different aspects of her hypothesis (as results are diversified).

Beyond coverage and diversity, another interesting dimension of group quality is its *rating distribution*. As it has been argued in previous work [4], groups with *homogeneous* ratings may be more appealing to some applications, while groups with *polarized* ratings are preferred by others. Indeed the rating distribution in a group provides analysts with the ability to tune the quality of found

[1] *http://www.bookcrossing.com*
[2] *http://www.imdb.com*

groups according to specific needs. Example 2 is a good case for *homogeneity*. By reporting the average rating of 4.6 for young female reviewers, we know that most individuals in that group have high ratings. The following example shows how tuning the *rating distribution* of discovered groups leads to new discoveries when used alongside coverage and diversity.

Example 3. Following Example 2, Anna then looks at the *variance* of ratings in those groups and finds that *male teenager reviewers* has a higher variance comparing to two other groups. This potentially shows that not all male teenagers like romantic movies. Anna is more interested in a homogenous group, so she can either choose the second or third group or ask the system to find other groups specifically for males or teenagers.

Given an input set of rating records (e.g., Sci-Fi movies from the 90's, David Lynch movies, movies starring Scarlett Johansson), our problem is that of discovering a set of user groups. Even when the number of records is not very high, the number of possible groups that could be built may be very large. Indeed, the number of groups is exponential in the number of user attribute values and many groups are very small or empty. Therefore, given the ad-hoc and online nature of group discovery, our challenge is to *quickly* identify high quality user groups. We hence define desiderata that user groups should satisfy (local desiderata) and those that must be satisfied by the set of returned groups (global desiderata).

Local desiderata:

- *Describability*: Each group should be easily understandable by the analyst. While this is difficult to satisfy through unsupervised clustering of ratings, it is easily enforced in our approach since each group must be formed by rating records of users that share at least one attribute value, which is used to describe that group.
- *Size*: Returning groups that contain too few rating records is not meaningful to the analyst. We hence need to impose a minimum size constraint on groups.

Global desiderata:

- *Coverage*: Together, returned groups should cover most input rating records. While ideally we would like each input record to belong to at least one group, that is not always feasible due to other local and global desiderata associated with the set of returned groups.
- *Diversity*: Returned groups need to be different from each other in order to provide complementary information on users.
- *Rating Distribution*: Ratings in selected groups should follow a requested distribution (e.g., homogeneity).
- *Number of groups*: The number of returned groups should not be too high in order to provide the analyst with an at-a-glance understanding of the data.

A candidate solution is a group-set that verifies all above desiderata. Finding such a group-set is a hard problem because of two reasons. First the pool of candidate group-sets is very large as any possible combination of attribute value pairs can form a group, and any number of groups can form a group-set. By having only 20 attribute value pairs, we end up with $1,048,575$ groups (i.e., $(2^{20}) - 1$) and over 10^{12} group-sets of size 5 (i.e., $1,048,575$ choose 5). The second reason of hardness is that diversity, coverage and rating distribution are conflicting objectives (Section 5.1), i.e., optimizing one does not necessarily lead the best values for others. Thus the need for a multi-objective optimization approach that will not compromise one objective over another. Such an approach would return *the set of all candidate group-sets* that are not dominated by any other along all objectives.

In this paper, we propose α-MOMRI, an α-approximation algorithm for user group discovery that considers local and global desiderata and guarantees to find group-sets that are α-far from optimal ones. Since α-MOMRI relies on an exhaustive search in the space of all groups, we propose h-MOMRI, a heuristic that exploits the lattice formed by user groups and prunes exploration in order to speed up group-set discovery. Both our algorithms admit a set of rating records of the form $\langle i, u, s \rangle$ and a constrained multi-objective optimization formulation [5] and return group-sets that satisfy the formulation and are not dominated by any other group-set. The contributions of this paper are as follows.

1. We formalize specific quality dimensions (coverage, diversity and rating distribution) which we find to be the most natural for discovering user groups on the Social Web. We exploit the semantics of these objectives to go beyond a generic approach.
2. We formalize the problem of discovering user groups as a constrained multi-objective optimization problem with quality dimensions as objectives.
3. We develop α-MOMRI, an α-approximation algorithm for user group discovery. Returned group-sets are instances of Pareto plans and are guaranteed to be α-far from optimal ones.
4. We develop h-MOMRI, a heuristic-based algorithm that exploits the lattice formed by user groups to speed up group discovery.
5. In an extensive set of experiments on MovieLens and BookCrossing datasets, we analyze different solutions of α-MOMRI and h-MOMRI and show that high quality group-sets are returned by our approximation and very good response time is achieved by our heuristic.

2 Data Model and Preliminaries

We model our database $\mathcal{D}$ as a triple $\langle \mathcal{I}, \mathcal{U}, \mathcal{R} \rangle$, representing the sets of items, reviewers and rating records respectively. Each rating record $r \in \mathcal{R}$ is itself a triple $\langle i, u, s \rangle$, where $i \in \mathcal{I}$, $u \in \mathcal{U}$, and s is the integer rating that reviewer u has assigned to item i. The values of s are application-dependent and do not affect our model.

$\mathcal{I}$ is associated with a set of attributes, denoted as $\mathcal{I}_A = \{ia_1, ia_2, \ldots\}$, and each item $i \in \mathcal{I}$ is a tuple with $\mathcal{I}_A$ as its schema. In other words, $i = \langle iv_1, iv_2, \ldots\rangle$, where each iv_j is a set of values for attribute ia_j. For example, for the movie *Kazaam* (1996) in MovieLens dataset, the set of attribute values are $\langle$`Paul M. Glaser`, $\{$`Comedy`, `Fantasy`$\}\rangle$ for the attribute schema $\langle$`director`, `genre`$\rangle$. Note that the attribute `genre` is multi-valued. Another example is for the book *Wild Animus* (2004) in BookCrossing dataset where the set of attribute values are $\langle$`Rich Shapero`, `Too Far`$\rangle$ for the attribute schema $\langle$`author`, `publisher`$\rangle$.

We also have the schema $\mathcal{U}_A = \{ua_1, ua_2, \ldots\}$ for reviewers, i.e., $u = \langle uv_1, uv_2, \ldots\rangle \in \mathcal{U}$, where each uv_j is a value for attribute ua_j. As a result, each rating record, $r = \langle i, u, s\rangle$, is a tuple, $\langle iv_1, iv_2, \ldots, uv_1, uv_2, \ldots, s\rangle$, that concatenates the tuple for i, the tuple for u, and the numerical rating score s. The set of all attributes is denoted as $A = \{a_1, a_2, \ldots\}$. We now define the notion of user group.

Definition 1 (User Group). *A group g is a set of rating records $\langle u, i, s\rangle$ described by a set of attribute value pairs shared among the reviewers and the items of those rating records. The description of a group g is defined as $\{\langle a_1, v_1\rangle, \langle a_2, v_2\rangle, \ldots\}$ where each $a_i \in A$ and each v_i is a set of values for a_i. By $|g|$, we denote the number of rating records contained in g.*

ID	Movie	Name	Gender	Age	Occup.	Rating
r_1	Toy Story	John	M	young	teacher	4
r_2	Toy Story	Jennifer	F	old	teacher	3
r_3	Toy Story	Mary	F	old	teacher	2
r_4	Titanic	Carine	F	old	other	4
r_5	Toy Story	Sara	F	young	student	3
r_6	Toy Story	Martin	M	young	student	5
r_7	Titanic	Peter	M	young	student	1

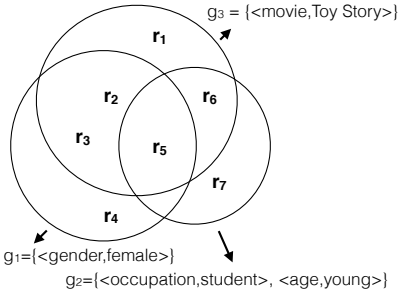

Fig. 1. Example Dataset and Group-set

For instance, the first group in Example 2, $g = \{\langle$`gender`, `female`$\rangle$, $\langle$`location`, `DC`$\rangle$, $\langle$`genre`, `romance`$\rangle\}$ contains rating records in MovieLens for romance movies whose reviewers are all females in DC. Figure 1 illustrates an example dataset with 7 rating records. The user group g_1 is for female reviewers with 4 rating records, and g_2 is for young students with 3 rating records. Note that there exists one record in common between two mentioned user groups (r_5). Note that a user group differs from a *where-clause* SQL query, since our objectives and constraints are not expressible as SQL predicates.

Given a rating record $r = \langle v_1, v_2 \ldots, v_k, s\rangle$, where each v_i is a set of values for its corresponding attribute in the schema A, and a group $g = \{\langle a_1, v_1\rangle,$

$\langle a_2, v_2\rangle, \ldots, \langle a_n, v_n\rangle\}, n \leq k$, we say that g covers r, denoted as $r \lessdot g$, iff $\forall i \in [1, n]$, $\exists r.v_j$ such that v_j is a set of values for attribute $g.a_i$ and $g.v_j \subseteq r.v_i$. For example, the rating $\langle$`female`, `DC`, `student`, $4\rangle$ is covered by the group $\{\langle$`gender`, `female`$\rangle$, $\langle$`location`, `DC`$\rangle\}$.

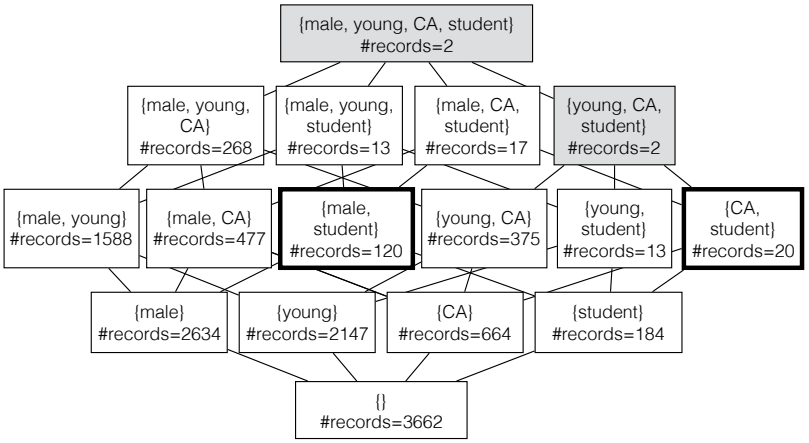

Fig. 2. Partial Lattice for the Movie *Toy Story*

Similarly to data cubes, the set of all possible groups form a lattice where nodes correspond to groups and edges correspond to parent/child and ancestor/descendant relationships. A partial lattice for rating records of the movie *Toy Story* is illustrated in Figure 2 where we have four reviewer attributes to analyze: `gender`, `age`, `location` and `occupation`. For simplicity, exactly one distinct value per attribute is shown in the Figure.

2.1 Group Quality Dimensions

We now define three quality dimensions for groups, i.e., coverage, diversity and rating distribution. We are given a set of rating records $R \subseteq \mathcal{R}$ and a group-set G.

Coverage is a value between 0 and 1 and measures the percentage of rating records in R contained in groups in G.

$$coverage(G, R) = |\cup_{g \in G} (r \in R, r \lessdot g)|/|R| \quad (1)$$

For instance, in Figure 1, $coverage(G, R) = 0.8$ where $G = \{g_1, g_2\}$ and R contains rating records for the movie *Toy Story*.

Diversity is a value between 0 and 1 that measures how distinct groups in group-set G are from each other. Diversity penalizes group-sets containing overlapping groups. To prioritize groups with few overlaps, the overlapping penalty

is considered exponential.

$$diversity(G, R) = 1/(1 + \Sigma_{g_1, g_2 \in G} | r \in R, r \lessdot g_1 \wedge r \lessdot g_2 |) \quad (2)$$

For instance, in Figure 1, $diversity(G, R) = 0.5$.

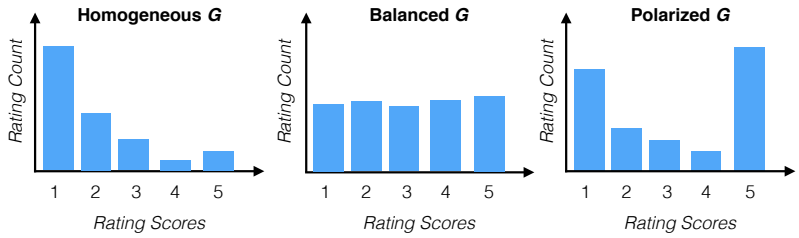

Fig. 3. Different Rating Distributions for a Group-Set

Rating Distribution. A group-set G may be characterized by its rating distribution. Figure 3 illustrates some examples of distributions. A rating distribution is a function over the set of ratings in the rating records of groups in G. Equation 3 shows an example of such a function which computes the average *diameter* of ratings. Other aggregation functions could be defined.

$$diameter(G) = avg_{g \in G}(max_{r \in g}(r.s) - min_{r' \in g}(r'.s)) \quad (3)$$

In Figure 1, $diameter(G) = 3$. We now explain different rating distributions in Figure 3.

Homogeneous. A homogeneous rating distribution shows that all users in G have approximately agreed on a unique score (i.e., "1" in Figure 3). We use this rating distribution when we are seeking a consensus between group members and to provide a *representative unique score* for the whole group-set. An example for this rating distribution is the movie *The Godfather* (1972) in IMDb, as 53.7% of ratings are for the highest score.[3]

Balanced. A balanced rating distribution shows that the preference of group members are equally distributed among scores. A user group with balanced rating distribution counts as a *neutral* group: there is no preference for any score. A neutral group can be used as a reference to see how other groups are biased towards a score.

Polarized. A polarized rating distribution shows that group members have the farthest possible preferences from each other over the set of rating records. A

[3] *http://www.imdb.com/title/tt0068646/ratings?ref_=tt_ov_rt*

real example for this rating distribution is the movie *Fifty Shades of Grey* (2015) in IMDb, as 28.8% and 15.9% of ratings are for the lowest and highest scores, respectively.[4]

Increasing/Decreasing. We can take into consideration many other distributions depending on problem needs and specifications. For instance, *increasing* rating distribution is the one where for each score s, the number of rating records with score s is larger than or equal to the one for $s-1$. *Decreasing* rating distribution is also the inverse of the above distribution. In these two rating distributions, there exists a total order between the number of rating records in consecutive scores. A group with increasing/decreasing rating distribution potentially represents rising/falling items, i.e., items which currently have relatively low/high acceptability but may eventually emerge as prominent popular/weak items.

Based on Definition 3, a small value of $diameter(G)$ leads a homogeneous group-set G and a high value leads a polarized group-set G.

2.2 Multi-Objective Optimization Principles

We propose to use the quality dimensions (coverage, diversity and rating distribution) defined as optimization objectives. When dealing with more than one dimension to optimize, there may be many incomparable group-sets. For instance, for a set of ratings R, we can form two group-sets, G_1 with $coverage(G_1, R) = 0.8$ and $diversity(G_1, R) = 0.4$ and G_2 with $coverage(G_2, R) = 0.5$ and $diversity(G_2, R) = 0.7$. Each group-set has its own advantage: the former has higher coverage and the latter has higher diversity. Another group-set G_3 with $coverage(G_3, R) = 0.5$ and $diversity(G_3, R) = 0.2$ has no advantage compared to G_1, hence it can be ignored. In other words, G_3 is dominated by G_1. In this section, we borrow the terminology of multi-objective optimization and define these concepts.

Definition 2 (Plan). *Plan p_i, associated to a group-set G_i for a set of rating records $R \subseteq \mathcal{R}$, is a tuple* $\langle |G_i|, coverage(G_i, R), diversity(G_i, R), diameter(G_i) \rangle$.

Definition 3 (Sub-plan). *Plan p_i is the sub-plan of another plan p_j if their associated group-sets satisfy $G_i \subseteq G_j$.*

Definition 4 (Dominance). *Plan p_1 dominates p_2 if p_1 has* better *or equivalent values than p_2 in every objective. The term* "better" *is equivalent to* "greater" *for* maximization *objectives (e.g., diversity, coverage and polarization), and* "lower" *for* minimization *ones (e.g., homogeneity). Furthermore, plan p_1 strictly dominates p_2 if p_1 dominates p_2 and the values of objectives for p_1 and p_2 are not equal.*

Definition 5 (Pareto Plan). *Plan p is Pareto if no other plan strictly dominates p. The set of all Pareto plans is denoted as $\mathcal{P}$.*

[4] *http://www.imdb.com/title/tt2322441/ratings?ref_=tt_ov_rt*

3 Problem Definition

We define our constrained multi-objective optimization problem as follows: for a given set of rating records R and integer constants σ and k, the problem is to identify all group-sets, such that each group-set G satisfies:

- $coverage(G, R)$ is maximized;
- $diversity(G, R)$ is maximized;
- $diameter(G)$ is optimized;
- $|G| \leq k$;
- $\forall g \in G : |g| \geq \sigma$.

The last constraint states that a group g should contain at least σ rating records, an application-defined threshold. For example, if we fix σ to 10 rating records, the groups highlighted in gray in Figure 2 will not be returned. Note that while we always maximize coverage and diversity, we may either minimize (e.g., in case of homogeneity) or maximize (e.g., in case of polarization) the diameter based on the analyst's needs.

We state the complexity of our problem as follows.

Theorem 1. *The decision version of our problem is NP-Complete.*

Proof. (sketch) It is shown in [4] that a single-objective optimization problem for user group discovery is NP-Complete by a reduction from the Exact 3-Set Cover problem (EC3). There, homogeneity is maximized and a threshold on coverage is satisfied. In our case, two new conflicting dimensions (diversity and coverage) are added. This means that the problem in [4] is a *special case* of ours, hence our problem is obviously harder. □

4 Algorithm

The main challenge in designing an algorithm for user group discovery, is the multi-objective nature of the problem. A multi-objective problem can be easily solved if *i.* it is possible to combine all objective dimensions into a single dimension (scalarization), or *ii.* if optimizing one dimension leads an optimized value for other dimensions. First, it is not possible in our problem to combine all objective dimensions into a single dimension [6]. We provide an intuition of the reason in the following example.

Example 4. Let us consider the *sum* aggregation function to combine coverage and diversity values of a plan into a single score. Let p_1 and p_2 be two plans corresponding to two group-sets G_1 and G_2 respectively, and $coverage(G_1, R)$=0.5, $diversity(G_1, R) = 0.8$, $coverage(G_2, R)$=0.6 and $diversity(G_2, R) = 0.1$. In this case, the aggregated score of p_1 is 1.3 and the score of p_2 is 0.7. Hence, we would incorrectly prune p_2 while it has a higher value for coverage.

Second, our objectives are *conflicting*, i.e., optimizing one does not necessarily lead to an optimized value for others (Section 5.1). For instance, a group-set may cover almost all input rating records but contains highly overlapping groups thereby hurting its diversity.

In this paper, we discuss 3 different algorithms for our problem: exhaustive, approximation and heuristic.

4.1 Exhaustive Algorithm

The exhaustive algorithm starts by calculating Pareto plans for single groups. Then it iteratively calculates plans for group-sets containing more than one group by combining single groups. At each iteration, dominated plans are discarded. The algorithm combines sub-plans to obtain new plans and exploits the *optimality principle* (POO) for pruning. POO is defined as follows.

Definition 6 (POO). *Given a maximization objective f (e.g., diversity, coverage, polarization) and plans p_1 and p_2 with sub-plans p_{11}, p_{12} for p_1 and p_{21}, p_{22} for p_2, if $f(G_{11}) \geq f(G_{21})$ and $f(G_{12}) \geq f(G_{22})$, then $f(G_1)$ cannot be lower than $f(G_2)$. The extension for a minimization objective is straightforward.*

This approach makes an exhaustive search over all combinations of user groups to find Pareto plans. This is both time and space consuming [6].

We propose two ways of improving the complexity of the exhaustive algorithm: *approximation-based* and *heuristic-based*. An approximation algorithm makes less enumerations with a theoretical guarantee on the quality of results. On the other hand, a heuristic can exploit the properties of the search space and prevent a brute-force execution.

4.2 Approximation Algorithm

For our approximation algorithm, we exploit the near-optimality principle (PONO) [14].

Definition 7 (PONO). *Given a maximization objective f (e.g., diversity, coverage, polarization) and $\alpha \geq 1$, let p_1 be a plan with sub-plans p_{11} and p_{12}. Derive p_2 from p_1 by replacing p_{11} by p_{21} and p_{12} by p_{22}. Then $f(G_{21}) \geq f(G_{11}) \times \alpha$ and $f(G_{22}) \geq f(G_{12}) \times \alpha$ together imply $f(G_2) \geq f(G_1) \times \alpha$. The extension for a minimization objective is straightforward.*

In Section 8, we formally prove that all our objectives (coverage, diversity and diameter) satisfy POO and PONO. Note that the functions for quality dimensions (coverage, diversity and diameter) are chosen in a way to satisfy POO and PONO.

PONO overrides POO (Definition 6). Thus a new notion of dominance is introduced in Definition 8 to be in line with PONO.

Definition 8 (Approximated Dominance). *Let $\alpha \geq 1$ be the precision value, a plan p_1 α-dominates p_2 if for every objective f, $f(G_1) \geq f(G_2) \times \alpha$ where $f \in \{coverage, diversity, polarization\}$ and $f(G_1) \leq f(G_2) \times \alpha$ where f is homogeneity.*

Algorithm 1: α-approximation MOMRI (α-MOMRI)

```
   Input: k, α > 1, R
   Output: Pareto result set P_α
 1 P_α ← ∅
 2 for all user groups g do
 3 |   p_g ← construct_plan(g)
 4 |   if p_g is not α-dominated by any other plan in P_α then P_α.add(p_g)
 5 end
 6 for n ∈ [2, k] do
 7 |   for group-sets G of size n do
 8 |   |   p_G ← construct_plan(g_G)
 9 |   |   if p_G is not α-dominated by any other plan in P_α then P_α.add(p_G)
10 |   end
11 end
12 return P_α
```

Definition 9 (Approximated Pareto Plan). *For a precision value α, plan p is an α-approximated Pareto plan if no other plan α-dominates p.*

Generating fewer plans makes a multi-objective optimization algorithm run faster [14]. This is because the execution time heavily depends on the number of generated plans. Thus a pruning strategy dictated by PONO is at the core of an approximation algorithm for multi-objective optimization.

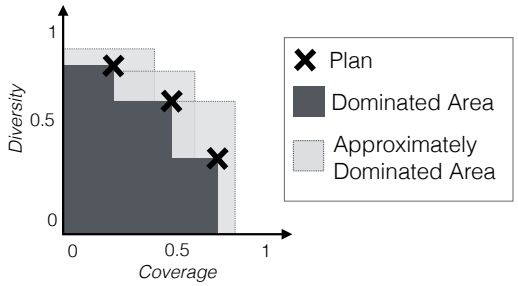

Fig. 4. Dominance Areas

Our α-approximation algorithm α-MOMRI is illustrated in Algorithm 1. The main idea is to exploit a dynamic programming approach. The algorithm begins by constructing a plan for each single user group (lines 2 to 5). We keep all non α-dominated plans of single groups in a buffer. Then it builds group-sets of size 2 up to size k using plans in the buffer (lines 7 to 11). After each iteration, we remove α-dominated plans from the buffer. At the end, we return the buffer content.

Algorithm 2: Heuristic MOMRI (h-MOMRI)

Input: k, α, R
Output: Result set $\mathcal{P}_h$
1 $\mathcal{P}_h \leftarrow \emptyset$
2 $\mathcal{N} \leftarrow$ Set of intervals on *diversity* values
3 **for** n *times* **do**
4 $\quad G_s \leftarrow random\_groupset(k)$
5 $\quad G_s^* \leftarrow SHC(G_s)$
6 $\quad interval \leftarrow get\_interval(G_s^*)$
7 $\quad \mathcal{N}[interval].add(G_s^*)$
8 **end**
9 **for** $interval \in \mathcal{N}$ **do**
10 $\quad$ Keep non-dominated plans in *interval* and add them to $\mathcal{P}_h$
11 **end**
12 $\mathcal{P}_h \leftarrow optimize\_diameter(\mathcal{P}_h)$
13 **return** $\mathcal{P}_h$

The crucial part of this simple algorithm is its pruning mechanism using the precision value α. In the special case of $\alpha = 1$, the algorithm operates exhaustively (as described in Section 4.1). If $\alpha > 1$, the algorithm prunes more and hence is faster. In the latter case, a new plan is only compared with all plans that generate the same result. But a new plan are only inserted into the buffer if no other plan approximately dominates it. This means that α-MOMRI tends to insert fewer plans than the exhaustive algorithm. Figure 4 helps illustrate this statement using two of our objectives: diversity and coverage. The exhaustive algorithm inserts new plans if they do not fall within the dominated area, but α-MOMRI inserts new plans if they neither fall into the dominated nor into the approximately dominated area.

4.3 Heuristic Algorithm

A heuristic algorithm has obviously its own advantages and disadvantages. Of course a heuristic algorithm does not provide any approximation guarantee. Eventually, it returns a subset of Pareto set. Nevertheless, the fact that it generates a subset of Pareto makes it faster.

Algorithm 2 illustrates our heuristic algorithm. The algorithm starts by making n different iterations on finding optimal points to avoid local optima (lines 3 to 8). At each iteration, the algorithm begins with a random group-set of size k called G_s (line 4). Then a *Shotgun Hill Climbing* [13] local search approach (*SHC*) is executed (Algorithm 3) to find the group-set with optimal value starting from G_s (line 5). *SHC* maximizes coverage. Diversity is already divided into intervals $\mathcal{N}$ for each of which a buffer is associated. The resulting group-set of *SHC* is placed in the buffer whose interval matches the diversity value of the group-set (line 7). Finally, n different solutions are distributed in different interval buffers. The algorithm then iterates over interval buffers to prune dominated

plans (lines 9 to 11). Based on Definition 4, a plan is pruned and removed from its buffer if it is dominated by other plans. Finally, for each interval, we report one unique solution that has the maximum/minimum value for diameter based on the requested distribution (line 12).

SHC operates on a generalization/specialization lattice of groups (as in Figure 2). Navigation of this lattice in a downward fashion satisfies a monotonicity property for coverage: given any two groups g_1 and g_2 where g_1 is the parent of g_2, the coverage of g_1 is no smaller than the coverage of g_2.

Note that in a bi-objective context, *SHC* can optimize each one of coverage and diversity. However, to benefit from the monotonicity property, we use *SHC* to optimize coverage. Nevertheless, if we optimize diversity using *SHC*, navigation in the generalization/specialization lattice is nothing but a random walk over the space of groups.

Algorithm 3: Shotgun Hill Climbing (*SHC*) Algorithm

Input: Group-set G, R
Output: Optimized group-set G^*

```
1  G* ← ∅
2  while true do
3  |  C ← ∅
4  |  for g ∈ G and each lattice-based parent g' of g do
5  |  |  G' ← G − {g} + {g}'
6  |  |  C.add(G', coverage(G', R))
7  |  end
8  |  let (G'_m, coverage(G'_m, R)) be the pair with maximum coverage
9  |  if coverage(G'_m, R) ≤ coverage(G, R) then
10 |  |  G* ← G
11 |  |  return G*
12 |  end
13 |  G ← G'_m
14 end
```

SHC verifies all local neighbors of a group for an improvement of coverage. If no improvement is achieved, it stops and returns the current group-set. For instance, consider the input group-set $G_s = \{g_1, g_2\}$ where $g_1 = \{\langle \texttt{gender}, \texttt{male}\rangle, \langle \texttt{occupation}, \texttt{student}\rangle\}$ and $g_2 = \{\langle \texttt{location}, \texttt{CA}\rangle, \langle \texttt{occupation}, \texttt{student}\rangle\}$. These two groups are marked in bold boxes in Figure 2. We obtain a coverage of 0.79 for G_s. Keeping g_2 fixed, the resulting combinations by swapping g_1 with its parents are either $g_3 = \{\langle \texttt{gender}, \texttt{male}\rangle\}$ or $g_4 = \{\langle \texttt{occupation}, \texttt{student}\rangle\}$. For instance, the coverage of $G'_s = \{g_2, g_3\}$ is 0.81. As we observe an improvement, we iterate on this new group-set G'_s to improve coverage.

4.4 Complexity Analysis

α-MOMRI: For each group-set of size between 1 and k, α-MOMRI verifies the whole buffer content β (as in Algorithm 1) for dominance. Thus the time complexity of α-MOMRI in the worst case is $\mathcal{O}(k.\binom{|\mathcal{G}|}{k}.|\beta|)$ where G is the set of all k user groups. Size of the buffer is dictated by the number of Pareto plans generated by the algorithm (hence a function of α). In case of an unlimited buffer (i.e., the case for the exhaustive algorithm), the complexity becomes $\mathcal{O}(k.\binom{|\mathcal{G}|}{k}.|\mathcal{G}|)$ Note that $|\mathcal{G}| \gg |\beta|$ for α-MOMRI. This is why the approximation algorithm can perform much better than the exhaustive algorithm.

h-MOMRI: The time complexity of $\mathcal{SHC}$ in the worst case is $T_{\mathcal{SHC}} = \mathcal{O}(h.2^{|A|})$ where h is the height of the generalization/specialization lattice and A is the set of all attributes. The complexity of h-MOMRI is then $\mathcal{O}(n.(T_{\mathcal{SHC}} + |\mathcal{N}|))$ where $|\mathcal{N}|$ is the number of diversity intervals.

Comparison: Concerning buffer size, it is always bound to n for h-MOMRI and potentially $n \ll |\beta|$. Also the execution of α-MOMRI depends on the size of the group space (G) which is not the case for the heuristic algorithm. Hence h-MOMRI is faster.

5 Experiments

We run 3 sets of experiments. The first set justifies the need for multi-objective optimization. In the second set, we vary different parameter values in order to find the most appropriate values. The last set is a comparative evaluation of α-MOMRI and h-MOMRI on the quality of retuned groups and the scalability of those algorithms.

We consider two different rating datasets for our study: MovieLens and BookCrossing. Both datasets have approximately the same number of ratings. BookCrossing has one order of magnitude more users and items. Ratings in MovieLens are expressed on a scale from 1 to 5 (higher values denoting higher appreciation) while in BookCrossing, it is from 1 to 10. We divide the ratings of the latter dataset by two, to make both datasets uniform. A *cleaning* phase was also necessary for BookCrossing as the structure is often broken (e.g., poor coded characters, lack of value, lack of separator, etc.) and this led to pruning 118,606 unstructured ratings to finally obtain 1,031,175 ratings. We briefly explain the dataset attributes we employ.

MovieLens Attributes: We consider four user attributes: `gender`, `age`, `occupation` and `zipcode`. The attribute `gender` takes two distinct values: `male` or `female`. We convert the numeric age into four categorical attribute values, namely `teenager` (under 18), `young` (18 to 35), `middle-age` (35 to 55) and `old` (over 55). There are 21 different occupations listed in MovieLens e.g., student, artist, doctor, lawyer, etc. Finally, we convert zipcodes to states in the USA (or to

`foreign`, if not in USA) by using the USPS zip code lookup.[5] This produces the user attribute `location` which takes 52 distinct values. Concerning item attributes, MovieLens only provides movie genres. Thus we enriched this dataset by crawling IMDb[6] using the OMDb API.[7] This gives us the `director`, `writer` and `release year` of each movie.

BookCrossing Attributes: There are only two attributes for each user in BookCrossing: `age` and `location`. Concerning `age`, we apply the same conversion we made for MovieLens. Note that in this dataset, the `age` attribute is missing for 110,776 users. Concerning `location`, we consider different levels (city, state and country) as different independent attributes, hence we end up with 4 different user attributes. Note that unlike MovieLens, users of BookCrossing are not located only in the USA. BookCrossing also offers information on each book (item), i.e., `writer`, `release year` and `publisher`.

We implement our prototype system using JDK 1.8.0. All scalability experiments are conducted on an 2.4 GHz Intel Core i5 with 8 GB of memory on OS X 10.9.5 operating system.

For our experiments, we consider four different sets of input rating records described in Table 1. Each item contains at least 50 ratings. We assume that it is straightforward to analyze less than 50 ratings, manually.

Dataset	Item (movie or book)	Characteristic
MovieLens	American Beauty	Highest number of ratings
	Celtic Pride	Lowest number of ratings
	Sanjuro	Highest avgerage rating
	Kazaam	Lowest average rating
BookCrossing	Wild Animus	Highest number of ratings
	Scarlet Letter	Lowest number of ratings
	Free	Lowest avgerage rating
	Ground Zero & Beyond	Highest avgerage rating

Table 1. Input Sets of Rating Records

For an input set of rating records, our algorithms return a set of group-sets. We now illustrate an example output of α-MOMRI. The same observation holds for h-MOMRI. Given a set of rating records R for the movie *American Beauty* in MovieLens, $k = 3$, $\sigma = 10$ and the request for minimizing the rating diameter (i.e., homogeneity), one of the returned group-sets is $G_1 = \{g_1, g_2, g_3\}$ where $g_1 = \{\langle$`gender`, `male`$\rangle\}$, $g_2 = \{\langle$`gender`, `female`$\rangle$, $\langle$`age`, `old`$\rangle\}$ and $g_3 = \{\langle$`gender`, `female`$\rangle$, $\langle$`location`, `CT`$\rangle\}$. The objective values for G_1 are as follows: $coverage(G_1, R)$=0.74, $diversity(G_1, R)$=0.25 and $diameter(G_1, R)$=0.38.

[5] *http://zip4.usps.com*
[6] *http://www.imdb.com*
[7] *http://www.omdbapi.com*

This group-set has a high coverage, as it only misses female reviewers who are neither old nor residents of Connecticut. It also has a high diversity, as only 3 female reviewers (out of 946) for *American Beauty* are both old and residents of Connecticut. Finally, it has also a low diameter, i.e., all groups in G_1 are homogeneous.

Another group-set for R is $G_2 = \{g_4, g_5, g_6\}$ where g_4=$\{\langle$`gender`, `male`$\rangle$, $\langle$`age`, `teen-ager`$\rangle\}$, $g_5 = \{\langle$`location`, `AZ`$\rangle\}$ and $g_6 = \{\langle$`age`, `old`$\rangle\}$. The objective values for G_2 are as follows: *coverage*(G_2, R)=0.1, *diversity*(G_2, R)=0.33 and *diameter*$(G_2, R) = 0.11$. While G_2 has a lower coverage than G_1, it has a better score for the two other objectives. Thus G_1 and G_2 are incomparable.

5.1 Need for Multi-Objective Optimization

What is the added value of multi-objective optimization? We compare first MOMRI with MRI [4], a single-objective approach for group discovery which some authors of this work have already proposed. MRI minimizes *diameter* and considers a lower bound on coverage *min_c*. Given a set of rating records R for the movie *American Beauty* in MovieLens, $k = 3$, $min\_c = 0.7$, one of the returned group-sets by MRI is $G_{MRI} = \{g_1, g_2, g_3\}$ where $g_1 = \{\langle$`gender`, `female`$\rangle, \langle$`age`, `young`$\rangle\}$, $g_2 = \{\langle$`occupation`, `student`$\rangle$, $\langle$`age`, `young`$\rangle\}$ and $g_3 = \{\langle$`gender`, `male`$\rangle$, $\langle$`occupation`, `student`$\rangle\}$. The objective values for G_{MRI} are as follows: *coverage*(G_{MRI}, R)=0.81, *diversity*(G_{MRI}, R)=0.03 and *diameter*$(G_{MRI}, R) = 0.13$. However, as diversity is not optimized, there exists huge overlap in groups: many young reviewers are also students.

In the same context, one returned group-set by MOMRI is the one we already discussed in Example 2: $G_{MOMRI} = \{g_4, g_5, g_6\}$ where $g_4 = \{\langle$`gender`, `female`$\rangle, \langle$`age`, `young`$\rangle\}$, $g_5 = \{\langle$`age`, `young`$\rangle, \langle$`location`, `DC`$\rangle\}$ and $g_6 = \{\langle$`gender`, `male`$\rangle$, $\langle$`age`, `teen` $-$ `ager`$\rangle\}$. The objective values for G_{MOMRI} are as follows: *coverage*(G_{MOMRI}, R)=0.79, *diversity*(G_{MOMRI}, R)=0.33 and *diameter*$(G_{MOMRI}, R) = 0.11$. This group-set has optimized values on all objectives. Specifically, it has a high diversity as only 2 female reviewers for *American Beauty* are both young and residents of DC. It also shows that *min_c* in MRI is a hard constraint and can easily miss a promising result which has a very high coverage but does not meet the threshold.

We already discussed that consistency of objectives transforms the multi-objective problem into a single-objective one that is trivial to solve (Section 4). In this experiment, we verify if our objectives (defined in Section 2.1) are consistent. We maximize coverage and observe how values of diversity and diameter evolve. To maximize coverage, we use Algorithm 3. Figure 5 illustrates the results for different sets of input rating records in Table 1. Each point illustrates the objective values for each of 20 runs. Note that this experiment is independent of the heuristic and the approximation algorithms.

We observe that in general, no correlation exists between the optimized value of coverage and other objectives. Thus each objective should be optimized independently.

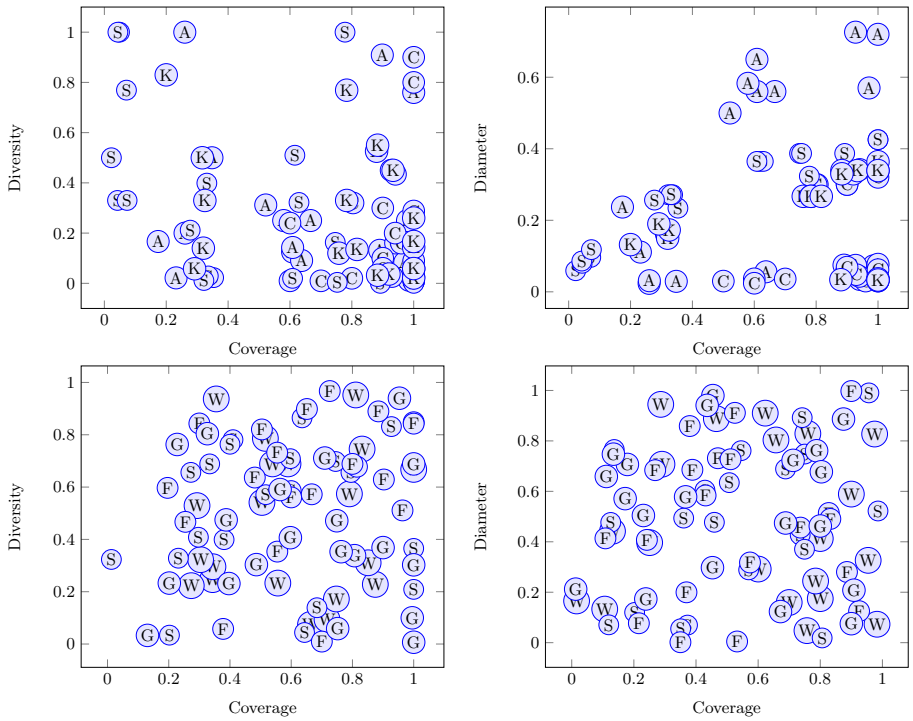

Fig. 5. Conflicting Objectives on MovieLens (top) and BookCrossing (bottom). Movie/Book title initials are illustrated on points.

5.2 Effect of Application-Defined Parameters

In this section, we examine the influence of different parameters of Algorithms 1 and 2. The parameters which are employed by h-MOMRI are number of intervals (*nbintervals*) and number of iterations (*nbiterations*). Also both algorithms employ two other parameters: minimum group size (σ) and maximum number of groups in a group-set (k). By default, we consider 10 intervals of diversity and 500 iterations for h-MOMRI and $\alpha = 1.5$ for α-MOMRI. For both algorithms, we consider $k = 5$ and we minimize diameter.

Minimum Group Size (σ) Not all combinations of attribute values can form a group. Because some combinations may not cover at least σ rating records. For instance, among rating records for *Toy Story* movie, there exists only 1 record which can be described by this label: $\langle$`male`, `young`, `lawyer`, `CA`$\rangle$. Thus for any $\sigma > 1$, this group would not be formed. In the first experiment, we illustrate the evolution of the number of groups by varying σ. Figure 6 illustrates the results for our 4 different sets of input ratings. The figure demonstrates a *long-tail* [7]: A few ratings are extremely frequent, but the majority of the dataset is composed of a large number of infrequent ratings. The long-tail transition is smoother in

case of MovieLens as it is denser, i.e., its average number of ratings per user is 4.14 times larger than BookCrossing. The long-tail reveals that choosing a fair value of σ is indeed challenging. In our experiments, we fix $\sigma = 10$ for both datasets, as this value is a border line between frequent ratings and the long tail.

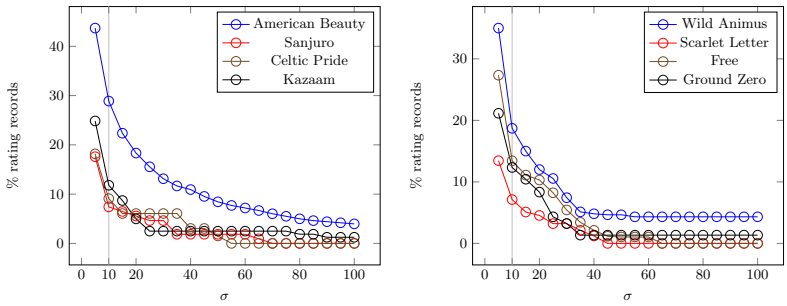

Fig. 6. Number of Groups as a Function of σ for MovieLens (left) and BookCrossing (right)

Number of Intervals and Iterations We examine the effect of other parameters on *execution time* and *number of solutions*. When there is more than one objective to optimize, there exists potentially many optimal solutions. Because those are incomparable (Example 4), it becomes tedious for an analyst to deal with thousands of solutions. On the other hand, a limited subset of these solutions may miss some interesting ones.

Figure 7 shows the effect of *nbintervals* on execution time and number of solutions. We vary *nbintervals* from 2 to 40. Obviously increasing *nbintervals* implies increasing result precision. However, we observe that it does not influence the size of the result space or the execution time. Each set of input ratings has almost a same value for all number of intervals. The order in which the values appear is in accordance with their number of input rating records. There exists different classes of values. For movies with less than 100 rating records, the execution time and the result space size are pretty similar. It is also the case for items with more than 1000 rating records (i.e., the movie *American Beauty*).

Figure 8 shows the effect of *nbiterations* on execution time and number of solutions. We vary *nbiterations* from 2 to 2000 to measure its effect on execution time and number of solutions. The hypothesis is that increasing the number of iterations leads to increasing the result space size. We observe that this hypothesis is only true when there is more than 1000 input rating records. In all other cases, the increase in number of solutions is negligible. Regarding the execution time, a linear behavior is observed which is far from being surprising.

Number of Returned Groups (k) Finally, we examine the effect of k on execution time and performance (Figure 9). We vary k from 2 to 10. In all sets

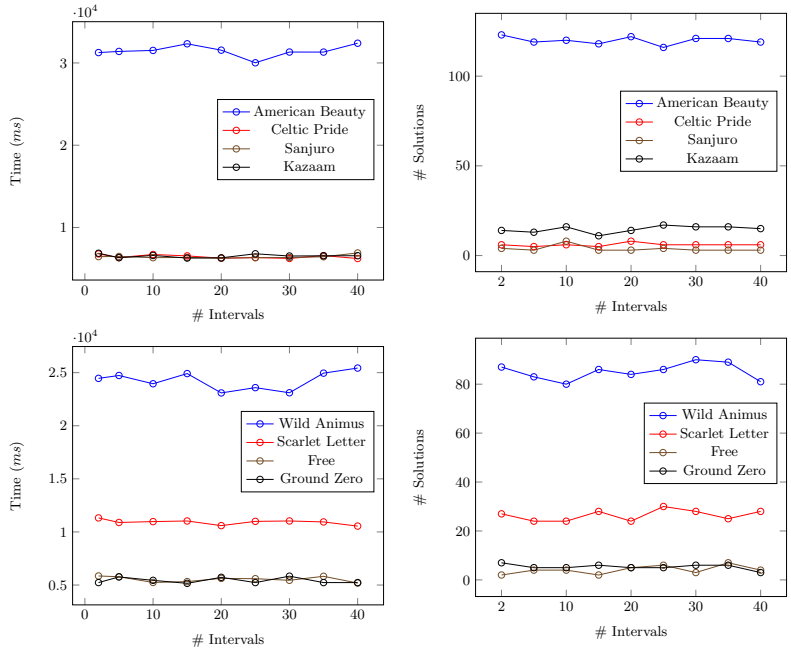

Fig. 7. Effect of *nbintervals* on Execution Time (left) and Result Space Size (right) for MovieLens (top) and BooCrossing

of input rating records, increasing k leads decreasing the size of the result space. Indeed, a bigger k means having bigger group-sets and less results. Nevertheless, when there are less than 1000 input rating records, the decrease is negligible. Same results hold for MovieLens.

5.3 Comparison of Algorithms

In this section, we compare h-MOMRI and α-MOMRI. Our hypothesis is that h-MOMRI has a manageable solution space size compared to α-MOMRI which leads to a reduced execution time.

First we compare the quality of algorithms regarding the dominance of solutions. In multi-objective optimization, if for two algorithms X and Y, the majority of X's solutions dominate Y's, it means that X is able to produce solutions with higher quality than Y. In this experiment, we make the same comparison between α-MOMRI and h-MOMRI. For this experiment, we need to compare each pair of α-MOMRI and h-MOMRI solutions. We count the number of times each algorithm dominates the other in pairwise comparison of their results. We consider $\alpha = 1.15$ for α-MOMRI and $nbintervals = 40$ for h-MOMRI. We denote the set of α-MOMRI solutions as $\mathcal{P}_\alpha$ and the set of h-MOMRI solutions as $\mathcal{P}_h$. We observe that for all sets of input rating records in Table 1, at least 62% of solutions in $\mathcal{P}_h$ are dominated by solutions in $\mathcal{P}_\alpha$. This is because α-MOMRI

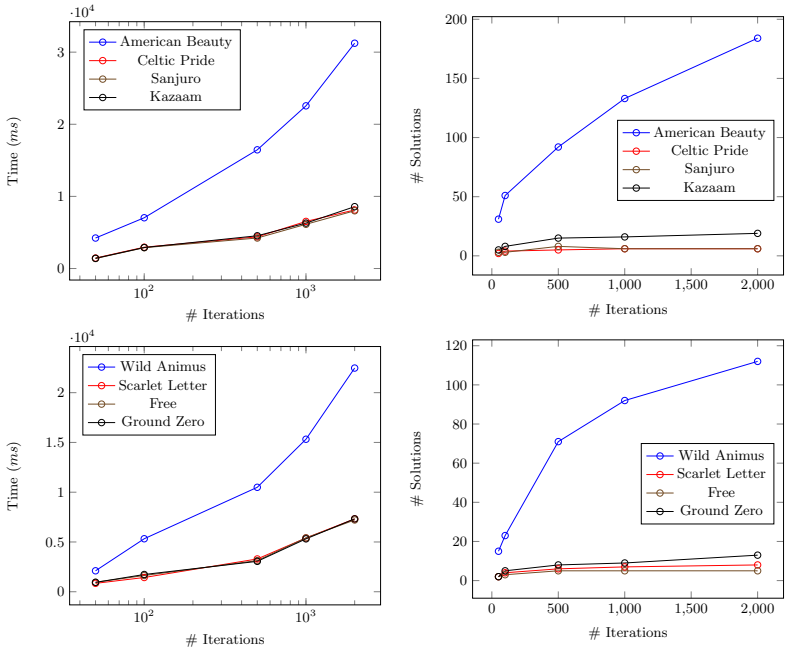

Fig. 8. Effect of *nbiterations* on Execution Time (left) and Result Space Size (right) for MovieLens (top) and BookCrossing (bottom)

generates the complete set of α-approximated Pareto plans, while h-MOMRI produces a subset. For instance, for the movie *American Beauty*, α-MOMRI produces 16 times more solutions than the heuristic algorithm. It is 14 times bigger for the book *Wild Animus*. Evidently the solutions in $\mathcal{P}_h$ are either as good as $\mathcal{P}_\alpha$'s or worse. Our results show that although α-MOMRI presents a huge set of all Pareto plans, h-MOMRI can return an acceptable representative subset where almost half of solutions are as good as the set $\mathcal{P}_\alpha$.

Concerning the huge difference in the size of solution sets, potentially a fairer comparison is to consider objective values to neutralize the influence of size. We observe that for all sets of input rating records in Table 1, h-MOMRI can achieve a supremacy over α-MOMRI in 39.4% of cases. This is a promising result for h-MOMRI which is in-line with our findings regarding the dominance comparison. We believe that the supremacy of h-MOMRI can be increased in two ways: *i.* by making a better balance of solution space size in each interval, and *ii.* by employing a more intelligent navigation mechanism for diversity and diameter as we do for coverage. We discuss the former in the next piece of experiments, while the latter is future work.

In the second comparative experiment, we analyze the distribution of solutions in h-MOMRI among diversity intervals. Note that the intervals have the same width. If the solutions are equally distributed among intervals, the probability of missing Pareto plans decreases. Because in this case, there exists enough

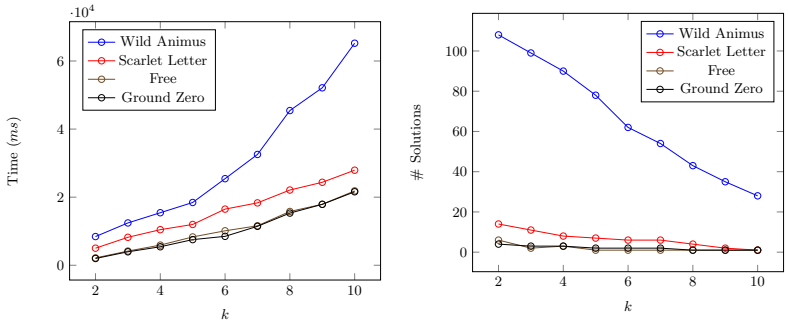

Fig. 9. Effect of the Number of Returned Groups (k) on Execution Time (left) and Result Space Size (right) for BookCrossing

instances in each interval which makes the probability of achieving Pareto plans statistically more powerful. For this experiment, we first observe a huge amount of empty intervals for most sets of input rating records. This is mainly because for some set of input rating records, the maximum possible diversity is not 1, but lower. In this case when we discretize diversity values into fixed-width intervals, many of them remain empty. Hence in this experiment, we discretize diversity values between zero and maximum possible diversity value for input rating records.

Figure 10 illustrates the results for different intervals and different sets of input rating records. The left chart illustrates the standard deviation for the number of solutions in intervals. If for a set of input rating records, all intervals contain the same number of solutions, then the standard deviation is equal to zero. Also, the right chart illustrates number of intervals with no solution, i.e., empty intervals.

We observe a high heterogeneity when $nbintervals < 10$ for all sets of input rating records and for both datasets. This means that by considering less than 10 intervals, we will potentially miss many Pareto plans. On the other hand, increasing the number of intervals leads to increasing the number of empty intervals which has the same consequence, i.e., missing Pareto plans. We then fix *nbintervals* to 10 as it exhibits the best tradeoff between heterogeneity and emptiness. This value of *nbintervals* increases the chance of discovering more Pareto plans in h-MOMRI, but as some amount of heterogeneity still remains even for $nbintervals > 10$, we cannot consider h-MOMRI as a safe replacement for α-MOMRI.

Now we compare α-MOMRI and h-MOMRI concerning their performance and the number of solutions they produce. We consider 3 different instances for each algorithm: for α-MOMRI, we consider instances with $\alpha = 2$ (A), $\alpha = 1.5$ (B) and $\alpha = 1.15$ (C), and for h-MOMRI, we consider instances with 5 (D), 10 (E) and 40 (F) intervals. We run this experiment with 4 items having the highest amount of rating records as items with fewer records exhibit similar behavior.

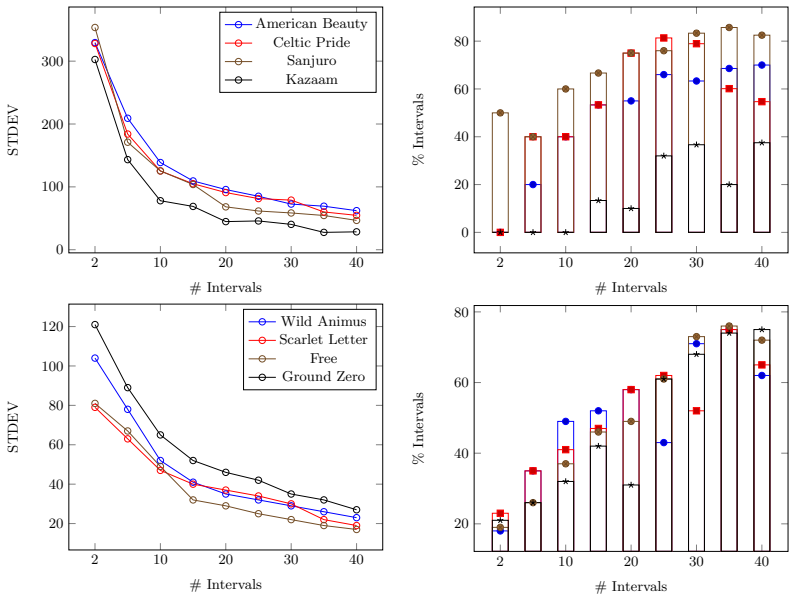

Fig. 10. Distribution of Solutions in Intervals in MovieLens (top) and BookCrossing (bottom)

Figure 11 illustrates the results. As expected, in general the number of solutions produced by h-MOMRI is one order of magnitude less than α-MOMRI in both datasets. In both algorithms, the number of ratings records play an important role and increases the the number of solutions.

5.4 Choosing between α-MOMRI and h-MOMRI

Both α-MOMRI and h-MOMRI are useful for analysts in different scenarios. α-MOMRI can be used in an *offline* context to produce an exhaustive set of user groups with a precision defined by α for further analysis. For instance, a movie rating website (like IMDb) can index user groups generated offline and execute various user queries like *'what are interesting groups of female teenagers who have rated romantic movies'*. On the other hand, in an *online* or *streaming* context, h-MOMRI is beneficial because it can immediately produce a representative subset of results. For instance, in a movie rating website an analyst can quickly observe interesting user groups of comedy and romantic movies.

6 Related Work

To the best of our knowledge, no approach has proposed and formalized the problem of discovering user groups for collaborative rating datasets by consid-

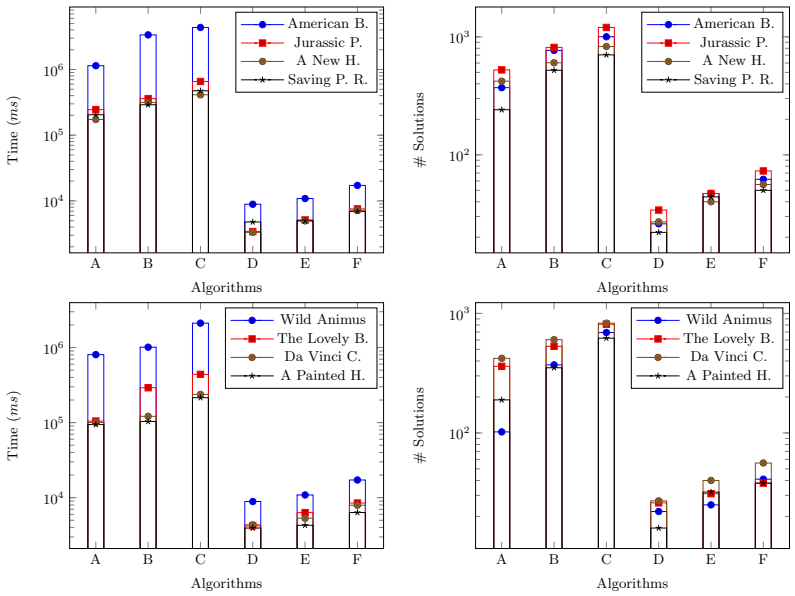

Fig. 11. Comparison of α-MOMRI and h-MOMRI Algorithms in Execution Time (left) and # Solutions (right) on MovieLens (top) and BookCrossing (bottom)

ering multiple *independent* and *conflicting* quality dimensions. Recent studies[8] have shown an interest in reporting statistics about pre-defined groups, as opposed to our work where we look to discover high-quality user groups on the fly. However our work does relate to a number of others in its aim and optimization mechanism.

There exist different approaches to solve a multi-objective problem [14, 15]. We already discussed that Scalarization does not work in our case (Section 5.1). Another popular method is ϵ-constraints [12] where one objective is optimized and others are considered as constraints. The approach in [4] can be seen as a relaxed ϵ-constraints version of our problem.

Another approach is Multi-Level Optimization [11] which needs a meaningful hierarchy between objectives. In our case, all objectives are independent and conflicting, hence using this mechanism is not feasible. In this work, we focused on [5, 14] mainly because of their recency and the adequacy of their data model to our problem.

User groups can be discovered by clustering methods [1–3, 9] where a single objective is optimized. Multi-Objective clustering [8, 10] is an improvement where clusters are obtained from n different clustering algorithms. This guarantees clusters with high quality in multiple dimensions. This is a two-step approach where *i*. each clustering algorithm, applied to one quality dimensionin our case, generates its own set of clusters, *ii*. a *goodness* measure picks target

[8] *http://blog.testmunk.com/how-teens-really-use-apps/*

clusters by combining results of all algorithms. However, the definition of a goodness measure is subjective and does not guarantee that all desired objectives are optimized.

MOMRI scans data only once as the pruning technique in α-MOMRI considers all objectives at the same time and ddetermines if a candidate group-set should or not be kept for further comparisons. On the other hand, another challenge of clustering which has received less attention, is *information overload*, i.e., on real data, there exists usually millions of clusters which make the analysis tedious. Using h-MOMRI, the analyst receives a manageable subset of high quality results in a reasonbale time. More (precise) results are returned by reducing α for α-MOMRI or increasing *nbintervals* for h-MOMRI.

7 Conclusion and Future Work

In this paper, we investigated the question of finding the best group-sets that characterize a database of rating records of the form $\langle i, u, s\rangle$, where $i \in \mathcal{I}$, $u \in \mathcal{U}$, and s is the integer rating that user u has assigned to item i. We showed that the problem of finding high-quality group-sets is NP-Complete and proposed a constrained Multi-Objective formulation. Our formulation incorporates local and global group desiderata. We proposed two algorithms that find group-sets as instances of Pareto plans. The first one α-MOMRI, is an α-approximation algorithm and the second, h-MOMRI, is a heuristic-based algorithm. Our extensive experiments on MovieLens and BookCrossing datasets show that our approximation finds high quality groups and that our heuristic is very fast. without compromising quality.

Our work can be improved in many ways. In particular, we plan to perform an extensive user study to be able to evaluate the quality of returned group-sets. An online poll (about movies or books) could be used to build a ground-truth and will be used to evaluate the usefulness of our group-sets. Also, we plan to investigate an extensive analysis of rating distributions for our algorithms using some dispersion measures.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.
2. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD*, 1993.
3. B. Amiri, L. Hossain, and J. Crowford. A multiobjective hybrid evolutionary algorithm for clustering in social networks. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, 2012.
4. M. Das, S. Amer-Yahia, G. Das, and C. Yu. Mri: Meaningful interpretations of collaborative ratings. *VLDB*, 2011.
5. P.-F. Dutot, K. Rzadca, E. Saule, and D. Trystram. *Multi-objective scheduling*, chapter 9. Chapman and Hall/CRC Press, 2009.

6. S. Ganguly, W. Hasan, and R. Krishnamurthy. *Query optimization for parallel execution*, volume 21. ACM, 1992.
7. S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. In *WSDM*, 2010.
8. R. Jiamthapthaksin, C. F. Eick, and R. Vilalta. A framework for multi-objective clustering and its application to co-location mining. In *Advanced Data Mining and Applications*, pages 188–199. Springer, 2009.
9. M. Kargar, A. An, and M. Zihayat. Efficient bi-objective team formation in social networks. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2012.
10. M. H. Law, A. P. Topchy, and A. K. Jain. Multiobjective data clustering. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–424. IEEE, 2004.
11. A. Migdalas, P. M. Pardalos, and P. Värbrand. *Multilevel optimization: algorithms and applications*, volume 20. Springer Science & Business Media, 1997.
12. C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS*, 2000.
13. S. J. Russell and P. Norvig. Probabilistic reasoning. *Artificial intelligence: a modern approach*, 2003.
14. I. Trummer and C. Koch. Approximation schemes for many-objective query optimization. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014.
15. G. Tsaggouris and C. Zaroliagis. Multiobjective optimization: Improved fptas for shortest paths and non-linear objectives with applications. *Theory of Computing Systems*, 45(1):162–186, 2009.

8 Appendix: Optimality and Near-Optimality Proofs

In all of the following theorems, we consider two group-sets G and G' and two sub group-sets G_1, G_2 for G and G'_1, G'_2 for G' such that:

- $G_1 \cup G_2 = G$, $G_1 \cap G_2 = \emptyset$;
- $G'_1 \cup G'_2 = G'$, $G'_1 \cap G'_2 = \emptyset$;
- $|G| = |G'|$, $|G_1| = |G'_1|$ and $|G_2| = |G'_2|$;
- User groups in $\{G, G', G_1, G_2, G'_1, G'_2\}$ are distinct, i.e., each group cannot appear more than once in a group-set;
- $\forall g_1 \in G_1 \wedge g_2 \in G_2, g_1 \nsubseteq g_2 \wedge g_2 \nsubseteq g_1$;
- $\Sigma_{g_1 \in G_1, g_2 \in G_2} |g_1 \cap g_2| \leq \Sigma_{g'_1 \in G'_1, g'_2 \in G'_2} |g'_1 \cap g'_2|$.

8.1 Optimality Proofs

Theorem 2. *Coverage (Equation 1) satisfies POO.*

Proof. Given the fact that:

- $coverage(G, R)$ is a monotone function;
- $coverage(G_1 \cup G_2, R) = coverage(G, R)$;

Then the left part of the POO implication $coverage(G_1, R) \geq coverage(G'_1, R) \wedge coverage(G_2, R) \geq coverage(G'_2, R) \rightarrow coverage(G, R) \geq coverage(G', R)$ can be transformed to $coverage(G_1 \cup G_2, R) \geq coverage(G'_1 \cup G'_2, R)$ and then $coverage(G, R) \geq coverage(G', R)$, i.e., the right part of the implication, hence, the proof. □

Theorem 3. *Diversity (Equation 2) satisfies POO.*

Proof. In Equation 2, the compoenent $(\Sigma_{g,g' \in G} | r \in R, r \lessdot g \wedge r \lessdot g'|)$ computes the amount of overlap. We use the notation ov_G to denote this component. Thus $diversity(G, R) = 1/(1 + ov_G)$. Obviously whenever ov_G increases, $diversity(G, R)$ decreases. Thus we transform the POO implication to $ov_{G_1} \leq ov_{G'_1} \wedge ov_{G_2} \leq ov_{G'_2} \rightarrow ov_G \leq ov_{G'}$. It is obvious that larger overlaps in G'_1 and G'_2 compared to G_1 and G_2 lead a larger overlap in G' compared to G. It is true only if there is no overlap between sub group-sets. □

Theorem 4. *Diameter (Equation 3) satisfies POO.*

Proof. We consider *homogeneity* in this proof. The extension to polarization is straightforward. For simplicity, we convert the formulation to the following form: $diameter(G) = avg_{g \in G}(diff(g))$.

Step 1. The left part of the POO implication $diameter(G_1, R) \leq diameter(G'_1, R) \wedge diameter(G_2, R) \leq diameter(G'_2, R)$ is then equal to $avg_{g \in G_1}(diff(g)) \leq avg_{g \in G'_1}(diff(g)) \wedge avg_{g \in G_2}(diff(g)) \leq avg_{g \in G'_2}(diff(g))$. It can be transformed to $(avg_{g \in G_1}(diff(g)) \times |G_1|) \leq (avg_{g \in G'_1}(diff(g)) \times |G_1|) \wedge (avg_{g \in G_2}(diff(g)) \times$

$|G_2|) \leq (avg_{g \in G'_2}(\mathit{diff}(g)) \times |G_2|)$ (i.e., multiplying a constraint to both parts of inequalities).

Step 2. As *summation* is a monotone function, we merge two parts of the conjunction to obtain the following: $(avg_{g \in G_1}(\mathit{diff}(g)) \times |G_1|) + (avg_{g \in G_2}(\mathit{diff}(g)) \times |G_2|) \leq (avg_{g \in G'_1}(\mathit{diff}(g)) \times |G_1|) + (avg_{g \in G'_2}(\mathit{diff}(g)) \times |G_2|)$ and then $((avg_{g \in G_1}(\mathit{diff}(g)) \times |G_1|)/|G|) + ((avg_{g \in G_2}(\mathit{diff}(g)) \times |G_2|)/|G|) \leq ((avg_{g \in G'_1}(\mathit{diff}(g)) \times |G_1|)/|G|) + ((avg_{g \in G'_2}(\mathit{diff}(g)) \times |G_2|)/|G|)$ (i.e., dividing the whole inequality by $|G|$).

Step 3. Recall $|G| = |G'|$ then $((avg_{g \in G_1}(\mathit{diff}(g)) \times |G_1|)/|G|) + ((avg_{g \in G_2}(\mathit{diff}(g)) \times |G_2|)/|G|) \leq ((avg_{g \in G'_1}(\mathit{diff}(g)) \times |G_1|)/|G'|) + ((avg_{g \in G'_2}(\mathit{diff}(g)) \times |G_2|)/|G'|)$. Based on the definition of average function, the expression is equal to $avg_{g \in G}(\mathit{diff}(g) \leq avg_{g \in G'}(\mathit{diff}(g))$, i.e., the right part of the formula, hence the proof. □

8.2 Optimality Proofs

Theorem 5. *Coverage (Equation 1) satisfies PONO.*

Proof. Given the same facts in the proof of Theorem 2, the left part of the PONO implication can be transformed to $\mathit{coverage}(G_1 \cup G_2, R) \geq \mathit{coverage}(G'_1 \cup G'_2, R) \times \alpha$ and then $\mathit{coverage}(G, R) \geq \mathit{coverage}(G', R) \times \alpha$, i.e., the right part of the implication, hence, the proof. □

Theorem 6. *Diversity (Equation 2) satisfies PONO.*

Proof. We reuse the notation we introduced in the proof of Theorem 3 and define $\mathit{diversity}(G, R) = 1/(1+ov_G)$. The PONO implication for diversity based on ov_G is $ov_{G_1} < (ov_{G'_1} \times \alpha) \wedge ov_{G_2} < (ov_{G'_2} \times \alpha) \rightarrow ov_G < ov_{G'} \times \alpha$. As *summation* is a monotone function, then we can transform the left part of the implication to $ov_{G_1} + ov_{G_2} < (ov_{G'_1} \times \alpha) + (ov_{G'_2} \times \alpha) \Rightarrow ov_{G_1} + ov_{G_2} < \alpha \times (ov_{G'_1} + ov_{G'_2})$ $\Rightarrow \mathit{diversity}(G_1, R) + \mathit{diversity}(G_2, R) \geq \alpha(\mathit{diversity}(G'_1, R) + \mathit{diversity}(G'_2, R))$ $\Rightarrow \mathit{diversity}(G, R) \geq \mathit{diversity}(G', R) \times \alpha$, hence the proof. □

Theorem 7. *Diameter (Equation 3) satisfies PONO.*

Proof. We consider *homogeneity* in this proof. The extension to polarization is straightforward. For simplicity, we convert the formulation to the following form: $\mathit{diameter}(G) = avg_{g \in G}(\mathit{diff}(g))$.

Step 1. The left part of the PONO implication $\mathit{diameter}(G_1, R) \leq (\mathit{diameter}(G'_1, R) \times \alpha) \wedge \mathit{diameter}(G_2, R) \leq (\mathit{diameter}(G'_2, R) \times \alpha)$ is then equal to $avg_{g \in G_1}(\mathit{diff}(g)) \leq (avg_{g \in G'_1}(\mathit{diff}(g)) \times \alpha) \wedge avg_{g \in G_2}(\mathit{diff}(g)) \leq (avg_{g \in G'_2}(\mathit{diff}(g)) \times \alpha)$. It can be transformed to $(avg_{g \in G_1}(\mathit{diff}(g)) \times |G_1|) \leq (avg_{g \in G'_1}(\mathit{diff}(g)) \times |G_1| \times \alpha) \wedge (avg_{g \in G_2}(\mathit{diff}(g)) \times |G_2|) \leq (avg_{g \in G'_2}(\mathit{diff}(g)) \times |G_2| \times \alpha)$.

Step 2. As *summation* is a monotone function, we merge two parts of the conjunction to obtain the following: $(avg_{g\in G_1}(\mathit{diff}(g))\times|G_1|)+(avg_{g\in G_2}(\mathit{diff}(g))\times|G_2|) \leq (avg_{g\in G'_1}(\mathit{diff}(g)) \times |G_1| \times \alpha) + (avg_{g\in G'_2}(\mathit{diff}(g)) \times |G_2| \times \alpha)$ and then $((avg_{g\in G_1}(\mathit{diff}(g)) \times |G_1|) /|G|) + ((avg_{g\in G_2} (\mathit{diff}(g)) \times |G_2|) /|G|) \leq (\alpha \times ((avg_{g\in G'_1} (\mathit{diff}(g)) \times |G_1|) /|G|)+ ((avg_{g\in G'_2} (\mathit{diff}(g)) \times |G_2|)) /|G|)$.

Step 3. Recall $|G| = |G'|$ then $((avg_{g\in G_1} (\mathit{diff}(g)) \times |G_1|) /|G|) + ((avg_{g\in G_2} (\mathit{diff}(g)) \times |G_2|) /|G|) \leq (\alpha\times((avg_{g\in G'_1} (\mathit{diff}(g)) \times |G_1|) + (avg_{g\in G'_2}(\mathit{diff}(g))\times |G_2|)) /|G'|)$. Based on the definition of average function, the expression is equal to $avg_{g\in G} (\mathit{diff}(g) \leq \alpha \times avg_{g\in G'} (\mathit{diff}(g))$, i..e, the right part of the formula, hence the proof. □