Building User Profiles to Improve User Experience in Recommender Systems

Anisio Lacerda
Univ. Federal de Minas Gerais, Brazil
and Zunnit Technologies, Brazil
anisio@dcc.ufmg.br

Advisor: Nivio Ziviani
Univ. Federal de Minas Gerais, Brazil
nivio@dcc.ufmg.br

ABSTRACT

Recommender systems are quickly becoming ubiquitous in many Web applications, including e-commerce, social media channels, content providers, among others. These systems act as an enabling mechanism designed to overcome the information overload problem by improving browsing and consumption experience. Crucial to the performance of a recommender system is the accuracy of the user profiles used to represent the interests of the users. In this proposal, we analyze three different aspects of user profiling: (i) selecting the most informative events from the interaction between users and the system, (ii) combining different recommendation algorithms to (iii) including trust-aware information in user profiles to improve the accuracy of recommender systems.

General Terms

Algorithm, Performance, Experimentation, Economics

Keywords

Recommender Systems, Novelty, Diversity

1. INTRODUCTION

Recommender systems are mechanisms devoted to overcoming problems that are inherent to information overload, providing intelligent information access and delivery, potentially improving browsing and consumption experience [10]. Recommender systems benefit from a detailed representation of the users' interests to personalize the results and improve experience. In order to understand the preferences of users, recommender systems require a proper *representation* of their interests and intents.

In this work we focus on generating compact and effective user profiles from user past behavior in recommender systems scenarios. The task of building the right profile can be divided in two phases: (i) finding the appropriate descriptive features and (ii) designing models to explore these features to achieve the systems goal. We model the input data for recommender systems as interactions between entities. The generated data are pairs of elements coming from two sets of objects called dyads [5]. Formally, the data consists of dyads $\{(r_i, c_i)\}_{i=1}^n$ (i.e., pairs of objects) with associated labels $\{y_i\}_{i=1}^n$.

WSDM'13, February 4–8, 2013, Rome, Italy.

The dyadic prediction task consists in determine labels for unobserved dyads, that is for pairs (r', c'). Given that we are interested on the interaction between users and items in recommender systems, the dyads are composed of pairs (users, items). A side-information is additional evidence for each member of a dyad. Thus, when considering a movie recommender system, age and gender are examples of side-information about users; when considering items, title and category of the movie are examples of side-information about items. For the same movie recommender system, the set of ratings for a movie is an example of labels.

An important problem related to the absence of side-information is known as cold-start problem, where the test set contains a dyad (r', c') where at least one of r' or c' is not present in any training set dyad. In recommender systems, it occurs when the system needs to predict how a user will rate a movie that is yet to be released, and thus has no existing ratings. In the following sections, we present some initial results on using the category of items to alleviate this problem.

Normally, Web dyadic data shares the following aspects [5]:

- High-dimensional: The two involved sets are often very large, e.g., the set of distinct users and all movies for (user, movie) dyadic data.
- Sparse: Measurements are sparse relative to all possible dyads, e.g., a user does not rate all movies.
- Dynamic: Web dyadic data keeps growing every single second in terms of both the observed dyads and the dimensionality, e.g., new users join movie recommender systems and rate movies they are interested all the time.

In order to alleviate these issues, we divided the task of building the user profile into two phases: (i) weighting accordingly the most descriptive dyads and (ii) designing models to explore these dyads to achieve the systems goal. We will also consider the list of items returned when extracting user preferences.

We will consider a simple method for user profile updating based on their age in the system, as we know this feature has a great influence in the user behaviour. According to [7], in recommender systems, new users have different needs from experienced users. New users may benefit from methods that generates highly ratable items, as they need to establish trust and rapport with a recommender before taking advantage of the recommendations it offers. Hence, the system needs to be able to evolve user profiles according to their needs as they age in the system.

2. OBJECTIVES

There are several tasks and challenges associated with modern recommender systems. A typical task can be stated as follows: given a set of items that are known in advance to be associated with a user (i.e., past preference data), the system must return a ranked list of suggested items, so that more interesting items appear in the top of the rank [10].

The notion of interestingness, however, may be subjective, encompassing a broad repertoire of measures, including: (i) accuracy (how well the suggested items meet the user's information need), (ii) novelty (how well the system promotes unknown items to the users), and (iii) diversity (how different the suggested items are with respect to each other). In order to find a proper balance between these metrics, we intent to: (i) select the most informativeness dyads to build the user profile, and (ii) use a linear combination of the relevance estimated by n different existing recommendation algorithms.

Recommender Systems based only on dyadic data (Collaborative Filtering – CF) rely on a simple intuition: items appreciated by people similar to someone will also be appreciated by that user. The process of comparing two users involves comparing the preferences they provided to items, which can be compromised by high-dimensional and sparse properties of the dyadic data. Our intent is to search for similar users as CF does but to search for trustable users by exploiting the social graph and incorporated this information on user profiles.

Following we summarize our research questions:

1. How can we select the most informative events to build an descriptive user profile?
2. How can we combine different algorithms to balance accuracy, novelty, and diversity?
3. How can we incorporate trust-aware network metrics in user profiles?

3. METHODOLOGY

In Table 1, we present the properties of the datasets that we will use in this research. For movie recommendation, we will use the MovieLens dataset [8]. As representative of trust-aware recommender systems we will use the Epinions dataset [6] and the Flixster dataset [13]. For music recommendation, we will use the Lastfm with an implicit preference dataset from [2].

The testing methodology we adopted in this paper is similar to the one described in [3], which is appropriate for the top-N recommendation task. For the measures of precision, recall and F1 we will use the definitions presented in [1]. In the case of implicit feedback (i.e., Last.fm), we normalized the observed item access frequencies of each user to a common rating scale [0,5], as used in [11].

In order to measure the novelty of the recommendations, we will use a popularity-based item novelty model proposed in [11], so that the probability of an item i being seen is estimated as:

$$P(seen|i_k) = \frac{|u \in U | r(u,i) \neq \emptyset|}{|U|} \quad (1)$$

where U denotes the set of users. Since the testing methodology supposes that most of the 1,000 additional unrated items are not relevant, we used the metrics in the framework proposed in [11] without relevance awareness. The novelty of a top-N recommendation list from R presented to user u is therefore given by:

$$nov(R(N)) = EPC(N) = C \sum_{i_k \in R}^{i_N} rd(k)(1 - p(seen|i_k)) \quad (2)$$

where $rd(k)$ is a rank discount given by $rd(k) = 0.85^{k-1}$[11] and C is a normalizing constant given by $1/\sum_{i_k \in R}^{i_N} rd(k)$. Therefore, this metric is rank-sensitive (i.e. the novelty of the top-rated items counts more than the novelty of other items). As is the case with precision and recall, we average the EPC@N value of the top-N recommendation lists over the test set.

We used a distance based model in order to measure the diversity of the recommendation lists. Once again, we used the metrics from [11] without relevance-awareness. The recommendation diversity, therefore, is given by:

$$div(R(N)) = EILD(N) = \sum_{\substack{i_k \in R \\ i_l \in R \\ l \neq k}}^{i_N, l_N} C_k rd(k) rd(l|k) d(i_k, i_l) \quad (3)$$

where $rd(l|k) = rd(max(1, l-k))$ reflects a relative rank discount between l and k, and $d(i_k, i_l)$ is the cosine similarity between two items, given by:

$$d(i,j) = \frac{|\mathbf{U_i} \cap \mathbf{U_j}|}{\sqrt{|\mathbf{U_i}|}\sqrt{|\mathbf{U_j}|}} \quad (4)$$

such that $\mathbf{U_i}$ denotes the users that liked item i, and $\mathbf{U_j}$ denotes the users that liked item j.

4. ANSWERING THE RESEARCH QUESTIONS

In Figure 1, we present an overview of the user profiling process, which is divided into three different steps. First, we investigate approaches to select the most informative instances to build the user profile. Second, we are interested on strategies to combine the informative selected instances and other evidence to match the most relevant items to users. Third, given that the user preferences changes over time, we investigate methods to reflect these changes.

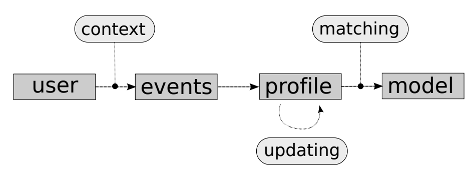

Figure 1: Overview of the user profiling process

4.1 Using Context to Select the Most Informative Events

In this section, we investigate the user context to select the most informative training instances. We start by discussing how different ranked lists are produced for each user, so that each one is particularly effective with respect to a specific interestingness measure (i.e., accuracy, diversity, or novelty). Then, we discuss a future direction to aggregate the multiple ranked lists that were produced, resulting in a final ranked list.

4.1.1 Related Work

In [2], the authors introduced the notion of "topic diversification", which ensures diversity by balancing suggestions across different topics. In terms of novelty, we adopt a metric that is closer to the metric proposed in [3], which assumes that less popular items are less likely to be widely known by users, and thus a penalty is imposed depending on the frequency or popularity of the suggested items.

Table 1: Statistics of Datasets.

	Datasets			
	Movielens	Epinions	Flixster	Lastfm
# users	6,040	49,290	1,000,000	992
# items	3,883	139,738	49,000	1,300,000
# preferences	1,000,209	664,824	8,200,000	19,150,868
Trust statements	–	487,181	26,700,000	–
Rating range	1 - 5	1 - 5	0.5 - 5.0	–
Feedback	Explicit	Explicit	Explicit	Implicit

4.1.2 Learning Optimal Ranked Lists for Multi-Objective Recommender Systems

The first step of our approach consists in performing consecutive queries over a large set of events E, where each event $x \in E$ consists of a set of (user, item) dyads and the respective labels. Each query selects an event in E exhibiting a property P which is related to a specific interestingness measure M. The selected event is appended into the training-set D, which represents the user profile, and a classifier is updated accordingly. This querying procedure iterates selecting examples that exhibit the desired property and updating the classifier, until a desired level of performance is achieved.

We used three different query strategies to select examples from E: (i) *accuracy-oriented*: the event selected is the one that the classifier is more uncertainty about, *diversity-oriented*: selects examples from E that are not redundant with respect to events already included into D, and *novelty-oriented*: selects events from E for which the corresponding topk items are less popular.

The baselines used were a naive recommendation approach ("most popular") as a lower bound, and we expect that all proposed approaches outperform this naive approach in all evaluation metrics considered. Further, we also employ a strong baseline, WRMF [4], which is a state-of-the-art weighted matrix factorization approach. We conducted five-fold cross validation, and the results reported for each evaluated recommendation approach correspond to the av- erage of the five trials. Statistical significance tests were performed using a 2–sided paired t–test with p$-$value < 0.05.

In Table 2 we show the results obtained by each evaluated recommendation approach in the MovieLens subsets. As expected, suggesting only the most popular items leads to poor recommendation performance in all evaluation metrics. In contrast, WRMF was the best performer in terms of precision. It outperformed our accuracy-oriented approach, with differences that are up to 7.6%. Closer recommendation performance was obtained only at p@10 and for subsets composed of moderate and large number of tags per object.

On the other hand, WRMF performed extremely poor in terms of diversity and novelty. This trend holds for all three subsets. As expected, our diversity-oriented approach was the best performer in terms of diversity, and our novelty-oriented approach was the best performer in terms of novelty. Also, the precision achieved by our diversity- and novelty-oriented approaches was much higher than the precision achieved by the naive ("most popular") approach.

4.2 Combining Different Algorithms to Balance Accuracy, Novelty, and Diversity

In this section, we investigate the hypothesize that it is possible to properly aggregate different recommendation algorithms, so that the resulting hybrids balances the level of accuracy, diversity and novelty in its suggestions.

4.2.1 Related Work

Historically, the typical goal of a recommender system is to maximize accuracy as much as possible in predicting and matching user information needs, often by considering individual delivered items in isolation [10]. More recently, however, it has become a consensus that the success of a recommender system depends on other dimensions of information utility, notably the diversity and novelty of the suggestions performed by the system [10]. More specifically, even being accurate, obvious and monotonous recommendations are generally of little use, since they do not expose users to unprecedent experiences. In the rest of this section we present previous related work, detail the current state of our work[1], and finally discuss the ongoing research.

4.2.2 Pareto-Eficient Hybrid Algorithm to Combine Recommender Systems

In order to provide suggestions that accounts for accuracy, diversity, and novelty, we investigate a hybrid recommendation approach that combines well-established recommendation algorithms (e.g., simple algorithms as well as representative of the state-of-the-art). Our proposed hybridization approach consists in finding appropriate weights for the constituent algorithms. By considering each dimension (i.e., accuracy, novelty and diversity) as a separate objective, we reduce the hybridization task to a multi-objective optimization problem, in which we search for the optimal combination of weights that maximizes accuracy, diversity and novelty.

Since the considered objectives are potentially conflicting, we employ an evolutionary search for optimal hybrids. Evolutionary algorithms denote a class of optimization methods that are characterized by a set of candidate solutions (aka individuals) called a population, which is maintained during the entire optimization process. The population of individuals evolves towards better (and potentially optimal) solutions by employing genetic operators, such as reproduction, mutation and crossover. In our context, each individual represents a possible combination of weights (i.e., a possible hybrid).

Optimal hybrids lie in the so-called Pareto frontier [15], and are optimal in the sense that no hybrid in the frontier can be improved upon without hurting at least one of its objectives. Therefore, the evolutionary algorithm evolves the population towards producing hybrids that are located closer to the Pareto frontier, and then a linear search returns

[1]A detailed version of the current state is presented in [9].

Table 2: Recommendation effectiveness for different subsets of the MovieLens dataset.

Approach	Precision			Diversity			Novelty		
	p@3	p@5	p@10	d@3	d@5	d@10	n@3	n@5	n@10
most popular	.244	.212	.169	.350	.365	.391	.026	.071	.232
WRMF	.352	.319	.241	.377	.392	.413	.290	.391	.545
accuracy-oriented	.332	.307	.226	.418	.434	.451	.332	.449	.607
diversity-oriented	.319	.304	.217	.442	.455	.475	.340	.455	.614
novelty-oriented	.274	.238	.196	.422	.451	.473	.358	.469	.629

the most dominant hybrid [15], which is likely to balance accuracy, novelty and diversity.

We conducted a systematic evaluation involving different recommendation scenarios, with explicit user feedback (i.e., movies from the MovieLens dataset). We selected eight recommendation algorithms to provide the base for our hybrids. To represent latent factor models, we selected PureSVD with 50 and 150 factors (PureSVD50 and PureSVD150) [3]. As a baseline for constituent method we used "MostPopular". As hybrid baseline, we used a voting-based hybrid based on Borda-Count (BC) and STREAM as baseline, a stacking-based approach with additional meta-features.

In Table 3 we show the accuracy results (recall and precision) over different values of N. Since both EPC(novelty) and EILD(diversity) are rank-sensitive metrics, we only presented their values for N = 20. There is a clear compromise between accuracy, novelty and diversity of these algorithms. The constituent algorithm that provides the most accurate recommendations is PureSVD50. TopPop provided the most diverse recommendations, although it performs significantly worse in accuracy and novelty. Regarding the performance of the baselines in the MovieLens dataset, STREAM performs worse then PureSVD50 on accuracy, maintaining the same level of novelty and performing better in terms of diversity. Borda Count performed poorly on accuracy and reasonably well in terms of novelty and diversity.

We compared PO-acc (Pareto-Optimal Accuracy) with PureSVD50, which is the stand-alone algorithm with the most accurate recommendations. PO-acc performs much better on diversity (and equally well on novelty). We compared PO-nov (Pareto-Optimal Novelty) with Pure-SVD150, which presented the most novel recommendations to the users, with reasonable accuracy. PO-nov performs slightly better on novelty than PureSVD150, but performs much better in terms of accuracy, and slightly on diversity. Finally, we compared PO-div (Pareto-Optimal Diversity) with MostPopular, the algorithm with the most diverse recommendations. PO-div loses very slightly on diversity, while improving on accuracy and novelty.

Hybrids in the Pareto frontier can be selected according to a certain need, allowing the recommender system to adjust the compromise between accuracy, novelty and diversity, so that the recommendation emphasis can be adapted dynamically according to the needs of each user (i.e., new users may benefit more from more accurate suggestions, whereas older users may require more novel and diversified suggestions). The ongoing research on these topic is to determine the exact moment during the user life-time usage of recommender system that is more appropriate to provide more diversified or novel items. In other words, we are interested on precisely identify the moment that users are more open to receive diversified results.

4.3 Incorporating Trust-Aware Metrics in User Profiles

In this section we present our proposal for enhancing Recommender Systems by use of trust information in the Slope One algorithm. In Section 1, we mentioned that web dyadic data shares some important aspects, high-dimensionality, sparsity, and dynamics. In other words, in popular domains such as movies, the number of items is very large while the number of items rated by every single user is in general small. This means that it is very unlikely two random users have rated any items in common and hence they are not comparable. In order to alleviate this problem we investigate a method to incorporate trust information into the user profiles. The rest of this section is organized as follows, first we briefly present the Slope One Algorithm and an extension known as Weighted Slope One. Then we show the current state of our research and discuss the results. Finally, we present the ongoing research path that we intent to follow.

4.3.1 Related Work

There are also other variants of the Slope One algorithm in the literature. For example, in [12], the authors propose an algorithm based on Slope One and user-based collaborative filtering to improve the recommendation performance of the second one. This approach addresses the issue by using the Slope One to fill the missing ratings and then using the user-based collaborative filtering to produce recommendations. Zhang [14] applies the Slope One algorithm in the same way as[12]. The difference between both approaches is that the first one uses an item-based collaborative algorithm to perform the recommendations.

4.3.2 A Collaborative Filtering Method Using User Trust-Aware Evidences in Slope One Algorithm

In order to incorporate trust information in the user profiles, we investigate an appropriate method to extend the Slope One, which is a family of algorithms for collaborative filtering, introduced in [12]. The Slope One basic principle is of "popularity differencial" between users and items. In a pairwise fashion, it determines how much better one item is liked than another. In order to measure this differential it subtracts the average rating of the two items. In turn, this difference can be used to predict another userâĂŹs rating of one of those items, given their rating of the other. We choose the Slope One algorithm because it is easy to extend and provide competitive suggestions.

One of the drawbacks of Slope One is that the number of ratings observed is not taken into consideration. Thus, we proposes variants of the Weighted Slope One, which performs predictions based on the items already rated by the user and these items are always equally considered. Specifically, we investigate the use of a measure of personalized item prediction efficiency for users, which is named as item

Table 3: Results for Recommendation Algorithms on the MovieLens dataset, with the three objectives (i.e., accuracy, novelty, and diversity). The recommender methods variants are grouped into: (i) constituent algorithms, (ii) hybrid baselines, and (iii) our proposed hybrids. We used the symbols: †, •, ⋄ to point out our method and the respective baseline. For instance, PSVD150 is the baseline with respect to the selected PO-nov individual. For each group, the best results for each metric are in bold. Underlined values means that the selected individual and the respective baseline are statistically different (95%).

		Accuracy								Novelty	Diversity
	Algorithm	R@1	R@5	R@10	R@20	P@1	P@5	P@10	P@20	EPC@20	EILD@20
Consts.	PSVD50 †	**0.1900**	**0.4155**	**0.5402**	**0.6643**	**0.1900**	**0.0831**	**0.0540**	**0.0332**	0.8070	0.1667
	PSVD150 •	0.1237	0.3203	0.4450	0.5658	0.1237	0.0641	0.0445	0.0283	0.8519	0.1375
	TopPop ⋄	0.0722	0.2061	0.2895	0.3994	0.0722	0.0412	0.0289	0.0200	0.7079	**0.2598**
Baselines	STREAM	**0.1792**	**0.3961**	**0.5169**	**0.6426**	**0.1792**	**0.0792**	**0.0517**	**0.0321**	0.8078	0.1914
	BC	0.0473	0.1657	0.2639	0.4352	0.0473	0.0331	0.0264	0.0218	**0.8210**	0.1609
	EW	0.1562	0.3574	0.4752	0.5980	0.1562	0.0715	0.0475	0.0299	0.7441	**0.2284**
Our Hybs	PO-acc †	**0.1999**	**0.4227**	**0.5432**	**0.6705**	**0.1999**	**0.0845**	**0.0543**	**0.0335**	0.7977	0.1897
	PO-nov •	0.1513	0.3725	0.4854	0.6111	0.1513	0.0745	0.0485	0.0306	**0.8597**	0.1411
	PO-div ⋄	0.0820	0.2271	0.3269	0.4470	0.0820	0.0454	0.0327	0.0223	0.7138	**0.2563**

usefulness, as a component of the weighting system of the Weighted Slope One algorithm. We propose to use the concept of web of trust to compute the weights [6].

We compare the Weighted Slope One (WSO) with our four proposed strategies: (i) Input Degree Based Slope One (IDSO), relies on the number of users that trusts on the active user; (ii) Linear Item Usefulness Slope One algorithm (LIUSO), relies on the linear difference between the maximum mean average error (MAE) and the MAE of the current item; (iii) Exponential Item Usefulness Slope One algorithm (EIUSO), relies on the exponential value of the previous difference; (iv) Mixed Slope One (MSO), is a linear combination of the previous strategies.

Table 4: Results for Flixster dataset

Threshold	Metric	WSO	IDSO	LIUSO	EIUSO
4	MAE	.741	1.954	.734	.714
	RMSE	1.059	2.499	1.056	1.058
	Precision	81.2%	59.8%	80.7%	79.2%
	Recall	42.0%	33.3%	44.2%	50.2%
	F1	55.4%	42.7%	54.8%	61.5%
	Gain (F1)	–	−12.7%	−0.6%	6.1%
Mean	MAE	.741	1.954	.747	0.714
	RMSE	1.059	2.499	1.068	1.058
	Precision	60.4%	52.1%	59.2%	61.2%
	Recall	53.8%	37.6%	52.5%	69.2%
	F1	56.9%	43.7%	55.6%	64.9%
	Gain (F1)	–	−13.2%	−1.3%	8.0%

In Table [4], we show the results obtained by each method in the Flixster dataset. Note that we split the experiment into two subsets by considering only ratings greater than 4 and ratings greater than the mean of all ratings. All results are statistically significant. The IDBSO method was poor in all thresholds considered. On the other hand, the IDBSO, EIUSO, and MSO methods perform better than baseline from 2.2% to 6.%.

5. CONTRIBUTIONS

By addressing the research topics, this work foresees the following contributions:

- A set of new performance-oriented approaches that produce ranked lists of suggested items that are either accurate, diverse and novel.
- A hybridization technique for combining different recommendation algorithms, following the Strength Pareto approach.
- A set of methods to incorporate trust information in Collaborative Filtering systems.

6. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval (Second edition)*. Pearson, 2011.

[2] O. Celma and P. Herrera. A new approach to evaluating novel recommendations. In *ACM RecSys*, pages 179–186, 2008.

[3] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pages 39–46, 2010.

[4] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 263–272, 2008.

[5] D. Kenny, D. Kashy, william Cook, and J. Simpson. *Dyadic Data Analysis*. The Guilford Press, 2006.

[6] P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, pages 29–33, 2006.

[7] S. McNee, J. Riedl, and J. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstract*, pages 1097–1101, 2006.

[8] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. Movielens unplugged: experiences with an occasionally connected recommender system. In *IUI*, pages 263–266, 2003.

[9] M. T. Ribeiro, A. Lacerda, A. Veloso, and Z. Nivio. Pareto-efficient hybridization for multi-objective recommender systems. In *ACM RecSys*, pages 9–16, 2012.

[10] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[11] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *ACM RecSys*, pages 109–116, 2011.

[12] P. Wang and H. Ye. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. In *ISECS*, pages 152–154, 2009.

[13] R. Zafarani and H. Liu. Social computing data repository at ASU, 2009.

[14] D. Zhang. An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing. In *ISECS*, pages 215–217, 2009.

[15] E. Zitzler and L. Thiele. Multi-objective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. on Evolutionary Computation*, 3(4):257–271, 1999.