

# DT-BAR: A DYNAMIC ANT RECOMMENDER TO BALANCE THE OVERALL PREDICTION ACCURACY FOR ALL USERS

Abdelghani Bellaachia<sup>1</sup> and Deema Alathel<sup>2</sup>

<sup>1,2</sup>School of Engineering and Applied Science,  
The George Washington University, Washington, DC, USA  
<sup>1</sup>bell@gwu.edu, <sup>2</sup>atheld@gwmail.gwu.edu

## ABSTRACT

*Ant colony algorithms have become recently popular in solving many optimization problems because of their collaborative decentralized behavior that mimics the behavior of real ants when foraging for food. Recommender systems present an optimization problem in which they aim to accurately predict a user's rating for an unseen item by trying to find similar users in the network. Trust-based recommender systems make use of trust between users. T-BAR was the first successful application of an ant colony algorithm to trust-based recommender systems but it lacked the ability to deal with cold start users. In this paper we propose a dynamic trust-based ant recommender (DT-BAR) that solves the problem of cold start users by locally initializing the pheromone level on edges using the dynamically changing information within each neighborhood as ants pass by. DT-BAR increases the communication between ants and emphasizes the importance of trust in the pheromone initialization process.*

## KEYWORDS

*Ant Colony Optimization, Artificial Agents, Bio-inspired Algorithms, Recommender Systems, Trust.*

## 1. INTRODUCTION

Collaborative filtering (CF) is the most common technique applied to recommender systems (RS) to suggest to users items that may be of interest to them [1]. CF techniques generate recommendations based on the items that are highly rated by users similar to the active user [2]. Such techniques suffer from their inability to deal with cold start users who have rated only a few items, which makes it hard for the system to find similar users. There have been several attempts to overcome this problem, such as allowing the users to express their trust level in other users in the network in hopes of utilizing the trust to enhance the system's performance. Such systems are known as trust-based recommender systems (TBRS) and one of the major trust-based algorithms applied to such systems is Massa et al.'s MoleTrust [3], which at the time it was presented showed to outperform traditional CF techniques and Golbeck's TidalTrust [4]. The MoleTrust approach is based on propagating the trust over the network to predict the active user's trust in other distant users and then use the propagated trust to reach the target users with the rating for the target item. However, we recently proposed our ant colony inspired algorithm, Trust-Based

Ant Recommender (T-BAR), which surpassed the performance of other algorithms for TBRS, including MoleTrust [5]. T-BAR's major advantage over other popular algorithms is that unlike other algorithms that only consider a single user at the end of each search path, it considers all users with a rating for the target item that are encountered in the search process. The paths are constructed by combining both trust and semi-similarity between the users, which guarantees the quality of the constructed solutions.

## 2. RELATED BACKGROUND

### 2.1. Trust-Based Recommender Systems

RS provide an adaptive web environment for users by filtering the vast amount of information available online and delivering the right information to the right user. At their core RS suggest to users items that may be of interest to them. One of the major techniques applied to accomplish this goal is CF, which recommends items that were liked by likeminded users in the network. However, in recent years many researchers shifted their focus to TBRS based on the popular belief that users tend to trust people they know rather than depending on a RS to find similar *unknown* people in the network. The incorporation of trust in such networks results in neighborhoods of trust where a web of trust for a user  $x$   $WOT_x$  refers to the group of users that are trusted by user  $x$ . In this manner, users trusted through chains of trust (i.e. friends of friends) are considered part of an extended WOT. Trust metrics in TBRS usually utilize trust by propagating it to reach a wider range of users who would not be otherwise reached if traditional CF techniques were used [4, 3].

### 2.2. Ant Colony Optimization Algorithms

Ant colony optimization (ACO) is a family of algorithms that falls under swarm intelligence. An ACO algorithm applies a probabilistic technique to solve optimization problems by mimicking the behavior of ants when they forage for food [6, 8]. In the context of RS, the active user can be considered the ants' nest while the target users that have a rating for the target item are considered the *good* food sources [5]. Within such a framework, artificial ants are dispatched from the active user into a network of users connected through multiple WOT and they construct paths that lead to possible target users in the solution space. Just like real ants, artificial ants deposit pheromone on paths leading to *good* users, and just like in real ant colonies, pheromone on edges evaporates as time passes by to allow exploration of other possible solutions. The combined effect of pheromone deposit and evaporation increases the probability of other ants traversing *good* paths while decreasing the probability of crossing others. The ants are usually dispatched from the active user over several iterations so the system moves from an unstable stage, where no solution is necessarily better than another, to a stable one where certain paths emerge as being the best solutions leading to the *best* food sources.

### 2.3. Trust-Based Ant Recommender

T-BAR [5] is a bio-inspired algorithm that is based on the ant colony system (ACS) algorithm proposed by Dorigo et al. [9]. T-BAR is the first successful application of an algorithm derived from ACS to TBRS. When compared to other popular algorithms for TBRS [4, 3], T-BAR greatly enhanced the accuracy of rating predictions and ratings coverage especially for heavy raters. However just like other techniques, it suffers from its inability to deal with cold start users.

At each step within an iteration in T-BAR, an ant  $k$  located at user  $x$  calculates the probability  $p_{xy}^k$  of crossing the edge connecting to a user  $y$  that belongs to  $WOT_x$ . The edge that yields the highest probability is the one that is crossed. The probability  $p_{xy}^k$  is calculated as:

$$p_{xy}^k = \frac{(\tau_{xy})^\alpha (\eta_{xy})^\beta}{\sum_{z \in WOT_x} (\tau_{xz})^\alpha (\eta_{xz})^\beta} \quad (1)$$

where  $\tau_{xy}$  is the pheromone level on the edge  $xy$ ,  $\eta_{xy}$  is the trust  $T_{xy}$  expressed by user  $x$  towards user  $y$ , and  $\alpha$  and  $\beta$  are influence parameters. Each ant stops constructing its solution once a certain search depth  $d$  is reached.

Pheromone update is accomplished on two levels in T-BAR, a local one and a global one. The *local pheromone update* occurs whenever an ant  $k$  crosses an edge  $xy$ , which results in adjusting  $\tau_{xy}$  using:

$$\tau_{xy} = (1 - \rho) \cdot \tau_{xy} + \rho \cdot \tau_{xy}^0 \quad (2)$$

where  $\rho$  is the pheromone evaporation coefficient and  $\tau_{xy}^0$  is the initial pheromone level on  $xy$  calculated within each  $WOT_x$  as [10]:

$$\tau_{xy}^0 = \frac{1}{\sum_{z \in WOT_x} T_{xz}} \quad (3)$$

The *global pheromone update* takes place at the end of each iteration where first each constructed path's quality is evaluated by calculating its *path trust*  $PT_k$  [5, 3] and the paths that satisfy  $PT_k \geq PT_{threshold}$  are considered the best paths so far. The pheromone level on the edges belonging to the best paths is further incremented by a quantity that is proportional to the path's  $PT_k$ . At the end of the last iteration, the system should have converged to the *best* solutions and the *good* users found on those paths are used to predict the rating for the target item  $i$ .

### 3. PROPOSED DYNAMIC TRUST-BASED ANT RECOMMENDER

Our proposed Dynamic Trust-Based Ant Recommender (DT-BAR) is a different variation of T-BAR that focuses on solving the problem of cold start users. DT-BAR applies a dynamic approach based on ACO algorithms' probabilistic methodology and emphasizes on utilizing trust and information sharing among ants to predict item ratings for users in a TBRS.

#### 3.1. Rationale behind Proposed Approach

T-BAR's success is greatly credited to incorporating trust to strengthen the paths constructed by ants based on user popularities and user similarities and thus to ultimately find *good* users with *good* quality solutions [5]. In other words, the quality of *good* users reached through T-BAR was high due to its ability to find users that have many items in common with the active user *and* that have a high trust level. T-BAR's approach guarantees great results for heavy raters because such users have rated many items and therefore it is easier for the algorithm to find *good* quality solutions. However, the same argument cannot be applied to cold start users due to their lack of item ratings, which would explain T-BAR's inability to perform well for them.

A solution to the cold start problem would require expanding the ants' exploration scope of the solution space to increase the probability of finding better users that can contribute to providing more accurate predictions. In T-BAR, the ants' edge selection mechanism is dictated by the trust between two users and the evaluation of the best paths is determined by the number of co-rated items between users. For cold start users, the ants would be guided only by trust, so the role of trust needs to be further emphasized for such users to compensate for the lack of item ratings and co-rated items.

To overcome this problem we propose to:

- 1) Allow the artificial ants to share more information among them about the paths that have been explored and thus to support the exploration of the other undiscovered paths.
- 2)

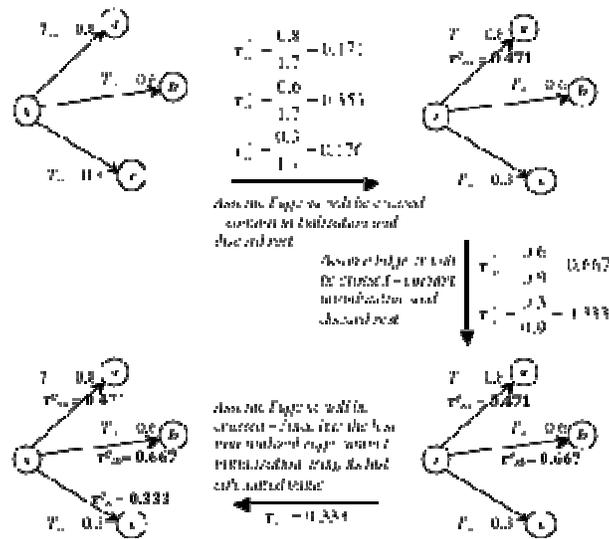


Figure 1. Example of DT-BAR's pheromone initialization process

- 3) Increase the role of trust in the pheromone initialization process so that the initial pheromone levels on edges within  $WOT_x$  would reflect the different trust levels on those edges.
- 4) We propose to accomplish both goals during the pheromone initialization process. To attain the first goal, we suggest allowing each ant to commit the initial pheromone level only on the edge that will be crossed while discarding the other initializations. This is closely related to the second goal though, because if we opt to calculate the initial pheromone level in a manner similar to the way it is done in T-BAR (Eq. 3), then the probability of crossing undiscovered edges would still be relatively unaffected especially if the previously crossed edges keep accumulating pheromone on them. Therefore, the initial pheromone level should be calculated in a way that reflects the trust assigned to each edge within  $WOT_x$ .

### 3.2. DT-BAR Algorithm

DT-BAR follows the same methodology proposed in [5] to predict ratings for unseen items for the active user in TBRS. DT-BAR's major difference from T-BAR is evident in the pheromone initialization step. Instead of calculating a single initial pheromone level  $\rho_{xy}$  for all nodes within

$WOT_x$ , our proposed algorithm would calculate a different value for each edge within  $WOT_x$  that reflects its associated trust level as follows:

$$\tau_{xy}^0 = \frac{T_{xy}}{\sum_{z \in WOT_x^*} T_{xz}} \quad (4)$$

where  $WOT_x^*$  refers to users in  $WOT_x$  with *uninitialized* edges. After calculating the probabilities of crossing those edges, DT-BAR allows an ant  $k$  to commit the pheromone initialization only on the edge that yielded the highest probability. So we can see how committing the initialization only on the crossed edges serves as a message to subsequent ants about which edges have been explored and thus those edges are not included later in the pheromone initialization of edges in the dynamically updated  $WOT_x^*$ . Also, DT-BAR emphasizes the role of trust on the different edges with  $WOT_x$  by calculating an initial pheromone level that is proportional to  $T_{xy}$ . Figure 1 is an example that demonstrates DT-BAR's pheromone initialization process.

The effect of the introduced change in the pheromone initialization will impact:

- 1) The probability  $p_{xy}^k$  (Eq. 1) because the initial pheromone level on an edge  $xy$  determines the initial probability of crossing that edge and since the calculated  $\tau_{xy}^0$  will dynamically change as ants continue to cross edges, we would expect the probability of selecting uncrossed edges to increase as ants keep passing by the neighborhood.
- 2) The local pheromone update of  $\tau_{xy}$  (Eq. 2) because it involves the initial pheromone level to determine the amount of pheromone to be deposited. So, once an edge is initialized to a value that is large enough to increase its probability of being crossed, then that *successful* initial pheromone level would be committed and will determine the rate at which pheromone is deposited on that edge.

## 4. EXPERIMENTAL EVALUATION

### 4.1. Dataset and Evaluation Metrics

The Epinions dataset was used to test our proposed algorithm DT-BAR. Epinions is one of the few publically available datasets that provides access to both user ratings and explicit trust values between users. It is composed of 49,290 users who rated 139,738 unique items at least once and 487,181 distinct trust statements. The two major user categories of utmost importance in Epinions are *cold start users*, who comprise more than half of the users in the dataset, and *heavy raters*. *Cold start users* are users who rated less than 5 items each in the dataset, while *heavy raters* are the ones who rated more than 10 items each [11]. Some other user and item categories that are worth considering in Epinions include *opinionated users* who rated 5 or more items and whose standard deviation is greater than 1.5; *black sheep users* who rated 5 or more items and the average distance of their rating for item  $i$  with respect to the mean rating of item  $i$  is greater than 1; *niche items* that received less than 5 ratings each; and *controversial items* received ratings whose standard deviation is greater than 1.5.

We applied the *leave-one-out* technique to measure DT-BAR's prediction ability. We compared our obtained results to the results reported from running a basic CF algorithm that uses the Pearson Similarity and to Massa's MoleTrust algorithm [3]. Massa's work is one of the early major techniques applied to TBRS and it has been compared to many contributions in the literature [12, 13, 14, 15]. The results of our empirical evaluation were analyzed in terms of the

*Mean Absolute Error* (MAE) along with the *ratings coverage* (RC) [16], which measures an algorithm's ability to predict a rating for a certain item (regardless of the accuracy of the prediction). One of the major drawbacks in the MAE is that it has the same weight for all user categories in the dataset without taking into account, for example, the percentage of cold start users and heavy raters in the system, so in the case of Epinions the error for heavy raters shadows the one for cold start users [3]. In order to gain a better understanding of an algorithm's performance the *Mean Absolute User Error* (MAUE) is usually used because it averages the MAE by the number of users in each category. In the same perspective, the *users coverage* (UC) represents the percentage of users in the dataset for which the algorithm was able to provide with at least one prediction.

## 4.2. Experimental Results

We compare DT-BAR's performance results to the ones reported by running several other algorithms on the Epinions dataset, namely: *CF* which is Massa's implementation of a basic CF algorithm that uses the Pearson Similarity [11], *MT* which is the MoleTrust algorithm [3], and T-BAR which is our basic trust-based ant recommender [5].

At first glance, a quick look at Table 1 and Table 2 would show that T-BAR has the best overall performance since it achieves the lowest MAE of  $\sim 0.3$  and the highest ratings coverage of 93%.

Table 1. The MAE of the algorithms across different views.

Views	Algorithm			
	CF	MT	T-BAR	DT-BAR
All	0.843	0.832	0.298	0.723
Cold start users	1.094	0.674	1.459	0.714
Heavy raters	0.850	0.873	0.212	0.778
Controversial items	1.515	1.425	1.995	1.629
Niche items	0.822	0.734	0.572	0.222
Opinionated users	1.200	1.020	1.308	0.411
Black sheep	1.235	1.152	1.973	0.812

Table 2. The RC of the algorithms across different views.

Views	Algorithm			
	CF	MT	T-BAR	DT-BAR
All	51%	28%	93%	84%
Cold start users	3%	11%	91%	55%
Heavy raters	58%	31%	93%	87%
Controversial items	45%	25%	59%	39%
Niche items	12%	8%	48%	84%
Opinionated users	50%	23%	94%	34%
Black sheep	56%	24%	77%	37%

Table 3. The MAUE of the algorithms across different views.

Views	Algorithm			
	CF	MT	T-BAR	DT-BAR
All	0.938	0.790	1.203	0.790
Cold start users	1.173	0.674	1.581	0.784
Heavy raters	0.903	0.834	0.282	0.806
Controversial items	1.503	1.326	1.967	1.750
Niche items	0.854	0.671	0.896	0.323
Opinionated users	1.316	0.938	1.262	0.498
Black sheep	1.407	1.075	1.973	0.895

Table 4. The UC of the algorithms across different views.

Views	Algorithm			
	CF	MT	T-BAR	DT-BAR
All	41%	47%	96%	68%
Cold start users	3%	17%	97%	50%
Heavy raters	86%	80%	93%	90%
Controversial items	16%	12%	92%	73%
Niche items	11%	10%	74%	85%
Opinionated users	61%	61%	94%	39%
Black sheep	68%	61%	81%	43%

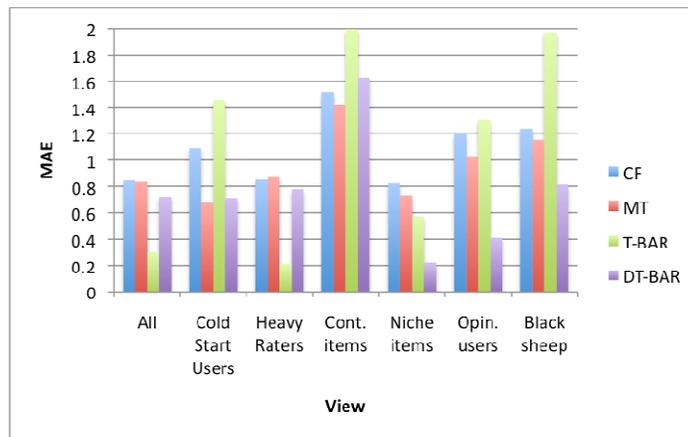


Figure 2. The MAE of the algorithms across different views

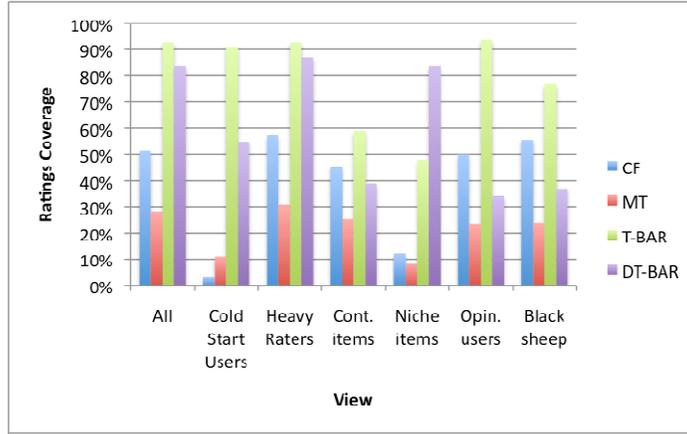


Figure 3. The RC of the algorithms across different views

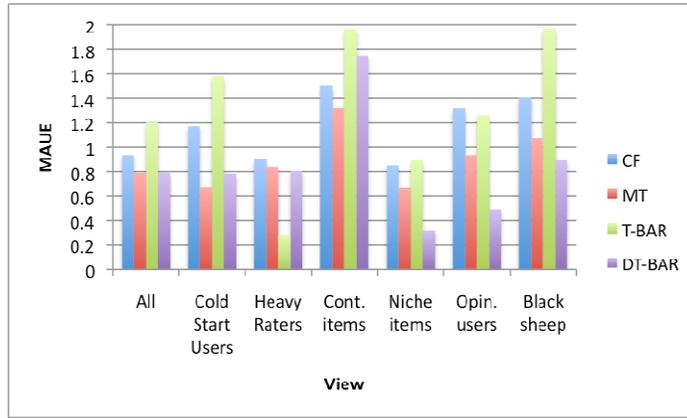


Figure 4. The MAUE of the algorithms across different views

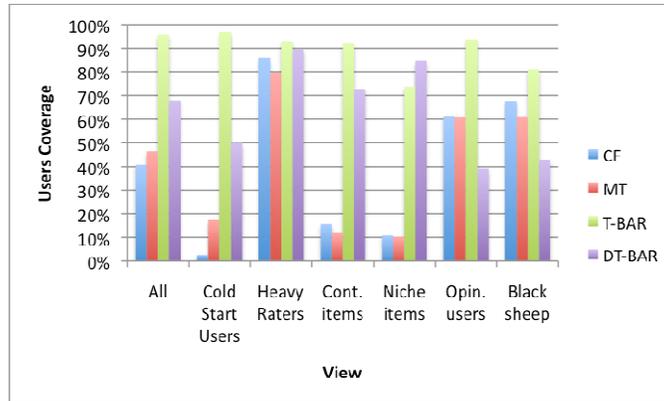


Figure 5. The UC of the algorithms across different views

However, a closer look at Table 1 and remembering how the MAE is calculated we can see how the overall accuracy does not reflect the results for the majority of users in the dataset, i.e. cold start users, but is rather affected by the ones for heavy raters due to the big difference in the accuracy between the two user categories. The same observation can be noticed with respect to

CF and MT. However, DT-BAR does not suffer from the same problem because it achieves almost similar MAEs for both cold start users and heavy raters ( $\sim 0.7$  and  $\sim 0.78$  respectively) which results in the overall MAE of 0.723 not being misleading as it is in the case of the other three algorithms. DT-BAR's overall RC of 84% is a bit lower than T-BAR's 93% but given DT-BAR's acceptable consistent overall performance, the drop in coverage can be considered reasonable. On the other hand, Table 3 and Table 4 show how the MAUE and UC are better indicators of all four algorithms' performances since the overall accuracies are direct reflections of the results obtained for cold start users. Even when averaging the results by the number of users/items in each category, DT-BAR outperforms the other algorithms because its overall MAUE of  $\sim 0.8$  reflects the results achieved for both cold start users and heavy raters (both  $\sim 0.8$ ). One might argue that DT-BAR's MAUE levels for the two major user categories are close to the ones obtained by MT, however Table 4 shows that DT-BAR provides better levels of UC (50% vs. 17% for cold start users and 90% vs. 80% for heavy raters). When it comes to heavy raters alone, T-BAR is still considered to beat all other algorithms in terms of both accuracy and coverage by reaching a MAUE as low as  $\sim 0.28$  and a UC of 93%.

Another major advantage of DT-BAR is its superior performance for niche items, which can be a challenge to deal with in RSs just like cold start users due to the scarcity of ratings available for those items. DT-BAR achieved a low MAE of 0.2 and a RC of 84%, which are better than any of the results achieved by the other three algorithms. Our proposed algorithm also achieved lower MAE and MAUE for black sheep and opinionated users however that came with the price of lower coverage percentage for both user categories. DT-BAR's performance for controversial items was almost similar to the average performance achieved by the other algorithms.

Overall, it is evident that T-BAR is always the better choice if the dataset is composed mostly of heavy raters since it provides an amazing performance that outperforms the ones achieved by all other algorithms (MAE  $\sim 0.2$  and 93% RC). On the other hand, if the distribution of user categories in the dataset is unknown then DT-BAR would be a more suitable option since it delivers an acceptable consistent performance across the different discussed views. Another case where DT-BAR should be considered is when a RS consists of a substantial number of niche items whose accuracy of predictions could affect a user's confidence in the system's performance.

## 5. CONCLUSION

In this paper we proposed our Dynamic Trust-Based Ant Recommender (DT-BAR) to achieve a consistent performance for the two major user categories in RS: cold start users and heavy raters. Proposed algorithms in the literature can only deliver good results for one user category at the expense of the other, but DT-BAR managed to balance the performance by attaining a consistent acceptable accuracy levels across the two categories. DT-BAR achieves that by allowing the artificial ants to share information about the explored edges and by initializing the pheromone level on edges to values proportional to their corresponding trust level. DT-BAR allows each ant to locally calculate the initial pheromone level for each edge within the neighborhood but only commit the initialization on the edge to be crossed which would serve as a message to the other ants to indicate which edges have been crossed and at the same time increase the possibility of crossing other edges by dynamically calculating the new initial pheromone level using the newly available information about the uninitialized edges.

The initial pheromone level is a determining factor in any ACO algorithm's convergence speed and accuracy, and DT-BAR's proposed dynamic pheromone initialization approach proved to be successful in terms of allowing the ants to expand the scope of their edge exploration, which benefited both cold start users and niche items. Our proposed algorithm loosely preserved the

ants' ability to exploit the good discovered paths for heavy raters, which resulted in finding *acceptable* good users with a rating for the target item in general. However, DT-BAR's performance for heavy raters is not as good as the ones achieved by T-BAR but at least it matched the average results obtained by a basic CF algorithm or Massa's MoleTrust (in other words, it was not worse than their results for heavy raters).

DT-BAR's expanded exploration mechanism proved to be feasible for cold start users and niche items. In general, DT-BAR achieved an acceptable, good, consistent performance for both cold start users and heavy raters as opposed to the other algorithms in the literature.

## REFERENCES

- [1] Resnick, Paul & Varian, Hal R., (1997) "Recommender systems", *Commun. ACM*, Vol. 40, pp 56-58.
- [2] Schafer, J. Ben, Frankowski, Dan, Herlocker, Jon & Sen, Shilad, (2007) "Collaborative filtering recommender systems", *The Adaptive Web*, Springer-Verlag.
- [3] Massa, Paolo & Avesani, Paolo, (2007) "Trust-aware recommender systems", *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp 17-24.
- [4] Golbeck, Jennifer, (2005) "Computing and applying trust in web-based social networks," Doctoral Dissertation, University of Maryland at College Park.
- [5] Bellaachia, Abdelghani & Alathel, Deema, (2012) "Trust-Based Ant Recommender (T-BAR)", *IEEE Conference of Intelligent Systems*, Sofia, Bulgaria, pp 130-135.
- [6] Dorigo, Marco, (1992) "Learning and natural algorithms", Doctoral Dissertation, Politecnico di Milano.
- [8] Dorigo, Marco, Bonabeau, Eric & Theraulaz, Guy, (2000) "Ant algorithms and stigmergy", *Future Generation Computer Systems*, Vol. 16, pp 851-871.
- [9] Dorigo, Marco & Stützle, Thomas, (2004) "Ant colony optimization", MIT Press.
- [10] Bellaachia, Abdelghani & Alathel, Deema, (2014) "A Local Pheromone Initialization Approach for Ant Colony Optimization Algorithms", *IEEE International Conference on Progress in Informatics and Computing*, Shanghai, China.
- [11] Massa, Paolo & Avesani, Paolo, (2004) "Trust-aware collaborative filtering for recommender systems", *Proceedings of the Federated International Conference on the Move to Meaningful Internet*, Larnaca, Cyprus, Springer-Verlag, pp 492-508.
- [12] Avesani, Paolo & Massa, Paolo, (2005) "Moleskiing.it: A Trust-aware recommender system for ski mountaineering", *International Journal for Infonomics*, pp 1-10.
- [13] Golbeck, Jennifer, (2006) "Generating predictive movie recommendations from trust in social networks", *Lecture Notes in Computer Science*, Vol. 3986, pp 93-104.
- [14] O'Donovan, John & Smyth, Barry, (2005) "Trust in recommender systems", *Proceedings of the 10th International Conference on Intelligent User Interfaces*, San Diego, California, USA, pp 167-174.
- [15] Victor, Patricia, Cornelis, Chris, Cock, Martine De & Teredesai, AM, (2008) "Key figure impact in trust-enhanced recommender systems", *AI Commun.*, Vol. 21, pp 127-143.
- [16] Herlocker, Jonathan L., Konstan, Joseph A., Terveen, Loren G. & Riedl, John T., (2004) "Evaluating collaborative filtering recommender systems", *ACM Transactions on Information Systems*, Vol. 22, pp 5-53.

**AUTHORS**

**Abdelghani Bellaachia** received his B.S. and M.S. degrees in electrical engineering from Mohammadia School of Engineering, Rabat, Morocco in 1983. He later received his second M.S. degree in computer science from the George Washington University, Washington, DC, in 1991. He earned his Ph.D. from the same university in software systems a year later. Since then he has been a faculty member at the George Washington University and is currently an associate professor there. His research interests include data mining, multi-lingual information retrieval systems, bio-informatics, design and analysis of algorithms, and parallel processing.



**Deema Alathel** is a doctoral student at the George Washington University, Washington, DC. She received her B.S. and M.S. degrees in computer science from King Saud University, Riyadh, Saudi Arabia. She has worked at the Institute of Public Administration in Riyadh for 2 years as a faculty staff member. Her research interests include data mining, information retrieval systems, and bio-informatics.