



Improving Friends Matching in Social Networks Using Graph Coloring

Omayya Murad¹, Azzam Sleit², Ahmad Sharaiah³

¹st The University Of Jordan, Amman Jordan, umaiya.murad@gmail.com

²nd The University Of Jordan, Amman Jordan, azzam.sleit@ju.edu.jo

³rd The University Of Jordan, Amman Jordan, sharieh@ju.edu.jo

Abstract

Recently, most of people have their own profiles in different social networks. Usually, their profiles have some brief description about their personnel picture, family members, home town, career, date of birth etc. which indicate other people know some general information about others. In social networks, usually friends recommendation is done by finding the most mutual friends and suggest them to be friends. In this paper, we will introduce an algorithm, with a linear time complexity, that helps people to get not only good friends but also have same characteristics.

Keywords

Algorithm, Graph coloring, Social Networks, Matching Friends.

Academic Discipline And Sub-Disciplines

Computer Science

SUBJECT CLASSIFICATION

Computer Science Classification

TYPE (METHOD/APPROACH)

Algorithms

I. Introduction

In social networks, graphs can be used to represent the social network. A graph G can be defined as $G = [V, E]$, where V is the set of vertices in the graph and E is the set of edges. Each edge $e \in E$ connects two distinct vertices in the graph, Cormen et al. [7], [11]. Graphs can be used to solve many kinds of problems; one of these problems is the social networks problems. Obviously, each vertex $v \in V$ will present a person profile and each edge connects two vertices if there is a friendship between the two people presented by the vertices.

The graph coloring problem is such a graph labeling that we use to color graph elements under some constraints with minimal colors. It colors graph vertices such that no two adjacent vertices hold the same color, which we call it vertex coloring. It can be inferred that if two vertices have the same color, then they have no relation that connects them together. Moreover, edges can be colored by assigning a color for each edge that no two adjacent edges have the same color. Graph coloring can be used in different fields such as biochemistry, electrical engineering and computer science, Rossi, Rayan A, and Nesreen K. Ahmad [20]. It can be used specifically in scheduling, resource allocation and timetabling, Cormen et al. [7].

Chalupa (2011) [5], applied coloring algorithm with a proper heuristic in social networks in order to define the G graph and its complementary \bar{G} , where $G = [V, E]$ represents the social network graph, every $v_1, v_2 \in V$ relates to two distinct people and every $e \in E$ is an edge, which implies that v_1 knows v_2 . On the other hand, $\bar{G} = [\bar{V}, \bar{E}]$ defines a complementary graph that every $\bar{v}_1, \bar{v}_2 \in \bar{V}$ relates to two distinct people and every $\bar{e} \in \bar{E}$ is an edge that inform \bar{v}_1 does not know \bar{v}_2 . Essentially, the social network will have sub-groups V and \bar{V} , where V will hold the list of friends who are in the friendship list and the complementary group \bar{V} will hold the list of people who do not have friendship among each other. In this paper, we will concentrate on the second group \bar{V} to find new friends for people whose identifications are matching as much as others with similar identifications. This will help people to get the most proper friends. Other coloring algorithms can be added such as those introduced in [10][2][4].

Graph coloring is not an easy job and it can be considered as a NP problem. In [18], they presented an algorithm, which can find the Maximum Independent Set (MIS) in large sizes graphs. They apply their algorithm Finding Maximum Independent Set (FMIS) over several graphs of orders: 20, 30, ..., 2000. Sharieh and his colleagues in [18] claimed that the complexity of FMIS can be considered as $O(1.0052n)$ for a graph of n nodes which is better than other algorithms.

On the other hand, in [2], finding the MIS can be done based on the weights of the nodes, while the weight of the node depends on the degree of the node. In [2], Al-Jaber and Sharieh presented an algorithm for finding MIS according to weights of the nodes with $O(N^2)$ complexity in the worst case of a graph of N nodes.

Finding the MIS can be very helpful in clustering, which is based on partitioning a large number of entities into a fixed number of groups based on different number of attributes[17], such as assigning students to proper groups as authors presented in [16], [12].

Rossi & Musolesi in 2014 [19], demonstrated different techniques for identifying the users social networks from their check-in data. Actually, they presented three strategies. The first, spatio-temporal trajectory relates to the time they check-



in. The second, using the frequency they visit the same place, respectively. The third strategy was a hybrid between previous two strategies. Some of datasets they created enabled them to classify more than 80% of the users correctly.

In [15], author presented a Artificial neuro fuzzy logic system for detecting human emotions, which can be used in social networks to detect the user's emotion during he/she is checking out his/her profile of timeline which can be used to match friends emotions in check-in places. Although it will be as another indicator of the friends matching, but for simplicity in this research we will use the three identifications presented in [19].

Dunbar, R. [9] claimed that there is a limited number of friends that can be handled by a human being, so finding a good friend who holds the same characteristics is not an easy job. Obviously, it requires too much time to know your partner's characteristics such that what he/she likes or dislikes, where he/she likes to go, and why. All of these questions can be used when you are searching for a new good friend who matches your personality issues or when a person asks you for creating friendship. On the other hand, when people applying for a new job it is a good chance for the managers to get a brief description about the candidates before calling them to the interviews even if they do not know each other.

In this paper, we will present a new algorithm that will be helpful in finding friends whose characteristics are matching more than the people you may know and exist in your friendship list. This algorithm can be used also for finding the proper people for the proper position in a specific company. When a manager knows the characteristics of this person, he/she will be able to decide which one will be the best candidate to be chosen.

The rest of this paper is organized as follows . Section II presents Literature Review. Section III present the proposed algorithm. Section IV presents the algorithm analysis. Finally, the conclusion is presented in Section V.

II. Literature Review

In [14], Malkawi et al. used coloring algorithm to present a new exam scheduling. The exam scheduling algorithm considered as an NP-hard problem that could not be solved in polynomial time, also it has many strict constraints, such as a student should not have 2 exams at the same day. Another one, is to reduce the gaps between students exams. In [14], authors did their best to achieve accuracy, fairness and optimal time period. Authors claimed that the complexity time needed for their algorithm is linear in time complexity.

When we want to implement social networks by graphs, we need a large graph size to be used because social networks usually have a huge number of participants. In [8], Donderiaa V & K. Janaa P presented a novel scheme for graph coloring that is dealing with large graph problems. Authors presented their own algorithm and they called it NGC algorithm to color large graphs using minimum number of colors. They show the time complexity of NGC algorithm as $O(n^3)$. They compared their NGC algorithm with DSATUR algorithm presented by Brelaz D [4], using different number of graph sizes 7 - 600 nodes. Although NGC algorithm used more colors than DSATUR algorithm for the number of nodes 60, 520 and 600. It shows less running time for all cases, according to the comparisons that authors did in [8].

Social networks are very large networks. There are different kinds of personal information for their users and also they can be considered as good indicators of their users interests. In [13], H. Lee D & Brusilovsky P presented a study over "CiteULike" which investigated if the social connection might indicate user similarity, which depends on the strength of the their connection.

In [3], the authors showed that if a local algorithm has constant memory and constant lookahead is arbitrarily worse than the global optimal. They also compared local algorithms and they showed there is a scenario for every local algorithm that it performs arbitrarily worse than another local algorithm.

Usually, friends recommendation of two people (x,y) is built according to common friends who exist in both friendship lists for both x and y and have some common similarities such as the name of the company they are working in as shown in Guy, I et al[10].

Alvin Chin, Bin Xu and Hao Wang [6] showed a new algorithm to recommend friends using physical locations such as friends attending the same workshop or conference. They added some social interactions such as sending and receiving messages and ask some questions to each others. They based their study on meeting for proximity; common interest and common friends for homophily. This can be used in studying physical locations cases, like workspaces and conferences. In their results, they mentioned that for people in the conference case, the main two reasons to recommend a friend is that they may know each other or maybe they faced each other before. But, what about people who do not know each other and they have common identifications? In this paper, we will show a new algorithm that will help these people to be good friends.

III. Algorithm $M(G, \delta, P)$

In this section, a new algorithm will be presented, which will be used to improve matching friends on social networks. This algorithm will be implemented using four main stages. Assume that we want to match new friends for a person δ , as follows.

First Stage: The algorithm will apply the best graph coloring algorithm to color the network. One algorithm with low complexity time was presented by Malkwai and his colleagues [14], because it has a linear run time complexity. The output of the algorithm will be a colored graph, which will give us two distinct groups. The first group L holds the people who know each other and thus they are connected in the social network. The second is \bar{L} group, which is the one that has the people who are not connected to each others. In other words, the people who do not exist in the friendship list of the required person δ .



Second stage: the algorithm will find the characteristics of the required person \bar{b} by applying Rossi & Musolesi 2014 algorithm [19]. Also it will be applied over all vertices in group \bar{C} to get their identifications and record them.

Third Stage: the second group \bar{C} will be used to search in all the vertices (personal profiles) in it and get the characteristics of each person p ; then determine if any characteristic matches the characteristic of the required person \bar{b} . Here we will create a new weighted graph that holds the vertices of people such as p who has the same characteristic of \bar{b} . Obviously this will be done by adding an edge that connects p and \bar{b} . On the other hand, the weight of this edge will implicate the amount of matching characteristics between p and \bar{b} . The more weight value the edge has, the more matched characteristic p and \bar{b} have. Notice that the characteristic will indicate by applying the Rossi & Musolesi 2014 [19] algorithm with all kind of techniques they presented. At the end of this step, we will get a new weighted graph \bar{G} , which represents all people p who have at least one common characteristic with the required person \bar{b} .

Fourth Stage: we will go through the new graph \bar{G} and find the maximum weighted edge, which will connect the required person \bar{b} with the most matching person P . These are the nodes represent the persons with the largest number of common characteristics between each others.

Algorithm $M(G, \bar{b}, P)$

// Input : G and \bar{b} , where

// G is the social network graph, \bar{b} the profile for
a person we want to find new friends for
him/her

// output : P , is the most matched person found.

1. Obtain the social network using a graph $G = [V, E]$.
2. Check all identifications of \bar{b} using techniques shown in [19].
3. Assign a vertex for each personal profile in the network, where $v_i \in V$.
4. For each vertex $v_i \in V$ and for all v_j in the friendship list of v_i draw an edge $e_i \in E$, between v_i and v_j .
5. Apply the graph coloring algorithm presented in [14], which will produce two subgroups P and \bar{P} , such as, P holds all \bar{b} 's friends and \bar{P} will hold all vertices related to people who are not in the \bar{b} 's friendship list (i.e. people do not know \bar{b} and visa verse, which will be searching area to get new proper match-able friends for \bar{b}).
6. For each $\bar{v}_i \in \bar{P}$, apply all techniques in [19], and find every ones identifications.
7. Compare each identification for both \bar{b} and \bar{v}_i if they have the same identification draw an edge between \bar{b} and \bar{v}_i and the number of matching identifications to w_i value, which is the weight of the edge between \bar{b} and \bar{v}_i .
8. Go through the graph and arrange the edges descending according to their weight.
9. The maximum weight w_h will lead to the most match-able vertex \bar{v}_h that will be suggested to be a good friend for \bar{b} .

Now, to demonstrate the idea let us take an example. Assume we have an undirected graph $G(V, E)$, where V is a set of vertices such as $\{v_1, v_2, v_3, \dots, v_{21}\}$, and E is a set of edges and each edge connects v_i and v_j where $i \neq j$, as shown in Figure 1. Assume that each vertex v_i relates a person profile in Facebook, and an edge will exist between v_i and v_j if v_j and v_i are friends and v_j is in v_i 's friendship list and visa verse. Notice that in real life, such a graph will be very huge but for simplicity, we will use a graph with only 20 vertices to show the idea behind the proposed algorithm. Apply Rossi L & Musolesi M [19] algorithm that will give the identification of each person according to the check-in places he/she published them in using his/her Facebook profile. These person-identifications will be presented in a matrix M that will hold a row for each vertex v_i (person) and the columns will be filled by the person's identifications id_i . In this example, we will assume only 3 identifications, while we can use more algorithms to get more identifications such as people emotional states. If this person has its value as 1, number 1 will be replaced in $M[v_i, id_j]$. On the other hand, if vertex v_i (person) has not the identification id_i , then a 0 will be placed in $M[v_i, id_j]$ (as shown in Table 1 which shows the rows of the first 8 vertices)

Figure1 Graph $G[V, E]$ before coloring

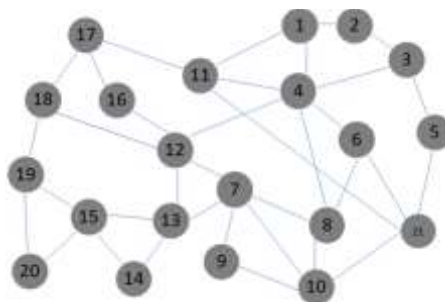




Table 1 First 8 person's identifications

Vertex No.	id1	id2	id3
1	1	1	1
2	1	1	1
3	0	1	1
4	1	1	1
5	1	0	1
6	0	1	0
7	1	1	1
8	0	0	0

Now we will apply Malkwai et al [14] coloring graph algorithm. This will color all vertices in the graph, such as each adjacent vertices are related to profiles of friends where there is an edge that connects them, with different colors. Consequently, if there are two vertices have no friendship, they will have the same color, as shown in Figure 2.

We will categorize the vertices according to their colors because all same colored vertices are non-friends according to the friendship list in their profiles. In this case, the algorithm colored the graph vertices by three colors, which are red, green and orange. This will be implemented in another matrix $L[C,Vno]$ that will hold a column for each color and the vertices number those have the same color of the column's label, as shown in Table 2.

Figure2 Graph $G[v,E]$ of 20 persons' profiles after coloring

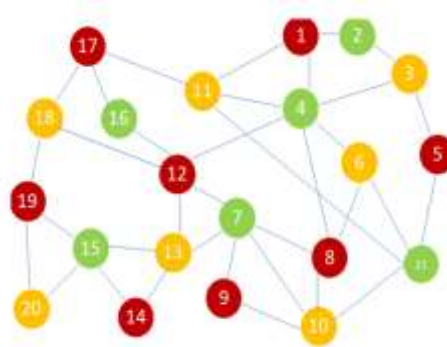


Table 2 The groups of non-friend people

Red	Orange	Green
1, 17, 5, 8, 9, 12, 19, 14	3, 6, 11, 10, 13, 18, 20	2, 4, 7, 16, 15, 21

At this point, we will check every vertex in each group with all group members. Remember that each group contains people who are non-friends. We will take each vertex and add it to matrix $N[v_i, v_j, w_i]$, where v_i is the person we will search for new friends for him/her, v_j is the person who is in the same group (i.e. there vertices colors are the same) and w_i (will be initialized by 0) is the weight of the edge connects v_i with v_j . The weight w_i will be increased by one if v_i and v_j have the same identification. Clearly, it will be at least zero and at most 3, as shown in Table 3.

After going over all vertices in the colored graph, we can see the friends to be suggested in each group. So let us take v_1 as an example and show the instance relates to it from matrix $N[v_i, v_j, w_i]$, where N is the matrix that holds the combination between vertices and v_i . We can decide which friends are the best choice to be friends, because they have more common identifications (ie. more weight value). This can be inferred by the weights of each pair of vertices. The more weight they have, the more common identifications are matched. Clearly, to suggest friends, we will choose the vertices connected with v_1 with the highest w_i value. However, these values can be sorted in descending order to pop-up the highest ones using a fast sorting algorithm such as the algorithm presented by Abu Dalhoun and his colleagues in [1].



Table 3 N[v1,vj,wi] Matrix of relation of v1 and the other vertices

v1	vj	wi
v1	v17	1
v1	v5	2
v1	v8	3
v1	v9	0
v1	v12	2
v1	v19	3
v1	v14	1

According to Table 1, v19 and v8 have 3 identifications in common with v1, which implies that we have to suggest them to be friends more than v9 who has no common identifications with v1. So, this algorithm provides us a solution for friend's recommendation to people who do not know each other but their identifications are common; that we believe they will be such good friends.

Note that the algorithm excludes all nodes with different colors than red one, who might have a high weight of identification as well. For example, v1 is not a friend with v7, v10, v13, v21 and others. In fact, the list of nodes with different colors is larger than the list of nodes with the same color (7 nodes with orange color and 6 green – total 13), while the nodes with red color are only 8). This means that the subset of nodes, which we investigate to suggest is $8/21 \approx 40\%$ of the total potential nodes (for this example).

IV. Algorithm Analysis

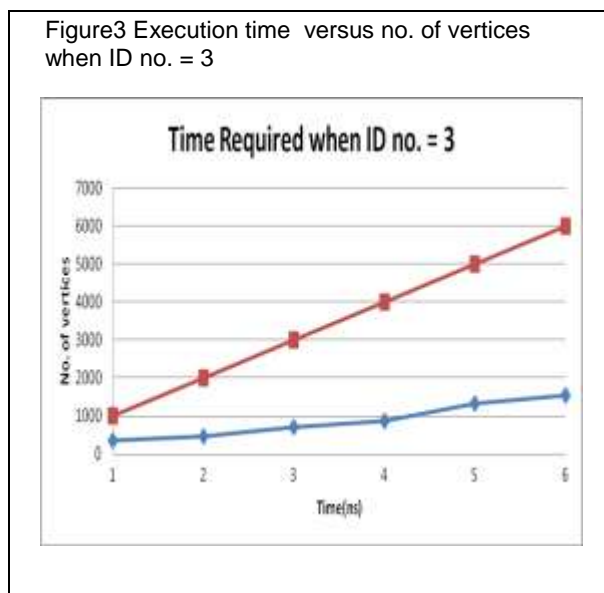
The time complexity of $M(G, \bar{b}, \bar{P})$ will be as follows.

Step number (1) will take a constant time $O(c)$ because it will just check the required person \bar{b} profile. Steps (2-4) will take $O(V+E)$ steps to draw the graph, which is needed to draw the required graph from the social network. Step (5) will take $O(V)$ steps as the coloring algorithm [14]. Step number (6) will take $O(c*V) = O(V)$, while step number (7) will take $\leq \bar{P} \leq V$ steps because in the worst case it will be $O(\bar{P})$ if \bar{b} has common identification with each $\bar{v}_i \in \bar{P}$. Step number (8), in the worst case, will take $O(\bar{P})$, that will happen when all $\bar{v}_i \in \bar{P}$ has similar identifications with \bar{b} . Last step will take $O(\bar{P})$ to get the maximum weight value. So the complexity for the whole algorithm will be $O(V+ \bar{P}) \approx O(2*V) \approx O(V)$ which can be considered as a linear complexity.

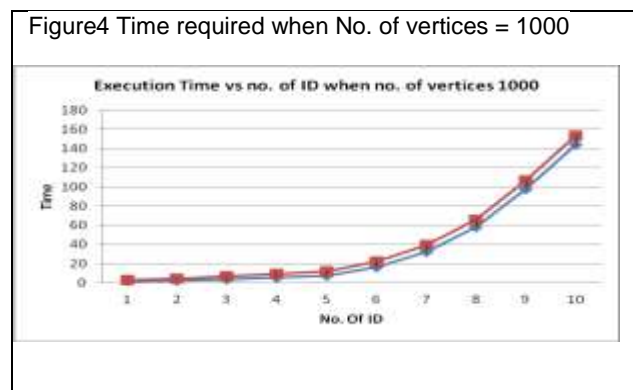
V. Experiments

In this section, we will present the experiments done for different number of identifications and different number of personnel profiles.

Figure 3, present the execution time required when the personnel profile defines three different identifications and show that the execution time is increasing whenever the number of vertices in a graph is increasing. We tested the algorithm with used up to 6000 vertices, although facebook rules do not allow more than 5000 people to be friends to a specific user.



In Figure 4 shows results when the proposed algorithm was tested using random identifications from 1 to 10 and, the graph has up to 1000 vertices. It is clear that whenever the number of identifications increase the time of matching increases.



VI. Conclusions

In this paper, we present an algorithm that can be used in social networks to help people to get new friends. These new friends will be chosen after matching both of the new friends identifications. We can use the proposed graph coloring algorithm to get people who are not currently connected with each other, although they will be perfect couples. The person who has more common identifications with the one we want to recommend friend for him will be the first choice to suggest his friendship. We tested the algorithm using random graph with 1000 nodes with up to 10 random identifications. The results show that the execution time increases proportionally the number of the identifications required for matching personnel profiles. Also, we tested the proposed algorithm with three different identifications and up to 6000 vertices. Absolutely, the more number of vertices, the more time required for matching the vertices.

References

- [1] Abu Dalhoum, A., Kobbay, T., Sleit, A., Alfonseca, M. & Ortega, A. (2012). Enhancing QuickSort Algorithm using a Dynamic Pivot Selection Technique. WULFENIA Journal, Austria. Vol 19, No. 10;Oct 2012
- [2] Al-Jaber, A., & Sharieh, A., - Algorithms Based on Weight Factors for Maximum Independent Set. Jordan: University of Jordan, 1999.
- [3] Brandes, P., & Wattenhofer, R. (2012). On finding better friends in social networks. In Stabilization, Safety, and Security of Distributed Systems (pp. 266-278). Springer Berlin Heidelberg.
- [4] Brélaz, D. (1979). New methods to color the vertices of a graph. Communications of the ACM, 22(4), 251-256.
- [5] Chalupa, D. (2011). On the Ability of Graph Coloring Heuristics to Find Substructures in Social Networks. Information Sciences and Technologies, Bulletin of ACM Slovakia, 3(2), 51-54
- [6] Chin, A., Xu, B., & Wang, H. (2013, May). Who should I add as a friend?: A study of friend recommendations using proximity and homophily. In Proceedings of the 4th International Workshop on Modeling Social Media (p. 7). ACM.
- [7] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms, 2009. Possíveis Questionamentos
- [8] Donderia, V., & Jana, P. K. (2012). A novel scheme for graph coloring. Procedia Technology, 4, 261-266.
- [9] Dunbar, R. (2010). How many friends does one person need. Royal Society for the Encouragement of the Arts, Manufactures and Commerce, London, 18.
- [10] Guy, I., Ur, S., Ronen, I., Perer, A., & Jacovi, M. (2011, March). Do you want to know?: recommending strangers in the enterprise. In Proceedings of the ACM 2011 conference on Computer supported cooperative work (pp. 285-294). ACM.
- [11] Hertz, A., & de Werra, D. (1987). Using tabu search techniques for graph coloring. Computing, 39(4), 345-351.
- [12] Karypis, G., & Kumar, V. (1995, December). Analysis of multilevel graph partitioning. In Proceedings of the 1995 ACM/IEEE conference on Supercomputing (p. 29). ACM.
- [13] Lee, D. H., & Brusilovsky, P. (2010, June). Social networks and interest similarity: the case of CiteULike. In Proceedings of the 21st ACM conference on Hypertext and hypermedia (pp. 151-156). ACM.
- [14] Malkawi, M., Hassan, M. A. H., & Hassan, O. A. H. (2008). A New Exam Scheduling Algorithm Using Graph Coloring. Int. Arab J. Inf. Technol., 5(1), 80-86.
- [15] Malkawi, M., & Murad, O. (2013). Artificial neuro fuzzy logic system for detecting human emotions. Human-Centric Computing and Information Sciences, 3(1), 1-13.

- [16] Salah, I., Sleit, A., Al-Sharaeh, S., Huneiti, A., & Obeed, N. (2005). Efficient Method for Assigning Students to Proper Groups. *Editorial Advisory Board e*, 21(2), 249-358.
- [17] Sleit, A., Abusharkh, S., & AlMobaideen, W. (2010). Enhancing Modularity-Based Graph Clustering. *World Applied Sciences Journal*, 9(9), 984-996.
- [18] Sharieh, A., Al Rawagepfeh, W., Mahafzah, M., & Al Dahamsheh, A. (2008). An algorithm for finding maximum independent set in a graph. *European Journal of Scientific Research*, 23(4), 586-596.
- [19] Rossi, L., & Musolesi, M. (2014, October). It's the way you check-in: identifying users in location-based social networks. In *Proceedings of the second ACM conference on Online social networks* (pp. 215-226). ACM.
- [20] Rossi, Rayan A, and Nesreen K. Ahmad. "Coloring large complex networks." *Social Network Analysis and Mining* 4, no. 1 (2014), pp. 1-37

Biography



Omayya Murad is a PhD. Candidate of computer science at The University Of Jordan. She received her MS degree from Middle East University, Jordan and BS from Petra University, Jordan. Murad was an editor in RCL (Risk Consult Limited), Jordan (2014-2015). Murad was a teacher at Dubai National School, United Arab of Emirates (2002-2007). The research interests of Murad include artificial intelligence and expert systems.



Prof. Azzam Sleit is the Former Minister of Information and Communications Technology, Jordan (2013-2015). He currently functions as the Dean of King Abdulla II School for Information Technology and a Professor of Computer Science with the University of Jordan. Before joining the University of Jordan in 2005, Dr. Sleit was the Chief Information Officer at Hamad Medical/ Ministry of Public Health, Qatar. Dr. Sleit has over twenty five years of experience and leadership working at all levels of government, private and international sectors. Before joining Hamad Medical, Dr. Sleit was the Vice President of Strategic Group & Director of Professional Services of Triada, USA, where he introduced the NGram Technology and Associative Memory Structures. The application of NGram technology helped Ford Motor Company to identify patterns of auto-parts failure and State of Michigan to recognize patterns of child abuse. Dr. Sleit served as the Regional Manager of Professional Services with Information Builders, USA. From 1993 to 1996, Dr. Sleit was in charge of MetSource, a strategic unit of Metlife responsible for providing outsourced IT health insurance services for large companies such as AT&T, ABB. Dr. Sleit has participated in numerous research activities related to Cloud Computing, Imaging Databases, Data Mining, Health and Management Information Systems and Software Engineering. He authored more than seventy refereed research papers published in reputable journals and conferences. Since 1987, Dr. Sleit has taught Computer Science courses at various universities in the United States and Middle East maintaining high teaching standards. Dr. Sleit holds B.Sc, M.Sc. and Ph.D. in Computer Science. He received his Ph.D. in 1995 from Wayne State University, Michigan.



Professor Ahmad Sharieh had two bachelor degrees: one in Mathematics and one in Computer Sciences. He had master degree in Computer Science and High Diploma in Teaching in Higher Education. He had PhD in Computer and Information Sciences from Florida State University 1991.

Sharieh worked as Assistant Professor in Fort Valley College (1991-1992 USA), The He worked as Associate Professor with Amman Arab University for Graduate Studies (AAUGS) (2003-2004 Jordan) and The University of Jordan (1992-2016 Jordan). He is a professor of Computer and Information Sciences at UJ.

He held several administration and academic positions: Chairman of Computer Science Department at UJ, Assistant Dean of Research Deanship at UJ, Chairman of Computer Science Department at AAUGS, Chairman of Central Tender Committee at UJ, Dean of King Abdullah II School for Information Technology (KASIT) at UJ, and Director of University Development Affairs at UJ, Dean (Executive President) of Sur University College/ Oman 2008-2015.

Prof. Sharieh initiated (as Coordinator or member) several academic and training programs: KASIT, Master degree in Computer Science, Master degree in Computer Information Systems, High Diploma in Information and Communication Technology in Education, E-government Diploma, Information Technology College at Amran University (Yemen) (Under establishment), PhD in Computer Science, and E-competence center at UJ, bachelor in Mechanical and Architecture Engineering; and Master of MBA at SUC.

He published about 60 scientific research articles in journals and conferences, and authored and prepared thirteen books. He gained grant for eight research projects from UJ and Europe. He developed several software systems such as: Teaching Sign Language, e-learning Modeling and Simulation, and Online (Automated) Exams. He is on the editorial board of several journals and conferences and a referee of several others. His research areas are in: Distributing Systems, Parallel Processing, Pattern Recognition, Software Engineering, Wire/Wireless Communication, Modeling and Simulation, and Cloud Computing.