Meaningful Learning through Identifying Differences and Similarities between Certain Problems and Algorithms

Malik Jahan Khan, Ashraf Iqbal, Yasser Hashmi
Lahore University of Management Sciences,
Lahore – Pakistan
{jahan, aiqbal, yasser}@lums.edu.pk

Abstract - **Some typical optimization problems like knapsack problem, activity selection problem, coin change problem and subset sum problem are taught in different courses of theoretical computer science, or in the different modules of the same course. A few of these problems can be solved in polynomial time but the rest are NP-complete problems. Even though, every problem has its own nature and description; however after changing certain constraints, they become equivalent to each other. These problems are discussed independently in most textbooks which makes it harder for the students to interlink them. Students get confused very easily while transitioning between different problem domains even if one problem is described in the language of another problem. In this paper, we have carefully analyzed and investigated links between the aforementioned problems and have also diagnosed the major causes of failure in understanding. One of the primary reasons students face difficulty is that they have a tendency to rote learn. We suggest that these problems and their algorithms should be taught simultaneously in a generic language using super-ordinate learning, so that they may be able to maintain the currently missing links. We assert that this effort will empower the notion of meaningful learning.**

Key Words - Algorithms, integrative reconciliation, meaningful learning, progressive differentiation.

INTRODUCTION

Learning new concepts is a complex but useful process. It may either lead to meaningful learning or rote learning. Meaningful learning builds strong concept maps [1], in which concepts are properly linked with each other. Rote learning fails during retrieval of knowledge from the existing concepts. If these concepts are not linked with each other properly then a lot of problems of reasoning, inferencing and problem solving arise [1]. In the course of “Analysis of Algorithms” taught at Lahore University of Management Sciences (LUMS), students are encouraged to devise and understand algorithms using discovery-based learning [2,3]. There are some traditional optimization problems in theoretical computer science like knapsack problem, activity-selection problem, subset-sum problem and coin change problem [4,5]. All of them apparently belong to entirely different domains. But if we investigate them in depth, then some similar patterns arise. Traditionally, in the classroom and various standard textbooks, these problems are discussed individually in different chapters and their similarities are overlooked. As far as solutions are concerned, these problems belong to different domains of problem complexity. Some of them can be solved easily and some are very hard to solve, also known as NP-complete problem. In algorithms, difficulty is usually measured in terms of time complexity. Some problems take polynomial amount of time and some take exponential amount of time. During the various offerings of the course “Analysis of Algorithms”, “Graph Theory and Algorithms” and “Computational Biology” [3], we have observed that students always suffer when they are asked to identify the similarities and differences between various algorithms. Also, if one problem is described in the language of a different problem, student fail to identify the underlying nature of problem and they focus more on the language of problem, overlooking the objectives and constraints of that problem. This leads them to entirely incorrect solutions [6,7,8]. We conducted comprehensive survey in the form of questionnaires, structured interviews and think-aloud protocol. Our intended subjects included students of undergraduate and graduate program who had taken at least one of the above mentioned courses. Besides the survey, our previous data in the form of exams and quizzes [3] also reflect a lot of problems to tackle such questions. After analyzing all these evidences, we present a panoramic way of teaching these problems by carefully identifying similarities and differences between these problems and their solutions, so that meaningful learning can take place. We also conclude that most of the time, students rote-learn these concepts in terms of the language of the problems, and they fail to identify the underling interesting hidden constraints and objectives which make the problem nature easy or complex [9,10,11,12,13,14].

PROBLEM DOMAIN

Knapsack problem, activity-selection problem, subset problem and coin change problem are typical optimization problems in the field of theoretical computer science and have a lot of practical implications [4,5]. All standard textbooks of this area discuss these problems and their solutions in isolated manner. Most of the time, learner is not able to capture their commonalities and differences. Here, we describe these problems as discussed in the literature:

I. Knapsack Problem

A thief robbing a house finds some items. Each item has a weight and some associated worth. Thief has a knapsack whose weight capacity is limited. Thief has to pick some items intelligently such that knapsack is filled no more than its capacity and total worth of the robbed items is maximum [4].

II. Activity-Selection Problem

The manager of a conference hall has to schedule various activities in that hall. Each activity has some defined start time and finish times. The hall cannot be used by more than one activities at a time. Total duration of the hall opening is also defined. Activities can be scheduled within that time only. Objective is to select some of the mutually compatible activities such that maximum number of activities may be selected. Mutually compatible activities are those activities which are non-overlapping with each other [4].

III. Subset-Sum Problem

We are given a set of integers, and a fixed integer, say target integer. We have to select some of the integers from the given set such that they exactly sum up to the target integer [4]. There are also some variants of this problem.

IV. Coin Change Problem

We are given some denominations of various coins and an amount, say target amount. We have to find change of that amount using the coins given in the collection of denominations such that minimum number of coins is used to generate the change [5]. There are also some variants of the same problem, like we may be given finite supply of the coin denominations, or we may be given infinite supply of the coin denominations.

MEANINGFUL LEARNING

Learning is a continuous process which adds new information to the existing information repository. If learner is interested to relate new information with the existing one instead of just adding it, then he is attempting meaningful learning. When knowledge structures are properly organized then it enhances meaningful learning [1,15]. If one concept is introduced and learner feels difficulty to relate it with the existing knowledge, then perhaps he is being pushed towards rote learning. When perceived regularity of an event or object is quite clear in the mind of the learner, then he may feel comfortable to integrate it within the existing knowledge structure. Progressive differentiation is a continuous process which refines the concept meanings and associated regularities in cognitive structure with the addition of new knowledge. Sometimes, existing concepts don't need to be refined contextually, but they are reconciled in an integrative fashion with the new concepts to enrich the cognitive structure or concept map. In the process of meaningful learning, direct integration of the new concepts with the existing ones is a very complex procedure. There are some perceptual barriers which resist this integration. These barriers can be easily and effectively crossed, if there are some facilitating concepts called subsuming concepts, in Ausubel's assimilation learning theory [1]. To empower meaningful learning, existing concepts and new concepts should be properly linked. Teacher and learner both should focus on creating these links. Creation of these missing links is a continuous process which removes the perceptual barriers.

TEACHING ALGORITHMS

It has been frequently observed that most of the teaching resources discuss and teach various related problems and their solutions, independently. When the learner is learning one of the problems, he only concentrates on that problem because no learning support or motivation is provided to think it in a generic way. This leads to rote learning the concepts [7,8,15,16]. These problems can be discussed in generalized ways instead of just focusing on a single example. When a learner is forced to focus on a particular example like activity-selection problem, language (vocabulary) of that problem hinders meaningful learning and learner concentrates more on words of that problem instead of realizing the objectives to be achieved and restrictions imposed by the problem. We gathered data in various forms to support these claims. Various examinations and quizzes of the course "Analysis of Algorithms" [3], questionnaires, structured interviews and sessions of think aloud protocol support our claims. Students who had taken this course and know the definitions of aforementioned problems get easily confused if any of these problems is described in a general way or its vocabulary is altered. For example, the subset sum problem can be described as: "We have to schedule some activities in a hall. Each activity has a fixed duration, but there is no bound on start time and finish time of activities. We have to select some of the activities such that hall is 100% utilized." If we carefully look at this description, it exactly maps to the description of subset-sum problem. But if rote learning takes place in the mind of learner, then he will try to map it to activity-selection problem with minor modification because of the vocabulary used like "activities", "hall" etc. So, if he attempts in this way, it will lead him to incorrect solution because activity selection problem and subset sum problem belong to entirely different domains of problem complexity. Activity selection problem can be solved in polynomial amount of time and subset sum problem can only be solved in exponential amount of time. Though both the problems seem similar apparently, but both of them need entirely different solution strategies. In the rest of the section, we will discuss and present analysis of the data, we gathered to witness such problems in learning and their solution to enhance meaningful learning.

I. Structured Interviews

We conducted structured interview containing four questions. Each question was described in the vocabulary of activity-selection problem. We also provided a list of names of well-known problems and asked the audience to map the description to one of the known problems. Our subjects included 25 students of undergraduate and graduate program who had taken the course "Analysis of Algorithms" and were familiar with the standard definitions of these problems at least. Following list shows the questions asked from every one:

1. We are given *n* activities (with given fixed start time and fixed finish time). All these activities may be conducted in a single hall which is available from morning to evening, both times fixed. The objective is to select (some of) the activities so that maximum number of compatible activities can take place in the given hall. What is the idea to achieve this objective?
2. We are given *n* activities (with given fixed start time and fixed finish time). All these activities may be conducted in a single hall which is available from morning to evening, both times fixed. The objective is to select (some of) the compatible activities so as to maximize the sum total utilization of the hall. What is the idea to achieve this objective?
3. We are given *n* activities (with no fixed start time and finish time). All these activities may be conducted in a single hall which is available from morning to evening, both times fixed. The objective is to select (some of) the activities so that maximum number of activities can take place in the given hall, while not exceeding the hall availability. What is the idea to achieve this objective?
4. We are given *n* activities (with no fixed start time and finish time). All these activities may be conducted in a single hall which is available from morning to evening, both times fixed. The objective is to select (some of) the activities so that hall is 100% utilized. What is the idea to achieve this objective?

Figure 1 shows the results of these questions:

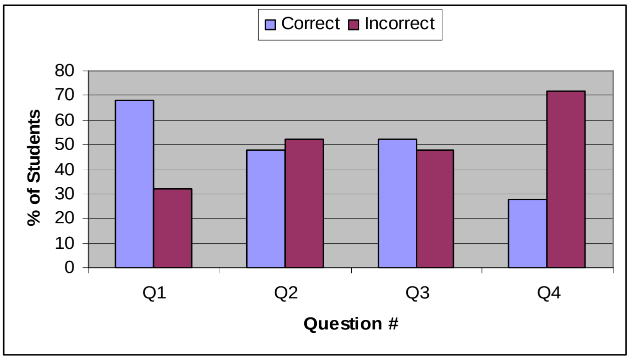

FIGURE 1
RESULTS OF STRUCTURED INTERVIEWS

II. Sessions of Think Aloud Protocol

We conducted 5 sessions of think aloud protocol. Two important issues were observed during these sessions:

1. In the first attempt, people think in terms of language of the problem instead of its structure which makes them away from the correct solution while mapping a given problem to a known problem. If mapping is provided, they confess their incorrect approach. But they fail to realize the complexity of the solution and overlook the key causes which control the complexity of the solution. Some problems are so simple that they can be solved using a greedy approach, some problems need dynamic programming to be applied and some problems are very hard to solve exactly. For such hard problems, approximate solutions are found.
2. They utter mistakes while they are convinced that problem is different and its language is misleading. They fail to properly map the problem from one domain onto an equivalent problem in some other domain. This observation also supports that there are some missing links in their concept maps, highlighted in figure 6. If we help them to add those links, they feel comfortable and may avoid mistakes.

III. Students Feedback Forms

During the previous offering of the course "Analysis of Algorithms" [3], we conducted anonymous feedback sessions at the end of every lecture. In these sessions, students had to fill up a form to mention the concept well-understood and concepts not understood. At the end of the course, we analyzed the data collectively lecture-wise, and the results shown in the figure 2 were obtained. Overall learning was good, and number of concepts well-understood remained much higher that those not understood. From lecture 9 to 12, a disrupting pattern came into scene, and when we investigated back then we found that in these lectures, aforementioned optimization problems and their variations were discussed along with their recursive formulations. Significant ratio of the students failed to understand and interconnect these problems. This increased frequency of misunderstood or poorly understood concepts demands the researchers to properly investigate the underlying problems in learning.

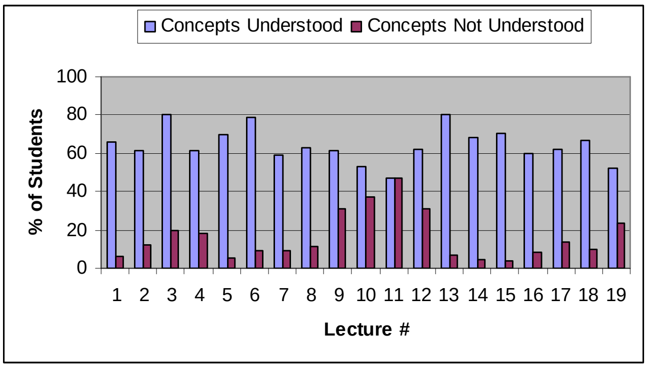

FIGURE 2
RESULTS OF FEEDBACK COLLECTED AFTER EVERY LECTURE

IV. Examinations and Quizzes Data

We also analyzed the questions of examinations and quizzes of the same course [3] which were related to these problems. Two quizzes were conducted and 51 students participated in them. Quiz 1 was described in terms of the activity-selection problem but it was actually imposing some strict constraints and it was going to be equivalent to subset-sum problem. In this quiz, students performed poorly. Quiz 2 was described in terms of the activity-selection problem and it was a variation of the activity-selection problem. Therefore students attempted it in relatively better way. Figure 3 shows the results of quizzes.

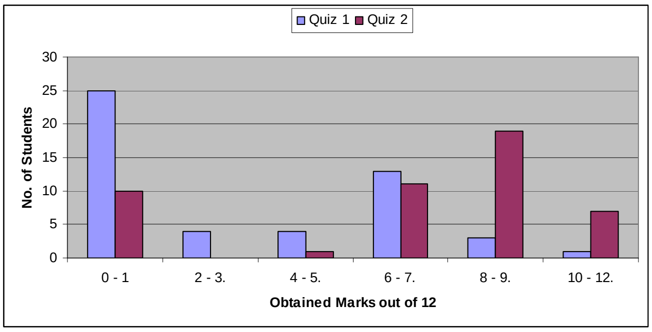

FIGURE 3
RESULTS OF QUIZZES

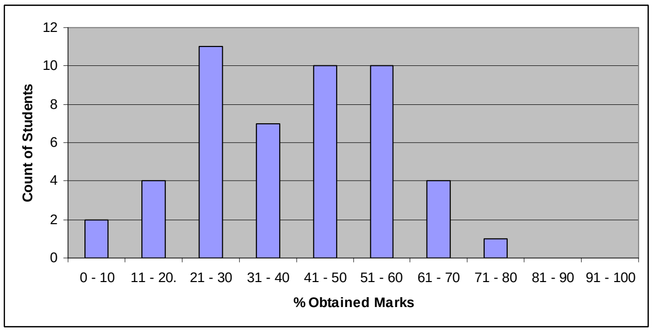

FIGURE 4
RESULTS OF EXAM

In one of the midterm examinations of the same course [3], eight questions were asked. All of them were described in the language of activity-selection problem, but they were attacking entirely different kinds of problems. Figure 4 represents the percentage marks of all students.

OUR WORK

After analyzing the whole data, it is quite obvious that there are some serious problems in learning these concepts. Though students know all these concepts individually, but they miss some very important links between those concepts. These missing links have been highlighted in figure 6. In this section, we propose a panoramic way of teaching and learning such problems and discovering their algorithmic solutions. We think that this way of teaching and learning will enhance meaningful learning.

I. Progressive Differentiation

Learning is a continuous process and as this process proceeds, cognitive structure becomes more and more precise and strong [1]. At the early stages of learning, some concepts may be poorly or vaguely understood. So, their perceived meanings are also vague. With the acquisition of new knowledge, those vague meanings start becoming more precise and specific. So, in this continuous learning process, concepts progressively differentiate, and after certain period of time, they achieve a desirable level of specificity [1].

In our proposed model of learning and teaching algorithms, we suggest to keep the concepts very vague at initial stages of learning, so that learner is forced to remove ambiguities from these concepts and make them more specific. Specifically, when any of the aforementioned problems are discussed, learner should not be spoon fed with the description of these standard problems at earlier stages. Start teaching using generic terms, e.g. object selection problem.

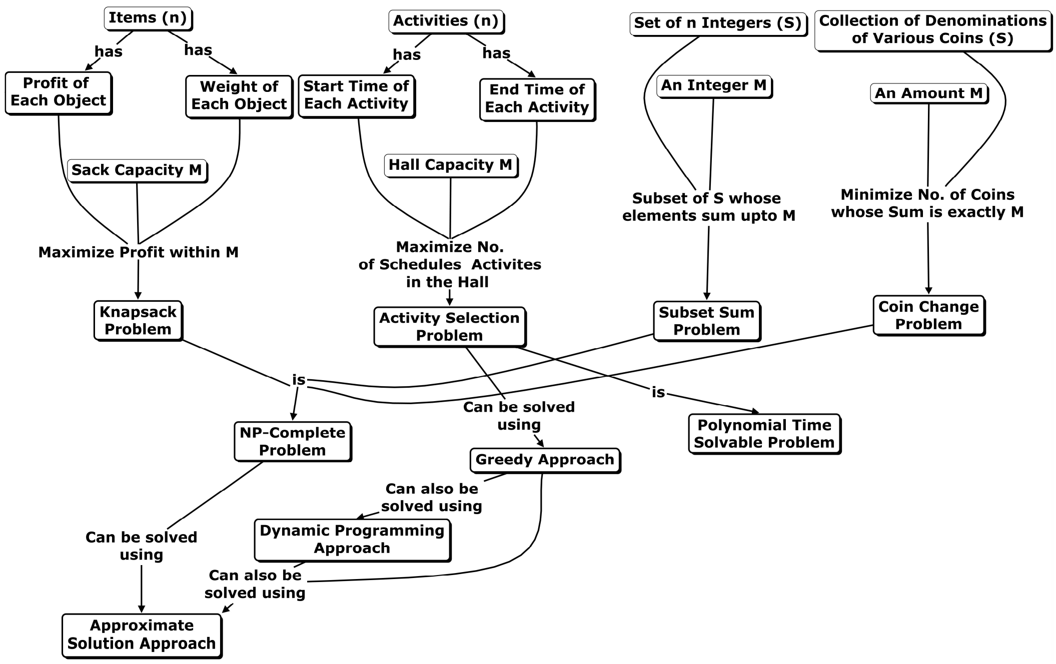

FIGURE 5
WEAK CONCEPT MAP

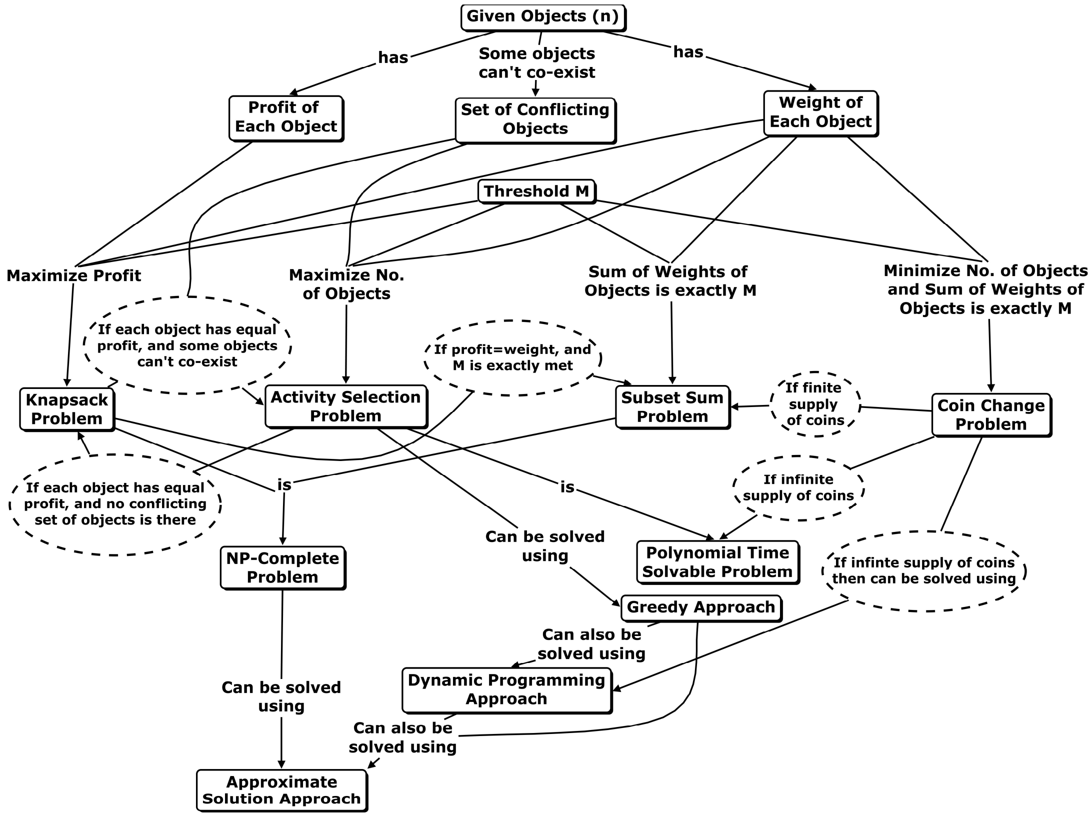

FIGURE 6
EXPERT CONCEPT MAP

Here, objects are not defined. Later on, these objects may be specified as activities, items to be robbed, some integers or coins. Similarly, when we talk about object selection, no constraints or objectives have been mentioned. Let them progressively differentiate in the cognitive structure of the learner.

II. Integrative Reconciliation

Learner acquires new concepts and integrates them with the existing concepts through some meaningful linkages. Sometimes, existing links may need to be modified. During teaching, when object selection problem is presented as: “Select some objects such that their total weight doesn’t exceed the given threshold, and total profit is maximized”. Learner is moving towards knapsack problem. If this knowledge exists and now learner is posed a different problem: “Select some objects such that threshold is met to maximum possible level and profit of each object is the same as its weight.” Now, this is the crucial perceptual barrier. In conventional teaching, this new posed problem is taught in such a way that learner fails to link it with the previous problem, because, this problem is a variation of activity-selection problem. At this point, integrative reconciliation [1] is desirable otherwise meaningful learning will be harmed. Teacher should take the previous problem and make appropriate transition to transform into this problem, so that student is able to link the previous concepts with this new set of concepts, as highlighted in the expert concept map of the figure 6.

III. Concept Assimilation

There are two broader categories of the concepts [1]: primary concepts and secondary concepts. Primary concepts are the concepts involving core knowledge. Secondary concepts are supporting concepts which help to acquire new concepts. Ausubel’s assimilation learning theory [1] names these secondary concepts as “subsuming” concepts. Subsuming concepts facilitate flow of knowledge though the perceptual barriers and provide a platform so that new concepts may be linked with previous concepts [1,17]. Dotted links in the expert concept map of the figure 6 are the subsuming concepts, which are missing in a weak concept map, as shown in figure 5.

IV. Panoramic Picture: Super-ordinate Learning

In our proposed model of teaching these problems, we suggest to introduce some super-ordinate concepts, so that commonalities between these problems become apparent in the cognitive structure of the learner. All problems are actually optimization problems which have to achieve certain objectives in different practical domains. Followings are the salient guidelines to empower knowledge through meaningfully learning these problems and their algorithms:

1. Pose the problem in generic terms, like object selection problem, shown in expert concept map of figure 6.
2. Define the attributes of objects, like weight, profit etc, and mention the restriction of certain total weight threshold.

3. Highlight that we have to optimize our selection procedure to achieve certain objectives under certain constraints.
4. Now, start defining constraints and objectives of one of the problems, say knapsack problem.
5. Once the learner is comfortable in understanding the problem, then as an example describe the traditional knapsack problem in its own language. So that, he may realize a practical application of this constrained object selection problem.
6. Now, change the constraints and objective of object selection problem, and ask the students to transform the previous problem into this new problem. This transformation done by the learner with the involvement of the teacher as a facilitator enhances progressive differentiation and integrative reconciliation. If this transition phase is carried out by the learner, then he will have better insight of the changing complexity domains.
7. Repeat this process for each problem and keep on progressively integrating it with the existing cognitive structure. Eventually, this process will end up with a panoramic picture as shown in the expert concept map of the figure 6. In this paradigm, general terms like objects act as super-ordinate concepts. This paradigm is very similar to discovery-based learning. In this paradigm, learner discovers the missing or bridging concepts himself with little bit support of the teacher. So, this is a discovery-based learning which uses super-ordinate learning at its core.

CONCLUSION

In this paper, we highlighted some problems and perceptual barriers of meaningful learning. We observed these problems in various offerings of the theoretical computer science courses like Analysis of Algorithms and Graph Theory. To support the claims of existence of these problems, we used data of quizzes and examinations, student feedback, structured interviews and think aloud sessions. In conventionally teaching, if we use an example to teach a problem, then students get involved in the vocabulary of that example and take less of care of the underlying constraints and objective functions which really matter to make a problem trivial or hard. Finally, we suggested a panoramic way of teaching and learning these problems and their algorithmic solutions which is very close to discovery-based learning. Instead of using the example-based teaching method, we should discuss problems at an abstract level. This abstraction may be introduced by discussing all such problems in a common abstract language. It will prevent students to think more in terms of the problem's language and will help them to think more about the constraints and objectives of the problem.

ACKNOWLEDGMENT

This research was supported by Higher Education Commission, Pakistan and Lahore University of Management Sciences, Pakistan. We gratefully acknowledge this support.

REFERENCES

[1] Novak, J., D., "Leaning, Creating and Using Knowledge", Lawrence Erlbaum Associates, Inc., 1998.

[2] Polya, G., "Mathematical Discovery: On Understanding, Learning, and Teaching Problem Solving", John Wiley and Sons, 1981.

[3] Iqbal, M., A., "Course Material of Analysis of Algorithms", Lahore University of Management Sciences, Pakistan, 2006.

[4] Cormen, H., T., Leiserson, E., C., Rivest, L., R., Stien, F., C., "Introduction to Algorithms", 2nd Edition. The MIT Press, 2001.

[5] Jones, N., C., Pevzner, P., A., "An Introduction to Bioinformatics Algorithms", The MIT Press, 2004.

[6] Iqbal, M., A., Hashmi, Y., "Bridging over Problems of Learning in Finding Strongly Connected Components", In Proc. of Cognition and Exploratory Learning in Digital Age (CELDA), 2005, pp. 78-87

[7] Iqbal, M., A., Tahir, S., "Should We Teach Algorithms", Iranian Journal of Electrical and Computer Engineering, Vol. 2, No. 2, 2003, pp. 134-140.

[8] Iqbal, M., A., Tahir, S., "Do not Teach Algorithms", IEEE International Multitopic Conference (INMIC), 2001. pp. 82-86.

[9] Orton, A., "Learning Mathematics", 3rd Edition, Continuum Publishers, 2004.

[10] Cooney, T., J., Hirsch, C., R., "Teaching and Learning Mathematics in 1990s", The National Council of Teachers of Mathematics, 1990.

[11] Rishel, T., W., "Teaching First: A Guide for New Mathematicians", The Mathematical Association of America, MMA Notes # 54, 2000.

[12] Meier, J., Rishel, T., "Writing in the Teaching and Learning of Mathematics", The Mathematical Association of America, MMA Notes # 48, 1998.

[13] Iqbal, A., I., Hashmi, Y., "Meaningful Learning of Graph Algorithms", Oral Presentation in 3rd International Conference on Teaching in Mathematics at Undergraduate Level, 2006.

[14] Mustafa, M., Qadeer, K., Iqbal, M., A., Hashmi, Y., "Problems in Learning Graph Algorithms", In Proc. of Cognition and Exploratory Learning in Digital Age (CELDA), 2005, pp. 401-404.

[15] Kamii, C., Dominick A., "To Teach or Not To Teach Algorithms", Journal of Mathematical Behavior, Vol. 16, No. 1, 1997, Elsevier Science, pp. 51-61.

[16] Iqbal, M., A., "Problems in Learning Theoretical Computer Science", In Proc. of 1st IEEE International Conference on Information and Communication Technologies (ICICT), 2005, pp. 68-68.

[17] Polya, G., "How to Solve It: A New Aspect of Mathematical Method", Princeton University Press, 1988.