Delayed Guidance

A teaching-learning strategy to develop ill-structured problem solving skills in engineering

Aditi Kothiyal
IDP in Educational Technology
IIT Bombay
Mumbai, India
aditi.kothiyal@iitb.ac.in

Bipin Rajendran
Dept. of Electrical Engineering
IIT Bombay
Mumbai, India
bipin@ee.iitb.ac.in

Sahana Murthy
IDP in Educational Technology
IIT Bombay
Mumbai, India
sahanamurthy@iitb.ac.in

***Abstract*—Problem solving is an important skill for engineering graduates to develop. However, in most traditional engineering classrooms, students practice solving well structured problems which is not sufficient because engineers need to be able to solve real world problems which are ill structured. Therefore it is important for engineering students to be trained in ill-structured problem solving. In this paper, we present a new teaching-learning strategy for ill-structured problem solving based on productive failure and instructor prompts. Students do unguided in-class exploration of engineering problems in groups, after which they receive instruction on problem solving strategies. We conducted a pilot study by incorporating our teaching-learning strategy in a senior electrical engineering course. We found that students developed several ill-structured problem solving skills and showed a range of problem solving behaviors and heuristics.**

I. Introduction

Problem solving is an important ability for engineering graduates to develop [1]. Engineering workplace problems are ill-structured and involve design, troubleshooting, system analysis, decision-making etc [2]. They have characteristics such as incomplete descriptions, multiple and conflicting goals and multiple solutions and solution paths [2]. In most engineering classrooms, students practice solving well-structured problems which are different from ill-structured engineering problems. It has been reported that the skill of solving well-structured problems does not translate to ill-structured problems [2]. Therefore engineering students need to be explicitly trained in ill-structured problem solving.

Ill-structured problem solving is a hard cognitive skill to learn because it requires competence in the sub-skills of problem space construction, generating possible solutions, constructing arguments for solutions, evaluating and monitoring solutions and assessing alternate solutions [3]. It has been recommended that problem solving be embedded in existing engineering courses and class time be spent solving ill-structured problems so that domain knowledge and problem solving reinforce each other [4]. There are several inquiry and problem based methods [5],[6],[7] which go beyond making students solve well-structured problems using known procedures and push students to solve problems beyond their current ability. The salient features of these methods are inquiry centered around an ill-structured and complex problem solving task that students solve collaboratively. The timing and type of scaffolding varies depending on the particular method. Methods like these are desirable for ill-structured problem solving training [2].

Current methods for ill-structured problem solving training in engineering include problem solving courses based on some of the above mentioned methods such as project-based learning (PjBL) [8], [9] and problem based learning (PBL) [10],[11], and training in specific ill-structured problem solving skills [12], [13]. These methods require extensive curriculum redesign [2] and typically include large engineering problems [14] which are highly ill-structured [3]. Often, students work on one problem over several weeks spread across the semester.

There are other types of engineering problems which are moderately ill-structured [3] in that the discipline/domain on which the problem is based is known to the students. These problems are routinely part of the large engineering problems described above. Solving these types of problems involve several of the skills required for solving highly ill-structured problems. In this paper, we propose and evaluate a classroom teaching-learning strategy that embeds moderately ill-structured problems in existing engineering courses for the development of these ill-structured problem solving skills as recommended by [4] and [2] among others. However since these are moderately ill-structured, they can be solved in the duration of one lecture of one-and-a-half hours. Our intention is to develop students' ill-structured problem solving skills by solving these moderately ill-structured problems and prepare them to solve large engineering problems in the future.

Our teaching-learning strategy, called Delayed Guidance or DG, is based on unguided student exploration of moderately ill-structured problems in groups followed by problem solving instruction. Similar methods [15],[7] have been show to be effective in engineering design and ill-structured problem solving. In the DG strategy, the instructor provides students an ill-structured problem to solve. Students work in groups, discussing various possible methods to solve the problem for $30 - 40$ minutes. No guidance, clarifications or hints are given at this stage and the instructor only provides affective support. After $30-40$ minutes, the instructor discusses groups' solutions with the whole class and gives them problem solving question prompts and strategies which students then practice on other problems.

Our broad research goal was to get an in-depth understanding of the effect of DG on the activity of ill-structured problem solving. We conducted a pilot study in a senior electrical engineering course with one group receiving direct instruction (control) and the other receiving DG intervention (experimental). The instructor received a short training on the DG strategy at the beginning of the semester. We conducted an ill-structured problem solving post test and detailed student interviews regarding their problem solving strategies. The student solutions were content analysed to identify the development of ill-structured problem solving skills and the interviews were transcribed and content analysed to identify the problem solving strategies used by students.

The results show that students in the DG group showed statistically significantly higher measures of some ill-structured problem solving sub-skills. Student interviews showed that students in the experimental group showed several problem solving behaviours not present in the control group, while offering insights into behaviours that could be improved. Thus, our pilot study demonstrates that DG is a promising teaching-learning strategy for incorporating moderately ill-structured problems in engineering courses and developing students' ill-structured problem solving skills and sub-skills.

II. Background and Related Work

According to the ABET accreditation criteria for engineering programs, problem solving or "an ability to identify, formulate, and solve engineering problems" [1] is one of the eleven outcomes that all graduating engineers should possess. By engineering problems, we mean problems that are ill-structured, are aggregates of well-structured problems, have multiple and often conflicting goals and could be solved in several different ways. Their success is rarely measured by engineering standards, they have non-engineering constraints and problem solving knowledge is distributed among team members, requiring extensive collaboration. Further, engineers rely primarily on experiential knowledge while solving them, often encounter unanticipated problems, use using multiple forms of representation and require good communication skills among team members [2].

We observe that domain and structural knowledge alone is not sufficient to solve these types of problems. Students need a repertoire of problem solving heuristics, metacognition and epistemic cognition, argumentation skills, thinking skills, representation practices, persistence, motivation, the ability to deal with ambiguity, and experience [3]. In most engineering classrooms, however, the emphasis remains on well-structured problems which students practice solving using concepts and familiar procedures. Ill-structured problem solving is mostly relegated to the last few weeks of a course where students do a project, or separate problem solving courses, even though it has been recommended that instructors include problem solving instruction in the classroom, with different kinds of problems within existing engineering courses [4], [2].

Ill-structured problems are solved by a process which includes steps like constructing the problem space, identifying the stakeholders and constraints, generating possible solutions, evaluating solutions and choosing one based on argumentation and beliefs, and finally implementing, monitoring and revising the solution if necessary [16]. Proficiency in all these aspects of ill-structured problem solving skills are necessary to become a good problem solver.This is because, even though not all the steps are needed to solve all problems, competence in all the steps of the process and following a systematic method when confronted with a new problem is a hallmark of expertise in problem solving [17]. Therefore engineering students need to develop these ill-structured problem solving skills via instruction and practice [12].

There are several research-based strategies for teaching engineering students ill-structured problem solving, some of which are detailed in [12], [8], [10],[11],[9], [18]. The majority of this work deals with incorporation of PBL and PjBL in engineering classrooms [14], [19] because the salient feature of these pedagogies is a problem or project as a means of organizing student learning of concepts and skills, emphasis placed on the learning process, self-directed learning and collaboration [14]. However there are differences between these two pedgagogies; projects tend to be closer to real life engineering problems, often take months to complete and result in concrete outputs such as a product or a report. Most importantly, projects are directed more towards the application of knowledge and learning of transferrable, professional skills while PBL is directed towards acquisition of the theoretical concepts on which engineering is based [19]. Depending on the specific learning goals of the instructor, course and college, the scope of PBL and PjBL can range from one unit to one course to the entire curriculum [19].

Even among the PBL/PjBL teaching-learning methods, there have been variations in which students receive instruction in improving specific PS skills concurrently with, or prior to, attempting the PBL/PjBL problems [12], [20]. This is necessary because PBL/PjBL assumes that students possess or will develop in the process of solving, the necessary problem solving skills; and this is not necessarily true [21]. What is needed is to identify target behaviours and give opportunities to students to practice the skills and receive feedback until they have achieved mastery [21]. Using question prompts as scaffolds while students solve ill-structured problems has been found to improve students ill-structured problem solving [22]. These scaffolds have also been applied in engineering to improve students' ill-structured problem solving skills [18], [23].

However, rather than providing these scaffolds immediately, delaying guidance to students has been established as an effective strategy for improving conceptual understanding and problem solving skills in various teaching-learning methods such as "A time for telling" [6] and"Productive failure" [7]. These methods are based on the theories of constructivism, collaborative and active learning and students explore the domain via case studies or ill-structured problems without direct instruction at the beginning. Students are pushed to do a task which is just outside his/her current cognitive ability. This provides an opportunity for the student to apply and prepare their prior or implicit knowledge on the subject, which can be built upon in the following form of feedback or instruction. Further, in the case of productive failure [7], attempting to solve ill-structured problems and "failing" enables learners to generate and refine several representation and solution methods for ill-structured problems. Such an effect was also found in a study of engineering design skills [15] wherein it was found that students who "failed" at an early decision making task and received feedback, ultimately performed better at problem structuring than students who did not fail initially. This shows that initial "failure" can be productive in the learning of an engineering design skill, when it is followed by appropriate feedback to the student after the "failure" has occurred. In other words, it is useful for students to not be guided or scaffolded from the start. We describe our teaching-learning strategy based on these theories in the next section.

III. Delayed Guidance strategy

We developed a teaching-learning strategy called delayed guidance (DG) for the development of students' ill-structured problem solving skills based on the theories of productive failure [7] and question prompts [22]. Initial "failure" while solving ill-structured problems in "Productive Failure" [7] and in a decision making task while learning engineering design skills [15] is useful because it make students aware of the abilities they possess (or not), and prepares them for the instruction or feedback that is to follow. We seek to leverage this awareness and preparedness that the student develops after exploring an ill-structured problem for a fixed time and provide instructor prompts to guide them in subsequent ill-structured problem solving. We are targeting domain-focused, moderately ill-structured problems [3] as they can be solved in a class of one-and-a-half-hours. Thus, it is possible for instructors to repeat this intervention as often as possible in the duration of their course, thereby giving students practice in developing ill-structured problem solving skills [12], [21].

DG is in spirit like PjBL in that student learning is organized around a moderately ill-structured problems and the goals are knowledge application and development of skills; where we differ is the manner in which we apply the strategy. Firstly, we scaffold the development of ill-structured problem solving skills via instructor prompts as students work on ill-structured problems. Secondly, we introduce a mandatory delay in guidance as a way to prepare students to effectively use these instructor prompts. During this time students are not allowed to use any scaffolding apart from collaborating with their peers. We are hence able to leverage the benefits of inquiry and scaffolding simultaneously. A DG session consists of four stages adapted from [7] and is described below.

A. *Group ill-structured problem solving*

During the first stage, students work for 30-40 minutes in predefined groups of 3 or 4 on a moderately ill-structured problem. The students are not allowed to consult the text book, the Internet or the instructor during this stage. They may however refer to their own previous notes and perform simulations on their laptops if they feel it necessary. The idea is for students to attempt the problem with whatever knowledge and experience they already have, which allows for in-depth exploration of the problem space as they face difficulties in completing the solution. The instructor walks around observing student work and motivating them to keep working, without giving any hints related to the domain or the process and not answering any student questions at this stage. This is stage for which the instructor needs a small training to hold back from guiding students in problem solving. The essential learning activity in this stage is group discussion which allows students to explore the problem space together, clarify goals and constraints, and develop and refine solutions via argumentation and articulation. This stage is expected to develop students' skill in problem space construction and solution development.

B. *Class discussion of student solutions*

During the next stage which lasts 15-20 minutes student groups present their solutions which the instructor summarizes on the board for everyone to see. When all the groups have presented their solutions, the instructor encourages all the groups to comment on the pros and cons of each others' solutions which leads to a rich, intense debate interspersed with student questions and instructor responses. At an appropriate time when the instructor feels that the solutions have been explored sufficiently and no new information is being added by the debate, he/she moves to the next stage. The essential learning activity in this stage is class discussion by way of pros-cons analysis and argumentation. Students are expected to develop the skill of evaluating solutions and assessing alternative solutions at this stage.

C. *Instructor prompts*

In this stage, the instructor provides scaffolds for ill-structured problem solving processes contextualized to the domain of the ill-structured problem from stage III-A that he wishes the students to learn. Examples of these prompts follow in the next section. This stage lasts 15-20 minutes and is interspersed with student questions, instructor responses and class discussion. The essential learning activity for students is to apply these prompts to their solutions and refine them. The students are expected to learn the skill of revising and monitoring solutions at this stage.

D. *Practice problem*

In this stage, students individually work on another similar ill-structured problem from the domain. This stage is 20 minutes long and while group problem-solving is discouraged to ensure individual practice, the instructor walks around answering student questions. The essential learning activity for students is to apply the instructor prompts and the skills learned in the previous stages and practice ill-structured problem solving. The students are expected to refine and develop all the ill-structured problem solving skills learned in the first three stages.

IV. Implementation

A. *Course Format*

The setting for our study was a senior undergraduate/graduate elective course called Neuromorphic Engineering with an enrollment of 36 students. The course ran for 15 weeks during the Autumn 2013 semester at our institute and included lectures, tutorials, literature review and a project. The class met twice a week for an hour-and-a-half and the instructional objectives for the course were:

1) Develop engineering intuition about mechanisms of neuronal and synaptic interactions.
2) Understand how learning/adaptation arises in biological systems.
3) Understand strategies for implementing learning circuits in software/hardware.

Our teaching-learning strategy was applied by the instructor during problem solving tutorials which were conducted once

How much did the Germans know? - a WWII mystery

- During World War II, the city of South London was hit by 537 bombs. For the sake of analysis, the total area of the city was divided into 576 squares, each with area $1/4$ sq. km.
- The statistics of the actual hits/square is below:

Number of hits/sq	Number of squares
0	229
1	211
2	93
3	35
4	7
5	1

- Looking at this statistics, can you tell if the Germans were targeting specific areas of if they were bombing the city without any prior information?
- Can you test your hypothesis by a computer simulation ?

(a) Example of a moderately ill-structured problem

An LIF neuron receives 10 independent stimulii through 10 synapses of identical strength ($w_e = 1$). The arrival rates at the 10 synapses linearly increase from $0.01/\text{ms}$ to $0.05/\text{ms}$. A spike at time t_k results in a current flowing into the neuron given by the expression
$I_{app,k}(t) = I_c w_e \left[e^{-(t-t_k)/\tau} - e^{-(t-t_k)/\tau_s}\right] h(t - t_k)$
Where I_c is the minimum DC current necessary to initiate spikes in the neuron, and $\tau = 15$ms and $\tau_s = \tau/4$.
Your task is to design a neural circuit such that an output spike is generated only when there is a persistent rising edge in the total current arriving at the above input neuron.
Quantify the performance of your network by calculating the number of false positives and false negatives at the output of the neuron A for a total simulation time of 10 sec. You are free to come up with a suitable definition for the term "persistent rising edge".

(b) Example of a practice problem

Fig. 1: Examples of problems used in the intervention

a month by designating the regular lecture as a tutorial. The instructional goal for these tutorials was improving students ill-structured problem solving skills using moderately ill-structured problems from the domain of Neuromorphic Engineering.

B. Learning Materials

1) Moderately ill-structured problem: The first ill-structured problem presented to the students should be complex enough that the students are unable to solve in the allotted 30-40 minutes while at the same time interesting enough that they remain cognitively engaged for the entire duration discussing with each other and attempting to find a solution. This cognitive engagement will serve to prime their prior knowledge and skills and prepare them to receive the following instruction. An example of such a moderately ill-structured problem in the domain of Poisson distributions, is shown in Figure 1a.

C. Instructor prompts in stage III-C

The prompts were designed according to the instructional design given in [16] in order to support all the stages of ill-structured problem solving. As an example, for the ill-structured problem of designing neural networks to compute, the prompts shown in Table I were used.

In addition to this material, the instructor also provided students with a `MATLAB` program to test and optimize their design of neural networks so that fluency with `MATLAB` programming was not a factor in student practice of ill-structured problem solving skills.

D. Practice Problem

This is an ill-structured problem given in stage III-D and is related to, but simpler than, the moderately ill-structured problem presented in stage III-A of the intervention and its purpose is to allow students to apply the prompts and ill-structured problem solving lessons learned from group discussions and practice their ill-structured problem solving skills. The goal is that the problem should be tractable by an individual so that the practice is not daunting. An example for the domain of neural network design is given in Figure 1b.

V. Methodology

A. Research Questions and Design

Our broad research goal was to get an in-depth understanding of the effect of DG on the activity of ill-structured problem solving. To do this, we took a two-pronged approach and analysed both students behaviours while solving ill-structured problems and the development of the ill-structured problem solving skills. The specific research questions were:

1) Does DG lead to the development of ill-structured problem solving skill and sub-skills?
2) What problem solving behaviours and heuristics do students use while learning with DG?

To achieve our goal, we performed a pilot study and collected both qualitative and quantitative data. RQ1 was answered by doing a post test only quasi-experimental quantitative study with two matched groups and random assignment to experimental and control groups. Our interventions were direct instruction (DI) for the control group and DG for the experimental group. RQ2 was answered with a qualitative study in which we did semi-structured interviews with a subset of students from both the experimental and control groups. We then compare and relate the results from the quantitative and qualitative studies in order to strengthen and refine the strategy.

B. Sample

Our sample was based on convenience and for the quantitative study, our sample size was 25 out of the 36 students (16 experimental, 9 control) enrolled in the course Neuromorphic Engineering. While we began the study with all 36 students, some students did not participate in all tutorials and hence were excluded from the study. For the qualitative study, our sample consisted of 10 students (6 experimental, 4 control) out of the 25 students who participated in the entirety of the quantitative study. Both the samples were a mix of undergraduate and graduate students.

C. Data Sources and Instruments

A survey was conducted at the start of the course which tested for prior knowledge in probability, motivation to take the course and general mathematical problem solving skill, and was content validated by three experts. To evaluate student learning after the intervention, a post-test was conducted consisting of an open design problem. The duration of the post-test was 2.5 hours at the end of which students answer sheets and any program code they had written was collected. While we conducted a post-test after each of the experiments, we are reporting results from the third experiment here as the first two are considered training for students and instructor in the new teaching-learning strategy.

The students were interviewed with a semi-structured protocol to understand their problem solving method. Students' were shown the post test question and their answer scripts and asked to respond to the four interview questions below. All interviews were audio recorded.

1) What do you do before you begin solving a problem such as this one?
2) What do you do as you solve the problem?

TABLE I: Instructor prompts for ill-structured problem solving given during stage III-C of the DG intervention

Ill-structured problem solving sub-skill	Prompts given for this sub-skill	Nature of prompt
Construction of problem space (PS)	1) For a given set of parameters that determine the dynamics of neurons, there is a unique relationship between input current and output spike rate and spike arrival time. 2) Networks that we have worked with are slow. So if you want a faster response, you need to adapt/ modify the networks to fit this new time scale.	Domain prompt Process prompt - cognitive
Generating and defending solutions	Network design involves adjusting signal propagation paths to obtain desired objectives. This can be done by: a) adding neurons in the information pathway b) appropriately weighting the signal strengths in the pathways c) modulating the time constants of neurons and/or synapses	Process prompt - cognitive
Evaluating and revising solutions	Think about things like: 1) What happens when you change one connection to the other? 2) How does changing the time constants of the neurons or synapses affect the output of a neural network? 3) What is the role of excitatory vs. inhibitory synapses? 4) What happens if I change the threshold voltage of the neuron?	Process prompt - metacognitive

3) What do you do once you have worked out a solution?
4) What do you believe is the most important step in solving a problem such as this one?

D. Procedure

The students were divided into experimental and control groups on the basis of their scores on the relevant parts of the survey such that the two groups were matched for prior knowledge (t-test, $p = 0.572$), motivation (Mann-Whitney, $p = 0.392$), general problem solving skill (Mann-Whitney, $p = 0.192$), distribution of undergraduates and graduates (Mann-Whitney, $p = 0.522$) and CPI (t-test, $p = 0.253$) and remained unchanged throughout the semester. Both groups were taught by the same instructor using different interventions and at different times.

Three experiments were conducted across the semester and different types of ill-structured problems like hypothesis testing, design, etc were used. The DI intervention had the same stages as the DG intervention except their order was changed as shown below.

1) Instructor prompts, 15-20 minutes
2) Practice problem, 20 minutes
3) Group ill-structured problem solving, 30-40 minutes
4) Class discussion of student solutions, 15-20 minutes

Thus, the control group got training in ill-structured problem solving going from simple to complex ill-structured problems, using instructor prompts without any initial delay. Apart from this difference in order, the exact same set of learning materials was used for both groups at each stage.

The interventions were conducted in the tutorial sessions of the course and lasted one-and-a-half hours for each group. The students were divided into groups of three for the intervention, which were mostly unchanged across the semester unless students were not present for the tutorial, in which case new groups were created on the spot. However, care was taken to ensure that the group structure was minimally disturbed.

After every intervention, a post-test was conducted four days later which was individual and lasted 2.5 hours, but students were allowed to use their notes from class and computers for programming.

E. Analysis

To answer RQ1, we performed content analysis [24] of the transcripts of students' post-test answer scripts, and created program code to identify the ill-structured problem solving skills used by them. The expert solution (in this case the instructors' solution) and literature about ill-structured problem solving skills served as the starting point for the coding. The unit of analysis was one idea in the solution, which could be a piece of text describing one argument, a fact or a concept; an equation, a figure or some other representation. For example, a diagram of a network was coded as "Construction of representation", as was a matrix; while an argument like "this can be ensured by adjusting the spiking current threshold for 3 and using different synaptic time const for A , C" was coded as "Constructing argument". However some codes were emergent (eg, Reflection) and we had a total of 15 codes grouped into four skill categories shown in Table II.

Each code represents the application of an ill-structured problem solving sub-skill such as "Construction of problem space" by the student at that point in the problem solving process. The sum of frequencies of all codes in a category is a measure of that sub-skill. For example, the sum of the frequencies of the codes "Select and explain solutions", "Constructing argument" and "Providing evidence" is a measure of the sub-skill "Generating and defending solutions". The overall ill-structured problem solving skill is measured by the sum of the frequencies in all four categories. The frequencies of codes related to overall skill and the sub-skills construction of problem space, generating and defending solutions, evaluating and revising solution and reflection in the two groups were compared using the Mann-Whitney test and the effect size of the intervention on the frequencies of codes was calculated using Pearson's correlation coefficient *r*.

To answer RQ2, we transcribed and analysed student interviews using content analysis [24]. The responses were initially coded along the categories of the four questions asked to the students regarding construction of problem space, generating and defending solutions, evaluating and revising solutions and the most important step in problem solving. However one category emerged from student responses, namely, reflection on the problem solving process. Within each category we initially coded students' description of their problem solving method using emergent codes. The codes describing non-unique/overlapping behaviours and heuristics in the problem solving method were then merged. The final list of observed behaviours and heuristics is given in Table V along with the results of the analysis.

TABLE II: Codes used for content analysis of student solutions to answer RQ1,"Does DG intervention lead to the development of ill-structured problem solving skills?"

Category	Code	Definition (Adapted from literature)
Construction of problem space (PS)	Construction of representation	Construct representations for given problem/solution
	Use of representation	Uses representations to solve problems
	Make assumptions	Make suitable and valid assumptions while solving problem
	Justify assumptions	Justify assumptions made in the course of solving the problem.
	Define problem	Problem clearly and completely stated.
	Generate sub-goals	Specific goals for problem solution is clearly stated.
	Identify information	Known factors and constraints are identified.
	Seek needed information	Pieces of needed information discussed.
Generating and defending solutions (GD)	Select and explain solutions	A solution is selected or developed, with explicit explanation on how the solution works. The explanation should include the interrelationship between different components, the inputs, the neurons in the network, the synapses, etc.
	Constructing argument	Coherent and persuasive reasons are provided to support the proposed solution, and factors or constraints are discussed. Explain why particular inputs were chosen, particular time constants, etc
	Providing evidence	Evidence to support the argument is strong and relevant. The evidence has been tested, or based on previous experience or real examples.
Evaluating and revising solutions (ER)	Evaluate solutions	The proposed solution is evaluated, and constraints are discussed, supported with reasoning. A statement is made about the effectiveness or benefits of the solution. The pros and cons of the solution are discussed, supported with relevant evidence (e.g. from the program), as well as how the constraints can be overcome.
	Revise Solution	The proposed solution is revised based on information obtained from the evaluation performed. The limitations highlighted by the evaluation are overcome by modifying the parameters of the solution appropriately.
	Assess alternative solutions	Alternative solution is stated, and the viability of the solution(s) is discussed. At least one optional solution is discussed. Reasons are given on why an option is selected over the other(s), with constraints discussed.
Ref	Reflection	Any comment about the problem or problem solving process

VI. RESULTS

A. RQ1: Does DG intervention lead to the development of ill-structured problem solving skill and sub-skills?

The differential and inferential statistics of the frequency of codes related to overall skill is shown in Table III. The results indicate that the experimental group (DG) had a significantly higher frequency of codes related to overall skill than the control group (DI). Further, the intervention had a medium-high effect on the frequency of codes.

TABLE III: Overall ill-structured problem solving skill: Differential and Inferential Statistics

Expt Mean (SD)	Ctrl Mean(SD)	p (Mann-Whitney)	Significant at 0.05?	Pearson's r	Effect size
34 (13.6)	22 (9.2)	0.025	Yes	0.44	Medium-high

The differential and inferential statistics of the frequency of codes related to each of the sub-skills are shown Table IV. These results indicate that in the sub-skill of "Generating and defending solutions" the experimental group (DG) had significantly higher frequency of codes than the control group (DI). Further, the intervention had a medium-high effect on the frequency of codes related to this sub-skill. On the sub-skill of "Evaluating and revising solutions" the difference between the frequencies of codes of the two groups was not significant (but approached significance); however the intervention had a medium effect on the frequency of codes related to this sub-skill. On the sub-skill of "Constructing the problem space", the difference between the frequency of codes of the two groups was not significant and the intervention had a low-medium effect on the frequency of codes related to this sub-skill. Finally, on the sub-skill of "Reflection" the difference between the frequency of codes of the two groups was not significant and the intervention had an opposite medium effect on the frequency of codes related to this sub-skill.

TABLE IV: Frequencies of each sub-skill-Differential and Inferential Statistics

	Expt Mean (SD)	Ctrl Mean(SD)	p (Mann-Whitney)	Significant at 0.05?	Pearson's r	Effect size
PS	8.4 (7.2)	5.8 (3)	0.33	No	0.214	Low-medium
GD	19.9 (7.3)	12.8 (6.2)	0.016	Yes	0.455	Medium-high
ER	5.3 (4.6)	2.3 (2.1)	0.07	No	0.348	Medium
Ref	0.4 (0.5)	1.1 (1.6)	0.33	No	-0.309	Medium

PS: Construction of problem space
GD: Generating and defending solutions
ER: Evaluating and revising solutions
Ref: Reflection

B. RQ2: What problem solving behaviors and heuristics do students use while learning with DG?

The results of the content analysis, namely the behaviours and heuristics observed in students' problem solving methods are shown in Table V. From the results we observe that while students from both the experimental and control groups show several common behaviours and heuristics while constructing the problem space such as "Connect to prior knowledge and experiences" and "Identify relevant information", the experimental group showed unique behaviours such as "Use intuition to identify a solution" and "Transform problem to a known problem". Similarly, while generating and defending solutions, both groups used the heuristic of "Generate sub-goals" and "Find a path to the goal", but the control group also did "Perform approximations" while the experimental group had several heuristics like "Adapt a known solution method", "Backtrack from goals to givens" and "Get on a different solution track". While evaluating and revising solutions, the control group did "Verify that assumptions made are reasonable" and the experimental group "Do sanity checks of solution" and "Quantify error in the solution". Finally we see that while both groups articulated their own problem solving process,

TABLE V: Analysis and results of RQ2, "What problem solving behaviors and heuristics do students use while learning with DG?"

	Construction of problem space	**Generating and defending solutions**	**Evaluating and revising solutions**	**Reflection**	**Most important step**
1	Connect to prior knowledge and experiences	Generate sub-goals	Compare solution with goal/hypothesis	Articulate own problem solving process	Initial idea of the solution
2	Draw diagrams; write equations, data, etc	Find a path to the goal	Verify solution by simulating	Recognize importance of diagrams, equations, models in problem solving	Understanding the problem
3	Identify relevant information	Reason from givens to goals using conceptual knowledge	Correlate solution with theory	Articulate role of trying various solution paths	Identifying goals/hypotheses
4	Make assumptions about the problem	Use prior knowledge and experiences	Compare solution with the original information		Making assumptions
5	Identify goals/ hypothesis/constraints	Perform approximations	Identify reasons for solution not working		Connecting to prior knowledge
6	Understand the problem	Adapt a known solution method	Redefine your goals/hypotheses		Evaluating the solution
7	Use intuition to identify solution	Backtrack from goals to givens	Identify alternate solution methods		Identifying relevant information
8	Transform problem to a known problem	Get on a different solution track	Recognize uncertainty in the problem solving process		
9		Back up and restart solution	Verify that assumptions made are reasonable		
10		Iterate the solution generation process	Do sanity checks of the solution		
11		Provide arguments for the solution	Compare generated data to givens		
12		Simulate the model/system	Quantify error in solution		

Behaviour shown by both groups	Behaviour shown by experimental group	Behaviour shown by control group

students in the experimental group "Recognize the importance of diagrams, equations and models in problem solving" and "Articulate the role of trying various solution paths".

VII. Discussion and Conclusions

The answer to our RQ1 is that DG leads to the development of students' overall ill-structured problem solving skill with a medium-high effect size. Further, we also see significant (generating and defending solutions) and near significant (evaluating and revising solutions) improvements in some sub-skills with a medium effect size. However, we find that the DG strategy does not lead to the development of the sub-skills of constructing the problem space and reflection.

The answer to our RQ2 is that the students in experimental group show more behaviours and heuristics to construct the problem space, generate and defend solutions and evaluate and revise them. Further, students in the experimental group reflect more on the role of various factors in their problem solving process. Finally, students in experimental and control group consider different steps of the problem solving process important. These results together show that students in the experimental group show a wider range of problem solving behaviours and heuristics than those in the control group. However, there are some behaviors and heuristics which students in the experimental group do not use and which are shown to be useful for engineering ill-structured problem solving [12] such as using approximations and verifying assumptions. Students are not aware of the importance of these behaviours; hence in future implementations we need to include prompts to improve student awareness and introduce some feature in the strategy to make students to practice them.

Comparing the answers to RQ1 and RQ2, we observe that for each sub-skill, several codes observed in student solutions are related to the behaviours seen during their interviews. For example, for the sub-skill of "Construction of problem space", "Define problem" is related to "Understand the problem", "Identify information" is related to "Identify relevant information" and "Construction of representation" is related to "Draw diagrams; write equations, data, etc". However there are some behaviours which students report doing, which we do not see evidence of in their written solutions. Examples of this are "Use intuition to identify solution" and "Transform problem to a known problem"; these are heuristics which are often used in ill-structured problem solving in engineering [12]. Further, even though analysis of student solutions did not show statistical evidence of the development of the sub-skills of constructing the problem space and reflection, student interviews highlight that students in the experimental group show a broader range of behaviours and heuristics in these categories which demonstrates their competence. Students are not inclined to consider the initial step of construction of problem space as part of the "solution" and hence did not write it down in the solutions which they submitted, even though they report during interview that they think about the problem, draw and write things out in "rough work" which is not submitted. Hence we could not find evidence of application of this sub-skill in their submitted solutions. Thus, this pilot study shows that for future implementations, we should refine the strategy in order to emphasize that students explicitly construct the problem space, as this is an important ill-structured problem solving sub-skill [12].

Our results are comparable with other methods for improving ill-structured problem solving skill and sub-skills, such as the development of twelve problem solving skills specific to engineering through a series of workshops over four years reported in [12]. Our method is different in that we incorporate ill-structured problem solving within the regular courses taken by engineering undergraduates and while we have not obtained development of all the skills in [12], we have obtained the results mentioned above from one course with three ill-structured problem solving sessions. We conjecture that by incorporating this intervention in all engineering courses and having students solve ill-structured problems routinely, we can expect the same level of ill-structured problem solving skill development as reported in [12].

There are several PBL/PjBL based courses [9],[10],[8], [11] that teach engineering students design and ill-structured problem solving by incorporating semester long projects that students solve in teams. These courses teach students skills such as self-directed learning, information seeking, problem definition, design process, project planning, teamwork etc which are required for real world engineering projects. In this intervention, we focus on a subset of these skills such as information seeking, problem definition and design process which are required for moderately ill-structured problems and develop them. We believe that students will be able to apply these skills as they work on real world engineering projects; however they will need explicit training in skills like teamwork and project planning, which are not the focus of this intervention.

In order to smoothly execute the intervention, the instructor (the second author of this paper) received a set of guidelines including a timeline of how the intervention would proceed, do's and dont's during the four stages of the intervention and a brief 10-15 minute training on how to handle various scenarios in the classroom. In addition, the instructor prepared the moderately ill-structured problems, practice problems and ill-structured problem solving prompts that were used in our study. However, despite the extra effort required from the instructor, the instructor felt comfortable implementing the experimental intervention and perceived qualitative differences in the problem solving process of students in the experimental and control groups. In particular, the instructor perceived greater independence and innovativeness from students in the experimental group which was attributed to the unguided exploration phase. Overall, the instructor felt enhanced engagement in the class, and found the experience highly rewarding.

As this is a pilot study and our sample size is small, and owing to the confounding variables of affective abilities, metacognition and epistemic cognition, generalizability is not our goal. Nonetheless, this pilot study offers several insights on how DG works for the teaching-learning of ill-structured problem solving skill and sub-skills and opens several avenues for further work. These include refining the strategy based on the insights gained in order to better scaffold students for the development of the skills which have scope for improvement.

References

[1] "Criteria for accrediting engineering programs, 2014 - 2015," ABET, Tech. Rep., October 26, 2013. [Online]. Available: http://www.abet.org/eac-criteria-2014-2015/

[2] D. Jonassen, J. Strobel, and C. B. Lee, "Everyday problem solving in engineering: Lessons for engineering educators," *Journal of engineering education*, vol. 95, no. 2, pp. 139–151, 2006.

[3] A. Kothiyal, "Notes on problem solving: A literature review of problem-solving, with emphasis on ill-structured, engineering and estimation problems," Indian Institute of Technology, Bombay, Technical Report (TR-ET-002-2014), 2014. [Online]. Available: http://www.et.iitb.ac.in/ aditi/lib/exe/fetch.php?media=problemsolvingtr.pdf

[4] P. C. Wankat and F. S. Oreovicz, *Teaching engineering*. McGraw-Hill New York, 1993.

[5] J. R. Savery and T. M. Duffy, "Problem based learning: An instructional model and its constructivist framework," *Educational technology*, vol. 35, no. 5, pp. 31–38, 1995.

[6] D. L. Schwartz and J. D. Bransford, "A time for telling," *Cognition and Instruction*, vol. 16, no. 4, pp. 475–5223, 1998.

[7] M. Kapur, "Productive failure," *Cognition and Instruction*, vol. 26, no. 3, pp. 379–424, 2008.

[8] M. Bozic, S. Cizmic, D. S. Pavlovic, and M. T. Escalas Tramullas, "Engineering practice: Teaching ill-structured problem solving in an internship-like course," in *Global Engineering Education Conference (EDUCON), 2014 IEEE*. IEEE, 2014, pp. 721–726.

[9] A. Shekar, "Project-based learning in engineering design education: Sharing best practices," in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2014.

[10] S. N. Kalnins, S. Valtere, J. Gusca, K. Valters, K. Kass, and D. Blumberga, "Cooperative problem-based learning approach in environmental engineering studies," *Agronomy Research*, vol. 12, no. 2, pp. 663–672, 2014.

[11] M. Williams and E. Ringbauer, "PBL field deployment: Lessons learned adding a problem-based learning unit to a traditional engineering lecture and lab course," in *ASEE Annual Conference and Exposition, Conference Proceedings*, 2014.

[12] D. R. Woods, A. N. Hrymak, R. R. Marshall, P. E. Wood, C. M. Crowe, T. W. Hoffman, J. D. Wright, P. A. Taylor, K. A. Woodhouse, and C. Bouchard, "Developing problem solving skills: The mcmaster problem solving program," *Journal of Engineering Education*, vol. 86, no. 2, pp. 75–91, 1997.

[13] S. Kimmel, F. Deek, and H. Kimmel, "Applying problem solving heuristics to a freshman engineering course," in *Proceedings of the 2001 American Society for Engineering Education Annual Conference*, 2001.

[14] E. De Graaf and A. Kolmos, "Characteristics of problem-based learning," *International Journal of Engineering Education*, vol. 19, no. 5, pp. 657–662, 2003.

[15] M. Mavinkurve and S. Murthy, "Comparing self-learning behavior of low and high scorers with ediv," in *Proceedings of the 21st International Conference on Computers in Education*. Asia-Pacific Society for Computers in Education, 2013.

[16] D. H. Jonassen, "Instructional design models for well-structured and iii-structured problem-solving learning outcomes," *Educational Technology Research and Development*, vol. 45, no. 1, pp. 65–94, 1997.

[17] J. Adams, S. Kaczmarczyk, P. Picton, and P. Demian, "Problem solving and creativity in engineering: findings of interviews with experts and novices," in *International Conference on Engineering Education ICEE 2008*, 2008, pp. 28–31.

[18] W. Zheng, M. L. Wang, and J. Yin, "Correlation analysis of scaffolding creative problem solving through question prompts with process and outcomes of project-based service learning," 2013.

[19] J. Perrenet, P. Bouhuijs, and J. Smits, "The suitability of problem-based learning for engineering education: theory and practice," *Teaching in higher education*, vol. 5, no. 3, pp. 345–358, 2000.

[20] R. Pimmel, "Cooperative learning instructional activities in a capstone design course," *Journal of Engineering Education*, vol. 90, no. 3, pp. 413–421, 2001.

[21] D. R. Woods, "Helping your students gain the most from PBL," *Problem-based learning: Educational innovation across disciplines. Singapore: Temasek Centre for Problem-based Learning*, 2000.

[22] X. Ge, "Scaffolding students' problem solving processes on an ill-structured task using question prompts and peer interactions," Ph.D. dissertation, The Pennsylvania State University, 2001.

[23] A. Anand, A. Kothiyal, B. Rajendran, and S. Murthy, "Guided problem solving and group programming: A technology-enhanced teaching-learning strategy for engineering problem solving," in *Technology for Education (T4E), 2014 IEEE Sixth International Conference on*. IEEE, 2014.

[24] K. Krippendorff, *Content analysis: An introduction to its methodology*. Sage, 2012.