In this technical report, we review problem solving literature with emphasis on engineering estimation problems. Specifically we discuss problem solving strategies, cognitive and affective abilities that affect problem solving behavior, expert problem solving behavior and strategies for teaching this expert behavior.
Introduction ........................................................................................................................................... 4
  1.1 What is estimation ability? ............................................................................................................. 6
  1.2 Who should read this report? ...................................................................................................... 7
2 The continuum of problems and the place of estimation problems in it .............................. 7
  2.1 Features of well and ill structured problems .............................................................................. 8
  2.2 Classification schemes .................................................................................................................. 9
3 Problem Solving Ability ............................................................................................................... 11
  3.1 Problem solving strategies ........................................................................................................... 11
  3.1.1 General problem solving strategies .......................................................................................... 12
  3.1.2 Ill-Structured problem solving strategies .................................................................................. 15
  3.1.3 Solving Strategies for Engineering problems: ........................................................................ 20
  3.1.4 Solving Strategies for estimation problems ............................................................................. 24
  3.2 Cognitive and affective abilities .................................................................................................. 26
  3.2.1 Cognitive and affective abilities for general problem solving .................................................. 26
    3.2.1.1 Familiarity: .......................................................................................................................... 26
    3.2.1.2 Domain and Structural Knowledge: ..................................................................................... 26
    3.2.1.3 Cognitive Controls: ............................................................................................................. 27
    3.2.1.4 Metacognition ................................................................................................................... 27
    3.2.1.5 Epistemological Beliefs: ..................................................................................................... 27
    3.2.1.6 Affective and conative abilities:........................................................................................ 28
    3.2.1.7 Practices: ............................................................................................................................ 28
  3.2.2 Cognitive and affective abilities for ill-structured problem-solving ........................................ 28
  3.2.3 Cognitive and affective abilities for engineering problem solving ........................................ 30
  3.2.4 Cognitive and affective abilities needed for estimation problem solving ......................... 32
4 Expert behavior in problem solving ............................................................................................... 33
  4.1 Expert-novice differences in general problem solving............................................................... 33
  4.2 Expert-Novice differences in ill-structured problem solving ...................................................... 37
  4.3 Expert novice differences in engineering problem-solving ......................................................... 38
5 How do people fare on estimation tasks? ....................................................................................... 40
6 Teaching - Learning strategies for problem solving ................................................................. 42
  5.1 General problem solving teaching-learning strategies ................................................................. 42
  5.2 Ill-structured problem-solving teaching-learning strategies ....................................................... 45
  5.3 Teaching-learning strategies for engineering problem solving: ............................................. 47
5.4 How to teach estimation? .................................................................49
6 Conclusions ......................................................................................52
7 References........................................................................................53
8 Appendix A – Mahajan’s tools for dealing with complexity ..............57
9 Appendix B – Linder’s Effective actions ..........................................59
10 Appendix C – Comparison of measurement estimation strategies ....61
11 Appendix D – Quality criteria for engineering problem solving (adapted from [46]) 62
12 Appendix E – Comparison of instructional studies in measurement estimation (adapted from [9]) .................................................................65
13 Appendix F – Instructional Studies in general problem solving ..........67
14 Appendix G – Instructional Studies in Ill structured and engineering problem solving .................................................................71
15 Appendix H – Comparison of features of teaching-learning strategies using problem-solving as an activity (adapted from [61]) ..........................74
1 Introduction

As engineering educators we are often faced with the task of explaining to our students what it means to be an engineer and ensuring that our curriculum helps them learn to become so. As defined in [1], there are several performance outcomes required of an engineering graduate and in this report we choose to focus on one specific outcome namely 3.e,

“an ability to identify, formulate and solve engineering problems.”

This outcome has been alternately defined in the Washington accord [2] in two parts:

i) “Define, investigate and analyze complex problems” which means to “Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.”

ii) “Design or develop solutions to complex problems” which means to “Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.”

As quoted by the author in [3], Gagne believed that "the central point of education is to teach people to think, to use their rational powers, to become better problem solvers." This becomes very evident in engineering as highlighted by Jonassen et. al. [4] who interviewed several practicing engineers to understand the work they did and found that their work entailed solving workplace problems with the following characteristics (among others),

1. Workplace problems are ill-structured. Practicing engineers are commonly dealing with incomplete information.
2. Workplace problems may be combinations of structured problems.
3. Workplace problems have multiple, conflicting goals.
4. There is no one, single good way to solve the problem. Often different ways need to be considered and/or applied.
5. Most constraints are non-engineering.
6. Workplace problems are often solved through collaboration.

It is evident from this list that a major part of an engineers’ job is to solve complex and ill-structured problems in the workplace. Engineering Estimation is also complex and ill-structured problem because firstly, the goals are conflicting in that we are not clear how accurate an estimate is required. Secondly, we are not sure where to start and what method to use to get a “good” estimate. Thirdly, we have to apply multiple concepts from physics, mathematics and other sciences in order to analyze the problem. Finally, we have to make several assumptions and simplifications in order to make the
analysis tractable. We claim for these reasons that estimation is a “complex” problem as defined in the Washington Accord [2] and also an ill-structured problem as defined in literature [3], [4], [11], [12], [13].

Indeed, estimation is not limited to engineering alone, but is a commonplace and important activity in everyday life. For example, we may need to estimate how big a space in our home is to decide if a table we intend to buy will fit or not. Or estimate how much cash you will need to buy vegetables for a week and whether you will have enough time to grab coffee before class. In engineering, consider for example, that you have an idea for a new smart phone. You will have to estimate what the life of your battery will be if all the different capabilities you plan to integrate into it are run simultaneously. This estimate will decide if such a smart phone is a feasible idea at all.

In addition to the above mentioned situations, estimation is used in engineering for a variety of reasons listed below [5], [6], [8], and [9]:

1. To set up, find parameters for and evaluate detailed analyses. This includes the following scenarios:
   a. Checking on the reasonableness of results – “sanity check”.
   b. When the choice of calculation of exact answer depends on the expected size of the answer, e.g., can we assume linearity or not.
   c. Need upper and lower bounds.
   d. Need to decide how to better our assumptions.
2. When the exact calculation is too difficult e.g. be because of non-linearity or because an equation cannot be solved exactly.
3. To establish feasibility of a design.
4. To evaluate the impact of a change in the input at the output.
5. When tools like a calculator or computer are not available.
6. When exact values of information or data are not available.
7. When the actual equations governing a situation are not available.
8. When complete accuracy is not necessary.
9. To eliminate candidate design solutions.
10. To sketch out potential paths to a solution.
11. To plan projects or experiments.
12. In the selection of materials or components.

From this discussion we conclude that engineering estimation is an important problem that engineers must learn to solve.

As mentioned in Section 1, there are no experimental studies on the effectiveness of instruction in engineering estimation. However [7] provides an excellent review of research studies on people’s measurement estimation abilities. Some studies have found that only 20-30% children and ~50% adults gave “correct” responses and the percentage errors ranged from 10% - 100%. However these studies also highlight that
estimation ability improves with age and instruction. All of these results indicate that estimation ability can be improved with practice and experience. So it is important to design teaching strategies to improve students’ engineering estimation abilities.

1.1 What is estimation ability?
In [5], the authors defined a framework to explain most estimation activity. It consists of a set of effective actions and associated information that people use while solving estimation problems. The manner in which these effective actions are employed by an individual depends on a set of mediating characteristics like individual knowledge, mental abilities and beliefs, context, etc. Thus the particular solution that an individual comes up with depends on how the mediating characteristics influence their usage of effective actions. The effect of the mediating characteristics can be offset by compensation methods like guessing and brainstorming. Together the effective actions and compensation methods account for most of what people do while making estimates. In other words,

\[
\text{Individual estimation ability} = f(\text{effective actions, compensation methods, mediating characteristics})
\]

Thus this framework is useful for understanding how people perform estimation, why they may be unsuccessful in making good estimates and how we can guide/teach the estimation activity.

In a similar manner, Jonassen [3] and Schoenfeld [10], among others [11], [12], [15], [16], describe problem solving ability as being a function of the individual’s knowledge base (domain, structural, conceptual and procedural knowledge), problem solving strategies, metacognition, epistemological beliefs, affective and conative abilities and practices of the domain (we will elaborate on these in later sections). Thus we have,

\[
\text{Individual problem solving ability} = f(\text{knowledge base, problem solving strategies, metacognition, epistemological beliefs, affective and conative abilities, practices of the domain})
\]

As in the case of any field, there exist expert problem solvers who exhibit a problem solving ability that we would like our students to emulate. In other words, the goal of instruction is to translate individual problem solving ability to the level of expert ability. Thus we have,

\[
\text{Individual problem solving ability + Instruction} \rightarrow \text{Expert problem solving ability}
\]

In the case of general, ill-structured and engineering problem solving the expert behavior is quite well understood, as we elaborate later. So it is possible to tailor instruction to fill in the gaps in individual problem solving ability and take it to the expert level. However in the case of estimation problems, while we know that there
exists a difference between expert and novice behavior [5], the expert or target behavior is not well-understood. Thus the problem of instructional design for estimation problem solving is a significant and hard one.

In the remainder of this report we elaborate on individual problem solving ability with emphasis on engineering estimation (Section 4), expert problem solving behavior (Section 5) and teaching strategies for problem solving (Section 6). Before we delve into these, we present an overview of the types of problems and their differences (Section 3).

1.2 Who should read this report?

- If you are an ET PhD student looking to get an understanding of problem solving, read the entire report.
- If you are an engineering instructor looking for problem solving teaching strategies, read section 6.
- If you are an ET researcher trying to get an understanding of problem solving ability or expert problem solving behavior, read sections 4 and 5.
- If you are either of 1, 2 or 3, you should of course begin by reading section 3.

2 The continuum of problems and the place of estimation problems in it

There is general agreement among researchers in problem solving that problems can broadly be classified into two types, namely well-structured and ill-structured problems [3], [10 -17] even though they are sometimes referred to by different names. In his 1992 chapter on mathematical problem solving, Schoenfeld [10] provides two dictionary definitions for the word problem which neatly encapsulate the differences between the two types of problems, without using the terms well-structured and ill-structured.

Definition 1: “In mathematics, anything required to be done, or requiring the doing of something.”

Definition 2: "A question... that is perplexing or difficult."

A well-structured problem is thus a problem according to definition 1. An ill-structured problem is a one that fits definition 2. In [15], Woods et. al. elucidates this difference for engineering problems. They contrast “problem solving” with “exercise solving”. They define problem solving to be the “processes used to obtain a best answer to an unknown, or a decision subject to some constraints. The problem situation is one that the problem solver has never encountered before; it is novel. An algorithm or procedure to be used to solve the problem is unclear; problem solving requires much mental work.” On the other hand the latter refers to the “recalling of familiar solutions from previously-solved problems.” Thus according to these definitions, problems are ill-structured and exercises are well-structured.
2.1 Features of well and ill structured problems

A problem has three components: an initial state, a goal state and a method or procedure to eliminate the gap between these two states. For a well-structured problem, these three components are either clearly identified in the problem description or familiar based on the information given in the problem statement [3], [11], [13], and [16] and [23]. For example, back of the textbook problems in mathematics and other STEM disciplines come under this category.

Ill-structured problems are those for which some or all of the three components of a problem are not clearly defined in or evident from the problem description [3], [11-17], [23]. Real-world problems like engineering problems, life decisions, designing a house, creating art etc fall in this category. The description of these problems is unclear and/or incomplete and the information needed to solve them may not be available in the problem statement. In fact it may not even be clear from the problem statement if there is a problem at all. It is after the solver creates his own representation of the problem, that the goal state becomes evident and probable paths to the solution emerge. Further, the exact nature of the problem will depend on the assumptions made by the problem solver [18] during problem representation. This in turn affects the solution of the problem. Thus it is obvious from this discussion that such problems have multiple possible solutions and solution paths; there is no best solution only a best possible one.

In Table 1, we present a comparison of the features of well and ill structured problem given in [13]. As it can be seen from the table, ill-structured problems possess additional features which make them harder to solve.

Table 1 – Comparison of well and ill structured problems

<table>
<thead>
<tr>
<th>Well structured problems</th>
<th>Ill structured problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present all elements of the problem</td>
<td>Appear ill-defined because one or more of the problem elements are unknown or not known with any degree of confidence</td>
</tr>
<tr>
<td>Have vaguely defined or unclear goals and unstated constraints</td>
<td></td>
</tr>
<tr>
<td>Are presented to learners as well-defined problems with a probable solution</td>
<td>Possess multiple solutions, solution paths, or no solutions at all that is, no consensual agreement on the appropriate solution,</td>
</tr>
<tr>
<td>Possess correct, convergent answers,</td>
<td>Possess multiple criteria for evaluating solutions.</td>
</tr>
<tr>
<td>Possess less manipulable parameters,</td>
<td></td>
</tr>
<tr>
<td>Engage the application of a limited</td>
<td>Have no prototypic cases because case elements are differentially important in different contexts and because they interact</td>
</tr>
<tr>
<td>Present uncertainty about which concepts,</td>
<td></td>
</tr>
</tbody>
</table>
number of rules and principles that are organized in a predictive and prescriptive arrangement with well-defined, constrained parameters, rules, and principles are necessary for the solution or how they are organized,

Involves concepts and rules that appear regular and well-structured in a domain of knowledge that also appears well-structured and predictable,
Possess relationships between concepts, rules, and principles that are inconsistent between cases,

Have a preferred, prescribed solution process. Have no explicit means for determining appropriate action,

Possess knowable, comprehensible solutions where the relationship between decision choices and all problem states is known or probabilistic
Require problem solvers to make judgments about the problem and defend them.

<table>
<thead>
<tr>
<th>Type of Problem</th>
<th>Data</th>
<th>Methods</th>
<th>Goals</th>
<th>Jonassen’s problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Given</td>
<td>Familiar</td>
<td>Given</td>
<td>Logical, algorithmic</td>
</tr>
<tr>
<td>2</td>
<td>Given</td>
<td>Unfamiliar</td>
<td>Given</td>
<td>Story</td>
</tr>
<tr>
<td>3</td>
<td>Incomplete</td>
<td>Familiar</td>
<td>Given</td>
<td>Mathematical proofs.</td>
</tr>
<tr>
<td>4</td>
<td>Incomplete</td>
<td>Unfamiliar</td>
<td>Given</td>
<td>Strategic performance, Design, Estimation/Approximation</td>
</tr>
<tr>
<td>5</td>
<td>Given</td>
<td>Familiar</td>
<td>Open</td>
<td>Rule-using, exploratory studies</td>
</tr>
<tr>
<td>6</td>
<td>Given</td>
<td>Unfamiliar</td>
<td>Open</td>
<td>Decision-making*, troubleshooting*, diagnosis-solution*, case analysis, dilemmas*, MPPs,</td>
</tr>
</tbody>
</table>

2.2 Classification schemes

Researchers have gone beyond listing features of these two types of problems, and come up with detailed classification schemes. Next we describe two such classification schemes and relate the two.

Johnstone (adapted here from [11]) developed a scheme based on the status of the three components of a problem namely the data or givens, the means or solution methods and the goals as they are described in the problem statement. It is explained in Table 2 below.

Table 2 (adapted from [3])
Estimation/Approximation

| 7  | Incomplete | Familiar | Open | How to fit a method to a problem? |
| 8  | Incomplete | Unfamiliar | Open | Research |

*: We assume that when we say a problem goal is given it is a single fixed goal, then this classification is valid.

The entries in red are our own examples and not from [3].

In this classification it is clear that the well-structured textbook problems that students typically encounter in engineering education are of types 1 to 3, while real world ill-structured problems are of types 4 to 8.

--

In [3] the author collected and performed a cognitive task analysis of hundreds of problems. He then identified the attributes of these problems and classified them into eleven types based on these characteristics. His classification is such that from left to right the problems represent a continuum from well to ill structured (Figure 1). To some extent this classification represents a hierarchical structure of problems and the skills needed to solve the well structured problems on the left serve as pre-requisite to solving the ill structured ones on the right. The eleven types in order are Logical, algorithmic, story, rule-using, decision-making, troubleshooting, diagnosis-solution, strategic performance, case analysis, design, dilemmas. Case-analysis problems, for instance, require that problem solvers to be able to solve decision making and aspects of troubleshooting in order to be able to solve case problems. Similarly, decision making requires rule using and story problems as prerequisite and so on [3].

We have attempted to classify these eleven types according to Johnstone’s scheme and our classification is given in the rightmost column of the table 1. We have also situated some other types of problems known to us which were not categorized by [3] into the table and these are indicated in red.

To situate estimation problems in these classification schemes, consider the following two definitions of estimation

- An analysis to determine all quantities to some level of specificity [5].
- Making decisions or selecting from a multitude of options based on incomplete or unavailable details or data [6].

As per the first definition the goal of the problem – determining a quantity to some level of specificity – is given while the data and methods are not clear. Thus it is a problem of type 4. However according to the second definition, estimation is like a decision-making task and hence falls under type 6. For the purposes of this report, since
we are focusing on finding an approximate value for a quantity, estimation is a problem of type 4, an ill-structured problem.

Now let’s try to classify estimation according to Jonassen’s scheme [3]. As we will see later, estimation involves many of the same cognitive tasks that design does such as modeling a system, making assumptions, decision making on which model to choose and mathematical calculations. Thus while it is not possible to place it exactly without a rigorous cognitive task analysis of the kind Jonassen performed in [3], we conjecture that estimation problems would lie near design problems, perhaps just to the left of design (see figure 1). So it is an ill-structured problem according to this definition as well.

Thus we have established that estimation problems are ill-structured in nature and will concentrate only on these types of problems for the rest of the report.

Takeaway: There are two types of problems, well and ill structured and two classification schemes to describe the continuum of problems. Estimation is an ill structured problem according to either classification scheme.

3 Problem Solving Ability

As explained in Section 2.1, problem solving ability is a complex function of several variables. Researchers [12] agree that the solvers’ experience with and knowledge of the situation together determine if and how a solution will be reached to a problem. Thus there are a set of internal and external factors that affect problem solving [3], [23]. The internal factors are those that describe the problem solver and external factors are those that describe the problem. External factors include things like problem structure, complexity, domain specificity and problem context [3], [23]. By individual factors we mean the individual differences between problem solvers [3], [23] which exist along dimensions like domain, structural and procedural knowledge, attitude, motivation, metacognition, epistemic beliefs, etc. For the purposes of this report we consider only the internal factors affecting problem solving behavior as these are the factors we would like to change via instruction. In Section 4.1 we expand on the problem solving strategies available for solving different kinds of problems while in Section 4.2 we focus on the individual cognitive and affective abilities that affect problem solving ability.

3.1 Problem solving strategies

Research has established that well-structured problem-solving strategies and abilities are necessary (though not sufficient) for the solving of ill-structured problem solving [41]. Further in engineering, it is first necessary for the students to understand and apply scientific principles in well-structured problems, before proceeding to the ill-structured real world problems. Therefore we first review general problem solving research (beginning with well-structured problems) as it has evolved over the past 50 years. This will lay the foundations for the description of ill-structured problem-solving strategies.
3.1.1 General problem solving strategies

The strategy used to solve a problem will depend on its structure. If the initial state is given and the goal is clear, the problem reduces to finding the best way to go from the initial state to the goal state. This process of searching for solution paths has been extensively studied [11-13], [16], [19-23]. If however the initial state and/or goal state are not explicitly specified as is often the case in ill-structured problems, then the solver has to follow a different approach, namely one in which the initial state, goals and constraints of the problem are first identified. The solution strategy will then depend on these identified problem components [10-14], [16, 17]. Various classes of problems have their own task characteristics and these greatly determine the behavior of the problem solver and the strategies for finding solutions to these problems [12]. Further as Jonassen says, while it is assumed that the learning of well-structured problem solving transfers to ill-structured problems as well, research has demonstrated that performance on well-structured problems is independent of performance on ill-structured problems as the latter engage a different set of epistemic beliefs [23 and references therein].

Nevertheless, in this subsection we review some general problem solving strategies as these serve as a starting point for ill structured problem solving strategies. In the next sub-section on ill-structured problems, we will identify the modifications necessary to the well-structured problem solving strategies to solve ill-structured problems.

Table 3 – Comparison of well-known problem solving strategies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem representation</td>
<td>Understand the problem</td>
<td>Planning and abstraction processes</td>
<td>Identify</td>
<td>Recognize or identify the problem. Define and represent the problem mentally.</td>
<td></td>
</tr>
<tr>
<td>Search for solutions</td>
<td>Devise a plan</td>
<td>Search processes</td>
<td>Explore</td>
<td>Develop a solution strategy. Organize his or her knowledge about the problem.</td>
<td></td>
</tr>
<tr>
<td>Implementing solutions</td>
<td>Carry out the plan</td>
<td>Act</td>
<td>Allocate mental and physical resources for solving the problem. Monitor his or her progress toward the goal. Evaluate the solution</td>
<td>Look Backward</td>
<td>Look backward</td>
</tr>
</tbody>
</table>
In his classic and widely read book [24], Polya describes the following four steps in the problem-solving process:

1. Understand the problem: This step is also known as problem representation. Polya encourages solvers to draw diagrams and graphs if necessary.
2. Devise a plan: In this step the solvers find a connection between the data and the unknown by trying to find similar problems that they have encountered before or finding auxiliary problems if a direct connection cannot be found. Ultimately the goal of this step is to come up with a plan to the solution.
3. Carry out the plan: In this step solvers implement their solution also check that each step is correct.
4. Look backward: Finally, solvers examine their solution for correctness and understanding. This means that solvers try to see what they have learned from this problem, whether the results and methods can be carried forward to another problem and what if anything they would do differently next time.

In [19], [20] and references therein, Simon et. al. describe the General Problem Solver (GPS), a computer program that is capable of solving problems the way humans do. It is based on the information processing theories of learning and describes two processes underlying human problem solving. The first are the planning and abstraction processes in which the solver generates simpler problem spaces by abstracting from the concrete details of the original problem and creates a plan for a problem solution in this simpler space. The second processes are the search processes in which the solver uses heuristics to find and reduce the difference between the initial and goal states.

Based on the GPS model, Bransford & Stein (1984) proposed another problem solving called the IDEAL (identify, develop, explore, act, look back) problem solver. The IDEAL process for solving problems includes identifying potential problems, defining and representing the problem, exploring possible strategies, acting on those strategies, and looking back and evaluating the effects of those activities [13].

In [16] the authors explain the problem solving cycle described by psychologists like Bransford, Stein, Hayes and Sternberg. The cycle consists of stages in which the problem solver does the following:

1. Recognize or identify the problem.
2. Define and represent the problem mentally.
3. Develop a solution strategy.
4. Organize his or her knowledge about the problem.
5. Allocate mental and physical resources for solving the problem.
6. Monitor his or her progress toward the goal.
7. Evaluate the solution for accuracy.

This cycle does not imply that every solver proceeds through these stages sequentially while solving problems. The stages are called a cycle because they are a
part of an iterative process in which once these stages are complete, the solver often needs to repeat the cycle before a solution is found. Thus successful problem solvers need to be flexible in apply strategies. This process is exemplified in the following example from [11] in which the author describes a study using the Chinese ring puzzle wherein all the subjects who successfully solved the puzzle went through a process of exploration. They initially randomly made progress towards the solution, then backtracked, then moved forward and backtracked again. Subjects repeated this process until they understood the structure of the puzzle and then directly proceeded towards the solution. What these successful solvers were essentially doing was exploring the problem using various heuristics such as trial and error and means-ends analysis until they understood the puzzle and an algorithm suitable to solving it. Once that was done, they applied the algorithm to solve the problem.

All of these strategies are neatly synthesized into the following three step problem solving strategy by [12], [13].

Figure 2 – General problem solving strategy

```
| Problem representation | Search for solutions | Implement solutions |
```

1. **Problem representation:** This step consists of the solver understanding the problem to identify its attributes, namely the givens, goals and constraints. In so doing the solver interprets the problem statement and creates his representation of the problem. A good representation can make the problem easier to solve while an improper one in which the solver adds unnecessary constraints or incorrect goals can make the problem difficult or even impossible to solve. This step also helps the solver link the problem to his prior knowledge and activate problem related schema. Again a good representation can aid this process and guide the search for solutions which is the next step.

2. **Search for solutions:** This can be done in several ways listed below.
   a. Recalling analogical problems: This method involves recalling a previously solved problem and applying that solution method to the current problem [24], [25]. Mahajan [9] mentions this strategy (calling it abstraction) as one of the techniques that students should have in their repertoire to perform estimation effectively. Hence it is worthwhile for us to consider a bank of generally applicable analogical problems in engineering domains that can guide our students in performing estimation.
   b. Means-ends analysis: As described in [19] this is a heuristic in which the problem solver after identifying the goals to be achieved, selects methods to achieve those goals. In other words, this is a heuristic for reducing the discrepancy between the initial and goal state by applying problem solving
methods or finding the best means to achieve the desired ends. This method however has some disadvantages. Firstly, it is not guaranteed to find a solution. Secondly, as demonstrated in [26], the cognitive load imposed by the process of working towards the goals impedes the process of schema acquisition, i.e., focusing on the attributes of the problem in order to understand the kind of problem that exists.

c. Decomposing and simplifying – Finding sub-goals: In this strategy, the solver breaks the problem down into smaller problems which he/she will find easier to solve compared to the original problem. While this strategy can reduce the solution search space if the solver has knowledge of a sub-goal that can be achieved in fewer steps, there is little advice on how to apply this heuristic to actual problems, i.e., how to choose sub-goals. This method requires that the solver have complete knowledge of the techniques and problem solving domain [13 and references therein]. Theoretically this is an ideal strategy but research has shown that creating sub-goals doesn’t always help [12 and references therein] and providing a sub-goal to solvers can often increase their confusion as they don’t know how to proceed once the sub-goal has been achieved. Mahajan [9] extensively uses this strategy in the form of divide-and-conquer to solve estimation problems. Hence it is important for us to think of ways to teach students to perform this sub-goaling in an efficient way.

d. Generate and Test: This is the least efficient method of searching for solutions and most used by novice solvers [12], [13]. Hence it is not recommended for instruction.

3. Implementing solutions: This final step involves solvers trying out the generated solutions and evaluating to see if it works. If it does then the problem has been solved. If not a new hypothesis needs to be generated or the old one needs to be modified. This is the step at which solvers who are unaccustomed to failure may need extensive coaching and prompting regarding aspects of the existing solution that may be used to generate a new solution [13].

Takeaway: General problem solving strategies have three major steps – problem representation, search for solutions and implementation of solutions. The search for solutions is performed using heuristics like analogies and decomposition which are applicable even in the case of estimation problems.

3.1.2 Ill-Structured problem solving strategies

According to situated cognition, (ill-structured) problems arise all the time, no matter what you are doing and context and experience affects problem solving by influencing the manner in which the problem solver [65]

1. Performs framing and registration (or problem space construction),
2. Interacts with the environment, uses external representations to support problem solving, adds structure to the environment and performs epistemic actions
3. Uses the resources in the environment and generates scaffolds/affordances in the environment to support problem solving
4. Uses the knowledge about the context, the environment and has a host of methods, heuristics, etc to solve problems

In Voss & Post (1988) and Sinnott (1989) the authors examined the think-aloud protocols of solvers working on ill-structured problems (social science problems in the former case and everyday problems in the latter) [12], [13], [41-43] and described a sequence of thinking processes that solvers indulged in, namely, representation, solution and monitoring & evaluation processes. We compare the two sets of processes in Table 4 below.

Table 4 – Comparison of ill-structured problem solving steps

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>recognizing that there is a problem</td>
<td>processes to construct problem space</td>
</tr>
<tr>
<td>finding out exactly what the problem is</td>
<td></td>
</tr>
<tr>
<td>searching and selecting some information about it</td>
<td></td>
</tr>
<tr>
<td>developing justification by identifying alternative perspectives</td>
<td></td>
</tr>
<tr>
<td>organizing obtained information to fit a new problem situation</td>
<td></td>
</tr>
<tr>
<td>generating some possible solutions</td>
<td>processes to choose and generate solutions</td>
</tr>
<tr>
<td>deciding on the best solution by the solver’s perception of problem constraints</td>
<td></td>
</tr>
<tr>
<td>implementing the solution and evaluating it by developing arguments and articulating personal belief or value</td>
<td></td>
</tr>
</tbody>
</table>

These results are synthesized into a seven step problem-solving strategy for ill-structured problems in [13] and explained below. However situated cognition argues that the framing, registration (problem space construction) and search processes are intertwined as the problem may be reformulated as the solver interacts with the environment during problem solving [65].
1. **Learners Articulate Problem Space and Contextual Constraints:** In this first step, the solver needs to create a problem representation. To do so, he needs to understand the problem statement and the context in which it is situated [65]. In ill-structured problems the context is very important because the problem emerges from the situation [13 and references therein]. According to situated cognition, a problem is tied to a concrete setting and solved using locally available materials, social and cultural resources and scaffolds [65]. The goal of representation in this case, rather than identifying or classifying problem type as in the case of well-structured problems, is to gather as much information about the problem and context as possible [12],[13],[41-43],[65]. Then this information has to be compiled to create a problem space that contains all possible problem states, operators and constraints [12],[13],[41-43],[65] by examining all possible causes of the problem. At this step solvers have to reflect on what they know about the problem domain – have they read, heard or seen anything about it, what they believe about it and what their own biases are [13]. They also have to reflect on how to collect information about the domain, where to look and whom to ask. As in the case of well-structured problems, better prior knowledge – which in this case is the domain knowledge – will greatly improve problem solving ability [13].

2. **Identify and Clarify Alternative Opinions, Positions, and Perspectives of Stakeholders:** In order to fully appreciate the complexity of the problem, the solver must first recognize and identify the multiple, often divergent views and opinions that exist on the issues involved in the problem [13]. Part of the problem representation process for ill-structured problems, thus, includes identifying the various stakeholders in the situation, their respective goals and their perspectives on the problem.

3. **Generate possible problem solutions:** As mentioned several times already, ill-structured problems will have multiple solutions. These different solutions are generated by the solver considering the goals of the various stakeholders in the problem and deciding how each one would achieve their goals. Another way is to identify the different causes of the problem and then generate solutions to alleviate the causes. Either way it is important to note that the solutions generated will be a
function of the problem representations constructed by the solver [12],[13], [41-43],[65]. Thus after considering different problem representations and the alternative solutions they give rise to, the solver has to evaluate them and either select one or synthesize a solution from his own mental model of the problem [12],[13],[41-43].

4. **Assess the Viability of Alternative Solutions by Constructing Arguments and Articulating Personal Beliefs:** Once several solutions have been generated, solvers have to select or construct a viable solution and be able to justify how they came to that decision. The best solution will be the one for which the solver can provide the most arguments in favor of; in order to do so the solver must gather evidence in support of and against each solution [13]. By debating the solutions, either with themselves or collaborators, the solver will refine his/her problem representation and decide on a final solution.

5. **Monitor the problem space and solution options:** In ill-structured problem solving, the mental model of the problem space developed by the solver is not fixed as in the case of the schema of well-structured problems. This model evolves as the solver considers what he knows and believes, what others believe and generates arguments to support his/her initial solution [13]. Thus ill-structured problem solving engages what Kitchner called epistemic cognition whereby solvers consider how much they know, how certain they are of what they know and how they know it to be true. Quoting Kitchner (1983) [13] says, epistemic cognition "leads one to interpret the nature of the problem and to define the limits of any strategy to solving it." In other words, the solver must first decide whether the problem is solvable at all. This monitoring process, according to Sinnott (1989) relies heavily on a variety of memories of the solver especially those that are related to prior problem solving experiences and these may either support or impede the problem solving process [13].

6. **Implement and monitor the solution:** Once the solution has been implemented the solver must monitor the solution to examine how it is working. Based on this performance, solvers may need to adapt the solution and after reflecting on it, learn and extrapolate from it as this is important to transfer the solution to related problems. Thus the result of this stage is that solvers posses better integrated mental models [13].

7. **Adapt the solution:** If it is possible to try out the solution, then the problem-solving process becomes an iterative process of implementation, monitoring and adapting the solution based on feedback [13].

In [14] the authors performed a think aloud protocol analysis of professionals from four different professions (architecture, engineering, medicine and law) as they attempted to solve an ill-structured problem. Their protocols were coded into nine basic actions, plus nine meta-actions (which describe or refer to any of the basic actions) that these professionals employed to solve the problem. While these actions provide a
framework for the ill-structured problem solving process of these professionals, it is important to observe that these actions are not taken in any particular sequence.

1. **Recall**: To recall facts specifically mentioned in the problem.
2. **Read**: To read words, phrases or statements from the problem.
3. **Assume**: To take for granted or suppose some fact or perceived fact not mentioned in the problem and to use it.
4. **Know**: To be certain of some knowledge and to use it.
5. **Infer**: To conclude from evidence or premises or to adopt as a logical consequence.
6. **Evaluate**: To fix the value of; or to examine and judge, information specified in the problem.
7. **Calculate**: To calculate the numerical value of information.
8. **Query**: To utter a question, inquiry or a doubt, and
9. **Recommend**: To counsel or advise that something be done to solve the problem.

Based on extensive qualitative and quantitative analysis of the protocols, the results on the ill-structured problem solving process of these professionals can be summarized as follows:

1. With the exceptions of architects, professionals use more actions than meta-actions.
2. In general individuals belonging to the same professions have higher correlation coefficient than individuals belonging to dissimilar professions.
3. They found significant evidence for profession specific general problem solving strategies that are significantly similar for individuals in the same professions and differ across individuals in different professions.
4. Common actions for engineers are five out of eight namely, assume, infer, recommend, evaluate and query. Analysis indicates repeated use of the actions recommend, infer, assume and query in solving the problem. Further, assume, query and evaluate actions lead to recommend actions.
5. Common actions for lawyers are six out of eight namely, know, infer, query, evaluate, assume and recommend. In this case know and infer actions lead to recommend actions and it was further found that there was repeated use of assume and recommend actions in solving the problem.
6. The authors also found that general cognitive strategies (from the training in their respective domains) sometimes help and sometimes hinder professionals from solving complex and ill-structured problems.

In Table 5 we compare the solving strategies of well and ill structured problems in order that we may understand how they are different and why instruction in well structured problem solving is not sufficient to solve ill structured problems.

Table 5 – Comparison of well and ill structured problem solving strategies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem representation</td>
<td>Learners Articulate Problem Space and Contextual Constraints</td>
<td>Recall; Read; Know</td>
</tr>
<tr>
<td></td>
<td>Identify and Clarify Alternative Opinions, Positions, and</td>
<td>Know; Assume; Infer; Query</td>
</tr>
</tbody>
</table>

### Perspectives of Stakeholders

<table>
<thead>
<tr>
<th>Search for solutions</th>
<th>Generate possible problem solutions</th>
<th>Calculate; Know; Infer; Evaluate; Recommend</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Assess the Viability of Alternative Solutions by Constructing Arguments and Articulating Personal Beliefs</td>
<td>Infer; Evaluate; Recommend</td>
</tr>
<tr>
<td></td>
<td>Monitor the problem space and solution options</td>
<td>Meta-know; Meta-calculate; Meta-evaluate; Meta-recommend; Meta-infer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implement the solution</th>
<th>Implement and monitor the solution</th>
<th>Calculate; Evaluate; Meta-calculate; Meta-evaluate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Adapt the solution</td>
<td>Calculate; Evaluate</td>
</tr>
</tbody>
</table>

**Takeaway:** The ill-structured problem solving strategy comprises seven steps, not all of which may be necessary for every type of ill-structured problem. There are a few steps more than the general problem solving strategy, which explains why instruction in general problem solving does not transfer to ill-structured problems. There are certain common actions taken in each step of the strategy.

### 3.1.3 Solving Strategies for Engineering problems:

In [36] Wankat & Oreovicz propose a strategy based on the problem solving strategy used by Donald Woods and his colleagues [15] at McMaster University. Their strategy has six operational steps and a prestep which focuses on motivation.

![Figure 4 – Engineering problem solving strategy](image)

1. **I can:** Step 0 is a motivation step. Since anxiety can be a major detriment to problem solving, this step helps with the solvers’ self-confidence.

2. **Define:** Step 1, the define step, is the step in which the solver lists the knowns and unknowns, draws a figure, and perhaps an abstract figure which shows the fundamental relationships. The figures are critical since an incorrect figure can lead
to an incorrect solution. The goals and constraints of the problem are clearly identified.

3. **Explore**: Step 2 is the explore step. This step was originally missing from the strategy but was added when its importance to expert problem solvers became clear. This step can also be called “Think about it,” or “Ponder.” During this step the solver asks questions and explores all dimensions of the problem [36]. Is it a routine problem? If so, the expert will quickly solve the problem using a working forward strategy. If it is not routine, what parts are present? Which of these parts are routine? What unavailable data are likely to be required? What are the alternative solution methods and which is likely to be most convenient and accurate? Does this problem really need to be solved, or is it a smoke screen for a more important problem? Many experts determine limiting solutions to see if a more detailed solution is really needed. Note that determining limits is also an estimation technique [9] and thus estimation can be an important part of solving an engineering problem.

4. **Plan**: In this step, the solver sets up the steps of the problem. For long problems a flowchart may be useful. The appropriate equations are written and solved without numbers.

5. **Do it**: Do it, step 4, involves actually putting in values and calculating an answer. The separation of the plan and do it stages makes for better problem solvers in the long run. Separating these stages makes it easier to check the results and to generalize them since putting in new values is easier.

6. **Check**: Checking includes internal checks for errors in both mathematical manipulations and number crunching, and evaluation with external criteria to check for consistency. A very useful method employed by expert problem solvers is to compare the answer to the limits determined in the explore step. The answer should also be compared to “common sense.”

7. **Generalize**: The last step, generalize, includes thinking about what has been learned about the content. How could the problem be solved more efficiently in the future? If the problem was not solved correctly, what should have been done? Solvers should collect feedback and then resolve incorrect problems.

In Table 6 we compare the engineering problem solving processes with general ill-structured problem solving processes and observe that the engineering problem solving steps map nearly one-on-one with the general ill-structured problem solving steps explained above. The exceptions are that engineering problem solving may not involve certain steps of ill-structured problem solving like “Assess the Viability of Alternative Solutions by Constructing Arguments and Articulating Personal Beliefs” and “Monitoring the problem space and solution options” except in engineering design in which the alternative solutions have to be considered and justified based on the goals of
the organization (rather than personal beliefs) and the monitoring process depends greatly on the history and beliefs of the organization itself.

Table 6 – Comparison of engineering and ill-structured problem solving strategies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Define</td>
<td>Problem Definition</td>
<td>Learners Articulate Problem Space and Contextual Constraints</td>
</tr>
<tr>
<td>Explore</td>
<td>Identify and Clarify Alternative Opinions, Positions, and Perspectives of Stakeholders</td>
<td></td>
</tr>
<tr>
<td>Plan</td>
<td>Idea generation</td>
<td>Generate possible problem solutions</td>
</tr>
<tr>
<td>Plan</td>
<td>Engineering Analysis</td>
<td>Assess the Viability of Alternative Solutions by Constructing Arguments and Articulating Personal Beliefs</td>
</tr>
<tr>
<td>Plan</td>
<td></td>
<td>Monitor the problem space and solution options</td>
</tr>
<tr>
<td>Do it; Check</td>
<td>Design refinement</td>
<td>Implement and monitor the solution</td>
</tr>
<tr>
<td>Generalize</td>
<td></td>
<td>Adapt the solution</td>
</tr>
</tbody>
</table>

In [45], Sobek & Jain characterize the relationship between twelve engineering design process variables and project outcomes and design quality. Even though there has been a lot said on the engineering design process [53], we choose to discuss this paper here because it treats engineering design as a problem solving process rather than as a general design process. We hope to get some insight into the process that is engineering design problem solving and extrapolate to general engineering problem solving. In [45] each design-related activity received two codes. The first is level of abstraction at which it occurred which is three levels.

i. Concept design (C) addresses a problem or sub-problem with preliminary ideas, strategies, and/or approaches. Common concept design activities are identifying customer needs, establishing the design specifications, and generating and selecting concepts.

ii. System level design (S) defines the needed subsystems, their configuration and their interfaces.

iii. Detail design (D) activities focus on quantifying specific features required to realize a particular concept, for example defining part geometry, choosing materials, or assigning tolerances.

The coding scheme also defined four categories of design activity consistent with engineering design research [53].
i. Problem definition (PD) implies gathering and synthesizing information to better understand a problem or design idea through activities such as: stating a problem, identifying deliverables, and researching existing technologies.

ii. Activities in idea generation (IG) include exploring qualitatively different approaches to recognized problems, such as brainstorming activities, listing of alternatives, and recording “breakthrough” ideas.

iii. Engineering analysis (EA) involves formal and informal evaluation of existing design/idea(s), e.g., mathematical modeling and decision matrices.

iv. Finally, design refinement (DR) activities include modifying or adding detail to existing designs or ideas, deciding parameter values, drawing completed sketches of a design, and creating engineering drawings using computer-aided design software [45].

Quantitative and Qualitative analysis indicates that conceptual and system level problem definition, and system level idea generation and engineering analysis have significantly positive impacts on project quality; whereas (somewhat counter-intuitively), concept level idea generation and design refinement, and detail problem definition and design refinement have significantly negative impacts[45]. The results however do not mean that all of the design time should be spent doing system level engineering analysis and no detailed engineering analysis.

An interesting trend in the results is the overwhelmingly positive effect of problem definition activities at the concept and system levels. These activities involve such things as: understanding the client’s needs, gathering additional information about the problem, searching the Internet to learn about existing technologies, studying a textbook to learn about the behavior of a certain material, and so forth. In other words, problem definition included anything where solvers attempted to gain a better grasp of the problem space. **Putting this trend together with the trend that idea generation doesn’t always positively impact quality, it seems that it may be more productive for solvers to learn about existing technologies, and to learn about existing solutions to similar or analogous problems, than to brainstorm for ideas** [45]. **This result is consistent with other general problem solving research which suggests that constructing the problem space is arguably the most important step in problem solving [11-13],[16].**

On the basis of this research, the authors [45] suggest that traditional design processes should be modified in certain ways (also see Table 6 for a comparison).

i. Since understanding the problem is so important, this should be done by seeking out solutions to similar problems in the past, and understanding why/how they worked in order to ascertain their applicability to the current problem. **The problem definition activity should extend and merge with the generating alternatives phase. Notice that these suggestions make the first two steps of the engineering**
problem solving process identical to steps 1 through 3 of the general ill-structured problem solving strategy [13].

ii. Particularly with complex problems, problem-solving can and should occur at different levels of abstraction. This is consistent with the general problem strategy of sub-goaling ([12], [13] and references therein).

**Takeaway:** From the comparison in Table 6 we establish that there is an overarching engineering problem solving strategy which affects problem solving ability. Problem definition is the most important step in engineering problem solving. Thus engineers should actively seek out solutions to similar or analogous problems first.

### 3.1.4 Solving Strategies for estimation problems

In his course on approximate methods in science and engineering, Mahajan [9] demonstrates a range of tools to approximate a physical quantity. However the tools that he describes can be used to analyze any situation—from the trivial “what is the mass of air in this room” to the complex “what is the size of a hydrogen atom”—from fundamental principles, by creating a simplified model of the world in our heads which allow us to think of things simply. These tools are of three types (Figure 2). The first set of tools proceed by organizing the complexity in a problem, the second set proceed by eliminating apparent complexity and the final set employs “lossy” compression, i.e., helps to discard real complexity and create simple models. We note that Mahajan does not describe an estimation or approximate analysis process similar to the problem solving strategies discussed above, but only offers examples and hints as to how these tools can be used to tackle estimation problems.

Figure 5 [9]

<table>
<thead>
<tr>
<th>How to handle complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organize complexity</td>
</tr>
<tr>
<td>Divide and conquer</td>
</tr>
<tr>
<td>Abstraction</td>
</tr>
<tr>
<td>Discard complexity</td>
</tr>
<tr>
<td>Discarding fake complexity</td>
</tr>
<tr>
<td>Discarding actual complexity</td>
</tr>
<tr>
<td>Symmetry and conservation</td>
</tr>
<tr>
<td>Proportional reasoning</td>
</tr>
<tr>
<td>Dimensional analysis</td>
</tr>
<tr>
<td>Special cases</td>
</tr>
<tr>
<td>Lumping</td>
</tr>
<tr>
<td>Limiting cases</td>
</tr>
<tr>
<td>Springs</td>
</tr>
</tbody>
</table>

The details of Mahajan’s methods are given in Appendix A.
As mentioned in Section 2.1, the authors in [5] defined a framework to explain most estimation activity. They did so by analyzing a large set of solutions to different problems in engineering practice and research given by a varied set of people from undergraduate and graduate students to industry and academia practitioners and several fields of engineering. Their analysis resulted in the identification of eight effective actions that are sufficient to explain the solutions to a large number of estimation problems. We observe a couple of things about these actions:

1. The authors do not attribute any order to these actions as in the case of the general problem solving process. In fact a solution need not even require the usage of all these actions.

2. Since these actions “describe what was effectively done and not actually done”, a set of actions describing a solution do not correspond to a unique solution process. Different solvers can arrive at different (or perhaps even the same) solutions using different subsets and different sequences of these effective actions. **In other words, different solvers can apply the same effective action in different ways and at different points in their estimation process. Their usage of the effective actions and their estimation process depends upon their cognitive and affective abilities.**

These effective actions and compensation methods are summarized in Appendix B.

In [8], the author offers some suggestions on how engineers should approach engineering analysis.

1. The first step in engineering analysis is to model the physical system that one is trying to analyze.
2. The second step is to validate the model through a test or by relaxing the assumptions. The nature of the estimate that we will obtain (rough vs. detailed) will depend on the nature of the model we use.
3. Foundation facts, depending on the engineering specialty, play an important part in the estimation process and so engineers should carry these with them.
4. There are several simple formulae, basic facts, rules of thumb, key relations and equations that are valuable in estimation. These must be easily accessible to the solver.

Shakerin [6] synthesizes the above tools and strategies to perform estimation in science and engineering as follows: “Successful engineering estimation is performed based on knowledge of dimensions and units, basic laws of physics and modeling, the ability to relate and compare, and common sense. Like many other attributes, an engineer’s ability to estimate is enhanced and strengthened by experience and gaining professional judgment.”

While the above strategies focus on engineering estimation, we now take a step back and explore strategies used for measurement estimation to see if we can borrow them
in engineering. [7] presents an in-depth review of research in the strategies for performing measurement estimation. We present a comparison of measurement estimation strategies with engineering estimation in Appendix A. **We observe from this comparison that there are some heuristics which can be applied across the board for estimation problems and perhaps even other types of problem solving.**

| Takeaway: While there exist several heuristics to solve estimation problems and perform approximate analyses in engineering, literature does not contain an overarching strategy that can be applied while solving estimation problems. |

3.2 Cognitive and affective abilities

In this section we elaborate on some of the internal factors that affect problem solving ability [3], [23]. Several of these factors are identified from the expert-novice studies in problem solving which will be described in the next section; the abilities that affect problem solving ability are those that experts have been found to employ successfully in problem solving.

3.2.1 Cognitive and affective abilities for general problem solving

3.2.1.1 Familiarity:

Familiarity with a type of problem is perhaps the strongest predictor of an individual’s ability to solve a problem. Solvers experienced in solving a certain type of problem will have well-developed schema for that problem type [26], [28], [29] and hence be able to proceed to the solution directly. However as shown in [25], the ability to solve a certain problem type doesn’t necessarily transfer to other types of problems or even the same problem type described differently. Therefore problem-solving instruction will have to be targeted towards facilitating this transfer.

3.2.1.2 Domain and Structural Knowledge:

As explained in [3], strong domain knowledge base is necessary for problem understanding and generating solutions. However it is important to remember that the knowledge base may contain things that are not true [10] as solvers may bring misconceptions and misremembered facts to problem situations and these can affect problem solving ability adversely. In addition, this domain knowledge needs to be well organized in the solvers’ memory so that related concepts are connected together and can be accessed easily. It has been found that the manner in which physical concepts and equations are connected in memory (schema) is an important predictor of problem solving ability ([3], [12], [23] and references therein). For instance, when solvers need to make decisions during problem solving, they may have several options available to them. They may fail to pursue some options because they overlooked them or because they didn’t know of their existence [10]. The former is an issue with not seeing the right connections or of poor structural knowledge. The latter case is a matter of not having
the right knowledge at all. Thus both these domain and structural knowledge are necessary for problem solving [10].

3.2.1.3 Cognitive Controls:

Individuals vary in their cognitive styles and controls, which represent the way they think and reason about information [23]. Cognitive controls, such as field independence, cognitive complexity, cognitive flexibility, and category width are most likely to affect problem solving ability [23]. While there is some experimental evidence for the fact that field independents are better problem solvers [references in 23] and individuals with higher cognitive complexity and cognitive flexibility should be better problem solvers owing to their better reasoning ability [references in 23], further research is needed to substantiate these claims about cognitive controls.

3.2.1.4 Metacognition:

Metacognition is described as the awareness of how one learns, the ability to judge the difficulty of a task, the monitoring of understanding, the use of information to achieve a goal and the assessment of learning progress [23]. Monitoring and assessing progress during problem solving and acting in response to these assessments are the core components of metacognition or self-regulation in problem solving [10]. The role of metacognition has received considerable attention in problem solving research ([23], [10] and references therein). Studies have found that when solving problems, good problem solvers work to clarify their goals, understand the concepts and relationships among the elements of a problem, monitor their understanding and choose and evaluate actions that lead toward the goal [23]. Problem solving requires knowing not only what to monitor but also how to monitor one’s performance and sometimes unlearning bad habits [23].

3.2.1.5 Epistemological Beliefs:

Problem solving, especially ill-structured, often requires the solver to consider the truth of ideas and multiple perspectives in evaluating problems or solutions. The ability to do so depends partially on the solvers’ underlying epistemic beliefs about knowledge and how they come to acquire it. Learners’ stage of epistemological development will decide how they incorporate these multiple perspectives and ideas into their analysis and evaluate them to arrive at a decision [23]. In [10], Schoenfeld describes how students with the belief that a typical homework problem should be solvable in a few minutes, will give up working on a problem after a few minutes of unsuccessful attempts, even though they might have solved it had they persevered. Schoenfeld [10] also found that students abstract their beliefs about formal mathematics in large measure from their experiences in the classroom and students’ beliefs shape their behavior in ways that have powerful and often negative consequences. Thus it is very important that instruction on problem solving include ways to positively affect students’ epistemic beliefs about the nature of the enterprise as this can affect their problem solving ability.
3.2.1.6 Affective and conative abilities:

Cognitive and metacognitive processes are a necessary but insufficient requirement for problem solving, which is also affected by several affective (like self-confidence) and conative (like motivation) elements as well as perseverance [3]. Knowing how to solve problems and believing that you know how to solve problems are often two different things. Research has shown that when students are faced with a problem in particular domains, some of them will disengage immediately believing it to be too difficult while some would continue to try finding a solution [10], [23]. Thus self-confidence will predict the level of sincere effort and perseverance that will be applied to solving the problem and thus problem solving ability. Personal biases also affect problem solving ability; for instance, solvers will create a problem representation that reflects their world view and biases. Because of this reason what is a problem for one person may not be a problem for another person [12], [16], [21].

3.2.1.7 Practices:

In [10], Schoenfeld quotes Resnick regarding mathematics instruction: ‘Becoming a good mathematical problem solver -- becoming a good thinker in any domain -- may be as much a matter of acquiring the habits and dispositions of interpretation and sense-making as of acquiring any particular set of skills, strategies, or knowledge. If this is so, we may do well to conceive of mathematics education less as an instructional process (in the traditional sense of teaching specific, well-defined skills or items of knowledge), than as a socialization process.’

Indeed expertise in any domain is more than just acquiring a set of skills of the domain [10 and references therein] – instead it is about acquiring a way of thinking, a way of seeing, a set of values and perspectives of the domain. In the case of engineering estimation, this theory perhaps explains the differences between senior engineering students and practicing engineers. Since the latter are a part of the engineering community they have acquired the values and perspectives of the community and hence perform engineering estimation better, as is the case in the other communities of practice [10 and references therein].

3.2.2 Cognitive and affective abilities for ill-structured problem-solving

Hong [41] classifies the abilities needed to solve ill-structured problems as follows.

3.2.2.1 Cognition:

As in the case of well-structured problems, evidence has also been found [Voss et al 1991, referred here from 41] that if a solver does not have or employ substantial domain knowledge, the solving strategy will lead to inadequate solutions. Further structural knowledge is important, for example, in the construction of the problem space as evidenced by the fact that experts have the ability to seek out critical and related information and terminate immediately after a sufficient amount of information has been collected [Voss & Post 1988, referred here from 41]. Solving ill-structured
problems may require domain-specific structural knowledge in several different domains [3].

3.2.2.2 Metacognition:

As described in Section 4.2.1, a lot of research on metacognition has centered on mathematical problem solving [10], which tends to be well-structured. In [41] Hong et. al. found evidence that metacognitive skills are more important to ill-structured problem-solving than well-structured problems. This is because the goals and methods are ill-defined so solvers have to use metacognitive skills such as changing strategies, modifying plans and reevaluating goals [41]. Metacognition has the following parts.

- Knowledge of cognition: Knowledge of cognition includes three sub processes such as "knowledge about self and about strategies, knowledge about how to use strategies, and knowledge about when and why to use strategies" (Schraw & Dennison, 1994, adapted here from [41]). Thus, domain-independent general strategies (for example, decomposition) are required to make progress on unfamiliar problems [10]. Additionally, information gathering search strategies and argumentation strategies are needed to solve unfamiliar problems [41].

- Regulation of cognition: Monitoring (of one’s own cognitive efforts, the effects of these efforts, the progress of the activity, one's emotional reactions), evaluation (of the potential usefulness of information), and planning (revising based on reflection, evaluation and monitoring) are called regulation of cognition as a part of metacognition [41].

However there is no research on the exact role of metacognition in ill-structured problem-solving and it is reasonable to expect that the specific metacognition and self-regulation skills needed will depend on the type of problem being solved [3].

3.2.2.3 Non-cognitive variables:

In general ill-structured problems possess multiple representations. The solvers must choose one which is relevant and useful in solving the problem based on their own cognitive and affective knowledge. Sinnott (1989) (adapted here from [41]) found that solvers personal beliefs, emotions and unrelated thoughts at the time of problem-solving often directed this choice and Voss (1988) (adapted here from [41]) suggested that solvers construct goals from his/her personal life. Further non-cognitive variables motivate the solver to continue through the process of problem solving [13] and create continual awareness of the problem which improves information gathering and use of search strategies [41].

1. Epistemic Cognition: Solvers’ epistemic cognition influences how individuals understand the nature of problems and decide what kinds of strategies are appropriate for solving them [17]. Kitchners’ [17 and references therein] claims that in addition to cognition, both metacognition and epistemic cognition have to be employed by the solver for ill-structured problems were partially tested by Gregory Schraw and his colleagues [17]. Their results showed that 1) performance on the
well-structured task was independent of performance on the ill-structured task; 2) epistemic beliefs explained a theoretically significant proportion of variation in ill-structured solving; and 3) epistemic beliefs failed to explain meaningful proportion of variation in well-structured solving. An example of epistemic beliefs is provided by Greeno (quoted here from [3]); the prevalence of algorithmic problem solving in mathematics and other STEM disciplines, leads to “a belief by students that mathematical problems are solved by applying procedures that a person may or may not know.”

In addition to epistemic beliefs, other components of epistemic cognition which influence ill-structured problem-solving ability are strategic planning and identifying constraints and assumptions [17 and references therein].

2. Justification skills are needed when the solvers identify the causes of a problem and select a solution because they have to provide a viable and cogent argument in favor of their chosen alternatives [13]. The solver’s epistemic cognition is an important component in order to develop justification (Kitchner 1983, from [41]). For example, if individuals believe all problems have a single solution, they may try to apply the procedures to obtain a single valid solution rather than considering potential alternative solutions [41]. This was observed by Shekoyan [17] in the case of physics multiple possibility problems (MPPs) where novices did not consider alternative problem scenarios at all.

In Table 7 we compare the cognitive and affective abilities needed for different types of problem solving.

3.2.3 Cognitive and affective abilities for engineering problem solving

The abilities described in previous sections are operationalized by Woods et. al. [15] into the following set of problem solving skills for engineers which they have used to develop a problem solving program for engineering undergraduates.

1. Being aware of the processes used;
2. Using pattern matching to quickly decide whether a situation is a problem or an exercise;
3. Applying a variety of tactics and heuristics; (This actually fits with problem solving strategies)
4. Placing an emphasis on accuracy (as opposed to speed.);
5. Being active by writing down ideas, creating charts and figures;
6. Monitoring and reflecting on the process used;
7. Being organized and systematic;
8. Yet being flexible (keeping options open, seeing the situation from many different perspectives and points-of-view);
9. Drawing on the pertinent subject knowledge and objectively and critically assessing the quality, accuracy and pertinence of that knowledge and data;
10. Being willing to risk and cope with ambiguity, welcoming change and managing distress;
11. Being willing to spend time reading, gathering information and defining the problem (as opposed to equating problem solving with “doing something” despite its pertinence);
12. Having an overall approach that uses fundamentals rather than trying to combine various memorized sample solutions.

Knowledge of these traits and abilities identified from research are employed to create strategies for teaching problem solving that target the development of these skills and abilities. We review some of these strategies in Section 6.

Table 7 – Comparison of cognitive and affective abilities needed for different types of problem solving

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiarity</td>
<td></td>
<td>Using pattern matching to quickly decide whether a situation is a problem or an exercise;</td>
<td>Knowledge; Visualization skills</td>
</tr>
<tr>
<td>Domain and structural knowledge</td>
<td>Cognition</td>
<td>Being active by writing down ideas, creating charts and figures; Drawing on the pertinent subject knowledge and objectively and critically assessing the quality, accuracy and pertinence of that knowledge and data;</td>
<td></td>
</tr>
<tr>
<td>Metacognition</td>
<td>Metacognition</td>
<td>Being aware of the processes used; Placing an emphasis on accuracy (as opposed to speed.); Monitoring and reflecting on the process used;</td>
<td></td>
</tr>
<tr>
<td>Epistemological beliefs</td>
<td>Epistemic cognition (includes epistemic beliefs and justification skills)</td>
<td>Yet being flexible (keeping options open, seeing the situation from many different perspectives and points-of-view); Being willing to spend time reading, gathering information and defining the problem (as opposed to equating</td>
<td>Mental abilities and beliefs; Anchoring effects</td>
</tr>
</tbody>
</table>
problem solving with “doing something” despite its pertinence);

<table>
<thead>
<tr>
<th>Cognitive controls</th>
<th>Affective and conative elements</th>
<th>Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Being willing to risk and cope with ambiguity, welcoming change and managing distress;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Preferences or biases;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Being organized and systematic; Having an overall approach that uses fundamentals rather than trying to combine various memorized sample solutions.</td>
<td></td>
</tr>
</tbody>
</table>

3.2.4 Cognitive and affective abilities needed for estimation problem solving

We did not find any literature on the components of problem solving ability for estimation problems. However in [5] the authors comment on a set of mediating characteristics which influence the manner in which solvers take effective actions during problem solving. Some of these are:

1) Knowledge

2) Mental abilities and beliefs: These are shaped by experience. For example, students believe they don’t have enough knowledge to solve a problem.

3) Characteristics of situation like resources available and nature of surroundings. (This is a feature of the problem itself and thus an external factor affecting problem solving.)

4) Visualization skills

5) Anchoring effects: This is a cognitive bias describing the human tendency to give more importance to the first piece of knowledge offered (the “anchor”) when making a decision. This effect is important for people who provide a value based on less knowledge.

6) Preferences or Biases: People have a bias towards a certain method, equation, principle or system. For example, mechanical engineering students prefer to use the drag equation even though there may be simpler ways to solve a problem.

See Table 7 for a comparison of these abilities with other problem solving abilities. *We observe that this is not a comprehensive list of abilities needed for estimation and should be added to the abilities needed for engineering problem solving. In fact there may be other abilities which need to be identified in order to design effective instruction strategies for estimation problem solving.*
There is a plethora of research studying the differences between experts and novices in terms of domain knowledge and problem solving skills [11], [26-34]. This line of inquiry aims to understand how experts’ knowledge, knowledge organization and problem solving ability differs from that of novices so that instruction can be tailored with the goal of getting novices to behave like experts. The expert novice differences are studied by analyzing the verbal or written protocols of experts and novices as work on problems.

The first dimension along which experts differ from novices is knowledge and knowledge organization. [26] describe the key principles of experts' knowledge and we reproduce them here because as we will see in the next section, well-organized domain knowledge [3], [10], [11], [21-23] is one of the most important requirements for successful problem solving.

1. Experts notice features and meaningful patterns of information that are not noticed by novices.
2. Experts have acquired a great deal of content knowledge that is organized in ways that reflect a deep understanding of their subject matter.
3. Experts' knowledge cannot be reduced to sets of isolated facts or propositions but, instead, reflects contexts of applicability: that is, the knowledge is conditioned on a set of circumstances.
4. Experts are able to flexibly retrieve important aspects of their knowledge with little attentional effort.
5. Experts have varying levels of flexibility in their approach to new situations.

In the Table 8 we summarize some key expert-novice differences during problem solving and elaborate later.

Table 8 - Expert novice differences in general problem solving

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Expert</th>
<th>Novice</th>
<th>Supporting Research Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>Have up to 50000 chunks of familiar stimuli, each with several pieces of information</td>
<td>No or smaller chunks</td>
<td>Chase &amp; Simon 1973, deGroot 1965</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Organized</td>
<td>Organized around</td>
<td>Eylon &amp; Reif 1984, Chi,</td>
</tr>
</tbody>
</table>

Takeaway: Different types of problems require different cognitive and affective abilities and we need to identify these in order to design effective instruction for each type of problem solving.
The first differences between the performance of expert and novice problem solvers come from studies by deGroot (1965) and Chase & Simon (1973). These researchers found that when an expert and a novice were shown a chess position for only five seconds the experts could recall it with very high accuracy while the novice players could not. Further studies found that this was not because of the inherently superior memory of the experts because upon being shown a random chess board the experts had as much difficulty recalling the positions as the novice. As explained in [12], Chase & Simon found that this was because the experts perceived groups of chess pieces together called chunks. While the novices also performed chunking, their chunks were significantly smaller than that of the experts.

The second difference lies in the knowledge organization of experts and novices. A large body of knowledge is a pre-requisite to expertise in problem solving; however this knowledge must be organized and indexed so that it can be accessed quickly when needed. Research [29] has found that the large set of chunks held in the experts’ memory serves as an index to the experts’ factual and procedural knowledge. In other words, the indexed memory is organized as a set of productions or condition-action pairs. When a pattern is recognized (i.e. the condition of a production is satisfied) then the action is automatically invoked. In this model, the condition is the index and the
action is the content. Thus recognition of a pattern from memory invokes along with it the set of actions and strategies for problem solving that may be suitable when the pattern is present.

The third and perhaps most significant difference between experts and novices lies in the way they categorize problems [28]. The authors in [28] found that the novices grouped problems according to their surface features like the physical objects described in the problem statement while experts classified problems according to their "deep structure", i.e., according to the physical laws that will be used to see the problem. The authors also found that with learning, novices start behaving like experts in their identification of problems by deep structure rather than surface features which was confirmed by instructional studies in [32]. Further experiments [26], [28], [29] found that experts appear to associate his/her principles with procedural knowledge about their applicability, reinforcing previous results about productions. In other words, experts' schema contain a great deal of procedural knowledge, with explicit conditions for applicability. Novices' schema, on the other hand, may be characterized as containing sufficiently elaborate declarative knowledge about the physical configurations of a potential problem, but lacking abstracted solution methods. Their surface-oriented categorizations yield equations associated with these surface components. So novices were unable to formulate solution methods at intermediate levels of abstraction between general prescriptions for how to proceed and highly specific equations.

For experts, the process of problem representation occurs over a span of time and involves interplay between the problem statement and the knowledge base, even during the reading of the problem [28]. Literal cues from the problem statements are transformed into second-order (derived) features which activate a category schema for a problem type. This schema is organized by a physics law. It guides completion of the problem representation and yields a general form for the equations to be used in problem solution. For novices, problem representation is organized by schemata for object categories, for example, "spring problems" or "falling bodies." These yield equations specific to problems at these levels, and much of the process of problem representation involves substituting variables in these equations. This difference was also observed by Simon & Simon (1978) as mentioned in [29].

The fourth difference between experts and novices was that the former solved the problems in less than one-quarter of the time required by the novice and with fewer errors [29]. The fifth difference was that the novice solved most of the problems by working backward from the unknown problem solution to the given quantities while the expert usually worked forward from the givens to the desired quantities [29]. Since novices had little experience with the domain they seemed to require goals and sub-goals to direct their search. The management of goals and sub-goals takes considerable time and places a substantial burden on limited working memory. Sweller [26] later elaborated on this difference. Experts are able to work forward immediately because
they recognize each problem and each problem state from previous experience and know which moves are appropriate due to the presence of schema. Novices, not possessing appropriate schema, are not able to recognize and memorize problem configurations and are forced to use general problem-solving strategies such as means-ends analysis when faced with a problem. Means-ends analysis places a significant cognitive load on the solver’s working memory which impedes schema acquisition which is the key to deep conceptual understanding [26].

A few additional differences between experts and novices adapted from [31] and [36] are summarized here.

1. Experts often make solving problems look easy, which causes novices to mistakenly think that they should be able to understand and solve problems easily too.

2. Experts use qualitative representations extensively, while novices have trouble with representations.

3. Experts have many techniques to redescribe the problem statement, they take time defining tentative problem and may redefine several times while novices have difficulty redescribing the problem statement and tend to jump to conclusions.

4. Experts are able to draw inferences from incomplete data while novices cannot. This is especially important in engineering problems which are ill-defined.

5. Experts are able to analyze complex problems in parts, proceed in steps and look for patterns.

6. Experts spend time exploring the problem and representing it in multiple ways while novices tend to jump to the equations.

7. Experts tend to do limiting cases analysis before proceeding with analytically intensive problems to get an idea of the solution. This is especially useful in engineering analyses.

8. One of the most important differences lies in the fact that experts monitor their progress, check for accuracy and evaluate their result which novices do not do.

Finally, we make one comment on the difference between professors and students when working on a problem which was unfamiliar even to the professors. In [35] the authors found that when faced with such a problem, even professors had difficulty in solving the problem under the time constraint. With one exception, none of the subjects in either group was able to solve the task in the time allowed. Still the professors initially employed a systematic approach, which included visualizing the problem, considering various conservation laws, making simplifying assumptions and examining limiting cases. After finding that familiar techniques were not fruitful, they made incorrect predictions based on one of two equally important factors. The responses of students who were given the same problem reflected no overarching strategies or
systematic approaches, and a much wider variety of incorrect responses were given. This is an important result in the context of estimation problems because even experts will often find themselves facing a new situation when dealing with estimation. This study demonstrates that in such a scenario a systematic approach along with experience and intuition gives the solver the best chance at arriving at a good estimate.

**Takeaway:** Expert-novice analysis shows that gaps exist between experts and novices along every dimension of the problem solving ability. Hence instruction is needed to bridge these gaps.

4.2 Expert-Novice differences in ill-structured problem solving

In this subsection, we elaborate on the additional differences that surface between experts and novices in the solving of ill-structured problems. Differences in epistemic cognition were described in the recent research by Shekoyan [17] on the solving of multiple possible problems (MPPs) in physics. In this study, he compared the performance of professors and undergraduates by conducting think aloud protocol analysis as they solved a physics MPP. Shekoyan assumed that a problem solver is engaging in epistemic cognition if she/he keeps asking herself/himself epistemic questions while solving a MPP problem or makes statements which can be considered responses to these questions. The questions are “How do I know this?”, “Am I making any assumptions?”, “Are these assumptions valid?”, “Are there alternative reasonable assumptions?” and “Are there other possible outcomes?” The problems presented by Shekoyan were such that the problem situation was not explicit and several possibilities needed to be considered to successfully solve the problem. He found that:

1. Almost all of these possibilities were considered by all but one of the experts. In other words, experts asked and/or answered more questions of the type described above. Only one novice exhibited expert-like behavior. Thus epistemic cognition level of experts is higher than the epistemic cognition level of novices.

2. Novices almost always needed at least one epistemic prompt from Shekoyan to consider the options and in some cases did not consider the possibilities even after several prompts. So we conclude that novices on average lacked the necessary level of epistemic cognition to be successful in solving MPP problems beyond a basic level of ill-structuredness.

3. Further in the case of experts the effectiveness of the prompting was 100%. In the case of novices, although the effectiveness of prompting was different for different students, overall it was 50% effective. **This suggests that while epistemic questioning can be improved with instruction, better teaching-learning strategies are needed to improve students’ epistemic questioning.**

**Takeaway:** Experts and novices differences in epistemic cognition affect their ability to solve ill-structured problems. Therefore good teaching-learning strategies need to be designed to improve students’ epistemic cognition.
4.3 Expert novice differences in engineering problem-solving

In [45], Adams et. al. conducted twenty five semi-structured interviews with engineering undergraduates, academics and professional engineers with the aim of understanding the differences between expert and novice problem-solvers in engineering. Their analysis results in the following observations:

1. Responses to the question “What qualities do you think make a good engineering problem solver?” indicate a continuum from students to professionals on the importance placed on process skills such as thinking and reflection.

2. Responses tend to suggest an increased importance placed on method and strategy by professionals.

3. While students were able to identify a range of important skills these were often presented in a rather disconnected way.

4. Academics identified a set of issues but like students tended to focus on the understanding of the question and learning by practicing rather than the importance of identifying and developing method.

5. Motivation also varies across this continuum with students being motivated by more immediate and tangible commodities such as grades and employment opportunities. Professionals tended to take a different view of motivation, relating this to having an enquiring mind and the desire to explore engineering issues more liberally.

The list of skills that emerged from analysis of responses is listed in the first three columns of Table 9 adapted from [45].

Table 9 – Comparison of student, academic and professional engineer problem solving skills

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of question</td>
<td>Experience and Practice</td>
<td>Naturally enquiring mind</td>
<td>Ongoing learning</td>
</tr>
<tr>
<td>Practice</td>
<td>Trial and error</td>
<td>Asking questions and asking others</td>
<td>Ongoing learning; self-assessment; Finding information; Continuous improvement; Balance</td>
</tr>
<tr>
<td>Analytical skills</td>
<td>Prior knowledge</td>
<td>Scoping of problems</td>
<td>Representation of a text problem; Representation using scientific concepts; concrete goal; how things work; finding information; external representation</td>
</tr>
<tr>
<td>Variability in terms of reflection</td>
<td>Questioning and listening</td>
<td>Balance</td>
<td></td>
</tr>
<tr>
<td>Ownership</td>
<td>Motivation:</td>
<td>Sorting information; synthesis</td>
<td>Motivation to make things better</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>Math skills</td>
<td>Good understanding of what to be achieved</td>
<td>Making things basic/simple</td>
<td>How things work?; representation using scientific concepts; concept of estimation; units; recognizing opportunities; cost effective; Balance;</td>
</tr>
<tr>
<td>English skills</td>
<td>Skills and not content or knowledge</td>
<td>Thinking skills</td>
<td>Schema- deep understanding; language of the discipline; scientific concepts; assumptions; self –assessment; strong learning process; ongoing learning; external representation; metacognition</td>
</tr>
<tr>
<td>Consulting with others</td>
<td>Thinking in different ways (pictures)</td>
<td>Verbal, visual and math representation; external representation</td>
<td></td>
</tr>
<tr>
<td>Looking from other viewpoints</td>
<td>Logic skills/ process skills</td>
<td>Framework (schema) for procedural knowledge;</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Analysis and application of analysis</td>
<td>Units; calculation; concept of estimation; recognizing opportunities; cost effective; representation using scientific concepts; assumptions;</td>
<td></td>
</tr>
<tr>
<td>Priorities and focus</td>
<td>Recognize what you do and don't know</td>
<td>Balance; scientific honesty; ongoing learning; continuous improvement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flexibility in method</td>
<td>Balance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reflection on method and having other strategies</td>
<td>Metacognition; Reviewing a solution; continuous improvement; error checking; self –assessment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Risk taking</td>
<td>Balance</td>
<td></td>
</tr>
</tbody>
</table>
As part of their research, Don Elger and his colleagues [46] identified a set of thirty criteria for quality performance in problem solving which together describe the standard practices of the engineering community, i.e. what experts do (full details in Appendix D). They assessed eight out of these thirty by surveying faculty and student beliefs on the performance objectives. They chose to focus on problem solving that involves calculations based on mathematical representations of scientific concepts, analysis and modeling, which is a good description of all engineering analysis problems [1], [2], and [4]. However this definition of quality is not intended to be universally acceptable [46].

In Table 9, we compare the expert behaviors that emerged from [45] and [46] and find that the former provides an overall comparison of expert vs. novice problem-solving behavior and the latter focuses on the engineering problem-solving method of experts. Further the trait called balance [46] defined as “Engineers balance persistence and perspective; they work hard, but they will take appropriate action if the effort to solve the problem becomes too much. By appropriate action, we mean performances such as asking for help, letting it go, or redefining the problem so that it is simpler” is the most important characteristic of the expert engineer’s affective ability.

**Takeaway:** Both the set of engineering expert behaviors in Table 9 are not actual behaviors demonstrated by experts as they solved problems, but expert beliefs on what their behavior should be like. Thus there is a difference between this expert-novice analysis and the one performed for general and ill structured problem solving; the latter highlights actual expert behavior during the solving of a problem and so is more useful to design effective instruction. Still these defined engineering expert behaviors can guide us as we choose appropriate instruction for estimation problem solving.

In this section we present results from two studies comparing experienced and novice estimators. The first study comes from Benjamin Linder [5] who compared the performances of experienced practicing engineers with senior engineering students on engineering estimation tasks. The second set of results come from Joram et al [7] who review several studies and compare the performance of adults and children on measurement estimation tasks. We summarize the results of the first paper in Table 10 and second paper in Appendix E.

The students in the study in [5] were seniors in mechanical engineering at MIT and five top engineering universities. The practitioners studied were all attendees of a plenary talk at an American Society of Engineering Education conference. Practitioners involved in academics were chosen because they have knowledge and backgrounds similar to senior students. The setting was chosen because of its similarity to the lecture setting used for the students. Engineers have median experience between 26 and 30.
years. The authors make the following observations based on the detailed description of student responses:

1. Students have difficulty making rough estimates because they don’t have a sound understanding of fundamental engineering concepts, much lesser in fact than was expected.

2. Students did not relate the estimates they made to their physical significance. Also, they don’t have reference values for the quantities they are estimating and have difficulties working with units.

3. The authors suggest that students knowledge of units, or lack of it, may be an indicator of their deep conceptual understanding of engineering concepts and hence a useful predictor of their ability to make rough estimates. If students have great difficulty associating the correct units with quantities, they will also have difficulty making estimates since estimates involve several quantities.

Some overall conclusions drawn from the results in [7]:

1. Children and adults performed poorly in the early studies on measurement estimation. Although accuracy of estimates improved with age, reasonable estimates were given by only about 30 to 40% of high school seniors and fewer than 50% of adults, on average.

2. In the next set of studies, again increases in accuracy with age were observed. However the average accuracy level of children and adults in these studies was low and performance was found to vary widely across attribute being estimated.

   Again these studies only serve to highlight that there exist differences between expert and novice estimators’ performances. However what in their behavior contributes to these differences is not understood from this research. Therefore we don’t know what is the target estimation behavior that we want to teach our student engineers.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Expert performance</th>
<th>Novice performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimating the drag force on a cyclist</td>
<td>All of the mechanical engineers’ answer values were on the right order of magnitude. The estimates of the electrical engineering practitioners only varied over 2 orders of magnitude.</td>
<td>The students’ answer values indicate the right order of magnitude; however, their answers cover six orders of magnitude not including outliers.</td>
</tr>
<tr>
<td>Estimating the energy stored in a 9V battery.</td>
<td>The mechanical engineering practitioners did noticeably better than the students, and the electrical engineering practitioners did substantially better. Their answer values indicate the right order of magnitude and do not vary by more than four orders of magnitude.</td>
<td>The MIT seniors' answer values do not indicate the right order of magnitude and cover a range of nine orders of magnitude not including outliers. The seniors at other universities responded similarly. Their answer values range by ten orders of magnitude.</td>
</tr>
</tbody>
</table>

**Takeaway:** While expert behavior is quite well understood for general, ill-structured and engineering problem solving, it is not so in the case of estimation problem solving. This makes instructional design a harder problem. Instructional design will have to be preceded by expert analysis to identify target behavior.

### 5 Teaching - Learning strategies for problem solving

As observed from the previous section, there are differences between the problem solving ability of experts and novices and the goal of instruction should be to reduce these differences. There are two approaches to reduce these differences.

1. Identify the difference between experts and novices along each dimension of the problem solving ability and design instructional activities to reduce this difference. Thus by performing the set of learning activities the novices' overall problem-solving ability is improved towards the expert ability.
2. Teach a problem solving method similar to the experts’ method, ensuring that the novice understands and emulates it. Then via extensive practice the novices' problem solving ability tends to the expert ability.

Different researchers prefer different teaching approaches. In this section, we review literature on studies performed in teaching problem solving.

#### 5.1 General problem solving teaching-learning strategies

In this subsection, we describe some strategies for classroom teaching of problem solving and review the results of some instructional studies. We begin with the method of [13], in which Jonassen proposes an instructional design for supporting problem solving which has six steps. This design is supposed to teach students the problem solving strategy proposed by Jonassen [13] for general problem solving. Similar recommendations have also been made by Heller and Hungate (as referred from [10]) based on detailed studies of cognition.

1. Make tacit processes explicit
2. Get students talking about processes
3. Provide guided practice
4. Ensure that component procedures are well learned
5. Emphasize both qualitative understanding and specific procedures
1. **Review Prerequisite Component Concepts, Rules, and Principles**

2. **Present conceptual or causal model of the problem domain**: Instructional studies [13 and references therein] have concluded that providing concrete, conceptual models for learners improves conceptual retention, reduces verbatim recall, and improves problem-solving transfer. One way to support learner construction of conceptual models is via concept maps which convey important structural knowledge and helps learners to build appropriate domain-specific problem representations [13].

3. **Model Problem Solving Performance in Worked Examples**: Worked examples are useful because they help learners to construct useful problem schema. In fact research has shown that problem-solving performance improved more after studying as few as two worked examples than from solving several well-structured problems because worked examples show not only the process used for solving the problem but also some of the reflective thinking that is essential to that process which novice learners may not possess [13 and references therein]. In [13] the authors also offer several suggestions on how to construct worked examples; for the sake of brevity we do not reproduce them here. We only make one comment that they should probably not be developed using experts to model the process since experts have automated the process to the extent that they often do not articulate all steps. Rather, it is important to use an intermediate learner as a model, someone who is competent and experienced in solving this type of problem, but someone who can still articulate all aspects of the problem state and solution.

Chi, as described in [11], based on instructional studies into the use of worked examples in the teaching of problem solving, described a method by which students can learn from worked examples. This method includes two processes called inferential processes and self-repairing. When the solution omits some information, the learner employs the inferential process to explain to themselves what is missing. When there is a conflict between the learner's mental model and the scientific model described in the worked example, the learner must self-repair, i.e., recognize that there is a defect in his/her mental model and address it to gain the learning benefit from the worked example.

Since viewing of worked examples is a passive process, questions arise about the transferability of problem-solving skills acquired from merely viewing worked
examples. There are a lot of studies [13 and references therein] which have compared worked examples with more exploratory approaches, finding that learning by trial and error takes more time than rule-based learning, but that it promotes transfer of learning to new tasks. As mentioned in [26], since learners may spend time on incorrect solution paths, they may fail to acquire good mental models. Other research has also found that students receiving worked examples solved related problems effectively while learners in the exploratory group solved more far transfer problems [13 and references therein]. So in order to support transfer of learning we need to supplement our problem solving instruction strategy.

4. **Present practice problems**: A combination of worked examples plus extended practice is most likely to facilitate the acquisition of problem schema and the transfer of those schema to novel problems [13]. Practice problems should be presented to the learner in the form in which they will be assessed and the inclusion of irrelevant problem information should be consistent between worked examples, practice problems, and assessment problems. This is the most popular instructional strategy in STEM classrooms [10], [11], [15], [36] including engineering estimation [5-9].

5. **Support the Search for Solutions**: One approach is to provide analogical problems and make the problem elements of the analogical problem obvious in order to assist learners in mapping the previous problem onto the new one [13]. Another support strategy is to provide advice or hints (scaffolding) on breaking down the problem into sub-problems that can be more easily solved by highlighting relevant cues or providing a solution template. Prompts may also be given on operators or actions that can be taken to solve the sub-problems. Finally, it is essential to provide adequate feedback about learners’ attempts to solve the problem [13].

6. **Reflect on problem state and problem solution**: Learners can be asked to create tables or databases of problem types and solutions. Also asking questions after the problem solving process requiring learners to compare initial problem conditions with solutions can help students create deep understanding. Several studies on how to improve students’ reflection or self-regulation have been done in Mathematics and reviewed in [10].

In Appendix F, we review several instructional studies on problem solving that have demonstrated learning gains in the STEM disciplines. These studies give us hints on how to design general problem solving teaching strategies.

**Takeaway**: The expert problem solving method can be taught using a six step process. The instructional activities in each step are supported by extensive research into their effectiveness for learning the particular part of the problem solving method they target.

There is another set of teaching-learning strategies that simultaneously target students’ conceptual understanding and problem solving ability by engaging them in problem solving, typically real-world problems. These include problem-based learning
(PBL)[61], model-eliciting activities (MEA)[60], contrasting cases (CC)[57], productive failure (PF)[58][63], game based learning (GBL)[59][56], anchored Instruction (AI)[64] and Inventing to prepare for future learning (IPL)[62]. Each of these differs in the role that the problem plays in the instruction process, the scaffolds provided to the student and other dimensions. These differences are summarized in Appendix H.

5.2 Ill-structured problem-solving teaching-learning strategies

We begin by summarizing Jonassen's [13] six step strategy for teaching ill-structured problem solving which is based on existing research. A detailed review of instructional studies with ill-structured and engineering problems is presented in Appendix G. These studies can offer us guidelines on how to design engineering problem solving instruction.

Figure 7: Instructional design for ill structured problem solving

1. **Articulate problem context**: Instruction should contain an inventory of domain related knowledge; not merely a list of rules, principles and concepts, rather the information should be related to the context of the problem.

2. **Introduce problem constraints**: As novice learners may not be able to identify the constraints of the problem themselves, instruction has to identify these for them.

3. **Locate, select and develop cases for learners**: Instruction should include representative real world cases for students to study. These can be obtained from practitioners in the field. The problems should be interesting and challenging, but solvable. The problems should have multiple solutions and several possible problem spaces to be relevant. Also rather than having a single, large, mega case, it is better to have more number of smaller cases as this promotes transfer [13]. According to cognitive flexibility theory [48-51], “the heart of instruction is the cases that include the contextualized problems that learners must solve.”

4. **Support knowledge base construction**: It is not necessary to provide all of the information that learners need in order to solve a problem; students should be expected to search for some of the information. However, providing a structured knowledge base will scaffold this information collection activity [3]. Learners must be made aware of and be able to differentiate between the multiple perspectives and interpretations of the problem. Cognitive flexibility theory conveys problem complexity by presenting learners with multiple representations or perspectives on the information [13], [18], [48-51]. Therefore cognitive flexibility theory serves as a good model for ill-structured problem-solving instruction [13].
5. **Support Argument construction**: In this step students must be guided to think about what is known and what isn’t. This guidance can take the form of model arguments for a related problem or prompts for learners during the solving of a problem. The former can be done by presenting the learner with the arguments constructed for a related problem by different solvers. Argument templates or checklists can also be used. If prompting is used, the learners should be provided with a series of judgment prompts or questions such as, “How do you know your position is correct? Is there a correct position? How did you come to hold this opinion? What did you base your view on?” and so on. The goal of these prompts is to get the student to examine every point of view and select the best one based on reasoning and evidence.

6. **Assess problem solutions**: Assessment of student solutions is done by considering both the problem solving process and the product, namely the final solution. The final solution can mostly be evaluated on the basis of its viability. We have to check whether the problem was alleviated. Can the student explain how the problem was solved by his/her solution? Were the constraints satisfied? Are all the stakeholders satisfied with the solution? Are all the issues resolved? In terms of process, we have to consider whether the student considered important perspectives, presented sound arguments in favor of and against alternative solutions, whether domain knowledge was correctly used and so on.

In Table 11, we compare the well and ill structured problem solving teaching strategies to understand how the instruction needs to be different for ill-structured problems compared to well-structured problems.

### Table 11 – Comparison of well and ill structured problem solving teaching strategies

<table>
<thead>
<tr>
<th>Well structured problems</th>
<th>Ill structured problems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Review prerequisite component concepts, rules and principles</strong></td>
<td>Articulate problem context - Rules, principles and concepts of domain + information related to context</td>
</tr>
<tr>
<td>Present conceptual or causal model of the domain – convey structural knowledge of domain, helps conceptual retention and promotes transfer</td>
<td>Introduce problem constraints - Guide learners in the identification of constraints</td>
</tr>
<tr>
<td>Model problem solving performance in worked examples - Worked examples help construct problem schema</td>
<td>Locate, select and develop cases for learners - Cases help promote cognitive flexibility</td>
</tr>
<tr>
<td>Present practice problems - Practice problems help construct problem schema and promote transfer</td>
<td>Support knowledge base construction – provide a structured knowledge base to scaffold the information collection activity</td>
</tr>
<tr>
<td>Support the search for solutions – Scaffolds or hints of analogies or how to divide problem into sub-problems</td>
<td>Support argument construction – Scaffolds in the form of model arguments or argument templates to think about what is known and what isn’t</td>
</tr>
<tr>
<td>Reflect on problem state and problem</td>
<td>Assess problem solutions – consider both</td>
</tr>
</tbody>
</table>

In Table 11, we compare the well and ill structured problem solving teaching strategies to understand how the instruction needs to be different for ill-structured problems compared to well-structured problems.
solution – promote reflection by asking process and product and give feedback questions

**Takeaway:** The ill-structured problem solving instructional design also consists of six steps like the one for general problem solving; however Table 11 shows how the exact scaffolds and instructional activities need to be different in each step depending on what problem solving is being taught. As before, these activities are designed based on extensive research.

5.3 **Teaching-learning strategies for engineering problem solving:**

Over two a half decades Don Woods [15] and his colleagues at the McMaster University in Canada have performed studies in the teaching of engineering problem solving. They have defined problem solving skill (see section 4.2.3), identified a set of effective teaching methods to develop this skill, implemented it as a series of four required courses and evaluated its effectiveness. *Thus this teaching approach focuses on developing each dimension of the problem solving ability separately in contrast to Jonassen’s instructional designs described in the previous two sub-sections.*

1. The first step of the teaching process was identifying 37 distinct skills required for problem solving. These include 14 related to problem solving ability for well defined problems, five for solving ill defined problems, seven for interpersonal and group skills, four related to self-management, two for self-assessment, one for change management and four for lifetime learning skills [15]. Thus these cover the range of cognitive and affective abilities needed for engineering problem solving that we have discussed in section 4.2.

2. As part of their problem solving program, students take workshops to develop these skills beginning from the second year. The first course teaches well-defined, back of the textbook problem solving skill. The next course focuses on giving additional practice in the application of these skills. The third course teaches group problem solving while the final course in the final year is targeted towards developing the skills needed for open-ended problem solving and lifelong learning.

3. The authors [15] found that merely providing the students with problem solving practice or working out problems in class is ineffective in improving problem solving skills. Hence they developed a set of activities based on research in cognitive and behavioral psychology, with each activity focused on one particular skill. The activities were selected and developed based on Sternberg's [52] criteria. The goal of the activities was to give the students an opportunity to learn how to do the skill in a content-independent domain, compare their behavior with the target behavior and help them develop the target behavior through practice and prompt feedback. These activities were followed by bridging activities that allowed them to understand how the skill would be applied in a subject-domain and in everyday life.
Students were required to reflect on their experiences, monitor their progress and keep a journal to provide evidence of their growth.

Wankat & Oreovicz [36] offer a set of guidelines to instructors for teaching problem solving in engineering. *Their instructional approach is focused on teaching a research-based problem solving method that can develop problem solving ability.* Their suggestions include:

1. Embed problem solving in existing engineering courses so that specific knowledge and problem solving can reinforce each other.
2. Illustrate the problem solving strategy via worked examples. This will serve as scaffolding to students and encourage them to solve problems using the strategy.
3. Solve worked examples in class that are unfamiliar even to the instructor, detailing out every step and verbalizing as the problem is solved. The goal is to demonstrate the process that one goes through while solving new problems.
4. Require students to follow the same format when solving all problems, label steps in problem solving strategy, try out ideas, draw diagrams, define symbols, write equations and show algebraic calculations.
5. Give problems from apply-create levels of the Bloom’s taxonomy and try to cover all levels of the problem solving taxonomy (routines, diagnosis, strategy, interpretation and generation).
6. Teach and have students practice all steps of the problem solving strategy separately and encourage them to use multiple methods to solve the same problem.
7. Require students to find or estimate some of the physical constants they need.
8. Students can be made aware of their problem-solving process by having them verbalize what they are doing while solving problems. This can be done in class with the Whimbey-Lochhead pair method wherein one student is the solver while the second one is the recorder. The recorder must avoid solving the problem and leading the solver toward a solution. Once the problem has been completed, the recorder and the problem solver discuss what the problem solver did while solving the problem with the goal being that the solver understands how he/she goes about solving a problem and if and how that process needs to be modified.
9. Problem solving can also be taught using discovery methods like problem based learning, simulation, case study, guided design etc wherein students work on realistic engineering problems.
10. Allow students to work in groups to learn problem solving. This gives students the opportunity to understand difficult problems as each group member will have a different perspective on it and brainstorming during the explore step becomes easy.
11. The goal of problem solving instruction is not to give students the solution but to enable students to find the solution, effectively on their own and to improve their problem solving skill in the process. The role of the instructor is to check that the students’ knowledge base is correct, help them understand the hierarchical structure of knowledge, scaffold them in the choosing of strategies or when they are stuck or when they are clearly headed down an incorrect path. According to guided
design, the entire problem and responses should be developed in advance so that students are guided through the steps of problem solving.

For more instructional studies in engineering problem solving see Appendix G.

Table 12 – Comparison between different engineering problem solving teaching strategies

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Focused on developing skills.</td>
<td>Focused on developing method.</td>
</tr>
<tr>
<td>Identified a set of problem solving skills and target instructional activities to develop these skills. Activities are designed based on Sternberg’s criteria [52].</td>
<td>Use a seven step problem solving strategy to design instruction. This is similar to the ill-structured problem solving strategy as seen in Table 6.</td>
</tr>
<tr>
<td>Students progress from well-defined problem solving skill to open-ended ill-defined problem solving skill in a sequence of four workshops.</td>
<td>Students learn problem solving in their engineering courses and solve problems by systematically applying the problem solving strategy on different types/levels of problems.</td>
</tr>
<tr>
<td>Students learn targeted activities to develop each skill in a domain independent manner; this is followed by bridging activities to learn how to apply these skills in subject domain and everyday life.</td>
<td>Students learn problem solving strategy from worked examples, watching instructor solve problems and solving practice problems in class. Problem solving can also be taught using methods like problem based learning, simulation, case study, guided design, etc.</td>
</tr>
<tr>
<td>Students are required to monitor and reflect on their progress.</td>
<td>Students can be made aware of their problem solving process by doing TAPPS. The goal of problem solving instruction is not to get students to solve a problem but to improve their problem solving ability.</td>
</tr>
</tbody>
</table>

One important aspect for engineering educators to consider is that since engineering is a situated activity [66], we must consider the theories of problem solving and situated cognition which argue for the importance of framing and registration, interactivity and epistemic actions, environmental resources and scaffolds and knowledge-rich problem solving [65]. These aspects must therefore be considered when designing instruction for engineering problem solving.

**Takeaway:** Engineering problem solving can be taught using two approaches as seen in Table 12. Each method has been widely researched and demonstrated improvements in students’ problem solving ability.

5.4 How to teach estimation?

Several authors have recognized the importance of estimation for engineers and the need for developing the estimation skill among student engineers [5], [6], [8], and [9]. In fact Linder [5] describes the differences in the characteristics of the learning activities of (mechanical) engineering and rough estimation/ engineering analysis which demonstrates that the learning that happens in current engineering curricula do not prepare students for rough estimation activities. This is because the learning activities are primarily well-structured in nature while rough estimation as explained previously
are ill-structured problems and as mentioned in [13], the ability to solve well-structured problems does not transfer to the ill-structured estimation problems. Therefore it is important for instruction to be tailored to developing students’ estimation skill explicitly. Below we present some guidelines by authors for activities that can support the learning of rough estimation. It is important to observe that these guidelines are not substantiated by experimental research and mostly suggestions by engineering instructors.

Table 13 - Comparison of estimation problem solving teaching strategies

<table>
<thead>
<tr>
<th>Author</th>
<th>Description of teaching method</th>
<th>Theoretical justification.</th>
</tr>
</thead>
</table>
| Linder (1999)| 1) Teach conceptual knowledge of estimation and estimation problem solving skills.  
2) Increase the number of rough estimation activities done by students.  
3) Include learning activities that have characteristics like those of rough estimation activities. These include engineering analysis, sketching, building, explaining and diagnosing. Basically activities that have students use their knowledge rather than supporting resources, activities where students have to select relevant information and balance different types of information. Students should be exposed to ill-structured problems. | Targeted towards developing students ability in using effective actions and compensation methods identified by the authors. |
2) Illustrate its application with an example from a particular domain.  
3) Repeat with examples from different domains. This helps students understand the essential features of the tool in a domain independent manner.  
4) Provide practice in the usage of a tool in practice problems.  
5) Present more practice problems without                                                                                                                                 | The larger instructional approach followed in the course is similar to Jonassen’s [13] instructional design for the teaching of ill-structured problems. However the teaching of each estimation tool is via Woods et al [15] activity teaching strategy where each activity is an estimation tool. |
| Dunn-Rankin (2001) | **1)** Provide several back-of-the-envelope calculation activities for students to work on individually or in a group.  
**2)** Also include some rhetorical questions that can add to the discussion.  
**3)** Whenever possible, tie the numerical values to everyday physical objects and activities. This helps students develop an intuition for reasonable values for physical quantities. | Practice and experience are important components of estimation. |
| Shakerin (2006) | **1)** Students should be encouraged to practice estimation and be made aware of its importance through short exercises with everyday objects and activities.  
**2)** Estimation activities should be included in courses at all levels, especially the lower division courses where the foundations of engineering are established like units, dimensions and basic engineering concepts. | **1)** The more we ask students to estimate, the better they are at using these skills in future courses and their careers.  
**2)** Units, dimensions and basic engineering concepts are the foundations to perform good estimation and hence students should receive a lot of practice in these. |
| Joram (1998) (Measurement estimation) | **1)** Guess and check or practice with feedback  
**2)** Strategies instruction – reference point strategy | **1)** Informational feedback which helps estimators hone their skill.  
**2)** Reference points may serve as mnemonics for associations between specific magnitudes and their corresponding measurements. |
As mentioned in Section 1, these instructional strategies in Table 13 are not validated via research on their effectiveness. Further their foundation in learning theories and expert-novice analysis is also weak. *So our future research will be focused on designing effective instructional activities for estimation problems based on expert-novice analysis and learning theories.*

6 Conclusions

In this technical report, we reviewed problem solving literature with the goal of identifying suitable teaching-learning strategies for engineering estimation problems. We studied the dimensions of problem solving ability including problem solving strategies, analyzed the differences between expert and novice problem solving behavior and finally identified suitable instructional activities for different kinds of problems. This literature will now guide us as we proceed to design instruction for engineering estimation problems.
7 References


Spiro, R. J., & DeSchryver, M. (2009). Constructivism: When it’s the wrong idea and when it’s the only idea. In meeting of the American Educational Research Association, 2007; An earlier version of this chapter was presented by the first author at a debate on constructivism at the aforementioned conference. Routledge/Taylor & Francis Group.


8 Appendix A – Mahajan’s tools for dealing with complexity

1. Methods of organizing complexity
   a. Divide and Conquer:
      Divide-and-conquer is essentially the decomposing and simplifying problem solving strategy described in section 3.2 and it can be applied irrespective of the problem domain. The key step in this strategy is replacing the quantities that one knows little about with quantities about which one knows more [9]. While we do pay a small price in accuracy because of combining many quantities together, this is offset by the benefit of accessibility of estimates. The hard problem of making a good estimate is converted into a set of simpler sub-problems each of which needs one estimate. Thus the estimate is constructed from the estimates obtained by solving smaller problems and this process can be represented by a tree. The easier the smaller sub-problems are to solve or the more we know about them, the better will be the estimate constructed from them. Hence the division is the most critical step in estimation and the teaching of estimation.

   b. Abstraction:
      Abstraction is the process of creating reusable parts that can be used for solving other problems [9]. Recursion is a special case of abstraction in which the other problem is a version of the original problem. Diagrams are a very useful abstraction because they don’t contain irrelevant details and capture the essential features of the problem.

2. Methods for discarding “fake” complexity
   a. Symmetry and conservation:
      Symmetry simplifies the problem without any cost in accuracy. It is a transformation that preserves the essential features of a problem [9]. The main ideas in this method are firstly, when there is change look for something that doesn’t change (conservation) and secondly, look for operations that leave this quantity unchanged (symmetries) [9].

   b. Proportional reasoning:
      Proportional reasoning is one implementation of symmetry. The idea behind this strategy is to do the analysis using quantities that are unchanged by the specific symmetry operation of taking the ratio of two changing quantities [9].

   c. Dimensions:
      A dimensionless group is a kind of invariant and hence an abstraction [9]. Dimensionless groups can be used to perform dimensional analysis in the following three steps: 1) Find the relevant groups. 2) Form dimensionless groups. 3) Use the groups to make the most general dimensionless statement. 4) Add physical knowledge to narrow the possibilities.
3. **Methods of discarding actual complexity**
   
a. **Easy (special) cases:**
   This tool is based on the principle that correct solution works in all cases including the easy ones [9]. We can use the analysis we perform on the easy cases to help us guess and check solutions. As an example, we can consider the 1D case before the 2D case.

b. **Lumping:**
   The principle behind lumping is to replace a complex, changing process with a simpler, constant process [9]. It is a complementary method to dimensional analysis in that this method removes as much mathematical complexity as possible from the problem in order to focus on the physical reasoning [9]. One lumping technique is to discretize continuous quantities.

c. **Probabilistic reasoning:**
   In this method we recognize that the information that is available to us is incomplete and use probabilities to reflect this incompleteness of knowledge [9]. This is the Bayesian interpretation of probability and is based on two ideas. The first is that probabilities are subjective and reflect our state of uncertainty about a hypothesis. The second is that by collecting evidence our level of uncertainty changes.

d. **Springs:**
   This tool is based on the assumption that every physical process contains a spring [9]. Even though this statement is not strictly correct, it can be applied whenever a physical system satisfies the characteristics of spring motion namely, springs have an equilibrium position and springs oscillate.

e. **Limiting cases:**
   This strategy involves analyzing the situation in limiting cases and connecting them together to make the analysis tractable and comprehensible [9].
## Appendix B – Linder’s Effective actions

<table>
<thead>
<tr>
<th>Effective action/Compensation method</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1. Identify a problem system          | a) Identify a system or systems consistent with the problem formulation.  
  b) Construct a representation of the object.  
  c) Choose a system that is familiar or easy to think about.  
  d) Identify a class of things that is consistent with the problem formulation.  
  e) Construct a schema for how objects function or behave.  
  f) Add information to the representation of the object or system. |
| 2. Identify a quantity with a system. | a) Identify quantities introduced in the process of problem solving with aspects of the system under consideration. |
| 3. Provide a value for a quantity.    | a) Provide a particular value for a quantity.  
  b) Provide a representative value for a quantity like average, minimum or maximum.  
  c) Provide a range of values for quantity.  
  d) Provide a unit for the value.  
  e) Provide a value for classes of objects or things.  
  f) Provide standard values for objects.  
  g) Provide a value for an object in a non-numerical form. |
| 4. Count a set of things              | a) Convert non-numerical knowledge of a number of objects into a numerical value by counting. |
| 5. Compare two objects for a quantity | a) Compare two objects for a quantity of the same type that they have in common.  
  b) Visualize two objects juxtaposed.  
  c) Visualize the smaller object repeated until the larger object is matched in dimension and count the number of repetitions of the smaller object.  
  d) Use absolute numerical values to calculate ratios of the quantities that are difficult to visualize.  
  e) Identify objects with the same orientation to compare with. (e.g. Length of bed vs. height of room)  
  f) Identify objects in the same conceptual scope to compare with. (e.g. Height of room vs. height of person instead of height of car) |
| 6. Identify a relationship between quantities | a) Provide a value for a quantity indirectly by identifying a relationship between it and other quantities.  
  b) Identify and use many kinds of relationships to solve a problem.  
  c) Use a definition to identify a relationship between quantities.  
  d) Identify a geometric relationship between quantities.  
  e) Use physical laws to identify a relationship between quantities.  
  f) Identify a constitutive relationship among quantities.  
  g) Identify and use proportionality relationships between quantities.  
  h) Carry out computation to evaluate the relationship.  
  i) Use computational estimation techniques to evaluate a
2. Change the system scope ("has a" action)
   a) Change the scope of a representation to include more or different information.
   b) Change the scope of a representation such that further actions can be taken.
   c) Consider additional aspects of the object or system.
   d) Consider the object or system as an aspect of a larger system.
   e) Divide spatial characteristics of an object into regular sections and then work with each section.
   f) Identify different conceptual systems that an object can be a part of.
   g) Select a conceptual system for the object to be part of that allows further actions to be taken.

3. Identify a similar system ("is a" action)
   a) Identify another system that is similar to the one under consideration.
   b) Identify a similar system that has only one quantity in common with the original system.
   c) Identify a similar system with two or more quantities in common such that these quantities are in the same relation to each other.
   d) Relate the results of actions taken on the similar system back to the original system.

4. External representation
   Writing, sketching to compensate for limited working memory and or visualization skills

5. Guessing
   Guess a value or relationship to compensate for unknown information or inability to recall. Then accuracy of estimate limited by accuracy of guess. Guess may be revised if estimate seems unreasonable.

6. Brainstorming
   To compensate for being stuck, ask self questions to identify information needed to proceed

7. Providing a range of values
   To compensate for inability to provide a single value

8. Considering the consequence of an action
   To compensate for not realizing that their initial actions would require subsequent actions that they would not be able to complete

9. Using units and dimensional analysis
   To compensate for not recalling a relationship accurately
## 10 Appendix C – Comparison of measurement estimation strategies

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental measurement</td>
<td>Mental instrument application</td>
<td>Mentally line up measurement instrument with object to be measured.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>Unit iteration.</td>
<td>Mentally divide object to be measured into units, count number of units and report sum as result.</td>
<td>None.</td>
</tr>
<tr>
<td>Transform the unit</td>
<td>Recall reference point</td>
<td>Recall an object equivalent to a standard unit, then apply another strategy such as unit iteration with the object substituting for standard unit.</td>
<td>Abstraction/recursion</td>
</tr>
<tr>
<td></td>
<td>Prior knowledge</td>
<td>Use prior knowledge about part of object to be estimated.</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>Comparison</td>
<td>Recalls reference point, whose measurement is known, that is equivalent in magnitude to object to be estimated.</td>
<td>Symmetry and conservation</td>
</tr>
<tr>
<td>Transform the object</td>
<td>Decomposition/recomposition</td>
<td>Prior to estimating, subdivide object to be estimated into smaller continuous parts. The estimate for each smaller part is obtained using one of processes above. These estimates are summed or multiplied to get the estimate of the larger object.</td>
<td>Divide and conquer</td>
</tr>
<tr>
<td></td>
<td>Use of subdivision cues</td>
<td>Use physical aspects of object to be estimated to mentally subdivide the object.</td>
<td>Divide and conquer</td>
</tr>
<tr>
<td></td>
<td>Rearrangement</td>
<td>When subdivision is difficult, estimators mentally rearrange part of the object to be estimated prior to sub-dividing.</td>
<td>Symmetry and conservation</td>
</tr>
<tr>
<td>Other</td>
<td>Squeezing</td>
<td>By making estimates that are a little more or a little less than the true measurement, the estimators get progressively closer to the true measurement.</td>
<td>Limiting cases.</td>
</tr>
</tbody>
</table>
11 Appendix D – Quality criteria for engineering problem solving (adapted from [46])

<table>
<thead>
<tr>
<th>Trait/Dimensions</th>
<th>Expert behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Understanding and Communication</strong></td>
<td></td>
</tr>
<tr>
<td>1 Schema – deep understanding</td>
<td>Engineers organize their knowledge using structures (schema) in which the various pieces of knowledge are linked to one another (think of a spider web). Schemas are hierarchical—at the top of the pyramid are broad disciplinary concepts, which link to sub concepts, which link to facts and details.</td>
</tr>
<tr>
<td>2 Language of the discipline</td>
<td>When communicating to others, engineers apply the concepts and ideas of our discipline. These concepts and ideas are communicated in a way that is valid and accurate and in a way that reflects an underlying schema.</td>
</tr>
<tr>
<td>3 Effective for audience</td>
<td>Engineers match the communication to the audience.</td>
</tr>
<tr>
<td>4 Verbal, visual and math representations</td>
<td>To enhance the efficiency and effectiveness of communication, engineers blend three types of communication (visual, verbal and mathematical)</td>
</tr>
<tr>
<td>5 Effective process of documentation</td>
<td>Engineers document their work on paper as they proceed (they avoid backfilling). Documentation is organized and a peer can quickly skim and understand the work. Documentation is complete, meaning it is appropriately annotated and labeled, and presents essential details. Documentation is archival, meaning it can be located and used long after the project or task is completed. Documentation is cost effective, meaning the engineer does not spend unnecessary time creating the documentation.</td>
</tr>
<tr>
<td><strong>Scientific Concepts and ongoing learning</strong></td>
<td></td>
</tr>
<tr>
<td>6 Scientific concepts</td>
<td>Engineers interpret the world using scientific concepts such as Ohm's law, equilibrium, and the ideal gas law. When faced with an unfamiliar problem or situation, engineers use scientific concepts to guide their actions—that is they use concepts to create understanding, to make predictions, to make decisions, to solve problems and to perform other similar actions.</td>
</tr>
<tr>
<td>7 Scientific Honesty</td>
<td>Engineers &quot;bend over backwards&quot; to convince themselves that their work accurately reflects phenomena in the real world. That is, engineers seek to discover what is true by carefully checking their results against real-world observations and well-establish scientific laws. Engineers report their results honestly, and they reveal relevant concerns and uncertainties. Engineers do not manipulate their results in order to prove that a certain point-of-view is valid.</td>
</tr>
<tr>
<td>8 Assumptions</td>
<td>Engineers continually assess the validity of the underlying assumptions. Engineers communicate and document major assumptions.</td>
</tr>
<tr>
<td>9 Self-assessment</td>
<td>Engineers assess their knowledge by continually questioning and probing their own understandings and conceptualizations.</td>
</tr>
<tr>
<td>10 Ongoing learning</td>
<td>Engineers learn in an ongoing and continuous process that is embedded in problem solving. Engineers assume responsibility for their learning and understanding. When understanding is in question, the engineer takes appropriate action such as asking a</td>
</tr>
<tr>
<td></td>
<td>Question or Self-Study.</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------</td>
</tr>
<tr>
<td>11</td>
<td>Strong Learning Process</td>
</tr>
<tr>
<td></td>
<td>Engineers use effective means to learn. Engineers relate newly-learned knowledge to what they already know and to the physical world. Engineers organize their knowledge into schema. Engineers continually test and modify their schema.</td>
</tr>
</tbody>
</table>

**Engineering Process**

| 12 | Recognizing Opportunities |
|    | Engineers recognize opportunities to apply analysis based on math and science to enhance quality or reduce cost. Engineers avoid applying analysis when it is inappropriate, unnecessary or too expensive. |

| 13 | Cost Effective |
|    | Engineers match the precision and accuracy of an analysis to the needs of the task and the quality of the data available. Simple models (e.g. lumped parameter, algebraic equations) are used for rough estimates. Complex models (e.g. ordinary and partial differential equations) are used when accuracy is attainable and when the needs justify the cost. |

| 14 | Metacognition |
|    | Engineers have self-knowledge, awareness and control as they problem solve. Engineers can describe in words and pictures how they think and monitor their thinking. |

| 15 | External Representation |
|    | Engineers sketch diagrams and put information on paper as a means to reduce the demands on short-term memory. |

| 16 | Framework (schema) for Procedural Knowledge |
|    | Engineers have a basic process (schema) that they apply to each new problem. Characteristics of this process are: 
   a) Personal. Each engineer has adapted the process to fit their unique style and they have ownership. 
   b) Robust. The same basic process works for many different types of problems. 
   c) Flexible. The process is continually adapted to fit each new problem. The process is continuously improved. 
   d) Active. Engineers are active: talking, questioning, sketching, etc. Engineers avoid “sitting and thinking.” 
   e) Effective. The process usually guides the engineer to a solution. The process is time efficient. 

   Stages of this process are:
   a) Problem Formulation. This is the process of interpreting the problem, creating a representation, and creating a specific goal.
   b) Strategy and Planning. This is the process of considering alternative ways to solving the problem, identifying a potential solution path and creating a simple plan.
   c) Action. This is carrying out the calculations and other actions associated with a solution. 
   d) Review. This is reviewing the process of solution and the solution itself. |

**Problem Formulation**

| 17 | How Things Work? |
|    | Engineers create representations that are founded on knowledge of how things work in the physical world. When this knowledge is lacking, engineers locate and learn the information. |

| 18 | Representation Using Scientific Concepts |
|    | Engineers create representations that are founded on scientific concepts. Engineers simplify real-world problems by retaining factors that strongly influence the physics and eliminating extraneous details that have small effects on accuracy. |

| 19 | Representation of a Text Problem |
|    | Engineers create their own interpretation of a text problem. This interpretation is complete (considers all the given information) and valid. Engineers see text problems as they might exist in the
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Concrete goal</td>
<td>Engineers create a well-defined goal. This goal satisfies criteria such as specific, strategic (i.e. useful to the overall project), unambiguous, meaningful, and accurate.</td>
</tr>
<tr>
<td>21</td>
<td>Finding information</td>
<td>Engineers locate relevant technical information such as material property data, standards data, and empirical parameters. Engineers assess this information, and make appropriate and valid decisions considering factors such as accuracy and applicability.</td>
</tr>
<tr>
<td>22</td>
<td>Units, calculation and concept of estimation</td>
<td>Units are carried and canceled on each calculation, and this process is well organized and documented. When practical, numbers and symbols are assigned units. The engineer knows the meaning and relative size of each unit.</td>
</tr>
<tr>
<td>23</td>
<td>Calculations</td>
<td>Calculations follow the rules of effective documentation (organized, complete, simple, archival).</td>
</tr>
<tr>
<td>24</td>
<td>Concept of estimation</td>
<td>Engineers make estimates that balance accuracy requirements with cost. Engineers seek an accuracy that is acceptable for the task—that is, they do not make a complex analysis when a simple analysis will suffice. Engineers can make valid estimates of the uncertainty of a calculation. Engineers can make quick estimates of complex problems.</td>
</tr>
<tr>
<td>25</td>
<td>Accuracy</td>
<td>Completed work is free of errors. Throughout the solution process, decisions, communication and other factors are based on valid interpretations of engineering knowledge.</td>
</tr>
<tr>
<td>26</td>
<td>Error checking</td>
<td>When proceeding through a problem, engineers routinely double-check most steps of their work.</td>
</tr>
<tr>
<td>27</td>
<td>Reviewing a solution</td>
<td>After completing a calculation, engineers use a simple method such as a quick estimate to evaluate the solution. Engineers review their results and make inferences about the problem as it exists or might exist in the physical world.</td>
</tr>
<tr>
<td>28</td>
<td>Troubleshooting</td>
<td>Since getting stuck is a natural part of problem solving, engineers develop methods for troubleshooting that work in most cases. Once stuck, engineers systematically apply their methods, proceeding in a calm and deliberate fashion.</td>
</tr>
<tr>
<td>29</td>
<td>Kaizen (continuous improvement)</td>
<td>To improve performance, engineers purposefully review. This review focuses on identifying strengths (practices that are effective for problem solving) and deltas (changes that bring about improvements in the future). To take advantage of the process of social learning, engineers involve others in these reviews.</td>
</tr>
<tr>
<td>30</td>
<td>Balance</td>
<td>Engineers balance persistence and perspective; they work hard, but they will take appropriate action if the effort to solve the problem becomes too much. By appropriate action, we mean performances such as asking for help, letting it go, or redefining the problem so that it is simpler.</td>
</tr>
</tbody>
</table>
### 12 Appendix E – Comparison of instructional studies in measurement estimation (adapted from [9])

<table>
<thead>
<tr>
<th>Name of study</th>
<th>Participants</th>
<th>Attributes estimated</th>
<th>Scoring</th>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassell (1941); Wilson and Cassell (1953)</td>
<td>6573 2nd through 10th grade student</td>
<td>Height, width, distance</td>
<td>Correct and reasonable estimates specified for each item; Percentage error acceptable for correct estimates varied by object</td>
<td>18% students on average at each grade level gave correct responses. Accuracy increased with grade level. Adults gave 19-56% correct responses.</td>
</tr>
<tr>
<td>Corle (1960)</td>
<td>147 5th and 6th grade students</td>
<td>Length, circumference</td>
<td>Percent error</td>
<td>Percent error was 96% on average for 5th graders and 56% on average for 6th graders (excluding outliers)</td>
</tr>
<tr>
<td>Corle (1963)</td>
<td>96 college juniors 368 elementary school teachers</td>
<td>Length, circumference</td>
<td>Percent error</td>
<td>Average percent error was 21-53% No differences between undergraduates and teachers.</td>
</tr>
<tr>
<td>Forrester, Latham and Shire (1990)</td>
<td>70 5-8 year olds</td>
<td>Length</td>
<td>Analyses performed on log transformed scores Classified methods used to make estimates</td>
<td>Children underestimated on length. Accuracy increased with age.</td>
</tr>
<tr>
<td>Forrester and Shire (1994)</td>
<td>67 8-11 year old children</td>
<td>Length</td>
<td>Percent error scores. Classified methods used to make estimates</td>
<td>Average percent error was 10% for 8-9 year olds and 7% for 10-11 year olds Younger children underestimated more often than older children, but only on relatively long items</td>
</tr>
<tr>
<td>Study</td>
<td>Sample Size</td>
<td>Measurements</td>
<td>Methodology</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Siegel et al (1982)</td>
<td>140 students in grade 2 through 8 and 10 adults</td>
<td>Length, height</td>
<td>Estimates classified as reasonable (&lt; 100% error) and accurate (&lt;= 50% error) Classified responses into strategies Objects requiring a single unit were more accurately estimated than those requiring fractional or multiple units Least accurate estimates were made for objects requiring mental decomposition before estimating Accuracy increased with age for all but those items lending themselves to a regular decomposition strategy.</td>
<td></td>
</tr>
<tr>
<td>Wilson (1936)</td>
<td>2056 3rd through 12th grade students</td>
<td>Height, width</td>
<td>Responses classified as “fair estimates” or “wrong” (criteria not given). Example was fair estimates ranged from 28-34 inches for a 30 inch object. Reasonable estimates increased with grade from an average of 3% students in 3rd grade to an average of 31% students in 12th grade.</td>
<td></td>
</tr>
<tr>
<td>Author (s)</td>
<td>Participants</td>
<td>Intervention strategy</td>
<td>Assessment</td>
<td>Outcomes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>------------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Heller &amp; Reif [54]</td>
<td>24 undergrad students in a second introductory physics course</td>
<td>Two different prescriptive models consisting of a three steps each for generating initial descriptions to mechanics problems corresponding to two experimental groups. One control group. Teacher provides detailed instructions on how to execute steps of the model along with question prompts at relevant junctures.</td>
<td>Student solutions graded with respect to four performance measures including overall correctness of solution.</td>
<td>Students following one particular model outperformed control group and the group following the other model.</td>
</tr>
<tr>
<td>Gaigher, Rogan &amp; Brown (from [11])</td>
<td>N/A</td>
<td>Structured strategy based on several previous frameworks with seven steps</td>
<td>N/A</td>
<td>Authors argue that the results of the study “found that students who had been exposed to the structured problem-solving strategy demonstrated better conceptual understanding of physics and tended to adopt a conceptual approach to problem solving.”</td>
</tr>
<tr>
<td>Ogilvie (from [11])</td>
<td>N/A</td>
<td>Problem solving instruction using context-rich problems</td>
<td>N/A</td>
<td>Only small decrease in frequency of students using plug and chug methods. But increase in student use of diagrams and thinking about concepts before looking for an equation.</td>
</tr>
<tr>
<td>Chi, Bassock,</td>
<td>N/A</td>
<td>Learning from worked examples.</td>
<td>N/A</td>
<td>Stronger students spent time on a worked</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Group</td>
<td>Metacognitive Question Prompts</td>
<td>Classroom Observation of Student Behavior</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>-------------------------------</td>
<td>------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>et. al.; Ferguson – Hessler &amp; de Jong (from [11])</td>
<td></td>
<td>Example trying to understand aspects of the example that they did not get initially. When solving problems they went back to the example only in search of a specific idea. Weaker students took everything for granted and spent very little time in deeper understanding. When solving problems they try to use the worked example as a template.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schoenfeld (from [10])</td>
<td>College math course students</td>
<td>Metacognitive question prompts during group problem solving. Instructor circulates through the room and may ask the students one of three questions at any time: a) What (exactly) are you doing? (Can you describe it precisely?) b) Why are you doing it? (How does it fit into the solution?) c) How does it help you? (What will you do with the outcome when you obtain it?). Students must prepare in advance by thinking of the answers to these questions.</td>
<td>Classroom observation of student behavior.</td>
<td>By the end of the course students become habituated to thinking about and answering these questions.</td>
</tr>
<tr>
<td>Lester, Garofalo &amp; Kroll (from [10])</td>
<td>One regular and one advanced seventh grade</td>
<td>To foster students metacognitive development. Total instruction time focusing on metacognition in the experiment was 16.1 written tests, student interviews, observations of individual and pair problem-solving sessions,</td>
<td>&quot;Developing self-regulatory skills in complex subject-matter domains is difficult. It often involves 'behavior modification', unlearning</td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td>Hours over 12 weeks - average 35.7% of the mathematics classroom time during the instructional period. The teacher (a) serve as external monitor during problem solving, (b) encourage discussion of behaviors considered important for the internalization of metacognitive skills, and (c) model good executive behavior.</td>
<td>Videotapes of the classroom instruction.</td>
<td>Inappropriate control behaviors developed through prior instruction.</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schoenfeld 1980 [38]</td>
<td>Seven upper-division science majors</td>
<td>Explicit use of the strategies was the difference between the instruction for the two groups. Strategies included drawing diagrams, look for an inductive argument, consider a logical alternative (contradiction/contra positive), consider a similar problem with fewer variables and try to establish sub goals. Expert model of the problem solving process designed consisting of 5 steps</td>
<td>Pre and post tests.</td>
<td>Learning gains were obtained in pre and post tests of the course.</td>
</tr>
<tr>
<td>Heller 92 [39]</td>
<td>Student s of an introductory physics course</td>
<td>Problem solving strategy based on research in expert behavior in physics problem solving. The problems constructed for practice and assessment were context rich problems which have the</td>
<td>Students were assessed using a reliable and valid measurement scale for problem solving performance and a rating scale for problem difficulty and it was found</td>
<td>The instructional approach improved the problem solving ability of students at all levels.</td>
</tr>
</tbody>
</table>
characteristics that goal variable was not explicit, assumptions need to be made, more information than necessary is provided in the problem statement, some information is missing that can be estimated or is common knowledge. Problem solving strategy has 5 steps: visualize the problem, physics description, plan, execute and evaluate. Students solved these problems in recitation and lab sessions in carefully managed co-operative groups where they practiced the prescribed strategy. that better problem solutions emerged by students working in groups than individuals solving problems alone.
## 14 Appendix G – Instructional Studies in Ill structured and engineering problem solving.

<table>
<thead>
<tr>
<th>Study</th>
<th>Participants</th>
<th>Intervention strategy</th>
<th>Assessment</th>
<th>Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shekoyan 2009 [17]</td>
<td>Introductory level physics course students</td>
<td>Using MPPs in cooperative group solving activities in the recitation and homework. Instructors scaffold student work by prompting students to ask epistemic questions during the problem solving steps.</td>
<td>The students' solutions to MPPs were graded based on the rubric. The rubric and the guidelines represented the elements of cognitive apprenticeship.</td>
<td>Positive trend in epistemic cognition enhancement. Students constructed a better conceptual understanding of physics. Not only did students engage in epistemic cognition, but also it did not depend on whether the students were high-achieving or low-achieving.</td>
</tr>
<tr>
<td>Ge 2001 [42]</td>
<td>115 college students participated in the experimental study, and 19 of them participated in the comparative, multiple-case study.</td>
<td>A mixed study design, combining an experimental study with a comparative, multiple-case study, was applied. The experimental study was conducted to measure the students' problem-solving outcomes on an ill-structured task, as demonstrated by the four problem-solving processes, in four different treatment conditions: individuals with question prompts (IQ), individuals without question prompts (IC), peers with question prompts (PQ), and peers without question prompts (PC). The comparative, multiple-case study, through observation, interviews, and think-aloud protocols, was carried out to gain insights into students' problem-solving processes, especially their cognition and metacognition, as influenced by question prompts or peer interactions.</td>
<td>Rubrics for rating problem solving report.</td>
<td>The study confirmed the findings of previous research on the effectiveness of question prompts in facilitating students' cognition and metacognition and also showed the benefits of peer interactions, which were contingent upon group members' active and productive engagement in peer interactions, that is, questioning, explaining, elaborating and providing feedback among peers.</td>
</tr>
<tr>
<td>Reference</td>
<td>Participants</td>
<td>Study Design</td>
<td>Problem-Solving Processes</td>
<td>Scoring Rubric</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>--------------</td>
<td>----------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Bixler 2007 [43]</td>
<td>Seventy-nine college students participated in the study; 40 in the control group and 39 in the experimental group.</td>
<td>A mixed study design was used, integrating quantitative with qualitative methods. The quantitative method empirically measured individual’s problem-solving outcomes on an ill-structured task. Four ill-structured problem-solving processes were measured: problem representation, developing solutions, making justifications, and monitoring and evaluation. The qualitative method investigated contextual information about the individual’s problem-solving process, such as how it affected their motivation and problem representation, solution development, making justifications, and monitoring and evaluation of solutions. This was accomplished via observation, interviews, and think-aloud protocols.</td>
<td>Scoring rubric for measuring ill-structured problem solving process</td>
<td>The results of the experimental study showed that students working with question prompts significantly outperformed students without question prompts in all four problem-solving processes. While no students in the experimental group actually claimed to be more motivated by prompts, question prompts had a positive role in subjects' motivation as evidenced by reduction of frustration and stress, an increase in self-efficacy, an increase in strategic behavior, and by providing a fail-safe environment for learning.</td>
</tr>
<tr>
<td>Raviv, Morris &amp; Ginsberg 2005 [55]</td>
<td>College and high school students</td>
<td>Teaching systematic ways of thinking. Students are exposed to TRIZ, Lateral Thinking, Mind Mapping and the Eight Dimensional Methodology for Innovative Thinking. Multiple methods are taught since some are not necessarily suited for everyone or for every problem.</td>
<td>Number of ideas generated by students counted and compared with pre-intervention levels.</td>
<td>On average, students generate more than twice as many solutions after completing the course. The quality of the answers were not considered in this study. Although we can be certain that students create more solutions in general, what causes this change is only speculated.</td>
</tr>
<tr>
<td>Stojcevski 2008 [55]</td>
<td>First semester of the undergraduates electrical</td>
<td>A framework for teaching students skills for solving design problems was developed. The reason for selecting this scheme of</td>
<td>A Pre and Post questionnaire was used. The pre-questionnaire</td>
<td>The student responses indicated that though many students have a tendency towards using deep learning</td>
</tr>
</tbody>
</table>
problem solving is that the stages directly relate to the cognitive process required in problem solving and the scheme is general enough so that it is applicable in most contexts of design problems. The framework has the following elements: 1. Presenting the problem to the learners 2. Supporting learners in representing the problem 3. Supporting learners in finding solution 4. Reflection in learning of problem solving 5. Assessing student learning in problem solving. The experimental study was done via a Problem/Project Based Learning (PBL) course. A library of cases and troubleshooting examples was available to the students. was used to establish the baseline for the design of the material for the training classes. The post questionnaire was used to evaluate the efficacy of the course.

approaches, most use surface approach to learning. Student responses also indicate that they learn best if the learning is contextual, which could help them to relate what they are learning with their existing sense of reality. They also seem to support social setups for learning and being able to learn something within a variety of contexts. Responses also indicate that concrete and tangible knowledge is important for the engineering students. Students in general seem not been able to properly identify and characterize types of problems.
## 15 Appendix H – Comparison of features of teaching-learning strategies using problem-solving as an activity (adapted from [61])

<table>
<thead>
<tr>
<th>TL strategy</th>
<th>Problem and its role</th>
<th>Process</th>
<th>Scaffolds</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBL [61]</td>
<td>Realistic ill-structured problem provides focus for conceptual learning and problem solving strategies.</td>
<td>Identify facts of the problem, generate ideas, identify content to be learnt, student directed learning, revisit, evaluate, reflect</td>
<td>Facilitator for learning who models reasoning process. Collaboration enables idea negotiation and knowledge sharing</td>
</tr>
<tr>
<td>CC [57]</td>
<td>A well-structured problem and its solution by multiple methods so students can identify similarities and differences between solution methods.</td>
<td>Side by side comparison of multiple solutions to problems, labeling of certain steps, generate explanations, do practice problems.</td>
<td>Labeled multiple solutions (worked examples). Pair work to leverage benefits of collaboration. Prompts for students to generate explanations while studying worked examples.</td>
</tr>
<tr>
<td>PF [58], [63]</td>
<td>Complex problem provides focus for generation (or invention) phase and student exploration, cause students to struggle with core concepts and procedures.</td>
<td>Generate representation and solution methods for complex problems, given canonical solution and direct instruction (consolidation or instruction phase), practice</td>
<td>Collaboration, affective support for persistence, metacognitive scaffolds, role-playing scripts, contrasting cases, representational scaffolding</td>
</tr>
<tr>
<td>IPL [62]</td>
<td>Complex problem with contrasting cases for which students have to “invent a solution”, helps students identify distinctive features and discern the underlying structure</td>
<td>Invention activities, classroom discussion, direct instruction and practice</td>
<td>Collaboration, Teacher questions regarding student inventions while they are working.</td>
</tr>
<tr>
<td>MEA [60]</td>
<td>A thought-revealing, model-eliciting, open-ended, realistic, client-driven problem for resolution provides focus for students to construct a mathematical model/system that addresses the needs of the client</td>
<td>Integrate learning from previous courses with new information; Reinforce the concepts that are currently being covered; and Discover a concept that has yet to be formally introduced. Document model, do self-evaluation of work.</td>
<td>Collaboration, Guidance by instructor, reflection tools</td>
</tr>
<tr>
<td>GBL [56], [59]</td>
<td>A scenario in an educational game provides the focus for student learning of different types of problem solving</td>
<td>Generate solutions using tools in the game, multiple iterations, reflection and evaluation, decision-making</td>
<td>All tools available in the game to develop solutions, ordered problems, repetition, just-in-time and on-demand information, situated meanings</td>
</tr>
<tr>
<td>AI [64]</td>
<td>Video-based narrative ending with complex problem which provides a shared experience for students to understand how knowledge can support problem-solving.</td>
<td>Guided planning and subgoal generation.</td>
<td>Teacher engages students prior knowledge, models problem solving strategies, provides content instruction as needed. Collaboration enables negotiation of ideas and sharing of strategies.</td>
</tr>
</tbody>
</table>