2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #52538

# BYOE: Design Rationale behind a Circuit Troubleshooting Laboratory Experiment Through the Lenses of Problem Complexity, Ill-Structuredness, Knowledge Types, and Troubleshooting Process

**Mr. Christopher Lowell Romeo, University at Buffalo, The State University of New York**

Christopher Romeo is currently a PhD candidate in the Engineering Education Department at SUNY University at Buffalo, where he is working under the advisement of Dr. Andrew Olewnik. His general research interest areas include problem typologies, problem-based learning, student problem-solving processes, pedagogy improvement, and design of learning experiences. He joined the ASEE as a graduate student in 2024.

**Dr. Andrew Olewnik, University at Buffalo, The State University of New York**

Andrew Olewnik is an Assistant Professor in the Department of Engineering Education at the University at Buffalo. His research includes undergraduate engineering education with focus on engineering design, problem-based learning, co-curricular involvement and its impact on professional formation, and the role of reflection practices in supporting engineering undergraduates as they transition from student to professional.

**BYOE: DESIGN RATIONALE BEHIND A CIRCUIT TROUBLESHOOTING LABORATORY EXPERIMENT**

Through the Lenses of Problem Complexity, Ill-Structuredness, Knowledge Types, and Troubleshooting Process

# Abstract

BYOE: Troubleshooting is a vital problem-solving skill for practicing engineers, so it is imperative that engineering students learn how to troubleshoot effectively. However, troubleshooting is rarely explicitly taught to engineering students, and traditional undergraduate engineering classes and laboratories seldom provide students with dedicated opportunities to practice doing troubleshooting. There is a need to develop tailored problems and exercises that can be used to teach students how to troubleshoot and allow them to intentionally practice and develop their troubleshooting skills.

To address this need, we created a circuit troubleshooting problem that is meant to be used as a laboratory experiment in an undergraduate electrical engineering class. This paper presents the design rationale behind that problem through the theoretical lenses of complexity, ill-structuredness, knowledge type application, and the troubleshooting process. We explain how we designed this problem to leverage complexity and ill-structuredness in very intentional ways, and we walk through the process of solving this problem using the expert troubleshooting process. We also explain the unique ways in which different knowledge types (conceptual, structural, procedural, domain/experiential, and strategic) are necessary to apply during that process to make progress towards a solution to the problem.

This problem serves as an example of a troubleshooting problem that electronics instructors can integrate into their existing circuits classes as a laboratory experiment. It can be used to explicitly teach students troubleshooting strategies and procedures. Besides this, we also argue that troubleshooting problems in general can be a particularly effective pedagogical tool for students to learn specific technical concepts and skills, because they provide a meaningful context for students to understand when and how to apply them. We offer some concluding remarks for engineering instructors regarding how they could design their own troubleshooting problems by purposefully manipulating the same theoretical aspects we used to create ours. We propose using these theoretical dimensions to reflect on the design of class exercises (troubleshooting or otherwise) because they can be calibrated in different ways to intentionally require students to critically think about, and judiciously apply, desired concepts, skills, tools, and procedures.

*Keywords: troubleshooting, problem design, circuits, complexity, ill-structuredness, types of knowledge, troubleshooting process*

# 1. Introduction and Purpose

Troubleshooting is a vital problem-solving skill for practicing engineers, and the ability to effectively troubleshoot problems is a crucial part of good engineering practice [1], [2]. Capable engineers must be able to identify, diagnose, and solve issues that naturally arise within existing devices, structures, processes, and systems, so it is essential that professional engineers be proficient in this skill. Moreover, troubleshooting problems are one of the most common types of problems encountered by working engineers in all disciplines [3]. Troubleshooting problems inevitably arise across many engineering pursuits, whether intentionally troubleshooting an explicit malfunction or addressing an unforeseen issue that spontaneously emerges within the context of a broader technical endeavor (like design, prototyping, etc...). Since fixing problems is a core competency for practice [4], it is imperative that engineering students learn how to troubleshoot effectively.

However, engineering students are rarely explicitly taught how to troubleshoot. Traditional undergraduate engineering classes and laboratories seldom provide students with dedicated opportunities to practice doing troubleshooting. Though troubleshooting scenarios inevitably arise during lab experiments or class projects, they are not usually leveraged as intentional learning opportunities for students, nor are students explicitly taught troubleshooting strategies or procedures in these contexts [5], [6], [7]. Engineering instructors even acknowledge that troubleshooting skills are vital to student success in their courses; yet, in many cases, these same instructors don't provide students with troubleshooting exercises to hone their skills, nor do they directly assess students' troubleshooting abilities [8]. Instead, students are just expected to somehow fix problems that emerge during a typical lab experiment or class project as an implied part of the activity, without being taught how to do so. And if they can't troubleshoot the issue successfully, then that can prevent them from succeeding with the intended objectives of the activity and achieving desired learning outcomes.

So, there is a need to develop tailored problems and exercises that can be used to teach students how to troubleshoot and allow them to **intentionally** practice and develop this important skill, instead of just relying on exposure to inevitable errors during learning activities as teaching tools. To address this need, we created one such circuit troubleshooting problem that can be used as a laboratory experiment in an intermediate-level undergraduate electronics class. This paper presents that problem and the design rationale behind it, explained through the theoretical lenses of complexity, ill-structuredness, knowledge type application, and the troubleshooting process.

First, we describe the circuit itself and how it is intended to operate. Then, we explain theoretical aspects of problems and discuss how they apply to the specific problem we created, which we designed to leverage those aspects in very intentional ways. We explain how to make the circuit a troubleshooting problem, why it is complex and ill-structured, and what students must do to contend with those aspects of the problem while trying to solve it. Then, we walk through the process of solving this problem using the theoretical troubleshooting process (which is representative of expert approaches) as a framework to illustrate that procedure. We also explain the unique ways in which different knowledge types (conceptual, structural, procedural, domain/experiential, and strategic) are necessary to apply during that process – i.e., what they are, and how they must be operationalized in this specific context to make meaningful progress while engaging this specific circuit problem.

# 2. The Circuit Problem

The circuit we designed for this troubleshooting lab experiment is shown in Fig 1, and it was adapted from the one featured in [5] in their telephone troubleshooting exercise. It functions as a nightlight, using a photoresistor and an op-amp configured as a non-inverting amplifier to control the brightness of an LED at the output. Depending on the light level in the room, the LED will turn on, and its brightness will increase or decrease accordingly.

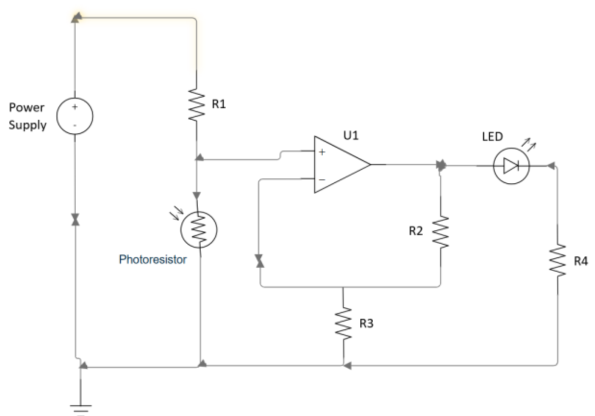


***Figure 1.*** *Nightlight circuit schematic. The circuit uses a photoresistor and an op-amp configured as a non-inverting amplifier to control the brightness of an LED.*

The circuit uses a basic resistor voltage divider network, comprised of R1 and the photoresistor, to adjust the variable voltage at the non-inverting input of the operational amplifier. As the amount of light hitting the photoresistor increases, its resistance decreases, and so the voltage at the input of the op-amp decreases, and vice versa. The op-amp is configured as a non-inverting amplifier, which scales the voltage seen at the input by a certain amount and sends it to the output. The gain of the op-amp is fixed, determined by the values of the two resistors (R2 and R3) in the feedback loop connecting its output back to its inverting input. The LED is powered by the amplified voltage output from the op-amp, and resistor R4 is used to limit the current flowing through it.

So, as more or less light is allowed to hit the photoresistor, a varying voltage is passed through the op-amp, scaled by the gain, and sent to the LED, which determines the LED brightness (or whether it is on or off altogether). Ideally, the circuit is calibrated such that the LED remains off in normal room lighting conditions; enough light hits the photoresistor so that the voltage output by the op-amp is too low to turn on the LED. Then, when the room becomes dark enough (or when the photoresistor is sufficiently covered), the voltage passed to the LED increases, eventually becoming high enough to turn the light on and illuminate the surroundings. Further darkening of the room will increase the relative brightness of the LED proportionally. The circuit can be calibrated to behave this way by making several adjustments:

1. Change the value of R1 to adjust the range of possible input voltages
2. Adjust the gain of the op-amp by changing the values of R2 and R3
3. Change the value of R4 to limit the current flowing through the LED more or less

The main challenge for students in this problem is to fine-tune the circuit so that it operates in this intended way, but there are a few different ways to achieve that, so the problem is ambiguous. If the circuit is misbehaving, you have different options for components that you can modify, or ways you can redesign the circuit, to fix it, depending on exactly how the circuit is malfunctioning. Now, regardless of the specific issue, the high-level goal of solving this troubleshooting problem is still the same – i.e., to achieve a circuit that functions as a nightlight, over a properly-calibrated spectrum of brightness that is proportional to the light level in the room. However, due to the ambiguity of the problem, and the unknown nature of the specific issue in the circuit, solutions are not immediately obvious, and so solving the problem is not trivial. Moreover, there are multiple solutions you could implement, each of which will have pros and cons, and certain solutions may be more efficient, effective, complete, or appropriate than others for addressing a specific issue.

Because of the ambiguity of the problem and the negotiability of possible solutions, there are a few other pitfalls for students to avoid as well. Even if the circuit is technically working, you don't want the LED to be too dim compared to how dark the room is. Conversely, you don't want the LED to immediately turn on and stay fixed at maximum brightness without varying, regardless of how the light level in the room changes (the op-amp output saturates to the supply voltage in this case). Similarly, you also don't want the voltage output from the op-amp to become exceedingly high (assuming the supply voltage even allows this). If it does, it will push too much current through the LED and cause it to burn out, which you do not want to be possible at the high end of circuit operation. In any of these cases, the circuit does not operate properly as a nightlight, and so none of these solutions would be considered very effective. So, students are tasked with finding *optimal* circuit parameters that will achieve the desired functionality beyond just a minimal working version of the circuit, while simultaneously avoiding these pitfalls.

# 3. Theoretical Frameworks for Problem Design

## 3.1 Making This Circuit a Troubleshooting Problem

Generally, the goal of solving a troubleshooting problem is to identify and eliminate a fault within some system, and to implement a solution that will repair the system and return it to its normal operating behavior [9]. To turn this circuit into a troubleshooting problem, you can deliberately insert a fault somewhere that will disrupt its intended behavior as a nightlight. There are a few (non-trivial) options for faults to choose from for this particular circuit.

For example, you can choose values for R2 and R3 such that the gain of the op-amp is too low or too high. If the gain is too low, the output voltage will be too low, and the LED will always stay off regardless of the light level in the room, or it will just be too dim compared to how dark the room is. If the gain is too high, the output voltage will be too high (saturated to supply voltage), and the LED will always stay on and fixed at maximum brightness. It will not vary in brightness regardless of how the light level in the room changes. In the extreme case, the LED will just burn out. Another fault option is to choose the value of R1 to be significantly larger or smaller than the resistance range of the photoresistor. This would cause the range of possible input voltages to the op-amp to be entirely too low or high, respectively, for correct nightlight operation, regardless of how the light level in the room changes. A third fault option is to choose the value of R4 to be too low or too high. In the former case, it will allow too much current to flow through the LED, and the light will hit maximum brightness too quickly, or even burn out completely; in the latter case, it will limit the LED current too much, and the light won't be able to become bright enough regardless of how dark the room becomes.

As mentioned in the previous section, depending on the specific fault you choose to put in this circuit, it can be troubleshot in different ways, and there are different appropriate solutions to fix it and restore it to correct operation. To illustrate and discuss different aspects of troubleshooting this circuit, for the rest of this paper, we will select the case where the feedback resistor values are such that the gain of the op-amp is too high. So, the op-amp output saturates to supply voltage, and the LED is always on and fixed at maximum brightness, regardless of how the light level in the room changes (assume it cannot burn out). Now that the mechanics of how the circuit operates have been established, next we explain how our approach to designing this problem was informed by the theoretical lenses of complexity, ill-structuredness, knowledge type application, and the troubleshooting process.

## 3.2 Complexity and Ill-Structuredness of Troubleshooting Problems

Troubleshooting problems are ill-structured [2] for a few reasons. First, troubleshooting requires the solver to familiarize themselves with the system at play. To some degree, they must understand how the system is supposed to function correctly to then be able to tell when it is not, as well as to know when it is fixed. Besides that, they must sufficiently understand the specific fault within the system, where it is located, and the scope of the system that it affects, before they even can begin to solve the problem. In other words, you must figure out exactly what is wrong with the system before you can try to fix it. Accomplishing these preliminary tasks usually requires substantial work on the part of the *troubleshooter*, because this information is not given to them at the outset, contrary to how it would be with well-structured, textbook problems. All the solver has when they start engaging the problem is a system that is evidently malfunctioning and needs to be fixed. Beyond that, they have no other information or details about the system or the specific issue within it, so just by looking at the system-level behavior, it is still unclear exactly what the problem is that they need to solve. So, the solver *themself* must do this work to give structure to the problem by sufficiently understanding the system they are working with and defining exactly what is wrong with it, before working on possible solutions to fix it. In fact, structuring a troubleshooting problem in these ways is part of solving the problem and is a crucial step of the troubleshooting process itself.

Similarly, while the high-level goal of troubleshooting is known – i.e., you want to fix the system and return it to normal operation – ambiguity surrounding the nature of the fault in the system makes exact solutions to achieve that goal initially unclear as well. As demonstrated with the example circuit in this paper, a malfunctioning system can be caused by multiple different faults, so depending on which one it is, there can be multiple possible solutions (and thus paths to those solutions) to repair the system and ultimately achieve the same end result. The ill-structuredness of troubleshooting problems is due to a lack of familiarity with the system that is malfunctioning, the lack of clarity around what the exact issue might be, and, consequently, uncertainty as to which specific solution you should use to fix it.

The complexity of troubleshooting problems stems from the number of components or functional blocks within the system at play, and how they interact with each other [2]. Complex systems have many functioning parts that interact with each other in complicated ways to achieve overall system behavior; one component affects how others function, and vice versa. A malfunctioning system can be caused by any component or functional block within it being faulty, so the troubleshooter themselves must determine which one contains the fault. They must distinguish which parts of the system are broken and thus responsible for the abnormal behavior, and which are still functioning correctly (and which would otherwise continue to function properly if the fault in the system was repaired). However, complex interactions between system components can make it very unclear as to where the exact issue lies. Because of this interconnectedness, the faulty part of a system usually impairs how the otherwise "healthy" parts of the system function, which further obfuscates what the true culprit is and where it is located.

So, to determine which parts of a system are relevant to eliminating the fault (and which are not), the solver must decompose the system into its constituent components and analyze those – both individually, and as they are connected to other components. Likewise, any solution that the troubleshooter implements (by changing one component or one functional block of the system) will have cascading effects on the other parts of the system. This adds even more complexity to contend with when troubleshooting, because the solver must fix the issue while simultaneously maintaining proper functionality of the other system components that are working correctly, and without introducing new faults or disrupting other parts of the system. Reducing the complexity of a troubleshooting problem by narrowing down the problem space – i.e., possible faults and locations within the system – overlaps with resolving ill-structuredness, because it is towards the same goal of sufficiently understanding the system at play and isolating the fault within it. So, it is similarly required to holistically solve the problem, and it is an equally important step of the troubleshooting process as well.

### 3.3 Complexity and Ill-Structuredness of This Circuit Problem

We designed this circuit troubleshooting problem to be complex and ill-structured in a few ways. First, it lacks structure because students themselves must understand how the circuit (the system) is supposed to function at a high level, before they can even begin to fix it. They must know how a nightlight generally works to be able to tell that the circuit is malfunctioning. Then, they must familiarize themselves with the specific components and functional blocks within this circuit, how they are supposed to work, and how they interact with each other to create the overall functionality of the entire circuit. This circuit consists of several components connected in different ways, and electrically-speaking, it has three main functional blocks – the voltage divider network, the non-inverting amplifier, and the output LED. To give structure to this problem, students must be able to identify these three distinct functional blocks; understand how each circuit block works and what components they consist of; and most importantly, figure out how these parts of the circuit interact with each other when connected the way they are. Decomposing the circuit into these blocks and components overlaps with, and transitions into, resolving the complexity of the problem as well.

Once they gain a sufficient understanding of how the circuit works, students must then figure out what the exact fault is, narrow down where it is located within the circuit (i.e., which component or functional block is faulty), and assess how many of the other components the fault affects. In the case we are considering, the fault is that the op-amp feedback resistor values are incorrect, and the gain is too high; however, this fault is not easy to determine right away. Looking at overall circuit behavior, students would only observe that the LED remains on and at full brightness, regardless of the light level in the room or how they may manipulate the photoresistor. But this malfunction could be caused by several things: an issue with the input voltage divider network, or an incorrect current-limiting resistor, not only an inappropriate op-amp gain. So, students must be able to determine that the feedback resistors are the true culprit of the abnormal behavior, while the other components in the circuit are fine as they are, and would work properly if the op-amp gain is fixed. Likewise, students must come up with and implement a solution that corrects the op-amp gain while simultaneously maintaining the functionality of the voltage divider network and the output LED. Ideally, they change the feedback resistor values appropriately, but if they misattribute the issue in the circuit to a different cause, and thus attempt any number of other solutions, those could introduce new faults into the circuit and/or disrupt the other functional blocks.

## 3.4 Solving This Circuit Problem with the Troubleshooting Process

A few problem-agnostic and context-independent troubleshooting processes have been proposed in the extant literature, **which represent expert approaches to troubleshooting**. These theoretical processes include:

1. Jonassen and Hung's model [2], [3]
2. Johnson's Technical Troubleshooting Model (TTM) [10], [11]
3. Ross and Orr's DECSAR method [12]
4. Schaafstal and Schraagen's task analysis process [13]

We offer a combined theoretical process model synthesized from all four of those (Fig. 2).

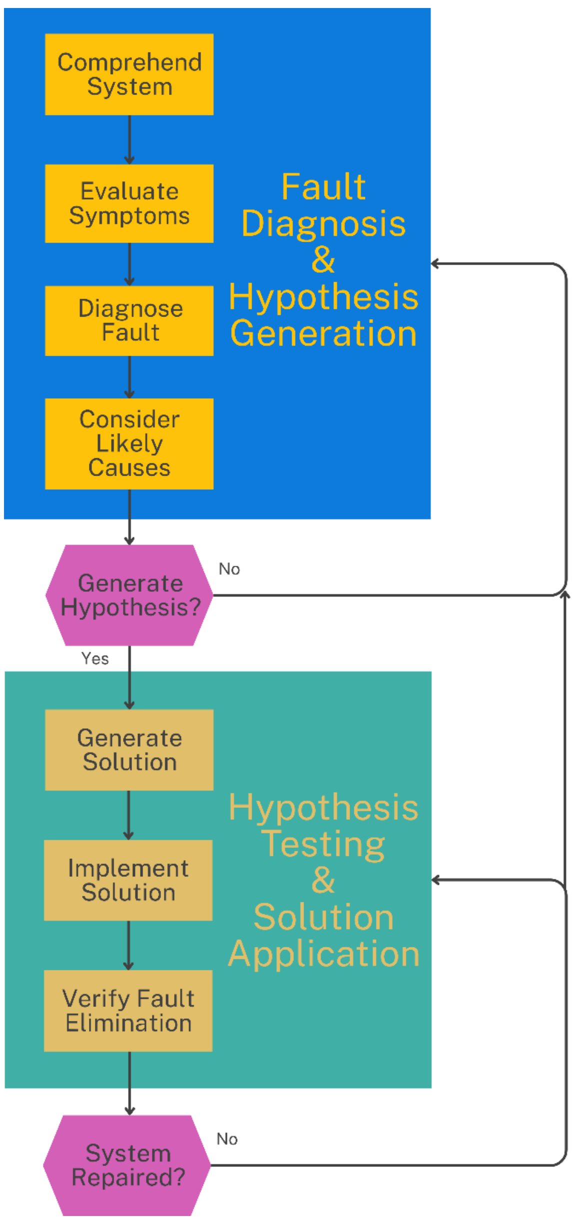


***Figure 2.*** *Combined theoretical troubleshooting process model*

To successfully troubleshoot a problem, you must engage in the troubleshooting process, which can be divided into two major phases. In the former phase, you attempt to identify, understand, and isolate the fault within a given system, with the goal of generating a hypothesis as to the likely cause of that fault. In the latter phase, you attempt to generate and implement a solution that will address the hypothesized cause of the fault, eliminate it, and repair the system. Each phase of this process consists of several crucial steps (usually performed iteratively). To illustrate the process of successfully troubleshooting this circuit, we list those steps below and describe what it would look like for students to apply them to this specific problem.

## Fault Diagnosis and Hypothesis Generation Phase

1. Thoroughly comprehend the system at play. Form expectations of normal function and expected outputs. Determine when and how the system is behaving abnormally.

   This has already been discussed with this problem and the fault case we are considering.

2. Evaluate observable symptoms to identify the fault and locate it within the system. Collect information necessary for sufficiently understanding the nature of the fault and the scope of the system it affects, which is usually done through observations and running appropriate tests. Narrow down possibilities for what the fault could be. Distinguish which components and functional blocks of the system could be faulty, and which are still functioning correctly.

   With this problem and the fault case we are considering, students would have to observe that the LED stays on and bright, regardless of the light level of the room. They should then test the circuit in different ways. As mentioned, they should manipulate the photoresistor and observe the same result (no change in output).

   After this, they should take strategic voltage and current measurements at various points around the circuit using a multimeter to gain more nuanced information that helps them narrow down what the true problem is and where it lies. Ideally, students would measure that the output voltage of the op-amp remains saturated at the supply voltage, regardless of how the circuit input changes (amount of light hitting the photoresistor). They may also measure that the input voltage of the op-amp does vary appropriately as the light level changes. All this information should indicate to students that the problem lies somewhere within the amplifier block of the circuit, because the voltage divider block is functioning correctly, and the output LED would otherwise function properly if the voltage supplied to it was appropriate. This testing and measuring helps eliminate areas of the circuit that could be broken and narrow down possibilities for the fault. Students may also measure the values of different resistors around the circuit, after which they should find that the voltage divider resistor and current-limiting resistor values are probably appropriate, which should reinforce the belief that those blocks of the circuit are functioning correctly.

3. Consider likely faults and their causes, and connect those with known symptoms, observations, and measurements. Attempt to cleanly distinguish the correct cause of the identified fault from other possibilities. Generate a hypothesis for what the fault is and what is causing the system to malfunction.

   Linking symptoms of a malfunctioning system to likely faults and causes is a particularly-challenging part of the troubleshooting process [13], [14]. If students can take the necessary measurements described above and narrow down where the fault could be in the circuit, they should reason that the likeliest fault is that the op-amp feedback resistors are incorrect. If they know how a non-inverting amplifier works and the formula for the gain, they should realize that the feedback resistor values are inappropriate and could be causing the gain to be too high. This in turn would cause the output voltage to always be high (which they also measured) and the LED to be constantly on at full brightness, which would explain the abnormal system behavior that they observed.

**Hypothesis Testing and Solution Application Phase**

4. Generate an appropriate solution that sufficiently addresses the hypothesized cause of the fault. Implement the solution to repair the system.

   In this case, students should replace the op-amp feedback resistors with ones that decrease the gain to a more appropriate level.

5. Evaluate the implemented solution. Verify that the issue has been fixed (i.e., the fault has been eliminated) and that the system functions normally again. Verify that all system components function properly together and that no new faults have been introduced.

   Once the feedback resistors are replaced, students should verify that the LED grows brighter and darker appropriately, and that the whole circuit functions properly as a nightlight. They may also measure the input and output voltages again to verify that they vary as they should for correct circuit operation. Students may need to try different feedback resistor values and calibrate them until they find a combination that makes the circuit operate as desired.

Students may struggle with any of the steps of this troubleshooting process. If they hypothesize the wrong fault, implement a solution that doesn't fix the true issue, or even introduce new faults into the circuit, then they must go back and iterate relevant steps of the process until they holistically solve the problem.

## 3.5 Types of Knowledge Required to Troubleshoot this Circuit

In addition to following the high-level, structured process above, successful troubleshooting requires the effective application of five specific types of knowledge [2]. We define what these knowledge types are and explain how they are relevant to troubleshooting this specific circuit. Students must possess and apply these types of knowledge to be able to solve this problem.

1. **Conceptual Knowledge** – Knowledge of higher-level, fundamental concepts relevant to a specific system you are troubleshooting. Necessary for all parts of the process.

   Students need to have knowledge of relevant circuits components and concepts, including voltage, current, resistors, op-amps, and LEDs, as well as how voltage divider and non-inverting amplifier configurations are supposed to work, too. This knowledge is vital for forming expectations of how the circuit is supposed to function properly, determining when it is behaving abnormally, identifying and isolating the fault, and implementing a solution that will repair the circuit correctly.

2. **Structural Knowledge** – Knowledge of important variables and constraints relevant to a specific system you are troubleshooting, and the functional relationships between them, which mathematically model how the system behaves. Usually expressed as equations.

   Students need to know the equations that govern the behavior of resistors, op-amps, and LEDs. Specifically, they need to know Ohm's Law to model how the voltage divider block should work; the formula for the gain of an op-amp to model how the non-inverting amplifier block should work; and the I-V characteristic of an LED (with current-limiting resistor) to model how the output should work. This knowledge is crucial to mathematically modeling how the circuit operates and comparing expected behavior to reality.

3. **Domain/Experiential Knowledge** – Familiarity with common practices and assumptions of a particular field, as well as the specific system you are troubleshooting, which is gained from past experiences working with similar systems. Usually manifests as familiarity with common issues that could arise within such systems or common failure modes of constituent components, as well as common causes of those faults and appropriate solutions for them.

   If students have past experience working with circuits like this one, or the components that make up this circuit, it may make it significantly easier for them to identify the cause of the fault and remediate it. If they have analyzed resistors, op-amps, and LEDs before, especially in similar configurations, they may more efficiently realize that the feedback resistors should be changed, and that the voltage divider and LED sections are fine as they are.

4. **Procedural Knowledge** – "knowing how", i.e., knowledge of well-defined procedures used to analyze a specific system, like standard testing procedures, steps of performing computations, or proper ways to use certain tools/equipment. Necessary for performing tests and collecting information about symptoms of an aberrant system while troubleshooting.

   Students need to know how to use a multimeter to take strategic voltage and current measurements at various points around the circuit to collect useful information for narrowing down fault possibilities. Students also need to know how to perform basic circuit analysis to model how the different circuit components interact the way they are connected, and to calculate what different voltages should be at different points around the circuit (in conjunction with structural knowledge), especially at the input and output of the amplifier.

These calculations/expectations should complement their measurements to distinguish which functional blocks of the circuit are working as intended, and which are not. As explained in the troubleshooting process in the previous section, students must take certain measurements and combine those with their calculated expectations to ultimately determine that the voltage divider and LED blocks of the circuit are functioning properly, but the non-inverting amplifier block is where the fault truly lies.

5. **Strategic Knowledge** – Knowledge of troubleshooting strategies, which can allow the solver to more efficiently narrow down the correct fault and its associated cause. Helps the troubleshooter more intentionally and strategically search for potential faults and causes within a malfunctioning system, and assists them with generating more effective solutions or coming up with alternatives, if necessary.

   Studies have shown that students – and technical novices in general – frequently use less efficient or practical strategies for identifying and isolating the fault, such as random trial and error, unintentional tinkering, or process of elimination (i.e., exhaustively testing and eliminating all possibilities one by one) [2], [5], [6], [7], [11]. If students can make use of more effective troubleshooting strategies, such as tracing or discrepancy detection, that should enable them to more efficiently reduce the problem space, as well as generate and pursue more intelligent hypotheses and solutions. The topographic strategy, which involves breaking down the system into its functional blocks to locate the fault, is another example of a more effective troubleshooting strategy that we used as part of illustrating how to solve this circuit problem via the exemplar troubleshooting process in the previous section.

# 4. Concluding Remarks for Engineering Instructors

## 4.1 Value of Troubleshooting Problems as an Instructional Tool

The circuit presented in this paper serves as an example troubleshooting problem that electronics instructors can integrate into their existing circuits classes as a laboratory experiment/exercise. This problem can be used to explicitly teach students the troubleshooting process, introduce them to more effective troubleshooting strategies and procedures, and provide them with an opportunity to intentionally practice and develop this important problem-solving skill.

However, beyond this specific problem, we believe that troubleshooting problems in general can be a particularly effective pedagogical tool for students to learn specific technical concepts and skills, because they provide a meaningful (and bounded) context within which students can apply them. As shown with the circuit problem we designed, troubleshooting requires students to apply different types of knowledge in strategic and intentional ways to figure out what the fault is and fix it. Problems like this present an interesting and non-trivial challenge to students because they must know when and how to use the knowledge and tools at their disposal to solve a real, practical problem with a tangible (but deceptively simple) goal. Students must think critically about which concepts and skills they have learned from class are relevant to fixing the issue at hand and judiciously apply them to solve the problem – and they must do this to a much greater extent than they are usually required to with the well-structured, textbook problems they are often given in engineering classes. At the same time, problems like this demonstrate to students the value of what they're learning in class by serving as an authentic use-case for it, applied to a scenario with tractable solution criteria. Troubleshooting problems like the one presented in this paper can be tailored to require students to operationalize a variety of different circuits concepts, equations, tools, testing procedures, and analysis methods in specific ways, depending on what the circuit consists of and how it is broken. Likewise, this type of problem can be used to creatively assess a student's mastery of these different constructs as well.

## 4.2 Potential for Instructors to Design Their Own Troubleshooting Problems

To take advantage of the potential instructional value of troubleshooting, we propose that engineering instructors design their own troubleshooting problems by purposefully manipulating the same theoretical aspects we used to design ours. We encourage instructors to use complexity, ill-structuredness, and knowledge types to think about the design of their own class exercises and try to leverage these dimensions in intentional ways. For example, you can increase the complexity of a circuit like the one presented here by adding components to it and/or connecting them together in different configurations, thereby complicating the way they interact with each other and making it more challenging for students to identify and isolate the fault. You can structure the problem less for students by giving them less information about the circuit and/or adding more functionality to it, thereby increasing the amount of work they must do to familiarize themselves with the system they're troubleshooting. You can tailor the components of the circuit, and the nature of the fault within it, to require students to practice applying desired types of knowledge in specific ways, which would be done through executing the different steps of the troubleshooting process, as we illustrated with our problem. We believe that designing problems using these theoretical lenses can result in much more meaningful, purposeful, and educative experiences for engineering students in classes and laboratories. We encourage engineering instructors across disciplines to try creating their own troubleshooting exercises for the specific classes and technical content that they teach.

## 4.3 Future Work

The circuit troubleshooting problem presented in this paper is being used in an ongoing research study to investigate undergraduate engineering students' troubleshooting processes, as well as their ability to apply different types of knowledge in a troubleshooting scenario. The goal is to collect implementation data to understand why students might struggle with problems like this through the theoretical lenses introduced here, and to compare how they engage the problem to expert troubleshooting practices and strategies.

# References


[1] P. Daly, M. Anand, C. Abel, and A. Sabuncu, "Industry Perspectives on Mechanical Engineering Troubleshooting," in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 47617. doi: 10.18260/1-2--47617.

[2] D. H. Jonassen and W. Hung, "Learning to Troubleshoot: A New Theory-Based Design Architecture," *Educ. Psychol. Rev.*, vol. 18, no. 1, pp. 77–114, Mar. 2006, doi: 10.1007/s10648-006-9001-8.

[3] D. H. Jonassen, "Toward a design theory of problem solving," *Educ. Technol. Res. Dev.*, vol. 48, no. 4, pp. 63–85, Dec. 2000, doi: 10.1007/BF02300500.

[4] H. J. Passow and C. H. Passow, "What Competencies Should Undergraduate Engineering Programs Emphasize? A Systematic Review," *J. Eng. Educ.*, vol. 106, no. 3, pp. 475–526, Jul. 2017, doi: 10.1002/jee.20171.

[5] C. Crockett, A. Barnes, G. Prpich, and A. McAlister, "BYOE: Teaching and Assessing Troubleshooting Strategies in Circuits Courses," in *2025 ASEE Annual Conference & Exposition Proceedings*, Montreal, Quebec, Canada: ASEE Conferences, Jun. 2025, p. 56049. doi: 10.18260/1-2--56049.

[6] M. Kinsel, C. Crockett, N. Smith, and G. Prpich, "Circuit Troubleshooting Techniques in an Electrical and Computer Engineering Laboratory," in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon: ASEE Conferences, Jun. 2024, p. 48461. doi: 10.18260/1-2--48461.

[7] G. Prpich, N. Smith, C. E. Crockett, and A. Shrestha, "Building Better Engineers: Teaching Chemical Engineers to Troubleshoot in the Laboratory," presented at the 2024 ASEE Annual Conference & Exposition, Jun. 2024. Accessed: Oct. 22, 2024. [Online]. Available: https://peer.asee.org/building-better-engineers-teaching-chemical-engineers-to-troubleshoot-in-the-laboratory

[8] D. R. Dounas-Frazer and H. J. Lewandowski, "Electronics lab instructors' approaches to troubleshooting instruction," *Phys. Rev. Phys. Educ. Res.*, vol. 13, no. 1, p. 010102, Jan. 2017, doi: 10.1103/PhysRevPhysEducRes.13.010102.

[9] N. M. Morris and W. B. Rouse, "Review and Evaluation of Empirical Research in Troubleshooting," *Hum. Factors*, vol. 27, no. 5, pp. 503–530, Oct. 1985, doi: 10.1177/001872088502700502.

[10] S. D. Johnson, "Cognitive Analysis of Expert and Novice Troubleshooting Performance," *Perform. Improv. Q.*, vol. 1, no. 3, pp. 38–54, 1988, doi: 10.1111/j.1937-8327.1988.tb00021.x.

[11] S. D. Johnson, "Understanding Troubleshooting Styles To Improve Training Methods.," 1995, Accessed: Sep. 18, 2024. [Online]. Available: https://eric.ed.gov/?id=ED389948

[12] C. Ross and R. R. Orr, "Teaching Structured Troubleshooting: Integrating a Standard Methodology into an Information Technology Program," *Educ. Technol. Res. Dev.*, vol. 57, no. 2, pp. 251–265, 2009.

[13] A. Schaafstal and J. M. Schraagen, "Training of Troubleshooting: A Structured, Task Analytical Approach," in *Cognitive Task Analysis*, Psychology Press, 2000.

[14] A. Schaafstal, J. M. Schraagen, and M. Van Berl, "Cognitive Task Analysis and Innovation of Training: The Case of Structured Troubleshooting," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 42, no. 1, pp. 75–86, Mar. 2000, doi: 10.1518/001872000779656570.