

Query-Based Generation of Trigonometric Identity Problems and Solutions

Meredith Pearce
Furman University
meredith.pearce@furman.edu

Jonathan McKinney
Furman University
jonathan.mckinney@furman.edu

Chris Alvin
Furman University
chris.alvin@furman.edu

Abstract

We propose a technique for generating trigonometric identity problems (TIPs) and problem templates based on the difficulty of the desired problem. Specifically, given an expression ℓ and a query Q describing desired problem components and a desired measure of difficulty, we explore the space of tautological expression problem paths using a rule-based substitution technique. Once a tautological expression r and its corresponding solution path within the desired level of difficulty is achieved, a problem of the form $\ell = r$ is returned along with an underlying bi-directional solution path. We also demonstrate that our tool is able to solve a corpus of textbook TIPs as well as generate problems that extend beyond the confines of problem structures present in textbooks.

1 Introduction

A trigonometric identity problem is a classic Pre-Calculus problem that is solved by a student establishing a tautological proof between two expressions. In this work we present techniques for generating, solving, and analyzing the difficulty of high school TIPs based on user query. Among mathematics educators, this problem type readily exposes student strengths and/or weaknesses in algebra skills. Our work empirically shows that TIP is a misnomer since the main set of skills they test are algebraic, not trigonometric. For example, a student may find $x^2 - 1 = (x - 1)^2$ erroneous, but affirm that $\cos^2 x - \sin^2 x = (\cos x - \sin x)^2$.

Solving. As an introduction to our technique, we consider a textbook TIP (Demana et al. 2010):

$$\tan(2 \cdot x) = \frac{2 \cdot \cos(\frac{\pi}{2} - x) \cdot \cos(x)}{1 - 2 \cdot \sin^2(x)}. \quad (1)$$

Demonstrating the validity of such a tautology may require the application of algebraic tautologies (e.g., $x \cdot x = x^2$, $(1/x)^2 = 1/x^2$) and trigonometric tautologies which students are often asked to memorize (e.g., $\cos x = 1/\sec x$, $\cos^2 x + \sin^2 x = 1$, etc.). Students then decide whether to fix the left hand side (LHS) of the tautology as the start expression s with the goal expression g being the right hand side (RHS) of the tautology (or vice versa). A solution to

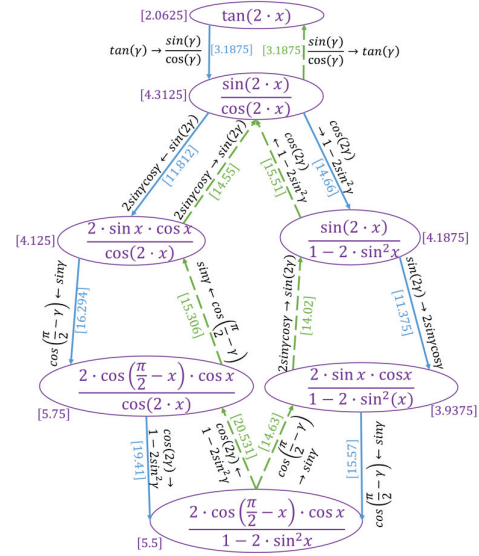


Figure 1: Sample solutions to the trigonometric identity problem from Equation 1.

the TIP is then a sequence of manipulations of the tautological expressions beginning with s and ending with g . For example, a student may identify $\tan 2x$ (LHS of Equation 1) as the start expression. Then, as shown in Figure 1 with the solid, downward directed edges, the student may initiate a sequence of substitutions with the intent to derive $g = \frac{2 \cdot \cos(\frac{\pi}{2} - x) \cdot \cos(x)}{1 - 2 \cdot \sin^2(x)}$. Each tautological expression in the solution path of a TIP is substantiated using the ‘known’ algebraic or trigonometric tautology that was used as the basis for a *substitution*; e.g., applying $\tan \gamma \rightarrow \frac{\sin \gamma}{\cos \gamma}$ results in $\tan 2x = \frac{\sin 2x}{\cos 2x}$ in Figure 1.

Difficulty. We measure the difficulty of a TIP by computing the cumulative difficulty of the constituent steps in a solution path. A solution to a TIP begins with a start expression, either LHS or RHS and identifies a sequence of steps from the start to goal (or vice versa). These choices impact perceived problem difficulty. Our first step in measuring problem difficulty is to assess the difficulty of the individual expressions in a shortest solution. We first compute domain

specific characteristics: such as the depth of nested divisions (e.g., $1/[1/\sin \gamma]$ is depth 2) or the presence of an absolute value or coefficient (e.g., $\sin 2\gamma$). Second, we compute characteristics of its syntax tree representation (e.g., height, width, number of operators, etc.).

The second difficulty step assesses the relative difficulty of a solver moving from one trigonometric expression (TE) to another. If a and b are TEs such that a can be derived from b using a single step (and vice versa), we compute the relative difficulty as a combination of the difficulty of both expressions and the substitution step. Last, we assess the overall difficulty of a problem as the sum of the transitions in a corresponding solution.

As an example, consider Figure 1. We label the difficulty of each individual TE node as well as each directed edge between nodes. Specifically, we observe a difficulty of $\tan 2x$ as 2.0625 and $\frac{\sin 2x}{\cos 2x}$ as 4.3125. Each edge has a perceived difficulty in the range $[1, 5]$ where 1 is considered easiest to apply when solving and 5 is most difficult. We then compute the edge difficulty using a trapezoidal method $\frac{1}{2}(2.0625 + 4.3125) \cdot 1 = 3.1875$ where 1 is the perceived difficulty of applying $\tan 2\gamma \leftarrow \frac{\sin \gamma}{\cos \gamma}$ or $\tan 2\gamma \rightarrow \frac{\sin \gamma}{\cos \gamma}$.

Generating problems. Given a query Q defining difficulty threshold (or a default value) for a TE, our approach to generating TIPs and their solutions exhaustively considers the set of trigonometric identities and a subset of algebraic identities common in solving this type of problem. Given a TE and the set of trigonometric and algebraic identities, we substitute to construct a set of TEs, repeating this process until other factors in Q (e.g., maximal number of steps, etc.) indicates we cease substituting and return a TIP.

2 Limitations of Notable Prior Work

Singh, et al (Singh, Gulwani, and Rajamani 2012) proposed a technique for generating ‘algebra’ problems. Their technique consists of abstracting a problem into a query Q (a structural *template*). Given the query, they use random testing to generate all permutations of problems with the same structure. This technique, while noteworthy, does little to generate unique and interesting problems for students beyond standard textbook structures. (Singh, Gulwani, and Rajamani 2012) state that the trigonometric identity problem

$$\frac{\sin A}{1 + \cos A} + \frac{1 + \cos A}{\sin A} = 2 \csc A \quad (2)$$

leads to 8 similar generated problems. Since each of the 8 TIPs are structurally identical, their corresponding solutions evidence similar algebraic and trigonometric deductions at each step. Thus, it is possible for a student to memorize a solution path inherent in the structure of a problem.

This mastery approach to teaching and learning can be effective in educating students requiring strong repetition, but it does not facilitate student learning beyond textbook structure. Our work attempts to build on (Singh, Gulwani, and Rajamani 2012), accomplishing three distinct goals with respect to TIPs. First, we generate TIPs with unique structures and thus unique problem templates (*queries* in the parlance

Table 1: Known Trigonometric Identities with Examples.

Classification	Example Known Identities
Tangent / Cotangent	$\tan \gamma = \frac{\sin \gamma}{\cos \gamma}$
Reciprocal	$\cos \gamma = \frac{1}{\sec \gamma}$
Pythagorean	$\sin^2 \gamma + \cos^2 \gamma = 1$
Even / Odd	$\cos(-\gamma) = \cos \gamma,$ $\sin(-\gamma) = -\sin \gamma$
Double-Angle	$\sin 2\gamma = 2 \sin \gamma \cos \gamma,$ $\cos 2\gamma = \cos^2 \gamma - \sin^2 \gamma$
Half-Angle	$\sin \frac{\gamma}{2} = \pm \sqrt{\frac{1 - \cos \theta}{2}}$
Sum and Difference	$\cos(\gamma \pm \nu) =$ $\cos \gamma \cos \nu \mp \sin \gamma \sin \nu$
Product to Sum	$\sin \gamma \sin \nu =$ $\frac{1}{2} [\cos(\gamma - \nu) - \cos(\gamma + \nu)]$
Co-Function	$\cos(\frac{\pi}{2} - \gamma) = \sin \gamma$
Product to Sum	$\sin \gamma + \sin \nu =$ $2 \sin(\frac{1}{2}(\gamma + \nu)) \cos(\frac{1}{2}(\gamma - \nu))$

of Singh, et al.). Second, our problem generation system operates by way of a query system where the user can input the level of difficulty, desired trigonometric identities, and algebra rules. Third, we empirically show that solving TIPs is strongly related to the algebra skills of the solver.

3 Representation

3.1 Expressions, Problems, and Solutions

We assume a *mathematical expression* is consistent with the rules of algebra (i.e., associativity, distributivity, etc.) over the standard mathematical operators: multiplication, division, addition, subtraction, and exponentiation. We consider an individual TE such as $\tan 2x$ in Figure 1.

Definition 1 (Trigonometric Expression). *A trigonometric expression is a mathematical expression explicitly or implicitly involving a trigonometric function (sine, cosecant, etc.).*

Both the LHS and RHS of Equation 2 are TEs explicitly using trigonometric functions. Whereas, non-coefficient integer constants such as 1 implicitly define a TE and occur in isolation (e.g., $\sin^2 \gamma + \cos^2 \gamma = 1$) or in many ‘standard’ identities such as $1/\sin \gamma$ or $1 + \tan^2 x = \sec^2 x$.

As described in Section 1, a TIP consists of two tautological TEs stated using the overloaded equals operator: $\ell = r$. A solution then derives a sequence of tautological steps between ℓ and r by applying *known* trigonometric identities and algebraic tautologies. We will refer to this set as *known identities*; trigonometric examples are given in Table 1. Most TIPs have many solutions of varying perceived levels of difficulty; therefore, we define such a problem alongside a solution.

Definition 2 (Trigonometric Identity Problem). *A trigonometric identity problem is a tuple $\langle \ell, r \rangle$ expressed as a statement of the form $\ell = r$ where ℓ and r are tautological trigonometric expressions.*

Definition 3 (Trigonometric Identity Problem Solution). *A solution to a trigonometric identity problem $p = \langle \ell, r \rangle$ is a sequence of tautological expressions ℓ, t_1, \dots, t_n, r or the reversed sequence r, t_n, \dots, t_1, ℓ .*

Two adjacent TEs in a solution are tautological rewritings of one another; thus, each rewriting is bi-directional. Hence, a student might solve the TIP in Equation 1 using any of the four solution paths depicted in Figure 1; there are other solution paths not depicted. We can capture all solutions to a given TIP (and more) in a graph.

Definition 4 (Trigonometric Expression Graph). A trigonometric expression graph G is a directed graph $G = (T, E)$ where T is a set of TEs and each edge $(a, b) \in E$ is mapped to a known identity facilitating the rewriting from $a \in T$ to $b \in T$: $\pi : E \rightarrow K$ where K is the set of known identities.

We link Definition 3 and Definition 4 noting that a solution to a TIP is a path in a corresponding graph; we omit proofs to our lemmas as trivial.

Lemma 3.1 (Solution and Path Correspondence). For a trigonometric identity problem $p = \langle \ell, r \rangle$ and a solution to p consisting of $n + 2$ nodes $S = \ell, t_1, \dots, t_n, r$ is a bijection onto a path in a trigonometric identity graph $G = (T, E)$.

Definition 4 defines a super-structure that may contain many TIPs. We define a more restrictive structure containing only tautological expressions.

Definition 5 (Tautological Trigonometric Expression Graph). A tautological trigonometric expression graph G is a trigonometric expression graph in which all trigonometric expressions corresponding to nodes in G are tautologies.

We make two claims about the relationship between a trigonometric expression graph (TEG) and the more restrictive tautological TEG due to the fact that tautological expressions form an equivalence class.

Lemma 3.2. Each disconnected component of a trigonometric expression graph is a tautological trigonometric expression graph. A tautological trigonometric expression graph consists of a single strongly connected component.

We build on the previous definitions to incorporate perceived complexity of TIPs by annotating TEGs. We say that a *difficulty descriptor* $d \in D$ is an aggregate structure composed of values describing elements of assessed difficulty. A *difficulty-based trigonometric expression graph* $G = (T, E)$ is a trigonometric expression graph in which all nodes in T map to a difficulty descriptor: $\delta_T : T \rightarrow D$. Similarly, all edges in E map to a difficulty descriptor: $\delta_E : E \rightarrow D$.

4 Difficulty Model

4.1 Difficulty of Trigonometric Expressions

We compute a perceived difficulty of a TE as an average of four measures in the range $[0, 10]$; each measure is bounded above by 10 so one measure does not skew the overall difficulty of a TE.

Measure 1: Trigonometric Function Count. Each of the basic trigonometric functions (sine, cosine, tangent) are assigned a weight of 2 for each occurrence while reciprocal functions (cosecant, secant, and cotangent) are perceived to be more complex and receive a weight of 3 for each occurrence. We compute the sum of the weights for a TE t . For example, $\tan 2x$ from Equation 1 has measure 2 for one occurrence of tangent.

Table 2: Sample crowdsourced difficulty measures: $[1, 5]$.

Difficulty →	Known Identity	Difficulty ←
1.0	$\tan \gamma \xrightarrow{\quad} \frac{\sin \gamma}{\cos \gamma}$	1.0
3.158	$\sin \gamma \xrightarrow{\quad} \cos\left(\frac{\pi}{2} - \gamma\right)$	3.316
3.623	$\frac{\tan \gamma + \tan \nu}{1 - \tan \gamma \tan \nu} \xrightarrow{\quad} \tan(\gamma + \nu)$	4.211

Measure 2: Multiple Angles. We consider a scalar s in a TE such as $\tan s\gamma$. If $s < 0$ (e.g., $\sin(-\gamma)$), we assess a score of 2 since a known odd / even identity (see Table 1) may need to be applied. If s is power of 2 (i.e., $s = 2^n$ for some integer $n \geq 1$), then the TE such as $\cos 2\gamma$ is expandable by a double-angle identity (Table 1). In this case, we assess a score of $2s$. Similarly, if $s = 2^n$ for some integer $n < 0$, then a half-angle identity (Table 1) applies: we assess a score of $\frac{1}{2s}$. If $|s| < 1$ and is not a power of 2, the expression is considered to already be in simplest form and is measured at 2. For $|s| > 1$ where s is not a power of 2, we may use a multiple-angle identity or sum / difference identity (Table 1) assigning a weight of 2^s . For example, we may rewrite $\cos 6\gamma$ as $\cos(2 \cdot 3\gamma)$ or $\cos(2\gamma + 4\gamma)$.

Measure 3: Algebraic Functions and Operators. TIPs and their solutions may include absolute value or exponentials. We count the number of occurrences n of absolute value and weight $n = 1 \Rightarrow 5$, $n = 2 \Rightarrow 8$, $n = 3 \Rightarrow 9$, $n = 4 \Rightarrow 9$, and $n \geq 5 \Rightarrow 10$. For exponentials of the form $\sin^s \gamma$, we base the weight on the value of s . Specifically, squaring is common ($s = 2 \Rightarrow 2$) while square roots ($s = \frac{1}{2} \Rightarrow 6$) are less common. For all other cases, if $s > 1$, the expression results in a score of 5; if $s < 1$, it scores 10.

Measure 4: Trigonometric Expression Structure. We represent each TE as a syntax tree maintaining operator precedence and associativity. We first average the height and width of the syntax tree. Second, we sum the number of operands (tree leaves) and the depth of nested divisions. For example, $\frac{2 \cdot \cos(\frac{\pi}{2} - x) \cdot \cos(x)}{1 - 2 \cdot \sin^2(x)}$ contains 7 operands and a nested division depth of 2 since $\frac{\pi}{2}$ is a subtree of two operands rooted with division. The two values are then averaged as the overall measure of TE structure.

4.2 Step-Based Problem Difficulty

A solution to a TIP is a sequence of tautological steps; each transition (step) with a perceived level of difficulty. For a difficulty-based TEG $G = (T, E)$ with $(a, b) \in E$, we compute the difficulty transitioning from a to b using a trapezoidal area method: $\delta_E((a, b)) = \frac{1}{2} [\delta_T(a) + \delta_T(b)] \cdot d_{(a,b)}$ where $d_{(a,b)}$ refers to a perceived difficulty measure; examples of perceived difficulty are given in Table 2 and are described in Section 7. As depicted in Figure 1, we compute $\delta_E(\tan 2x, \frac{\sin 2x}{\cos 2x}) = \frac{1}{2} (2.0625 + 4.3125) \cdot 1.0 = 3.1875$. Due to perceived difficulty, it is not always the case that $d_{(a,b)} = d_{(b,a)}$: $\cos^2 \gamma + \sin^2 \gamma \rightarrow 1$ is common while it is counter-intuitive when to apply $1 \rightarrow \cos^2 \gamma + \sin^2 \gamma$.


```

Query      ::= <TrigExpr, InclConstrs,
                ExclConstrs, DConstrs>

TrigExpr   ::= te |  $\in$ 
InclConstrs ::= <May, Must>
ExclConstrs ::= <May, Must, MayTExpr>
May        ::= IList
Must       ::= IList
IList      ::= set consisting of:
                any Known Identities
                or trigonometric expressions
                set of any algebraic constructs

MayTExpr   ::= set of any algebraic constructs
DConstrs   ::=  $\in$  | DConstrs min_prob_diff  $\leftarrow$  4
                | DConstrs max_prob_diff  $\leftarrow$  75
                | DConstrs min_algebra_steps  $\leftarrow$  0
                | DConstrs max_algebra_steps  $\leftarrow$  10
                | DConstrs min_trig_steps  $\leftarrow$  2
                | DConstrs max_trig_steps  $\leftarrow$  10
                | DConstrs average_diff  $\leftarrow$  5
                | DConstrs max_step_change  $\leftarrow$  4
                | DConstrs max_expr_diff  $\leftarrow$  7

```

Figure 2: Syntax of the query language.

Solution Difficulty. Let $p = \langle \ell, r \rangle$ be a TIP along with a corresponding difficulty-based trigonometric expression graph $G = (T, E)$. Also let S be a shortest path solution to p in G consisting of $n - 1$ steps: $S = t_1, \dots, t_n$ where $\ell = t_1$ and $t_n = r$. We compute the overall difficulty of p with shortest solution S as a sum of the difficulty of the edges in S : $\delta(p, S) = \sum_{1 \leq i < n} \delta_E(t_i, t_{i+1})$.

Since S is a solution to p , it follows that the reversed sequence $S_r = t_n, \dots, t_1$ is also a solution. Due to perceived difficulties, it is unlikely $\delta(p, S) = \delta(p, S_r)$. For example, the left path in Figure 1 has ‘downward’ difficulty 50.7035 while it has a reversed difficulty of 53.5745.

Problem Difficulty. A TIP may have multiple shortest path solutions; Figure 1 depicts two bi-directional shortest path solutions. We say the difficulty of a problem p , $\delta(p)$, is the minimum difficulty across all shortest path solutions (and their reversals) to problem p .

5 Query Language

The space of tautological TEs is infinite; therefore, we define a query language to prune the space and focus generation of problems that are more likely to be of interest to the user. As shown in Figure 2, a query Q is a tuple consisting of (1) a TE (terminal **te** in the grammar), (2) a set of known identities or expressions to be included in the resulting problem (InclConstrs), (3) a set of known identities or expressions to be omitted from a problem solution, and (4) a set of difficulty-based constraints (DConstrs). The result of such a query will be a TIP.

The first element in a query Q is a fixed trigonometric expression **te** that will be used as stimulus for problem generation. If the user does not provide an expression, an expression will be drawn from existing database of TEs. For example, any of the TEs in Figure 1 could be used to generate the TEG containing the problem in Equation 1.

Q can state which known trigonometric identities may or must be involved in the TIP (InclConstrs). A Must request may come with a cost. If a user requires a known identity k be used in a problem, it requires that *all* solution paths

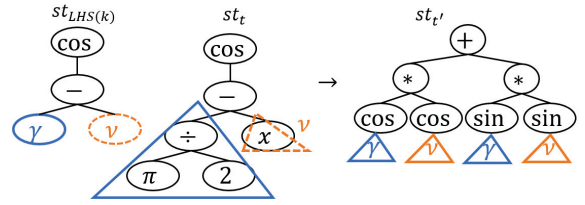


Figure 3: λ pattern matching $k \equiv \cos(\gamma - \nu) \rightarrow \cos \gamma \cos \nu + \sin \gamma \sin \nu$ to TE $t = \cos(\frac{\pi}{2} - x)$ resulting $t' = \cos \frac{\pi}{2} \cos x + \sin \frac{\pi}{2} \sin x$.

contain the use of k . For example, all solutions to the TIP in Equation 1 use the known identity $\tan \gamma = \frac{\sin \gamma}{\cos \gamma}$. Hence, Equation 1 could be generated if Q contained $\tan \gamma = \frac{\sin \gamma}{\cos \gamma}$ in the InclConstrs.Must list. Whereas Equation 1 could be generated if Q contained $\cos(\frac{\pi}{2} - \gamma) = \sin \gamma$ in the InclConstrs.May list. To facilitate solution generation of a particular TIP p , the user can include both the LHS and RHS of p in InclConstrs.Must.

Since a tautological TEG is a strongly connected component, a non-empty InclConstrs.Must set of constraints can be difficult to enforce in practice. Instead, Q can define a list of known identities that may appear in a shortest path solution to the problem, InclConstrs.May.

Similarly, the user can specify known identities that may (May) or must (Must) be excluded (ExclConstrs). In addition, the user also has the ability to request that a particular algebraic structure *may* be omitted in a problem. For example, the user might specify that factoring a quadratic expression or simplifying nested division should not be required. Since our techniques target mastery of trigonometric identities, we do not allow the user to define algebraic constraints that *must* not occur in the resulting problem.

Q allows the user to define difficulty-based constraints (DConstrs) as a means of pruning the space of TIPs by specifying problem-based constraints. If the user fails to define these constraints, default values are enforced (indicated using \leftarrow in Figure 2). For example, the user can restrict the number of known algebraic identities or trigonometric steps.

6 Problem and Solution Generation

We generate problems on-demand with an input query Q . We segment Q into two types: TEG generation information ($Q.TrigExpr, Q.Dconstrs$) and problem generation information ($Q.InclConstrs, Q.ExclConstrs$).

Applying known identities. Generating a TEG relies on an operation that takes a TE t and transforms it into TE t' by way of *applying* a known identity. We do so using a technique similar to applying transformations in λ -calculus. For example, we may apply the known identity $k \equiv \frac{\sin \gamma}{\cos \gamma} \rightarrow \tan \gamma$ to $t = \frac{\sin 2x}{\cos 2x}$. However, applying k requires we must recognize γ in k corresponds to $2x$ in t . The result of this pattern matching operation is $t' = \tan 2x$.

In general, our structural, λ -pattern matching algorithm for applying known identities to TEs proceeds as follows.

Let k be a known identity with a corresponding syntax tree representation for the LHS and RHS, $st_{LHS(k)}$ and $st_{RHS(k)}$, respectively. Let t be a TE with syntax tree st_t . We traverse the internal (non-leaf) nodes of st_t seeking a node c that equates to the root of $st_{LHS(k)}$ (e.g., cosine root nodes in Figure 3). Once c is found, our goal is then to traverse simultaneously both $st_{LHS(k)}$ and the subtree rooted at c in st_t as long as nodes containing non-free variables directly correspond; when correspondence fails, applying k fails. Once we encounter a node in $st_{LHS(k)}$ corresponding to a free variable (e.g., γ in Figure 3), we bind that free variable to the corresponding subtree in st_t . In Figure 3 we bind γ to $\frac{\pi}{2}$ and ν to x . Once our free variables are bound, we can perform the substitution on bound variables in $st_{RHS(k)}$ acquiring $t' = \cos \frac{\pi}{2} \cos x + \sin \frac{\pi}{2} \sin x$ in Figure 3.

Trigonometric Expression Graph Generation. We construct a TEG G using a breadth-first approach beginning with a stimulus TE $t_s = Q.\text{TrigExpr}$ being the ‘center’ of G . If $t_s = \text{nil}$, we look up a TE in a database. We expand outward from t radially using the distance, r , from t , until the radial distance is half the maximum number of steps specified by $Q.\text{max_steps} = Q.\text{max_algebra_steps} + Q.\text{max_trig_steps}$. Our main iteration step takes a pair (t, r) consisting of a TE in G and its distance from t_s from a worklist of unprocessed pairs. We then sequentially apply all known identities, $k \in K$ using our λ pattern matching to t . If applying k to t results in a TE t' not filtered out by Q , we (1) add t' one radial distance away from t to G (with appropriate edges) and (2) add the same pair to the worklist.

Generation. Let G be a TEG. Our first step in problem generation identifies the set of tautological TEGs (strongly connected components) of G , \mathcal{G} . Problem generation then chooses any two non-adjacent nodes in any $G_t \in \mathcal{G}$ (guaranteeing all problems require at least two steps to solve). Assume a query Q is defined with restrictions on problem generation: steps, difficulty, etc. For a tautological TEG G_t we compute the difficulty of all nodes and edges resulting in a difficulty-based tautological TEG, G_{dt} . Then, for G_{dt} , solution generation identifies the set of shortest path solutions \mathcal{S} among all non-adjacent pairs of nodes in G_{dt} . We then filter \mathcal{S} according to the constraints defined by Q .

7 Experiments

We are foremost interested in demonstrating the utility of our approaches by solving and analyzing the difficulty of problems found in textbook. Second, we describe some queries and results for problem generation. We first describe our difficulty-related experimental assumptions.

Crowdsourced Step Difficulties. As described in Section 4, difficulty of each step in a TIP solution is based on the perceived difficulty of applying a known trigonometric identity to a TE. To generate these perceived difficulties, we constructed a survey consisting of 46 known trigonometric identities (a sample is shown in Table 1). The survey presented each known trigonometric identity twice, once for each direction in a randomly permuted order as $\ell \rightarrow r$ or $r \leftarrow \ell$. Each of the 24 faculty (3) and undergraduate students (21) participants who satisfactorily completed the sur-

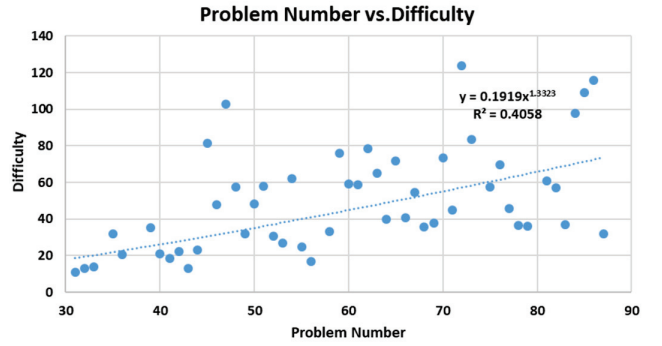


Figure 4: Problem number compared to problem difficulty.

Table 3: Statistics of our 90 problem corpus.

$n = 90$	Alg. Steps	Trig Steps	Total Steps	Prob. Diff.	Avg. Prob. Diff.	Abs. Cumul. Change in Diff.
Min	1	0	2	10.43	3.24	0.5
Median	3.5	2	6	36.42	6.84	4.78
Max	10	6	14	128.63	11.1	10.63
Mean	4.11	2.36	6.47	44.28	6.67	4.83
Std Dev	2.10	1.34	2.83	26.21	1.85	2.51

vey were asked to classify each known identity transition on a scale of 1 being the easiest to recognize and apply while 5 is the most difficult. We averaged the survey responses as the perceived difficulty of known, directional trigonometric identities; see Table 2.

Known Algebraic Identities. Solutions to most TIPs require algebra to solve. For example, showing $1 - 2\sin^2 x$ is a tautology with $(\cos x - \sin x)(\cos x + \sin x)$ requires factoring a difference of squares: algebraically, $\gamma^2 - \nu^2 = (\gamma - \nu)(\gamma + \nu)$. As our goal is to assess the relative difficulty of applying known trigonometric identities, we fixed the difficulty of *all* known algebraic identities to be 2, admittedly losing some nuance in assessing difficulty.

Solving. As a baseline, we verified that our techniques successfully solve textbook problems. Using a corpus of 90 TIPs from seminal mathematics textbooks (Demana et al. 2010; Stewart, Redlin, and Watson 2015), we defined a query Q with maximum problem steps 14, maximum problem difficulty 130, maximum trigonometric steps 8, and a subset of size 30 known identities of the 46 from Table 1. In order to focus the search of the solution space, for each individual problem $p = \langle \ell, r \rangle$, we defined $Q.\text{TrigExpr}$ and executed Q with either ℓ or r chosen randomly (with both ℓ and r as elements of $\text{InclConstrs}.\text{Must}$). With this set of constraints, we successfully identified a shortest path for all 90 TIPs; Table 3 gives summary statistics of our corpus.

We enumerate some interesting pedagogical results with respect to our corpus of textbook problems. (1) One problem did not require any trigonometric identity to solve. (2) With mean 4.11 algebra steps compared to 2.36 trigonometric steps per TIP, we state with confidence that TIPs assess student algebra skills more thoroughly than their knowledge

of trigonometry. (3) Students and educators often align their view of textbook problem difficulty with problem number: the later the problem appears, the more difficult it will be relative to the previous problems. To investigate, we took an ordered sequence of 51 problems from one section of (Demana et al. 2010) and plotted their problem numbers against their difficulty as shown in Figure 4. We note that these 51 problems were taken from two non-adjacent ‘groupings’ of problems in the text; however, the instructions indicated a continuation thus it was fair to investigate the problems a single grouping of problems. Although the distribution shows a positive correlation, it is not strong; this implies some refinement of our difficulty model is required and / or sequencing TIPs based on difficulty is not likely to be the case. (4) The absolute cumulative change in difficulty in Table 3 refers to the sum of the (absolute) changes between the difficulty of each TE in a shortest solution path. Figure 4 indicates that the distribution of absolute change is approximately normal (mean 4.83 and median 4.78). We observe Figure 1 solves a ‘typical’ problem and has absolute change 4.1875 and thus conclude that most TIP solutions have changes in difficulty commensurate with Figure 1.

Problem generation. Using the average textbook problem number of steps 6.47, we executed a sequence of 10 queries with $Q.TrigExpr$ set to different stimulus expressions such as $\csc x - \sin x$ and $\frac{\tan x}{\sec x}$ and the maximum number of steps set to 6. The result was an average of 1844 (std. dev. 1028.54) problems. As an example, from $\frac{\tan x}{\csc x}, \left(\frac{1}{\cos^2 x} - \tan^2 x\right), \frac{1}{\sin x}$ was generated. While the structure of the resulting expression does not conform to textbook structures, the problem is reasonable in terms of difficulty (16.17) taking only 3 steps. Such unique problem structures is not a strength of textbook problems. Coupling together our techniques to generate problem templates with Singh, et al. (Singh, Gulwani, and Rajamani 2012) would generate an abundant number of problems.

8 Related Works

Etzioni, et al. (Clark, Etzioni, and et al. 2019; Koncel-Kedziorski et al. 2015; Seo et al. 2015) have worked toward NLP understanding, diagrammatic reasoning, and solving of problems in Aristo. In (Clark, Etzioni, and et al. 2019) they describe their path toward addressing their grand challenge of passing Grade 8 New York Regents science exam. Our approach does provide a means of solving TIPs via our query language. However, our central contribution is a template generation procedure via substitution along with problem and solution generation. We also propose a schema for classifying the difficulty of a particular class of problems based on user sentiment of rules as opposed to crowd-sourced difficulties for a specific corpus of problems.

O’Rourke et al. (O’Rourke et al. 2019) proposed an answer set programming (ASP) model to generate practice problems and intelligent explanations of problems requiring students to solve algebraic equations. In some respects, edges in our TEGs mimic the underlying if-then production rules of the O’Rourke model thus providing a foundation for future work. Our goal is not to generate step-by-step expla-

nations, although that is possible, our goal is to foster greater diversity in generated TIPs. We feel (O’Rourke et al. 2019) lays the groundwork for ASP models with more complex problems. Trigonometry is a richer area since it relies on such algebraic knowledge, but prohibits using solving techniques (mixing the LHS and RHS of a TIP). TIPs are solved via a path of substitutions; the decision making surrounding those substitutions often being complex.

9 Conclusions and Future Work

Using a rule-based substitution technique, we have generated trigonometric identity problem solutions and problem templates based on user query. We have formalized a trapezoidal model for computing step-based difficulty in problems in concert with crowdsourced perceptions of difficulty. Our query-based approach proved to be efficient and effective in solving TIPs and generating such problems. Our technique can be used to assist educators with greater variety in problem structures thus assisting students in their learning. We believe our technique is applicable beyond the class of TIPs extending to any problem solvable using a strict, rule-based rewriting system. Future work includes abstracting our approach to a difficulty-based meta-model using Knuth-Bendix completion (Knuth and Bendix 1983).

10 Acknowledgements

This work was supported by the Furman University Department of Computer Science and The Furman Advantage.

References

- Clark, P.; Etzioni, O.; and et al. 2019. From ‘F’ to ‘A’ on the N.Y. regents science exams: An overview of the aristo project. *CoRR* abs/1909.01958.
- Demana, F.; Waits, B.; Foley, G.; and Kennedy, D. 2010. *Precalculus: Graphical, Numerical, Algebraic*. Pearson, eighth edition.
- Knuth, D. E., and Bendix, P. B. 1983. *Simple Word Problems in Universal Algebras*. Berlin, Heidelberg: Springer Berlin Heidelberg. 342–376.
- Koncel-Kedziorski, R.; Hajishirzi, H.; Sabharwal, A.; Etzioni, O.; and Ang, S. D. 2015. Parsing algebraic word problems into equations. *TACL* 3:585–597.
- O’Rourke, E.; Butler, E.; Tolentino, A. D.; and Popovic, Z. 2019. Automatic generation of problems and explanations for an intelligent algebra tutor. In *Artificial Intelligence in Education - 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I*, 383–395.
- Seo, M. J.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. 2015. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 1466–1476.
- Singh, R.; Gulwani, S.; and Rajamani, S. K. 2012. Automatically generating algebra problems. In *AAAI*.
- Stewart, J.; Redlin, L.; and Watson, S. 2015. *Precalculus: Mathematics for Calculus*. Cengage Learning, 7th edition.