THESIS

POSITIONING OF ANCHORS FOR PERSONAL FALL ARREST SYSTEMS FOR SLOPED ROOFS

Submitted by

Azin Heidari

Department of Construction Management

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2019

Master's Committee:

Advisor: Dr. Svetlana Olbina
Co-Advisor: Dr. Scott Glick

Dr. Rebecca A. Atadero

ABSTRACT

POSITIONING OF ANCHORS FOR PERSONAL FALL ARREST SYSTEMS

FOR SLOPED ROOFS

Construction worker falls account for about one-third of all construction fatalities increasing 53% from 2011 to 2015 with most fatalities in the roofing trade. A personal fall arrest system (PFAS) is an effective means of fall protection and required by the Occupational Safety and Health Administration (OSHA). However, PFAS anchor point placement is an issue evidenced by the number of fatalities caused by incorrect anchor positioning (Hinze and Olbina, 2008).

This research looks at a process to create a tool to optimize PFAS anchor points by: 1) Converting OSHA and American National Standards Institute (ANSI) regulations and standards for anchor point positions into computer-readable format; and 2) Developing a tool for optimization of locations of anchor points. A qualified field user performs the tool data input. Data include PFAS features (e.g. lanyard length) and project-related values (e.g. roof height). The tool then looks for the potential anchor locations that satisfy the fall clearance and swing hazard requirements. K-Nearest Neighbor Search (KNNS) algorithm was used as the optimization model for the tool. The tool was developed in Python programming language and was compiled into a standalone computer application with a user-friendly interface. The output of the tool includes optimized anchor points displayed both graphically and numerically. The tool results were validated using the K-fold Cross-Validation method and proved the tool output results to be adequately accurate. The contribution of this research is the development of an automated field-level process for steep sloped roofing companies that would help improve their safety practices.

DEDICATION

To my family and especially my husband
Thank you for all the love, patience, and support you have given me

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Background

Construction sites are dangerous places, statistics show that out of 4,693 worker fatalities in private sector in 2016, 991 or 21.1% were in construction, in other words, one in five worker deaths were in construction (OSHA, 2016). In 2015, construction industry accounted for the largest proportion of occupational fall fatalities in the USA, that is, 367 fatal falls out of total 800 for all industries (Figure 1) (CPWR, 2017). Construction accidents in the private sector (excluding highway collisions) were responsible for about 64.2% the construction worker deaths in 2015, according to the Bureau of Labor Statistics (2015). Four major causes of fatalities in construction were reported by Occupational Safety and Health Administration (OSHA) (2015b) for 2015:

1) Falls – 364 (38.8%)
2) Struck by Object - 90 (9.6%)
3) Electrocutions - 81 (8.6%)
4) Caught-in/between- 67 (7.2%)

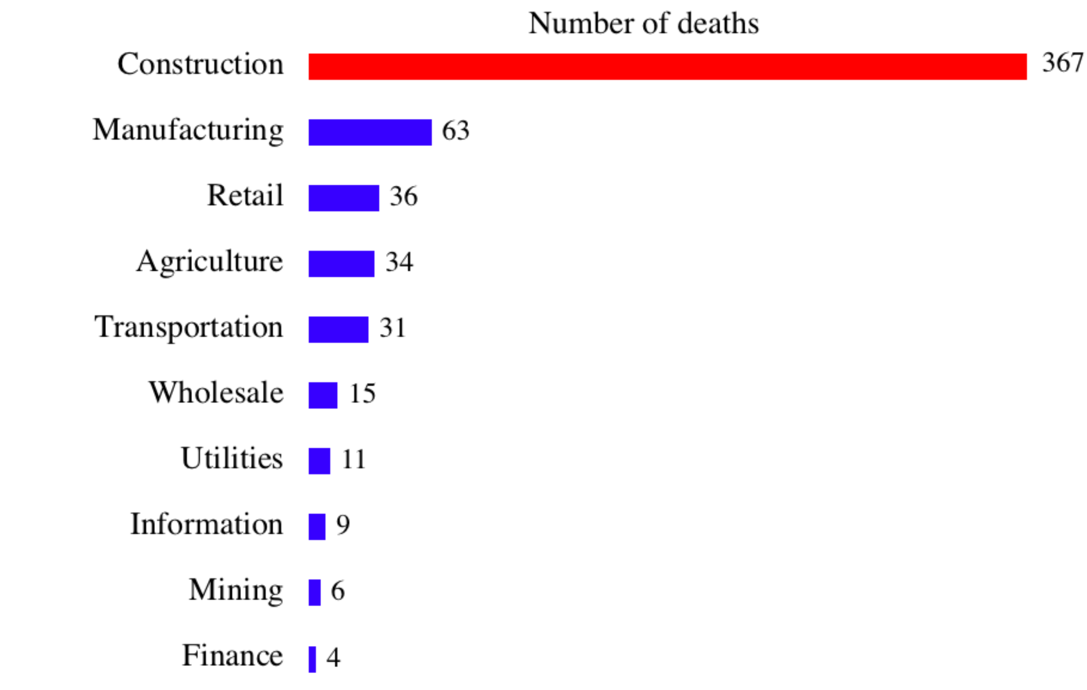

Figure 1: Number of fatal fall injuries by major industry, 2015 (All employment) (CPWR, 2017)

Additionally, the most violated safety standards are identified in order to determine the reasons associated with each of the hazard categories. The OSHA standards that were most frequently violated in 2016 included:

1) Fall protection, construction (29 CFR 1926.501)
2) Hazard communication standard, general industry (29 CFR 1910.1200)
3) Scaffolding, general requirements, construction (29 CFR 1926.451)
4) Respiratory protection, general industry (29 CFR 1910.134)
5) Control of hazardous energy (lockout/tag out), general industry (29 CFR 1910.147)
6) Powered industrial trucks, general industry (29 CFR 1910.178)
7) Ladders, construction (29 CFR 1926.1053)
8) Machinery and Machine Guarding, general requirements (29 CFR 1910.212)

9) Electrical, wiring methods, components and equipment, general industry (29 CFR 1910.305)

10) Electrical systems design, general requirements, general industry (29 CFR 1910.303)

Most of the violated standards pertain to fall protection in construction, and three out of the ten most violated standards are in the construction industry. The 29 CFR 1926.501 is the most cited regulation for noncompliance with fall protection standards and it includes two major clauses:

- OSHA 1926.501(a)(2) “The employer shall determine if the walking/working surfaces on which its employees are to work have the strength and structural integrity to support the employees safely. Employees shall be allowed to work on those surfaces only when the surfaces have the requisite strength and structural integrity”.
- OSHA 1926.501(b)(1) “Unprotected sides and edges. Each employee on a walking/working surface (horizontal and vertical surface) with an unprotected side or edge which is 6 feet (1.8 m) or more above a lower level shall be protected from falling by the use of guardrail systems, safety net systems, or personal fall arrest systems (PFAS).”

The terms that are specifically important in both clauses are structural integrity of the working surface and protecting edges 6 feet or more above the lower level. In other words, with regard to fall protection, two primary conditions of structural integrity of working surface and possessing fall protection gear must be met.

According to the Center for Construction Research and Training (CPWR, 2013), the risk of falling at the workplace depends on the type of construction trade. Roughly 60% of the construction workers work in trades that require working at heights at least once a month and many of these workers use ladders or scaffolds during half of their working hours. Many of construction trades such as roofer, drywall installer and ironworker require the worker to keep their balance on the working platform while performing the work. Roofers are the trade with the highest rate of fatal falls (Figure 2) as well as nonfatal falls (CPWR, 2017).

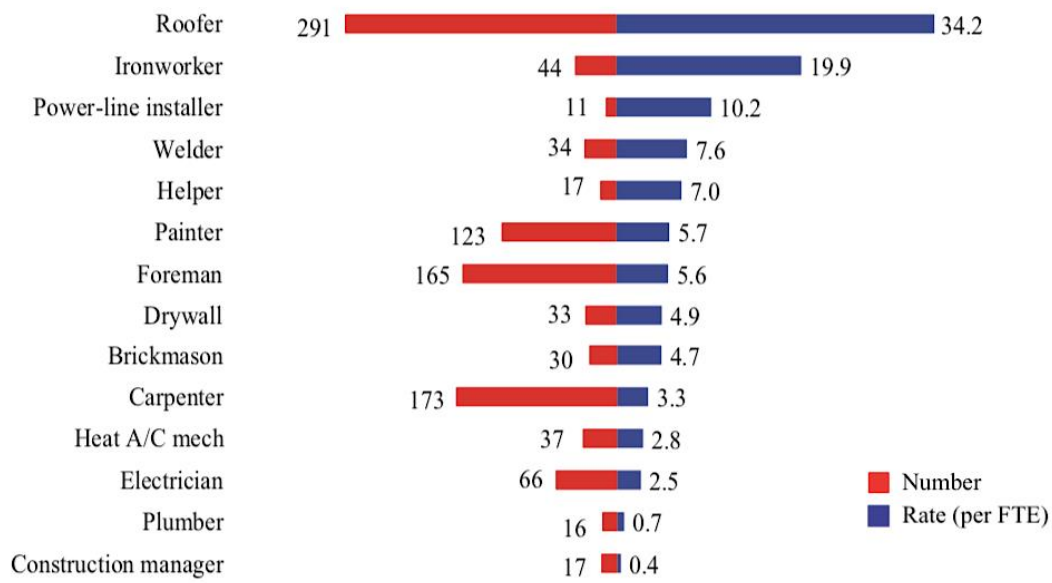

Figure 2: Number and rate of work-related fatalities from falls to a lower level in construction, selected occupations, sum of 2011-2015 (CPWR, 2017)

CPWR (2013) reports the distribution of fatalities from falls in construction by establishment size between 2008 and 2010 for both wage-and-salary workers. The 521 annual fatalities are categorized into five company sizes based on the number of employees. Based on this report, the greatest proportion (i.e. 54.7%) of the fatalities occurred in companies with 1-10 employees in the USA. The fall rate declines with the increase of the size of the company.

In conclusion, one of the major leading causes of deaths in the construction industry is falling from height and roofers are the trade most exposed to this risk. Additionally, small-size companies are ones that are exposed the most to a fall from height risk.

1.2. Problem statement

The initial chance of roofer falling to a lower level increases in the case of sloped roofs, in other words the larger the roof slope, the greater the risk of fall. There are two categories established with regard to the roof slope, low-slope roof (i.e. slope of 4 to 12 or less) and steep roof (i.e. slope of greater than 4 to 12). PFAS is required to be used for both slope categories by OSHA. One would expect that this system contributes greatly to mitigating the risk of fall for small residential projects, however, according to statistics, that is not the case. This is mainly because the use of PFAS by the residential industry is not yet fully embraced; discussed in detail in Chapter 2. An example of a common misconception is, most residential companies believe that the use of PFAS is financially burdensome and interferes with workers' activities. This mentality typically results in either the incorrect PFAS design by unqualified individuals or violating of the requirement of having fall protection.

A literature review of the fall arrest standards, showed that in the case of non-certified anchor points (i.e. the non-engineered selection of anchor points which occurs in most residential projects due to small project size) a qualified person (i.e. professional engineer (P.E.)) is required to design the system. Once designed, a competent person (i.e. safety manager/inspector) or the qualified person (P.E.) supervises the installation and maintenance of the anchor point. Nevertheless, in most of the residential projects, a qualified person whose sole responsibility is to design and supervise the installation of the PFAS is not present on the jobsite. Therefore, the PFAS ends up being installed by individuals with little or no professional training. This typically results in the improper installation of PFAS and raises a number of safety concerns. These concerns include but are not limited to: swing fall hazards, workers not moving anchor points as work progresses, not installing sufficient anchors, not installing anchors per manufacturer instructions, and improper slack on the rope. In essence, the majority of such problems stem from the poor selection of anchor points. These problems point out the need for developing a tool to help with the selection of anchor points in order to reduce the reliance on individuals as well as semi-automating the process of designing PFAS to make it financially feasible and simple to use, especially in smaller companies that may have limited resources.

1.3. Research Objectives

The research goal was to develop a semi-automated process and a computer tool for the optimization of anchor points based on the ANSI and OSHA regulations. This study aimed to achieve five research objectives.

1) Defining a set of rules for positioning the anchor points for PFAS.
2) Examining the rules for various scenarios with regard to different projects.
3) Developing an optimization model for making decisions about anchorage positioning.
4) Automating the optimization process by developing a windows-based computer tool.
5) Validating the tool results using the validation method specific to the optimization model.

In order to achieve the objectives of this study some questions had to be answered in order to narrow down the research scope which is defined in section 1.5. For example, what are the key standards and instructions in designing a PFAS and which of them is examined in this study? This question is answered in Chapter 4 Section 1. What are the scenarios that might not be covered by this tool and how the user can clearly distinguish them? This question is answered with the delimitations that the author set for this research.

1.4. Research Contributions

This study's contribution is improving the safety practices for residential roofing construction. The study developed a computer-based tool that integrates a semi-engineered procedure for selection of anchor points. The tool can be used to decide: 1) the type of deceleration device that must be used based on PFAS's specifications, and 2) The optimized locations for the anchorage that satisfies fall clearance and swing hazard requirements. The potential users of the tool are roofing trades working on the sloped roofs, typically steep slopes, in the case of small and mid-sized projects.

1.5. Research Scope and Delimitations

The research scope was narrowed down to address the fall accidents among roofers during different stages of roof construction. As stated in Sections 1.1 Background and 1.2 Problem Statement, there is a great demand for addressing the existing problems and shortcomings associated with PFAS with regard to this type of construction. Residential roofing includes both low slope and steep sloped roofs; however, this study's focus was only on steep sloped roofs.

The delimitation of this research is that only the basic type of PFAS were examined for a single worker with a weight between 130 and 310 pounds equipped with a deceleration device, a full body harness and a lanyard. Other PFAS types (e.g. horizontal lifeline with multiple users PFAS) were not investigated.

CHAPTER 2: LITERATURE REVIEW

In this Chapter, an overview on the fall accidents statistics is presented, as well as the state-of-art research related to PFAS scope, application and shortcomings. PFAS standards and instructions are carefully examined, based upon which ruleset for anchorage positioning is developed by this research. In addition, this chapter presents the application of information technology (IT) and building information modeling (BIM) in construction safety to date. Furthermore, optimization algorithms and their validation methods are discussed in the last section.

2.1. Construction Fall Accidents

According to the reports by National Safety Council (2013) and regulatory impact analysis by OSHA (2013), at least 95 fatalities and 68,000 injuries due to falls from elevations (i.e. covered under CFR, section 29 subpart M) occur every year. That is over 20% of all workplace injuries in the construction industry every year. OSHA (2015b) requires implementation of fall protection measures for work on a sloped roof that has one or more unprotected side or edge 6 feet or more above lower levels. Some of the required fall protection measures are shown in Figure 3.

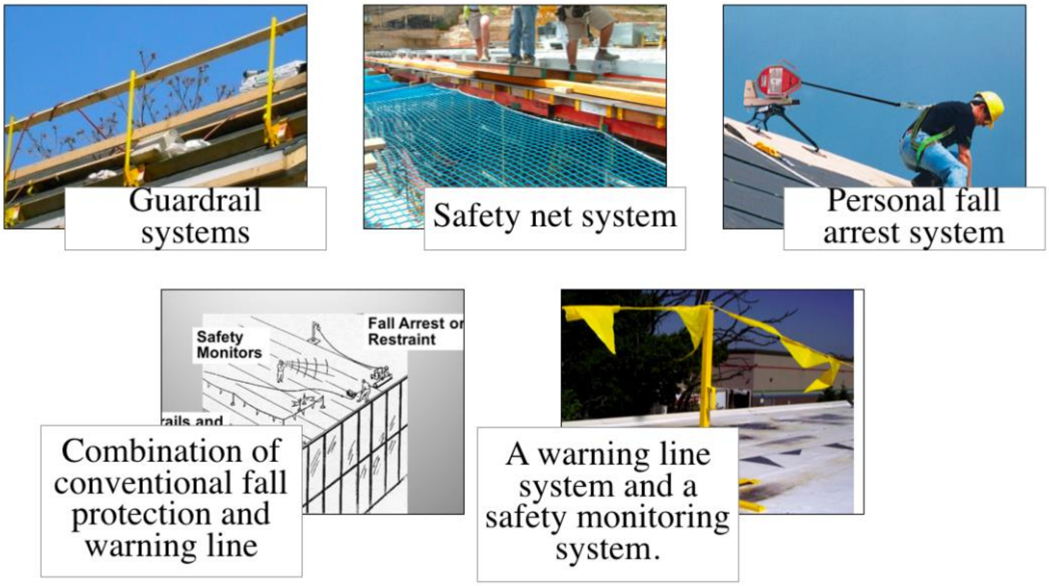

Figure 3: Fall protection measures (OSHA, 2015).

On the other hand, due to the recent evolutions in construction safety (e.g. advanced safety education and recent safety protocols), one might expect the rate of fatalities to decline. However, statistics implied that 364 out of 937 total deaths in construction in 2015 were due to construction falls, which indicates an ascending rate of fatalities (OSHA, 2015). In other words, falls are the leading cause of fatalities in construction in the USA and the rate of fall fatalities is increasing. According to OSHA (2015a), fall accidents are the result of following conditions:

- Unstable surfaces
- Misuse of equipment
- Failure to use safety equipment
- Human error

However, the data on the distribution of fall accidents per accident conditions are not available. CPWR (2013) reported that comprehensive fall accident data for the construction industry are not available on many levels. Falls are generally recorded by OSHA according to the nature of the injury and the surface involved, but these two categories have not been uniformly cross-referenced.

The research by Chi et al. (2005) is based on reports from Council of Labor Affairs of Taiwan and presents a database of fall accidents, categorized by various factors. These individual factors included company size, age of victims, roof slope, location of worker on roof (e.g. edges, surface), roof height, and passive and active prevention measures taken at the time of the accident. The authors reported the ratios between the cause of fall (e.g. inappropriate protection/improper use of Personal Protective Equipment (PPE)) and accident event (e.g. fall from roof edge and fall through roof surface). The authors concluded that the causes of fall and the accident events were directly linked with the specified factors. The authors indicated that the falls from roof edges were associated with bodily actions and being pulled down by a hoist, object or tool while falls through roof surfaces were associated with lack of complying scaffolds.

According to CPWR (2013), the risk of fatal falls depends on the type of construction trade. Between 2008 and 2010, the annual fatal falls for power-line installers were equal to 28.5 per 100,000 full time equivalent (FTEs), while roofers and ironworkers accounted for annual fall fatalities of 23.8 per 100,000 FTEs. However, the average annual falls fatalities of all construction occupations were reported to be 3.2 per 100,000 FTEs, which highlights the high exposure to the fall risk among the three trades (i.e. power-line installers, roofers, and ironworkers). For the nonfatal injuries, ironworkers, sheet metal workers and roofers have an average annual rate of nonfatal falls of 65.5 per 10,000 FTEs, which is higher than 36.2 per 10,000 FTEs for all construction workers. As for the cause of fatalities from falls in all employment categories, CPWR (2017) reported that 33% of the fatalities is due to “falls from the roof”, 24% to “falls from the ladder” and 15% to “falls from the scaffold and staging” (Figure 4). In summary, the major victims of fall incidents are roofers and ironworkers (47.6 deaths per 100,000 FTEs) who are at about five times greater risk compared to other construction trades.

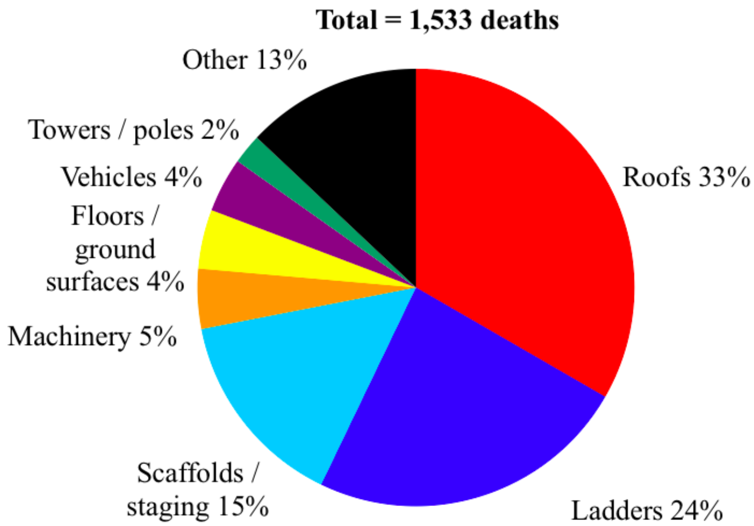

Figure 4: Fatal Falls to a lower level in construction by primary source, sum of 2011-2015 (CPWR, 2017)

Analysis of fall fatalities by the size of establishment showed that small companies (i.e. 1-10 and 11-19 employees) accounted for 55% of fall fatalities from 2008 to 2010. In addition, the safety inspection reports indicated that the majority of standard violations were related to fall protection equipment (Hinze & Huang, 2003). These statistics demonstrate the important impact of fall from roof hazards and the need for implementing more sophisticated fall protection safety measures. Safety is considered a great value for residential contractors because the majority of the contractors aim to prevent fatalities, litigation and extra unforeseen costs imposed on projects resulting from an accident. One reason is the construction contractors do not want their premiums to increase for workers compensation insurance. If a contractor has high Experience Modification Rating (EMR) due to high rate of fatalities and injuries in the previous projects, the insurance companies will impose higher premium rates for worker's compensation policy (Fisk & Reynolds, 2014).

In a research by Huang and Hinze (2003) fall accidents were categorized based on the type of facility being constructed (e.g. single family or duplex dwelling, multi-family dwelling, etc.) as well as nature of the construction effort (e.g. new construction, addition, maintenance, demolition, etc.). The nature of the construction was important in this research since the activities differed. The authors investigated relationships between fall rates and the cost of project and realized that 28.10% of the falls occurred on projects with the total cost under $50,000 followed by 18.80% for projects with the costs between $50,000 to $250,000. These two categories had the largest distribution of falls. In addition, the roofing, siding and sheet metal work categories were ranked first among the top 10 Standard Industrial Classification (SIC) codes associated with the largest number of falls (i.e. code SIC 1761). Additionally, the research by Kang et al. (2017) had a goal to update the statistics presented in the research by Huang & Hinze (2003) with most recent information from OSHA and CPWR reports. Results showed that the frequency of falls had increased by 1.8% from 2003 to 2017 for roofing, siding and sheet metal work.

2.2. Personal Fall Arrest System (PFAS)

2.2.1. PFAS scope, application and limitations

There are two types of fall safety equipment: 1) Fall arrest (e.g. PFAS which saves a worker already in the process of falling, also referred to as an active strategy) and 2) Fall restraints (e.g. such as safety guardrails and net that prevent access to hazardous locations also referred to as a passive strategy). The research by Kines (2002) was based on national data sets from Denmark and Sweden and carried out performed a descriptive analysis of the fatal and serious injury cases resulting from falls through roofs. Kines (2002) created several datasets of investigation reports such as the location of the incident, time of day, the height of roof and contact surfaces.

Furthermore, Kines investigated whether any active/passive fall protection system was in place at the time of the accident. The research mainly focused on the various time intervals in which the fall accidents happened. Findings demonstrated that falls from height predominantly occurred on farm buildings in the afternoon hours noting that worker's fatigue was a likely contributing factor to these circumstances. Kines (2002) emphasized the need for including Personal Protective Equipment (PPE) status in the injury reports in order to allow the researchers to examine the effectiveness of such equipment.

PFAS are proven to be an effective means of fall protection for workers. A PFAS consists of three key components that work in conjunction with each other (Figure 5).

- Anchorage connectors (i.e. permanent or temporary) (Figure 6)
- Body harness
- Deceleration device (e.g. hock absorbing lanyard or self-retracting lifeline)

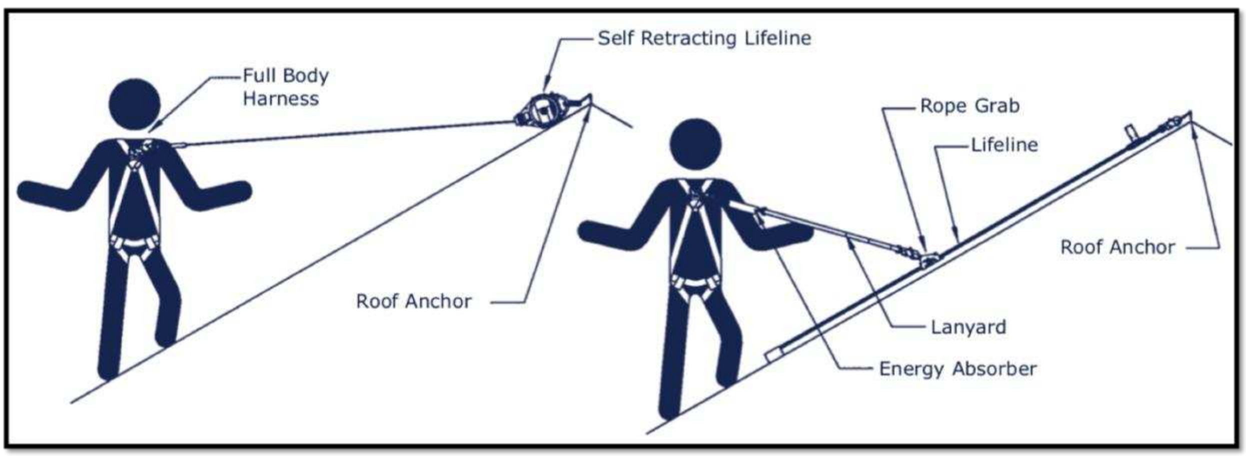

Figure 5: Main components of PFAS (NAHB, 2011).

Figure 6: Example of an anchor (left); roofer tying himself off to a permanent anchor (right) (NAHB, 2011)

Body belts became obsolete in 1990 and are not allowed to be used as fall protection anymore (OSHA, 2015a). ANSI (2016) stated that fall restraint (e.g. guardrails and nets) can only be used in areas with a slope of 0 to 18.4 degrees, equal to or less than a 4/12 slope. In comparison, PFAS can be used for all roof slopes, which resulted in the wide application of PFAS as the fall protection method in sloped-roofing construction.

According to OSHA (2015a), a low slope roof is “a roof having a slope less than or equal to 4/12 (vertical to horizontal)”, and a steep roof as anything greater than 4/12; in other words, a low slope roof angle is from 0 degrees to about 18.4 degrees, and a steep roof angle is from 18.5 degrees on up to 45 degrees or more, as shown in Figure 7. A pitched/sloped roof consists of various components: the structure, the insulating elements, the ventilation of the roof space, the roof covering, the water drainage system and appurtenances.

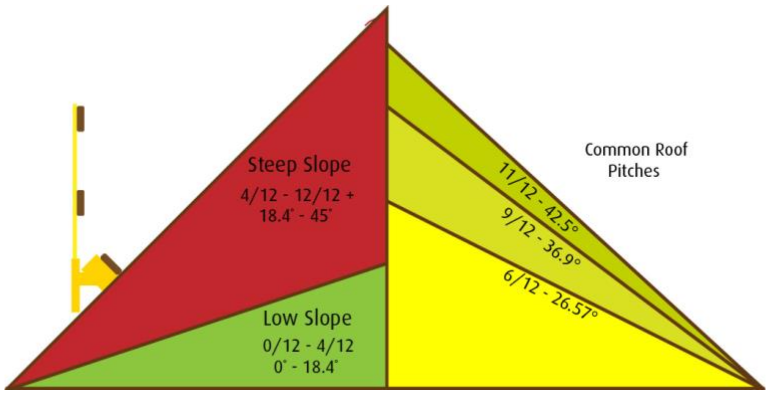

Figure 7: Sloped roof angles. The green triangle range is where the worker would be able to use guardrail systems with warning lines, personal fall arrest systems, safety nets, or a safety monitoring system, and the red triangle range is where guardrails with toe-boards, safety nets and, personal fall arrest systems are the only compliant options available (OSHA, 2015c).

OSHA standard (2015e) relates to roofing work on low-slope roofs and states that when engaged in roofing work on a low-slope roof that has one or more unprotected side or edge 6 feet or more above lower levels, workers must be protected from falling by:

- Guardrail systems,
- Safety net systems,
- Personal fall arrest systems,
- A combination of conventional fall protection systems and warning line systems, or
- A warning line system and a safety monitoring system.

OSHA standard (2015f) on working on steep roofs states that when working on a steep roof that has one or more unprotected side or edge 6 feet or more above lower levels, each worker must be protected by:

- Guardrail systems with toe-boards,
- Safety net systems, or
- Personal fall arrest systems.

Therefore, the use of PFAS is not limited to a specific slope and can be utilized in any project with unprotected edges 6 feet or more above the lower level. In other words, a PFAS is well-oriented toward sloped-roof construction.

However, PFAS have certain shortcomings and limitations as well, such as (OTM, 2016):

- Adds an additional 5000 lbs. static force to the structure in case of fall (Figure 8)
- May result in deformation of PFAS components either after each fall or over lifetime (Figure 8)
- May create hazards of swing, suspension trauma, serious injuries during the fall arrest since worker falls for a short distance before the PFAS stops the fall (Figures 8-9).
- Attachment of anchor points to the roof structure is done by screwing or drilling the anchors into the sheathing, which results in roof-membrane leaks (Figure 8)

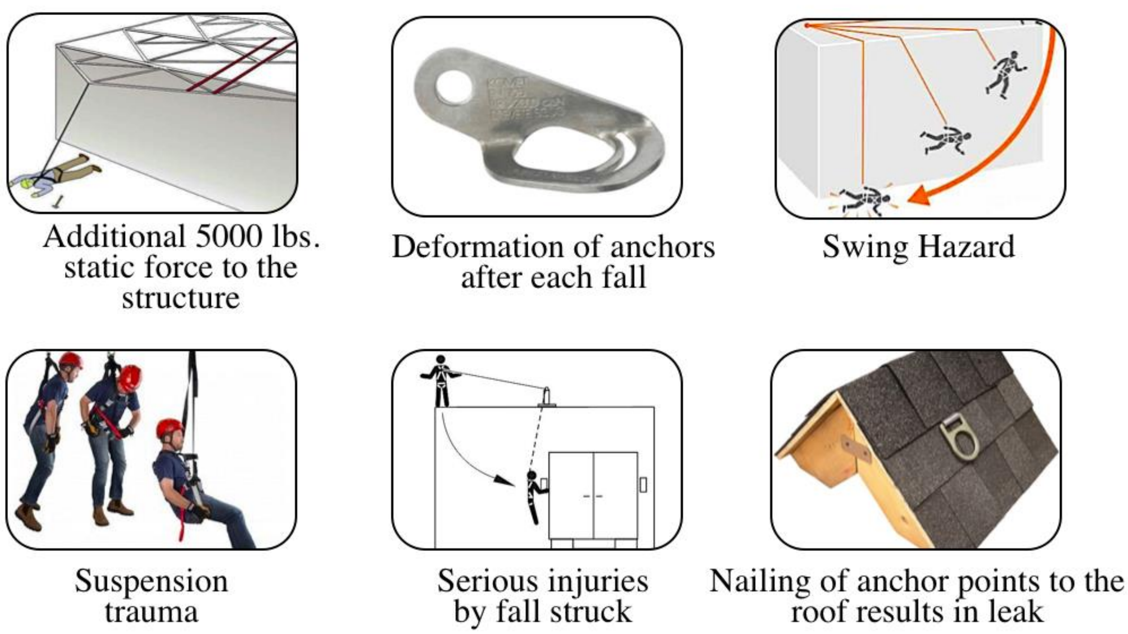

Figure 8: Shortcomings and limitations of PFAS (OTM, 2016)

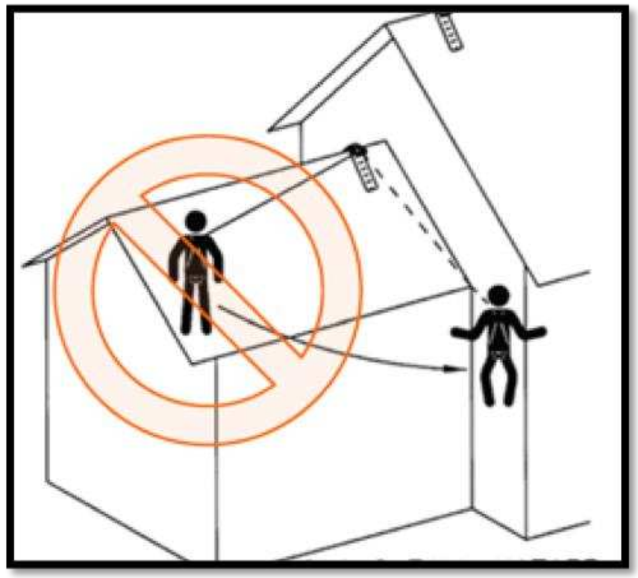

Figure 9: Swing hazard for PFAS anchor point (NRCA, 2017)

The research by Hinze & Olbina (2008) examined the fatalities caused by a PFAS failure to provide protection against fall hazard (i.e. PFAS not available or not properly used). These failures included: 1) No harness worn, 2) Lanyard unhooked (i.e. mobility freedom assumption), 3) Harness improperly worn (e.g. no training provided), 4) Removed harness, 5) Broken lanyard (e.g. deterioration due to age and extensive use), 6) Not 100% tie off (i.e. twin-leg and single leg risks), 7) Malfunction of personal fall arrest system components (e.g. the lanyard hooks are designed to withstand 5000 pounds static force and are not tolerant to side-loads and will break if exposed to such loads), 8) Tied off but killed (i.e. the PFAS does not prevent falls from occurring, but they are designed to restrict the distance that workers fall), 9) Structure collapse and improper anchor points. In addition, National Institute of Occupational Safety and Health (NIOSH) Fatality Assessment and Control Evaluation (FACE) reports by CPWR (2017) confirm the status of PFAS in construction falls mostly as “not present” and “not in use” (Figure 10).

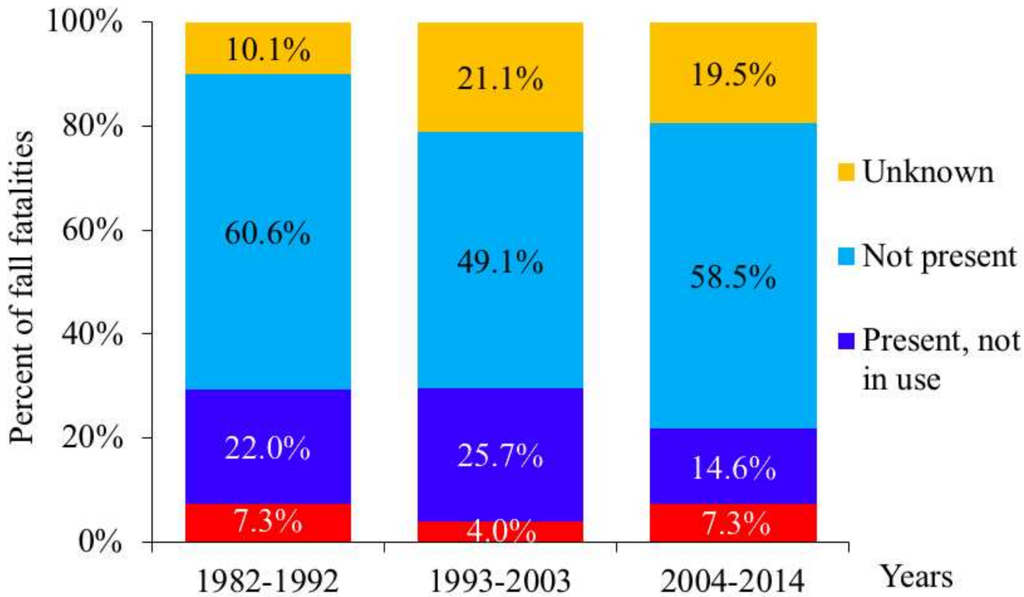

Figure 10: Fatal falls in construction, by Personal Fall Arrest System (PFAS) status, 1982-2014 (CPWR, 2017)

Perry et al. (2015) compared five construction trades including carpenters, electricians, ironworkers, painters, and roofers in terms of their compliance with safety regulations. Their findings indicated that the use of PFAS was 3% lower than applying other safety practices and that carpenters and roofers complied with safety regulations less frequently than other trades resulting in their higher risk of falls. The authors concluded that such reoccurring issues of PFAS noncompliance were due to misuse (e.g. not tying off to an appropriate structure, extending a retractable lifeline too far and negating swing fall clearance) or absence of PFAS.

2.2.2 PFAS in Residential Construction

In the next step, it is important to find out to what extent a PFAS is accepted by the industry. PFAS are widely used in the commercial projects, but they are often absent in residential construction, even when they are available on the site, they are rarely used. One of the technical reasons for it may be that it is highly challenging to find secure anchor points for a fall arrest system, especially during framing and before trusses are set and secured (Kaskutas et al., 2008). Cable (2006) and Stromme (2011) claimed that there is a bias in the residential industry that a 30 minutes exposure to the fall hazard without fall protection, in a low-height project, is reasonable. In addition, Johnson (2011) stated that fall protection for small projects can be financially burdensome. Also, according to Johnson (2011) tying off the worker to PFAS dramatically decreases worker productivity. Dong et al. (2017), using FACE reports from NIOSH, indicated that the ratio of falls from 6-15 feet, in relation to falls from any height, was increasing, from 13.9% for 1982-1992, to 24.0% for 1993-2003 and finally to 48.7% for 2004-2014. Falls from 16-30 feet for construction roofers have approximately the same ratio distribution. In addition, Dong et al. (2017) examined the availability of PFAS for the workers in fatal fall accidents using the FACE database. They concluded that among every four worker fatalities, almost half (i.e. two workers) had no access to PFAS while one worker had access to PFAS but was not using it at the time of accident.

Bethancourt (2011) indicated that there is a misconception of OSHA requirements and as long as the system maintains a safety factor of two, per its anticipated load, the anchor point is allowed to have a tolerance less than 5,000 lb. The author stated that many residential project administrators currently do not use PFAS and consider it unnecessary. Nonetheless, federal and state OSHA standards are the law and cannot be disregarded. Even though some OSHA regulations are between 15 and 40 years old, the ANSI Z.359 standards developed by American Society of Safety Engineers (ASSE) on PFAS are less than two years old and are applicable to all types and sizes of projects.

The research by Haslam et al. (2005) laid out a statistical foundation of analyzing the construction accidents in terms of circumstances under which accidents happened, the casual influences involved, and other causes. A data-collection method was selected that included interviews with the jobsite personnel after accidents. The key factors leading to hazardous situations included problems arising from workers or the work team (70% of accidents), workplace issues (49%), shortcomings with equipment (i.e. Personal Protective Equipment (PPE)) (56%), problems with suitability and condition of materials (27%), and deficiencies with risk management (84%). With regard to the shortcomings with PPE, the authors stated that despite the wide reliance on PPE as a good means of safety control, theoretically, they are often used only when compulsory. The construction stakeholders usually do not obtain such high cost equipment unless it is enforced by law. The findings showed that most of the construction workers find PPE uncomfortable and interfering with their ability to perform the job.

2.3. Anchorage in PFAS

The literature on PFAS suggested that the proper selection of an anchorage point is a vital component of trusting the worker's life on the system. According to Hinze & Olbina (2008), proper selection of roof anchors appears to be the most effective solution in reducing the high number of fatalities related to falls from a roof. However, most of the anchor points used during construction cannot remain on a roof permanently for future use. In many cases, they will be covered by roofing materials or removed at the end of the construction. The authors stated that there is a need for improving the roofers' safety since the existing safety gear for this occupation may be insufficient. Additionally, different roof-construction methods are being used in European countries to reduce risks of fall (e.g. hooks and ladder).

Kramer (2010) stated that safety professionals, when selecting an anchor point, should look for the appropriate location for clearances and swing fall, in addition to the strength of the anchor point. Free fall distance affects clearances and the maximum arresting force applied to the fallen worker and thus should be given adequate attention. Excessive rope slack, using vent pipes, stacks and topping concrete instead of a concrete slab are also common mistakes in anchorage positioning. If the anchorage is to become part of the building structure, the design of the anchorage should be completed by a qualified person (i.e. professional engineer (P.E.)) per building codes. In addition, on-site testing of the anchorage strength is not required but it may be necessary under some circumstances. Also, the competent person (e.g. safety manager) must ensure that appropriate equipment is used, and that procedures and training are in place.

Anchor points should be positioned with regard to certain parameters (OSHA, 2015a; NRCA, 2011; Kramer, 2010; ANSI, 2016):

1. The anchors must be tied off to the structural elements (rafters, trusses, etc.) that have sufficient structural strength and integrity.
2. The anchors must be located in a way that cause minimum interference with the task being performed; the anchors should not impede the activity and should permit worker to operate efficiently.
3. The anchors should provide maximum accessibility to certain parts of the roof.
4. Minimum/maximum number of anchor points is preferred (i.e. depending on the project).
5. Permanent/temporary locations: possibility of providing suggestions on keeping the anchors in place after construction is complete so that they may be used in future maintenance

6. The anchors must allow for proper rescue plans. In other words, the suspended worker should be easily accessible by ladder or walking surfaces for rescue.

2.4. PFAS Standards and Instructions

There are many rules and regulations developed over the years for PFAS. A list of the most important PFAS rules that researchers reviewed in this study are presented in Appendix 1. In essence, the correct interpretation of rules is of utmost importance regarding PFAS. This can be accomplished by asking a series of questions for exploring the rules further (Table 1).

Table 1: Standard clauses, regulations and recommendation with interpretation process.

OSHA Regulations		**The Research Interpretation Questions**
Clause Number	**Clause Description**	
PFAS manufacturer manual template for anchorage installation by OSHA (2013)	Fall clearance distance should be calculated with regard to Length of Lanyard (LL), Deceleration Distance (DD), Height of Suspended worker (HH) and Safety Factor, the worker should not hit the nearest obstruction in case of fall.	1. How to determine the accurate deceleration distance? What about the corners of the roof where there is hazards of fall from both sides? 2. How to determine that the slack of the rope doesn't exceed the maximum allowable length? 3. How to automatically calculate the deceleration distance for any scenario?
OSHA 29 CFR 1926.502(k)	A written fall protection plan consists of proper work, supervision and rescue procedures, appropriate products to be utilized, and training responsibilities.	1. How can the outcome of this research fit into the fall protection plan?
OSHA subpart M 1926.502(d)(15)	Anchorage used for attachment of personal fall arrest equipment shall be independent of any anchorage being used to support or suspend platforms and capable of supporting at least 5,000 pounds (22.2 KN) per employee attached.	1. How to determine the capability of the anchorage for supporting 5,000 forces? 2. How to determine how many workers are attached to a single anchor point? 3. How to test the anchor points' strength?

OSHA 29 CFR 1926.32 subparts (f) and (m)	Perform a hazard analysis to determine areas of risk, identify existing and predictable hazards in the surroundings or working conditions which are unsanitary, hazardous or dangerous to employees by a P.E or Qualified engineer.	1. What are the criteria for identification of a hazardous area? 2. Is there a checklist used by P.E. or Qualified engineer that could be used to automatically identify hazards? 3. What to extract from this analysis?
OSHA subpart M 1926.502(d)(4)	Dee-rings and snap hooks shall be proof-tested to a minimum tensile load of 3,600 pounds (16 KN) without cracking, breaking, or taking permanent deformation.	1. Is this regulation relevant to what this research covers? 2. How to assign specific requirements for different types of PFAS products?
OSHA subpart M 1926.502(d)(15)(i)	Anchorage shall be designed, installed, and used as part of a complete PFAS which maintains a safety factor of at least two.	1. How to maintain the safety factor? 2. Some references stated a safety factor of at least 3, What is the difference?
OSHA subpart M 1926.502(d)(16)	PFAS, when stopping a fall, shall limit maximum arresting force on an employee to 900 pounds (4 KN) when used with a body belt OR limit maximum arresting force on an employee to 1,800 pounds (8 KN) when used with a body harness.	1. How to choose between the two types of body device? 2. How to measure the maximum arrest force of the PFAS in a certain scenario?
OSHA subpart M 1926.502(d)(16)(iii)	PFAS, when stopping a fall, shall be rigged such that an employee can neither free fall more than 6 feet (1.8 m), nor contact any lower level.	1. How to measure this limit in a certain scenario? 2. How to address the possibility of hitting the exterior walls in case of fall?
OSHA subpart M 1926.502(d)(16)(iv) & 1926.502(d)(16)(v)	PFAS must bring an employee to a complete stop and limit maximum deceleration distance an employee travels to 3.5 feet (1.07 m); and have sufficient strength to withstand twice the potential impact energy of an employee free falling a distance of 6 feet (1.8 m), or the free fall distance permitted by the system, whichever is less.	1. How can free fall be controlled in different scenarios? Is it possible to consider a range of possible free fall distance for various cases?
OSHA subpart M 1926.502(d)(23)	PFAS shall not be attached to guardrail systems, Standard Railings, Ladders/Rungs, Scaffolding, Light fixtures, Conduit or Plumbing, Ductwork or Pipe Vents, Wiring Harnesses, Rebar, Lanyards, Vents, Fans, Roof Stacks or any item or structure not capable of meeting OSHA structural load requirements.	1. How to detect and exclude the elements that are not allowed for anchor points in a semi-structured procedure? 2. How to include all possible elements? What are the other allowable elements for anchorage, aside from beams?

OSHA subpart M 1926.502(d)(23) & 1926.502(d)(24)	PFAS shall not be attached to hoists except as specified in other subparts of this Part. When a personal fall arrest system is used at hoist areas, it shall be rigged to allow the movement of the employee only as far as the edge of the walking/working surface.	1. Should the hoists be included in the selection process?
OSHA 1926 Subpart M App C	The anchorage should be rigid and should not have a deflection greater than 0.04 inches (1 mm) when a force of 2,250 pounds (10 KN) is applied.	1. How to determine the deflection of the beam? 2. Which building code should be used for this purpose?
OSHA 1926 Subpart M App C	The frequency response of the load measuring instrumentation should be 500 Hz.	
OSHA 1926 Subpart M App C	During the testing of all systems, a test weight of 300 pounds plus or minus 5 pounds (135 kg plus or minus 2.5 kg) should be used. The test weight used in the strength and force tests should be a rigid, metal, cylindrical or torso-shaped object with a girth of 38 inches plus or minus 4 inches (96 cm plus or minus 10 cm). The test weight should fall without interference, obstruction, or hitting the floor or ground during the test.	
OSHA 1926 Subpart M App C	For lanyard systems, the lanyard length should be 6 feet plus or minus 2 inches (1.83 m plus or minus 5 cm) as measured from the fixed anchorage to the attachment on the body belt or body harness.	1. How to maintain the lanyard length in jobsite conditions? 2. What about other types of PFAS other than lanyard system?
OSHA 1926 Subpart M App C	For rope-grab-type deceleration systems, the length of the lifeline above the centerline of the grabbing mechanism to the lifeline's anchorage point should not exceed 2 feet (0.61 m).	
OSHA 1926 Subpart M App C	A system fails the force test if the recorded maximum arresting force exceeds 1,260 pounds (5.6 KN) when using a body belt, and/or exceeds 2,520 pounds (11.2 KN) when using a body harness.	1. How to measure the arresting force for every scenario?
OSHA 1926.502(d) (20)	When PFAS is used, the employer must assure that employees can be promptly rescued or can rescue themselves should a fall occur. The availability of rescue personnel, ladders or another rescue equipment should be evaluated. In some situations, equipment which allows employees to rescue themselves after the fall has been arrested may be desirable, such as devices which have descent capability.	1. How to anticipate a rescue plan for every scenario? 2. Does the rescue plan affect the selection of anchor points?

A list of standards developed for PFAS by ANSI/ASSE includes:

- ANSI/ASSE Z359.0-2012 Definitions & Nomenclature Used for Fall Protection & Fall Arrest
- ANSI/ASSE Z359.1-2007 Safety Requirements for Personal Fall Arrest Systems, Subsystems & Components
- ANSI/ASSE Z359.6-2016, Specifications and Design Requirements for Active Fall Protection Systems
- ANSI/ASSE Z359.7-2011, Qualification and Verification Testing of Fall Protection Products
- ANSI/ASSE Z359.12-2009, Connecting Components for Personal Fall Arrest System
- ANSI/ASSE Z359.13-2013, Personal Energy Absorbers and Energy Absorbing Lanyards
- ANSI/ASSE Z359.14-2014, Self-Retracting Devices for Fall Arrest and Rescue Systems
- ANSI/ASSE Z359.15-2014, Safety Requirements for Single Anchor Lifelines and Fall Arresters for Personal Fall Arrest and Rescue

In order to investigate fall protection-related rules, the questions why/where/when falls happen should be answered. Table 2 is the result of careful examination of the most-cited CFR code (OSHA, 2015b) that best answers such questions.

Table 2: Standard clauses from Title 29 Code of Federal Regulations (CFR) Subpart M - Fall Protection, 29 CFR 1926.501 (OSHA 2016)

OSHA Standard Number	Fall Protection Area	Requirements
1926.501(a)(2)	Walking/working surfaces on which the employees are to work	The employer shall determine if the walking/working surface have the structural integrity to support employees safely

1926.501(b)(1)	Unprotected sides and edges in a horizontal/vertical working surface which is 6 feet (1.8 m) or more above a lower level	Guardrail systems, safety net systems, or personal fall arrest systems
1926.501(b)(2)(i)	Worker constructing a leading-edge 6 feet (1.8 m) or more above lower levels	Guardrail systems, safety net systems, or personal fall arrest systems
1926.501(b)(2)(ii)	Worker who is not engaged in the leading-edge work, but works where leading edges are under construction 6 feet (1.8 m) or above	Guardrail systems, safety net systems, or personal fall arrest systems
1926.501(b)(3)	Employee in a hoist area shall be protected from falling 6 feet (1.8 m) or more to lower levels	Guardrail systems or personal fall arrest systems
1926.501(b)(4)(i)	Holes (including skylights) more than 6 feet (1.8 m) above lower levels	Personal fall arrest systems, covers, or guardrail systems erected around such holes
1926.501(b)(4)(iii)	Employee on a walking/working surface exposed to objects falling through holes (including skylights)	Covers
1926.501(b)(5)	Each employee on the face of formwork or reinforcing steel, 6 feet (1.8 m) or more to lower levels	Personal fall arrest systems, safety net systems, or positioning device systems
1926.501(b)(6)	Ramps, runways, and other walkways 6 feet (1.8 m) or more to lower levels	Guardrail systems
1926.501(b)(7)(i) 1926.501(b)(7)(ii)	Employee at the edge of an excavation 6 feet (1.8 m) or more in depth edge of a well, pit, shaft, and similar excavation 6 feet (1.8 m) or more in depth	Guardrail systems, fences, or barricades when the excavations are not readily seen because of plant growth or another visual barrier
1926.501(b)(8)(i)	Employee less than 6 feet (1.8 m) above dangerous equipment	Guardrail systems or by equipment guards
1926.501(b)(9)(i) 1926.501(b)(9)(ii)	Employee performing overhand bricklaying and related work or reaching more than 10 inches (25 cm) below the level of the walking/working surface on which they are working, 6 feet (1.8 m) or more above lower levels	Guardrail systems, safety net systems, personal fall arrest systems, or shall work in a controlled access zone
1926.501(b)(11)	Each employee on a steep roof with unprotected sides and edges 6 feet (1.8 m) or more above lower levels	Guardrail systems with toe-boards, safety net systems, or personal fall arrest systems
1926.501(b)(12)	Employee engaged in the erection of precast concrete members (including, but not limited	Guardrail systems, safety net systems, or personal fall arrest systems

	to the erection of wall panels, columns, beams, and floor and roof "tees") and related operations such as grouting of precast concrete members, who is 6 feet (1.8 m) or more above lower levels	
1926.501(b)(13)	Employee engaged in residential construction activities 6 feet (1.8 m) or more above lower levels	Guardrail systems, safety net system, or personal fall arrest system
1926.501(b)(14)	Employee working on, at, above, or near wall openings (including those with chutes attached) where the outside bottom edge of the wall opening is 6 feet (1.8 m) or more above lower levels and the inside bottom edge of the wall opening is less than 39 inches (1.0 m) above the walking/working surface	Guardrail system, a safety net system, or a personal fall arrest system
1926.501(c)	Employee is exposed to falling objects	Wear a hard hat or prevention of the possibility of objects falling

2.5. IT and BIM Application in Construction Safety

Information Technology (IT) is aiding construction researchers in keeping up with the new processes, technology and devices used today in the construction industry. A discussion on the state-of-art safety research with regard to Building Information Modeling (BIM) and IT appeared necessary, especially since every attempt in the past to improve compliance with safety standards has had some lessons to learn from.

Malekitabar et al. (2016) stated that the inherent uncertainty in construction projects should be addressed with the latest technology. The authors found that the state-of-art BIM safety tools lacked either generality or integrity. In other words, the existing safety tools were either incomplete or could not fully integrate their results into BIM models. The authors analyzed 363 incidents and based on this analysis, introduced five sets of safety risk drivers. They proved that the drivers imrpoved the identification of potential hazards up to 40%. Malekitabar et al. (2016) used Python (i.e. an object-oriented programming language) to make the risk drivers quantifiable for BIM software.

Wang et al. (2015) emphasized the importance of resolving fall and turn-over hazards on construction sites. The authors stated that a lot of heavy machinery turn-over accidents happen in the excavation phase, where there are many uneven surfaces and potential edges for fall. The authors developed a database of safety standards related to excavation using OSHA regulations. Their objective was to find ways to terminate the potential causes of turn-overs and falls prior to proceeding with excavation. This research focused on semi-automated identification of cave-in and fall hazards associated with working in excavated pits. The findings indicated that based on the assessment of geometric site conditions, a proper guardrail system could be proposed.

In research conducted by Zhang et al. (2013), the authors indicated that a major portion of safety measurements were still primary text-based checklists, rather than smart tools driven by recent technology. The tool developed in this research applies to the construction planning phase as well as the design phase. The objective of the research by Zhang et al. (2013) was to take advantage of BIM's potential for promoting safety measures in construction. The authors presented an automated system for recognition of sitework hazards and suggesting corrective actions. The authors limited the scope of rule-implementation to fall incidents, due to the large share of accidents resulting from the unavailability of PPE for workers' protection. The framework of the rule-checking system was a fusion of Work Breakdown Structure (WBS), BIM model, safety regulations and OSHA's best practices. Potential construction hazards and best solutions were mapped to the linked building objects and their associated rules from OSHA. The idea was that as the model updates, the tool re-runs the results. The authors discussed the context for the best performance of the tool. The rule-interpretation methodology contained the classification of holes with regard to their dimensions. For example, a hole with a diameter of 1.5 meters or more was considered hazardous and required a guardrail. The research investigated different cases of fall protection and categorized the gathered information. The automation also acceped human input in the decision-making process, which eliminated the possible errors in the results.

Solihin & Eastman (2015) developed a framework for defining rules for BIM automation. The authors identified three main issues in defining the set of rules: 1) Key concepts for rules are descriptive in nature at the beginning of this process and, therefore, the rules are complex and interdependent. To address this issue, the rules should be streamlined, and the dependencies should be clarified; 2) It is important to clarify when each individual rule is applicable and under what terms each rule will be satisfied. This is to avoid confusion in the interpretation of overlapping rules by the computer; and 3) Rules should be interpreted into computable Boolean forms (i.e. true or false) if possible.

2.6. KNNS Optimization Algorithm and Validation of the Results

K-nearest neighbors' algorithm (KNNS) is a form of pattern recognition algorithm that uses a non-parametric method for classification of the data. The input consists of k closest training examples in the data space. The KNNS algorithm is among the simplest machine learning algorithms and functions based on the local proximity. This algorithm is the basis of many advanced optimization algorithms such as artificial neural networks, supervised learning and data mining (Altman, 1992). Some other optimization algorithms were investigated for this research as well, such as Nearest Centroid Classifier, Artificial Neural Network (ANN) and Structured prediction. Nearest Centroid Classifier is a machine learning algorithm that computes centroid of each class of data and measures the distance of each point to the centroid of other classes (Dhillon

& Modha, 2002). Since the optimization dataset in this research is considered one single class, the researcher decided this method was not appropriate for this research. ANN consists of a framework for many different machine learning algorithms that work together and process complex data inputs (Graupe, 2013). ANN is a time-consuming and complex method that is unnecessarily complex for this research. Based on the analysis of these different optimization algorithms, the researcher selected KNNS as the best fit for this research because it works based on local proximity measurement and matches the level of computation needed for the optimization in this research.

The KNNS algorithm can be summarized as:

1. A positive integer k-number is specified, along with a new sample.
2. K entries in our database closest to the sample are selected.
3. The distance of the entries is measured and classified (Altman, 1992).

Over the years, many different methods have been developed for validating the results of a KNNS optimization model. Some of these methods are: confusion matrix, K-fold Cross-Validation and likelihood-ratio test (Dasarathy, 1991). Confusion matrix (also known as error matrix) allows the validation of different classes of data using a contingency table (Stehman, 1997). This validation method is proper for the optimization algorithms with more than one class of data, which was not the case in this research. The likelihood-ratio test is a statistical test for computing p-value and is based on the goodness of fit test for an alternative model against a null model (Dasarathy, 1991). This method has an advantage of validating the statistical optimizations; however, this validation model is not a good fit for the geometrical optimizations used in this research.

The K-fold Cross-Validation is commonly used to estimate the skill of a model for a given problem. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. This method is easy to understand and implement. This method uses a resampling procedure to evaluate the results in a limited data sample (Kuhn & Johnson, 2013).

The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias (Cawley and Talbot, 2010).

CHAPTER 3: RESEARCH METHODOLOGY

To achieve the research objectives, this study developed a tool for automating the optimal positioning of anchor points for a PFAS in residential roofing. Table 3 summarizes the research objectives, methods, and the software used as well as the software functionality for each step of the research.

Table 3: Research objectives, methods and software used.

Research Steps	Research Objectives	Research Methods	Software/ Strategy	Software Functionality
1	Defining a set of rules for positioning the anchor points in PFAS	Examining the ANSI standards and OSHA regulations and converting them into machine-readable rules	N/A	N/A
2	Examining the rules for various scenarios with regard to different projects	Rule-checking of fall clearance, fall swing, accessibility and structural strength	Manual Calculations	Developing a ruleset to be used as a database for the tool
3	Developing an optimization model for decision-making	Determining the decision matrix with regard to 1) decision variables 2) constraints 3) objectives	KNNS Optimization Algorithm	N/A

4	Automating the positioning of anchor points in form of a Windows-based computer application.	Developing an independent tool using computer programming	Python and Various Python Modules such as Numpy, TkInter, Sympy, Math and Pil	Developing a computer program with a simple user-friendly interface/GUI
5	Validating the tool results	Determining the proper algorithm for validation of the optimization model and performing the validation	K-fold Cross-Validation	N/A

3.1. Examining OSHA And ASSE/ANSI PFAS Regulations and Standards

The first research step was to translate the OSHA and ANSI regulations and standards into computer-readable formats. Rule clauses were written in a homogenous, simple and clear format and the conditions under which each rule would be satisfied were defined. To avoid ambiguities in computer interpretation, a set of questions regarding the coverage of rules were developed (Table 1).

The purpose of the rule checking was to support the decision-making process; in other words, the involvement of a qualified person (such as a professional engineer (PE)) would still be necessary. The process has an automated structure and engages a competent person (e.g. safety manager) by asking for certain PFAS and project-related values. It is expected that eventually, with getting feedback from tool users, the tool can be fully automated to free qualified or competent persons and enable them to focus on other areas of construction and design.

3.2. Evaluating the Regulations and Standards for Different Scenarios

The second research step examined the ruleset developed in the first step with regard to different scenarios and projects. These rules included fall clearance, structural strength, fall swing and accessibility. In addition, different projects with differing roof plans and PFAS characteristics were used to implement the rules. This phase was a crucial step for finalizing the database for the tool as every rule was thoroughly inspected and refined to have minimum overlap and contradiction with the other rules. As a result, the possible contradictory and overlapping clauses were eliminated. Two sample scenarios that demonstrate the process used in this step are discussed in Chapter 4.

3.3. KNNS Optimization Algorithm

The third research step was to develop an optimization model for the tool's automated decision-making. First, the role of optimization in defining optimal anchor points was determined. For the purpose of this research, an optimization model consisting of three major components was defined (Table 4). At this stage, the decision variables were narrowed down to fall clearance and swing hazard requirements to streamline the optimization process.

1) Decision variables
    - 1.1. Allowable distance of working platform from the roof edge
    - 1.2. Allowable angle of working platform in case of a nearby barrier that might lead to fall swing hazard
2) The constraints
    - 2.1. The roof height
    - 2.2. Distance of the roof edge to the closest lower obstruction
    - 2.3. The PFAS features

2.4. The roof dimensions

2.5. Location of barriers leading to swing hazard

3) The objective

3.1. Finding the optimal locations for positioning anchors

Table 4: The optimization matrix based on a predictive model

	Swing Hazard Angle #1	Swing Hazard Angle #2	Swing Hazard Angle #3	Swing Hazard Angle #4	Swing Hazard Angle #5	Swing Hazard Angle #6	
Fall Clearance CP #1	X1	X2	X3	X5	X6	X7	Accessibility Rank #1
Fall Clearance CP #2	X8	X9	X10	X11	X12	X13	Accessibility Rank #2
Fall Clearance CP #3	X14	X15	X16	X17	X18	X19	Accessibility Rank #3
Fall Clearance CP #4	X20	X21	X22	X23	X24	X25	Accessibility Rank #4
Fall Clearance CP #5	X26	X27	X28	X29	X30	X31	Accessibility Rank #5
Fall Clearance CP #6	X32	X33	X34	X35	X36	X37	Accessibility Rank #6
	Strength Req. Value #1	Strength Req. Value #2	Strength Req. Value #3	Strength Req. Value #4	Strength Req. Value #5	Strength Req. Value #6	

In the next step, an optimization algorithm was selected to act as a filter to the possible anchor point placement that satisfy both the fall clearance and swing hazard requirements. Various optimization algorithms were investigated, and the researcher determined that K-Nearest Neighbor Search (KNNS) algorithm was the best fit for this research. KNNS optimization is a robust and versatile algorithm for machine learning applications (Altman, 1992). KNNS is a form of proximity search, which is a problem of finding the point in a given set of points that is closest to a given point. Closeness is typically expressed in terms of a dissimilarity function: the less similar the objects, the larger the function values (Altman, 1992).

3.4. The Tool's Programming Logic

The fourth research step focused on programming the optimization model in Python 3.6 based on the ruleset developed in the second step of the research. First, the tool was programmed to ask the user for input values. These values included: 1) Building dimensions (e.g. roof height), distance between structural members (e.g. trusses or rafters) and suggested locations for working platform and anchor point, 2) PFAS specifications (e.g. lanyard length, harness stretch, etc.), 3) Worker information (e.g. worker weight and height), and 4) Location of physical barriers. There are also built-in default parameters for the input values that are used by the tool in the case the user chooses the default values. It is important to re-emphasize that this study was delimited to a PFAS designed for a single worker (i.e. with a weight between 130 to 310 pounds) equipped with a deceleration device (i.e. either a self-retracting device or a personal energy absorber) as well as a full body harness and a lanyard.

The basis for the Python programming was the "callback" definition, where a Python class was created to include all the definitions that put together different functions of the tool. The definition named "_init_(self)" was created for the sole purpose of collecting the user input values. The label command defined the text for each entry. The text used for each label included the description, the abbreviation and the unit for each value (e.g. roof height (RH) in feet). The entry command was used to collect the input values. Sixteen entries were defined for this tool. The "printt" definition was created in order to automatically insert the result output in the text pad in the main menu. As a result, the user can simply save the text pad file as a record of the tool run.

The “on_button” definition was in charge of the subsequent calculation of the tool. The “Enter” button corresponds to the “on_button” definition, where the tool runs the optimization for the entered input values. The function creates a matrix of points on the roof, inch by inch. Then through the “Euclidean distance” definition, the distance of each point from the input anchor is measured. The result matrix consists of all the points in the user-defined range of possible anchor points. The length of the result matrix varies and was approximately 43,000 points for Scenario 1 explained in Section 4.2.

In the next step, two sets of points, 8,000 in each set, were randomly selected from the result matrix. The fall clearance and swing hazard requirements were tested for each set by conditional statements such as if/else. The qualified points were then divided into two new sets of points. The final results came from the comparison of these two data sets and the mutual points were reported to the user of the tool through Matplotlib module. The Matplotlib module plots the result points in a coordinate plane, where the user can visually compare the results and make the final decision.

The experimental environment for the evaluation of the tool computation time is shown in Table 5. The launch time is less than six seconds and the average computation time is a little over a minute.

Table 5: Experimental Environment

Category	**Specifications**
Operating System	Windows 10
Memory	16 GB
CPU	Intel(R) Core i7-6700 3.40 GHz
Tool Launch Time	5.50 seconds
Average Computation Time	71.81 seconds

Lastly, the Python script was compiled into a standalone executable tool which runs in Windows operating system. The user of the tool would not need to install Python or any of its modules to be able to run the application.

3.4.1. The Tool's Graphical User-Interface (GUI)

Since the application was programmed in Python, it was necessary to compile the Python script into an executable file with a proper GUI so that the user would not need to install Python or its modules. Therefore, the Python script was compiled into a standalone executable tool which is compatible with the Windows operating system. In order to create a simple, user-friendly GUI for the tool and then compile the Python script, TkInter 3.7.1 was used. TkInter is the Python's de-facto standard GUI package. The TkInter advantage over other GUI toolkits is that it is an object-oriented toolkit that is the most compatible with Python and offers a wide array of useful widgets. The tool's TkInter designed GUI includes the main menu and different windows and buttons in the tool. A menu window was created as the dashboard of the tool, containing a command bar, a text widget, and four buttons. The buttons are used to navigate through different windows in the tool. The buttons on the command bar correspond to different events (input processes) of the tool such as creating a new file, saving the text pad and opening the help box.

3.5. Validation of the Optimization Method

The fifth step was to validate the optimization method developed in step three. To validate the results of the KNNS optimization algorithm, the K-fold Cross-Validation method was used.

Even though the basic concept of the K-fold Cross-Validation was used for the validation, the method differs from the original model due to difference between the KNNS original model and the optimization model developed for this tool. The general procedure of validation for this research was as follows:

1) The results dataset was shuffled randomly, and N number of result anchor points were selected.
2) The dataset (N points) was split into k-number of groups.
3) The points in each group were examined for swing hazard and fall clearance requirements.
4) The accuracy of results was examined by comparing the accuracy of the points within a group to the points in the other groups.

This method required developing a rule-checking Python code for each anchor point entry and was done through loops of rule-checking statements, thus automating the process.

CHAPTER 4: RESULTS

4.1. Extracted Rules for PFAS Tool

As discussed in Chapter 3, several standards and regulations from OSHA and ANSI were examined in order to accomplish the first research objective. The researcher extracted a ruleset for fall clearance and swing hazard requirements and calculations for rigid anchor points. As a result, the following list of assumptions and rules was created:

1) Assumptions (see also Figure 17 for graphical representation)
    a. Maximum number of users = 1 user (the tool does not support multi-user PFAS)
    b. Maximum arrest force = 1800 lbs. (F_{CLR})
    c. Maximum arrest load = 5000 lbs.
    d. Stretch-out of the harness = 1 ft.
    e. Safety factor = 2
    f. Maximum deceleration distance: 3-½" ft.
    g. D-ring shift = 1 ft.
    h. D-ring height (H_1) = 5 ft. For 6 ft. tall workers
    i. X_W = straightening of user + harness stretch
    j. Maximum deployment of shock absorber = 42 inches.
    k. Self-retracting device: Fall arrest force:
        i. Class A: 1350 pounds
        ii. Class B: 900 pounds
2) Rules
    a. Selection of the type of the connecting device:

i. If fall clearance is less than 18.5 ft., the connecting device should be self-retracting lanyard.

ii. If fall clearance is more than 18.5 ft. the connecting device should be self-retracting lanyard or personal shock absorber.

b. In case of swing hazard, free fall distance + harness stretch distance + swing drop distance should be less than 4 ft.

c. Personal energy absorber: fall Arrest force (F_{CLR}):

i. In case of a 6 ft. free fall, fall arrest force equals 700 pounds

ii. In case of a 12 ft. free fall, fall arrest force equals 950 pounds

d. Clearance requirement (C_P/C_A): Worst case of free fall distance + deceleration distance + stretch out + swing fall distance (if applicable) + clearance margin (2ft. or 3 ft. to be prudent)

e. Rope slack = length of lanyard + height of D-ring on user's harness

f. Free fall distance must be less than:

i. 6 ft. for shock absorber

ii. 4 ft. without shock absorber

4.2. Two Examples of the Examined Scenarios

The second research objective was accomplished in this step of the research. The rules were implemented for different scenarios of roof and PFAS features. Two examples of such scenarios are presented in Scenario #1 and Scenario #2.

Scenario #1: A 6-foot tall worker is working on a 6.5-foot-high roof and their height from the edge of the roof to the working platform equals one foot. The height of the edge of the roof to the top of the highest allowable obstruction (in this case, ground level) is 13.12 feet (Figure 11). In this scenario the roof slope is 5/12 and the worker's D-ring is below the anchor location. In addition, the swing hazard for 30- and 15-degree angles is assumed to exist.

The roof dimensions are 26 feet by 40 feet and the user weight is 170 pounds.

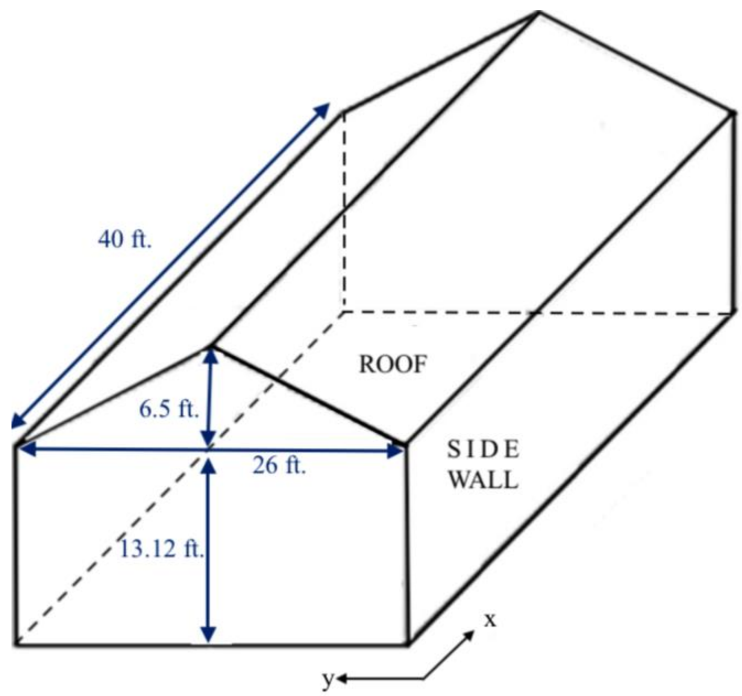

Figure 11: Isometric view of the house in Scenario #1.

Solution: Required clearance below the anchorage (C_A) (Figure 12):

C_A (required clearance below the anchorage) = LY (lanyard length, DV*: 3ft.) + DD (lanyard/lifeline stretch, DV:3.5 ft.) + X_{PEA} (deployment of the connecting device) + X_L (stretch of the lanyard) +MASD (D-ring shift, DV:1 ft.) + HI (back D-ring height, DV: 5 ft.) +X_W (harness stretch, DV: 1ft.) + safety factor (2 ft.)

$C_A = 3 + 3.5 + 1 + 5 + 1 + 2 = 15.5$

*DV=Default Value

According to the standards, 15.5 < 18.5, therefore: the connecting device should be a self-retracting lanyard.

FF (Free Fall) = 3 – (-0.5) = 3.5 ft. and 3.5 < 4, therefore: the free fall distance complies.

The addition of free fall distance, harness stretch distance and swing drop distance must be less than 4 ft.

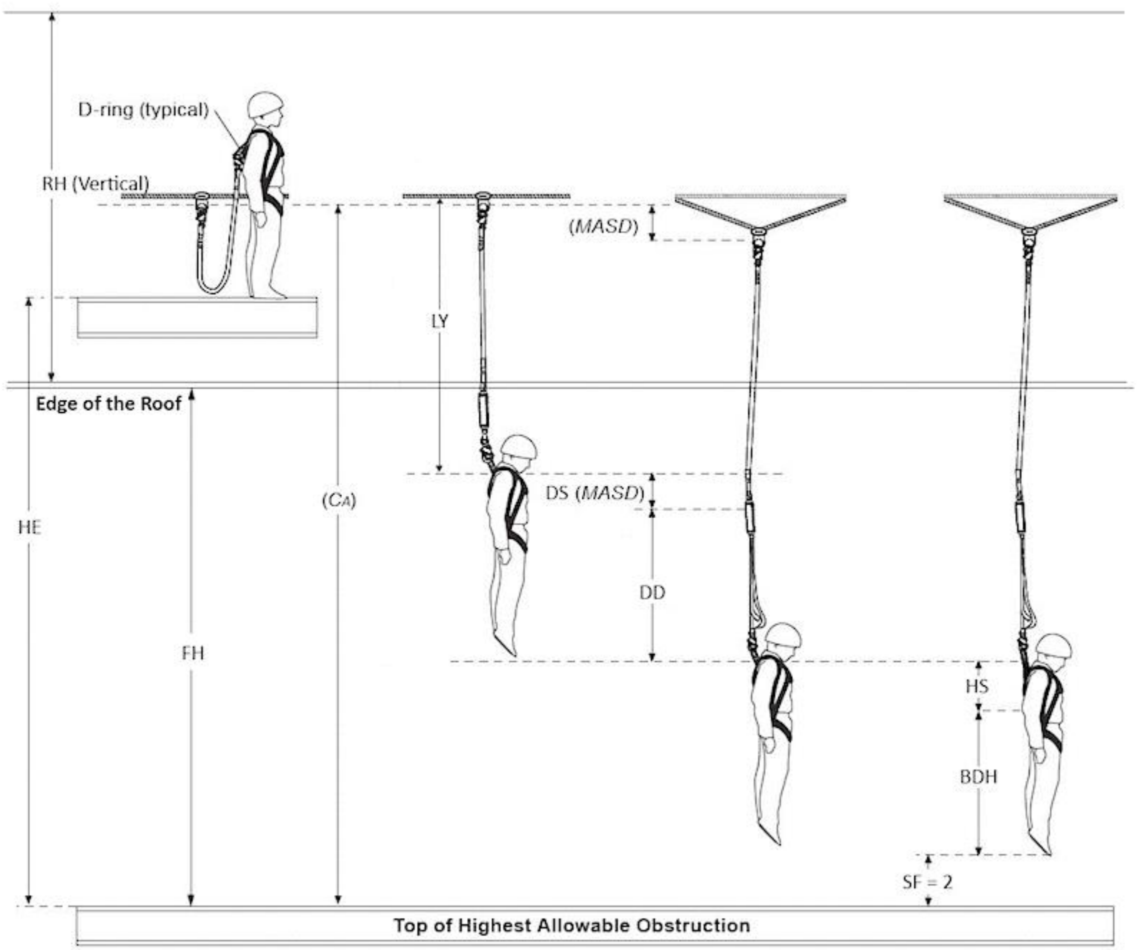

Figure 12: Fall clearance guide (adapted from ANSI 2016)

Since there is swing hazard, the swing fall distance (SFD) should be added to C_A to account for the worst-case scenario (Figure 13).

Q= 3 * sin (30) = 1.5 ft.

Q'= 3 * sin (15) = 0.77 ft.

S = 2 (sin (15) * 3) = 1.55 ft.

SFD= sqrt $[(1.55)^2 - (1.5)^2]$ = 0.39 ft.

S’= 2(sin (7.5) * 3) = 0.78

X= sqrt [$(0.78)^2 - (0.77)^2$] = 0.32

SDD = 0.39 – 0.32 = 0.07

3.5 + 1 + 0.07 = 4.57 ft. and 4.57 > 4 ft. therefore: the angle is NOT safe and there is swing hazard.

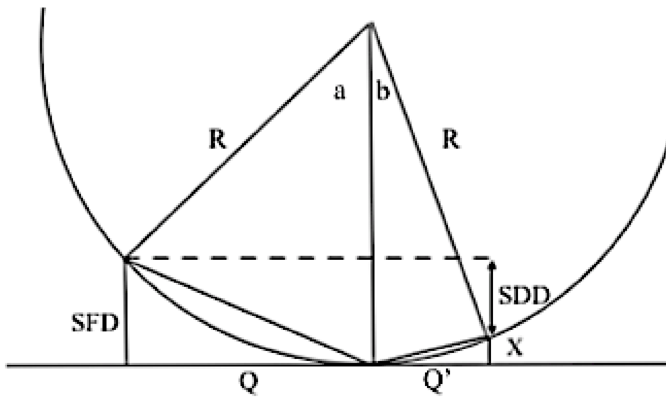

Figure 13: Swing hazard calculations.

Scenario #2: A 6-foot tall worker is working on a 7.5-foot-high roof with a slope of 3.75/12. Their working platform is 4.5 feet lower than anchorage. The lanyard is 5 feet long. The height of the edge of the roof to the top of the highest allowable obstruction is 10 feet (Figure 14). In this scenario, the worker’s D-ring is above the anchor location. In addition, the swing hazard for 60- and 30-degree angles is assumed to exist. The roof dimensions are 48 feet by 60 feet and the user weight is 190 pounds.

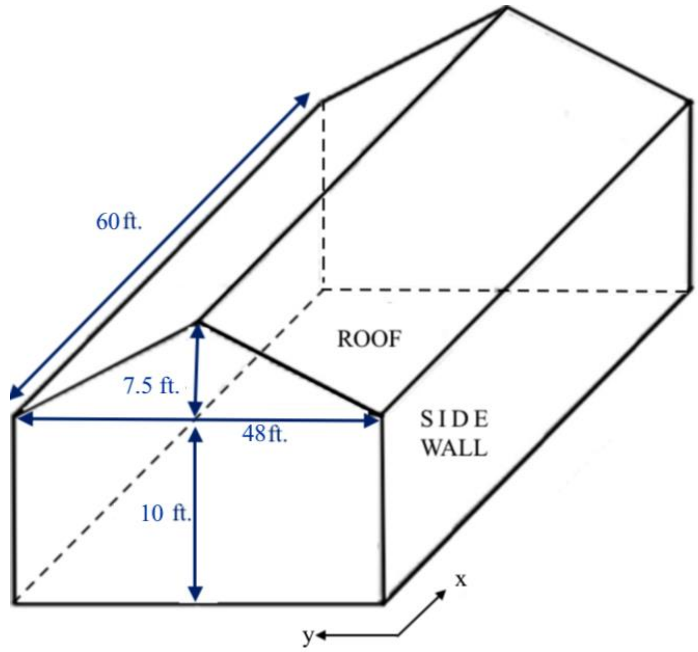

Figure 14: Isometric view of the house in Scenario #2.

Solution: C_A = LY (lanyard length, DV: 3ft.) + DD (lanyard/lifeline stretch, DV:3.5 ft.) + X_{PEA} (deployment of the connecting device) + X_L (stretch of the lanyard) +MASD (D-ring shift, DV:1 ft.) + HI (back D-ring height, DV: 5 ft.) +X_W (harness stretch, DV: 1ft.) + safety factor (2 ft.)

C_A= 5 + 3.5 + 1 + 5 + 1 + 2= 17.5

FF (Free Fall) = 5 – 4.5 + 5 = 5.5 ft. and 5.5 < 6, therefore: the free fall distance complies with the PFAS standards.

A retractable device with a personal shock absorber should be used as the connecting device.

Q = 5*sin (60) = 4.33

Q' = 5*sin (30) = 2.5

S = 2 (sin (30) * 5) = 5

SFD= sqrt [$(5)^2 - (4.33)^2$] = 2.5 ft.

S'= 2(sin (15) * 5) = 2.58

X= sqrt [$(2.58)^2 - (2.5)^2$] = 0.63

SDD = 2.5 – 0.63 = 1.87

5.5 + 1 + 1.87 = 8.37 ft. and 8.37 > 4 ft. Therefore, the working angle is not safe and might result in serious injuries in case of a fall.

4.3. Development of the Optimization Model

The KNNS was used to define the optimization problem in the third research step. The goal of a KNNS based optimization is to find a point which minimizes certain objective functions. In other words, a point p is an R-near neighbor of a point q if the distance between p and q is at most R (Figure 15) (Andoni, 2009).

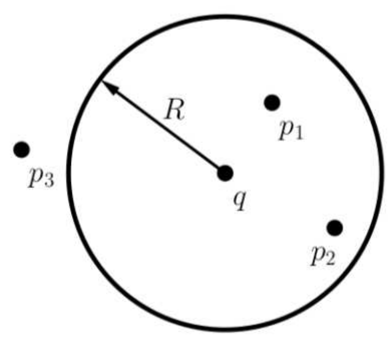

Figure 15: An illustration of an R-near neighbor query. The nearest neighbor of the query point q is the point p1. However, both p1 and p2 are R-near neighbors of q (Andoni, 2009).

The KNNS query problem was established for this research. A set of points were defined with respect to the roof dimensions in inches. For example, for a 50 foot by 12-foot roof, a set of 86,400 points was defined. From this set, N (in this case N=8,000) points were randomly selected. The number of iterations, M (in this case M=2) was selected. In each iteration, all 8,000 points were checked for fall clearance and swing hazard rules compliance. The points that complied with both rules were defined under a new set of points. Therefore, two sets of points (P and Q) were created after two iterations. The length of these sets of points varied since the initial 8000 points were randomly selected. In the end, the two sets of P and Q points were compared with each other and the mutual points were defined under a new set of points that was named S. The final set of points (S) represent the results of the tool.

Additionally, values of N and M could change based on the preference of the tool developer (i.e. the researcher). The researcher selected the values of N and M based on the system specifications and computation time.

In addition to the developing the optimization model, being able to evaluate the model precision and quality is also an important part of optimization. The validation of the optimization results is discussed later in the chapter.

4.4. Programming the Tool

In the fourth step of the research, the tool was programmed in Python based on the programming logic discussed in Chapter 3, for more details see Appendix 2. The result of compiling the Python code was a standalone application that incorporates the fall clearance and swing hazard requirements for the optimization process.

For the purpose of optimization and compilation, Python modules including Tkinter, Numpy, Sympy, Math and Matplotlib were utilized. First, a main menu was created to work as a dashboard for the tool. The main menu window was set to 600 x 500 pixels. The tool's main menu consists of a toolbar with conventional icons (e.g. open file, save file, copy & paste, etc.), a text pad, three buttons (fall clearance, swing hazard and roof geometry guides), and a START button (Figure 16).

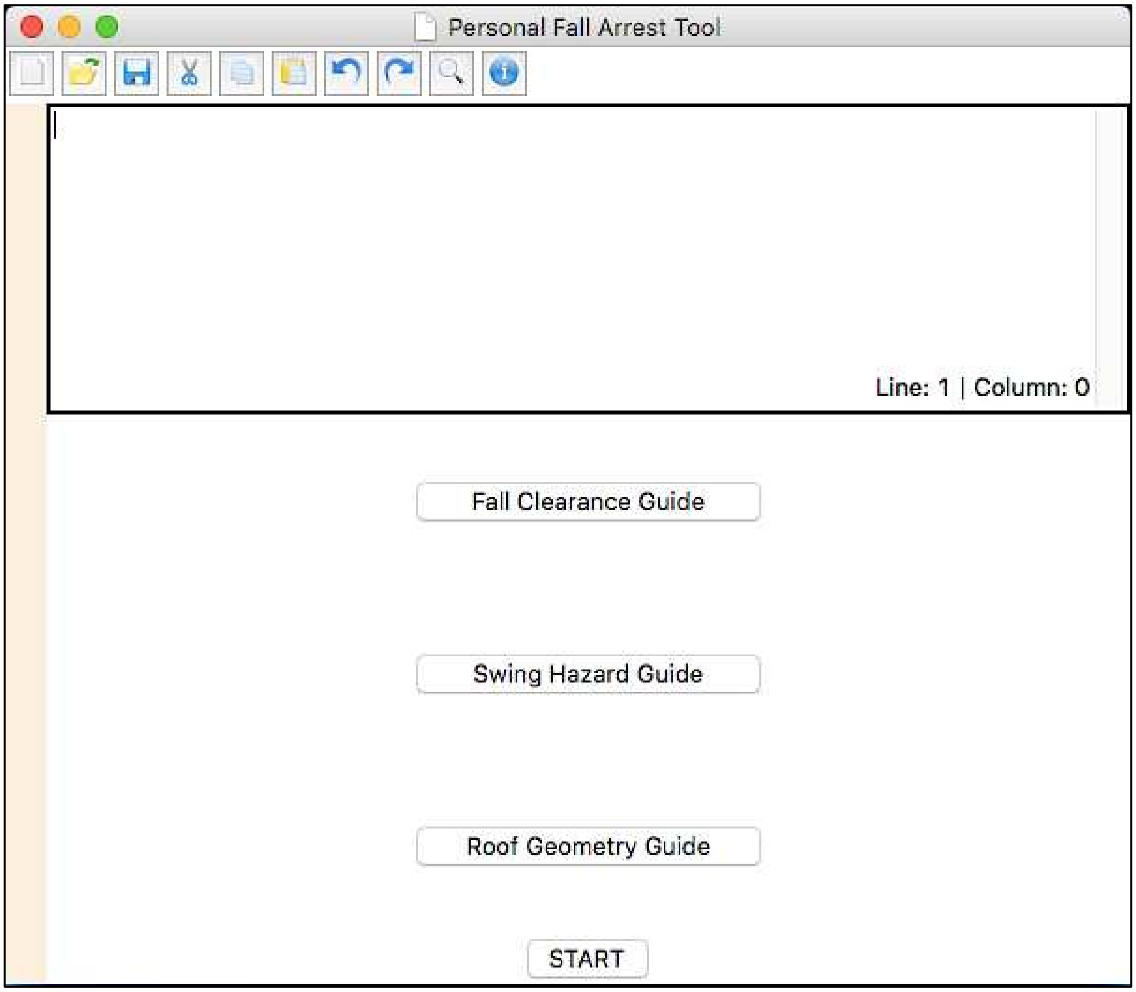

Figure 16: The PFAS tool's user-interface (main menu).

In addition to the main menu, three windows for diagrams were created that correspond to the three buttons on the main menu. To help the user have a better understanding of the input values, two diagrams, adapted from ANSI (2016), were used to present the tool's parameters. The values are slightly different than those used in the ANSI diagrams due to the tool's needs. These diagrams can be accessed by clicking on "Fall Clearance Guide" and "Swing Hazard Guide" buttons on the main menu (Figures 17 and 18).

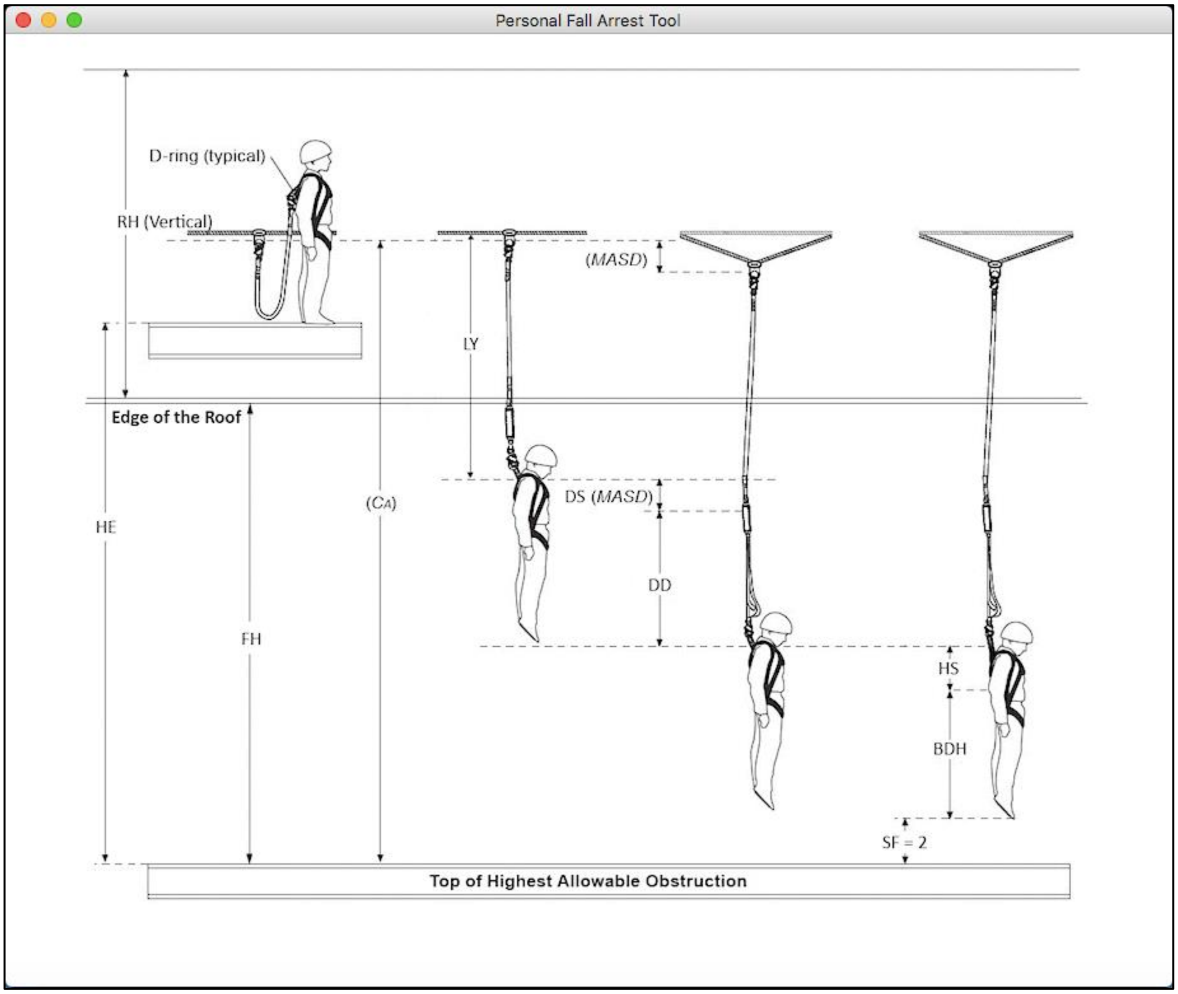

Figure 17: Fall clearance guide (adapted from ANSI 2016).

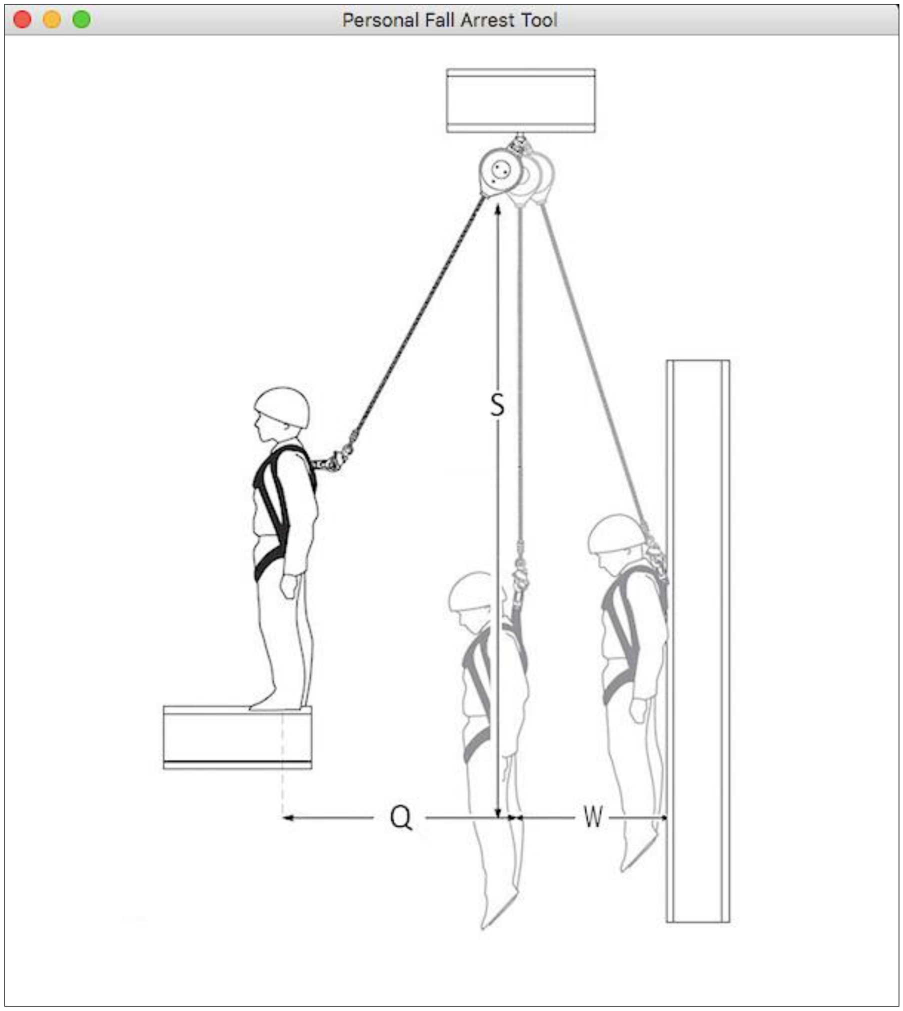

Figure 18: Swing hazard guide (adapted from ANSI 2016).

In addition, the third button corresponds to the “Roof Geometry Guide” diagram that helps the user identify the input values for roof dimensions and in interpreting the results (Figure 19). Roof width, roof length (at possible eave location), roof ridge, roof plane (RP) and the origin point of the coordinate plane are introduced in a 3D view of a gable roof. For example, this diagram informs the user that the roof width corresponds to half of the roof plan width (RW on Figure 19) and the rake-end height, from ridge to bottom of truss system, is used as the vertical axis value (RH on Figure 19), the distance from the ridge down the roof slope to the intersection with a wall where an eave may be placed. RW and RH are used in determining the location of the anchor points on the roof plane (RP), shown in green on Figure 19.

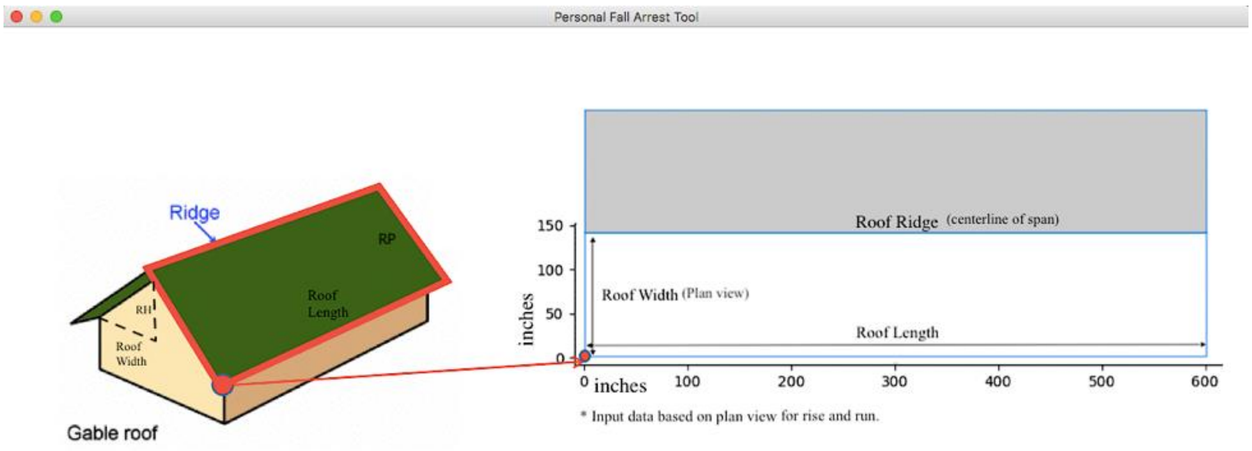

Figure 19: Roof geometry guide.

In order to gather enough information for the optimization, the qualified person (i.e. the tool user) needs to input project-specific values in the user input panel. These values include: 1) Building dimensions (e.g. roof height), distance between structural members (e.g. trusses or rafters) and suggested locations for the working platform as well as the initial anchor point, 2) PFAS specifications (e.g. lanyard length, harness stretch, etc.), 3) Worker information (e.g. worker weight and height), and 4) Location of physical barriers (e.g. a chimney).

The qualified person can also select default values of the input parameters that are built in the tool in the case that these default values could be applied to a project. The default values are based on some of the commonly found jobsite conditions. The input panel is accessed by clicking on the “START” button on the main menu. After clicking on the START button, a new window named user input panel opens up (Figure 20). In addition, the windows for fall clearance, swing hazard and roof geometry guides can remain open next to the user input panel, so that the user can refer to them simultaneously to the data input process.

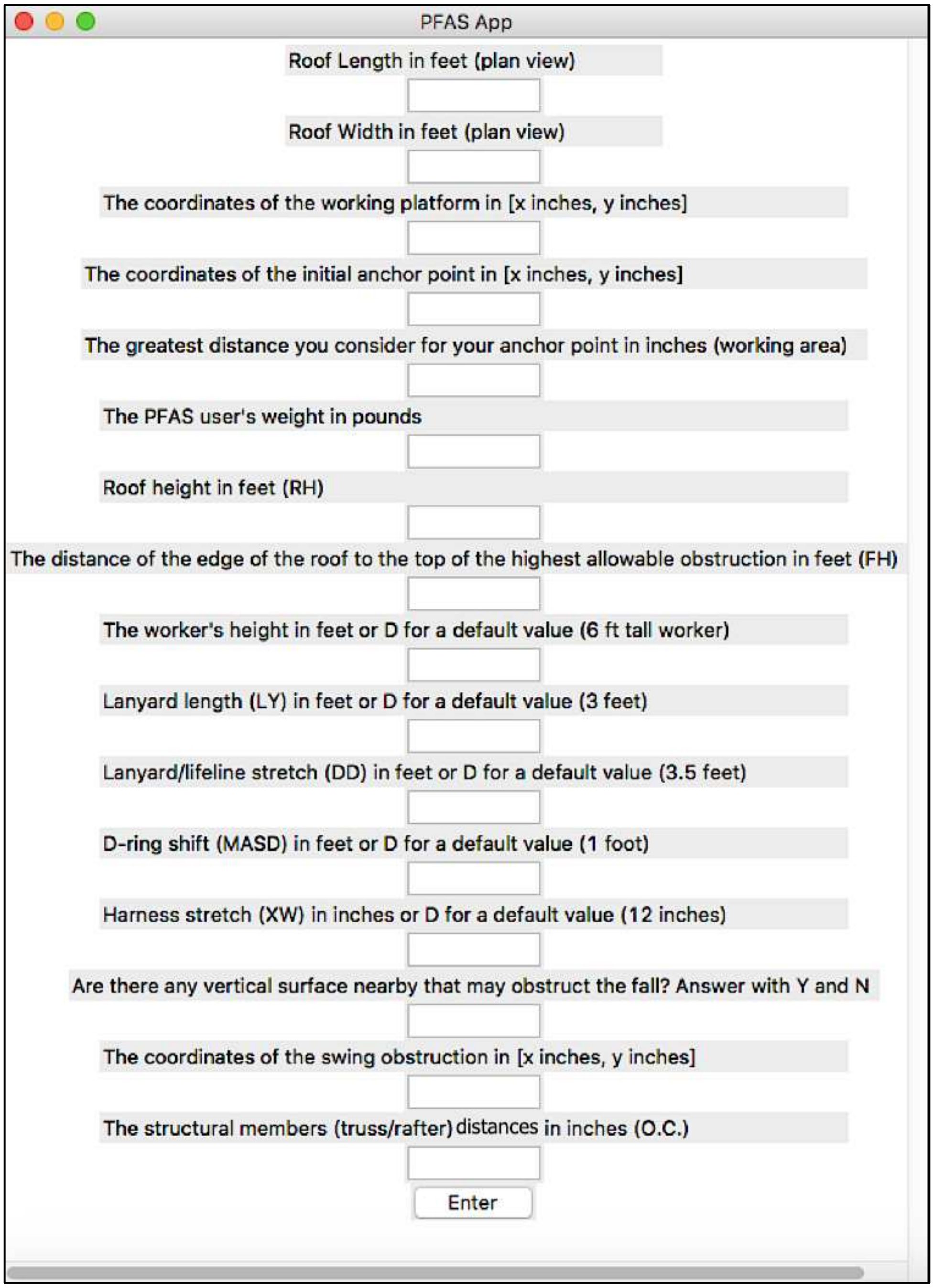

Figure 20: The user input panel.

The user input panel consists of many different values that the tool user will need to know. The user inputs the working platform as an initial point, knowing that the working platform is a dynamic value that changes as the worker moves the anchor in relation to the working area. The working area is defined as the area of a circle with a radius the length of the lanyard, centered at the initial anchor point. However, an initial point is needed to calculate the lanyard stretch value for the optimization process. The user data entry screen also requires the distance between trusses or stick built roof components (e.g. 16 inches or 24 inches on center).

The number of optimization parameters can be increased by improving the computer system specifications. This results in higher optimization accuracy; however, the default tool settings are selected in a way that the tool's launch time remains under six seconds for a typical system. The six seconds was chosen by the researcher in an effort to ensure that start up time did not impact the users desire to use the tool. Existing classical KNN queries mostly focused on reducing query processing time (Luo et al. 2018). The time taken for the optimization was a target code for the researcher and was done by decreasing the reducing query processing time. The user cannot change the optimization population number and the number of rounds of optimization in the tool, but these numbers can easily be modified in the Python script by the tool developer as needed.

After the tool evaluates the values entered in the user input panel, the tool runs the optimization. The tool then displays the results on a coordinate plane plotted in Matplotlib module (Figure 21). Any of the blue points, located on a structural member, can be selected by the tool user as the anchor point as shown in Figure 21. In addition, the numerical output of the optimization is automatically inserted into the text pad on the main menu (Figure 22).

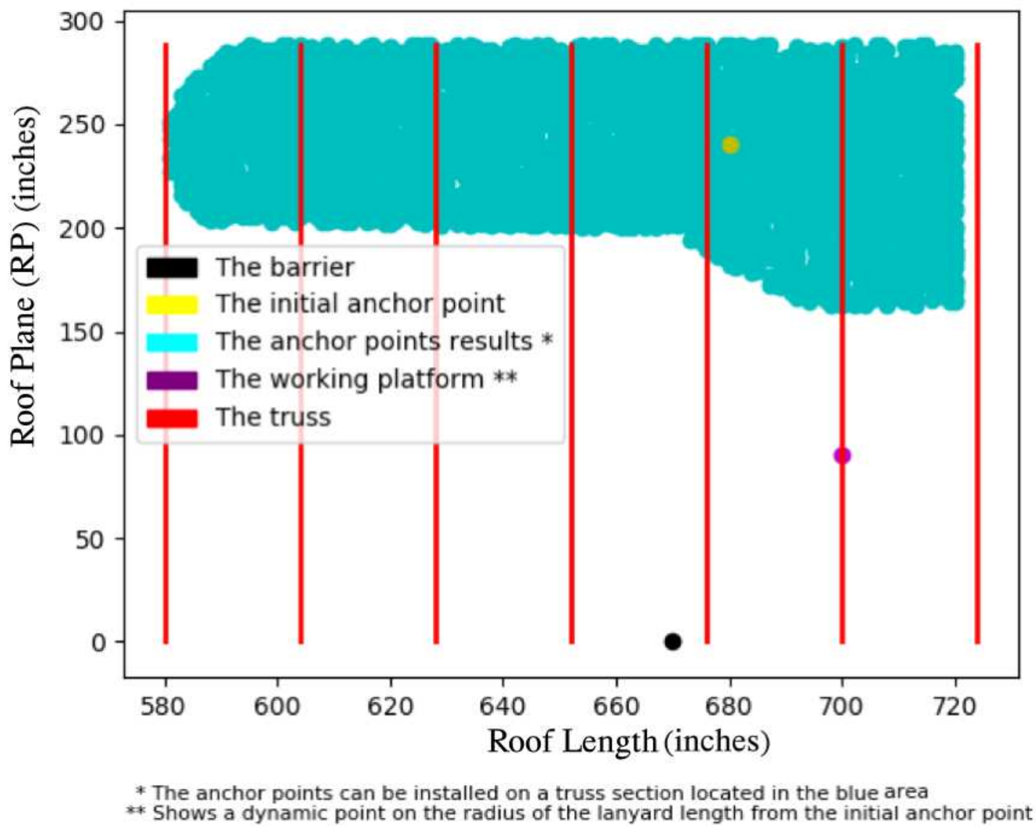

Figure 21: An example of the tool output coordinate plane.

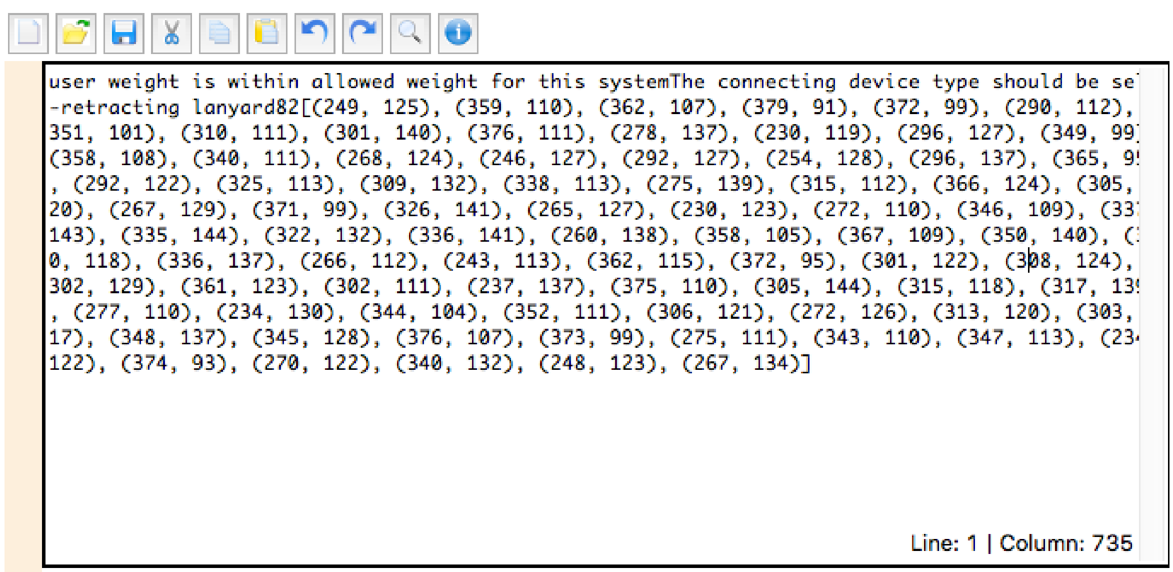

Figure 22: An example of the tool numerical output in the text pad on the main menu.

After running the calculations, the tool provides the following results to the user:

1) Whether the user weight is within allowed weight range for this tool.

2) The type of connecting device that should be used (e.g. self-retracting lanyard, shock absorber, etc.).

3) The anchor points within the user's working area input that satisfy all fall clearance and swing hazard rules.

With the use of this tool, the qualified person does not need to be present on the jobsite all the time. The qualified person gives instructions to the competent person on which structural members have sufficient structural strength to be selected to support the anchor point. For the rest of the job, the competent person can run the tool multiple times as the work progresses and the location of any barriers change, or as new barriers appear. The tool output for all iterations should be evaluated by competent person on the jobsite (foreman, project manager, superintendent) to make sure the suggested anchor point locations on structural members are used. This iterative process minimizes the need for the presence of the qualified person on the jobsite for determining the anchor points.

4.4.1. The Tool's Workflow

The tool workflow incorporates the following tasks:

1) Opening the fall clearance, swing hazard and roof geometry guides on the tool's main menu and having them open alongside the user input panel for reference,

2) Clicking on the START button to open the user input panel,

3) Entering the project values for the parameters in the user input panel:

  a. Roof length and width in inches: according to Figure 19, the tool refers to the length and width of one side of the gable roof (shown in plan view) as length and width of the roof (i.e. the gray side of the roof plan shown in Figure 19 is not included). Therefore, for the gable roof in this scenario (Figure 19), the roof width equals to 144 inches and the roof length equals to 600 inches.

b. The coordinates of the working platform, the anchor point and the swing obstruction: the user should enter these coordinates in inches in an [x, y] format based on the coordinate plane provided in Figure 19.

c. The greatest distance from the anchor point (working area) is determined by the lanyard length. The user should enter the greatest distance that they can work form the anchor point without negatively affecting the productivity and accessibility of the worker. As the work progresses the anchor point will need to be moved as required.

For the remaining values, the user should refer to the fall clearance and swing hazard guides (Figures 17 and 18) for further clarification.

4) Clicking the "Enter" button after all the input entries are filled in to run the optimization,

5) Saving the numerical results that are inserted in the text pad on the main menu,

6) Saving the report in a .txt file format for reference and as a documentation record.

In the next step, the user could evaluate the strength and accessibility requirements for the anchor point results obtained from the tool output. This step is not incorporated in the tool but can be done manually based on the results from the tool (since the locations of the structural members are shown in the results). For example, for the results in this case (Figure 21), the user limits the results of where to place the anchor to the areas with sufficient structural strength, such as the blue areas shown over trusses or stick built rafters.

4.4.2. Illustration of Tool Application

During the tool development, the tool was tested using 20 scenarios to verify the tool programming. This included the two scenarios explained in Section 3.2. For Scenario #1, the researcher ran the tool for a PFAS anchor point optimization in the case of a gable roof. The input values included: 1) The 6.5 foot distance from the roof ridge to the bottom chord of the truss system (RH); 2) A 13.12 foot height from the top of highest allowable obstruction to the edge of the roof (FH); 3) The tool's default values for the PFAS specifications; 4) Roof dimensions 40 feet by 13 feet and; 5) Within 80 inches, the greatest distance you consider for your anchor point (working area) as shown in the user inputs (Figure 23).

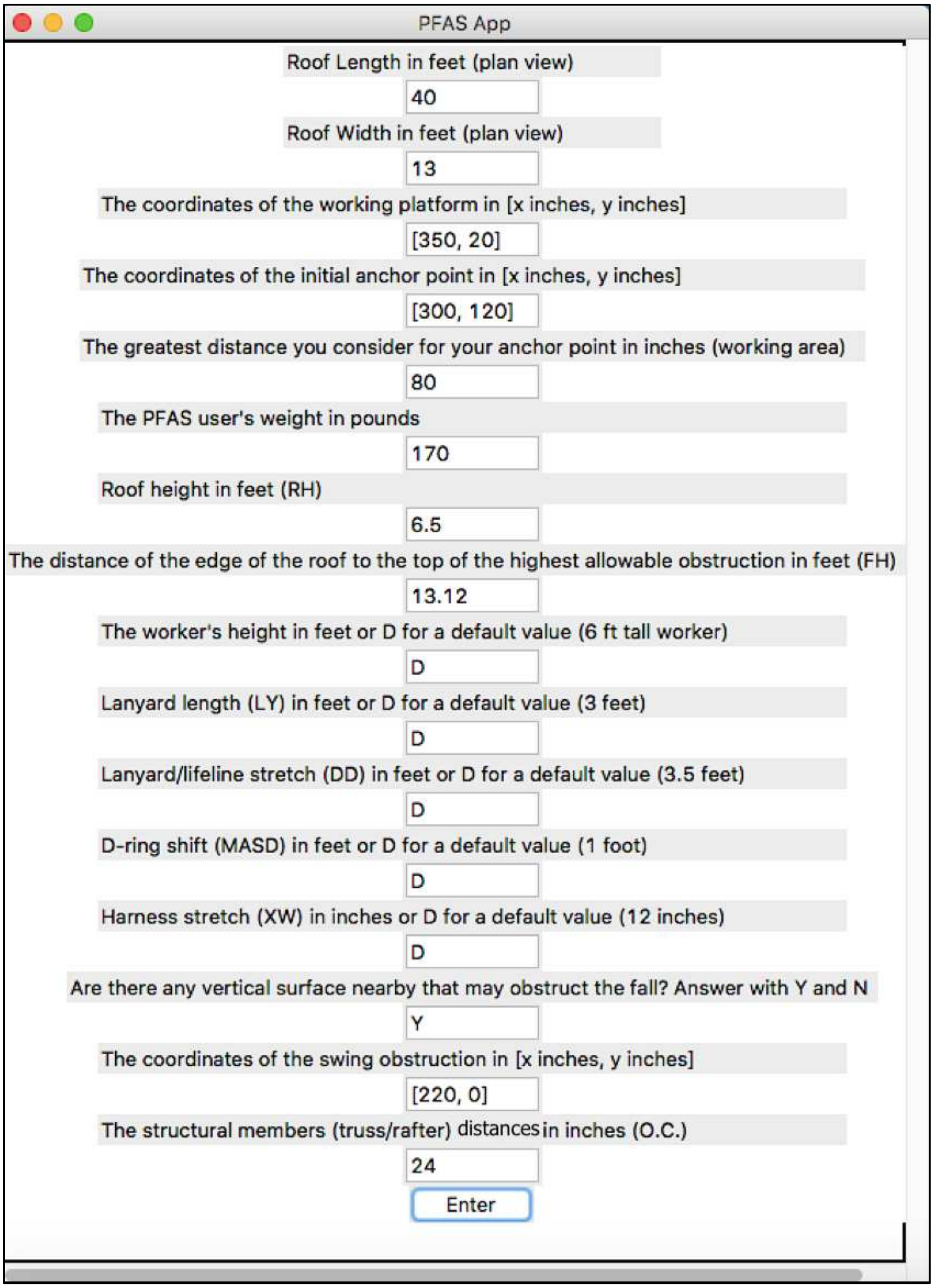

Figure 23: The user inputs for Scenario #1.

The tool output included the following results:

1) 15,663 points initially entered the optimization process.

2) 2,481 points were the results of the optimization.

3) User weight was within allowed weight for this system.

4) The fall clearance distance (C_A) equaled to 15.5 feet and the connecting device type should be self-retracting lanyard.

5) Some of the anchor points coordinates included: [(285, 81), (286, 144), (253, 146), (371, 91), (307, 109), (308, 101), (342, 140), (345, 152), (290, 126), (271, 128), (360, 83), (348, 99), (315, 106), (321, 135), (305, 144), (271, 145), (344, 149), (274, 119), (349, 156), (294, 114), (278, 152), (355, 96), (374, 140), (319, 78), (360, 117), (325, 147), (290, 151), (329, 114), (335, 78), (283, 124), (302, 125), (289, 152), (308, 108), (327, 143), (369, 118), (298, 70), (366, 129), (323, 82), (287, 120), (291, 125), (293, 156), (352, 125), (377, 107), (305, 137), (274, 110), (330, 147), (349, 149), (294, 117), (299, 122), (310, 85), (348, 152), (315, 154), (334, 95), (266, 135), (362, 138), (344, 69), (307, 123), (327, 132), (374, 114), (330, 155), (339, 129), (323, 91), (322, 108), (290, 112), (293, 133), (296, 119), (315, 120), (316, 106), (340, 96), (305, 130), (346, 139), (330, 154), (351, 85), (355, 110), (328, 76), (350, 151), (340, 137), (362, 154), (308, 114), (365, 127), (332, 72), (339, 134), (342, 145), (328, 149), (291, 107), (368, 146), (296, 126), (316, 97), (320, 100), (376, 129), (377, 117), (288, 93), (346, 146), (366, 81), (270, 130), (294, 103), (295, 111), (299, 104), (355, 151), (319, 101), (324, 112), (328, 115), (293, 85), (354, 69), (362, 133), (271, 119), (368, 105), (276, 98), (336, 130), (311, 140), (358, 89), (347, 95), (348, 71), (311, 77), (315, 118), (376, 136), (341, 96), (343, 139), (256, 141), (313, 76), (332, 152), (351, 91), (299, 97), (355, 156), (360, 81), (362, 155), (325,

119), (341, 119), (353, 86), (317, 72), (344, 90), (291, 136), (275, 121), (276, 105), (281, 98), (323, 78), (347, 84), (311, 74), (296, 108), (279, 125), (318, 136), (304, 156), (341, 105), (288, 107), (254, 144), (327, 74), (371, 148), (299, 102), (300, 104), (360, 88), (328, 97), (303, 128), (297, 149), (349, 106), (354, 87), (263, 133), (375, 128), (355, 135), (369, 111), (277, 96), (357, 75), (363, 156), (308, 90), (285, 111), (341, 114), (289, 84), (345, 119), (366, 106), (314, 84), (370, 89), (299, 135), (281, 133), (300, 103), (357, 146), (284, 150), (360, 95), (268, 120), (325, 121), (328, 104), (359, 142), (307, 92), (308, 86), (364, 91), (275, 135), (333, 122), (316, 150), (282, 149), (284, 126), (285, 104), (307, 135), (343, 150), (313, 103), (300, 94), (339, 77), (286, 153), (331, 125), (329, 103), (293, 73), (331, 138), (345, 99), (260, 134), (284, 112), (272, 129), (364, 82), (368, 149), (333, 115), (338, 126), (250, 153), (270, 145), (373, 93), (321, 79), (371, 137), (285, 89), (313, 104), (336, 71), (273, 125), (309, 156), (311, 136), (260, 141), (317, 108), (375, 111), (364, 73), (274, 148), (368, 140), (356, 83), (322, 136), (341, 156), (252, 156), (250, 149), (366, 140), (301, 148), (345, 74), (354, 136), (281, 152), (324, 94), (286, 139), (255, 151), (311, 129), (317, 101), (343, 95), (274, 143), (310, 140), (275, 154), (277, 130), (301, 124), (363, 120), (296, 79), (288, 142), (345, 83), (310, 101), (370, 125), (335, 103), (339, 96), (359, 93), (292, 106), (349, 71), (343, 84), (291, 82)]

6) The final results were graphically presented in Matplotlib (Figure 24).

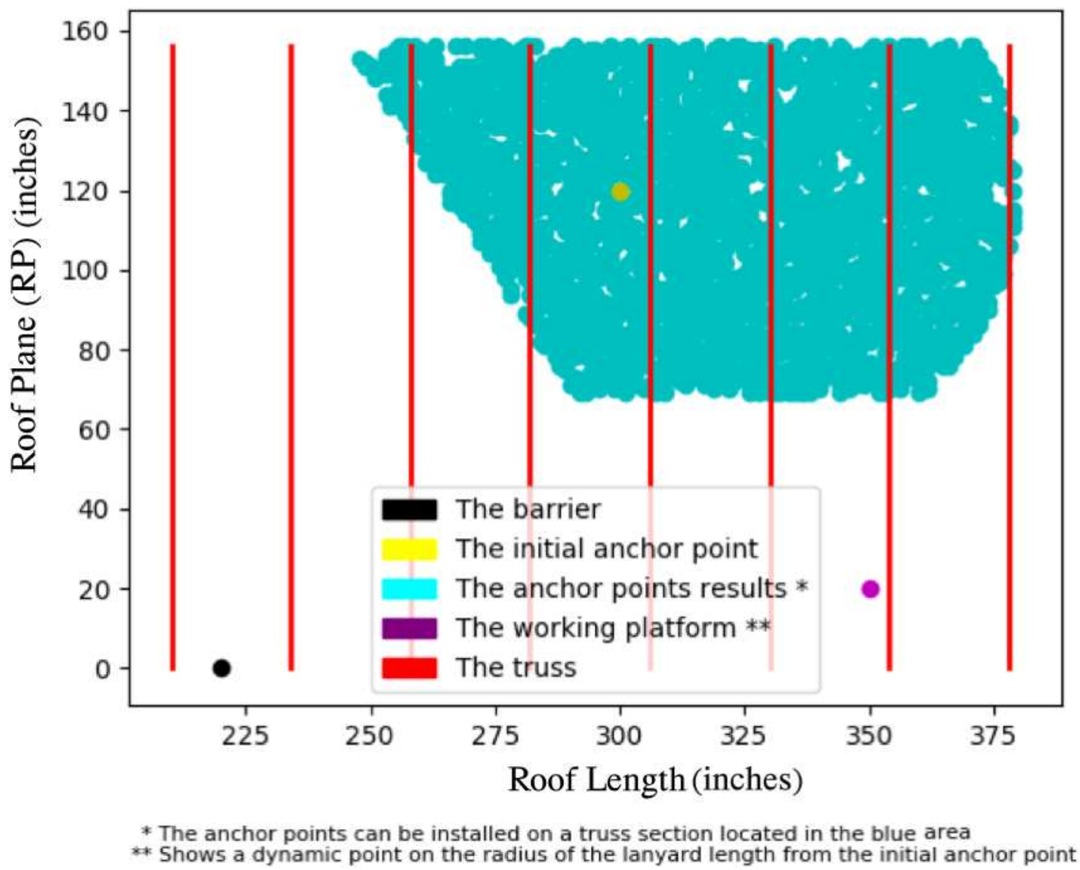

Figure 24: The tool output results for Scenario #1.

The tool was run for Scenario 1 and the output results are presented in Figure 24. The blue area demonstrates the possible anchor points over the entire area, not just those over the structural members where the anchor point would typically be installed. The locations of the roof trusses are shown as red lines to help the user pick the acceptable anchor points on the trusses as they are considered structural components of the roof. Without additional engineering and blocking, the points in between the trusses represent the roof sheathing, and may lack the structural strength required for anchor point (i.e. 5000 lbs.). The competent person can choose anchor points from the tool output as long as work is being done in the zone that was determined as the working area in the user input panel. The footnotes below the coordinate plane (Figure 24) help the user to better interpret the results of the tool.

Some of the results could be compared to the manual calculations for Scenario #1 in Section 3.2, such as required clearance below the anchorage (C_A) and the connecting device type. There was no difference between the tool results and the manually calculated results which showed that the tool provided accurate results for this scenario. However, some of the results for optimization could not be compared with manual calculations simply because the validation methodology was different (discussed in section 4.5 below).

The results of this scenario indicated that any point below the rake-end length of 63 inches will result in a swing hazard identified as point coordinates [220, 0] inches.

For Scenario #2, the researcher ran the tool for a PFAS anchor point optimization in the case of a gable roof. The input values included: 1) The 7.5 feet distance from the roof ridge to the bottom of the truss system (RH); 2) A 10 foot height from the top of highest allowable obstruction to the edge of the roof (FH); 3) The tool's default values for the PFAS specifications except for LY=5 ft; 4) Roof dimensions 60 by 24 feet; 5) Within 100 inches, the greatest distance you consider for your anchor point (working area) as shown in the user inputs (Figure 25).

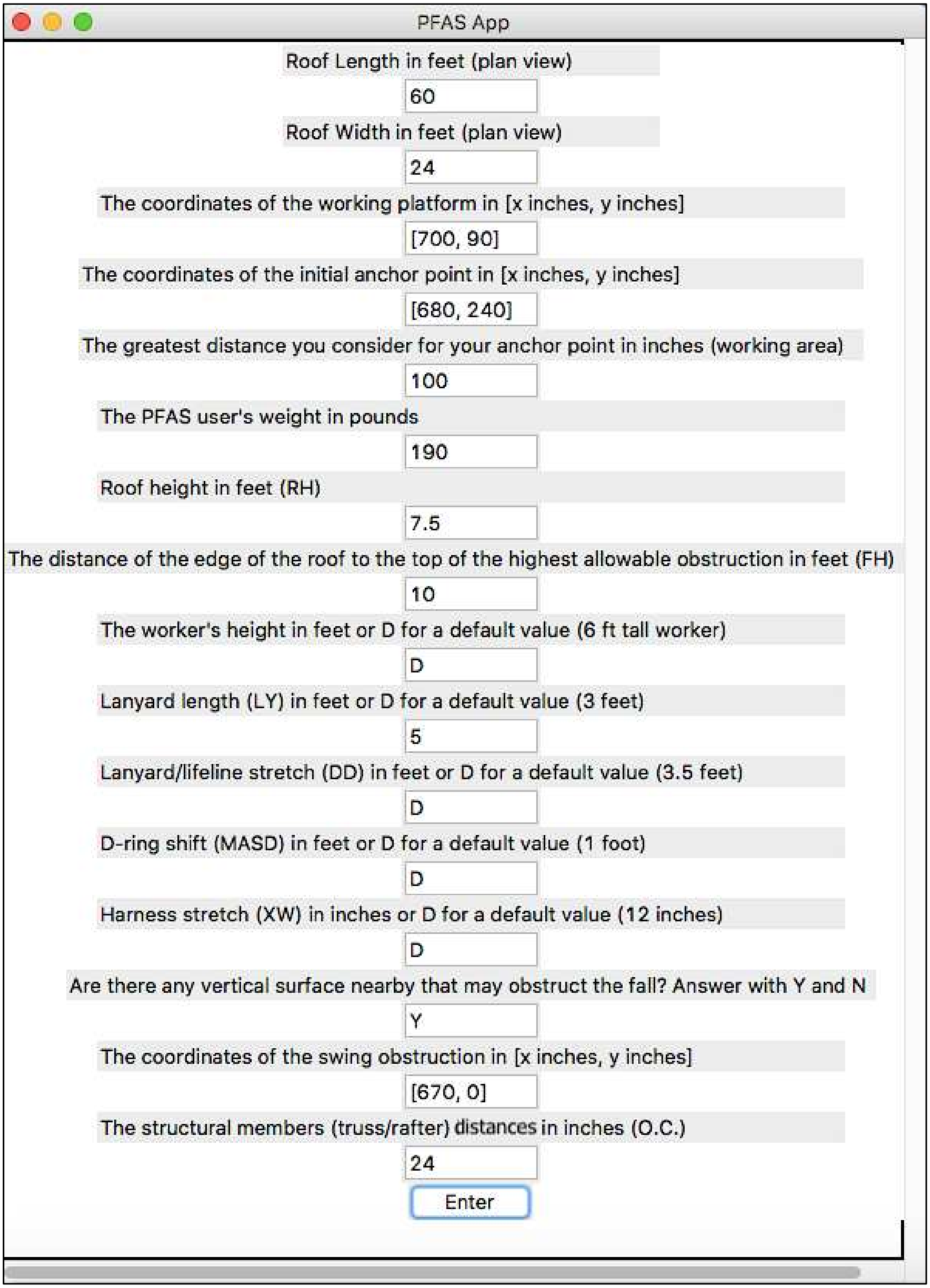

Figure 25: The user inputs for Scenario #2.

The tool output included the following results:

1) 18,405 points initially entered the optimization process.

2) 3,714 points were the results of the optimization.

3) User weight was within allowed weight for this system.

4) The fall clearance distance (C_A) equaled to 17.5 feet and the connecting device type should be self-retracting lanyard.

5) Some of the anchor points coordinates included: [(650, 223), (710, 274), (617, 273), (689, 173), (644, 265), (697, 228), (609, 269), (614, 225), (608, 237), (691, 229), (591, 284), (676, 248), (686, 212), (717, 287), (586, 227), (683, 209), (720, 255), (700, 220), (711, 263), (620, 242), (689, 280), (699, 261), (699, 200), (597, 236), (676, 257), (625, 211), (678, 220), (630, 229), (605, 278), (706, 192), (667, 228), (668, 238), (647, 220), (709, 188), (671, 234), (626, 288), (678, 214), (655, 257), (585, 229), (713, 229), (654, 203), (675, 271), (705, 278), (711, 282), (704, 180), (665, 229), (655, 225), (633, 288), (713, 209), (641, 284), (712, 163), (709, 165), (606, 278), (687, 220), (704, 255), (684, 215), (614, 207), (655, 218), (640, 257), (625, 264), (658, 267), (653, 228), (690, 273), (637, 203), (650, 270), (682, 197), (647, 264), (659, 235), (606, 237), (676, 285), (697, 255), (603, 280), (610, 279), (712, 176), (638, 230), (607, 235), (633, 208), (613, 240), (716, 201), (701, 213), (674, 288), (685, 242), (626, 242), (636, 267), (700, 201), (691, 219), (705, 165), (648, 232), (681, 197), (714, 189), (692, 269), (603, 270), (717, 170), (702, 226), (586, 213), (676, 250), (631, 278), (692, 180), (625, 255), (636, 207), (671, 269), (704, 264), (635, 246), (662, 276), (606, 225), (702, 282), (694, 236), (705, 208), (609, 260), (599, 265), (608, 230), (639, 226), (694, 262), (671, 212), (697,

213), (620, 231), (582, 229), (632, 210), (602, 218), (684, 262), (690, 241), (717, 187), (625, 244), (649, 217), (661, 252), (711, 276), (611, 281), (622, 229), (614, 262), (675, 279), (635, 248), (639, 282), (704, 172), (695, 277), (594, 287), (720, 179), (669, 239), (610, 282), (680, 235), (710, 213), (702, 246), (622, 283), (696, 266), (699, 226), (709, 215), (607, 286), (684, 239), (688, 234), (715, 214), (627, 227), (678, 204), (630, 245), (581, 227), (590, 235), (657, 275), (717, 194), (719, 251), (647, 204), (677, 210), (629, 247), (644, 270), (633, 236), (624, 288), (719, 264), (583, 223), (588, 234), (599, 216), (658, 259), (638, 206), (712, 257), (704, 238), (654, 234), (588, 222), (714, 247), (591, 250), (667, 215), (668, 223), (711, 288), (664, 231), (669, 252), (641, 201), (687, 265), (662, 252), (698, 275), (677, 232), (623, 241), (687, 236), (588, 225), (639, 231), (637, 225), (662, 200), (684, 265), (691, 181), (692, 182), (684, 251), (655, 254), (665, 288), (602, 229), (631, 207), (666, 219), (708, 208), (615, 215), (703, 168), (689, 188), (719, 286), (604, 265), (641, 228), (605, 268), (673, 203), (693, 184), (665, 272), (685, 226), (695, 234), (712, 251), (654, 241), (603, 212), (652, 269), (679, 219), (663, 244), (718, 239), (706, 241), (712, 249), (695, 197), (608, 213), (702, 244), (706, 166), (701, 216), (716, 183), (604, 231), (621, 278), (668, 236), (641, 220), (617, 205), (714, 190), (687, 258), (718, 275), (584, 246), (667, 236), (719, 266), (643, 277), (704, 202), (585, 210), (708, 267), (688, 262), (715, 235), (707, 287), (700, 195), (685, 273), (620, 241), (689, 281), (590, 206), (615, 233), (703, 199), (603, 216), (615, 216), (619, 285), (685, 188), (697, 238), (717, 211), (605, 210), (667, 265), (707, 249), (612, 233), (590, 261), (646, 235), (702, 200), (675, 256), (598, 240), (714, 256), (591, 205), (699, 283), (718, 222), (711, 283), (587, 265), (646, 277), (591, 235), (705, 265), (606, 229), (652, 272), (673, 253), (692, 272), (644, 260), (662, 239), (676, 237), (672, 197), (701, 227), (596, 271), (706, 238), (627, 265)]

6) The final results were graphically presented (Figure 26).

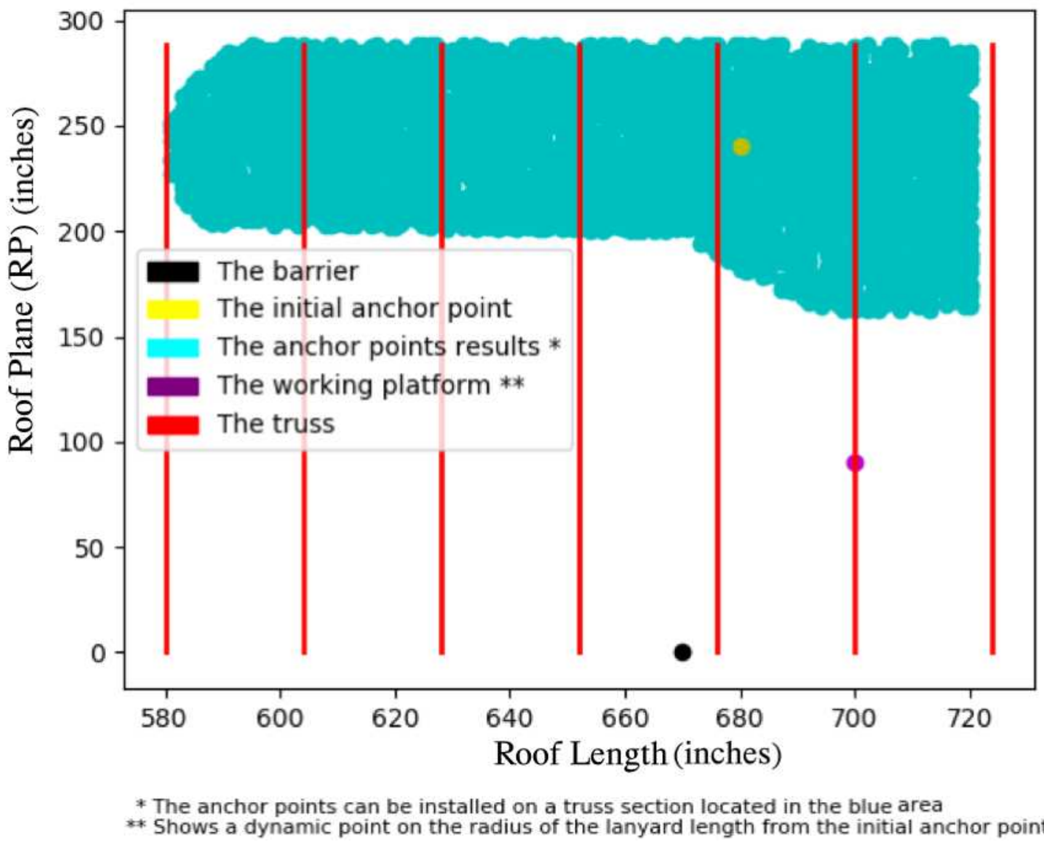

Figure 26: The tool output results for Scenario #2.

The tool was run for Scenario 2 and the output results are presented in Figure 26. The blue area demonstrates the possible anchor points over the entire area, not just those over the structural members where the anchor point would typically be installed. The locations of the roof trusses are shown as red lines to help the user pick the acceptable anchor points on the trusses as they are considered structural components of the roof. Without additional engineering and blocking, the points in between the trusses represent the roof sheathing, and may lack the structural strength required for anchor point (i.e. 5000 lbs.). The competent person can choose anchor points from the tool output as long as work is being done in the zone that was determined as the working area in the user input panel. The footnotes below the coordinate plane (Figure 26) help the user to better interpret the results of the tool.

The results obtained from the tool were then compared to the manually calculated results for the same scenario for required clearance below the anchorage (C_A) and connecting device type. There was no difference between the tool results and the manually calculated results which showed that the tool provided accurate results for this scenario. However, some of the results for optimization could not be compared with manual calculations simply because the validation methodology was different (discussed in section 4.5 below). The user can then consider the structural strength and accessibility requirements and then select the final locations for the anchor points.

Moving from left (Roof length of 580 inches) to right (Roof length of 720 inches) (see horizontal measurements on Figure 26), it can be seen that the curve of the roof plane (RP) coordinates (see vertical measurements on Figure 26) slopes down, meaning that the danger of swing hazard declines on the right side of the barrier. There is still a swing hazard on the left side of the barrier.

4.5. Validation of the Results

The validation of the accuracy of results was necessary, mainly because the programming of optimization model developed in Python should be evaluated. In order to validate the accuracy of optimization model, the K-fold Cross-Validation method explained in Chapter 3 was utilized. The K-fold Cross -Validation was used for validating five (k) groups of results. The validation was performed in the following steps:

1) 400 results from the tool (for a single scenario) were shuffled randomly, and 200 anchor points (N=200) were selected.
2) The dataset (200 points) was split into five groups (k=5).
3) The points in each group were examined for swing hazard and fall clearance requirements point by point. An automated rule-checking Python validation code for each anchor point entry was developed and the rule-checking was done through loops of rule-checking statements. In other words, each point in the final optimization results was tested to check if it satisfied the fall clearance and swing hazard requirements.
4) The accuracy of the results in each of the five groups was compared to each other, discussed below, and the average percentage of accuracy of all groups was determined. The closer the of all groups to one another (averages of all five groups were compared together), the more accurate the model.

Each of the criteria in the rule-checking (i.e. type of deceleration device, fall clearance and swing hazard verification) were given a numerical weight and the automated validation code calculated the scores for each of the five groups based on these weighted criteria. The results showed that 98.6% of the tool results satisfied all the requirements and that there was 1.4% error pertaining to the results that did not satisfy the swing hazard requirement. In other words, 3 out of 200 points, or 1.4% of cases had some error pertaining to the swing hazard calculations. That is mainly due to the complexity of swing hazard situations. For this reason, the tool incorrectly assumed the direction that swing would happen in those three cases.

In the next step, to examine the variation of results, two groups of tool results for a single scenario were compared (Figure 27). There were 235 points in each group for that scenario.

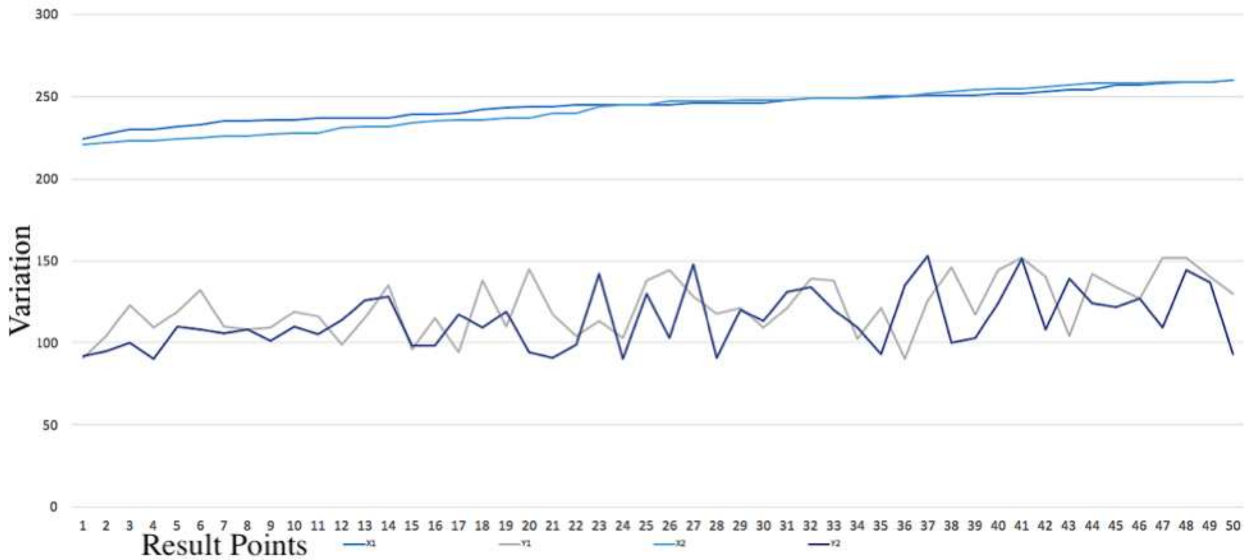

Figure 27: Variation of 50 out of 238 results. (Horizontal numbers are data points in each group)

Therefore, one point from one group and one from the other group were compared and the variation in their X and Y coordinates is shown in Figure 27. The same process was repeated for each pair of closest points. The average variation in the similar points coordinates equaled 0.13%. It showed that the results differed 0.13% each time the tool is run for the same input from a single scenario. Considering that the optimization algorithm in this research was based on the selection of the best fit of the data in the random dataset of points, the 0.13% variation in the results for each run is considered small in this study. However, due to the new nature of artificial intelligence (AI), it cannot be stated that any amount of error is acceptable concerning fall protection applications.

The optimization model and the tool results were partially validated in this step; however, as a part of the future study the tool results should be validated on the jobsite using an actual project.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

Falls from heights has been a major cause of construction fatalities over the last decade. This research identified this unsafe trend and aimed to contribute to the safety research in the fall protection area. PFAS have proven to be an effective means of protecting the worker's life in the case of a fall. However, statistics by Perry et al. (2015) show that PFAS are either not used or misused mainly due to human error. To address this problem, this research developed a tool for optimizing the location of PFAS anchors based on fall clearance and swing hazard regulations. As a part of this process, a ruleset that incorporated the fall protection and anchor point regulations and standards established by OSHA and ANSI was developed. The PFAS design process consists of many factors that need to be taken into account; however, the researcher found that fall clearance and swing hazard are of the utmost importance. The objectives of this research included: 1) Defining a ruleset for positioning the anchor points, 2) Examining the ruleset for different scenarios, 3) Developing an optimization model for the positioning of anchor points, 4) Automating the optimization model by creating a computer tool and 5) Validating the tool. First, the ruleset was developed using ANSI and OSHA regulations and standards and was tested for 20 scenarios of gable roofs with different dimensions and heights. Next, the KNNS optimization model was developed for the optimization of the anchor points for a PFAS system. It was based on the minimum Euclidian distance of the anchor point that the user initially had in mind with all the possible anchor points that satisfied the fall clearance and swing hazard requirements. The KNNS model was then coded in Python 3.6. using various Python modules such as Numpy, Sympy, Matplotlib and TkInter. TkInter was used to compile the script into a standalone computer tool with a user-friendly interface. The Matplotlib module was used to plot the result points on a coordinate plane as a graphical way of presenting results. Lastly, the results were validated using the K-fold Cross-Validation method to verify that the results met the tool requirements for fall clearance and swing hazard. The validation was also a means to measure the accuracy of the results.

The target users of this tool are small sized companies trying to protect their laborers working on steep sloped roofs (i.e. residential, roofing trades, small commercial, etc.). This group was chosen based on the literature identifying them as being most in danger of falling from height due to lack of proper safety equipment. The researcher developed and examined the ruleset for different scenarios without going into unnecessary details. Redundant details would have complicated the optimization model and increased the runtime of the tool significantly. The researcher ran the tool using different scenarios to make sure it ran smoothly and without any errors or breakdowns. Also, the researcher aimed to create a user interface that had a simple workflow to make the tool easy to work with.

5.1.1. Contribution of the Study

The contribution of this research is in improving worker safety by facilitating the use of PFAS in roofing construction. The optimization model helps the qualified person (e.g. PE) to select the safest anchor points and to ensure that the worker's productivity will not be decreased. Additionally, the use of the model makes the hazard assessment a conscious effort of the qualified person and helps the qualified person think about multiple hazardous scenarios. The tool helps eliminate uncertainties in PFAS design and errors in the anchor point calculations. As a result, the tool helps roofing contractors to comply with OSHA fall protection regulations and ensure that the worker's life can be fully trusted on the PFAS with lower cost and fewer technical team members (e.g. professional engineer, project manager) needed. In addition, the results of this study create a foundation for future research related to the use of automation for enhancing construction safety.

Moreover, roofing companies could benefit from an affordable and accessible tool that would help them to meet PFAS fall protection requirements. According to OSHA reports, a large number of small residential and roofing companies are not complying with the use of PFAS for sloped roofs. This tool could help OSHA accurately monitor company use of PFAS in the field. With the use of the tool, these companies could quickly produce anchor point reports and present them to OSHA as evidence of fall protection compliance, or a conscious effort to consider using s PFAS system. The companies violating fall protection regulations could be, for example, sanctioned or fined by OSHA if they fail to present reports produced by the tool.

5.2. Challenges and Limitations of the Study

The process for developing a tool typically follows a planned outline but may run into unanticipated issues. The following unanticipated challenges were encountered in this research:

1) The goal was to create a user-friendly tool by providing highly graphical user interface. This goal was partly addressed with use of TkInter and Matplotlib modules. However, most of the Python modules were unable to depict the calculations in a graphical way; thus, the problem remained partly unsolved. For example, the tool could benefit from graphical presentation of possible swing hazards and how they could lead to a fatal accident.

2) The KNNS optimization algorithm was at first intended to start the optimization with 10 rounds of 8,000 population data sets, but due to the limitations of the CPU and memory

capacities (see Table 5), the optimization happened at 2 rounds of 8,000 population data sets.

3) Another challenge was validating the optimization results. It was a challenge to find the proper method of validating the optimization results for the KNNS optimization. This is due to the need to modify the optimization method to accommodate the tool needs. No other acceptable validation methods were identified. Confusion matrix and likelihood-ratio validation methods were considered but were not selected for this study due to the inability to alter them to meet the needs of the tool validation.

5.3. Recommendations

5.3.1. Recommendations for Application of the Tool by Industry

The target users of this tool are small sized companies (i.e. residential, roofing trades, small commercial, etc.). They can take advantage of this tool by hiring a competent person to partly design the PFAS from the structural strength standpoint and then use the tool for calculations of fall clearance and swing hazard. Therefore, with use of the tool the PFAS design process will be much more affordable for small companies and it will also ensure the worker's life can be trusted using a PFAS system.

5.3.2. Recommendations for Future Research

There are still some obstacles to overcome in the design of the next versions of the tool to make it more reliable and user friendly. A future application of the tool may include the automatic design of PFAS with respect to all design requirements. Future research could focus on further development of the tool into a fully automated PFAS design. The design could include an optimization algorithm with respect to multiple aspects of PFAS such as strength calculations and accessibility for multiple types of projects with different roof geometries. As for the graphical improvements, the tool can benefit from an interactive 3D model exchange with design software such as Revit or SketchUp. The 3D model can be imported into the tool from the most commonly used BIM design software with different file formats (e.g. RVT, IFC, 3DS) and the tool could automatically recognize the project values from the 3D model as opposed to the user entering them manually.

Some of the future versions of the tool could include:

- Semi-automated detection of potential dangers associated with residential pitched-roof construction.
- Developing a framework for assisting the competent/qualified person in determining the optimal anchorage locations for larger projects with complex roof shapes.
- Increasing the tool’s accuracy.

In the future the tool should be tested on an actual construction site. The anticipated challenges for a successful field implementation include the ever-changing location of potential barriers for swing hazard and the structural limitations in selecting a proper anchor point. Future versions of the tool could include a mobile application with the purpose of facilitating the use of tool on the jobsite.

REFERENCES

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. *46 (3)*, pp.175–185.

Andoni, A. (2009). *Nearest neighbor search: the old, the new, and the impossible*. Retrieved from http://www.mit.edu/~andoni/, Doctoral dissertation, Massachusetts Institute of Technology.

American National Standards Institute (ANSI) (2017). Safety Requirements for Anchorage Connectors for Active Fall Protection Systems (ASSE Z359.18-2017). Retrieved from http://www.asse.org/ansi/

American National Standards Institute (ANSI) (2017), Minimum Requirements for a Comprehensive Managed Fall Protection Program (ASSE Z359.2-2017), Retrieved from http://www.asse.org/ansi/

American National Standards Institute (ANSI) (2016), Specifications and Design Requirements for Active Fall Protection Systems (ASSE Z359.6-2016), Retrieved from http://www.asse.org/ansi/

Bethancourt, J. (2011). OSHA rescinds residential fall protection exemption. *Professional Safety*, *56*(6), pp.66.

Cawley, G. & Talbot, N.L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. 11. *Journal of Machine Learning Research*, 2079–2107.

Cable, J. (2006). ASSE: Are you a “self-inter-preter” of OSHA fall protection standards? *EHS Today*. Retrieved from http://ehstoday.com/ppe/fall-protection/ ehs_imp_38294

Chi, C., Chang, T., & Ting, H. (2005). Accident patterns and prevention measures for fatal occupational falls in the construction industry. *Applied Ergonomics,36(4)*, pp.391-400.

Dasarathy, B. V. (1991). Nearest neighbor ({NN}) norms: {NN} pattern classification techniques. University of Michigan, MI: IEEE Computer Society Press.

Dhillon, I. S., & Modha, D. S. (2002). A data-clustering algorithm on distributed memory multiprocessors. Large-scale parallel data mining, Berlin, Heidelberg: Springer, pp. 245-260.

Dietz Harry (2017). [Graph illustration NRCA]. Disaster Responder Roofing Safety. Retrieved from http://www.elcosh.org/document/4250/d001531/Disaster%2BResponder%2BRoofing%2BSafety.html

Dong, X. S., Largay, J. A., Choi, S. D., Wang, X., Cain, C. T., & Romano, N. (2017). Fatal falls and PFAS use in the construction industry: Findings from the NIOSH FACE reports. *Accident Analysis & Prevention,102*, 136-143. doi: 10.1016/j.aap.2017.02.028.

Fisk, E. R., & Reynolds, W. D. (2014). Construction project administration. Boston: Pearson. Chapter 8, pp. 135-137.

Graupe, D., (2013). Principles of Artificial Neural Networks. 3rd Edition, World Scientific Publishers, pp. 253–274.

Haslam R.A., Hide S.A., Gibb A.G.F., Gyi D.E., Pavitt T., Atkinson S., Duff A.R., (2005) Contributing factors in construction accidents, *Applied Ergonomics*, *36(4)*, Pages 401-415, Retrieved from https://doi.org/10.1016/j.apergo.2004.12.002.

Hinze, W. J., & Olbina, S. (2008). Problem areas in personal fall protection. *Proceedings of CIB W* , *99*, pp. 406-18).

Huang, X., & Hinze, J. (2003). Analysis of construction worker fall accidents, *Construction Engineering and Management*, *129*(3), pp.262-271.

Johnson, B. (2011). Bad timing for new OSHA rules on fall protection? *Finance & Commerce*. Retrieved from http://nance-commerce.com/2011/06/ bad-timing-for-new-osha-rules.

Kaskutas, V., Dale, A.M., Lipscomb, H., and Evanoff, B. (2008). Development of the St. Louis audit of fall risks at residential construction sites. *International Journal of Occupational and Environmental Health*. *14*, pp.243-9.

Kang, Y., Siddiqui, S., Suk, S. J., Chi, S., & Kim, C. (2017). Trends of Fall Accidents in the US Construction Industry, *Construction Engineering and Management, 143(8),* 04017043.

Kines, P. (2002). Construction workers' falls through roofs: fatal versus serious injuries, *Safety Research*, *33(2),* pp.195-208.

Kramer, T. (2010). an interview with Thomas Kramer, chair of the ANSI Z359.6 and Z359.17 (Safety Requirements for Horizontal Lifelines for Personal Fall Arrest Systems) subcommittees. *ASSE Professional Safety*, *7*, pp.57-59.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. New York: Springer, *26*.

Luo, S., Kao, B., Li, G., Hu, J., Cheng, R., & Zheng, Y. (2018). TOAIN: a throughput optimizing adaptive index for answering dynamic k NN queries on road networks. *Proceedings of the VLDB Endowment*, *11(5),* pp.594-606.

Malekitabar, H., Ardeshir, A., Sebt, M. H., & Stouffs, R. (2016). Construction safety risk drivers: A BIM approach. *Safety Science, 82*, 445-455. doi:10.1016/j.ssci.2015.11.002.

NAHB Research Center (2011). [Graph illustration]. Section 2 Fall Protection Systems. Retrieved from https://slideplayer.com/slide/7313428/

National Safety Council (2013). Injury Facts (Rep.), 2013 Edition, Retrieved from http://www.mhi.org/downloads/industrygroups/ease/technicalpapers/2013-National-Safety-Council-Injury-Facts.pdf

Occupational Safety and Health Administration (OSHA) (2016), Commonly Used Statistics, Top 10 most frequently cited OSHA standards violated in FY2016, Retrieved from https://www.osha.gov/oshstats/commonstats.html

Occupational Safety and Health Administration (OSHA) (2015a), 1926.500(b)—Definitions, Retrieved from https://www.osha.gov/

Occupational Safety and Health Administration (OSHA) (2015b), Duty to have fall protection (29 CFR 1926.501), Retrieved from https://www.osha.gov/

OSHA (2015c) 1915.151(b), Personal Protective Equipment- Scope, application and definitions.

OSHA (2015d) 1926.502(d)(16)(iii), Steel Erection- Fall protection criteria and practices.

OSHA (2015e) 1926.501(b)(10), Fall Protection- Duty to have fall protection.

OSHA (2015f) 1926.501(b)(11), Fall Protection- Duty to have fall protection.

OSHA Technical Manual (OTM). (2016), UNITED STATES DEPARTMENT OF LABOR. Retrieved March 20, 2018, from https://www.osha.gov/dts/osta/otm/otm_v/otm_v_4.html

Perry, M. J., McQueen, A., & Kim, S. (2015). Evaluating Fall Safety Compliance among Skilled Trades in Construction, *CPWR Small Study Final Report,* Retrieved from www.elcosh.org

Shanahan, T., & Dietz, H. (2011). Roofing Industry Fall Protection from A to Z. National Roofing Contractors Association (NRCA), pp.10-25.

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, pp.69-82.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment*, *62(1)*, pp.77-89.

Stromme, M.H. (2011). Five fall protection myths: Clear up construction industry misconceptions. *Industrial Safety & Hygiene News*. Retrieved from www.ishn.com/articles/91777-5-fall-protection-myths

The Construction Chart Book: Fall Injuries and Prevention in the Construction Industry (2017). Silver Spring, MD: CPWR – The Center for Construction Research and Training. First quarter.

The Construction Chart Book: The U.S. Construction Industry and its worker (2013). Silver Spring, MD: CPWR – The Center for Construction Research and Training. Fifth Edition.

Wang, J., Zhang, S., & Teizer, J. (2015). Geotechnical and safety protective equipment planning using range point cloud data and rule checking in building information modeling. *Automation in Construction, 49*, pp.250-261. doi:10.1016/j.autcon.2014.09.002.

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building information modeling (BIM) and safety: Automatic safety checking of construction models and schedules. *Automation in Construction*, *29*, pp.183-195.

APPENDIX 1: PFAS RULES

A list of applicable PFAS rules for this study is presented. The ruleset developed in this research was based on this list.

1. ANSI/ASSE Z359.6-2016, Specifications and Design Requirements for Active Fall Protection Systems
    - Anchorage definition by OSHA: A secure point of attachment for equipment such as lifelines, lanyards, or deceleration devices.
    - The fall arrest system shall be used in the temperature range between 35˚F (2˚C) and 100˚F (38˚C).
    - The passive protection systems (i.e. guardrails and safety nets) may not be used as anchorage for PFAS.
    - Design strength may be determined by: 1) empirical testing 2) numerical techniques by design specifications (i.e. found on ACI 318, ACI 530, AISC 360 and NDS).
    - The allowed weight for PFAS user is in range of 130-310 pounds.
    - Drawings, specifications and procedures for PFAS shall be prepared by the qualified person/competent person.
    - The fall protection plan requires a statement defining the type of active fall protection system (travel restraint or fall arrest), a drawing showing the layout of the system, and that the design complies with ANSI and OSHA standards.
    - User inputs:
        i. Maximum number of workers permitted on the system

ii. MAF (i.e. Maximum arrest force that is allowed to be released on the worker body), is 1800 lbs. for test weight of 220 lbs. in accordance to ANSI Z.359.1.

iii. MAL (Maximum arrest load that occur to an anchor) which is 5000 lbs. (22.2 kN) per OSHA rules.

iv. Stretch out of full body harness

v. Safety factor (Ω) (i.e. 2, 2.5, 3 or 5)

- The PFAS shall be inspected prior to each use as well as once per year by a competent person.
- PFAS deceleration device can include personal energy absorbers, self-retracting devices or horizontal lifeline energy absorbers.

i. Personal energy absorbers: Also known as shock absorbing device, connects body wear to the lanyard. A shock absorber is a mechanical or hydraulic device designed to absorb and damp shock impulses by converting the kinetic energy of the shock into another form of energy (typically heat), which is then dissipated.

ii. Self-retracting devices: The retracted end of the lifeline unwinds from the drum under the tension created by the worker's normal movement below the device. When tension is released, the drum automatically retracts the lifeline. Once the speed at which the lifeline pays out reaches approximately 1.5 meters per second (5 feet per second), a velocity-sensing device engages a brake or locking mechanism that arrests the worker's motion *(works like car seat belt).* Self-retracting devices use on a sloped roof could be a problem because when the worker slips down the slope, he may not reach a high enough speed to engage the mechanism.

iii. Horizontal lifeline energy absorbers: This type of connecting device is mostly used for multi-user PFAS and sometimes only functions as fall restraint system.

- Under certain circumstances, it is impossible to design a fall-arrest system that will stop the fall within the available clearance. In such cases, the designer should either increase the rigidity of the anchorage, use a low-stretch full-body harness, restrict the maximum weight of worker who may use PFAS or removal of personal energy absorbers.
- When the swing fall cannot be practically prevented, and the user could potentially swing into an obstruction, therefore, Eq[1] should be satisfied to prevent such swing hazard.

  Free fall distance + harness stretch distance + swing drop distance < 4.0 ft.   Eq[1]
- For strength calculations of fall arrest systems:

  $A_{FAS}= m*F_{STR}$ Eq [2]   m=number of users, default=1 (Tables 6 to 9)

Table 6: Arrest Force (F_{STR}) of Personal Energy Absorbers for Strength Calculations:

F_{STR}	Default	Minimum	Wet or frozen condition
Compliant with ANSI Z359.13	1800 pounds	-	-
6-foot free fall	-	900 pounds	>1,125 pounds
12-foot free fall	-	1,350 pounds	>1,575 pounds
Not compliant with ANSI Z359.13	User Input		

Table 7: Arrest Force (F_{CLR}) of Personal Energy Absorbers for Fall Clearance Calculations:

F_{CLR}	Default
6-foot free fall	700 pounds or user input
12-foot free fall	950 pounds or user input
Non-Compliant with ANSI Z359.13	Manufacturer input

Table 8: Arrest Force (F_{STR}) of self-retracting devices for Fall Clearance Calculations: (i.e. FF limited to activation distance)

F_{STR}	Default
Dynamic Analysis	1800 pounds OR Class A: >1,350 Frozen Condition: >1,575 Class B: >900 Frozen Condition: >1,125
Static Analysis	1800 pounds OR Class A: >1,350 Frozen Condition: >1,575 Class B: >900 Frozen Condition: >1,125

Table 9: Arrest Force (F_{STR}) of self-retracting devices for Fall Clearance Calculations:

(when FF exceeds activation distance)

	F_{STR}	Normal	Frozen Condition
Internal Energy Absorber	Class A	>1,350	>1,575
	Class B	>900	>1,125
External Energy Absorber	In accordance with 5.3.1.		

- Strength requirements should be met via calculation of various loads on the structure:
  i. Dead Load (D)
  ii. Active Force (A)
  iii. Live Load (L)
  iv. Lateral Load (H)
  v. Self-Straining Load (T)
  vi. Equipment Load (Q)
- User inputs should be defined as mandatory and non-mandatory.
- Strength Design: All components of an active fall protection system shall have sufficient strength such that:

  $R_U \leq \phi R_n$, Eq[4]

  where:

  R_U=the required strength determined by the combination of factored load effects.

R_n= the nominal strength determined in accordance with the design specification for the material of the component.

ϕ= the resistance factor given by the design specification for a particular limit-state.

ϕR_n= the design strength.

- Clearance Requirement (C_P or C_A):

  Worst case of free fall distance + deceleration distance + stretch out +swing fall distance (if applicable) and a clearance margin (for rigid anchors shall not be less than two feet)

- Free Fall definition: Free fall is the unimpeded fall distance of the user. Free fall ends when all slack has been taken out of the fall arrest system so that further displacement of the user will be resisted by forces developed in the system.
- Rope slack= length of lanyard+ height of D-ring on user's harness Eq[5]
- Deceleration Distance definition: the distance over which a fall arrest system reacts to bring a falling user to a complete stop.

2. OSHA 29 CFR 1926.500 - .503

- Deceleration distance: This is the distance the lanyard stretches in order to arrest the fall. Deceleration distance must be no greater than 3.5 feet.
- Safety factor: A safety factor is an additional distance added to the total fall clearance distance to ensure there is enough clearance between the worker and the lower level after a fall. It is typically two feet (i.e. three feet to be prudent).
- D-ring shift: This is the distance the D-ring moves and the harness shifts when they support the worker's full weight. As the line tugs upwards, the harness can shift so the D-ring

location is higher on the worker than it was before the fall. This shift is often assumed to be one foot.

- Back D-ring height: The D-ring height is measured as the distance between the D-ring and the worker's shoe sole while the worker is wearing the harness. This height is often standardized as five feet for six-foot-tall workers.

3. International Safety Equipment Association (ISEA) Personal Fall Protection Guide

- Do not select anchorage locations that will require the user to work above the anchor as this will increase the potential free fall and total fall distances. When fall clearance is over 18.5 ft. (5.6m) there is sufficient total fall distance available and the six ft. lanyard is acceptable to use. Note that energy absorbing lanyards can expand up to 3.5 ft. (1.1m).

APPENDIX 2: PYTHON SCRIPT

As the result of the programming of the tool in Python, the following script was developed:

```
from Tkinter import *
import Tkinter as tk
import tkFileDialog
import tkMessageBox
import os
from tkinter import ttk
from numpy import *
from math import *
from sympy import Point, Line
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches


root = Tk()
root.geometry('600x500')
root.title('Personal Fall Arrest Tool')
root.iconbitmap('icons/pypad.ico')


def popup(event):
    cmenu.tk_popup(event.x_root, event.y_root, 0)


# theme choice
def theme():
    global bgc, fgc
    val = themechoice.get()
    clrs = clrschms.get(val)
    fgc, bgc = clrs.split('.')
    fgc, bgc = '#' + fgc, '#' + bgc
    textPad.config(bg=bgc, fg=fgc)


def show_info_bar():
    val = showinbar.get()
    if val:
        infobar.pack(expand=NO, fill=None, side=RIGHT, anchor='se')
    elif not val:
```

```
        infobar.pack_forget()


def update_line_number(event=None):
    txt = ''
    if showln.get():
        endline, endcolumn = textPad.index('end-1c').split('.')
        txt = '\n'.join(map(str, range(1, int(endline))))
    lnlabel.config(text=txt, anchor='nw')
    currline, curcolumn = textPad.index("insert").split('.')
    infobar.config(text='Line: %s | Column: %s' % (currline, curcolumn))


def highlight_line(interval=100):
    textPad.tag_remove("active_line", 1.0, "end")
    textPad.tag_add("active_line", "insert linestart", "insert lineend+1c")
    textPad.after(interval, toggle_highlight)


def undo_highlight():
    textPad.tag_remove("active_line", 1.0, "end")


def toggle_highlight(event=None):
    val = hltln.get()
    undo_highlight() if not val else highlight_line()


##########################################################################
def about():
    tkMessageBox.showinfo("About", "A tool for personal fall arrest system (PFAS) setup by
Azin Heidari")


def help_box(event=None):
    tkMessageBox.showinfo("Help", "For help email to chenyu.wu@outlook.com",
icon='question')


def exit_editor():
    if tkMessageBox.askokcancel("Quti", "Do you really want to quit?"):
        root.destroy()


root.protocol('WM_DELETE_WINDOW', exit_editor)
```

```
##########################################################################

# demo of indexing and tagging features of text widget
def select_all(event=None):
    textPad.tag_add('sel', '1.0', 'end')


def on_find(event=None):
    t2 = Toplevel(root)
    t2.title('Find')
    t2.geometry('300x65+200+250')
    t2.transient(root)
    Label(t2, text="Find All:").grid(row=0, column=0, pady=4, sticky='e')
    v = StringVar()
    e = Entry(t2, width=25, textvariable=v)
    e.grid(row=0, column=1, padx=2, pady=4, sticky='we')
    c = IntVar()
    Checkbutton(t2, text='Ignore Case', variable=c).grid(row=1, column=1, sticky='e', padx=2,
pady=2)
    Button(t2, text='Find All', underline=0, command=lambda: search_for(v.get(), c.get(), textPad,
t2, e)).grid(row=0,
                                                                                  column=2,
                                                                                  sticky='e' + 'w',
                                                                                  padx=2,
                                                                                  pady=4)

    def close_search():
        textPad.tag_remove('match', '1.0', END)
        t2.destroy()

    t2.protocol('WM_DELETE_WINDOW', close_search)


def search_for(needle, cssnstv, textPad, t2, e):
    textPad.tag_remove('match', '1.0', END)
    count = 0
    if needle:
        pos = '1.0'
        while True:
            pos = textPad.search(needle, pos, nocase=cssnstv, stopindex=END)
            if not pos: break
            lastpos = '%s+%dc' % (pos, len(needle))
            textPad.tag_add('match', pos, lastpos)
            count += 1
            pos = lastpos
```

```
        textPad.tag_config('match', foreground='red', background='yellow')
    e.focus_set()
    t2.title('%d matches found' % count)


##########################################################################
# Levaraging built in text widget functionalities

def undo():
    textPad.event_generate("<<Undo>>")
    update_line_number()


def redo():
    textPad.event_generate("<<Redo>>")
    update_line_number()


def cut():
    textPad.event_generate("<<Cut>>")
    update_line_number()


def copy():
    textPad.event_generate("<<Copy>>")
    update_line_number()


def paste():
    textPad.event_generate("<<Paste>>")
    update_line_number()


########################################################################
def new_file(event=None):
    global filename
    filename = None
    root.title("Untitled - Tkeditor")
    textPad.delete(1.0, END)
    update_line_number()


def open_file(event=None):
    global filename
    filename = tkFileDialog.askopenfilename(defaultextension=".txt",
                                          filetypes=[("All Files", "*.*"), ("Text Documents", "*.txt")])
```

```
    if filename == "":  # If no file chosen.
        filename = None  # Absence of file.
    else:
        root.title(os.path.basename(filename) + " - Tkeditor")  # Returning the basename of 'file'
        textPad.delete(1.0, END)
        fh = open(filename, "r")
        textPad.insert(1.0, fh.read())
        fh.close()
    update_line_number()


def save(event=None):
    global filename
    try:
        f = open(filename, 'w')
        letter = textPad.get(1.0, 'end')
        f.write(letter)
        f.close()
    except:
        save_as()


def save_as():
    try:
        # Getting a filename to save the file.
        f = tkFileDialog.asksaveasfilename(initialfile='Untitled.txt', defaultextension=".txt",
                                           filetypes=[("All Files", "*.*"), ("Text Documents", "*.txt")])
        fh = open(f, 'w')
        global filename
        filename = f
        textoutput = textPad.get(1.0, END)
        fh.write(textoutput)
        fh.close()
        root.title(os.path.basename(f) + " - Tkeditor")  # Setting the title of the root widget.
    except:
        pass


##########################################################################
# defining icons for compund menu demonstration
newicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/new_file.gif')
openicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/open_file.gif')
saveicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Save.gif')
cuticon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Cut.gif')
copyicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Copy.gif')
pasteicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Paste.gif')
```

```
undoicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Undo.gif')
redoicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/Redo.gif')



# Define a menu bar
menubar = Menu(root)

# File menu
filemenu = Menu(menubar, tearoff=0)
filemenu.add_command(label="New", accelerator='Ctrl+N', compound=LEFT, image=newicon,
underline=0, command=new_file)
filemenu.add_command(label="Open", accelerator='Ctrl+O', compound=LEFT,
image=openicon, underline=0, command=open_file)
filemenu.add_command(label="Save", accelerator='Ctrl+S', compound=LEFT, image=saveicon,
underline=0, command=save)
filemenu.add_command(label="Save as", accelerator='Shift+Ctrl+S', command=save_as)
filemenu.add_command(label="Exit", accelerator='Alt+F4', command=exit_editor)
menubar.add_cascade(label="File", menu=filemenu)

# Edit menu
editmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="Edit", menu=editmenu)
editmenu.add_command(label="Undo", compound=LEFT, image=undoicon,
accelerator='Ctrl+Z', command=undo)
editmenu.add_command(label="Redo", compound=LEFT, image=redoicon,
accelerator='Ctrl+Y', command=redo)
editmenu.add_separator()
editmenu.add_command(label="Cut", compound=LEFT, image=cuticon, accelerator='Ctrl+X',
command=cut)
editmenu.add_command(label="Copy", compound=LEFT, image=copyicon,
accelerator='Ctrl+C', command=copy)
editmenu.add_command(labe="Paste", compound=LEFT, image=pasteicon,
accelerator='Ctrl+V', command=paste)
editmenu.add_separator()
editmenu.add_command(label="Find", underline=0, accelerator='Ctrl+F', command=on_find)
editmenu.add_separator()
editmenu.add_command(label="Select All", accelerator='Ctrl+A', underline=7,
command=select_all)

# View menu

viewmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="View", menu=viewmenu)
showln = IntVar()
showln.set(1)
```

```
viewmenu.add_checkbutton(label="Show Line Number", variable=showln)
showinbar = IntVar()
showinbar.set(1)
viewmenu.add_checkbutton(label="Show Info Bar at Bottom", variable=showinbar,
command=show_info_bar)
hltln = IntVar()
viewmenu.add_checkbutton(label="Highlight Current Line", variable=hltln,
command=toggle_highlight)
themesmenu = Menu(viewmenu, tearoff=0)
viewmenu.add_cascade(label="Themes", menu=themesmenu)

# we define a color scheme dictionary containg name and color code as key value pair
clrschms = {
    '1. Default White': '000000.FFFFFF',
    '2. Greygarious Grey': '83406A.D1D4D1',
    '3. Lovely Lavender': '202B4B.E1E1FF',
    '4. Aquamarine': '5B8340.D1E7E0',
    '5. Bold Beige': '4B4620.FFF0E1',
    '6. Cobalt Blue': 'ffffBB.3333aa',
    '7. Olive Green': 'D1E7E0.5B8340',
}
themechoice = StringVar()
themechoice.set('1. Default White')
for k in sorted(clrschms):
    themesmenu.add_radiobutton(label=k, variable=themechoice, command=theme)

# About menu
aboutmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label="About", menu=aboutmenu)
aboutmenu.add_command(label="About", command=about)
aboutmenu.add_command(label="Help", command=help_box)

root.config(menu=menubar)

# shortcut bar and line number
shortcutbar = Frame(root, height=25)
icons = ['new_file', 'open_file', 'save', 'cut', 'copy', 'paste', 'undo', 'redo', 'on_find', 'about']
for i, icon in enumerate(icons):
    tbicon = PhotoImage(file='/Users/azinheidari/Desktop/icons/' + icon + '.gif')
    cmd = eval(icon)
    toolbar = Button(shortcutbar, image=tbicon, command=cmd)
    toolbar.image = tbicon  # http://effbot.org/tkinterbook/photoimage.htm
    toolbar.pack(side=LEFT)
shortcutbar.pack(expand=NO, fill=X)

lnlabel = Label(root, width=2, bg='antique white')
```

```
lnlabel.pack(side=LEFT, fill=Y)

# Text widget and scrollbar widget
#######################################
textPad = Text(root, undo=True)
textPad.pack(expand=YES, fill=BOTH)
scroll = Scrollbar(textPad)
textPad.configure(yscrollcommand=scroll.set)
scroll.config(command=textPad.yview)
scroll.pack(side=RIGHT, fill=Y)


# Info Bar
infobar = Label(textPad, text='Line: 1 | Column:0')
infobar.pack(expand=NO, fill=None, side=RIGHT, anchor='se')

# context popup menu
cmenu = Menu(textPad, tearoff=0)
for i in ('cut', 'copy', 'paste', 'undo', 'redo'):
   cmd = eval(i)
   cmenu.add_command(label=i, compound=LEFT, command=cmd)
cmenu.add_separator()
cmenu.add_command(label='Select All', underline=7, command=select_all)
textPad.bind("<Button-3>", popup)

####################################################
# Add events
# Binding events
textPad.bind('<Control-N>', new_file)
textPad.bind('<Control-n>', new_file)
textPad.bind('<Control-O>', open_file)
textPad.bind('<Control-o>', open_file)
textPad.bind('<Control-S>', save)
textPad.bind('<Control-s>', save)
textPad.bind('<Control-A>', select_all)
textPad.bind('<Control-a>', select_all)
textPad.bind('<Control-f>', on_find)
textPad.bind('<Control-F>', on_find)
textPad.bind('<KeyPress-F1>', help_box)

textPad.bind("<Any-KeyPress>", update_line_number)
textPad.tag_configure("active_line", background="ivory2")


import sys
```

```
def button1():
   novi = Toplevel()
   canvas = Canvas(novi, width = 1000, height = 800)
   canvas.pack(expand = YES, fill = BOTH)
   gif1 = PhotoImage(file = '1414.gif')
                          #image not visual
   canvas.create_image(50, 10, image = gif1, anchor = NW)
   #assigned the gif1 to the canvas object
   canvas.gif1 = gif1


button1 = Button(root,text ='Fall Clearance Guide',command = button1, height=5,
width=20).pack()

def button2():
    novin = Toplevel()
    canvas = Canvas(novin, width = 600, height = 650)
    canvas.pack(expand = YES, fill = BOTH)
    gif2 = PhotoImage(file = '1313.gif')
    canvas.create_image(50, 10, image = gif2, anchor = NW)
    canvas.gif2 = gif2

button2 = Button(root,text ='Swing Hazard Guide',command = button2, height=5,
width=20).pack()


def button6():
   novin = Toplevel()
   canvas = Canvas(novin, width=1250, height=455)
   canvas.pack(expand=YES, fill=BOTH)
   gif6 = PhotoImage(file='1515.gif')
   canvas.create_image(50, 10, image=gif6, anchor=NW)
   canvas.gif6 = gif6


button6 = Button(root, text='Roof Geometry Guide', command=button6, height=5,
width=20).pack()


def callback():
   class SampleApp(tk.Tk):
      def __init__(self):
         tk.Tk.__init__(self)
         self.title("PFAS App")
         self.geometry('600x850')
```

```
        yscroll = tk.Scrollbar(self, orient=tk.VERTICAL)
        xscroll = tk.Scrollbar(self, orient=tk.HORIZONTAL)
        yscroll.pack(side=tk.RIGHT, fill=tk.Y)
        xscroll.pack(side=tk.BOTTOM, fill=tk.X)

        self.canvas = tk.Canvas(self)
        self.canvas.pack(fill=tk.BOTH, expand=True)
        self.canvas['yscrollcommand'] = yscroll.set
        self.canvas['xscrollcommand'] = xscroll.set
        yscroll['command'] = self.canvas.yview
        xscroll['command'] = self.canvas.xview

        frame = tk.Frame(self.canvas)
        self.canvas.create_window(4, 4, window=frame, anchor='nw')  # Canvas equivalent of
pack()
        frame.bind("<Configure>", self._on_frame_configure)

        self.label_1 = ttk.Label(frame, width=30, text="Roof Length in feet (plan view)")
        self.entry2 = ttk.Entry(frame, width=10)
        self.button2 = ttk.Button(frame, text="Enter", command=self.on_button)
        self.label_2 = ttk.Label(frame, width=30, text="Roof Width in feet (plan view)")
        self.entry3 = ttk.Entry(frame, width=10)
        self.label_3 = ttk.Label(frame, width=60,
                        text="The coordinates of the working platform in [x inches, y inches]")
        self.entry4 = ttk.Entry(frame, width=10)
        self.label_4 = ttk.Label(frame, width=63,
                        text="The coordinates of the initial anchor point in [x inches, y inches]")
        self.entry5 = ttk.Entry(frame, width=10)
        self.label_5 = ttk.Label(frame, width=63,
                        text="The greatest distance you consider for your anchor point in inches
(working area)")
        self.entry6 = ttk.Entry(frame, width=10)
        self.label_6 = ttk.Label(frame, width=60, text="The PFAS user's weight in pounds")
        self.entry7 = ttk.Entry(frame, width=10)
        self.label_7 = ttk.Label(frame, width=60, text="Roof height in feet (RH)")
        self.entry8 = ttk.Entry(frame, width=10)
        self.label_8 = ttk.Label(frame, width=75, text="The distance of the edge of the roof to
the top of the highest allowable obstruction in feet (FH)")
        self.entry9 = ttk.Entry(frame, width=10)
        self.label_9 = ttk.Label(frame, width=60,
                        text="The worker's height in feet or D for a default value (6 ft tall
worker)")
        self.entry10 = ttk.Entry(frame, width=10)
        self.label_10 = ttk.Label(frame, width=60, text="Lanyard length (LY) in feet or D for a
default value (3 feet)")
        self.entry11 = ttk.Entry(frame, width=10)
```

```
        self.label_11 = ttk.Label(frame, width=60,
                       text="Lanyard/lifeline stretch (DD) in feet or D for a default value (3.5
feet)")
        self.entry12 = ttk.Entry(frame, width=10)
        self.label_12 = ttk.Label(frame, width=60,
                       text="D-ring shift (MASD) in feet or D for a default value (1 foot)")
        self.entry13 = ttk.Entry(frame, width=10)
        self.label_13 = ttk.Label(frame, width=60,
                       text="Harness stretch (XW) in inches or D for a default value (12
inches)")
        self.entry14 = ttk.Entry(frame, width=10)
        self.label_14 = ttk.Label(frame, width=65,
                       text="Are there any vertical surface nearby that may obstruct the fall?
Answer with Y and N")
        self.entry15 = ttk.Entry(frame, width=10)
        self.label_15 = ttk.Label(frame, width=60,
                       text="The coordinates of the swing obstruction in [x inches, y inches]")
        self.entry16 = ttk.Entry(frame, width=10)
        self.entry17 = ttk.Entry(frame, width=10)
        self.label_17 = ttk.Label(frame, width=60,
                       text="The structural members (truss/rafter) distances in inches (O.C.)")

        self.label_1.pack()
        self.entry2.pack()
        self.label_2.pack()
        self.entry3.pack()
        self.label_3.pack()
        self.entry4.pack()
        self.label_4.pack()
        self.entry5.pack()
        self.label_5.pack()
        self.entry6.pack()
        self.label_6.pack()
        self.entry7.pack()
        self.label_7.pack()
        self.entry8.pack()
        self.label_8.pack()
        self.entry9.pack()
        self.label_9.pack()
        self.entry10.pack()
        self.label_10.pack()
        self.entry11.pack()
        self.label_11.pack()
        self.entry12.pack()
        self.label_12.pack()
        self.entry13.pack()
```

```
    self.label_13.pack()
    self.entry14.pack()
    self.label_14.pack()
    self.entry15.pack()
    self.label_15.pack()
    self.entry16.pack()
    self.label_17.pack()
    self.entry17.pack()
    self.button2.pack()

def printt(self, xy):
    #self.label_16 = ttk.Label(width=60, text=xy)
    #self.label_16.pack()
    textPad.insert(END, xy)

def _on_frame_configure(self, event=None):
    self.canvas.configure(scrollregion=self.canvas.bbox("all"))

def on_button(self):
    RoofL2 = float(self.entry2.get()) * 12
    numb = float(self.entry17.get())
    RoofW2 = float(self.entry3.get()) * 12
    matrix = []
    for i in range(1, int(RoofL2) + 1):
        for j in range(1, int(RoofW2 + 1)):
            matrix.append((i, j))
    # print len(matrix)
    dim = eval(self.entry4.get())
    dim2 = eval(self.entry5.get())
    KKK = eval(self.entry6.get())

    def euclidean_distance(x, y):
        return sqrt(pow(x[0] - y[0], 2) + pow(x[1] - y[1], 2))

    dis = euclidean_distance(dim, dim2)
    resultmatrix = []
    for i in range(0, len(matrix)):
        if euclidean_distance(matrix[i], dim2) <= int(KKK):
            resultmatrix.append(matrix[i])

    AAAA = len(resultmatrix)
    self.printt(AAAA)

    mylist = []
    for i in range(0, len(resultmatrix)):
        mylist.append(i)
```

```
azinlist = random.choice(mylist, 8000)  # number of population
# print azinlist
azinlis = random.choice(mylist, 8000)

finalists2 = []
finalists = []
for i in range(0, 8000):
    finalists.append(resultmatrix[azinlist[i]])
    finalists2.append(resultmatrix[azinlis[i]])
#self.printt(finalists)
#self.printt(finalists2)
UW = float(self.entry7.get())
if (UW < 130):
    self.printt('user weight is less than the minimum allowed weight for this system')
elif (130 > UW > 310):
    self.printt('user weight is more than the maximum allowed weight for this system')
else:
    self.printt('user weight is within allowed weight for this system')
RF = float(self.entry9.get()) * 12
WH = self.entry10.get()
if (WH == "D"):
    WH = 6
LY = self.entry11.get()
if (LY == "D"):
    LY = 3
# A standard lanyard length is 36 inches.
DD = self.entry12.get()
if (DD == "D"):
    DD = 3.5
# Deceleration Distance (DD) is XPEA + XL
DS = self.entry13.get()
if (DS == "D"):
    DS = 1
if (WH == 6):
    BDH = 5
else:
    BDH = float(WH) - 1
# BDH is the back D-ring height/HI
HS = self.entry14.get()
if (HS == "D"):
    HS = 12
SF = 2

# clearance margin/CM
CA = float(LY) + float(DD) + float(DS) + float(BDH) + (float(HS)/ 12) + SF
```

```
        CI = float(CA) * 12
        #print CA
        if CA < 18.5:
            self.printt("The connecting device type should be self-retracting lanyard")
        else:
            self.printt("The connecting device type should be self-retracting lanyard OR personal
shock absorber")
        JJ = self.entry15.get()
        JJJ = eval(self.entry16.get())
        FinalList = []
        FinalList2 = []
        for i in range(0, 8000):
            xx, yy = finalists[i]
            iii = float(yy) + float(RF)
            if CI < iii:
                FF = (float(LY) * 12) + (dim[1] + 60) - yy
                if FF < 48:
                    if (JJ == "Y"):
                        Q = abs(xx - dim[0])
                        W = abs(JJJ[0] - xx)
                        S = abs(yy - dim[1])

                        p1, p2, p3, p4 = Point(0, 0), Point(0, float(S)), Point(-float(Q), 0),
Point(float(W), 0)
                        l1, l2, l3 = Line(p1, p2), Line(p2, p3), Line(p2, p4)

                        G = 180 - round(math.degrees(l1.angle_between(l2)), 0)
                        K = 180 - round(math.degrees(l1.angle_between(l3)), 0)
                        # print ('the angle in degrees between axis and working platform:', G)
                        # print ('the angle in degrees between axis and obstruction:', K)
                        GG = 2 * math.sin(math.radians(G / 2)) * float(LY) * 12
                        KK = 2 * math.sin(math.radians(K / 2)) * float(LY) * 12
                        SFD = math.sqrt(abs((math.pow(float(GG), 2)) - (math.pow(float(Q), 2))))
                        # SFD=Swing Fall Distance
                        A = math.sqrt(abs((math.pow(float(KK), 2)) - (math.pow(float(W), 2))))
                        SDD = SFD - A
                        # print ('Swing Drop Distance:', SDD)
                        RG = float(FF) + float(HS) + float(SDD)
                        if RG < 48:
                            FinalList.append(finalists[i])  # print ('There is no swing hazard')
                    else:
                        FinalList.append(finalists[i])

        #self.printt(FinalList)
        #printt(len(FinalList))
```

```
for i in range(0, 8000):
    xx, yy = finalists2[i]
    iii = float(yy) + float(RF)
    if CI < iii:
        FF = (float(LY) * 12) + (dim[1] + 60) - yy
        if FF < 48:
            if (JJ == "Y"):
                Q = abs(xx - dim[0])
                W = abs(JJJ[0] - xx)
                S = abs(yy - dim[1])

                p1, p2, p3, p4 = Point(0, 0), Point(0, float(S)), Point(-float(Q), 0),
Point(float(W), 0)
                l1, l2, l3 = Line(p1, p2), Line(p2, p3), Line(p2, p4)

                G = 180 - round(math.degrees(l1.angle_between(l2)), 0)
                K = 180 - round(math.degrees(l1.angle_between(l3)), 0)
                # print ('the angle in degrees between axis and working platform:', G)
                # print ('the angle in degrees between axis and obstruction:', K)
                GG = 2 * math.sin(math.radians(G / 2)) * float(LY) * 12
                KK = 2 * math.sin(math.radians(K / 2)) * float(LY) * 12
                SFD = math.sqrt(abs((math.pow(float(GG), 2)) - (math.pow(float(Q), 2))))
                # SFD=Swing Fall Distance
                A = math.sqrt(abs((math.pow(float(KK), 2)) - (math.pow(float(W), 2))))
                SDD = SFD - A
                # print ('Swing Drop Distance:', SDD)
                RG = float(FF) + float(HS) + float(SDD)
                if RG < 48:
                    FinalList2.append(finalists[i])  # print ('There is no swing hazard')
            else:
                FinalList2.append(finalists[i])

#self.printt(FinalList2)
#printt(len(FinalList2))

FFF = list(set(FinalList2).intersection(FinalList))
#self.printt(FFF)
self.printt(len(FFF))

#FFFF = [i for i in FFF if 380 >= i[0] >= 320]


self.printt(FFF)
x = dim[0]
y = dim[1]
z = dim2[0]
```

```
        w = dim2[1]
        v = JJJ[0]
        g = JJJ[1]
        uu = z - (KKK / 2) - 50
        uuu = z + (KKK / 2) + 50
        for i in range(uu, uuu, int(numb)):
            p2 = [0, RoofW2]
            p1 = [i, i]
            # plt.newline(p1, p2)
            plt.plot(p1, p2, linewidth=2, color='red')

        #plt.Circle((z, w), KKK)
        plt.scatter(*zip(*FFF), c='c')
        plt.scatter(x, y, c='m')
        plt.scatter(z, w, c='y')
        plt.scatter(v, g, c='k')
        #plt.text(0, (-20), 'text')
        txt = "The units for the table axis are inches" \
              "* The anchor points can be installed on a truss section located in the blue area" \
              "** Shows a dynamic point on the radius of the lanyard length from the initial anchor
point"
        plt.figtext(0.5, 0.01, txt, wrap=True, horizontalalignment='center', fontsize=8)
        #circle = plt.Circle((x, y), radius=float(LY)*12 , color='r', fill=False)
        #def show_shape(patch):
            #ax = plt.gca()
            #ax.add_patch(patch)
            #plt.axis('scaled')
            #plt.show()
        #show_shape(circle)
        cyan_patch = mpatches.Patch(color='cyan', label='The anchor points results *')
        m_patch = mpatches.Patch(color='purple', label='The working platform **')
        yellow_patch = mpatches.Patch(color='yellow', label='The initial anchor point')
        k_patch = mpatches.Patch(color='black', label='The barrier')
        r_patch = mpatches.Patch(color='red', label='The truss')
        plt.legend(handles=[k_patch, yellow_patch, cyan_patch, m_patch, r_patch])
        plt.show(block=True)

    app = SampleApp()
    app.mainloop()


b = Button(root, text="START", command=callback)
b.pack()


root.mainloop()
```