MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

ROBOTICS
514 BENJAMIN ST.
ANN ARBOR, MI 48104

TO: Alex Souris, Chief Technology Officer

FROM: Connor Williams (willcon@umich.edu), Research Engineer

SUBJECT: Development of Real-Time Automatic Muscle Fascicle Tracking Software

DATE: 28 February 2023

FOREWORD

Logiteque Software is interested in expanding its reach to exoskeleton technology to offset heavy work performed by people. Specifically, the company is looking into controlling ankle exoskeletons with real time analysis of ultrasound images of the calf muscles. I was asked to research methods previously taken to solve this issue and provide suggestions for how to proceed with the project. This has been completed and is ready to be shared. The purpose of this report is to state my findings and conclusions regarding automated ultrasonography.

very clear, nice!

SUMMARY

if you aren't going to refer to 'fascicle' as 'fiber' then why use the paranthesis?

There are three main categories of muscle fascicle (fiber) tracking algorithms: heuristic, optical flow, and deep learning. The first approach uses classical computer vision techniques, such as convolutions and line detectors, to locate and measure the fascicles. This approach can be very fast, which is critical since the software will run on a wearable system, but it can produce incorrect measurements. How much allowable error the system can tolerate is still unknown, but more work can be done to determine this.

The most popular approach is optical flow. This technique analyzes the motions of the ultrasound images over time and tracks specific points of interest. By selecting the endpoints of a fascicle, this algorithm can follow them through time. This is the method used by UltraTrack, the muscle fascicle tracking software most used to assess the accuracy of new algorithms. However, if the endpoints of the fascicle leave the frame, which regularly occurs during walking, the algorithm must extrapolate outside of what the ultrasound image contains. This approach also requires a human to select a fascicle before starting, since it only tracks points and does not truly detect fascicles, and requiring users to look at live ultrasound feed and select a fascicle may be cumbersome.

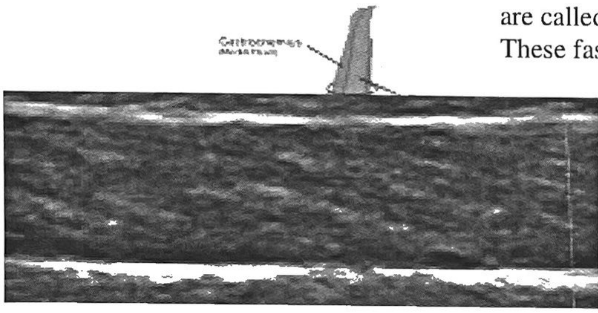
The final approach is deep learning. This approach involves gathering vast quantities of data, manually measuring the lengths of fascicles in every image, and training a machine learning model to measure in the same way a human would. If the model is designed to be simple, it can process very quickly. However, a simple model can yield simple behavior, and it may accurately measure the fascicle lengths. It also cannot transfer to different ultrasound systems, so if the company wishes to use cheaper systems to reduce the cost for the customers, we would need to gather much more data and train a new model.

start on new page

INTRODUCTION

Dr. Pridham and Dr. Stirling developed an exoskeleton control algorithm that utilizes tracked changes in calf muscle architecture to control ankle exoskeletons [1]. These architectural changes highlight when the muscle cells are using the most energy (ATP), and so the exoskeleton activates most during high energy usage. They hope this algorithm can reduce the metabolic cost of walking more than other algorithms.

Their primary muscle of interest is the soleus (shown in Figure 1a), whose purpose is to plantarflex the foot during walking. The soleus is attached to the knee on the upper end of the calf and the Achilles tendon on its lower end. The deep and superficial boundaries of the soleus are called aponeuroses, which are connected through fascicles (Figure 1b). These fascicles are what generate force in muscles, and Hill et al. developed a model that estimates energy usage based on their architecture [2]. Most importantly, if one can measure the length and velocities of these fascicles, one can implement Pridham and Stirling's algorithm.



center

Aponeuroses

Figure 1: (a) Diagram showing where the soleus is in relation to the lower leg, adapted from [3]. (b) Ultrasound image of the soleus, adapted from [4]. The left side is up the leg, the right is towards the foot, the top is towards the back of the leg, and the bottom is towards the front of the leg. The fascicles are the fine angled lines, while the aponeuroses are the bold horizontal lines.

PRIOR APPROACHES TO AUTOMATICAL FASCICLE TRACKING

This section describes three common algorithm archetypes that were developed to automatically measure the lengths of fascicles from ultrasound images.: heuristics, affine flow, deep learning.

Heuristics

As mentioned before, heuristics utilize classical computer vision techniques to extract features. One such technique is edge detection, where the computer can use simple procedures, such as Sobel filtering, to highlight edges in an image. From there, one could feed those edges into the Hough Line Detector to obtain equations that describe those edges. Such algorithms are very straightforward and easy to understand since they are typically built from intuitions.

Citation?

Seynnes et al. developed the Simple Muscle Architecture Analysis (SMA) that automatically detects both aponeuroses and fascicles in ultrasound images [5]. It takes very straightforward steps to do so: blur the image to remove noise, normalize the image for consistency across images, enhance tube-like structures, detect edges for line detection, and so on.
The SMA algorithm produces measurements that are similar to those obtained by professional examiners. Nearly all data from the study lies within a 95% limit of agreement according to their results. However, as mentioned in the paper, this algorithm is weak to changes in image quality. Such changes may include blurriness, average brightness, and contrast. The authors mention the

algorithm could work for analyzing images during walking, but the image quality criteria must always be met. People using such a system in their day-to-day lives may not have that necessary consistency, so a purely heuristic algorithm may not but sufficient.

Affine Flow

Affine flow is an extension of Lucas-Kanade optical flow [6]. Lucas-Kanade optical flow assesses the spatial and temporal gradients through an image sequence, which generates a field describing where bright points in a video have moved. The affine flow extension fits an affine transformation matrix to that field to describe pixel motion in four simple parameters: dilation, rotation, shear, and translation. An affine transformation matrix is useful because it can predict the motion of points not shown in the image. This means if a fascicle periodically leaves the ultrasound sensor's field of view, its motion can be extrapolated beyond the image bounds.

Affine flow is highly supported by research. Gillett at al. compared an implementation of this algorithm named "UltraTrack" [7] to three professional human examiners [8]. Each examiner analyzed three ultrasound videos three times each, while the UltraTrack software did the same. The researchers assessed within-examiner reliability and between-examiner reliability with this data using the coefficient of multiple determination (CMD). The CMD was between 0.98 and 0.99, which is remarkably high. Such a CMD indicates that UltraTrack, and thus, affine flow, is a reliable method to automatically measure fascicle lengths.

what do these mean?

Affine flow computes average motion in an image, not motion of individual pixels, so it is prone to drifting. This drifting is a severe issue and historically has prohibited this algorithm's use in real-time applications [9] and must be solved in creative ways. UltraTrack mitigates drift using "key-frame correction". After a video has been recorded, the user may select similar frames in the sequence, such as whenever the heel of the foot touches the ground, to use as anchor points for interpolation. While this key-frame correction solves the drifting issue in post-recording settings, this is not useful for live analysis.

include a visual for this?

Affine flow methods also require a region of interest (ROI) and starting fascicle to begin computation, since it does not truly track fascicles. This limitation would mean an exoskeleton user must view their ultrasound feed from their leg and select an ROI and initial fascicle before using the exoskeleton. Affine flow in conjunction with another algorithm may resolve this issue.

Deep Learning

Deep learning is a relatively new approach to muscle fascicle tracking. It takes pre-measured ultrasound footage, automatically tunes up to hundreds of billions of parameters, and tests itself against novel ultrasound footage. In essence, the aim of deep learning is for the computer to measure fascicles the same way a human would.

The use of deep learning for automated fascicle tracking was first performed by Rosa et al. [4]. The researchers recorded ultrasound data from six participants in five difference conditions: constrained ankle flexion, free ankle flexion, calf raises, walking, and random motions. The data was then labeled using UltraTrack and processed into a deep learning model.
The results of Rosa's model were not great. The correlation coefficient (r) during walking was low at 0.22, meaning there was a low correlation between the model's output for fascicle length and the true fascicle length. The other metric used was root-mean-square-error (RMSE), which was 2.36±1.03 mm. They did not use other metrics, such as peak error, to quantify performance, so it is unknown how much their model may "jump".

define

Deep learning models frequently do not generalize well. Rosa's approach, along with an approach by Cronin et al. [10], utilized data from high-end ultrasound sensors that cost over $10,000. Logiteque Software must develop an affordable exoskeleton for this product to be a success, and cheaper systems have significantly different image quality than the ones used in the literature.

SUGGESTED APPROACH TO AUTOMATIC FASCICLE TRACKING

Table 1 summarizes the pros and cons of each archetype, with major attributes highlighted.

	Pros	**Cons**
Heuristic: Using simple computer vision techniques (convolutions, frequency domain analysis, line detections) to extract features	Can be very fast	Depending on the components, the pipeline may need tuned for every user
	Very intuitive approach	Can be very jumpy
	Flexible approach, easy to add and remove pipeline components	Sensitive to changes in brightness and sharpness
Affine flow: Using the movement patterns between frames to predict where points of interest have moved to	Robust to changes in imaging technology	Could be computationally expensive
	Works even when fascicle endpoints are not visible	Fascicle endpoints can drift significantly, must be corrected with every step
	Very smooth tracking, little filtering is needed	Requires an initialized fascicle
	Greatly supported in research that studies its reliability and accuracy	
Deep learning: Gathering large quantities of labeled ultrasound data, designing a model, and training the model to mimic human measurements	If the model is small enough, it can execute very quickly	Can be extremely sensitive to other imaging technologies
	Simplifies the problem from the development standpoint - give the computer a ton of examples and let it figure it out	Model performance is relatively random
	Data sensitivity can be reduced by augmenting the training data	Can magnify biases found in human-labeled data

remove

consider coloring the entire row (not just highlighting text)

~~Table 1: Summary of archetypes.~~ duplicate diagram label

Stick to one term

An ideal automatic fascicle tracking algorithm could meld multiple approaches together to cancel out weaknesses using other strengths. One such algorithm could be the fusion of heuristic tools with affine flow. While affine flow is supported by research and very steady, its drifting makes it unusable for real-time analysis. Heuristics lack steadiness but produce more globally oriented measurements.

One such fusion could operate as follows:

1. Retrieve the first video frame from the ultrasound sensor
2. Define a region of interest by detecting aponeuroses with a heuristic
3. Detect a single fascicle with a heuristic and record the end points
4. Retrieve the next video frame from the ultrasound sensor
5. Use affine flow to update the fascicle endpoint positions
6. Detect aponeuroses with a heuristic
7. Constrain the fascicle endpoints to the aponeuroses
8. Repeat steps 4-8 until the video sequence ends

Steps 1-3 solve affine flow's first issue of needing a human-determined region of interest and initial fascicle. Steps 5-6 eliminate the drift produced by affine flow's update. This procedure theoretically resolves the primary issues of heuristic and affine flow methods.

The major flaw of melding approaches is the additional computation power necessary to perform the larger algorithm. An ideal final exoskeleton product should be lightweight, and so the computer performing the tracking should be very small. Additional research must be done to implement method-fusing algorithms and to assess both reliability and speed.

CONCLUSION

The key to optimal exoskeleton control may lay in actuating when energy usage of muscles is at its peak. A model developed by Dr. Pridham and Dr. Stirling utilizes the lengths of muscle fascicles to approximate energy usage and relates that to a torque profile that could be used to control an ankle exoskeleton. Heuristics, affine flow, and deep learned are previous attempts at automatic muscle fascicle tracking from ultrasound images; however, each one has major flaws that currently render them unusable for real-time analysis. A path forward may be to fuse multiple methods together to counteract specific weaknesses in each. → mention that this will require solving the computation problem

REFERENCES

[1] P. S. Pridham and L. Stirling, "Ankle exoskeleton torque controllers based on soleus muscle models," *PLoS One,* 2023.

[2] A. V. Hill, "The heat of shortening and the dynamic constants of muscle," *The Royal Society,* vol. 126, no. 843, 1938.

[3] westcoastsci, "westcoastsci," 6 June 2018. [Online]. Available: https://westcoastsci.com/blog/soleus-the-forgotten-muscle-for-runners/. [Accessed 28 Februrary 2023].

[4] L. G. Rosa, J. S. Zia, O. T. Inan and G. S. Sawicki, "Machine learning to extract muscle fascicle length changes from dynamic ultrasound images in real-time," *PLoS ONE,* 2021.

[5] O. R. Seynnes and N. J. Cronin, "Simple Muscle Architecture Analysis (SMA): An ImageJ macro tool to automate measurements in B-mode ultrasound scans," *PLoS ONE,* 2020.

Logiteque Software
2300 Hayward St
Ann Arbor, MI 48109

TO: Alex Souris, Chief Technology Officer (CTO)
FROM: Vishwa Patil
SUBJECT: Constructing an application that showcases people's blogs about their lifestyles
DATE: 02/06/2023

change format

Foreword

Moving to a new city can be one of the most challenging times in a person's life. BlogLocal is a platform for people to share their experiences about their day-to-day lives in a certain city, providing new members of the community with local tips and tricks. The intended audience is people of all ages and backgrounds in search of a community in a new location. We will rely mainly on people's goodwill and desire to share their experiences to gain blog posts, meaning that our platform must be extremely well maintained and easy to navigate. As users of forums such as Reddit, our team has identified that many of these forums are not designed aesthetically, and can often be challenging to understand as a new user. By retaining a clear focus on a core feature set, we will make it easy for users to surface the information that really matters: a personal view into day-to-day life in different areas. This memo discusses the findings of my research and development process and proposes how our unique user interface on the platform helps people into moving to a new town.

this seems more relevant for summary section

what research? need to provide problem statement explaining what you were tasked w/.

Summary

Based on the findings and the whole development process, BlogLocal superseded expectations. The findings showed that about 60% of the users were able to successfully move into a new neighborhood that was suggested by a blogger on the platform. Of the total users, there was a 70% retention rate of people coming back to view other blogs on BlogLocal. Specifically, they noted that they appreciated the unique and easy-to-follow UI that helped them navigate through blogs and upvote ones they really enjoyed. → should lead into next section (introduce coming topics)

Introduction

It's extremely important for individuals to better integrate into their new community when they move to a new city. If this isn't done in time or at all, it's super detrimental for them to fully live the way they want to live. They need to understand the local culture and shortcuts in order to fully comprehend what they need in order to establish a routine and schedule for themselves. Additionally, it helps these individuals appreciate diversity in their new city and country as a whole. This is important as it helps others be mindful and aware of differences among citizens and natives in the area. If these new individuals are assuming a high status role in this new community, they will definitely be interacting with those who they have not met before or worked with in the past. By being knowledgeable about this information, more people will be able to live in harmony with one another.

Seems subjective. hypothetical. Try to cite something here!

Moving forward with solutions to this problem, one that stood out to the team is utilizing the power of blogs and infusing that with a lifestyle format. This helps people gain ease of access to understandable and comprehensible information that aids them in assimilating them into their new area of residence. By implementing an easy to use UI on a mobile and desktop platform, this is helpful for them to view this information on their personal devices wherever they are. It'll educate them on certain landmarks that they travel to or pass by on the way to work or home, so they can take note of them and write down any information they need.

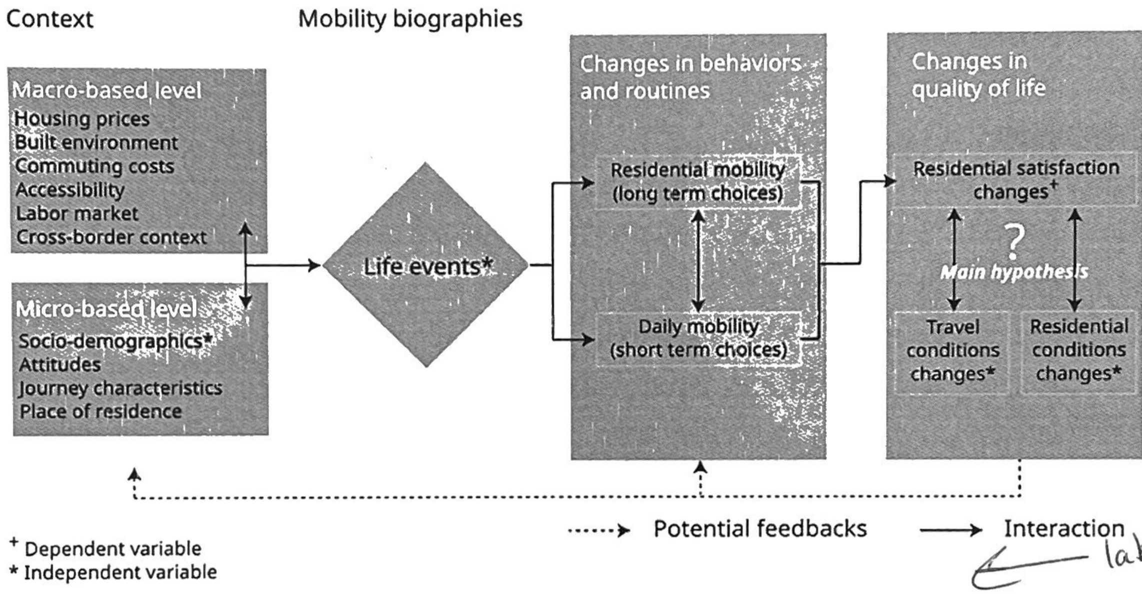
Commute Troubles

A specific group of people that I wanted to target for this application were people that have to commute long distances. They often have to figure out their own way to maneuver through certain routes and how to structure their day so that they are able to get to work at a reasonable time. Additionally, there has been a lot of dissatisfaction with

maybe instead of alluding to their route, talk about being in an unfamiliar area? Routes seem better suited for a map app.

this specific subtype of citizens. They have too many variables to control and it results in them becoming overwhelmed with trying to decide what to do. The figure below showcases the different factors and variables that influence the lifestyles of these commuters.

Design



label your figure

Our research and development team will be utilizing this framework to construct the layout and schema for what information the blogs should include. We decided there needs to be a centralized and uniform design of all the blogs because they all cannot differ in content. This would overwhelm and confuse viewers on the platform because they wouldn't know what blogs to view in order to understand a lifestyle in a specific urban or rural setting.

do you mean 'can'?

From observing this diagram, it can be seen that life events are the most volatile variable that impacts residential and daily mobility, residential satisfaction changes, and travel conditions. These have to be incorporated into the blogs content as much as possible as it will provide a very comprehensive viewpoint of the lifestyle. Specifically, if the blogger is okay with it, then they can share as much information as they can which will inevitably help the viewers gain as much knowledge and perspective as possible.

I got confused here: is the app for sharing travel tips, or lifestyle stuff?

Moreover, “moving home is often accompanied by major life events, such as forming or dissolving a partnership, having children or losing a loved one, and these events usually influence people’s sense of wellbeing. Residential histories are, therefore, closely interwoven with personal satisfaction and wider webs of wellbeing.” [2] By providing a raw and real look at someone’s personal life and story behind moving to a new city or home helps those gain an introspective perspective and understand whether they want to move in the first place or not.

nice citation!

Conclusion

From the results and findings, it is imperative that BlogLocal’s purpose is to help those gain information about a new area of residence. It goes a long way in terms of increasing satisfaction rates and building up multiple perspectives of a city or rural area which in turn develops into a cohesive view.

So this memo is about finding the app's purpose?

Works Cited

Carpentier-Postel, Samuel, et al. “Changes in Residential Satisfaction after Relocation: The Effects of Commuting. A Case Study of Luxembourg Cross-Border Workers.” *Case Studies on Transport Policy*, vol. 10, no. 3, 2022, pp. 1754–1766., https://doi.org/10.1016/j.cstp.2022.07.006. [1]

Nowok, Beata, et al. “Linking Residential Relocation Desires and Behaviour with Life Domain Satisfaction.” *Urban Studies*, vol. 55, no. 4, 2016, pp. 870–890., https://doi.org/10.1177/0042098016665972. [2]

HILLCOAT-NALLÉTAMBY, S., & OGG, J. (2014). Moving beyond ‘ageing in place’: Older people's dislikes about their home and neighbourhood environments as a motive for wishing to move. *Ageing & Society, 34*(10), 1771-1796. doi:10.1017/S0144686X13000482. [3]

Nisic, N., & Petermann, S. (2013). New City = new friends? the restructuring of social resources after relocation. *Comparative Population Studies*, *38*(1). https://doi.org/10.12765/cpos-2013-08 [4]

LOGITEQUE SOFTWARE
2300 Traverwood Dr,
Ann Arbor, MI 48105

MEMORANDUM

To: Alex Souris, Chief Technology Officer (CTO)

From: Pingbange Hu (pbb@umich.edu), Research Engineer (RE)

Date: 28th February 2023

Subject: Exploring Solutions for Improving the Efficiency of the Network Failure Detection

Overview

Foreword

The current network failure detection algorithm in our implementation is unable to cope with the growing size of our clients' network clusters, in particular, the current implementation involves solving large mixed-integer linear programming (MILP) problems, which have become too large for the state-of-the-art commercial solvers like `Gurobi` [1], `SCIP` [2], and `Cplex` [3]. Hence, Chief Technology Officer (CTO) Alex Souris assigned me to investigate in possible solutions to speed up the detection algorithm.

It is crucial to either come up with new detection methods or improve the speed of the MILP solver to address the declining performance of our product. This memo examines possible solutions by examining new detection algorithms developed in academia and industry and recent improvements in MILP solvers.

this could be more concise. I was confused reading this

Summary

This memo focuses on the issue of decreasing efficiency in our network failure detection algorithm due to the increasing size of our clients' network clusters. The current algorithm involves solving large mixed-integer linear programming (MILP) problems, which are too complex for existing commercial solvers: As reported, it takes more than 10 minutes to finish one detection cycle, but the ideal completion time, based on previous surveys, is under 3 minutes.

citation?

To overcome this challenge, this memo examines potential solutions by reviewing recent advancements in MILP solvers and discussing potential new detection algorithms ~~by~~ through the lens of theoretical computer science (TCS). The aim of the memo is to provide suggestions and recommendations for enhancing the performance of our network failure detection algorithm, and the following are two promising directions to explore: ML-driven solvers and randomized detection algorithms.

1 Introduction

1.1 Monitoring Network Failures

The task of network monitoring has been a persistent challenge, and it has become increasingly important with the growth of the internet. Many network issues, including disconnections, are investigated through the use of graph theory, which involves modeling cuts in a graph. A number of simple algorithms, such as breadth-first search [4], can be employed to assess the connectivity of a graph. These algorithms traverse the entire connected component containing the graph's starting point, and by examining whether this component encompasses the entire graph (or a significant portion of it), we can determine whether a severe network failure has occurred.

Defining the term "severe" is crucial in the context of network monitoring since it can greatly severe? impact the number of falsely reported failures resulting from only a few disconnections. To avoid this issue, a formal definition of severe failure is necessary. In practical terms, given a network of size n and a *tolerance level* $\varepsilon \in (0,1)$, a network failure is declared if the connected components of the network can be grouped into two parts, each with a size of at least εn. For example, consider a network of size n is broken into three parts, each with size $(1-\varepsilon)n, \varepsilon n/2, \varepsilon n/2$, respectively. In this case, we would like to declare a network failure, since we can group these components into $\{(1-\varepsilon)n\}$ and $\{\varepsilon n/2, \varepsilon n/2\}$, each with size at least εn. This problem is referred to as the "balanced network failure" problem in the academic community. With the problem now defined, the next step is to discuss potential solutions.

1.2 The Optimizations and Solvers Approach

Our current approach involves formulating the balanced connectivity problem as a MILP using the network flow technique [5, §8]. This formulation is then passed to modern optimization solvers like `Gurobi` [1], `SCIP` [2], and `Cplex` [3]. However, this check needs to be performed regularly, ideally every 5 minutes. Unfortunately, the current method takes 10 minutes to run, which is slower than the acceptable level. ~~Therefore~~, we will explore several data-driven solvers that have the potential to significantly speed up our current implementation in this short memo.

1.3 The Detection Sets Approach

The problem of balanced connectivity is a well-researched topic in the field of fine-grained complexity theory, particularly in the theoretical computer science community over the past few decades. The type of network failures that they study is known as (ε,k)-*failures*, which were introduced by Jon Kleinberg [6]. In this scenario, the adversary is limited to deleting up to k edges or nodes in the graph, which provides an additional condition for the problem.

To monitor such failures, we can assign some nodes as a set D of detectors, which is referred to as an (ε,k)-*detection set*. A detection set D is considered effective if for any (ε,k)-failure of the network, there exists a pair of nodes in D that disconnects, thereby detecting every significant failure. After obtaining a detection set, we can monitor network connectivity by conducting a vertex connectivity test between every pair of nodes in the detection set, which can be done efficiently. In this memo, we explore several practical applications of this approach and discuss their potential in our context.

(does this mean the following methods will be 'detection set approaches' as well? Maybe this needs its own section

2 Data-Driven MILP Solvers

With recent progress in machine learning, data-driven optimization solvers are gaining popularity for their promising performance [7]. The idea behind these solvers is to utilize machine learning algorithms such as reinforcement learning to solve the variable selection problem, which is a key component of modern MILP solvers. This sub-problem can be naturally formulated as a Markov decision process, and solved using reinforcement learning. The availability of state-of-the-art integer programming solvers provides a good dataset for imitation learning [8], as suggested by [7].

Hu [9] applied this technique to solve the traveling salesman problem (TSP), a problem that involves pure integer linear programming instead of mixed-integer linear programming. The study demonstrates a considerable improvement in performance using a data-driven solver compared to existing solvers. Table 1 and Table 2 from the same paper provide additional information on the results.

tables should appear in doc before being referenced in text?

Test Size	Avg. Walltime(s)		Avg. Improvement(s)			Avg. Improvement(%)		
	SCIP	GCNN	All	First 80	Last 20	All	First 80	Last 20
TSP10	0.507s	0.484s	0.022s	0.012s	0.063s	4.40%	3.41%	5.68%
TSP15	2.932s	2.764s	0.168s	0.090s	0.481s	5.73%	5.77%	5.71%
TSP20	50.794s	44.972s	5.822s	0.985s	25.174s	11.46%	7.14%	12.66%
TSP25	238.699s	231.872s	6.827s	3.527s	20.028s	2.86%	6.52%	2.05%

Table 1: Model Trained on TSP10 [9]

Test Size	Avg. Walltime(s)		Avg. Improvement(s)			Avg. Improvement(%)		
	SCIP	GCNN	All	First 80	Last 20	All	First 80	Last 20
TSP10	0.490s	0.461s	0.028s	0.020s	0.063s	5.80%	5.60%	6.07%
TSP15	2.822s	2.661s	0.161s	0.050s	0.605s	5.70%	3.31%	7.48%
TSP20	49.020s	47.181s	1.8389s	0.878s	5.683s	3.75%	6.58%	2.96%
TSP25	256.253s	239.864s	16.389s	3.515s	67.886s	6.40%	6.56%	6.36%

Table 2: Model Trained on TSP15 [9]

Therefore, it is promising to explore this approach further as it is a natural extension of our current implementation and requires only a change in the solver used. provide a little more detail here?

3 Randomized Detection Sets

The field of graph theory studies such a balanced cut problem in a detailed but abstract way for decades. As described in the introduction, a common technique to tackle this problem is by using detection sets ~~introduced by Jon Kleinberg [ ]~~, where we simply sample enough

omit

random nodes in the graph and do connectivity test between pairs of nodes, and it's theoretically guaranteed to be a valid algorithm with high probability. It's known that for any undirected graph $\mathscr{G}$, there exists an (ε, k)-detection set for both edge and vertex failures of size $\widetilde{O}(\frac{k}{\varepsilon})$ independent of n, constructed via random sampling [6, 10].

don't use the same structure, change it up!

One may further parametrize such bound with the edge-connectivity λ, i.e., the minimum number of edges needed to be removed in order to make the graph disconnects is λ. In particular, there is always a (ε, k)-detection set for edge failures with size $\widetilde{O}(\frac{k}{\lambda\varepsilon})$ obtained by random sampling [11].[1] In particular, we may focus on first finding small size network failures, i.e., when k is exactly λ or κ, depending on the type of the attack.

Detecting Small Failures. It has been shown in the literatures that a (ε, λ)-detection set for min-edge or min-vertex failures ($k = \lambda$ or $k = \kappa$) with size $\widetilde{O}(\frac{1}{\varepsilon})$ can be achieved [11]. The min-edge case is clear by considering $\widetilde{O}(\frac{k}{\lambda\varepsilon}) = \widetilde{O}(\frac{\lambda}{\lambda\varepsilon}) = \widetilde{O}(\frac{1}{\varepsilon})$, the min-vertex case is done by a much more complicated and refined arguments by considering shredders [12, 13] and 2-way cuts separately. In both cases, we see that there's a huge potential on this result: we can now detect any minimum attack (minimum in the sense that the adversary need to delete this many edges/nodes in order to make a valid attack) in constant time with respect to both the network size and the connectivity since it only depends on ε. In other words, it'll be much more efficient if the network is densely connected (i.e., λ or κ is large).

rephrase this for clarity: huge advantage?

Finding Small Balanced Separators. In addition to the above result, we can take another viewpoint: instead of passively detecting network failures, we can now find them actively and try to fix it. Such a problem is known as the *finding small balanced separators*,[2] which is an important and challenging open problem in algorithmic graph theory [14, 15, 16, 17, 18, 10, 19, 20].

Finding a minimal size α-separator is NP-hard. On the other hand, if we allow approximation in terms of both *size* and *balancedness* of the vertex separators (which is commonly known as pseudo-approximation), the problem becomes tractable. Specifically, Feige and Mahdian [10] showed that for any $2/3 \leq \alpha < 1$, in $n^{O(1)}2^{O(k)}$ time, we can find an α-separator of size k if such separator exists, except when there is an $(\alpha + \varepsilon)$-separator of smaller size in which case they find the latter. On the other hand, Sebastian Brandt and Roger Wattenhofe [19] proposed an algorithm which finds an $(\alpha + \varepsilon)$-separator of size $O(\varepsilon^{-1}k^{-2}\log^{1+o(1)} n)$ in time $O(\varepsilon k^3 m \log^{2+o(1)} n)$, which relaxes both the size and balancedness. The former is polynomial when $k = O(\log n)$, and the latter is almost linear when $k = O(\operatorname{poly}\log n)$. In the case that $k = \kappa$, ~~i.e.,~~ when the balanced separator is also a minimum vertex cut, in the manuscript by Hu and Saranurak [21], they proposed a randomized algorithm that finds an $(\alpha + \varepsilon)$-separator with size $k = \kappa \leq \varepsilon\sqrt{n}$ in $m^{1+o(1)}2^{O(\varepsilon^{-2}\log 1/\varepsilon)}$, which is almost linear and independent of κ.

By efficiently finding the vulnerable connection in the network, we're able to actively prevent attacks, which provides another incentive to further explore this direction.

[1] It's still an open problem whether we can achieve a similar bound, $\widetilde{O}(\frac{k}{\kappa\varepsilon})$, for vertex failures, where κ is the vertex-connectivity, defined analogously to λ.

[2] In the literature, people usually use α-separator instead of ε-balancedness, but they are essentially the same by interpreting $\alpha := 1 - \varepsilon$. Formally, an α-separator is a cut that separates the graph of size n into connected components of size smaller than αn, for some fixed α.

The Dynamic Viewpoint Using the detection set approach has the advantage of requiring only one sampling of D after specifying k assuming there are no new connections, and it works as long as there are no more than k edges or nodes deleted, making it more "dynamic" than the previous approach. However, this dynamicity is not in the usual sense, and we mention this as a potential future research direction in the conclusion.

4 Conclusion

The memo addresses the issue of decreasing efficiency in the network failure detection algorithm due to the increasing size of clients' network clusters. To enhance the algorithm's performance, two promising directions are suggested: exploring ML-driven solvers and randomized detection algorithms. Additionally, another potential direction is to formulate the problem in a dynamic way, allowing the network to undergo sequential updates and reporting severe network failures when detected. This approach eliminates the need for regular cold-starts, which should significantly speed up the ~~process~~. However, there is currently no existing solution to derive such an algorithm, making it a challenging task. very clear conclusion, nice!

"performance"?