Designing an Algorithm to Allocate Charitable Food Donations to Food Banks and Soup Kitchens

Submitted by

Meredith Benson

Computer Science

To

The Honors College

Oakland University

In partial fulfillment of the

requirement to graduate from

The Honors College

Mentor: Dr. Douglas Zytko, Professor of Human Computer Interaction

Department of Computer Science

Oakland University

April 1, 2022

Table of Contents

Abstract

The coronavirus pandemic has led to a drastic rise in food shortages and subsequent food insecurity, which is burdening food banks and soup kitchens. These charitable organizations now have more people relying on them, and are having a harder time obtaining the food they need. Previous research surrounding the subject has included algorithmic solutions, but each of the resulting algorithms fails to take all of the important considerations into calculation. This paper puts forth an algorithm that centralizes the process of food donation allocation and prioritizes the specific donation requests of food banks and soup kitchens. Interviews were conducted with nonprofit organizations belonging to the charitable food system in order to understand the process of food donation from the perspective of these nonprofits, and the findings were used to inform the algorithm's decision-making process. The algorithm was first designed in pseudocode, and the pseudocode was then translated into JavaScript for use on a website that provides an interface for donors to submit donations and nonprofit organizations to submit requests. The website uses the algorithm to facilitate the matching of donations to nonprofit organizations. This algorithm unifies the food donation allocation process for a community, maximizes the amount of food donations that are utilized, and assists food banks and soup kitchens in acquiring the donations they need.

Introduction

Food insecurity has increased by up to 170% since 2018 [1]. This is due, in large part, to the COVID-19 pandemic. The coronavirus outbreak has affected many aspects of life since its onset, but food insecurity has continued to be a serious concern for millions of Americans.

To understand the rise in food insecurity, it is important to first understand the supply chain issues that have amplified the problem. In the early stages of the pandemic, many factories were forced to close or reduce production in response to outbreaks among their employees. Anticipating lower productivity in the factories, shipping companies scaled back their schedules. However, demand for goods rose globally as more people began to work from home and required products to aid in the transition to virtual life. In response to the elevated demand, factories began to increase production. This boost in manufacturing created an increased need for shipping, both to import raw materials for use in production and to export the finished goods, which began to overwhelm the shipping industry. At the same time, the industry was being disrupted by the need to send supplies such as masks and other personal protective equipment to countries that generally do not have products to ship back to China, which resulted in shipping containers being abandoned in countries that could not use them. This caused a massive shortage of shipping containers, and this, coupled with the significant shipping delays, caused the cost of transportation to increase by upwards of a factor of ten. At ports, the influx of shipments caused further delays as there was not enough personnel or infrastructure to handle the vast amount of shipments being received [2]. Two years since the start of the pandemic, manufacturing and shipping industries are still trying to recover from the supply chain crisis, and the disarray is not forecasted to be resolved anytime soon [3].

Supply chain issues have deeply impacted most large industries around the globe, and the food industry is no exception. The issues have created lasting food shortages for many types of products. Nonprofit aid groups, such as food banks and soup kitchens, are among the most affected by these food shortages. Soup kitchens are organizations that serve hot or pre-prepared meals, often lunch, to people in the community who are in need. Food banks are organizations that collect and store charitable food donations that can then be distributed to local food programs, like food pantries, which distribute bags of food and groceries to be taken home and used by the recipients. Many food banks operate as food pantries and distribute their food to people in the community; for this reason, the term food banks will be used in this paper to encompass both types of organizations. Additionally, most soup kitchens and food banks are registered 501(c)(3) nonprofit organizations, which are charitable organizations that are regulated by the Internal Revenue Service (IRS), and will therefore be referred to interchangeably as nonprofit organizations.

Due to the food shortages, nonprofits are having a more difficult time acquiring the food they need to feed the families who rely on them. Food banks and soup kitchens obtain the vast majority of their food through one of two avenues: charitable donations from for-profit companies, and purchasing in bulk at wholesale suppliers. As for-profit companies are dealing with shortages and supply chain issues, they have fewer excess goods to donate to nonprofit organizations. And because of the shortages, wholesale suppliers have placed purchase limits on many of their in-demand items, greatly reducing an organization's ability to obtain the goods they need. The shortages have also resulted in steep inflation of the cost of many basic goods. The VP of Supply Chain at San Francisco-Marin Food Bank stated that due to the shortages, the cost of chicken drumsticks increased by 32 cents per pound in the last year, which costs them an extra $100,000 a month [4]. Nonprofits receive monetary donations that they use to pay for the items they purchase themselves, but the significant rise in prices has made it difficult for them to afford the vast amount of food needed.

Food waste is another important lens through which this problem should be viewed. Globally, about 30% of all food products go to waste, and it has been estimated that around 60% of that waste is avoidable [5]. Not only is this a problem because the discarded food could be helping to feed families living under the poverty line, but also because it is contributing to the pressing issue of climate change. In 2017, it was found that 14% of all human-related methane emissions originated from food breaking down in landfills [6]. Methane, a greenhouse gas that is 25 times more potent than carbon dioxide, is the second largest source of anthropogenic emissions of greenhouse gasses [7]. A global reduction in the amount of food that is wasted annually would be a significant step towards addressing the climate crisis we are facing. The United States Department of Agriculture (USDA) and the Environmental Protection Agency (EPA) are working together to eliminate 50% of national food waste by 2030, and they have presented a hierarchy that covers the most important routes towards that goal [8]. Second only to source reduction, redistribution of surplus food to feed hungry people is one of the most important areas of recovery that they have highlighted.

Food donation allocation, as is the topic of this paper, will address all of these issues. This topic lends itself to an algorithmic-based solution, as an algorithm has the ability to handle a large volume of information, both in regard to the amount of data it can use in its decision-making process and in regard to the amount of donations it can manage. This is vital to adequately address the issues mentioned above, as any given community contains dozens if not hundreds of local nonprofit organizations. An algorithm is also applicable to this scenario because it will use the same exact rules and factors in its decision-making process every time it allocates a food donation, proving it to be more equitable and less prone to bias than a person attempting to do the same thing. Creating an algorithm to augment the process of food donation allocation will help nonprofit aid groups manage the food shortages they are experiencing by increasing their access to and awareness of charitable food donations. It will align with the government's goal to reduce food waste by redistributing excess food to people in need, which in turn could beneficially impact the problem of global warming. And, ultimately, it will help decrease the food insecurity that so many people are facing today.

Literature Review

There has been a substantial amount of research into this specific issue in the past few years. A group of human-computer interaction researchers [9] assisted the nonprofit organization 412 Food Rescue in building a food donation allocation algorithm to do the work that was previously being done by human volunteers. They focused on participatory design and brought in stakeholders from all levels to take part in the final design of the algorithm in order to maximize the equity of the algorithm's decision-making process. They held individual interviews with the stakeholders in which they gave the stakeholder a theoretical donation and asked them to choose between two nonprofit organizations to which to allocate it based on factors such as proximity, nonprofit size, and regional food insecurity and income levels. The study investigated two different methods of allocation: a machine learning model that was trained using the binary decisions made by the stakeholders, and an explicit rule model that assigned scores to each factor that the algorithm took into consideration. When each type of model was tested against the human participants' decisions, it was found that the machine learning model matched with the stakeholders' decisions at a higher rate than the explicit rule model.

Another study [5] created an application to address food redistribution within the framework of a smart city, focusing on redistributing food directly to individuals in the community without the need for food banks. In this model, the donations were transferred directly between end users, rather than through nonprofit organizations, for individual allocation. The algorithm the app was based on relied on hard-coded processes that ordered a number of factors preferentially. For example, the algorithm considered if a receiver had a preferred donor and if a donor had preferred receivers, and the algorithm attempted to create matches where both donor and receiver included the other in their preferences list, with vicinity as the second most important factor when the preferences could not be met. Alongside donors and receivers, the app also included volunteers as an equally important part of the process. Because the food was being redistributed to individuals rather than organizations, volunteers were needed to transport food from donors to recipients if recipients could not arrange transportation to pick up donations or donors could not deliver them. The app facilitated volunteer matching along with recipient matching to every donation that was received by the system.

Looking at soup kitchens specifically, Akkerman, Bloemhof, Buisman, and Haijema [10] incorporated meal planning into the mathematical optimization model that they built to address donation allocation, in order to minimize food waste. The idea of this model was to help soup kitchens decide whether or not they should accept a donation, thus minimizing the chance of accepting a donation that they could not use. This reduced the risk of food waste at the nonprofit level. Unlike the others, this study took the specific food needs of the nonprofit into consideration. Although it did not address donation management on a large scale, ensuring that nonprofits do not accept donations they cannot use allows that donation to then be accepted by a nonprofit that can utilize it, which benefits the entire local charitable food system.

These three studies contribute important research in this area, however, they are all missing a key portion of the problem. The first study managed donation allocation for a local system of food banks and soup kitchens, but it did not consider a nonprofit's need for a specific type of food, which can lead to nonprofits receiving donations that they do not need or that they cannot use. When this is the case, the nonprofit must either throw the food out, creating avoidable waste, or try to donate the food they have received to another organization, which takes volunteer time and energy to organize. The second study allowed recipients to request both food type and food amount, but entirely removed food banks or soup kitchens from the scenario and instead opted to deliver meals directly to people in need. This is understandably an ideal scenario, but the entire system requires the framework of a smart city. It requires people in need to have smart devices with which they can make requests, and it requires easy access to transportation. This is not currently a realistic implementation. The algorithm developed during the course of this project will be applicable anywhere, but implementation will be focused in southeastern Michigan, which is largely suburban and rural. The app discussed in the second study could not be applied to this region as it currently exists. The third study did account for the specific types of food that a soup kitchen needs, but it failed to examine the larger system as a whole. Food banks are also an important agent in the charitable food system, with many soup kitchens relying on food banks for donations, and many food banks sending goods to one another to optimize usage. A centralized algorithmic system is needed to manage the process of donation allocation for a local charitable food network, and that will be the focus of this paper. This project will propose a unified food donation allocation system that will prioritize the specific needs of the nonprofit organizations that make up a local charitable food network.

Aims and Objectives

The overall goal of this project is to create a website that matches available food donations with the needs of nonprofit food banks and soup kitchens. The website will centralize food donation allocation, which will maximize donation usage while minimizing food waste. Specifically, the focus of this project is to design and implement the matching algorithm that will be used by the site to pair food donations with nonprofits in need.

Aims

1. To determine the factors that the algorithm should base its decision-making process on along with the importance of each of those factors.

2. To design and implement an algorithm that uses the aforementioned factors and their corresponding weights.

3. To create a website[1] that utilizes the algorithm to match donations to charitable organizations.

Objectives

1. Determining the different elements that factor into the algorithm will establish the information the algorithm uses to make decisions. The corresponding importances of each element determines the[1] weight with which each element should be considered in the algorithm.

2. Designing and implementing the algorithm will be the focus of the project. The algorithm will be used to help food banks and soup kitchens receive donations that they need and, by default,

[1] The algorithm discussed in this paper is being used as a part of a larger initiative to build a website that will provide an interface for donors and nonprofits. This is being done in conjunction with four other Oakland University students belonging to the Computer Science and Information Technology undergraduate programs.

will redirect donations that they do not need to an organization that has better use for them. The weighted factors will help it to determine which potential recipient organization is the best match for the donation.

3. Creating the website will provide an interface for donors to input the items that they have available for donation and for nonprofits to indicate what items they need. The algorithm will run on this website to facilitate the matching of donations to charities.

Methodology

The process of this project can be divided into two major subsections: research and implementation. To structure the progression of the project, a series of five sprints was outlined. Sprints are a concept belonging to the family of agile methodologies, a popular approach to project management for software development projects. They break up a development project into manageable two-week blocks, where each week begins with a planning phase and ends with a retrospective to assess the progress made during the sprint. The sprint schedule that was followed is shown below.

Table 1 Sprint schedule used to structure the project timeline

Sprint	Backlog Items
Sprint 1: Research	• Review the current research on the subject • Create a list of local food banks and soup kitchens, along with their location and contact information • Contact local nonprofits to inquire about interviews
Sprint 2: Design	• Hold interviews with representatives from food banks and soup kitchens • Analyze interview findings • Write pseudocode for the algorithm • Outline necessary database queries
Sprint 3: Implementation, part1	• Translate pseudocode into JavaScript code • Sign up for Google's Distance Matrix API and read

	documentation ● Implement Distance Matrix API to calculate driving distance between donor and receiver
Sprint 4: Implementation, part 2	● Connect JavaScript algorithm to back-end database using PHP ● Edge-case testing of algorithm
Sprint 5: Implementation, part 3	● Connect JavaScript algorithm to front-end website code ● Implement email system to message receivers

API, Application Programming Interface; PHP, Hypertext Preprocessor.

Research

A detailed understanding of how both food banks and soup kitchens organize, receive, and utilize donations was crucial to design an algorithm that will be useful to the nonprofit organizations that will ultimately use the website. As there is not currently much information about this process available online, interviews with representatives from both types of nonprofit organizations were necessary to gain this understanding.

First, a list of local nonprofit organizations was compiled, separated by whether the organization was a food bank or a soup kitchen. The two types of organizations operate differently, so it was important to have conversations with both. The organizations included on the list are all local to southeastern Michigan; although this concept could be applied to any community, it was most pertinent to focus on the local region. All of the organizations on the list with public email addresses were then sent an email explaining the basis of the project and asking if someone from the organization familiar with the food donation process would be available for an interview. A total of nine food banks and eleven soup kitchens were messaged; four replies were received. Of the four responses, interviews with three were arranged. The last organization had prohibitive scheduling conflicts and was not available for an interview. Interviews were set with the pantry manager of Rochester Area Neighborhood House (RANH), an intern from Golden Grizzly Pantry (GGP), and the executive director of East Side Soup Kitchen (ESSK).

The interviews were conducted in a semi-structured format. For the comfort of the participants, they were given the choice to conduct the interview over the phone or via virtual meeting using Google Meet. One interviewee chose to have the meeting over the phone, whereas the other two opted for a virtual meeting. Virtual meetings were preferred, as they allow for more human contact, which helps the conversation to feel more relaxed and put the participant at ease. The interviews were structured as a casual conversation, which was guided by a pre-prepared list of questions whenever the conversation came to a natural lull. This allowed for flexibility in the discussion, while also assisted in accurately gathering the information that was needed to understand the food donation process from the perspective of a nonprofit organization. In several cases, the semi-structured format also led to the discovery of new knowledge that was pertinent to the project, but had not been previously considered. The interview questions are listed below.

Fig. 1. Interview questions, ordered

General Interview Questions:

1. Where do you get the majority of your food?
    a. Do the donations come from individual people, from local businesses, or from someplace else? How common is each source?
    b. If you buy food, does the money for that come strictly from charitable donations?
2. How are the donations handled? Are they called in, are they dropped off?
    a. How many people work on that?
3. Do you have specific needs for food, or do you take anything you can get?
    a. When you have specific needs, can you get those needs from donors or do you purchase them yourself?
4. How specific is the need for a certain type of food? Do you need “chicken tenders” or do you just need “protein”?
5. Are you always able to get as much food as you need, or the types of food that you need?
6. What kind of donations can you accept? What rules out a possible donation?
7. What is the approximate turnaround time between accepting a donation and serving it?
8. How often do you have to throw out food donations? For what reasons?

9. How has the pandemic affected this process? Are you being affected by the shortages or the price increases?
10. Are you able to handle dietary preferences or restrictions, such as allergies or veganism? Are there many people who are affected by this?

Questions for Soup Kitchens:

1. Do you have a kitchen in the building? If not, where do you prepare the food?
2. Do you give your patrons a menu or meal schedule ahead of time? Do you ever have to change the menu?

Questions for Food Banks:

1. How do you decide what food and how much is allocated to each person or family that you serve?

With the permission of each of the interviewees, the conversations were recorded using a smart device. The recording was fundamental to the interview process, because it allowed the focus to remain solely on the conversation without worrying about the need for documentation. After the interviews were finished, the recordings were replayed and bulleted notes were taken on everything of interest. Afterwards, the notes were separated into a few different categories: responses to interview questions, knowledge that was unanticipated but useful, and general information that could be useful to some other aspect of the project. This created a structure to the findings of the interviews, to allow for easier access to the information while designing the algorithm.

Implementation

After findings from the interviews were synthesized, the next step was to begin to design the algorithm. This was done using pseudocode, an informal description of the program that is used to outline the flow of logic without the need for underlying details. The pseudocode allowed all of the details of how the algorithm should operate to be worked out, without being weighed down by the technical implementation. After the pseudocode was written, the next step was to translate the pseudocode into JavaScript code that could be run on the front end of the website.

The implementation level included many additional details that needed to be worked out, which are discussed in the following sections.

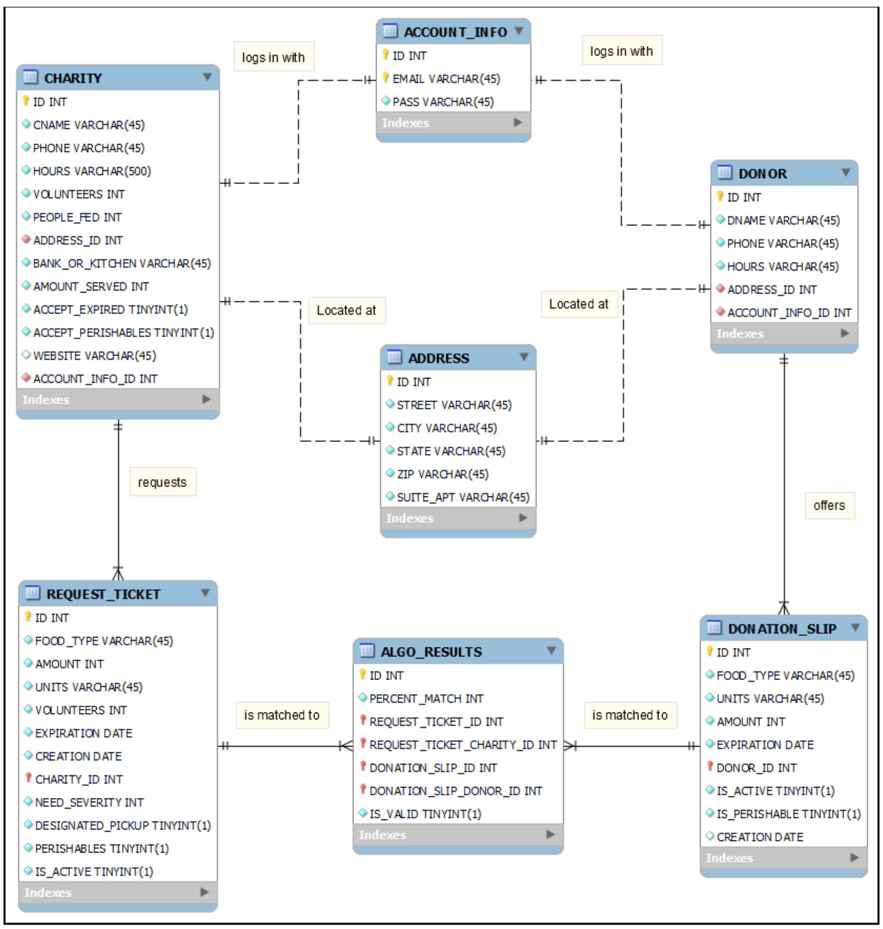
A. Development Environment

For version control and program organization, Github was used. It tracks changes that are made to code, allowing changes to be undone if necessary. The algorithm relies on Hypertext Preprocessor (PHP) code as well as Structured Query Language (SQL) queries, and Github allowed for easy organization of the different code files that the project is composed of. Visual Studio Code was chosen to be used as the code editor for its simple interface and ease of Github integration.

It was important to ensure that the JavaScript code was being tested in a browser environment while it was being written and modified. JavaScript has two main use cases: front end, in a browser environment, and back end, in a Node.js environment, and the way JavaScript operates is different between the two environments. In Node.js, a developer can specify exactly what version of JavaScript should be used, whereas each browser dictates which ECMAScript version of JavaScript will be run, which a web developer has no control over. Also, the browser has the ability to interact with the Document Object Model (DOM) that comprises the structure of a webpage and other Application Programming Interfaces (API) that only affect the browser, while Node.js does not have support for this [11]. In order to run JavaScript in a Chrome browser, a Visual Studio Code extension called Live Server [12] was utilized to start a live development server and open a port of the developer's choice in a Chrome window. Chrome was chosen as it is the most commonly used browser, with recent statistics showing that about 63% of internet users use Chrome [13].

The algorithm relies on a backed MySQL database to store all of the necessary information to facilitate the matching process. It stores information about both the charities and the donors, their request tickets and donation slips, and the results from the matching process, which are later used to send emails to potential recipients to notify them of a pending donation. The database was structured as shown in the entity-relationship diagram (ERD) below.

Fig. 2. Entity relationship diagram of the MySQL database[a]



[a] Both the database design and the diagram in Figure 2 were created by Stevie Warren, a member of the team working on the website implementation.

To communicate between the database and the JavaScript algorithm running on the front end, PHP middleware was used. The JavaScript code initiates contact with the PHP code by sending an HTTP POST request using JavaScript's Fetch API, a popular API for making Asynchronous Javascript and XML (AJAX) requests. AJAX techniques are preferable to synchronous methods, as they allow the request to be made in the background without blocking the rest of the JavaScript code from running. Once the PHP receives the POST request, it then uses the information included in the body of the request to query the database. When the query returns, the PHP will pass that information back to the JavaScript algorithm in the form of a promise. To manage the PHP files required of the project, the open-source software Cross-platform + Apache + MySQL + PHP + Perl (XAMPP) was used [14]. XAMPP allows both an Apache web server and a MySQL server to be started locally on the developer's device, which allows developers to easily run and test PHP files for web development projects.

B. Distance Calculation

During the design phase, the concept of calculating the distances between a donor and a potential recipient organization had been theoretical. Upon further investigation in the implementation phase, the problem was more complicated than it originally appeared. The website would be able to collect the addresses of both the donor and the nonprofit organizations, but a third-party software would be necessary to produce usable information about the addresses and the distance between them, as addresses are only meaningful within the context of a map. For this purpose, it was decided that the Distance Matrix API, an API provided as a part of the larger Google Maps API, would be used [15]. The API takes in two addresses and geocodes them for accurate latitude and longitude coordinates, and then returns the distance between them, amongst several other pieces of information. This API was also chosen for the fact that it is able to return the driving distance between two locations, rather than the direct Euclidean distance that could be calculated from locational coordinates. The API is able to make this calculation using Google's intricate road map system. The driving distance is much more relevant to a volunteer who needs to pick up and deliver a donation, and it was important to factor the most accurate distance measurement into the algorithm's decision making process. For this project, the algorithm sends two complete addresses to the Distance Matrix API, and the API returns the driving distance between them in kilometers, as well as the input addresses to confirm that the API geocoded them correctly. The kilometers are then translated into miles, and the number of miles is run through a helper function to assign the correct proximity scale value.

C. Helper Functions

To partition the logic that controls the algorithm's decision-making process, helper functions were written. There are a total of six helper functions that control various minor aspects of the technical implementation of the algorithm. There is a function that checks if an organization will be able to accept a donation, a function that checks if a donation is currently expired, a function that uses the Distance Matrix API to get the distance between a donor and a potential recipient, and three functions used to get the scale value for the factors that will be weighted by the algorithm. These scales are further discussed in the Results section. A description of each of these helper functions is included in the appendix.

Results

The research conducted through interviews with nonprofit organizations was used to make the design decisions that the algorithm centers around. Through the interviews with the nonprofits, a number of factors were identified as relevant to the considerations that should be made while allocating donations.

Expired Food

Some organizations will not accept expired food, whereas others will. This was determined to be an important factor from the responses received in regard to question eight during the interviews. GGP stated that they will not accept expired donations, and when they do receive donations of expired food, they will take the time to deliver it to a different food bank that will take food past its date. Both RANH and ESSK said that they will accept expired food, within reason. This is an important factor in deciding whether or not a donation should be allocated to an organization. If a donation is expired, it should not be allocated to an organization that does not accept expired food. If it is donated to such an organization, they will either need to use valuable time and resources to redonate it, or they will throw it away. This is a strict binary decision, and therefore is the first item to be checked by the algorithm. If an organization does not accept expired food, the rest of the matching process should be skipped, and the algorithm should move on to the next organization for consideration.

Perishable Food

Similarly, certain organizations will not accept perishable food. This was discovered during the interview with GGP, as they discussed how they do not have a refrigerator or a freezer due to a lack of space and therefore when they receive a donation of food that requires refrigeration, they must redonate it to a different food bank. They also mentioned that they sometimes receive donations of fruit such as bananas or clementines, and when that occurs they attempt to hand it out to anyone who stops by as soon as possible to prevent it from decomposing. An organization might also turn down perishable donations if they are only open a few times a week, as the food could spoil before it has the opportunity to be given out. This is another strict binary decision point, and for the same reasons as with expired food, an organization should not be allocated perishable food if they can not accept it.

Requested Food Type

After the above qualifications are validated, the database is then queried to find request tickets for the type of food that is being donated. This is the most important factor to the algorithm, and it is what sets this project apart from similar algorithms that have been created. During the interviews, it was verified that the organizations often have needs for specific types of foods. ESSK spoke about how they often arrange food drives with local organizations to collect the specific types of food that they need. For example, they recently partnered with a local Boy Scout troop to hold a cake batter drive. GGP stated that they go through bags of chips and water bottles frequently, and are always in need of those items. RANH even has a list of suggested donations for items they need posted on their website. If an organization is in need of a certain type of donation, the donation should be preferentially allocated to them. If there is only one organization requesting that type of food, the donation will be offered to them. If there is more than one organization requesting that type of food, all of the following factors must also be considered.

Weighted Factors

A. Need Severity

When deciding between more than one organization to allocate a charitable donation to, four additional factors will be considered by the algorithm. ESSK indicated that they need juice boxes for after-school meal programs, but there are strict purchase limits on them because of the food shortages; thus, they only have the ability to purchase a few packages at a time. In this situation, they are strongly in need of these donations, whereas other items, such as the cake batter mentioned previously, are more of a luxury and therefore less important. Each request ticket has a need severity assigned by the requester, stating how strongly they are in need of that item. There are three options for this factor: nice to have, general need, and urgent, with corresponding values of 1, 3, and 5. The higher the value, the higher priority the algorithm will place on assigning the donation to the respective nonprofit. This factor carries 40% of the weight, and has been deemed the most important consideration. If a nonprofit is in need of a certain type of food, and that type of food is available to be allocated to them, it should preferentially be given to them over another organization that could use it but does not specifically need it. This should only be considered within reason, however, as an organization that is too far away to receive the donation should not be preferred by the algorithm, which is why proximity, among other factors, will also be considered by the algorithm. It also holds significant weight because it is how this algorithm differs from those that have been created previously.

Table 2 Severity of item need and corresponding algorithmic value

Need Severity	Value
Nice to Have	1
General Need	3
Urgent	5

B. Proximity

The second most important factor to the algorithm is the proximity of the potential recipient to the donor. This is for several reasons. ESSK has only seven employees, as they discussed in their interview, and therefore relies heavily on volunteer assistance to run the kitchen and serve the meals. They have a single staff member who drives their van, meaning that staffer is in charge of all grocery runs, “Feed the Van” can drives that they host, and donation pickups. It was important to minimize the total time that volunteers or employees spend picking up the donations because their time is a valuable resource necessary for other operations of the nonprofit. Also, the longer distance food must travel, especially perishable food, the more risk it is at for damage while in transit. ESSK stated that they will throw out bruised fruit, citing that they do not wish to serve food to their patrons that they would not serve to their own family. It is also best for foods that must remain refrigerated or frozen to have short travel distances, because the nonprofit organizations may not be able to keep the items cold while in transit. For these reasons, the proximity factor is weighted with 30% and follows a scale that ranges from one to five, with five representing that there is less than five miles between the agents, one representing that there is over twenty miles of distance between the agents, and each value in between increasing by a value of five miles.

Table 3 Proximity between donor and potential recipient and corresponding algorithmic value

Miles	Value
≥ 20	1
15-20	2
10-15	3
5-10	4
< 5	5

C. Fulfillment

Another factor was discovered to be necessary during the implementation process. The question of how to handle a situation in which a request ticket does not need as much food as the donation has to offer had arisen, and it was decided to create another factor to be included in the weighting process. This factor, fulfillment, indicates if a request can be completely satisfied by the donation amount, over 50% satisfied by the donation amount, or under 50% satisfied by the donation amount. This is an important factor, as it is best to preferentially send donations to organizations that will be most assisted by them. It was also considered to keep fulfillment separate from the weighting process, and instead choose to send donations to the first ranked organization that can be completely fulfilled by the donation. This, however, works against the purpose of weighting several factors for consideration to determine the best nonprofit to which to allocate the donation, and it was decided that including the factor in the weighting scheme would more accurately target the issue. Most organizations will take any donations they can get, independent of whether it completely satisfies their need, and it is common for organizations to collect several donations to satisfy the need they have. Although it is better to avoid this situation, it should not be so important as to prevent an organization with a higher need from receiving a donation simply because there is another requester that could be completely satisfied.This factor received 20% of the weight within the algorithm, and the scale is shown below.

Table 4 Percentage amount a donation will be be able to fulfill a request and corresponding algorithmic value

Fulfillment	Value
< 50%	1
50%-99%	3
≥ 100%	5

D. Organization Size

The last factor that is considered during the weighting process is the relative size of the organization. The bigger an organization is, the more people they feed and therefore the more food they will require. This was also observed indirectly in the interviews. GGP, the smallest of the three organizations, mostly supplies shelf-stable food such as chips, soup packets, and cookies, and said that most of their shoppers are there for a quick snack rather than an essential meal. RANH, in contrast, is a much bigger organization and focuses on distributing well-rounded groceries following the MyPlate nutritional guidelines provided by the USDA [16], and requires much more food per month to feed everyone relying on them. It could be speculated that a large organization has more methods for obtaining food in place or more resources, but it is difficult to qualify that argument. However, because of the more uncertain nature of this factor, it only receives a weight of 10% in the matching algorithm’s calculations. It should be noted that since the two types of organizations operate differently, they follow different scales. Food banks are able to give out more food overall than soup kitchens, and food banks measure their productivity in pounds of food while soup kitchens measure productivity in number of meals served.

Table 5.1 Number of meals served per month in a soup kitchen and corresponding algorithmic value

Meals Served Per Month	**Value**
< 2,000	1
2,000 - 5,000	2
5,000 - 10,000	3
10,000 - 20,000	4
>= 20,000	5

Table 5.2 Pounds of food served per month from a food bank and corresponding algorithmic value

Pounds of Food Served Per Month	**Value**
< 1,000	1
1,000 - 10,000	2
10,000 - 100,000	3
100,000 - 1,000,000	4
>= 1,000,000	5

The scales for this factor were created based on research done on the sizes of local food banks and soup kitchens. Each local nonprofit organization was investigated, and the most recent numbers that they listed were recorded in the following table. This was done using self-reported data that was found either on their website or in self-published newsletters, which is used only as an estimate to allow us to understand the general scope of the amount of people that these organizations are able to reach.

Table 6.1 Number of meals served per month from individual soup kitchens

Organization Name	Meals Per Month	Reference
Baldwin Center	2,500	[17]
Capuchin (Conner)	27,345	[18]
Capuchin (Meldrum)	14,284	[18]
Crossroads of Michigan	1,200	[19]
East Side Soup Kitchen	5,000	[20]
God's Kitchen	8,333	[21]
Isabella Community Soup Kitchen	2,400	[22]
Matchan Soup Kitchen	1,254	[23]
South Flint Soup Kitchen	2,583	[24]
St. Leo's Soup Kitchen	3,333	[25]

Table 6.2 Pounds of food served per month from individual food bank[b]

Organization Name	Pounds Per Month	Reference
Capuchin Service Center	215,812	[18]
Food Bank of Eastern Michigan	4,250,000	[26]
Forgotten Harvest	4,320,000	[27]
Gleaners Community Food Bank	3,816,666	[28]
God's Helping Hands	32,047	[29]
Golden Grizzlies Pantry	1,600	
Matchan Pantry	21,220	[23]
Rochester Area Neighborhood House	22,000	
Safe Harbor Food Pantry	25,000	[30]
South Michigan Food Bank	1,166,666	[31]

[b] The values for Golden Grizzly Pantry and Rochester Area Neighborhood House come from their respective interviews

Lack of Request Ticket

In the case that there are no active request tickets for the type of food being submitted for donation, then a modified version of the weighting process will occur. First, the database will be queried to return all local organizations. For each organization, it will be checked whether they will accept expired or perishable food, in the case that the donation is either expired or perishable. The organizations that can accept the type of food being donated will move on to an adapted version of the weighting scheme. The adapted weighting scheme ranks organizations using proximity and organization size with weights 70% and 30%, respectively. Need severity is only included on a request ticket, and without an active request ticket, this factor cannot be used. The fulfillment factor relies on the amount requested on a request ticket, which means this factor also cannot be used when there are no active request tickets.

Algorithm

Included below is the pseudocode for the final version of the algorithm.

Fig. 3. Pseudocode for the matching algorithm

```
function canBeAccepted(donationSlip, organizationInfo):
        if(donationSlip.isExpired): //if donation is for expired food
                if(not requestTicket[i].acceptsExpired): //if organization does not accept
                        expired food
                        return False
        if(donationSlip.isPerishable): //if donation is for perishable food
                if(not requestTicket[i].acceptsPerishable)://if organization does not accept
                        expired food
                        return False
        return True //if neither of above conditions return False, then donation can be
                accepted
```

```
function matchingAlgorithm(donationSlip):
        variable requestTickets = db.query(request tickets for type of food included
                in donationSlip)
        if(requestTickets): //if there are active request tickets
        for(i in requestTickets): //for each request ticket do the following
                if(not canBeAccepted(donationSlip, requestTicket.orgInfo)): //if donation
                        cannot be accepted
```

```
                continue //move on to next request ticket
            else: //if donation can be accepted, move on to weighting process
                variable distance = getDistance(donationSlip,
                requestTickets.orgInfo[i])
                variable size = getSize(requestTickets[i].orgInfo)
                variable need = getNeed(requesTickets[i])
                variable fulfillment = getFulfillment(donationSlip,
                requestTickets[i])
                variable weight = 0.4*need + 0.3*distance + 0.2*fulfillment +
                0.1*size
                db.query(store organization and corresponding weight)

        else: //if there are no active request tickets
            variable organizations = db.query(all organizations in the area)
            for(i in organizations): //for all organizations do the following
                if(not canBeAccepted(donationSlip, organization)): //If donation
                cannot be accepted
                    continue //move on to next request ticket
                else: //if donation can be accepted, move on to weighting process
                    variable distance = getDistance(donationSlip,
                    organizations[i])
                    variable size = getSize(organizations[i])
                    variable weight = 0.7*distance + 0.3*size
                    db.query(store organization and corresponding weight)
end
```

Testing

In order to test the functionality of the algorithm, two classes were created. The first class, RequestTicket, represented the information submitted by a nonprofit when requesting a certain type of food. The second, DonationSlip, represented the information included by a donor when filling out the donation form on the website. Using these classes, imitation request tickets and donation slips were created in order to test the matching algorithm and all of its helper functions, to ensure that they operate accurately and as expected when integrated with the website. The helper functions and their purposes are described further in the appendix. Each helper function was first tested independently to ensure accuracy, and then the algorithm was tested as a whole. It was found that the algorithm operates correctly, using the information from a single request ticket and a single donation slip to assign a weighted value to the pairing. In the case that there are no active request tickets for the type of food in a donation, the donation slip is weighted against nearby organizations using proximity and organization size, and this process was also found to function correctly.

Discussion

The current charitable food system is driven by supply rather than demand. Individuals will decide that they have too many cans that they cannot use, and will drop them off at a local food bank. Farmers may decide they have too much of a certain crop, and will donate the excess to a nearby soup kitchen to prevent the produce from going to waste. People who hunt for sport can take their catches to a legal packing firm to be packaged for donation. For-profit companies will pull B-grade items that they believe their customers will not purchase off the shelf to deliver to a partnering nonprofit organization. Even when a nonprofit obtains food from another food bank, they often must still pay for it, and there is little guarantee that the food bank will have what they are in need of. Nonprofit organizations can purchase food from wholesale suppliers, but because of the food shortages they may not be able to purchase enough to satisfy their need.

Furthermore, the current process has no overarching management system. This means that each food bank and soup kitchen must find its own partnerships and make arrangements to receive the donations that they need. Because of this, charitable donations are often not used with maximum efficiency. Donations can be sent to organizations that cannot accept them. As explained by GGP during their interview, they receive regular donations for bricks of cheese, which they cannot accept because they do not have a refrigerator. They must then redonate the blocks of cheese to an organization that can use them. This causes further problems in that the more time certain items such as produce spend in transportation, the more likely they are to be bruised or damaged in the process, causing unnecessary waste. GGP makes a conscious effort to ensure that the food does not go to waste, but this is a luxury that not all nonprofits can afford. If they do not have the resources or the time to take the donation to an organization that can accept the food, a nonprofit will have to throw the items out. It is also possible for a donation to be sent to an organization that does not have a direct need for it but will be able to use it, while elsewhere in the community there is a nonprofit that has a strong need for that item. This is simply an issue of awareness, but because there is no central governing body, there is no straightforward way for donors to find the organization that is most in need of their donation. It is clear to see, however, that this is an issue. If we want to truly improve access to food and reduce food insecurity, we need to ensure that excess food is used as efficiently as possible. This will also mean a reduction in the amount of food that gets wasted, which is beneficial both to the members of the community and to the climate.

The algorithm proposed in this paper addresses both of these concerns. Food banks and soup kitchens are able to create request tickets to indicate the food they are in need of, shifting towards a more demand-centered charitable food system. Donors can fill out a donation form to indicate the food that they have available for donation, and the algorithm runs immediately upon submission to decide the best organization to which to allocate the food. This creates a centralized system that has a top-down view of the network of food banks and soup kitchens within a community, allowing donations to be allocated more efficiently to the recipients who will be able to use them best. This will also prevent donations from needing to be allocated and transported more than once. A donation must be approved and accepted by the food bank or soup kitchen before it is allocated to them, giving them control over which donations they receive.

And, expired or perishable donations will not even be offered to organizations that will not accept expired or perishable food, meaning that every donation offer they receive is for a type of food that they can accept.

In certain areas, centralized platforms such as this already exist, with the key exception that they have human volunteers running the back end and making all of the allocation decisions. For example, Food Rescue US states on their website, “The local Site Director will match donated food to a local nonprofit agency in the community that can best utilize it” [32]. This takes a lot of time and effort on the part of the site director, and that effort could be obviated through the implementation of this algorithm. Additionally, people are subject to bias and human error, and they may not allocate a food donation as efficiently as possible for these reasons. This algorithm will use the same considerations for every allocation it makes, ensuring that all decisions are equitable and that the donation will be used with maximum efficiency.

Further Research

This algorithm, however, is not without its limitations. One potential factor that was considered for inclusion in the weighting scheme was the food insecurity or poverty level of the neighborhood assisted by a given nonprofit. There is an atlas provided by the Economic Research Service (ERS) branch of the USDA that indicates both food access and income level of neighborhoods in the United States [33]. This would be a valuable factor to include, as it would be able to indicate more specifically which areas of a community have more inherent need, and areas with higher need could be weighted accordingly. A satisfactory incorporation of this factor was outside of the scope of this project, but could be a valuable area for future study.

Another decision point that was considered was how to handle large donations. For the purposes of this project, the fulfillment factor in the weighting scheme was created to represent this in the algorithm. Requesters whose request amount can be totally satisfied by the donation amount are preferred to requests that are unable to be satisfied by the amount of food in a donation, because it will be more meaningful to them. Ideally, though, the algorithm would be able to split up large donations to be given to two or more requesting charities. However, the amount of research conducted during the interviews was not sufficient to accurately inform the potential decision-making process behind splitting up the donations. For example, should a donation be split up if it means neither nonprofit will be totally satisfied? Should a donation still be allocated to the nonprofit ranked the highest if there is a lower-ranked donation that can be totally satisfied? If a donation can completely satisfy one nonprofit but only has a marginal amount of excess, should it be split up or should the excess be included in the donation? These questions and others will require more in-depth research to develop a refined solution.

Lastly, it would be pertinent to conduct further research into the potential of allowing donors to select to whom they would like to allocate the donation. Other similar systems, such as the one created by the group of ICT researchers [5] allow donors to specify the recipient they would like to see receive the donation, in the form of a ranked list. Allowing this, though, would directly impede the effectiveness of the algorithm, as it would give weight to a factor that does not help to maximize the efficiency of the allocation while subsequently taking weight away from the factors that do maximize efficiency. However, most food banks and soup kitchens have regular donors who are adamant about supporting them. As RANH mentioned in their interview, they have a donor who makes regular trips to both Costco and Panera to persuade them to give her excess food for the food bank. GGP talked about how there is a retired Oakland University professor who makes a regular donation of approximately $500 worth of food every other week to the pantry, because he understands the value of having a food bank on campus and wishes to support it. The donors who prefer to donate to specific organizations may choose not to use the algorithm for this reason, which could also negatively impact the overall efficiency of the system. In either regard, more research is required to determine the best course of action.

Conclusion

This project will culminate in a website available for use by food banks and soup kitchens to request and receive the donations that they are in need of, and by donors to register the food they have available for donation. An algorithm has been produced for use by the website to facilitate the matching of donations to recipient organizations based on a number of relevant factors. This benefits the local charitable food network, as it centralizes the process of food donation allocation and therefore reduces food wastage and minimizes the time and energy spent by nonprofits to redonate unusable items. This could benefit the patrons of the food banks and soup kitchens, as there would be more food available to be given out and the volunteers at these organizations would be able to spend more time focusing on feeding and serving their patrons. The research accomplished in this project may also draw attention to how the pandemic-related supply chain issues are affecting food banks and soup kitchens, a problem that is forecasted to last for years to come. This project could even inspire an organization with more resources to implement a similar system on a large scale, which would help to reduce food insecurity within the communities involved. This algorithm, and the website developed to support and promote its use, builds on previous technologic solutions designed to alleviate the problem of food insecurity and its widespread adoption could represent a needed advance in the field.

Acknowledgements

I have received much support and assistance throughout the writing of my thesis, and would like to acknowledge the many people who have helped me along the way.

First, I would like to thank my mentor Dr. Doug Zytko, who worked with me through several drafts of this paper to make it what it is today. The idea for this entire project sparked from some initial feedback you gave us at the beginning of this project, and your feedback and advice have continued to improve the project and this paper along the way. I would also like to acknowledge my proposal advisor, Dr. Susan Beckwith, for answering all of my questions and providing valuable insight into the process of writing both the proposal and the thesis paper.

I would like to thank the members of the nonprofit organizations that I interviewed for their invaluable insight into the food donation process. Johana Campos Sanchez from Golden Grizzly Pantry, Debra Cole from Rochester Area Neighborhood House, and Diane Keenan from East Side Soup Kitchen: thank you all so much for taking the time to sit down and answer my questions, and thank you for your expertise. Without it, this project would not have happened.

Additionally, I would like to acknowledge my fellow classmates who have been working on the website implementation. To Marcus Dyson, Talha Memon, Stevie Warren, and Lisa Zhang: thank you. Though the website was not the focus of this paper, your individual contributions to the website and to help me with the algorithm design and implementation were incredibly beneficial, and your hard work has not gone unnoticed.

Finally, I would like to thank my mother, Darlene Benson, for her unmatched knowledge of English grammar and for her guidance throughout the process. You were always there to listen patiently to my problems and to help me brainstorm solutions, and your time, effort, and encouragement helped to make this paper the best it could be.

References

1. Friedman E, Hudak K, Johnson J, Benjamin-Neelin S. Food bank donations in the United States: a landscape review of federal policies. Nutrients. 2020;12(12): 3764. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7762564/.
2. Goodman P. How the supply chain broke, and why it won't be fixed anytime soon. New York (NY): New York Times. 2021 Oct 22 [updated 2021 Oct 31; accessed 2022 Jan 27]. https://www.nytimes.com/2021/10/22/business/shortages-supply-chain.html.
3. Goodman P. A normal supply chain? It's 'unlikely' in 2022. New York (NY): New York Times. 2022 Feb 1 [updated 2022 Feb 6; accessed 2022 Feb 20]. https://www.nytimes.com/2022/02/01/business/supply-chain-disruption.html.
4. Sharma J. Higher prices, fewer drivers: Food banks exacerbated by the supply-chain crisis. New York (NY): Morning Brew. 8 Oct 2021 [accessed 2022 Jan 18]. https://www.morningbrew.com/retail/stories/2021/10/08/higher-prices-fewer-drivers-food-banks-exacerbated-by-the-supply-chain-crisis.
5. Mukhopadhyay S, Sanyal B, Sanyal S, Singh V, Xhafa F. A game theoretic framework for surplus food distribution in smart cities and beyond. Appl. Sci. 2021;11(11:5058)1-19. http://dx.doi.org/10.3390/app11115058.
6. Why should we care about food waste? Washington (DC): Department of Agriculture (US). [date unknown; accessed 2022 Feb 20]. https://www.usda.gov/foodlossandwaste/why#:~:text=Greenhouse%20gases%20generated%20from%20food,methane%2C%20a%20powerful%20greenhouse%20gas.
7. Importance of methane. Washington (DC): Environmental Protection Agency (US). [updated

2021 June 30; accessed 2022 Feb 25]. https://www.epa.gov/gmi/importance-methane#:~:text=Methane%20is%20more%20than%2025,due%20to%20human%2Drelated%20activities.

8. Food waste FAQs. Washington (DC): Department of Agriculture (US). [date unknown; accessed 2022 Feb 25]. https://www.usda.gov/foodwaste/faqs.

9. Chan A, Kahng A, Kim J, Kusbit D, Lee M, Lee S, Noothigattu R, Procaccia A, Psomas A, See D, et al. WeBuildAI: Participatory Framework for Algorithmic Governance. *Proc. ACM* Hum.-Comput. Interact. 2019;3(CSCW):1-16. https://dl.acm.org/doi/10.1145/3359283.

10. Akkerman R, Bloemhof J, Buisman M, Haijema R. Donation management for menu planning at soup kitchens. Eur. J. Oper. Res. 2019;272(1):1-3. https://www.sciencedirect.com/science/article/pii/S0377221718305289.

11. Differences between Node.js and the browser. San Francisco (CA): Node.js. [date unknown; accessed 2022 Mar 2]. https://nodejs.dev/learn/differences-between-nodejs-and-the-browser.

12. Dey R. Live Server [software]. Version 5.6.1. Ritwick Dey. 2017 Jun 27 [updated 2022 Mar 8].https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer&ssr=false#overview.

13. Browser market share worldwide. Dublin (IRL): Statcounter. 2022 Feb [accessed 2022 Mar 2]. https://gs.statcounter.com/browser-market-share.

14. XAMPP [software]. Version 8.1.2-0. Apache Friends. 2022. https://www.apachefriends.org/index.html.

15. Google Maps. Distance Matrix API [application programming interface]. Mountain View (CA): Google. 2022 Feb. https://developers.google.com/maps/documentation/distance-matrix/?hl=en_US/.

16. What is MyPlate. Washington (DC): Department of Agriculture (US). [date unknown; accessed 2022 Mar 19].

17. About us. Pontiac (MI): Baldwin Center; [date unknown; accessed 16 Feb 2022]. https://baldwincenter.org/about/.

18. Ministry statistics. Detroit (MI): Capuchin Soup Kitchen; 2017 [accessed 16 Feb 2022]. https://www.cskdetroit.org/about_us/ministry_statistics/.

19. Happy New Year. Detroit (MI): Crossroads of Michigan; [date unknown; accessed 12 Mar 2022]. https://crossroadsofmichigan.org/happynewyear/.

20. East Side Soup Kitchen History. Saginaw (MI): East Side Soup Kitchen; [date unknown; accessed 12 Mar 2022]. https://www.eastsidesoupkitchen.org/history/.

21. God's Kitchen. Grand Rapids (MI): Catholic Charities West Michigan; [date unknown; accessed 27 Mar 2022]. https://ccwestmi.org/what-we-do/food-pantry-programs/gods-kitchen/.

22. About ICSK. Mt Pleasant (MI): Isabella Community Soup Kitchen; [date unknown; accessed 12 Mar 2022]. https://isabellacommunitysoupkitchen.org/aboutUs.html.

23. December newsletter. Pontiac (MI): Matchan Nutrition Center; Dec 2021 [accessed 16 Feb 2022]. https://www.matchannutritioncenter.org/wp-content/uploads/2021/12/2021-December-Newsletter.pdf.

24. About us. Flint (MI): South Flint Soup Kitchen; [date unknown; accessed 12 Mar 2022]. http://southflintsoupkitchen.org/about-us.html.

25. About us. Detroit (MI): St Leo's Soup Kitchen; [date unknown; accessed 12 Mar 2022]. https://www.stleosoupkitchendetroit.org/about.

26. Food Bank of Eastern Michigan. Flint (MI): Food Bank of Eastern Michigan; 2021 [accessed 16 Feb 2022]. https://www.fbem.org/.

27. Forgotten harvest. Oak Park (MI): Forgotten Harvest; [date unknown; accessed 16 Feb 2022]. https://www.forgottenharvest.org/.

28. Give money. Detroit (MI): Gleaners Community Food Bank; 2019 [accessed 16 Feb 2022]. https://www.gcfb.org/give-money/#:~:text=Gleaners%20is%20the%20largest%2C%20most,45.8%20million%20pounds%20in%202019.

29. Statistics. Rochester Hills (MI): God's Helping Hands of Michigan; 2011 [accessed 16 Feb 2022]. https://www.ghhmichigan.org/statistics.

30. What we do. Bay City (MI): Safe Harbor; [date unknown; accessed 16 Feb 2022]. https://safeharborkitchen.com/.

31. South Michigan Food Bank. Battle Creek (MI): South Michigan Food Bank; [date unknown; accessed 16 Feb 2022]. https://smfoodbank.org/.

32. Rescue food. Stamford (CT): Food Rescue US. [date unknown; accessed 2022 Mar 19]. https://foodrescue.us/rescue-food/.

33. Economic Research Service. Washington (DC): Department of Agriculture (US). [date unknown; accessed 2022 Mar 19]. https://www.ers.usda.gov/data-products/food-access-research-atlas/documentation/.

Appendix

Table 7 Descriptions of each helper function used by the algorithm

Helper Function	Description
canBeAccepted Arguments: donation slip, request ticket Return type: boolean	This function takes a single donation slip and a single request ticket as input, and checks if the food is expired or perishable. If either of those are true, it will check if the requesting organization accepts expired or perishable food. It will return a value of true or false, indicating whether the requesting organization will be able to accept the donation.
isExpired Arguments: donation slip Return type: boolean	This function takes a donation slip as input, and checks whether the expiration date indicated for the donation has passed. This function is necessary because it is possible that the donation slips may not be allocated immediately upon entry into the system, and it is important to have updated information about the expiration status. If a donation is submitted before it expires, but is not allocated to an organization until after it expires, it is vital to ensure that the algorithm is aware that the donation is now expired, and should only be offered to an organization that will accept expired food. It uses the exact date and time that the algorithm is run for comparison against the given expiration date, and returns true or false depending on whether it is found to be expired.
getDistance Arguments: donation slip, organization profile Return type: integer	This function accepts a single donation slip and the profile information for the nonprofit, and makes the call to the Distance Matrix API to determine the driving distance between the potential recipient organization and the donor. The API returns the driving distance measured in kilometers, which the function will translate into miles. This function is asynchronous, meaning when called, it returns a promise that resolves to the driving distance.
getProximity Arguments: distance Return type: integer	This function is called after getDistance, and accepts the distance returned as input. It is used to relate the number of miles between the donor and potential recipient to the corresponding proximity scale value that is used by the matching algorithm. It returns the scale value as an integer.
getSize Arguments: organization profile Return type: integer	This function takes in an organization's profile information and uses it to determine the corresponding organization size scale value used by the matching algorithm. It checks whether the organization is a food bank or a soup kitchen, and uses that information to map it to the correct scale. It returns the scale value as an integer.
getFulfillment Arguments: request amount, donation amount Return type: integer	This function takes in the amount of food from the request ticket and the amount of food included in the donation, both measured in pounds. It divides the donation amount by the request amount to determine the exact percentage of the request that the donation is able to fulfill. This percentage is mapped against the fulfillment scale to determine the corresponding scale value, and then the value is returned as an integer.