

Towards a Socially Responsible Smart City: Dynamic Resource Allocation for Smarter Community Service

Huey-Ru (Debbie) Tsai

University of California, Los Angeles
tsai.debra@cs.ucla.edu

Min Kyung Lee

Carnegie Mellon University
mklee@cs.cmu.edu

Yasser Shoukry

University of Maryland
yshoukry@ece.umd.edu

Vasumathi Raman

vasumathi.raman@gmail.com

ABSTRACT

As a city becomes smarter, the integrated networks of engineered cyber and physical elements provide the capability to greatly improve the quality of life of its citizens. In order to leverage these capabilities to benefit all classes of society, we propose a framework that balances the supply and demand of available resources while maximizing the social welfare of people-in-need by utilizing cyber-physical infrastructure in smart cities. We show through numerical simulations that our proposed framework can reduce the amount of resources wasted by 25% through intelligently assigning the location of services and dynamically pairing resources to different homeless populations.

CCS CONCEPTS

- **Applied computing** → **Law, social and behavioral sciences**;
- **Computer systems organization** → **Embedded and cyber-physical systems**;

KEYWORDS

Algorithmic services, fairness, allocation, community service, smart city, service design, automation

1 INTRODUCTION

Every year, 3.5 million people in the US experience homelessness, with 1 in 30 children becoming homeless [1]. Despite numerous government-sponsored programs and efforts by nonprofit organizations, many homeless people live in abject conditions. According to the the U.S. Conference of Mayors 2013 Status Report on Hunger & Homelessness in 2013, 21% of people across the surveyed cities who need emergency food assistance received none [2, 3]. Moreover, in

This material is based upon work supported by the National Science Foundation under Grants Number #1651858 and #1651566. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or the U.S. Government. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

BuildSys'17, November 8–9, 2017, Delft, The Netherlands

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

ACM ISBN 978-1-4503-5544-5/17/11...\$15.00
<https://doi.org/10.1145/3137133.3137163>

all responding cities, emergency kitchens and food pantries had to reduce the quantity of food each person could receive per visit. In 78% of those cities, they had to reduce the number of times a person or family could visit a food pantry each month. In two-thirds of the cities, facilities had to turn away people due to a lack of resources.

Another major problem is the lack of real-time coordination among different community service efforts, which results in an inefficient system where available supply is not matched with demand. Despite recent successes in smart-city technologies and community-driven capabilities, the act of managing and coordinating services for communities of people in need is still a local, ad hoc effort. In particular, some food banks constantly have excess supply, whereas others do not have enough [4]. On the other hand, smaller food service establishments such as restaurants and private citizens often have perishable food to donate on a daily basis. These resources often go to waste because the supply is usually in small quantities, thus uneconomical for donors to transport it to distribution centers.

Rethinking smart city technologies to best serve those in need is essential for improving their access to resources including food, shelter and medical services. As a first step, we identify the problem of balancing supply and demand while maximizing the social welfare of both people-in-need and other citizens by utilizing the cyber-physical infrastructure in smart cities. We argue that by intelligently managing the efforts of the city, NGOs and private citizens, a smart city can optimally distribute the available supply of food, temporary shelter, health care and other services. The main contribution of this paper is an architecture for optimal resource allocation and assignment that achieves this goal along with preliminary numerical results that supports the proposed architecture.

2 SOCIALLY RESPONSIBLE SMART CITY

Even as technology permeates every corner of our lives today, the homeless population remains a largely underserved population. A major area where technology could have a large impact on the quality of life of the homeless population is in information dissemination about donated food and services, and an efficient, real-time management and distribution of donations. In this section, we start by reviewing the existing technologies that currently serve different people-in-need communities followed by the architecture of the proposed system that enhances existing technologies to facilitate the access of people-in-need to the available resources.

2.1 Existing Technologies

1. **Homeless initiated (“Pull” model):** In this model of information dissemination, homeless people are held responsible

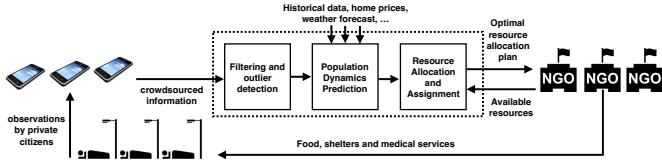


Figure 1: A cartoon of the architecture of the proposed solution.

for searching for the information they need. A California survey showed that 62% of homeless youth have access to feature phones [5]. Cities like San Francisco, CA have launched websites for the homeless that are accessible by phone and used geolocation APIs in order to retrieve their location and to provide information on food and shelters nearby [6]. Another piece of technology that supports a pull model is the federally-provided voicemail accounts that the homeless can access from free phone cabins. There are also many free computer labs, such as in public libraries, which allow the homeless to access the internet and search for data.

2. Community initiated (“Push” model): In this model, community coalitions and NGOs are responsible for delivering information to the homeless. For example, the New York City Department of Homeless Services deploys teams citywide to engage and encourage homeless individuals to move from the streets into existing shelters and to utilize drop-in center services [7].

We refer to the smart cities that employ the later model as socially responsible smart cities. To achieve this goal, we start by classifying the available resource (food, shelter, medical, etc.) providers into two broad categories based on location availability: (i) permanent and (ii) mobile. Examples of the first resource category are permanent shelters and food pantries. Examples of the second category include mobile food trucks and clinics.

2.2 System Architecture

We propose a subscription-based service model in which (permanent or mobile) food, medicine, shelter and other service providers use mobile technology to declare the available supplies and their location. This data is combined with crowdsourced real-time information of people-in-need population distribution reported voluntarily by private citizens (e.g. using mobile phone apps). The crowdsourced information is filtered by a web server in order to detect redundancies (e.g. same people-in-need reported by different private citizens), mismatches, and/or outliers.

We note that the information collected does not provide a continuous stream for the location of people-in-need; it provides the locations only as they are encountered by private citizens. Hence, the next step is to use machine learning algorithms along with mathematical models for population dynamics, historical data, home prices, weather forecast and other features in order to augment this sporadic stream of information to build a continuous estimate of the density and location of needy communities.

The final step is to fuse the information provided by the resource providers along with the estimate of the density and location of needy communities to calculate an optimal strategy for dynamically allocating resources to service locations (including routing information for transporting food between locations where appropriate) that minimizes the wastage of available resources while maximizing the social welfare and satisfying a set of spatio-temporal specifications to be fulfilled (e.g. maximum homeless population density in

specific district at particular time, minimum number of meals an individual needs per week, maximum distance traveled per day). The overall architecture of the proposed system is shown in Figure 1. Concurrent efforts are being made to implement the proposed system in collaboration with several NGOs and homeless service organizations. While mobile technologies are being developed to report the location of available resources and of people-in-need (PiN), along with physically distinguishing characteristics (e.g., hair color), mobile apps will be connected with community partners who can address the immediate needs of PiNs. [9] However, in the remainder of this paper, we focus only on the final step of the proposed system, namely, resource allocation and assignment.

3 RESOURCE ALLOCATION AND ASSIGNMENT

We note that our proposed methodology generalizes to services including temporary shelter and free medical treatment, but we will focus in this section on the particular problem of food distribution allocation and assignment. We denote by N_{NGO} the number of non-governmental organizations (NGOs) that are participating in food resource supply. Similarly, we denote by N_{PiN} the number of people-in-need. Each NGO is a tuple: $NGO_i = (s_i, c_i, r_i)$, where $i \in \{1, 2, \dots, N_{NGO}\}$ and s_i is the amount of food (services) available at the i th NGO, c_i is the x - y position of the NGO, and r_i is the radius of coverage or neighborhood around the NGO position for which this NGO can serve. As discussed before, we consider both permanent and mobile food suppliers where the location c_i of the former is fixed while the location c_i of the later is free to be assigned by our system. To differentiate between the two cases, we will use the notation NGO_i^f and NGO_i^m where the superscripts f and m stands for fixed and mobile, respectively. Similarly, each PiN is a tuple: $PiN_j = (h_j, l_j, id_j)$, where $j \in \{1, 2, \dots, N_{PiN}\}$ and h_j is the hunger level of the j th person-in-need person¹ and l_j is his location. Lastly, id_j represents the physical traits reported on the crowdsource mobile app. We assume that both hunger level h_j and food s_i have the same units, i.e., one unit of food is required to reduce the hunger level by one unit.

The objective of our distribution algorithms is to (i) assign the location of the mobile NGOs (resource allocation) and (ii) assign PiNs to NGOs (resource assignment or pairing) in a manner that maximizes the social welfare. In our framework we define social welfare by three criteria: (1) percentage of individuals serviced (or paired with NGO), (2) average hunger, and (3) percentage of food waste. While one can argue that the first and second criterion are redundant, it is important to note that the first criterion is needed to promote fairness and prevent the case where only some PiN are constantly serviced while others (in places that are far from NGOs) are constantly kept with no service. Formally, we introduce a binary indicator variable $I(NGO_i, PiN_j)$ which evaluates to one whenever the j th PiN is assigned (or paired) with the i th NGO. Using this notation, we can formally define the three social welfare objectives as:

$$J_{\% \text{ serviced}} = \frac{1}{N_{PiN}} \sum_{i=1}^{N_{NGO}} \sum_{j=1}^{N_{PiN}} I(NGO_i, PiN_j)$$

¹For a thorough overview of different measures for assessing and measuring hunger we refer the reader to [8].

$$J_{\text{hunger}} = \frac{1}{N_{\text{PiN}}} \sum_{j=1}^{N_{\text{PiN}}} h_j$$

$$J_{\text{waste}} = \sum_{i=1}^{N_{\text{NGO}}} s_i - \sum_{j \in \{k \mid I(\text{NGO}_i, \text{PiN}_k) = 1\}} h_j$$

Therefore, the problem of maximizing the social welfare can be defined as a search problem over the NGO locations c_i and pairings $I(\text{NGO}_i, \text{PiN}_j)$ such that:

$$\underset{c_i, I(\text{NGO}_i, \text{PiN}_j)}{\text{maximize}} \quad J_{\% \text{serviced}} + J_{\text{hunger}} + J_{\text{waste}} \quad (1)$$

$$\text{subject to :} \quad I(\text{NGO}_i, \text{PiN}_j) = 1 \Leftrightarrow \|c_j - l_j\| \leq r_j \quad (2)$$

$$\sum_{i=1}^{N_{\text{NGO}}} I(\text{NGO}_i, \text{PiN}_j) \leq 1 \quad (3)$$

$$\sum_{j \in \{k \mid I(\text{NGO}_i, \text{PiN}_k) = 1\}} h_j \leq s_i \quad (4)$$

where the first constraint ensures that a PiN must be in the neighborhood of an NGO to receive service, while the second constraint ensures that a PiN is assigned to at maximum one NGO. This in turn ensures that $J_{\% \text{serviced}}$ increases only if more people are assigned services. The maximum of $J_{\% \text{serviced}}$ is achieved if and only if all the individuals have been served. The third constraint ensures that an NGO can not serve more than the available resources.

We note that this above search problem is highly combinatorial. To reduce the complexity of the search problem, we rely on a sub-optimal solution in which we search for the NGO locations c_i separately from the PiN matching. That is, we search first over the possible locations of the NGOs that potentially lead to maximizing the pairing. Once we fix the NGOs locations, we search over all possible pairings for these fixed locations, i.e., we solve instead the following optimization problem subject to the previous constraints.

$$\underset{I(\text{NGO}_i, \text{PiN}_j)}{\text{maximize}} \quad \underset{c_i}{\text{maximize}} \quad J_{\% \text{serviced}} + J_{\text{hunger}} + J_{\text{waste}}$$

3.1 Resource Assignment and Pairing

We start by describing the algorithm for resource assignment and pairing (the outer maximization in (1)) while assuming that all NGO locations are fixed. We model the assignment problem as a max-flow problem where available services s_i are the source of the flow and the hunger level h_j is the target of the flow. More details regarding the assignment algorithm are included below:

Step 1: To generate a flow graph, each NGO and PiN is represented as a node. Edges exists between an NGO and a PiN when the PiN lies within the reachable radius of the NGO (constraint (2)). Each edge has a flow capacity of 1. Every node representing a PiN is connected to the sink pseudo-node, which does not correspond to any real PiN. The sink node is used to quickly identify if a PiN has already been assigned to an NGO.

Step 2: For each NGO_i , we attempt to push 1 unit of flow, corresponding to a unit of food, to the first neighboring PiN_j . If the edge between the PiN_j and the sink pseudo-node has not yet reached capacity, then we know that this pairing between the NGO_i and PiN_j is a valid pairing. If the edge between the PiN_j and the sink pseudo-node has reached capacity, signifying that this individual

Algorithm 1 RESOURCE PAIRING ALGORITHM

Input: PiN, NGO

Output: pairs

```

1: function RESOURCEPAIRING( $\text{PiN}, \text{NGO}$ ):
2:    $\text{pairs} = []$ 
3:   Generate flow graph  $g$  based on  $\text{PiN}$  and  $\text{NGO}$ 
4:   Sort  $\text{PiN}$  from hungriest to least hungry
5:   Sort  $\text{NGO}$  from lowest to highest available resources
6:   for  $\text{NGO}_i$  in  $\text{NGO}$  do
7:     for  $s_i$  units of flow do
8:       for  $l_j$  inside  $r_i$  of  $c_i$  do
9:         if flow( $\text{PiN}_j, \text{sink}$ ) = 0 then
10:          Append ( $\text{NGO}_i, \text{PiN}_j$ ) to  $\text{pairs}$ 
11:          Exit innermost for-loop
12:        else REVERSEFLOW( $\text{NGO}_i, \text{PiN}_j, \text{pairs}$ )
13:   return  $\text{pairs}$ 
14: function REVERSEFLOW( $\text{NGO}_i, \text{PiN}_j, \text{pairs}$ ):
15:   for  $\text{NGO}_{i'} \neq \text{NGO}_i$  in  $\text{PiN}_j.\text{suppliers}$  do
16:     if flow( $\text{NGO}_{i'}, \text{PiN}_j$ ) = 1 then
17:       for each  $\text{PiN}_{j'} \neq \text{PiN}_j$  inside  $r_{i'}$  of  $c_{i'}$  do
18:         if flow( $\text{PiN}_{j'}, \text{sink}$ ) = 0 then
19:           Remove ( $\text{NGO}_{i'}, \text{PiN}_j$ ) from  $\text{pairs}$ 
20:           Add ( $\text{NGO}_{i'}, \text{PiN}_{j'}$ ), ( $\text{NGO}_i, \text{PiN}_j$ ) to  $\text{pairs}$ 
21:       return
22:     else REVERSEFLOW( $\text{NGO}_{i'}, \text{PiN}_{j'}, \text{pairs}$ )

```

has already received flow from another $\text{NGO}_{i'}$, then we enter the reverse flow stage. In this stage, we attempt to push flow reverse through edge $(\text{NGO}_{i'}, \text{PiN}_j)$ to a new neighbor $\text{PiN}_{j'}$. Then, we are either able to reach the sink pseudo-node or otherwise attempt the reverse flow process again. This process is repeated until either we are able to successfully push flow to the sink pseudo-node or when we have exhausted all possibilities.

Step 3: If there is extra food available at a supplier after assigning food to all reachable individuals, then it is considered waste.

Note that the above max-flow algorithm optimizes only for $J_{\% \text{serviced}}$ and J_{waste} . To optimize for J_{hunger} as well, we order the nodes that represents PiN according to their hunger level. Therefore, the algorithm will always consider the PiNs with higher levels of hunger before assigning resources to the other PiNs. This process is summarized in Algorithm 1.

In Algorithm 1 line 4, we sort individuals by hunger and service requests in that order. If a certain individual has been serviced by a NGO in a previous round, his/her hunger will be reduced. This individual effectively gets moved to the end of the list. This gives the subsequent individuals on the list who were not serviced the opportunity to be assigned to NGOs, and hence maximizing J_{hunger} .

3.2 Resource Allocation

In the case where we have only permanent food suppliers, we can directly apply the pair-generating algorithm described in the previous subsection. Otherwise, we need to determine and recommend new locations for the mobile providers. In this scenario, we first obtain the resource pairing assignments from considering the resources provided by only the permanent NGOs. This allows us to identify the remaining individuals that were unassigned to any NGO. We would like to provide resources to these unserviced PiN by moving the mobile providers towards them. In other words, we would like to identify k cluster centers from the subpopulation of unassigned individuals, where k is the number of mobile providers available. This type of clustering problem is commonly solved by applying the K-Means algorithm.

Algorithm 2 SUPPLIER RESOURCE DISTRIBUTION**Input:** $PiN, NGO = NGO^f + NGO^m$

```

1: procedure RESOURCEDISTRIBUTION( $PiN, NGO$ ):
2:    $pairs = \text{RESOURCEPAIRING}(NGO^f)$ 
3:    $k = \text{COUNT}(NGO^m)$ 
4:   if  $k > 0$  then
5:      $centers = \text{MOBILESUPPLIERPLACEMENT}(pairs, PiN, k)$ 
6:     Update locations of  $NGO^m$  to  $centers$ 
7:      $pairs = \text{RESOURCEPAIRING}(NGO)$ 
8:   Distribute food according to  $pairs$ 
9: function MOBILESUPPLIERPLACEMENT( $pairs, PiN, k$ ):
10:    $centers, unfed = []$ 
11:   for  $PiN_i$  in  $PiN$  do
12:     if  $PiN_i$  not in  $pairs$  then  $unfed.append(PiN_i)$ 
13:   if  $\text{COUNT}(unfed) < k$  then
14:      $centers = unfed$ 
15:   else:  $db\_model = \text{DBSCAN.FIT}(unfed)$ 
16:      $kmeans\_model = \text{KMEANS.FIT}(db\_model.core\_points, k)$ 
17:      $centers = kmeans\_model.cluster\_centers$ 
18:   return  $centers$ 

```

However, applying K-means directly to unpaired PiNs forces every individual to be assigned to some center. Therefore, some individuals may be located far from its assigned cluster. These individuals would pull on the cluster center, potentially shifting the center's location such that it no longer covers most of the population density in the cluster. To remove the effect of these outliers, we first apply the Density-based spatial clustering of applications with noise (DBSCAN) algorithm to the unpaired population. The DBSCAN is able to distinguish between core points, those that are a part of a high density area, and non-core points, those that lie far from neighboring points. Finally, we apply K-means to the subset of core points to find reliable cluster centers. This process is summarized in Algorithm 2. Once the location for the mobile centers are identified, we rerun the pairing algorithm using the resource contributions from both stationary and mobile NGOs.

4 SIMULATION RESULTS

In order to test our resource allocation and assignment algorithms, we built a bounded 2-dimensional grid environment containing simulated NGOs and PiNs. Each tile in the grid corresponds to a roughly 330 feet by 330 feet city block. Multiple PiNs can occupy the same tile and share this space with an NGO, but only at most one NGO is allowed to be located per tile. We emulate a few simple migratory patterns in the following manner. Each PiN is randomly assigned one of four possible movement patterns: (1) stationary, (2) strictly along a horizontal path, (3) strictly along a vertical path, and (4) looping along a rectangular path. At each time step, the PiNs move according to their designated movement patterns.

Here, we present a comparison between the traditional resource allocation method in which NGOs distribute resources to PiNs in the vicinity without coordinating with other providers, and our proposed method of resource distribution while assuming the ground-truth location of all homeless people is known. Our results are based on an environment set to 13 tiles wide by 11 tiles long with 1 permanent supplier, 2 mobile suppliers, and 50 PiNs.

As shown in Figure 2, our proposed method shows a 25% reduction in food waste and average hunger, while the consistently reaching almost the same number of individuals. We can understand and explain the improvements from the perspective of our algorithm design. In the traditional allocation method, certain individuals could repeatedly receive resources from multiple centers,

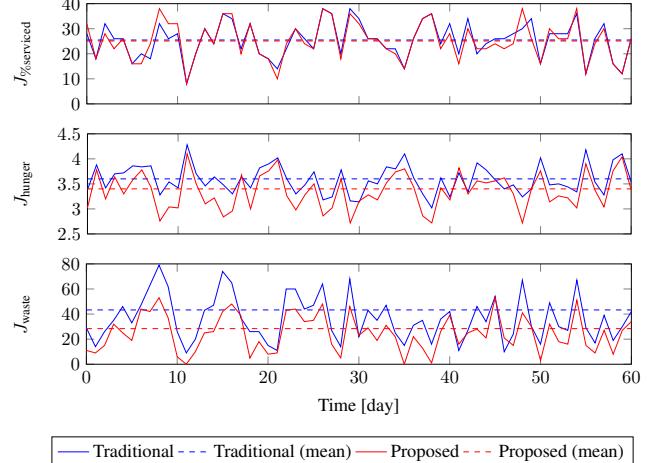


Figure 2: Performance of our framework vs traditional allocation methods across different criteria for measuring social welfare.

while slightly farther individuals would remain neglected. Our algorithm specifically avoids feeding the same individual multiple times unless other individuals have been first considered. This has a two-fold effect: (1) we reduce the amount of food waste since we are no longer over-feeding select individuals, and (2) we reduce hunger across more individuals.

5 CONCLUSIONS

While there have been a handful of recent initiatives using mobile technology to broadcast information about available resources to the homeless, to our knowledge, none have taken a formal modeling approach to ensure optimal resource distribution by coordinating multiple providers. In this paper, we take the first steps towards creating socially responsible smart cities. In particular, we propose a framework to allocate mobile resources and pair them with people-in-need in a manner that maximizes social welfare. We show that by smartly coordinating the efforts of NGOs, a smart city can reduce the resources wasted by 25%. In the future, we would like to adjust our algorithm to take into account how NGOs can share resources. We also plan to continue developing other aspects of our cyber-physical framework, such as crowd-sourced information gathering, population location prediction, and real-time implementation of the proposed framework.

REFERENCES

- [1] Barbara Sard, "Number Of Homeless Families Climbing Due to Recession," Center on Budget and Policy Priorities, Special Series: Economic Recovery Watch, January 2009.
- [2] The U.S. Department of Housing and Urban Development, "The 2014 Annual Homeless Assessment Report (AHAR) to Congress, PART 1: Point-in-Time Estimates of Homelessness," October, 2014.
- [3] The U.S. Department of Housing and Urban Development, "The 2013 Annual Homeless Assessment Report (AHAR) to Congress, PART 2: Estimates of Homelessness in the United States," October, 2014.
- [4] Invisible people (blog), <http://invisiblepeople.tv/blog/2012/08/why-i-support-regulating-the-public-feeding-of-homeless-people>
- [5] E. Rice, A. Lee, S. Taitt, "Cell Phone Use among Homeless Youth: Potential for New Health Interventions and Research," Journal of Urban Health, Vol. 88(6), pp: 1175–1182, Dec 2011.
- [6] <http://www.link-sf.com/>
- [7] NYC Department of Homeless Services, "Street Homelessness & Outreach Program", <http://www.nyc.gov/html/dhs/html/outreach/outreach.shtml>.
- [8] Laurie DeRose, Ellen Messer, and Sara Millman, "Who's hungry? And how do we know? Food shortage, poverty, and deprivation", United Nations University Press, 1998.
- [9] SOCIUS project website: <https://vin01188.github.io/sociusinfo/>