Connected Placement of Disaster Shelters in Modern Cities

Huanyang Zheng and Jie Wu
Dept. of Computer and Info. Sciences
Temple University

Road Map

- Introduction
- Model and Formulation
- Algorithms
- Experiments
- Conclusion

Introduction

Facebook Aquila

-Internet Access

-Disaster recovery

-Controllable mobility

Google Balloon

-Internet Access

-Search and rescue

-Not mobile

Introduction

Disasters and Shelters

-France has witnessed unprecedented attacks in Paris.

-Disaster shelters can handle large-scale emergencies.

-Hawaii uses disaster shelters for hurricanes.

Can we apply recent technologies to disasters?

-Facebook Aquila and Google Balloon can connect disaster shelters.

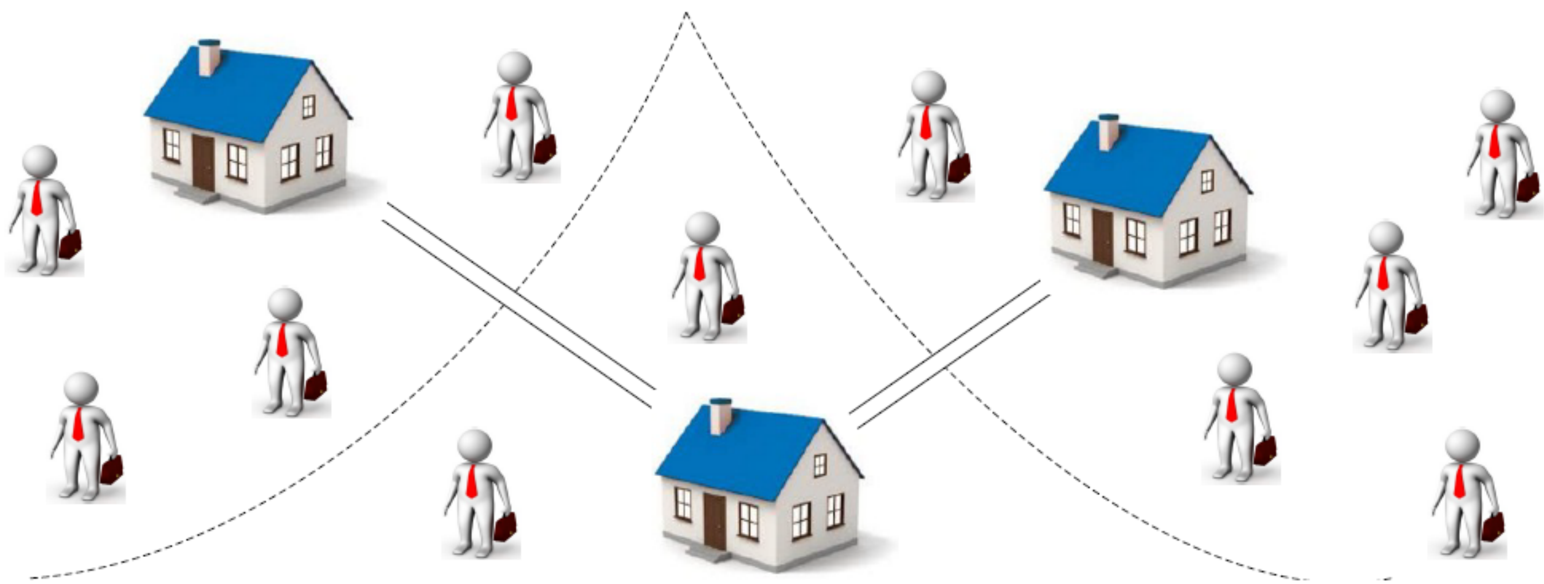
Introduction

Disaster recovery with Aquila and Balloon

-Connected placement of disaster shelters

-People go to shelters after disasters

-Aquilia and Balloon provide connections among shelters



Model and Formulation

Scenario

-Three-dimensional Euclidean surface represents city

-Use establishment cost to place disaster shelters

$P = \{p\}$ is a set of people with known locations

$S = \{s\}$ is a set of disaster shelters; l_s, c_s, and e_s are location, capacity, and establishment cost

Establishment cost depends on capacity

Function $e_s = F(c_s)$ is known

Model and Formulation

Objective

-Minimize the total establishment costs of shelters.

Constraints

1. Distance constraint: people must be assigned to disaster shelters within a certain range of r
2. Capacity constraint: disaster shelters must have sufficient capacity to hold incoming people
3. Connection constraint: the induced neighboring graph of the disaster shelters includes all disaster shelters and is connected (by Aquilia/Balloon) within range R

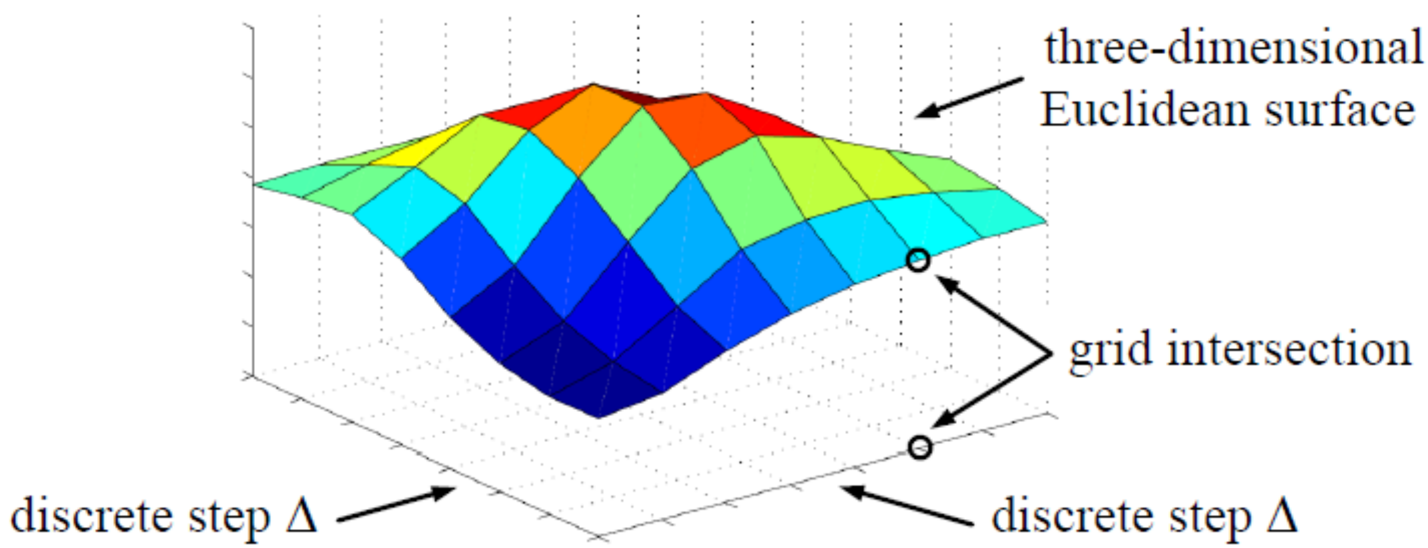
Model and Formulation

Problem hardness: NP-hard

By reduction from geometry set cover problem

Discrete framework

1. Disaster shelter locations are discrete



Two-Stage Algorithm

Two-stage algorithm

1. 1st stage: solve distance and capacity constraints by mapping to the set cover problem
2. 2nd stage: solve connection constraint by mapping to the Steiner tree problem

Approximation ratio: $a_1 + a_2 \times (1 + \lfloor 2r/R \rfloor \times a_1)$

- a_1 and a_2: bounds for set cover and Steiner tree
- r and R: range for distance and connection constraints

Two-Stage Algorithm

1: **for** each location, $l \in L$ **do**
2: **for** each discrete capacity, $c_{\min} \leq c \leq c_{\max}$ **do**
3: Calculate the set of covered people, P_s, for the disaster shelter, s, with $l_s = l$ and $c_s = c$. Map s to a set and P_s to elements in s. Use the establishment cost, $e_s = F(c_s)$, as the set weight.
4: For the mapped sets and elements, use an approximation algorithm to solve the geometric set cover problem. The resultant collection of selected sets are mapped back to the disaster shelters, including locations and capacities.
5: Map locations in L as nodes in a graph (nodes are connected if their distance is no larger than R). Use an approximation algorithm to solve a Steiner tree problem, which selects nodes to connect the terminals (disaster shelters in line 4). Selected nodes are recorded as new disaster shelters with minimum capacities.
6: **return** the resultant disaster shelters in lines 4 and 5.

Tree Growth Algorithm

Tree growth algorithm

1. -Iteratively establishes a shelter while maintaining the connection constraint with existing shelters

2. -Iteration terminates when the distance and capacity constraints are satisfied

Approximation ratio: $\frac{R}{\min_s \gamma_s} \log |P|$

R: range for connection constraint

r_s: range of disaster shelter

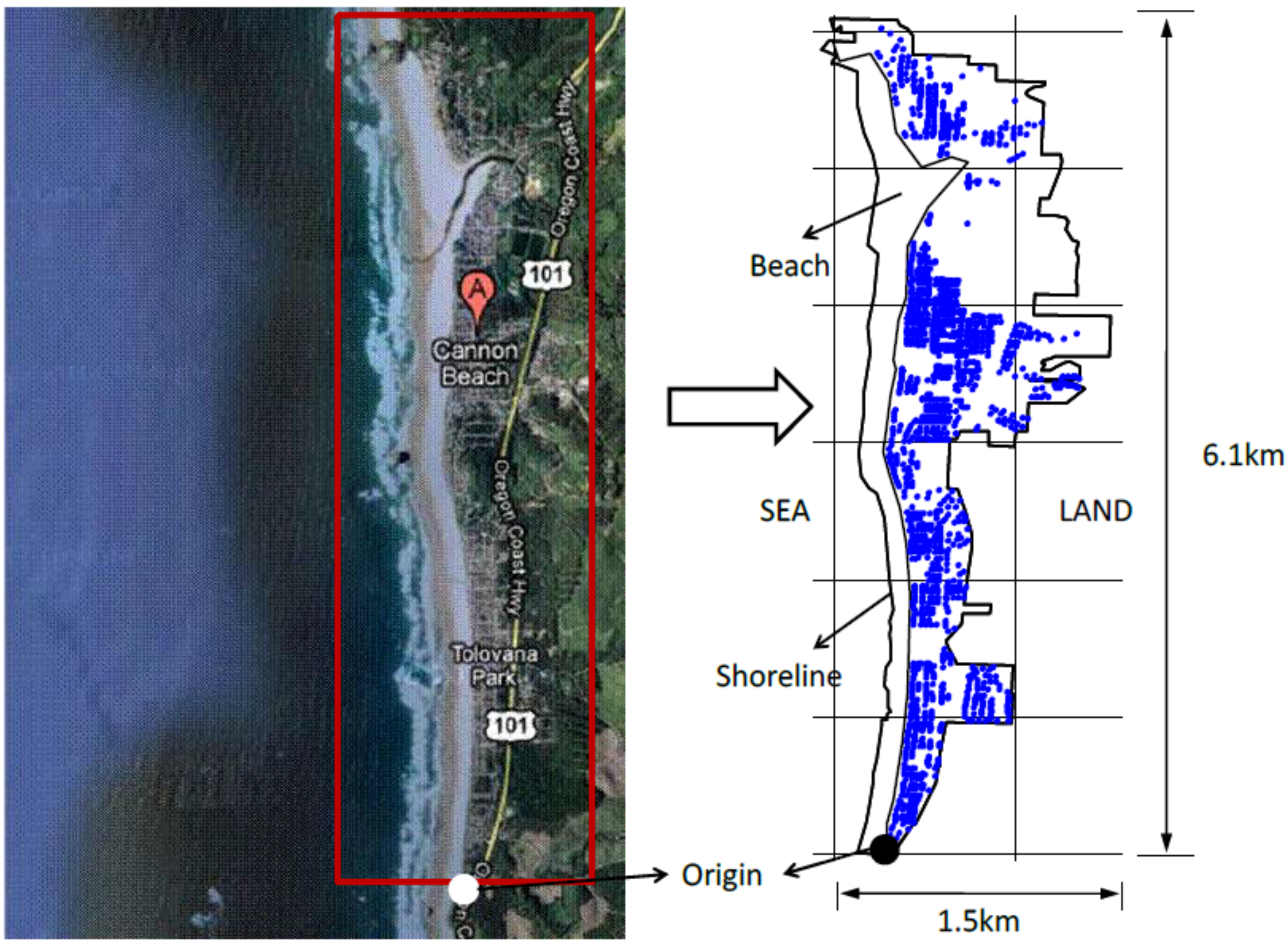
P: set of people

Tree Growth Algorithm

1: Initialize $S = \emptyset$.
2: Among locations $l \in L$, place a disaster shelter, s, with a capacity, c_s that minimizes $\frac{F(c_s)}{|P_s|}$.
3: Add s to S, and then, remove P_s from P.
4: **while** $P \neq \emptyset$ **do**
5: Find the set of locations, L', that are adjacent to S.
6: Among locations $l \in L'$, place a disaster shelter, s, with a capacity, c_s, that minimizes $\frac{F(c_s) \cdot D(l,S)}{|P_s|}$.
7: Add s to S, and then, remove P_s from P.
8: **return** S, their locations, and capacities.

Experiments

Data set for city of Cannon Beach, OR



Experiments

Comparison Algorithms

1. CLS: bounded clustering method. Disaster shelters are established at the center of each cluster to satisfy the distance and capacity constraints. To satisfy the connection constraint, a spanning tree is built.

2. DAC: divide and conquer method. DAC iteratively divides people into two clusters through a k-means algorithm. The centers of these two clusters are connected through placing disaster shelters with minimum capacities. The iteration terminates when the distance and capacity constraints are satisfied.

Experiments

Settings

1. Location discretization: 100m by 100m squared
2. Shelter capacity limits: 10 to 100 people

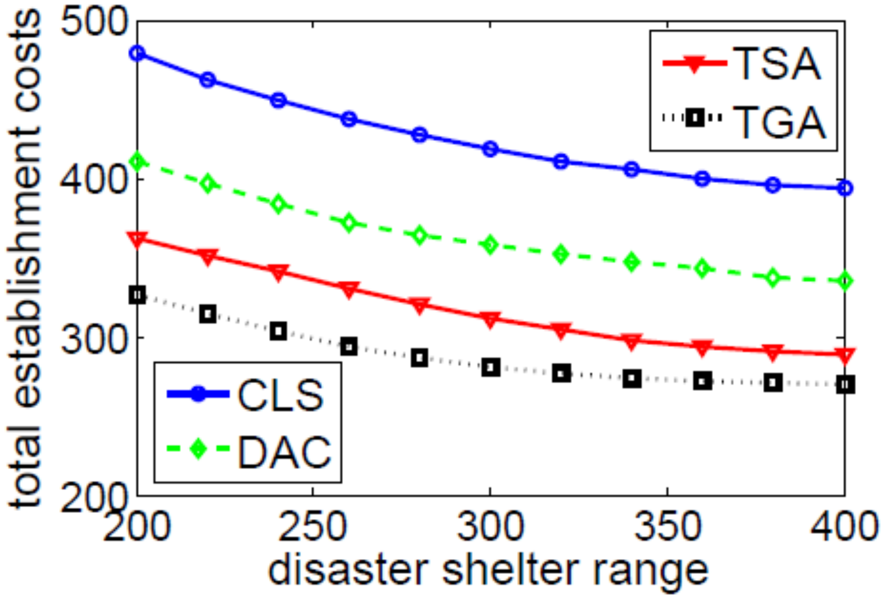
3. Establishment cost function
    1. Sub-linear: $F(c_s) = \sqrt{c_s}$
    2. Linear: $F(c_s) = 2.4 + 0.076 \times c_s$

Distance constraint: r ranges from 200m to 400m

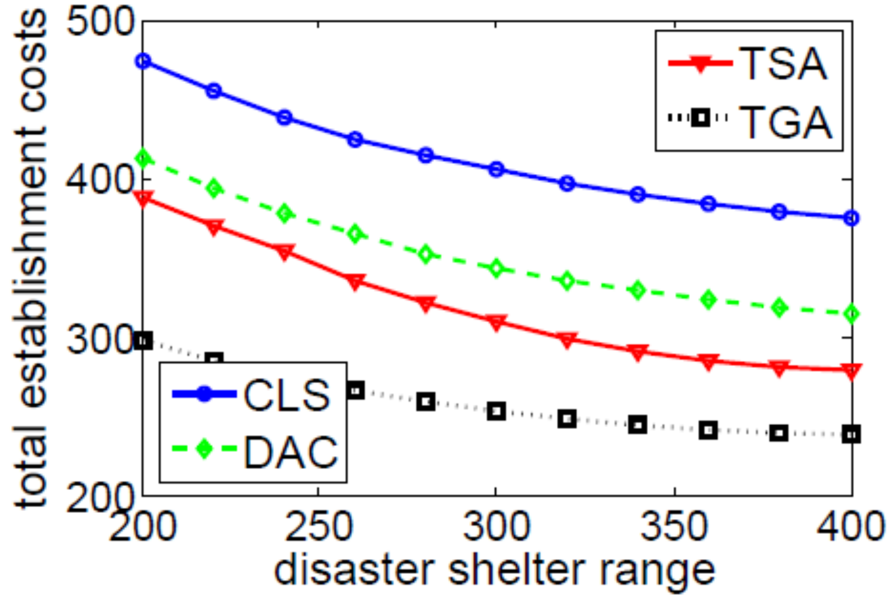
Connection constraint: R is 400m or 600m

Experiments

Sub-linear establishment cost



(a) $R = 400$m.

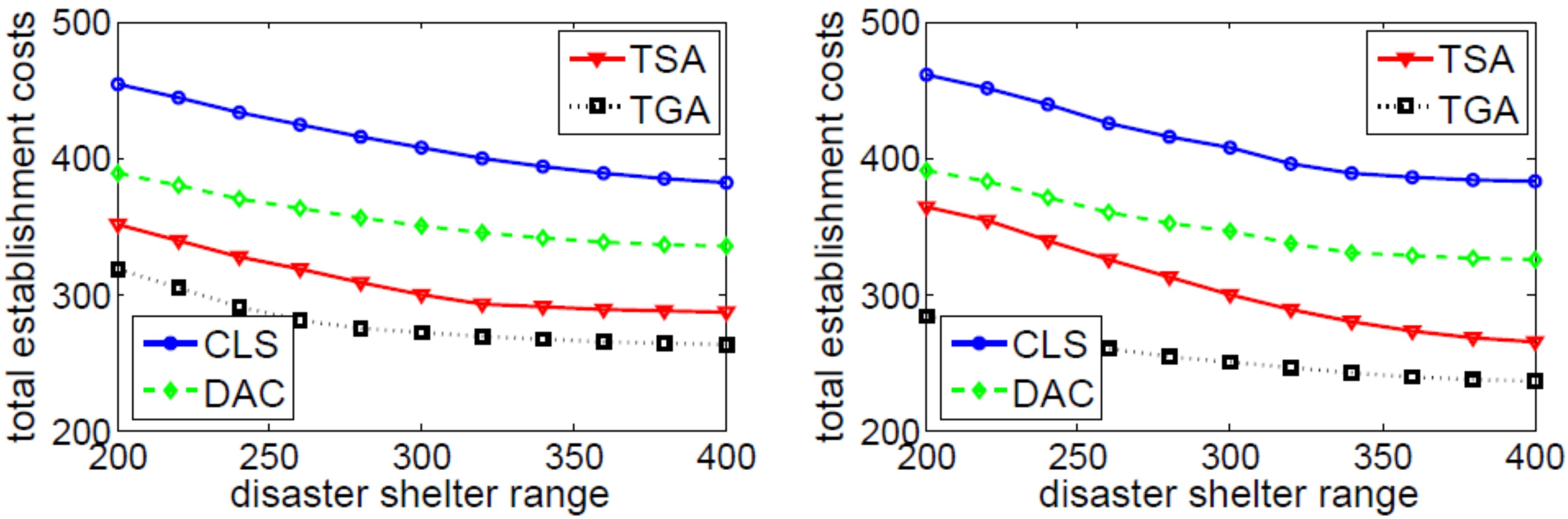


(b) $R = 600$m.

A larger disaster shelter range (for distance constraint) brings lower establishment costs

Experiments

Linear establishment cost



(a) $R = 400\text{m}$. (b) $R = 600\text{m}$.

Linear establishment cost function brings a higher total cost than sub-linear establishment cost function

Conclusion

Shelter placement

1. -Distance, capacity, and connection constraints
2. -Aquilia/Balloon is used to connect shelters

3. Two approximation algorithms
    1. -Two-stage algorithm
    2. -Tree growth algorithm