Assignment: CoIoTe

Problem description and rules

Roberto Tadei, Luca Gobbato, Dario Mana

Contents

Contents

- The problem description
- The optimization problem
- Your objectives
- Rules
- Material (Portale della didattica)

Problem description

Swarm Joint Open Lab

Dario Mana

CoIoTe Optimization

Connecting the Internet of Things

TIM

What is CoIoTe?

- It is a new service being developed by TIM/Telecom Italia which is capable of **connecting Internet of Things (IoT) objects to Internet, without** requiring the deployment of a **new ad-hoc wireless network**.

- PROBLEM: Smart City IoT objects need connection to Internet in order to **communicate measures and receive configurations**; for example:

  - **Dumpsters** have to send to the «cloud» their level of waste filling
  - Smart **street lights** can measure the level of filth that is obfuscating their light and, so, request intervention for cleaning
  - **Traffic lights** have to be reconfigured based on the traffic conditions
  - **Manholes** may send an alarm when they get hacked (for stealing metal) or are clogged by leaves
  - **Bus stops** show the times of arrival of public modes and detect their transit
  - **Bike sharing** columns detect the presence or absence of available bicycles
  - … … …

What is CoIoTe?

- STANDARD SOLUTIONS for connecting 'on-the-streets' objects to Internet are:
  1. Use the mobile network => TIM is very happy, but…
     - Very costly for the companies deploying smart objects within the city
     - Very battery hungry: batteries of deployed objects should last for months or even years without requiring human intervention
  2. Deploy a new ad-hoc wireless network: for example an LPWAN (Low Power Wireless Area Network):
     - LoRa – Long Range,
     - SigFox,
     - LTE-M – Machine to Machine

What is CoIoTe?

- STANDARD SOLUTIONS for connecting 'on-the-streets' objects to Internet are:
  1. Use the mobile network => TIM is very happy, but…
     - Very costly for the companies deploying smart objects within the city
     - Very battery hungry: batteries of deployed objects should last for months or even years without requiring human intervention
  2. Deploy a new ad-hoc wireless network: for exa
     LPWAN (Low Power Wireless Area Network):
     - LoRa -
     - Sig Fox
     - M

Example:
Dubai + LoRa

This is very costly for TIM!

What is CoIoTe?

- ALTERNATIVE SOLUTION: entering **CoIoTe**…
- Opportunistic connection for the Internet of Things: it leverages:
  1. **Domestic Wi-Fi networks** that trespass domestic walls and "invade" the street
  2. **Hotspot Wi-Fi networks** created on-the-fly by the **smartphones** of TIM users passing by* Internet of Things objects (e.g. dumpsters) needing to send data to Internet

* TIM, given consensus by the user, is able to detect his position

Users must be adequately Rewarded!

CoIoTe Assignment

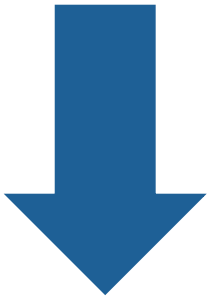
- Optimize a system capable of **assigning missions to mobile users**: users have to be sent near to, e.g., dumpsters within the city, in order to give the dumpsters Internet connectivity for some seconds. Such dumpsters need connectivity so they can send their waste filling level to the waste company main server => the waste company may then better schedule the emptying of dumpsters
- **Mobile users** will not participate without some kind of **reward**
- Trade-off between:
  1. the **level of connectivity service** guaranteed by TIM to the dumpster company and
  2. the **cost of engaging mobile users** so that they:
     1. change their habitual route and
     2. let TIM control the hotspot Wi-Fi network of their smartphones for some small amount of time

http://jol.telecomitalia.com/jolswarm/en/coiote-opportunistically-connecting-the-internet-of-things/

Optimization problem

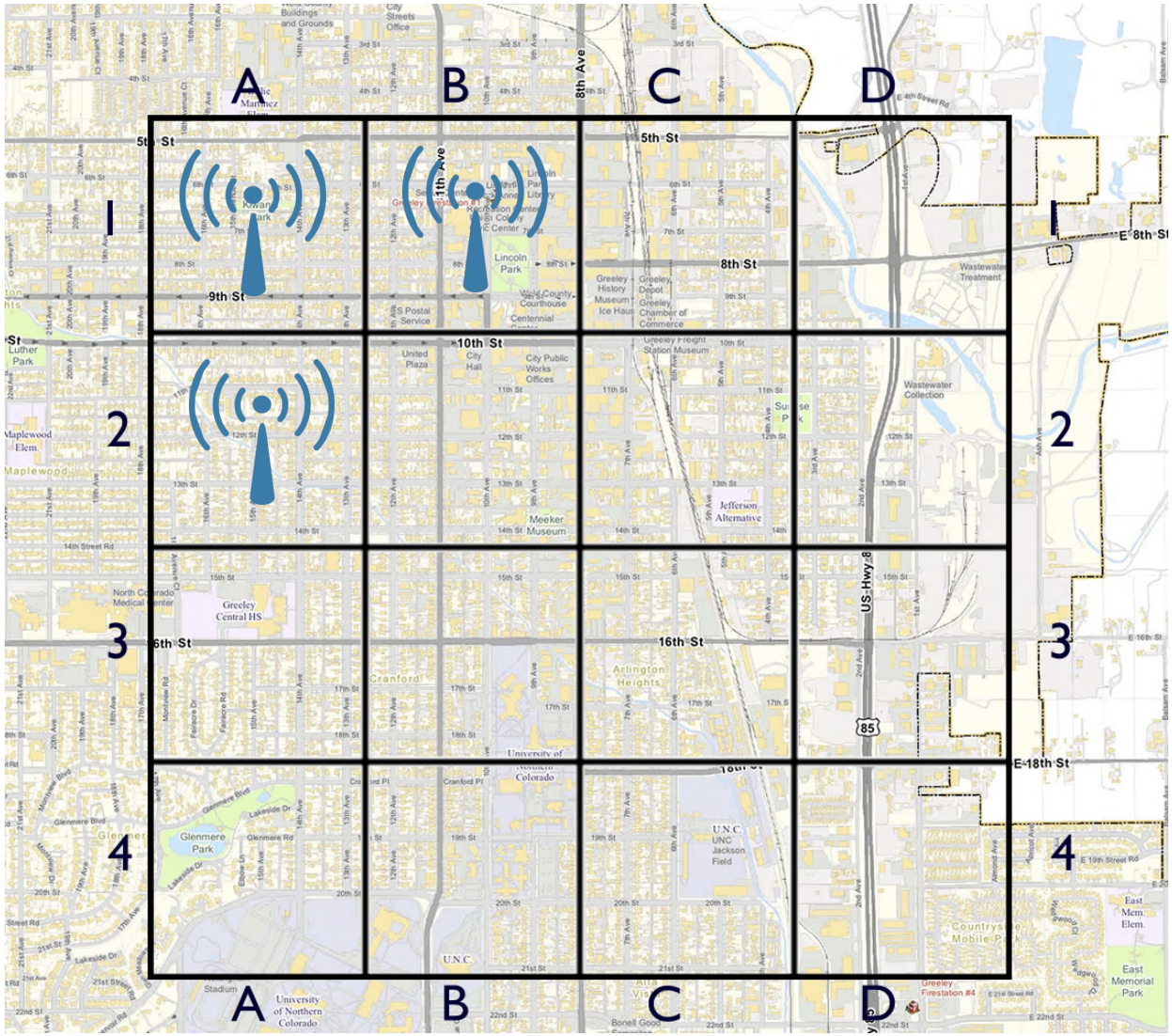
Optimization Problem

To ask to users to share the internet connection, we have to offer some reward



How to cover all the dumpsters minimizing the costs (rewards).

Optimization Problem

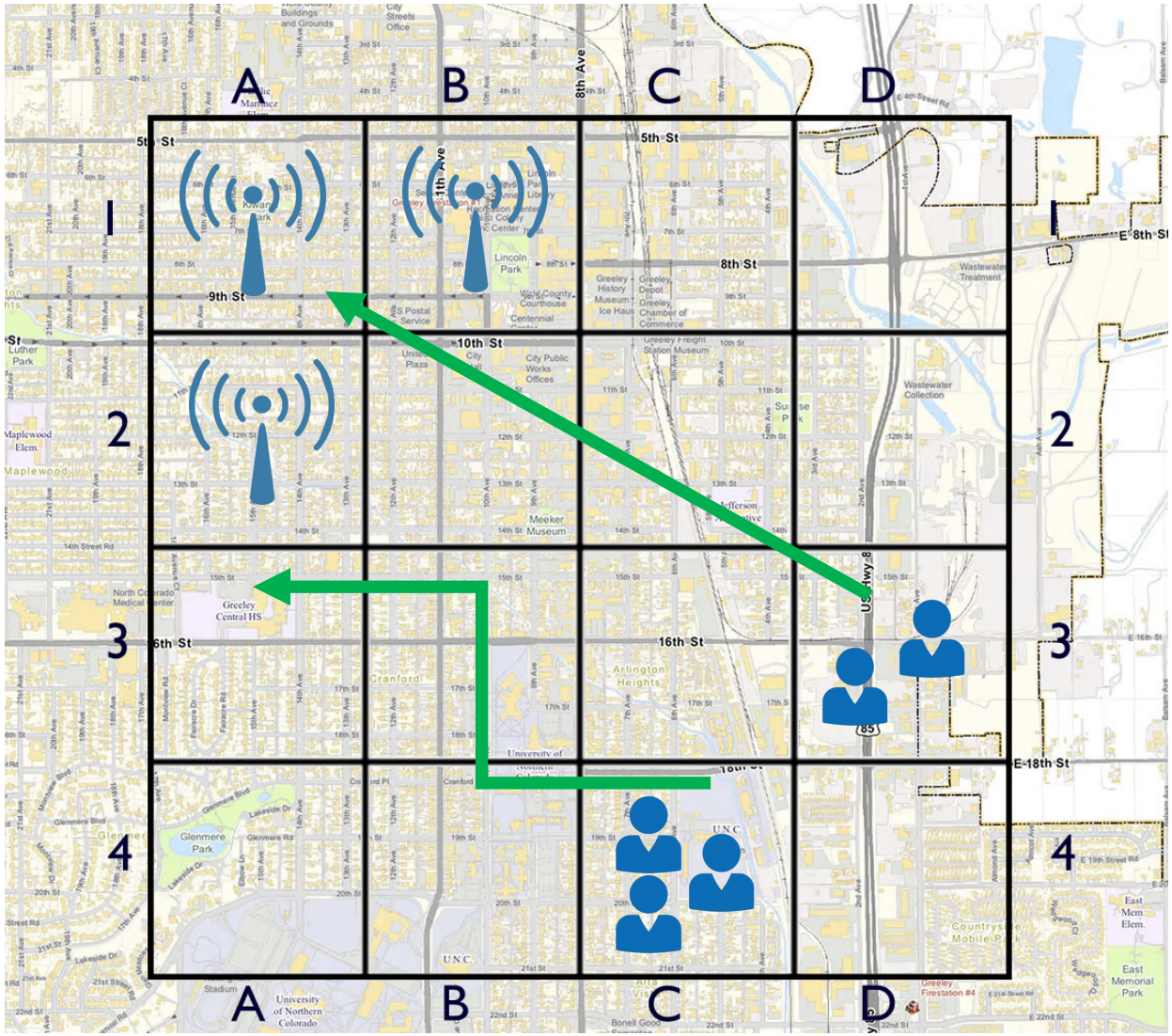


Antennas divide map in cells

People move through cells during the day

People can share connectivity with near dumpster

Optimization Problem

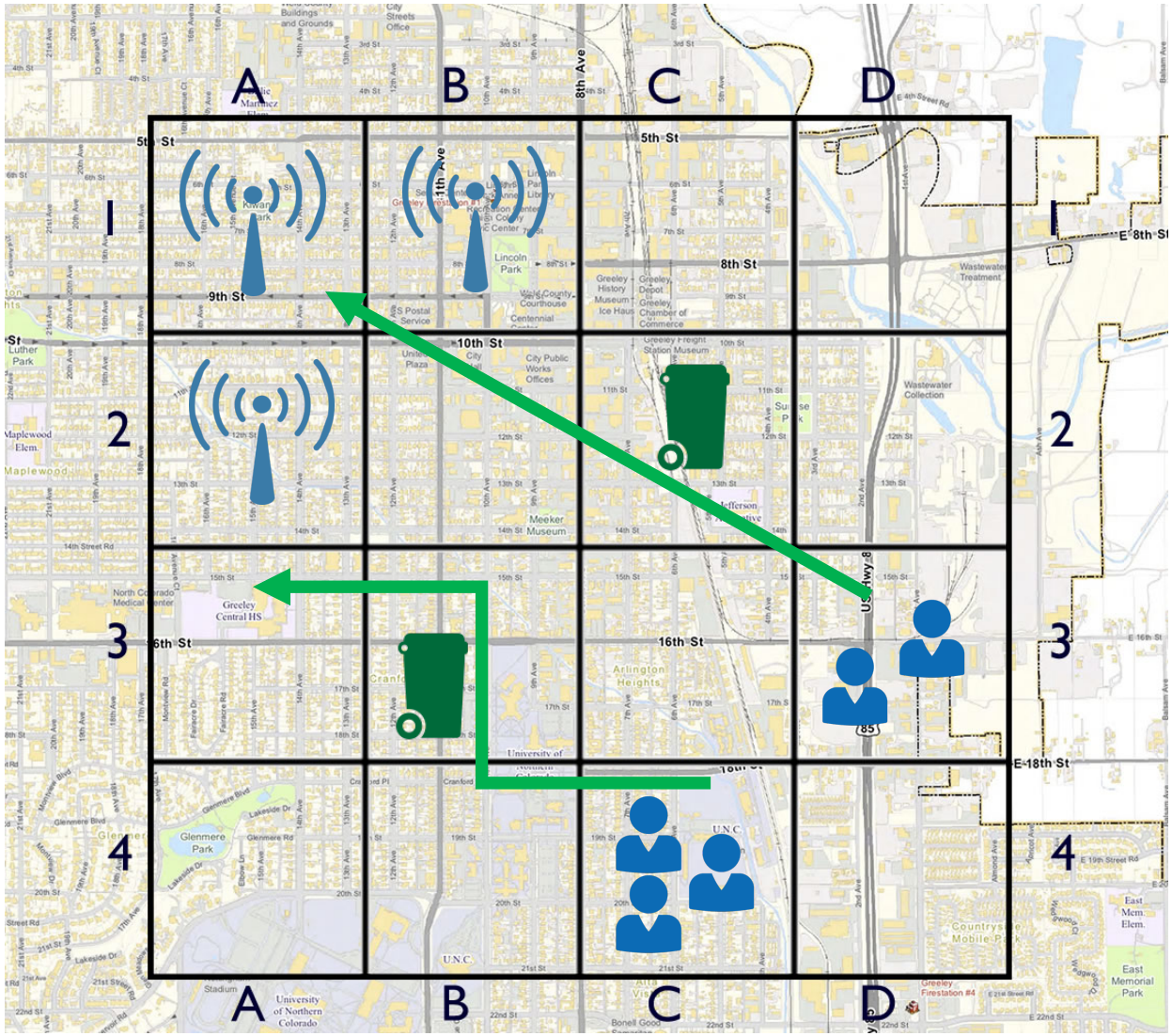


Antennas divide map in cells

People move through cells during the day

People can share connectivity with near dumpster

Optimization Problem



Antennas divide map in cells

People move through cells during the day

People can share connectivity with near dumpster

Optimization Problem

Objective function: The goal of the project is to **minimize the total amount of the reward**.

Constraint (1): **Every dumpster must be visited**

Constraint (2): The total amount of requests can not **exceed the number of people in the cell**

Domain: The number of people must be **an integer number.**

Optimization Problem

Sets

- $\mathcal{T}$ is the set of all time indexes. The cardinality of this set is T.
- $\mathcal{I}$ is the set of all cells. The cardinality of this set is I. We assume that the cells can be of three types: origin, destination and operational. These types define a partition of the set.
- $\mathcal{M}$ is the set of all users type. The cardinality of this set is M. We assume that the cells can be of three types.

Parameters

- c_{ij}^{tm} is the cost of the reward for a customer of type m in cell i at time t that goes in cell j.
- N_i is the number of tasks that must be done in the operational cell i during the time .
- n_m is the number of tasks that a customer of type m can do.
- θ_i^{tm} is the number of customer of type m in cell i during time step t.

Optimization Problem

Variables

- x_{ij}^{tm} is the number of customers of type m that are asked to do n tasks in cell j, starting from i at time t.

Optimization Problem

Our problem is then:

$$\text{minimize} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{m=1}^{M} c_{ij}^{tm} x_{ij}^{tm}$$

subject to:

$$\sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{i=1}^{I} n_m x_{ij}^{tm} \geq N_j \quad \forall\ j \ \in \ \mathcal{I} \tag{1}$$

$$\sum_{j=1}^{J} x_{ij}^{tm} \leq \theta_i^{tm} \quad \forall\ i \ \in \ \mathcal{I}\ t \ \in \ \mathcal{T}\ m \ \in \ \mathcal{M} \tag{2}$$

$$x_{ij}^{tm} \in \mathbb{N} \quad \forall\ i \ \in \ \mathcal{I}\ j \ \in \ \mathcal{J}\ t \ \in \ \mathcal{T}$$

Your objectives

Your objectives

- Develop a metaheuristic able to solve this problem
  - **Fast**: no more than 5 seconds
  - **Accurate**: solutions close to optimal solutions
- Program
  - C++, Java, python, matlab, …
  - Metaheuristic of your choice
    - Greedy
    - Tabu search (http://www.coin-or.org/Ots)
    - Genetic Algorithm (GALib http://sourceforge.net/projects/java-galib/)
    - …

Is my algorithm accurate??

- To measure the accuracy of the algorithm, you can compute the **optimality gap**
  - Solve the problem with your heuristic
  - Obtain an objective function value: v_{obj}
  - Given the optimal objective function: v_{opt}

$$\text{Optimality gap} = (v_{obj} - v_{opt})/v_{opt}$$

Optimality gap < 0 is impossible!!!

Rules

Mark

- Up to 10 points
  - Only with the exam: **20/30**
  - With the assignment: **30/30**

- The mark is a mix of
  - files you have uploaded are along our standard **(1 point)**
  - the optimality gaps **(6 points)**
  - the presentation **(3 points)**

Uploading files

- Upload your workgroup on the Web Portal (Section Elaborati) by **January 9th, 2017**

  - Zip file with name **Group_xx.zip**
  - Program, source and batch to solve all instances
  - Excel files with the results (format is given in Material)
  - Presentation: keynote, ppt, pdf

Chart and secret tests

- We run your program on our machine on instances provided in Portale della Didattica
- We test your program on a secret set of instances

- We build a chart sorting the total optimality gaps
  - Best groups get 6 points
  - Points decrease if optimality gaps increase
  - Acceptable optimality gap is smaller than **2%**

Our machine

- The machine will be used to test your heuristics has the following characteristics:
  - **Intel i7 2.80 GHz**
  - **4 CPUs, 8 threads**
  - **16 GB RAM**
  - **Windows 10 x64**

Presentation

- Max 9/10 minutes
- At least **half + 1** elements of the group must present the results (i.e. 3 for a group of 5 students)

- Topics
  - Your metaheuristic
  - Metaheuristic tuning
  - Results
  - Your considerations

Material (Portale della didattica)

Material

- Optimal solutions:
  - *optimal_solutions.csv*
- Instances to solve
- Code example in C++
  - Read an instance
  - Build a solution
  - Write a row in output file
  - Write the solution on a file
  - Can check the feasibility of a solution file
- Batch that solves all instances:
  - *solveAll.sh*

Instances

- Parameters
  - # cells: 30, 100, 300
  - # time periods: 1, 20
  - # user types: 3
  - Cost
    - U[2,10]
    - User type dependent: U[2,10](1+ln(2t+2))
  - 10 repetitions per combination

- Number of instances: 120

Output

- CSV files (excel)
  - One row for each instance
  - Columns
    - Instance name
    - Time
    - Objective function value
    - Number of people of type 1
    - Number of people of type 2
    - Number of people of type 3

Co_30_1_NT_0	0.014	1041	183	24	97
Co_30_1_NT_1	0.013	1756	246	111	172
Co_30_1_NT_2	0.017	2341	276	76	178