Monitoring by UAVs i.e. What? (some theses proposals)

Prof. Tiziana Calamoneri
Network Algorithms
A.y. 2017/18

Unmanned Aerial Vehicles (UAVs)



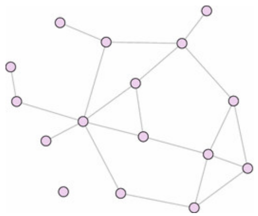
- UAVs are flying vehicles able to autonomously decide their route (different from drones, that are remotely piloted)
- Historically, used in the military, mainly deployed in hostile territory to reduce pilot losses
- Now, new applications in civilian and commercial domains:
  - weather monitoring,
  - forest fire detection,
  - traffic control,
  - emergency search and rescue

The Problem

Monitoring by UAVs

- Let be given an AoI whose map is known
- we have a fleet of m UAVs leaving from a safe location (v_0) each with a battery B
- in the AoI there is a set $S=\{v_1, \ldots, v_n\}$ of sites that must be examined (e.g. crumbled buildings after a hearthquacke)
- each site v_i needs a time t_i to be inspected
- each UAV must go back to v_o in order to recharge its battery when necessary; this takes time R, typically 5-10 times B
- we want to overfly $v_1, \ldots, v_n$ "as soon as possible" in order to collect data and possibly save people

??

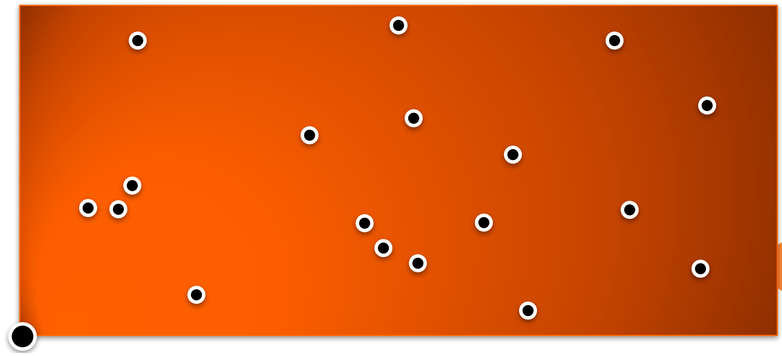


The Graph Model

The Graph Model (1)

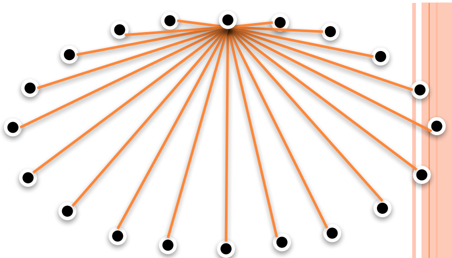
It is natural to model this problem as a graph problem:

- sites $v_1, \ldots, v_n$ + the depot v_0 are the $n+1$ nodes of the graph

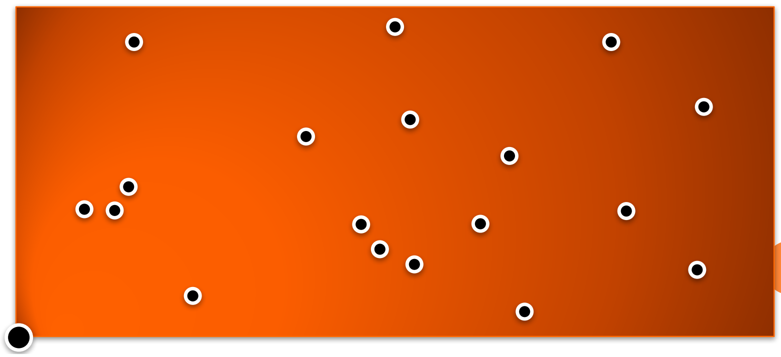


The Graph Model (2)

It is natural to model this problem as a graph problem:



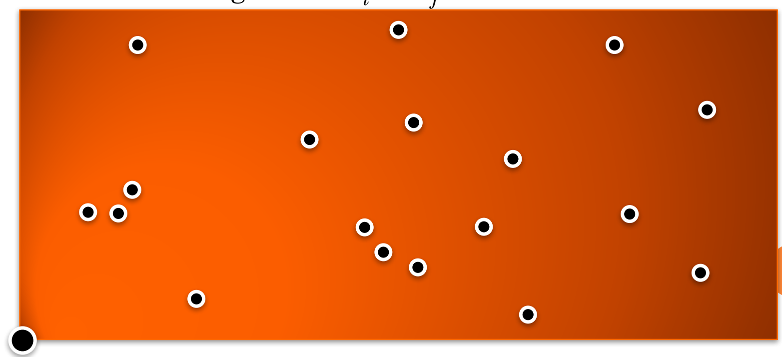
- there is an edge between each pair of nodes $\Rightarrow K_{n+1}$



The Graph Model (3)

It is natural to model this problem as a graph problem:

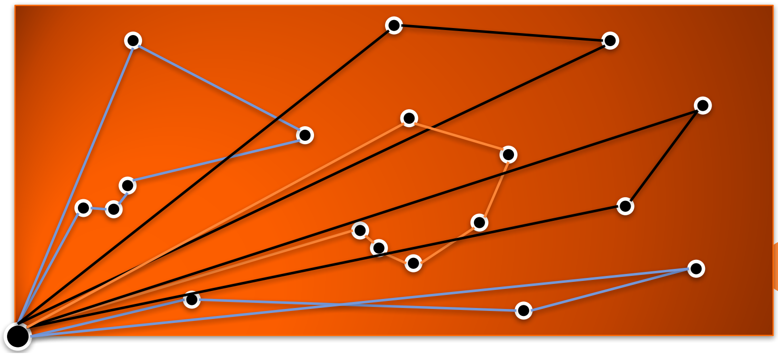
- Each UAV has a flying+inspection time bounded by B.
- for each pair of sites (v_i, v_j) we assume their distance (stored as an edge weight function $w(u_i, u_j)$) as the time a UAV needs to go from u_i to u_j.



The Graph Model (4)

- each UAV is characterized by a different color
- each UAV flies along a cycle (colored with the UAV color) and visits as many sites as it can (w.r.t. its battery constraint B), it goes back to the depot to recharge its battery (with time R) and it leaves again…

All sites need to be visited in the "shortest time".



The Graph Model (5)

What does it mean that the sites should be visited "as soon as possible"?

Different possibilities for the optimization function:

- Minimize the Total completion Time
- Minimize the Average Waiting Time
- Minimize the number of cycles
- …
- Note: Minimize the Overall Energy Consumption has no meaning

The Graph Model (6)

Similarities with many problems:

mTSP -multiple Traveling Salesperson

- m salespersons must overall cover n cities,
- objective: minimize the total length of the path
- no visiting times nor battery constraint

The Graph Model (7)

Similarities with many problems (cntd):

kTRPR -k-Traveling Repairperson Problem with Repairtimes

- given n points, construct k cycles, each starting at a common depot and together covering all the n points

calling the *latency* of a point the distance traveled (or the time elapsed) before visiting that point

- objective: minimize the sum of all latencies
- no battery constraint

The Graph Model (8)

Similarities with many problems (cntd):

mTRPD -multiple Traveling Repairperson Problem with Distance Constraints

- k repairpersons have all together to visit all the n customers
- they are not allowed to traverse a distance longer than a predetermined limit;
- Objective: minimize the total waiting time of all customers
- No repairtimes and not trivial to extend a solution by just adding them
- no. of cycles fixed to k

The Graph Model (9)

Similarities with many problems (cntd):

variants of **VRP** -vehicle routing problem

- Similar to mTRPD but there usually is a constraint on the number of visited customers per vehicle

The Graph Model (10)

Similarities with many problems (cntd):

variants of **MKP** -*Multiple Knapsack Problem*

- assigns a subset of n items to m distinct knapsacks,
- Objective: the total profit sum of the selected item is maximised
- no multiple rounds

The Graph Model (11)

Similarities with many problems (cntd):

TOP -team orienteering problem

- equivalent to the first round of our problem
- Objective: maximize the no. of covered sites
- Repeat many times until all sites have been covered does not seem a good idea…
- NOTE: From all these similarities we deduce that the problem is NP-hard and we cannot exploit any known result…

Monitoring an Area by UAVs

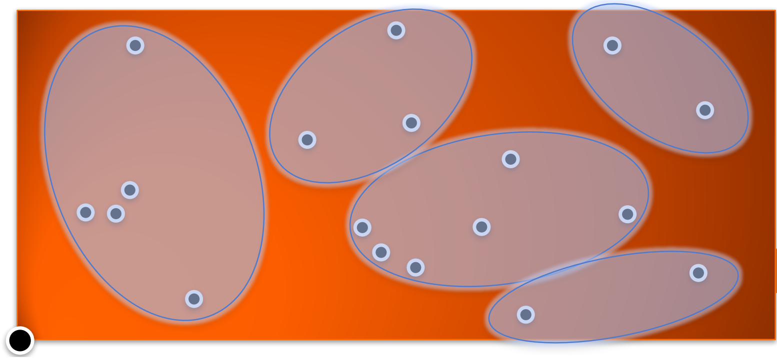
We have to study the problem by itself, going in several possible directions:

- due to its NP-completeness, approximate algorithms:
  - based on three main pahses:
    - clustering/matroid theory (greedy)
    - approximating TSP
    - scheduling
- MILP formulation
- reduction of the dimension of the problem

Monitoring an Area by UAVs (1)

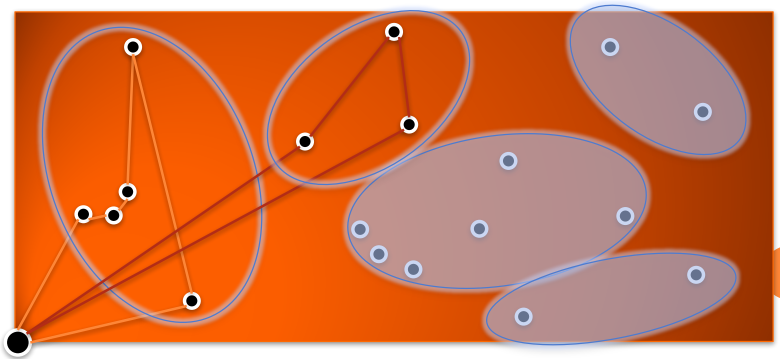
General idea: problem divided into three phases:

1. clustering: partition the sites so that each set can be covered (traversal+visiting times) within battery *B*.



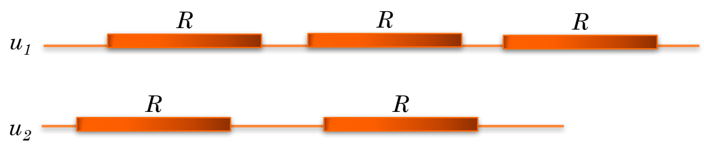
Monitoring an Area by UAVs (2)

2. approximating TSP: constructing a cycle covering all sites in each cluster (in fact performed together with the clustering, to guarantee the battery constraint)



Monitoring an Area by UAVs (3)

3. scheduling: all cycles must be distributed to UAVs so to guarantee min completion time or min latency



Monitoring an Area by UAVs (4)

Each one of the three phases can be implemented in several ways providing different solutions…

Problem 1: compare all the provided solutions in terms of goodness

Monitoring an Area by UAVs (5)

Instead of clustering sites, we can:

- enumerate all possible cycles passing through the depot that can be covered within battery B
- solve a min set cover.

Exploiting the fact that this system is a matroid, a greedy approach guarantees a very good approximation ratio but…

The no. of enumerated cycles is exponential in general…

Problem 2: reduce the space of the cycles so that the approximation ratio does not increase too much

Monitoring an Area by UAVs (6)

Provide a MILP formulation in order to compare optimum solutions of (small) instances with the approximate ones:

- in the graph model, the depot is transformed into two nodes, v_0^s and v_0^t and the graph is oriented
- define a variable family x_{ij}^k , $i,j=0, \ldots, n$, $k=1, \ldots, c$ such that it is =1 iff edge (i,j) is used by cycle k
- define a variable family z_i^k, $i=0, \ldots, n$, $k=1, \ldots, c$ such that it is =1 iff cycle k passes through node i
- …

Monitoring an Area by UAVs (7)

Provide a MILP formulation (cntd)

constrains:

- $\sum_{k=1}^{c} z_i^k \leq 1, \forall i = 1, \ldots, n$ (every site visited at least once)
- $\sum_{(v_i,v_j)\in E} w(v_i, v_j) x_{ij}^k + \sum_{i=1}^{n} t_i z_i^k \leq B, \forall k = 1, \ldots, c$ (battery constraint)
- $x_{ij}^k \leq z_i^k \; \forall (v_i, v_j) \in E, \forall k = 1, \ldots, c$ (the k-th cycle passes through site i iff it is in fact assigned to it)
- $\sum_{i=0}^{n} x_{ij}^k = 1 \; \forall j, k$ (only 1 edge per cycle enters in v_j)
- $\sum_{j=0}^{n} x_{ij}^k = 1 \; \forall i, k$ (only 1 edge per cycle comes out from v_j)
- …

Monitoring an Area by UAVs (8)

Provide a MILP formulation (cntd)

constrains (cntd):

- $\sum_{j=1}^{n} x_{0j}^{k} = 1$ *and* $\sum_{i=1}^{n} x_{i0}^{k} = 1\ \forall k = 1, \ldots, c$ (all c cycles go out from $v_0{}^s$ and enter in $v_0{}^t$)
- subtour elimination constraints
- Objective: minimize c [usually to avoid…]

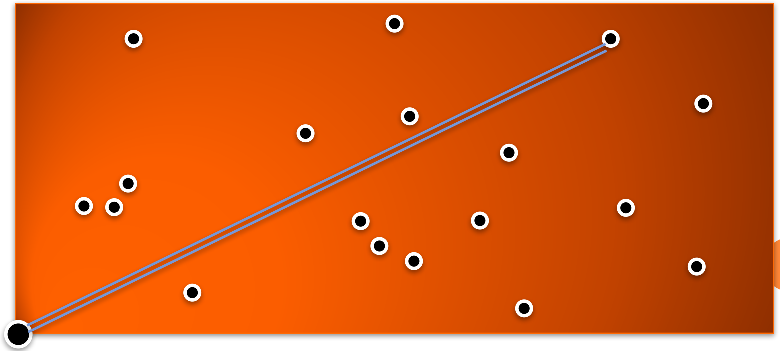
- NOTE: this formulation solves only the first two phases: the scheduling is missing…

Problem 3: provide a correct and complete formulation…

Monitoring an Area by UAVs (9)

Reduction of the dimension of the instance:

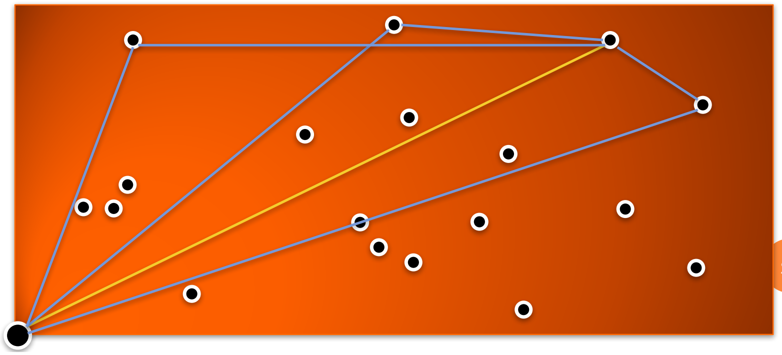
Property 1: if $\exists i$ s.t. $2\ w(v_0,v_i)+t_i=B \Rightarrow$ cycle v_0-v_i-v_0 is in every solution.



Monitoring an Area by UAVs (10)

Reduction of the dimension of the instance (cntd):

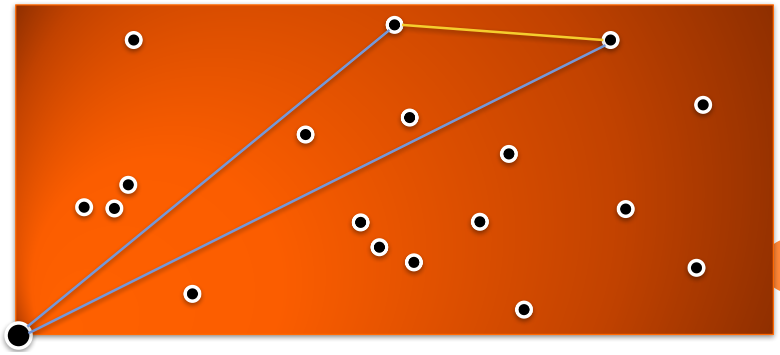
Property 2: if $\forall j$ it holds $w(v_0,v_i)+t_i+w(v_i,v_j)+t_j+w(v_j,v_0)>B \Rightarrow$ cycle v_0-v_i-v_0 is in every solution.



Monitoring an Area by UAVs (11)

Reduction of the dimension of the instance (cntd):

Property 3: if $\exists i,j$ s.t. $w(v_0,v_i)+t_i+w(v_i,v_j)+t_j+w(v_j,v_0)>B \Rightarrow$ edge (v_i,v_j) cannot enter in any solution.



MONITORING AN AREA BY UAVS (12)

Reduction of the dimension of the instance (cntd):

The main idea is that, before solving our problem on the given instance, we can reduce its dimension by forcing to be inside the solution the edges indicated by Properties 1 and 2, and to be outside the solution the edges indicated by Property 3.

Problem 4: given a general (e.g. random, real life, etc.) instance, how much can we expect to reduce its dimension?

OTHER OPEN PROBLEMS

- determining a tight approx ratio
- introducing cooperation
- better exploiting UAVs' capabilities
- …