MONITORING BY UAVS
I.E.
WHAT?
(SOME THESES PROPOSALS)

Prof. Tiziana Calamoneri

Network Algorithms

A.y. 2020/21

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

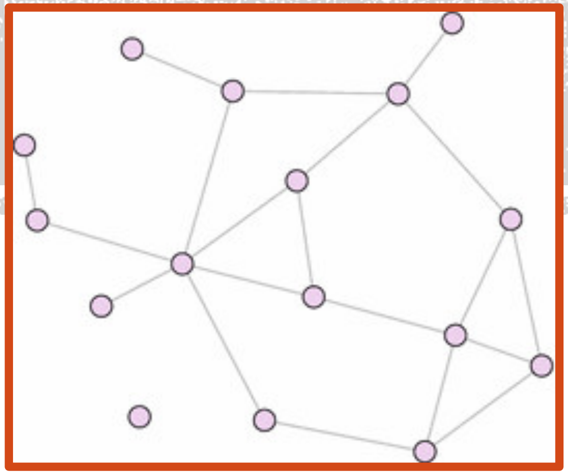
3

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 periodically go back to v_o in order to recharge its battery; this takes time R, typically 2.5-5 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

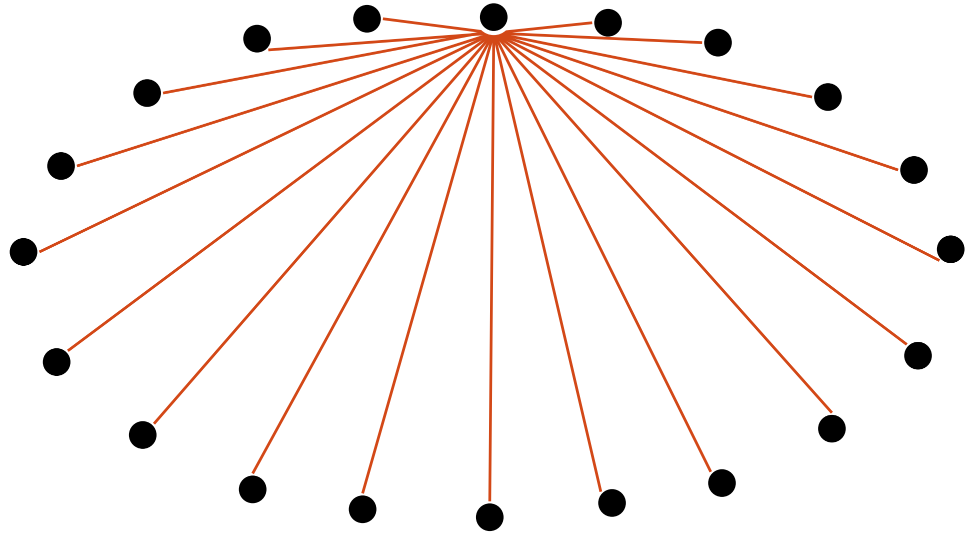


THE GRAPH MODEL (1)

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 think that it is possible to go from each node to every other node, so

there is an edge between each pair of nodes ⇒ K_{n+1}

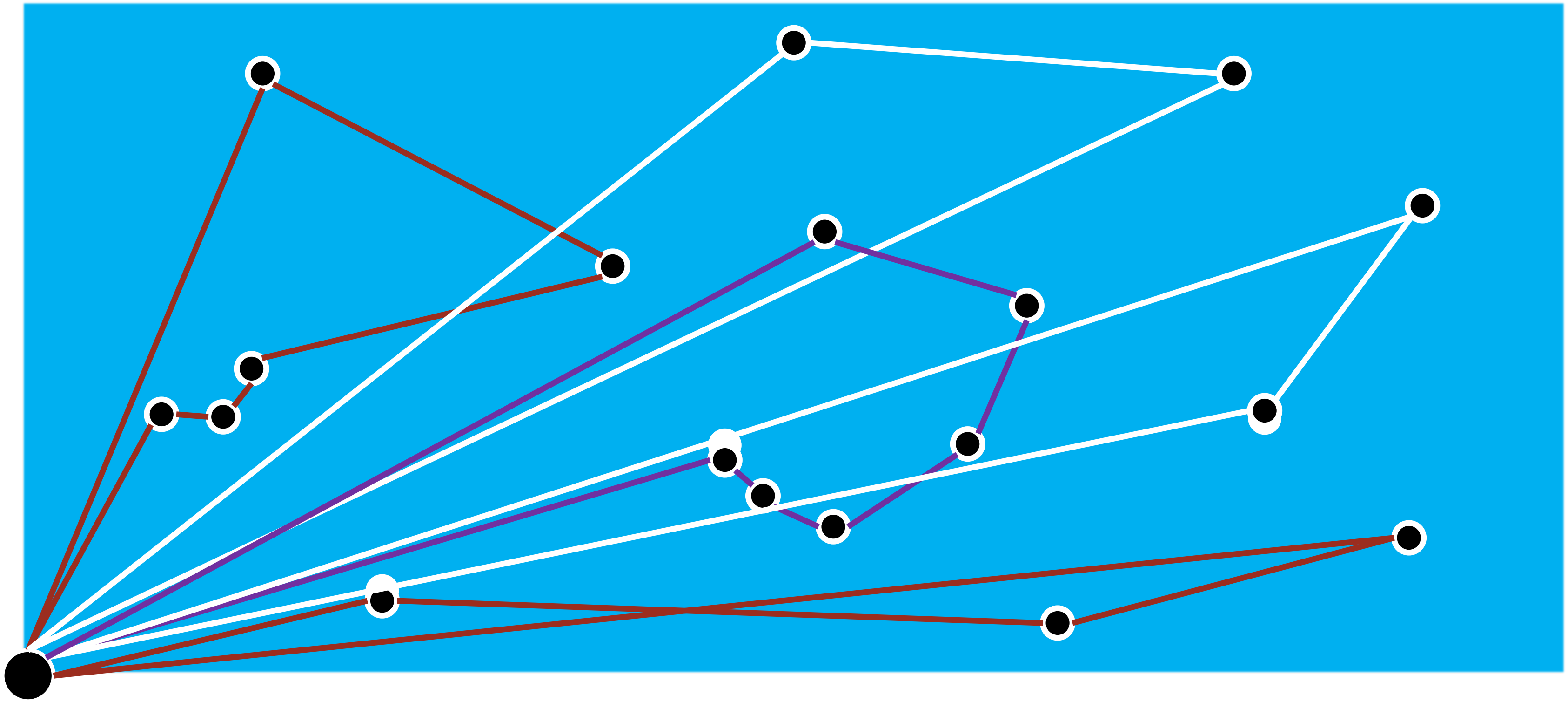
THE GRAPH MODEL (3)

- 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 in the "shortest time"?

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 (i.e. the total traversed distance) 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 custemers
- No repairtimes and it is not trivial to extend a solution by just adding them
- number 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 is usually a constraint on the number of visited customers per vehicle 

THE GRAPH MODEL (10)

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…

CONNECTION WITH RMCCP (1)

[C. & Tavernelli'19]

- A cycle cover $\mathscr{C} = \{C_1,\ldots,C_k\}$ for the site set V is a set of cycles s.t. each location of V belongs to at least one cycle.
- Given a fixed value $x \geq 0$, an x-bounded cycle cover is a cycle cover in which each cycle C has $cost(C) \leq x$.
- A rooted cycle C is a cycle where $v_0 \in C$. A rooted cycle cover is a cycle cover whose cycles are rooted cycles.
- The completion time of a (rooted) cycle cover $\mathscr{C}$ is

$$ct(\mathscr{C}) = max\ cost(C) \text{ on all } C \in \mathscr{C}.$$

CONNECTION WITH RMCCP (2)

- RMCCP (Minimum Bounded Rooted Cycle Cover Problem) requires to find, if it exists, a bounded rooted cycle cover of minimum cardinality.
- Definition. RMCCP:
  Input: $\langle G, v_0, d, x \rangle$ where $G = (V,E)$ is a graph, $v_0 \in V$ is called root, d is a distance defined on E and x is a positive number
  Output: an x-bounded rooted cycle cover of minimum cardinality, if it exists.
- RMCCP has been proved to be approximable first within $O(\log x)$ [Nagarajan & Ravi '12] and then within $O(\frac{\log x}{\log \log x})$ [Friggstad & Swamy '14].

CONNECTION WITH RMCCP (3)

RMCCP and our problem are tightly connected:

- Thm. If RMCCP can be approximated within α, our problem can be approximated within 5 α +1 (if the optimum solution has completion exceeding *b*).

On the other hand:

- Thm. If our problem can be approximated within γ, then MCRCCP can be approximated within 2 γ +1.

CONNECTION WITH RMCCP (4)

- In other words, our problem inherits the hardness of RMCCP. Notice that whether RMCCP admits a constant approximation algorithm or not is one of the major open problems in this area.
- Note. although RMCCP and our problem are so tightly connected, they are rather different, indeed the first one minimizes the number of cycles, while the second one the completion time.

CONNECTION WITH RMCCP (5)

All these results are true in general…

Special case:

- If $w(v_0,v_i) \leq xb$, where x is a fixed constant in the open interval $(0, 1/2)$, then our problem can be approximated within a constant.

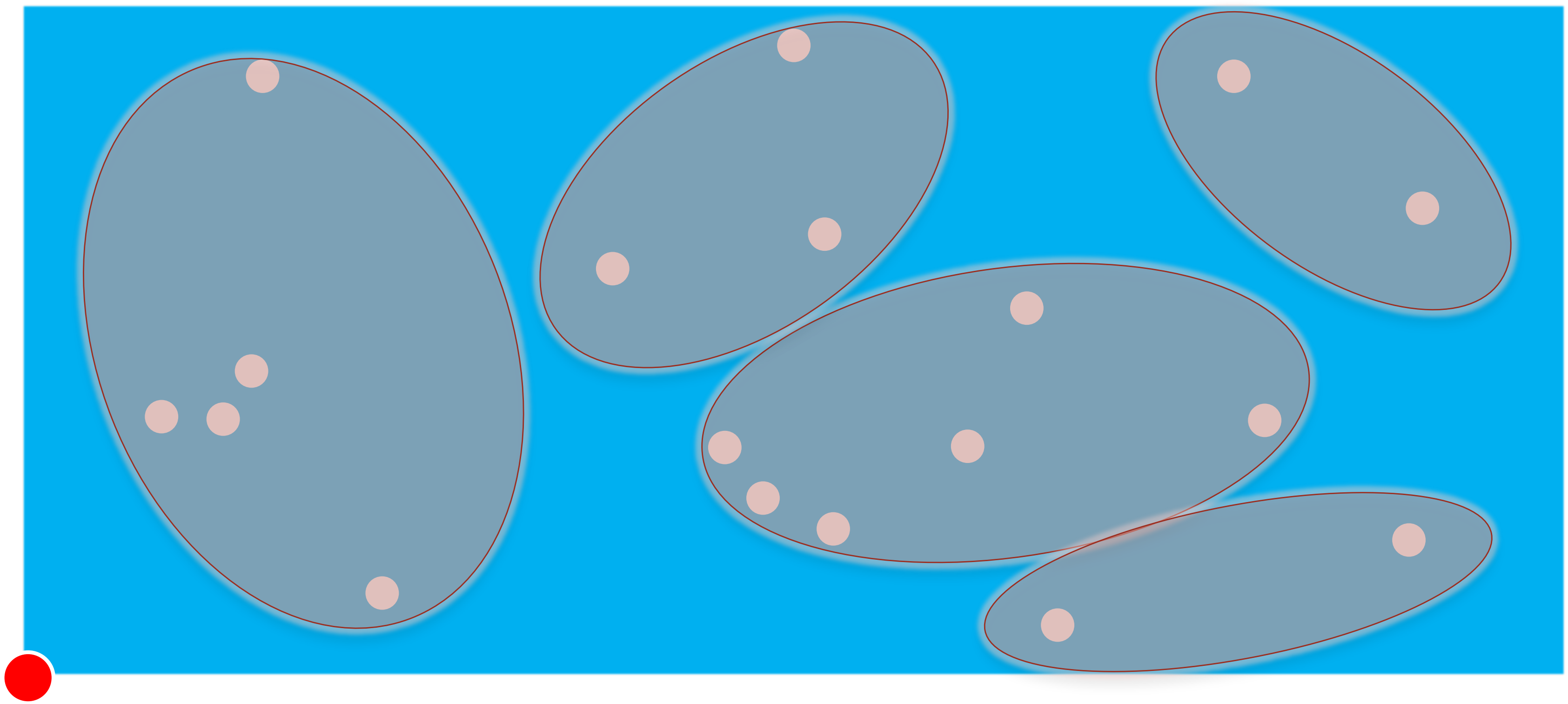
MONITORING AN AREA BY UAVS (1)

So, we cannot exploit other problems and we study it by itself, going in possible directions [C. & Dell'Orefice '18]:

1. due to its NP-hardness, heuristics based on three main pahses:
   - clustering/matroid theory (greedy)
   - approximating TSP
   - scheduling/bin packing
2. reduction of the dimension of the problem

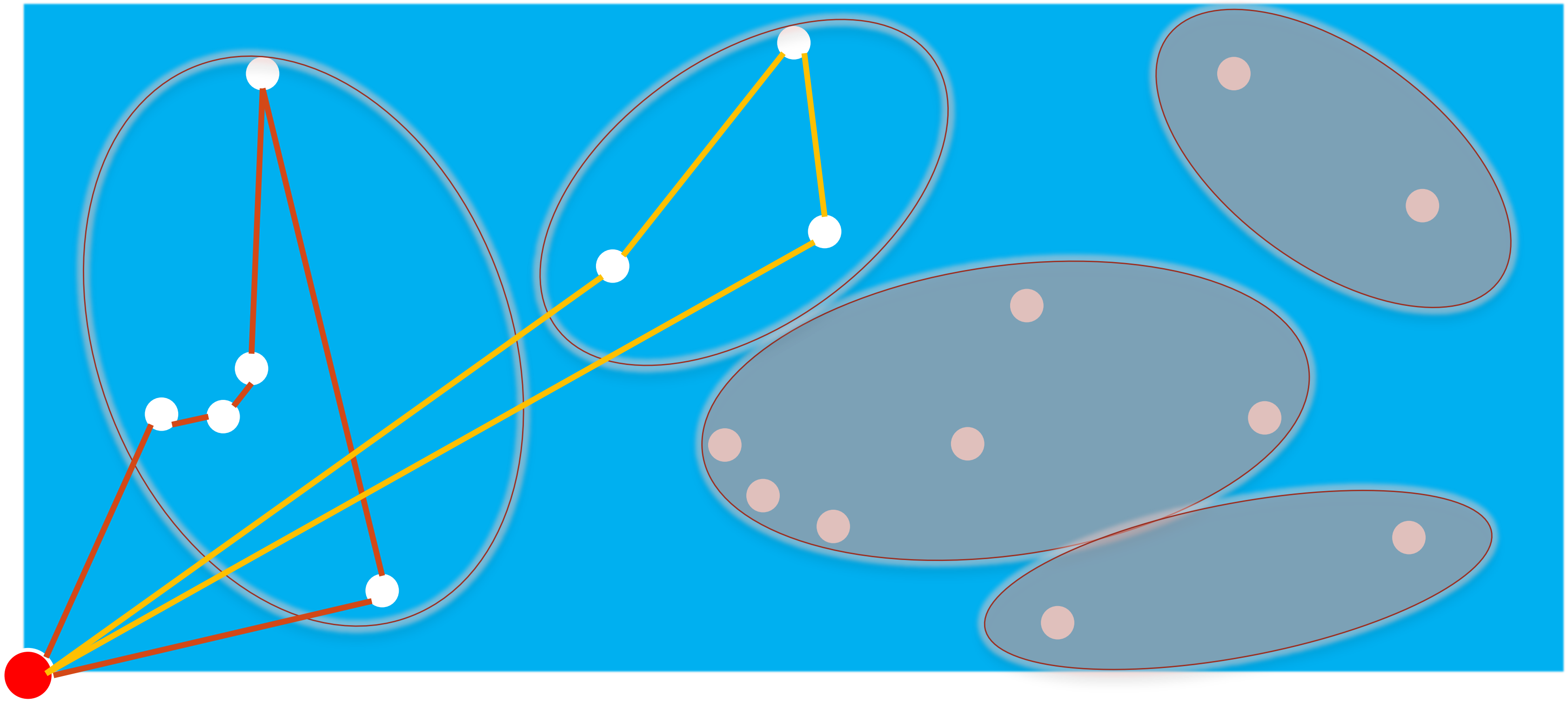
MONITORING AN AREA BY UAVS (2)

1. Heuristics based on three main pahses:
   - clustering: the idea is to partition the sites so that each set can be covered within battery *B.*



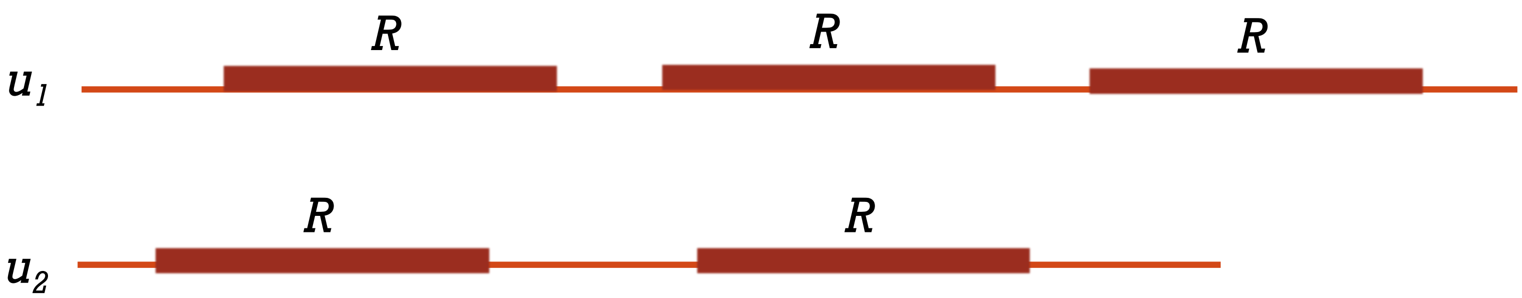
MONITORING AN AREA BY UAVS (3)

- 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 (4)

- scheduling: all cycles must be distributed to UAVs so to guarantee min completion time or min latency
- bin packing: when the order is not important



MONITORING AN AREA BY UAVS (5)

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 (6)

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…

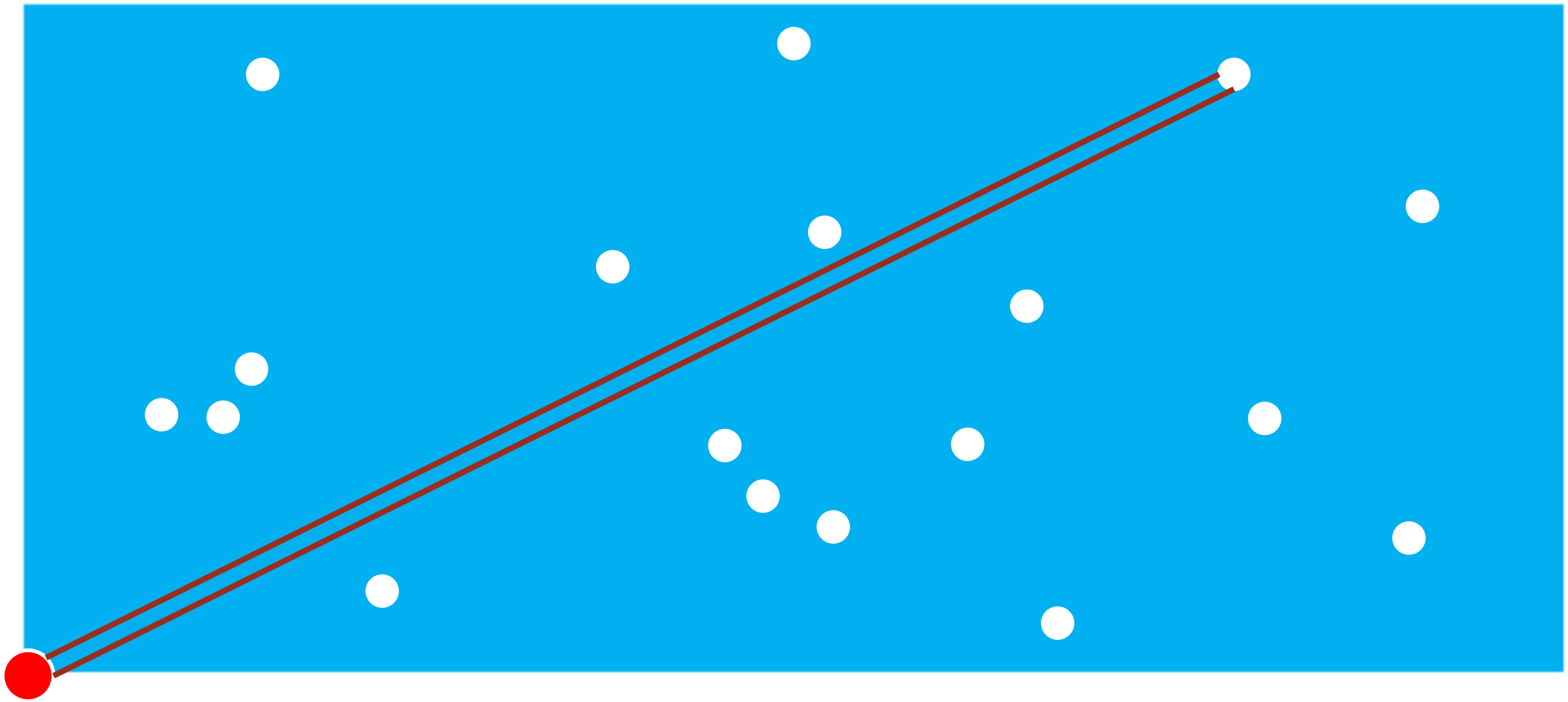
Problem 2: reduce the space of the cycles so that the approximation ratio does not increase too much

MONITORING AN AREA BY UAVS (10)

2. 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.

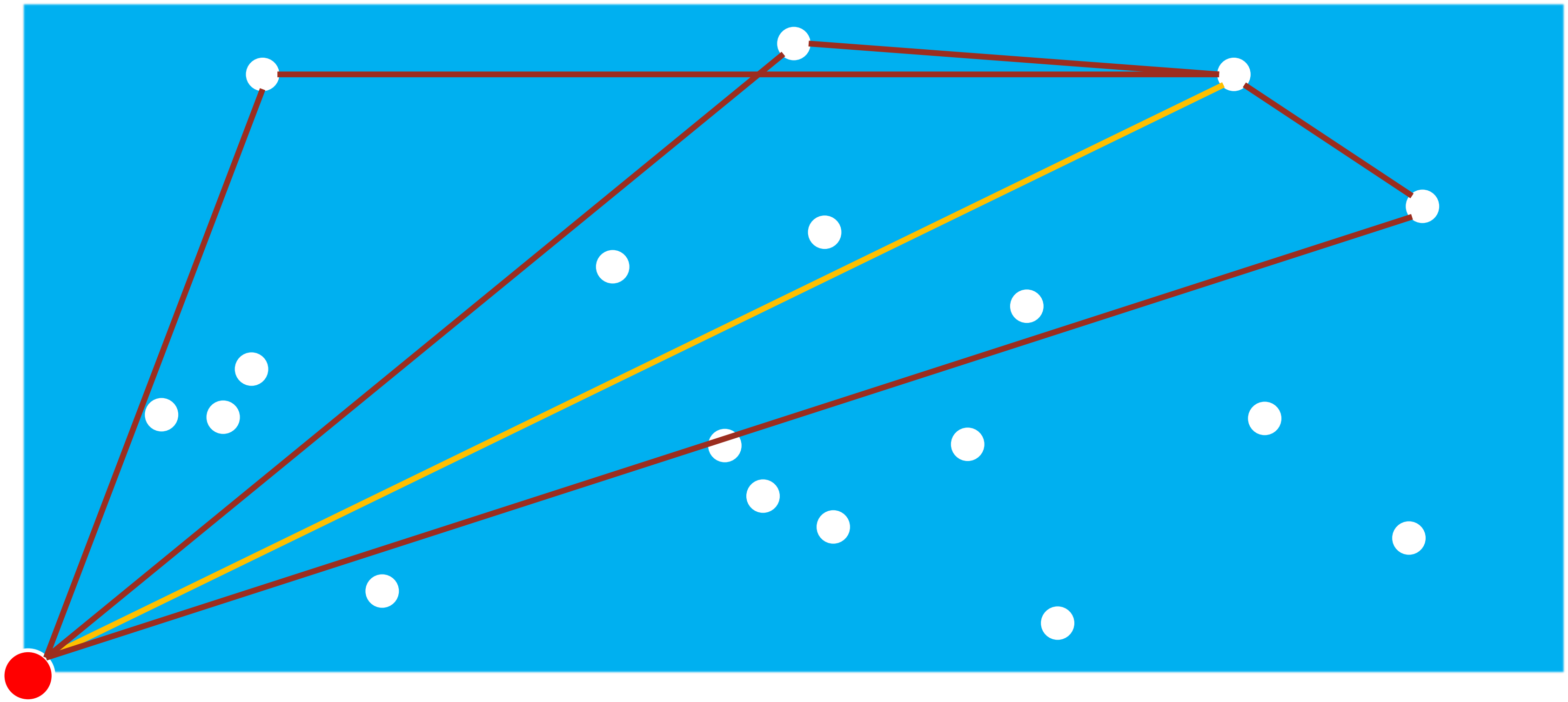


MONITORING AN AREA BY UAVS (11)

3. Reduction of the dimension of the instance (cntd):

Property 2: if $\forall i$ 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_j-v_0 is in every solution.

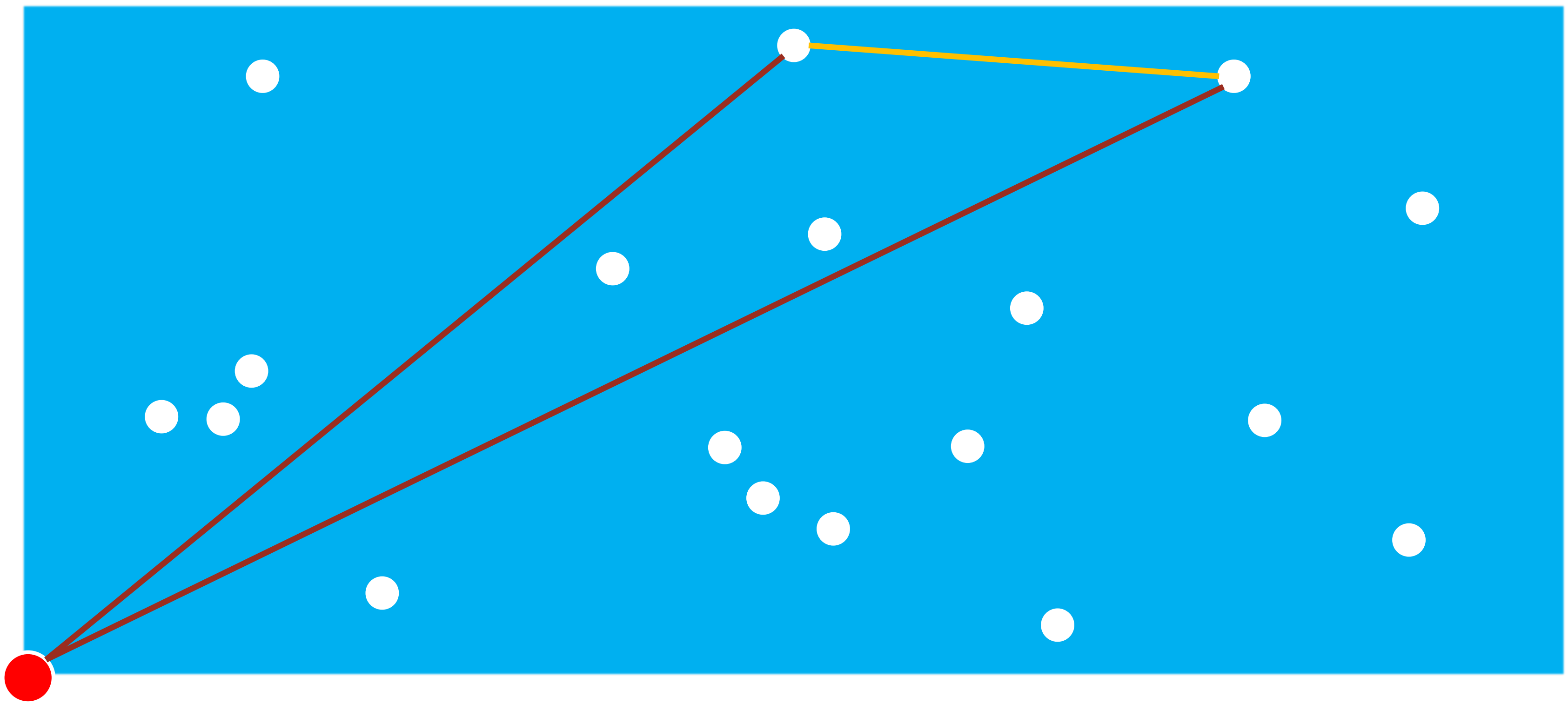


MONITORING AN AREA BY UAVS (12)

3. 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 (13)

3. 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?

OPEN PROBLEMS

- determining a tight approx ratio
- introducing cooperation
- introducing some “emergency criteria” able to dynamically change the UAVs’ behaviour (what if an injured person is detected? Shall we wait until the UAV is back?)
- …