TECHNISCHE UNIVERSITÄT DRESDEN

Simultaneous Placement and Scheduling of Sensors

Presenter: Peter Frohberg
Supervisor: Dr.-Ing. habil. Waltenegus Dargie
TU Dresden
Chair of Computer Networks

Dresden, June 16 , 2010

Outline

1. Introduction
2. Placement and Scheduling
3. ESPASS algorithm
4. Case studies

1. Introduction

- „Simultaneous Placement and Scheduling of Sensors“
  - Andreas Krause (Caltech) et. Al.
  - 2009

- Application examples
  - Traffic Control
  - Monitoring vulcano activities

Central questions

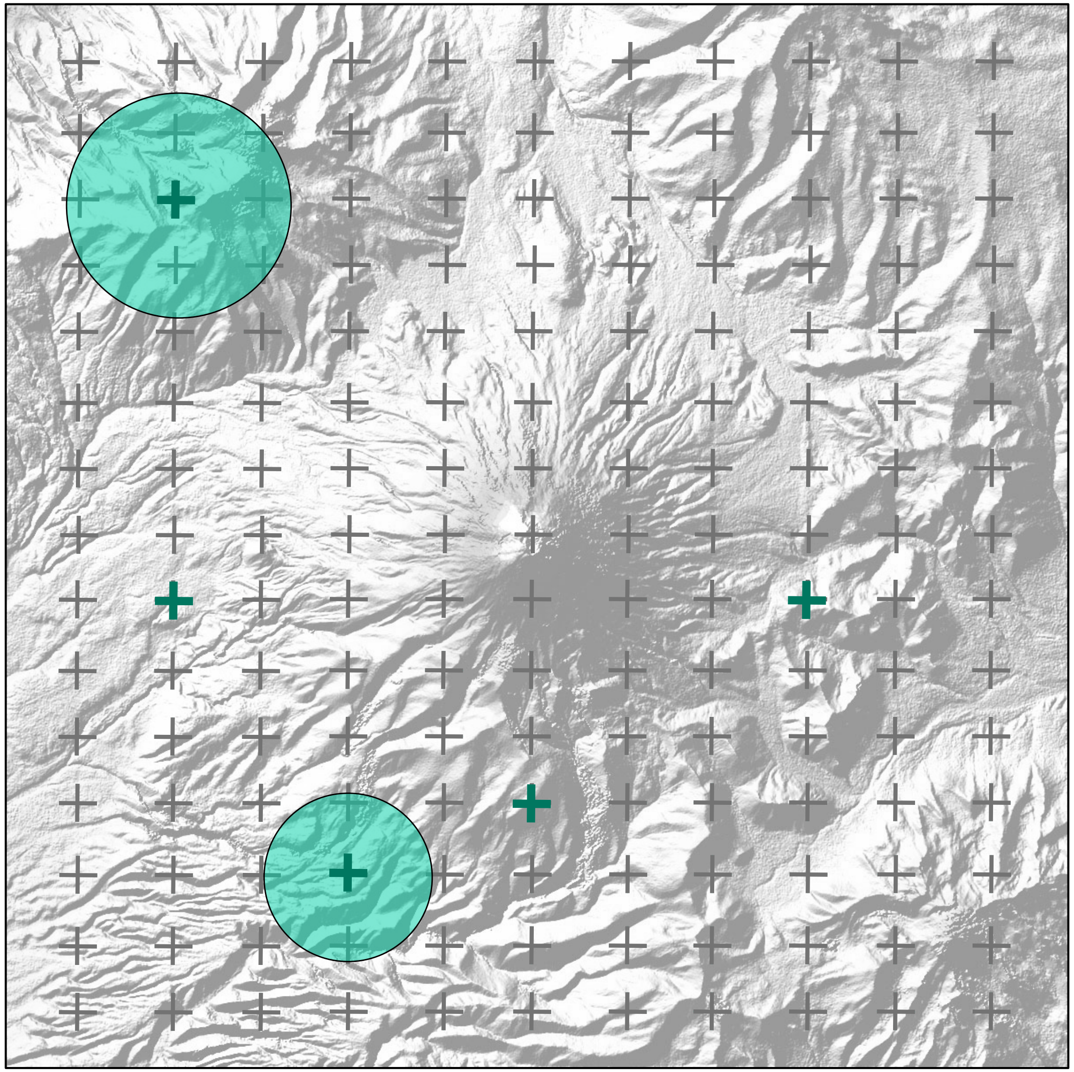
- **Where** should the sensors be **placed**?
- **When** should the sensors be **activated**?

→ Traditionally, sensor placement and scheduling have been considered separately from each other

TECHNISCHE UNIVERSITÄT DRESDEN

2.1 Placement of sensors

- Finit set of possible locations **V** = $\{s_1, s_2, \ldots, s_n\}$
- Goal: Select a small number of locations – **A** ⊆ **V** – for good measuring results.
- Sensing quality function **F(A)**
  - *F(A) = total area covered by sensing at locations A*
  - *or F(A) = prediction accuracy in traffic control*
  - *...*



+ Location $s \in V$

+ Location $s_i \in A$

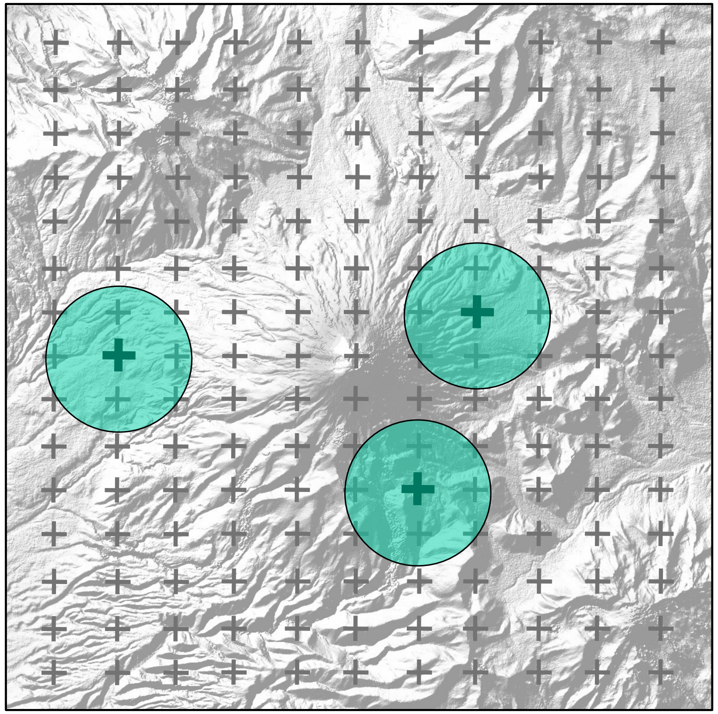
Area covered by a location $s_i \in A$

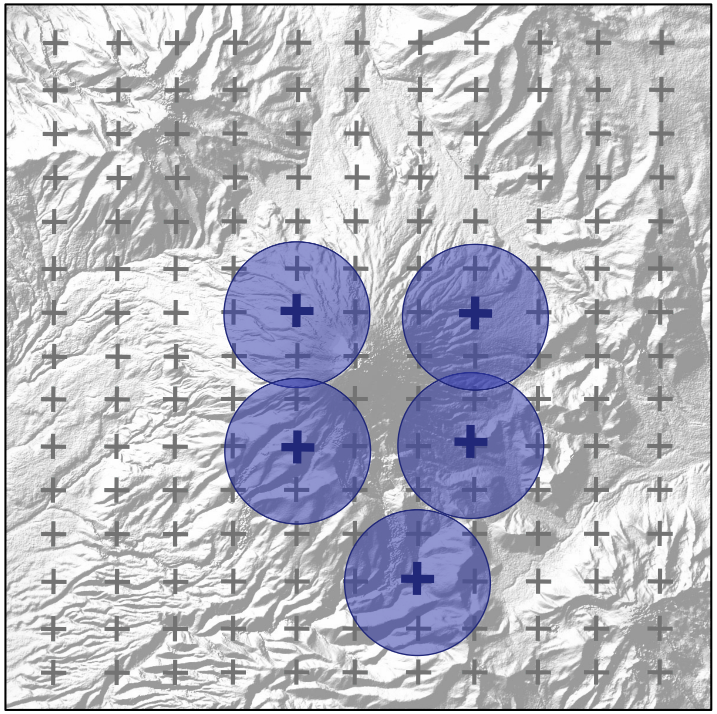
2.2 Scheduling

- Batterie lifetime can be increased by dividing sensors into groups and activate only one group per time.
- time slots **t** = {1, 2, ..., k}
- $s_i \in A_t$
- Calculating $\mathbf{F(A_t)}$ for every time slot

TECHNISCHE UNIVERSITÄT DRESDEN

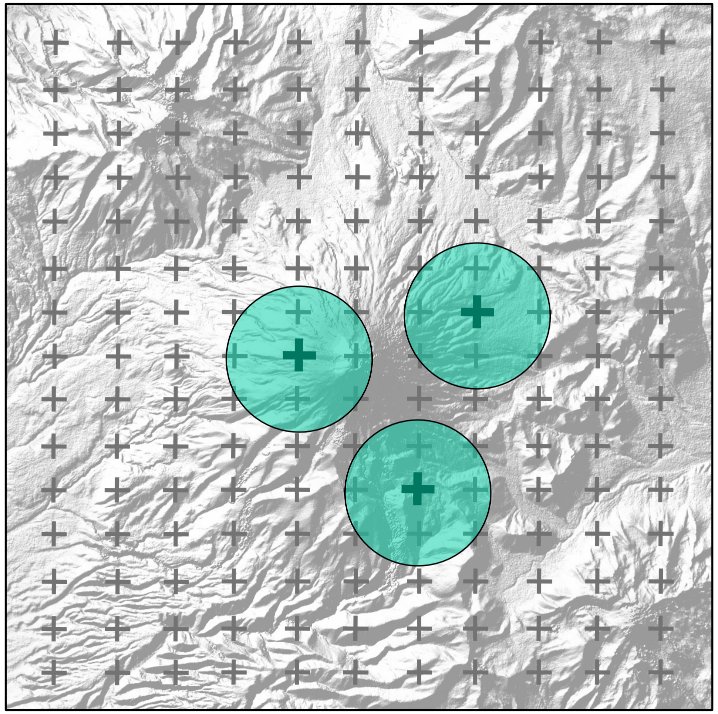
Average performance

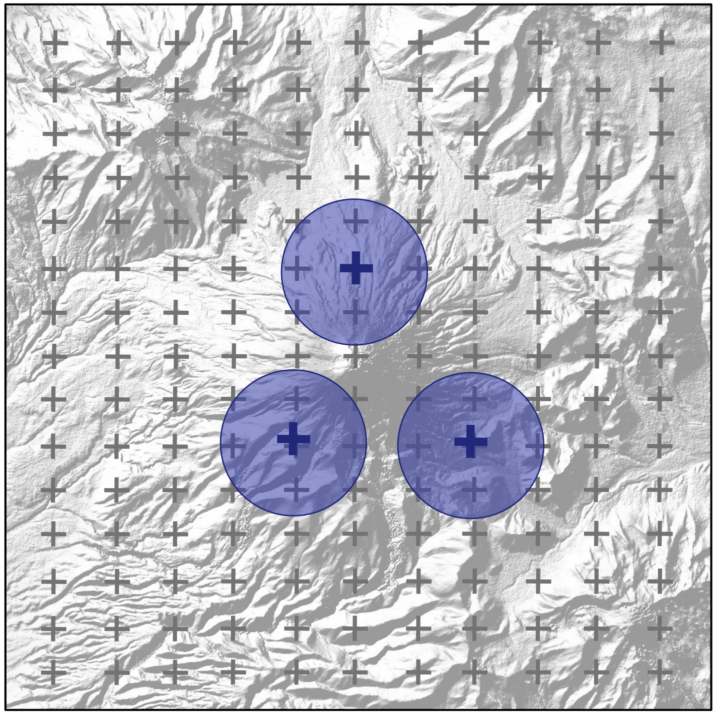




- Acceptable sensing quility in general using **average case**
- poor sensing quility in specific time slots possible

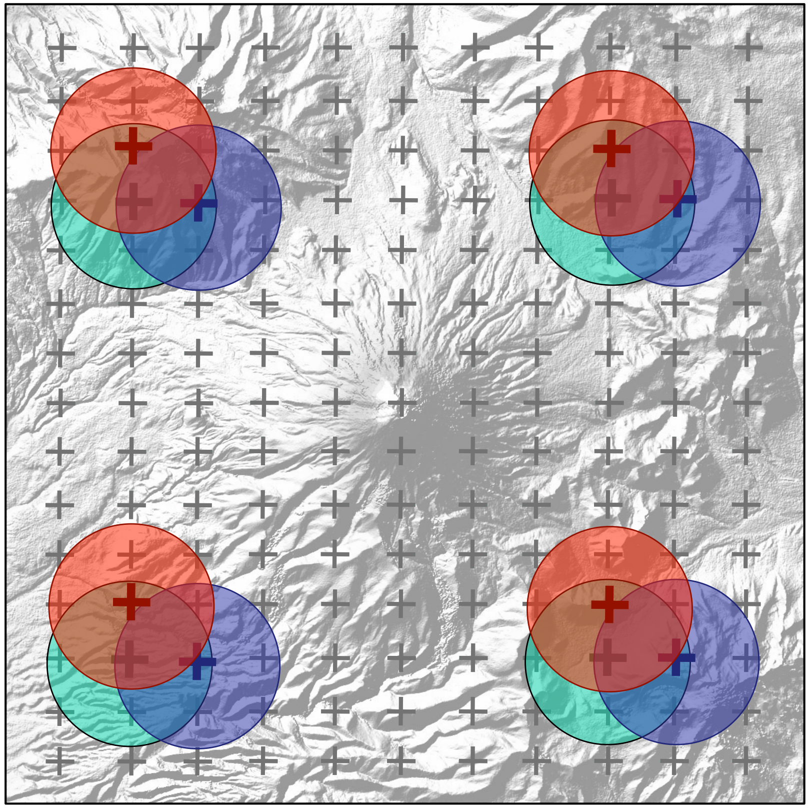
Well balanced solution





- **Minimum** sensing quility at every time slot garantied
- Performs generally better than average solution

3. Algorithm



- **Idea:** choose locations near to other sensors for every time slot.
- $F(A_1) \approx F(A_2) \approx \ldots \approx F(A_k)$

→ **Problem:**
switching from schedule mode to high density mode would not increase F(A).

3.1 GAPS

- **G**reedy **A**verage-case **P**lacement and **S**cheduling
- Near optimal solution for placement problem
- ***Short Description****:* For every sensor we have, GAPS picks a timeslot and a location which increases the total sensing quality at most.

TECHNISCHE UNIVERSITÄT DRESDEN

Algorithm GAPS (F, V, k, m)

Input:
- Quality sensing function *F*,
- Set *V* of possible locations,
- Number *k* of timeslots
- Number *m* of sensors we have

$A_t \leftarrow \varnothing$ for all t;

There is no sensor at any timeslot placed

for i = 1 to m do

For every sensor we do the following:

Do for every unused location and for every timslot the following:

foreach $s \in V \setminus (A_1 \cup \ldots \cup A_k)$, $1 \leq t \leq k$ do

$\delta_{t,s} \leftarrow F(A_t \cup \{s\}) - F(A_t)$;

How much increases the sensing function, when you add s to a timeslot t? Remember that value.

$(t^*, s^*) = \text{argmax}_{t,s}\ \delta_{t,s}$;

Choose the t,s combination with the highest δ value (highest F increase)...

$A_{t^*} \leftarrow A_{t^*} \cup \{s^*\}$;

... and put location s to set A_t.

- Example: GAPS(F, V, k=3, m=9), |V| = 50

- State before placing last sensor
  - A1 = {s11, s2, s37} F(A1) = 10
  - A2 = {s46, s43} F(A2) = 6
  - A3 = {s24, s15, s16} F(A3) = 8

- Finding best location and timeslot for last sensor:

F(A1 + s1) – F(A1) = 2	F(A1 + s36) – F(A1) = 4
F(A2 + s1) – F(A2) = 5	F(A2 + s36) – F(A2) = 2
F(A3 + s1) – F(A3) = 5	<u>F(A3 + s36) – F(A3) = 7</u>
...	...

- GAPS returns:
  - A1 = {s11, s2, s37} F(A1) = 10
  - A2 = {s46, s43} F(A2) = 6
  - A3 = {s24, s15, s16, 36} F(A3) = 15

GAPS

- O(k*m*n)

- Returns at least ½ of the optimal average-case score

- Performs well „on-average"
  → Krause interested in „balanced" perfomance

- Direct modification ***GBPS*** performs bad too.

3.2 ESPASS

- **E**fficient **S**imultaneous **P**lacement and **S**cheduling of **S**ensors

- Near optimal solution for scheduling problem
  (balanced schedule)

- Idea: every $F(A_t)$ must be above a minimum value.
  This value will be dynamically calculated

How did they get the idea? Speculations...

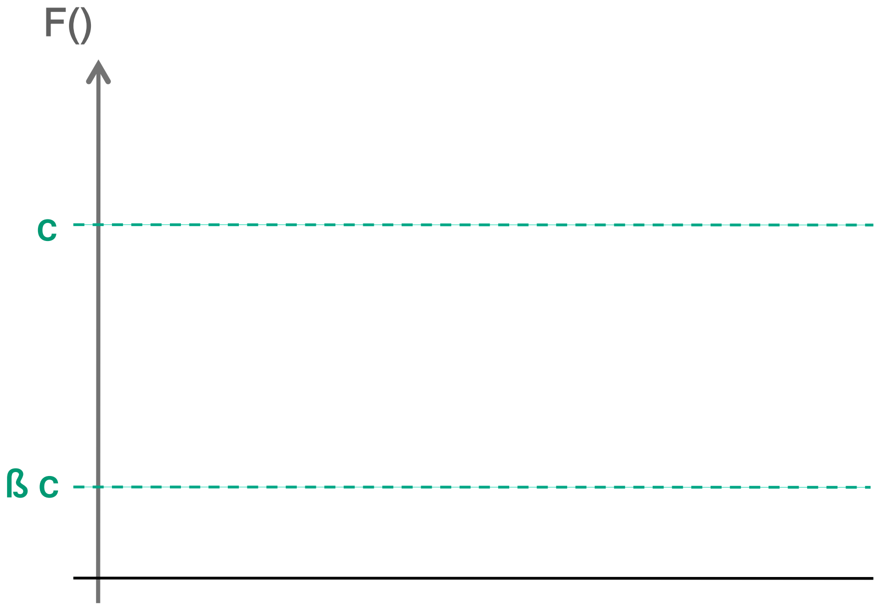
Introduction to ESPASS

- We have one „bucket" for each time slot
- In every bucket have to be n sensors, so that

$$\mathbf{F(A_t) > ßc}$$

→ **c** ... optimal value to balanced problem
→ **ß** ... fraction >0 of optimal sensing quality
→they prove that ß = ⅙

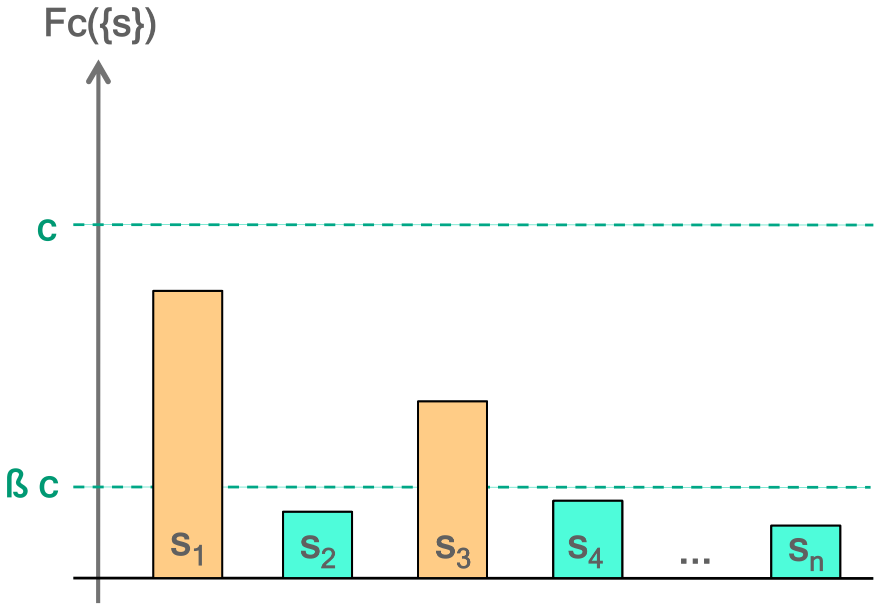
How ESPASS works



1. „Guess“ the optimal value c

TECHNISCHE UNIVERSITÄT DRESDEN

How eSPASS works



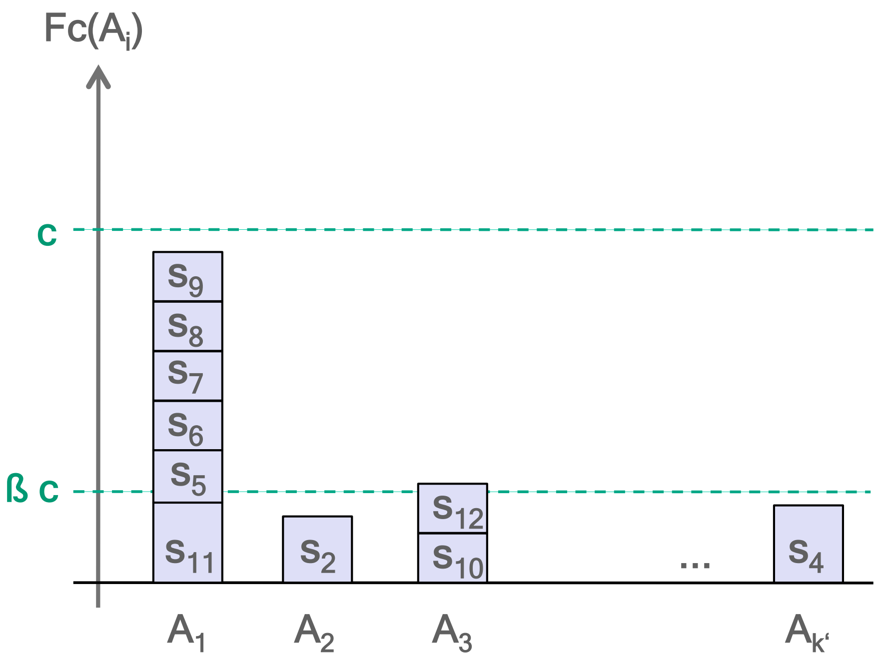
2. Call an element $s \in V$

   "big" if $Fc(\{s\}) \geq \beta c$

   "small" otherwise.

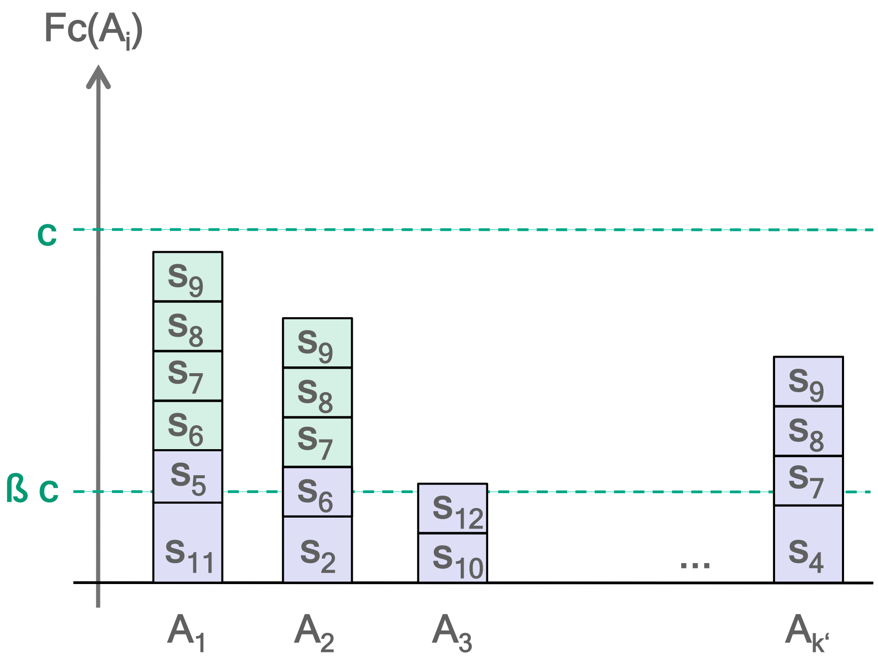
   Put each big element into a separate bucket. From now on, we ignore those satisfied buckets, and focus on the unsatisfied buckets.

How eSPASS works



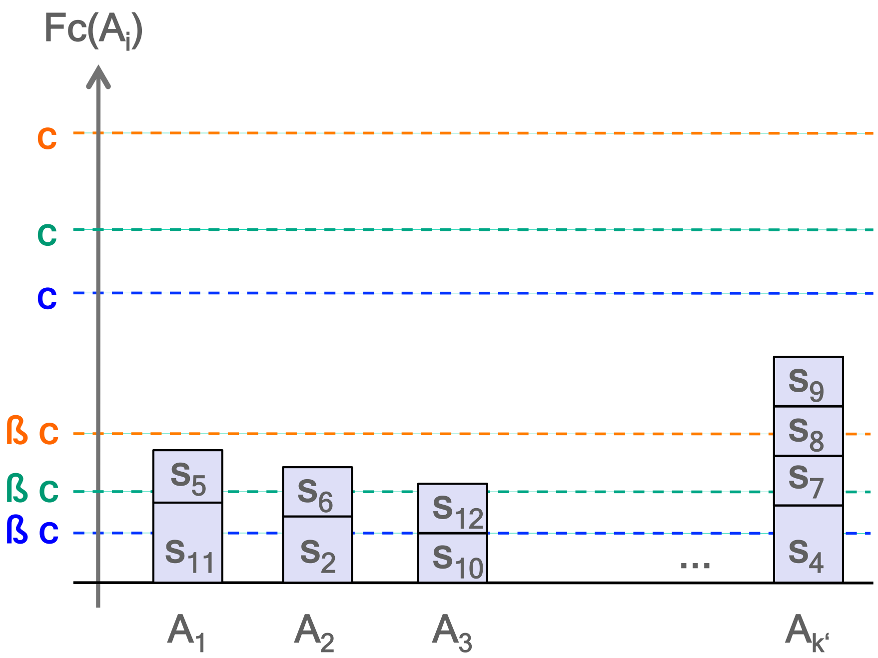
3. Run GAPS to optimize Fc and allocate the small elements to the unsatisfied buckets.

How ESPASS works



4. Pick a "satisfied" bucket A_t that contains sufficiently many elements and reallocate enough elements to an "unsatisfied" bucket to make it satisfied. Repeat this step until no more buckets are unsatisfied or no more reallocation is possible.

How ESPASS works



5. If all buckets are satisfied, return to step 1 with a more optimistic (higher) "guess" for c. If at least one bucket remains unsatisfied, return to step 1 with a more pessimistic (lower) guess for c.

TECHNISCHE UNIVERSITÄT DRESDEN

Algorithm ESPASS $(F, V, k, m, \varepsilon)$ — Tolerance parameter

$c_{min} \leftarrow 0;\ c_{max} \leftarrow F(V);\ \beta \leftarrow 1/6;$ — Declaration: F(V) would be the maximum value that would be possible

while $c_{max} - c_{min} \geq \varepsilon$ **do**

$c \leftarrow (c_{max} + c_{min}) / 2;$ — Dynamical calculation of c

$B \leftarrow \{s \in V : F_c(\{s\}) \geq c\beta\};$ — Get the „big“ elements

$k' \leftarrow k;$

foreach $s \in B$ **do** — Every „big“ element get it‘s own bucket

$A_{k'} \leftarrow \{s\};\ k' \leftarrow k' - 1;$

if $k'=0$ **then** $c_{min} \leftarrow c;$ — In case we have only big elements, we can increase c (indirectly increasing cmin and start once again)

$A_{best} \leftarrow (A_1,\dots,A_k);$

Continue with while loop;

$V' \leftarrow V \setminus B;\ m' \leftarrow m - |B|;$ — We now concentrate only on the possible locations left and run the GAPS algorithm place the sensors

$A_{1:k'} \leftarrow \mathrm{GAPS}(F_c, V', k', m');$

if $\Sigma_t F(A_t) < k'c/2$ **then** $c_{max} \leftarrow c;$ **continue**; — If the sensing quaility over all time slots is not good enough we start with a lower c.

else

while $\exists i,j: F_c(A_j) \leq \beta c,\ F_c(A_i) \geq 3\beta c$ **do**

foreach $s \in A_i$ **do**

$A_j \leftarrow A_j \cup \{s\};\ A_i \leftarrow A_i \setminus \{s\};$

if $F_c(A_j) \geq \beta c$ **then break**;

$c_{min} \leftarrow c;\ A_{best} \leftarrow (A_1,\dots,A_k);$

Here we relocate elements in case there are buckets with sufficiently many alements

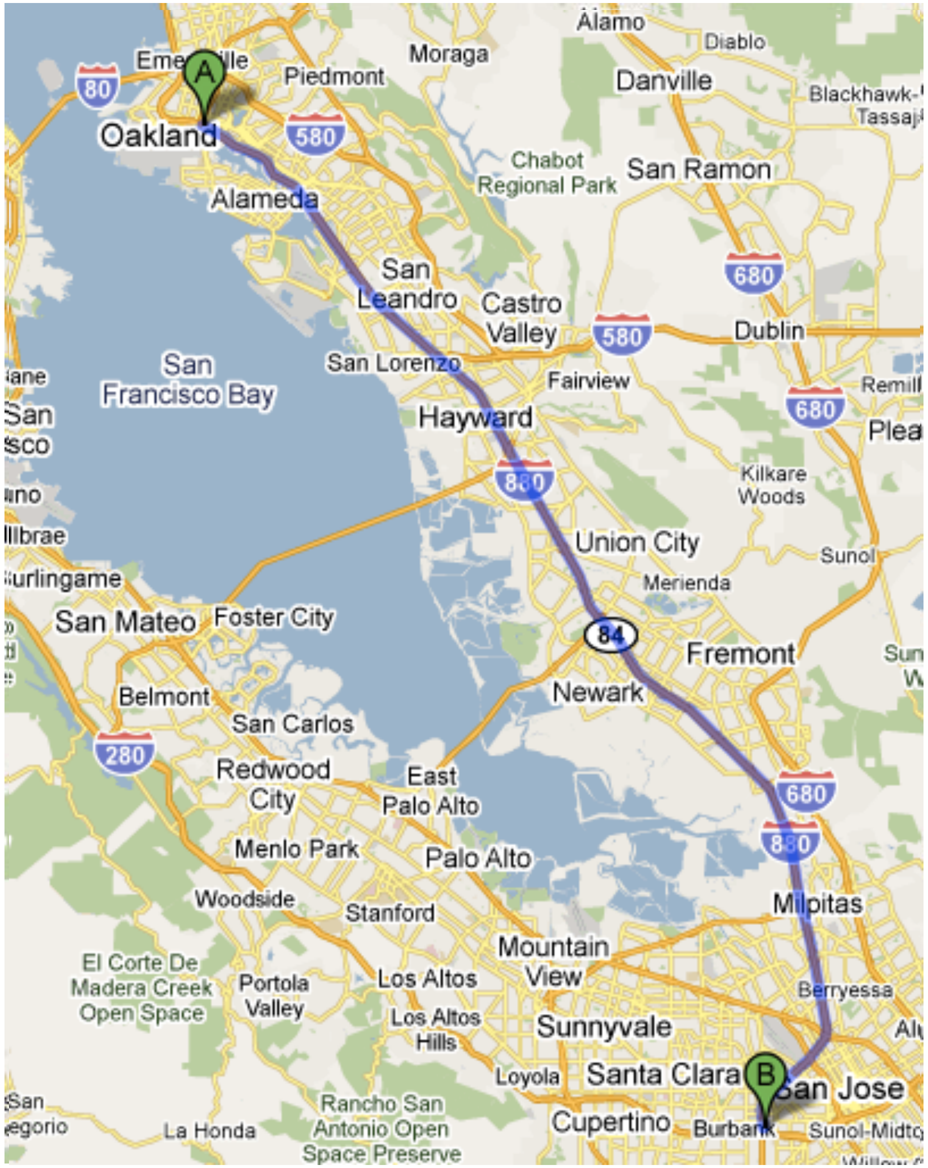
4. Case Studies

4.1. Highway Monitoring

4.1. Highway Monitoring – I-880

- Deployment strategy for replacing 10000 old sensors in California

- Interstate Highway 880 South
  - California
  - 35 miles
  - 3-5 lanes
  - 88 measurement sites, every 2 miles
  - 357 sensors
  - Speed information from lanes, between 8-11 AM, excluding weekends and holidays



ESPASS (F, V, k, m, ε)

- F = prediction accuracy (0…100)
- |V| = 100 possible locations
- k = 3 timeslots
- m= 20 sensors
- ε = 2 (termination condition)

1. Guess „c“
   - $c_{min} = 0$
   - $c_{max} = F(V) = 100$
   - **c** $= (c_{min} + c_{max})/ 2$ **= 50**
   - **ßc** = 1/6 * 50 **= 8,33**

2. Calculate F() for every possible location s_n.

F(s1) = 5	F(s21) = 6	F(s85) = 3
F(s2) = 7	F(s22) = 3	F(s86) = 7
F(s3) = 1	F(s23) = 7	F(s87) = 8
F(s4) = 0	F(s24) = 8	F(s88) = 8
F(s5) = 2	...	...
F(s6) = 4	F(s47) = 10	F(s97) = 1
F(s7) = 1	...	F(s98) = 2
F(s8) = 2	F(s56) = 2	F(s99) = 5
F(s9) = 7	F(s57) = 1	F(s100)= 1
...	...	

- One *big bucket* (one timeslot reserved) → k�much = 2
- One sensor at one location placed → |V‵| = 99, m‵=19
- A1 = {s47}
- A2 = A3 = ∅

3. Run GAPS (F,V,k,m)

- F = prediction accuracy (0...100)
- V = V‵ (|V‵| = 99)
- k = k‵ = 2
- m = m‵ = 19

3. Result GAPS

- A2 = {s1, s3, s5, s6, s7, s21, s22, s56, s85, s87, s100}
  F(A2) = 16 *(>ßc → satisfied)*
- A3 = {s2, s4, s8, s9, s57, s86, s87, s99}
  F(A3) = 11 *(>ßc → satisfied)*

➢ A2, A3 are satisfied → we don't need to relocate any sensor (skipping step 4)
➢ All timeslots are satisfied (perform better than ßc)

5. Find a new „c"

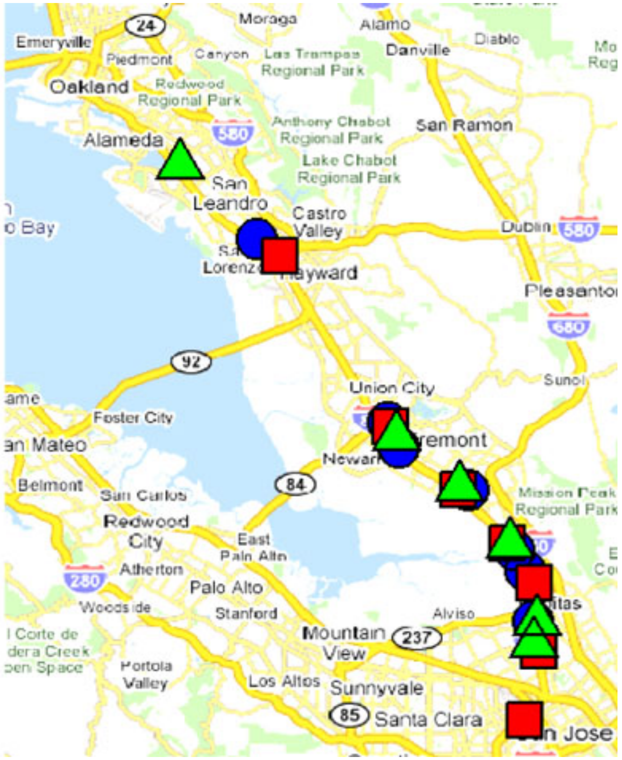
- c_{min} = c

➢ Start again

- **c** = (c_{min} + c_{max}) / 2 **= 75**
- **ßc** = 1/6 * 75 **= 12.5)**

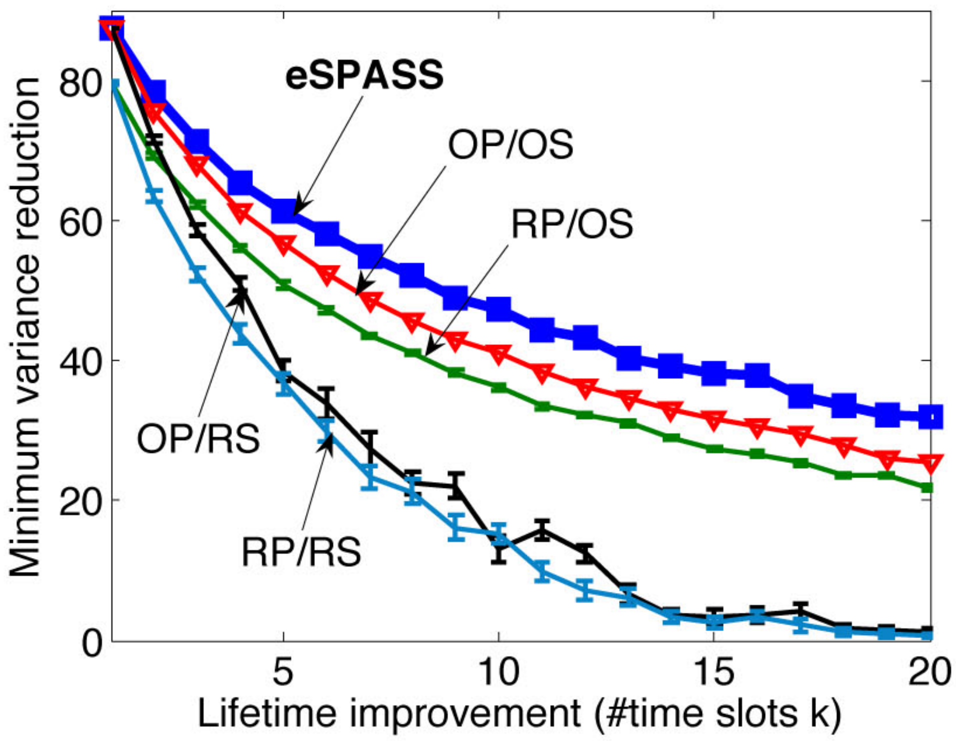
4.1. Highway Monitoring – I-880

stage-wise



simultaneous

4.1. Highway Monitoring – I-880



TECHNISCHE UNIVERSITÄT DRESDEN

4.2. Contamination detection

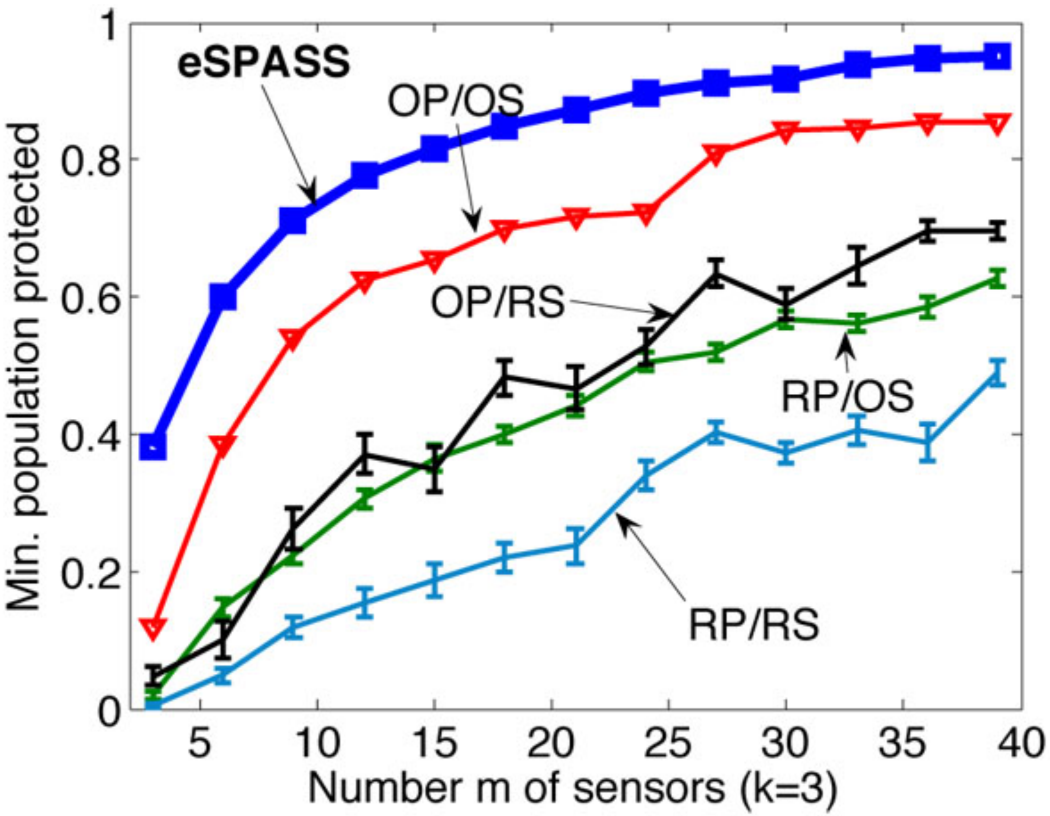
- Topic: Water Distribution Network
- Accidential or maliceaus intrusions can cause contaminents to spread over the network
- 2006 – „Battle of the Water Sensor Networks“
  - Best sensor placement for real metropolitan water distribution network
  - 12527 possible locations
  - Intrusion scenarios were defined

4.2. Contamination detection

- Goal: minimize impact measures, such as the expected population affected

- Type of Sensor: YSI 6600 sonde
  - 15 water parameters
  - Sends data 75 days long every 15 minutes

ESPASS at Battle of the Water Sensor Networks



Conclusion

- sensor placement and scheduling is important for the lifetime of a system
  → has been traditionally considered separately from each other.

- ESPASS algorithm optimizes simultaneously placement and scheduling
  → Balanced schedules
  → Network lifetime improvement between 33% and 100%

References

- A.Krause, R.Rajagopal, A.Gupta, and C.Guestrin, Simultaneous placement and scheduling of sensors, Information Processing In Sensor Networks, 181-192, 2009,
- A.Krause, R.Rajagopal, A.Gupta, and C.Guestrin, Simultaneous placement and scheduling of sensors, Tech. ReportCMU-ML-08-114, CMU, 2008.http://pems.eecs.berkeley.edu/
- http://maps.google.de
- http://earthobservatory.nasa.gov/IOTD/view.php?id=1816
- http://www.anhydre.eu/ysi/YSI%206600sondeV2.jpg
- http://www.flickr.com/photos/callmeack/3145284644
- http://www.flickr.com/photos/pinkcotton/4163902022/
- http://www.flickr.com/photos/kristofabrath/4595832841
- http://www.flickr.com/photos/artbandito/3627991896/
- http://www.flickr.com/photos/chrisjan99/1866889122