Online correlated orienteering on continuous surfaces

A problem in sea exploration

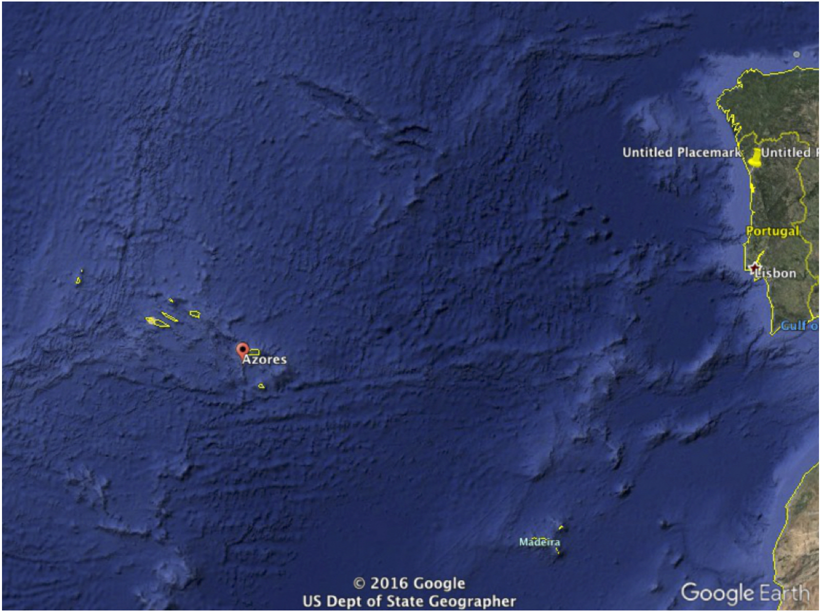
João Pedro Pedroso

INESCTEC and Faculty of Sciences, University of Porto[1]

ISCO, Marrakesh, April 2018

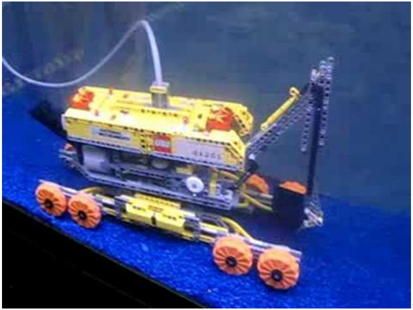
[1] Work partially supported by project "Coral - Sustainable Ocean Exploitation: Tools and Sensors/NORTE-01- 0145-FEDER-000036", financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

The problem (#1)



The problem (#2)

- Portugal: large area in the Atlantic
- Future: maybe exploit some of the resources in the seafloor
- Problem: seafloor contents unknown
- Need to fetch information about seafloor contents
  - send underwater robots
  - collect samples



The problem (#3)

- How to schedule a sea recognition trip?
- What is known:
  - maximum time the ship can spend on the trip
  - an empiric assessment about possibly interesting places
  - estimation for the time it takes to collect a sample (*probe*)
  - estimation of the ship's speed
    (though it depends on weather conditions)

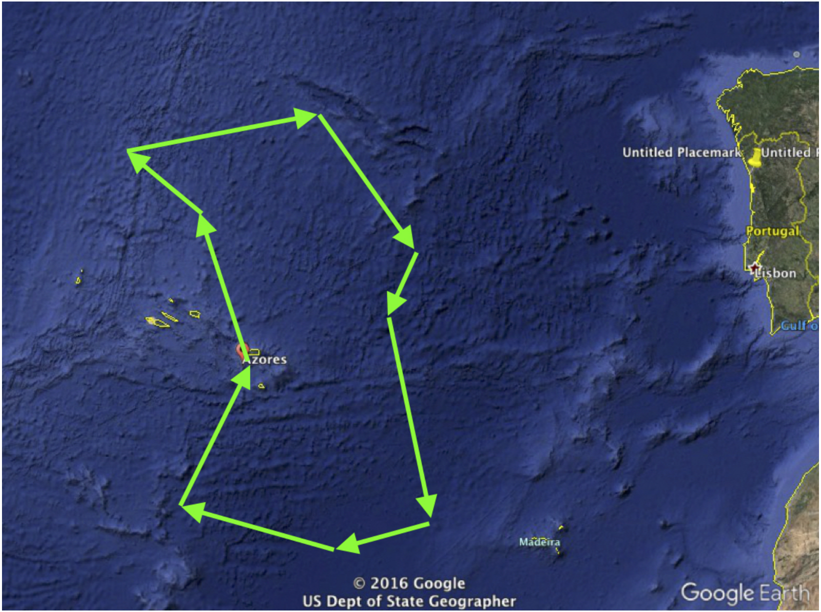
The problem (#3)

- How to schedule a sea recognition trip?
- What is known:
  - maximum time the ship can spend on the trip
  - an empiric assessment about possibly interesting places
  - estimation for the time it takes to collect a sample (*probe*)
  - estimation of the ship's speed
    (though it depends on weather conditions)
- How to define the problem mathematically?
  - ⚠when we collect a sample, the shape of the information landscape changes
  - → online problem

Background

- First relevant related problem: orienteering
  - visit subset of vertices
  - collect "prize" on visited vertices
  - limit on total trip time
- But our variant is very different of standard version
  - no clear objective:
    - "*maximize information*" about seafloor contents?
    - ...
  - no underlying graph:
    - select discrete set of points in continuous surface
    - virtually any point in the sea visitable from any other point

A possible solution



Background

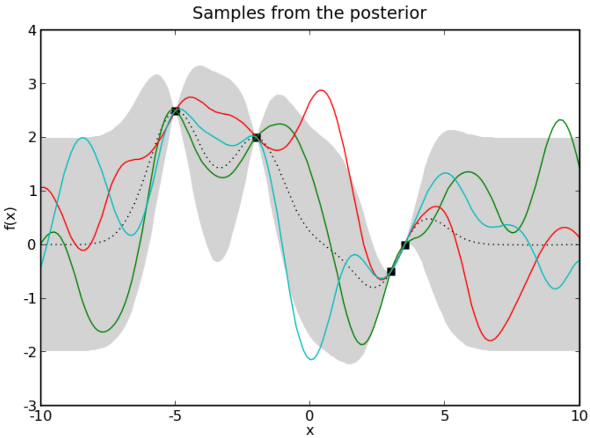
- Second relevant problem: *attractiveness* estimation
  - *how interesting is it to explore/probe a given point?*
- Defines our objective on orienteering

Background

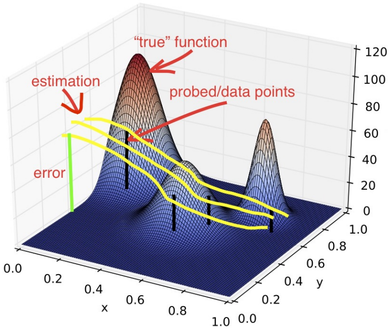
- Second relevant problem: *attractiveness* estimation
  - *how interesting is it to explore/probe a given point?*
- Defines our objective on orienteering
- Related problem: kriging
  - 1960's: Danie G. Krige, method for choosing mines
  - data: position of currently known mines
  - output: next position to probe
  - kind of interpolation

Background

- Second relevant problem: our choice: gaussian processes
  - "modern" version of kriging
  - works in function space
  - uses data to restrict to "*likely*" functions
  - gives information about expectation and standard variance



Visualization

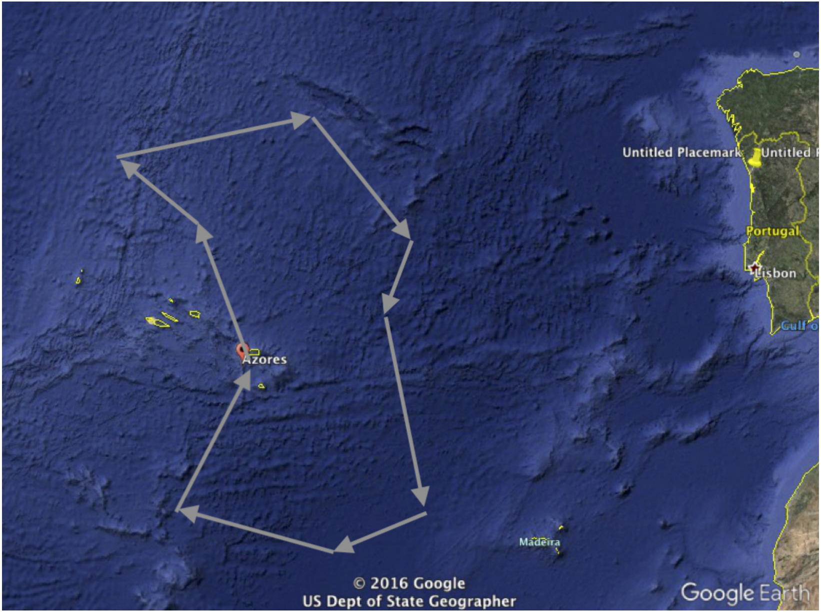


- Gaussian processes
  - Data: currently known seafloor contents at given points
  - Assign a numeric value to the contents at any other point
  - Also provide a value for the variance
  - However:
    - values on different points are correlated
- Our problem: selecting new data for the GP

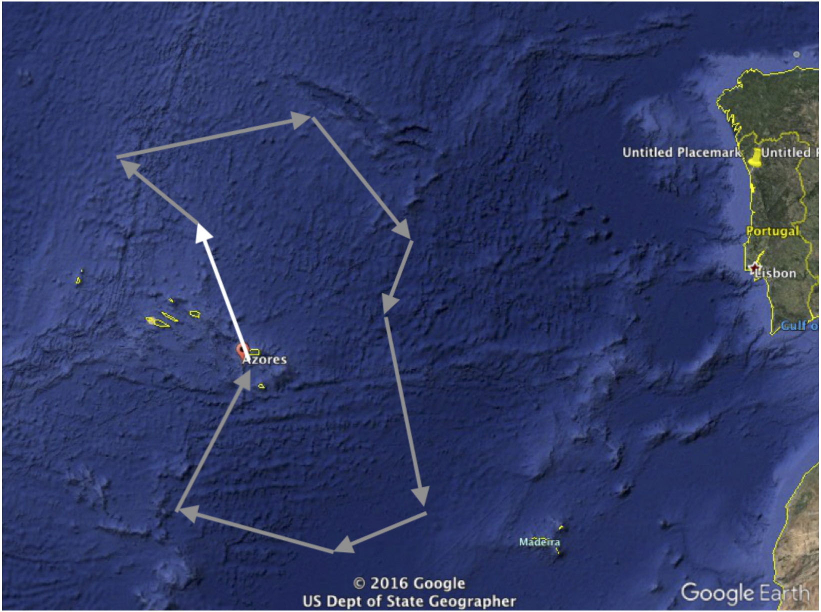
Online problem

- Probing → data set is being complemented dynamically
- Newly collected data influences the GP landscape
- Expedition plan may have to be updated in real time

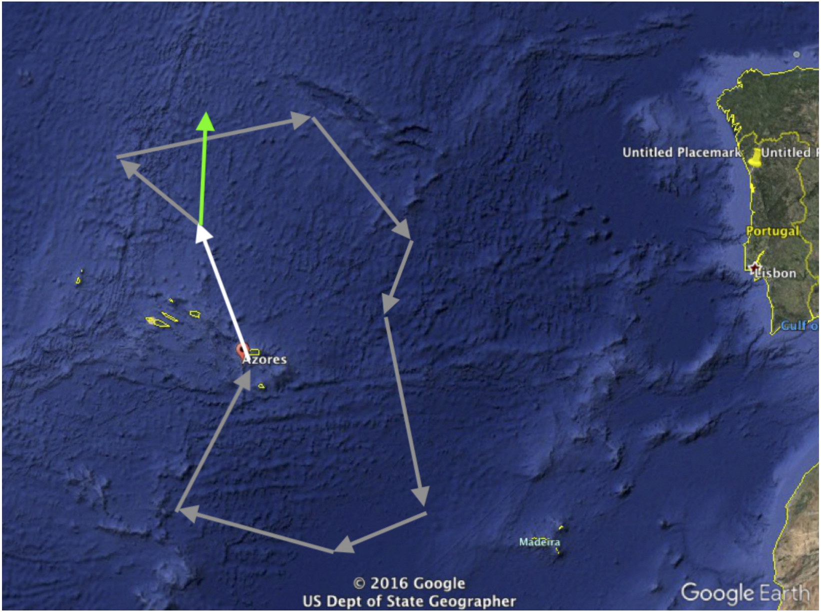
Initial solution



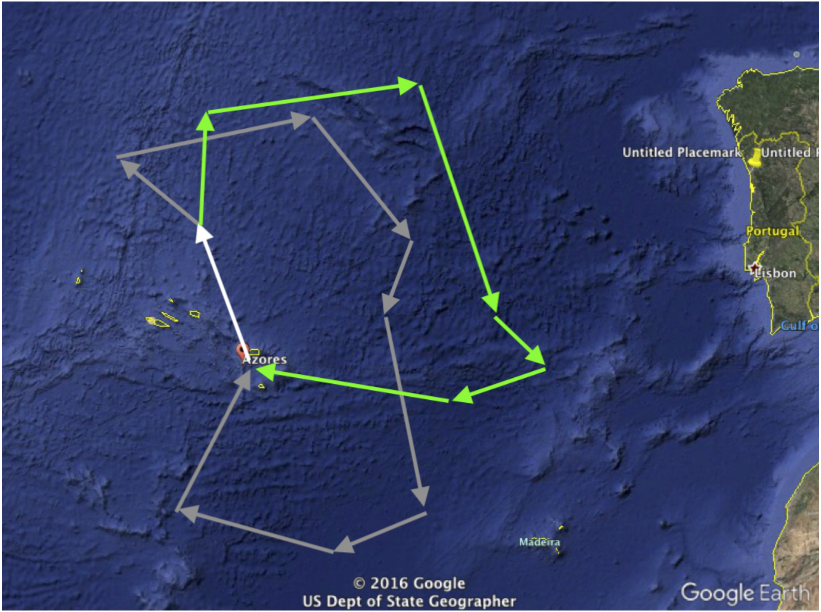
Update


Untitled Placemark
Untitled
Portugal
Lisbon
Gulf o
Azores
Madeira
© 2016 Google
US Dept of State Geographer
Google Earth

Update


Untitled Placemark
Portugal
Lisbon
Azores
Madeira
© 2016 Google
US Dept of State Geographer
Google Earth

Update


Untitled Placemark
Untitled P
Portugal
Lisbon
Gulf o
Azores
Madeira
© 2016 Google
US Dept of State Geographer
Google Earth

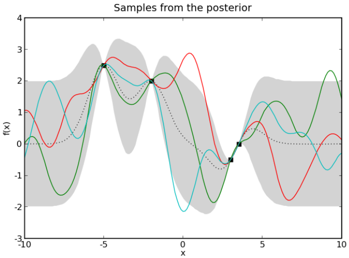
Our view: the global problem

1. Assessment: use currently available data
2. Orienteering:
   - select points for probing
   - generate new data
3. Estimation: using all data, predict values at new points

Our view: orienteering

- Orienteering trip: select a set of points to visit
  - these points will be probed for seafloor contents
  - after actual probing, we can reassess estimation allover the surface
- Objective:
  - at the end of the trip, have a best possible estimation allover the surface

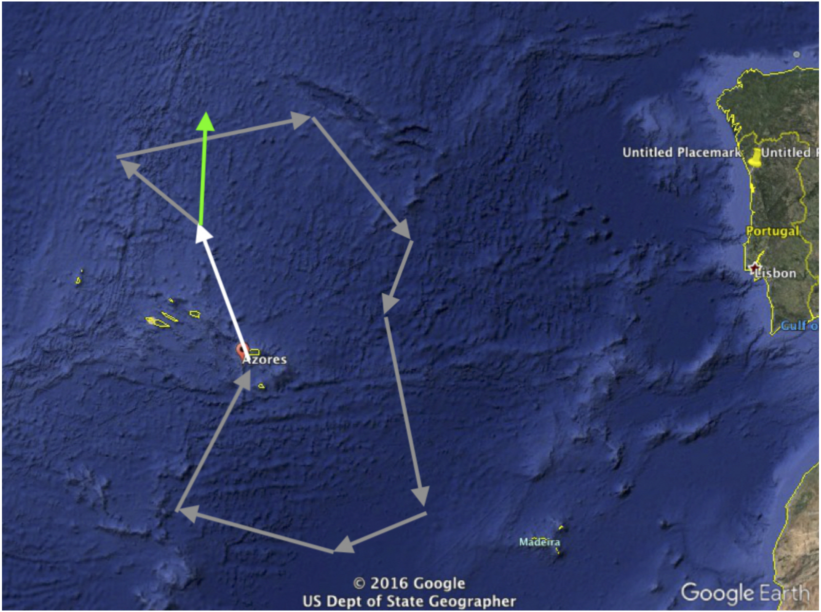
→ choose points with high uncertainty in current estimation



Our approach: static version

- Take known seafloor contents data
- Build assessment for attractiveness based on known points:
  - evaluate "attractiveness" (variance) on a fine mesh ⚠
- Orienteering: repeat:
  1. select point with highest variance
  2. find tour T with feasible length
     - if no such tour exists, break
  3. simulate probing that point; recompute "attractiveness" ⚠
- Probe: evaluate true function for all $(x, y) \in T$ ⚠
- Estimation: evaluate resource level allover the surface (GP)

Setting


Untitled Placemark
Portugal
Lisbon
Azores
Madeira
© 2016 Google
US Dept of State Geographer
Google Earth

Our approach: online version

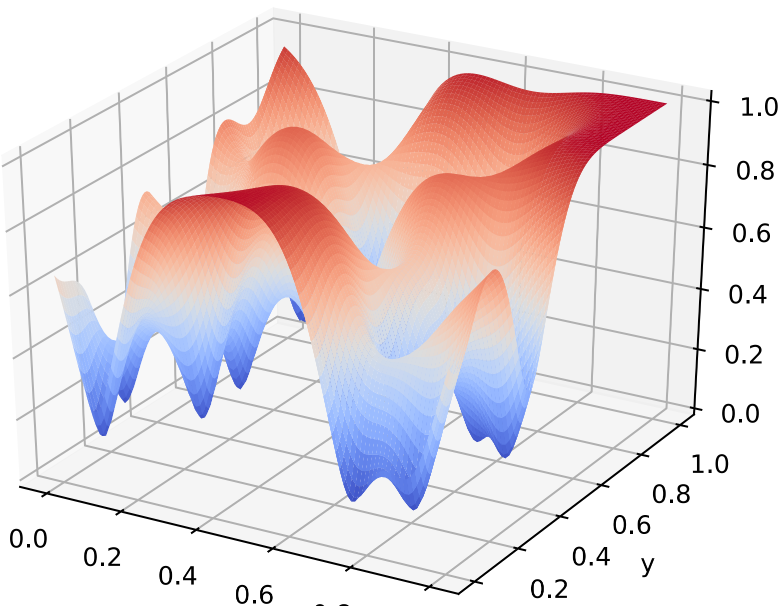
- Until there's no time for an additional probing, repeat:
  - given:
    - previous data
    - data collected in current position
  - determine remaining part of the trip
  - commit to the next point to visit

Algorithm

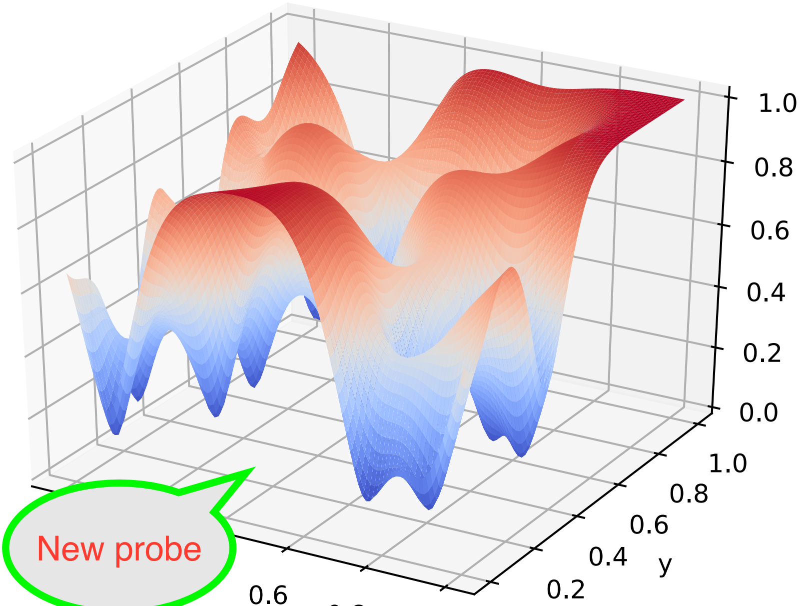
[Input: previous data + current and final positions]

1. Initialization
   - draw a random point within feasible region
   - if feasible, insert it into current path and repeat
2. Evaluation
   - train Gaussian Process with current data
   - for each point in current solution:
     - check variance with GP → ⚠correlations
     - assume expectation of GP = true value
3. Improvement: repeat until failing:
   - *insertion:* attempt a new probe; if feasible, insert in current trip
   - *random motion:* for each probe, attempt some points around it; if improved, move there
4. Update incumbent
5. Pertubation:
   - remove a random point from the solution

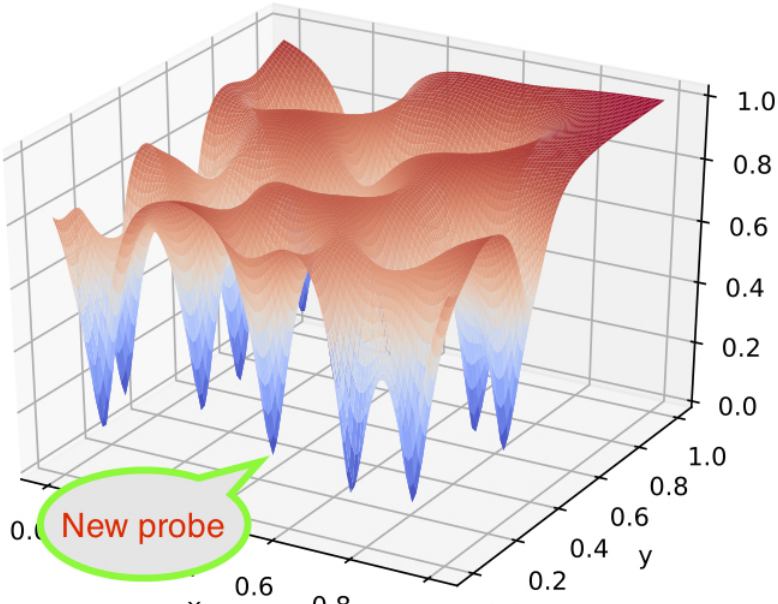
Probing



Probing



Probing

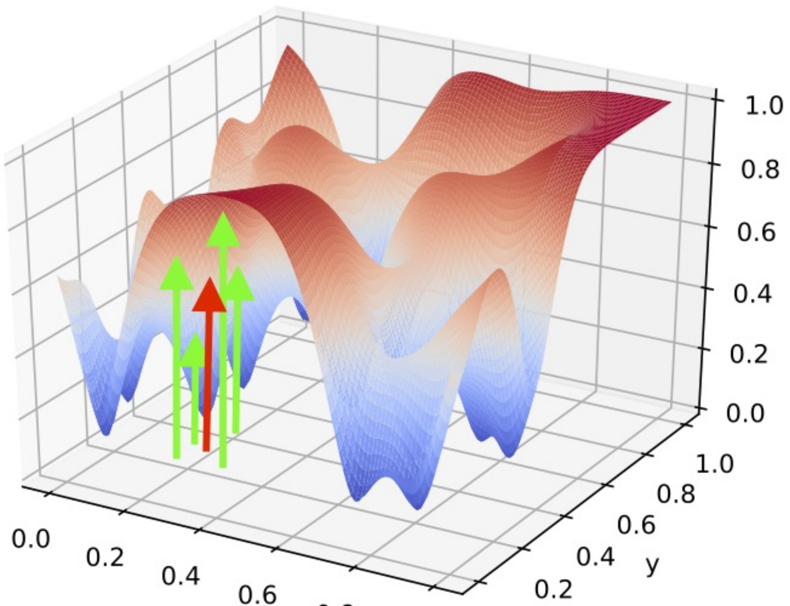


Algorithm

[Input: previous data + current and final positions]

1. Initialization
   - draw a random point within feasible region
   - if feasible, insert it into current path and repeat
2. Evaluation
   - train Gaussian Process with current data
   - for each point in current solution:
     - check variance with GP → ⚠correlations
     - assume expectation of GP = true value
3. Improvement: repeat until failing:
   - *insertion:* attempt a new probe; if feasible, insert in current trip
   - *random motion:* for each probe, attempt some points around it; if improved, move there
4. Update incumbent
5. Pertubation:
   - remove a random point from the solution

Neighborhoods

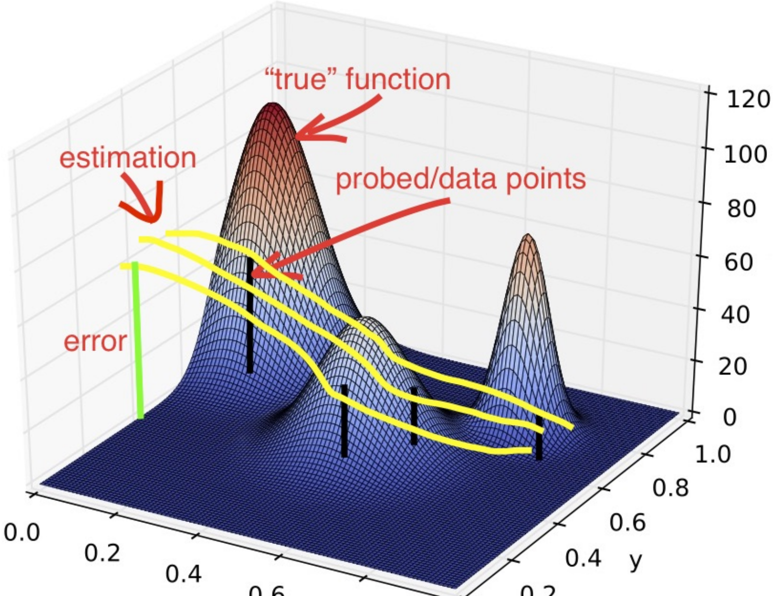


Algorithm

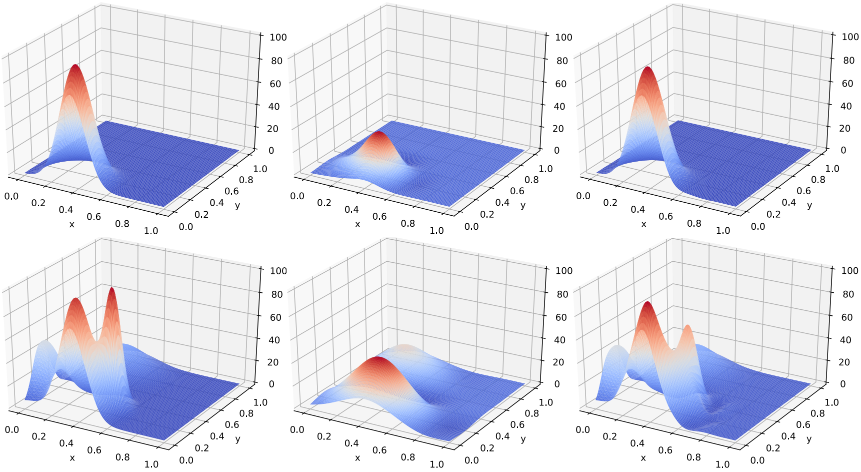
[Input: previous data + current and final positions]

1. Initialization
   - draw a random point within feasible region
   - if feasible, insert it into current path and repeat
2. Evaluation
   - train Gaussian Process with current data
   - for each point in current solution:
     - check variance with GP → ⚠correlations
     - assume expectation of GP = true value
3. Improvement: repeat until failing:
   - *insertion:* attempt a new probe; if feasible, insert in current trip
   - *random motion:* for each probe, attempt some points around it; if improved, move there
4. Update incumbent
5. Pertubation:
   - remove a random point from the solution

Benchmarking: integrate error over relevant surface



Some results



In summary

- First attempt to model and solve online version of this problem
- Method:
  1. assessment: initial estimation based on current data [ML]
  2. planning: construct a trip for probing new points → CO
  3. final estimation: use previous data + newly probed points [ML]
- Online version:
  - CPU usage important → dumb grid search too time consuming
  - selecting few evaluation points: borrowing ideas from metaheuristics
  - planning: MIP solvers are quick enough, if correctly employed
- Benchmarking:
  - compare "true" (artificial) function to predicted data