Water Resources Engineering
Exercise

Andrea Rinaldo (andrea.rinaldo@epfl.ch)
Luca Carraro (luca.carraro@epfl.ch)
Jonathan Giezendanner (jonathan.giezendanner@epfl.ch)

lecture: Tuesday 8:15-10 in room GR B3 30
exercise: Thursday 10:15-20 in room IN F 03

Goal

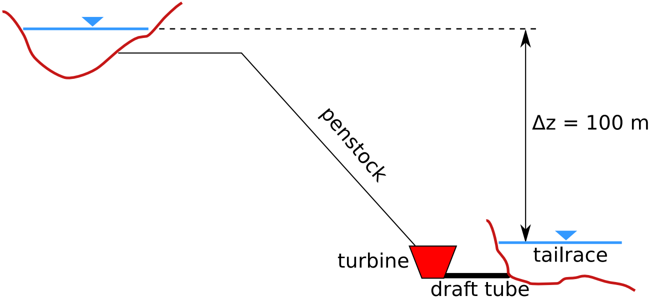
We need to find parameters for our model (Q_{mod}) which best describe our observations (Q_{obs})

What is Optimisation?

Find a set of parameters which optimises (maximises or minimises) target function.

Analytical Solution

Remember Exercise 6.2



Target function, power as a function of discharge:

$$P(Q) = \eta\gamma\Delta z Q - \eta\gamma K Q^3$$

Optimise (maximise) power, set derivative to zero:

$$\frac{dP}{dQ} = \eta\gamma\Delta z - 3\eta\gamma K Q^2 = 0$$

No Analytical Solution

- Too complicated
- Complex Simulation
- Non-existing (most cases)

$\Rightarrow$ Rely on sampling of solution space

No Analytical Solution

- Too complicated
- Complex Simulation
- Non-existing (most cases)

$\Rightarrow$ Rely on sampling of solution space

Brute Force

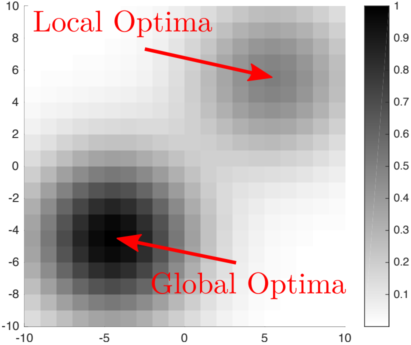
Sample every Parameter Combination

Example

- Optimise:
  $f(x,y) = exp\left(-\frac{(x+5)^2+(y+5)^2)}{5^2}\right) + \frac{1}{2} exp\left(-\frac{(x-5)^2+(y-5)^2)}{5^2}\right)$
- Domain: $x \in -10..10$; $y \in -10..10$

Example

- Optimise:
  $f(x,y) = exp\left(-\frac{(x+5)^2+(y+5)^2)}{5^2}\right) + \frac{1}{2}exp\left(-\frac{(x-5)^2+(y-5)^2)}{5^2}\right)$
- Domain: $x \in -10..10$; $y \in -10..10$



Feasible if

- Evaluation of the objective function does not take too long
- Not too many parameters
- Parameter range not too large

Possible alternative: Meta Heuristic Method

Accept fact that you are not guaranteed to find the best solution, but one possible solution

Possible alternative: Meta Heuristic Method

Accept fact that you are not guaranteed to find the best solution, but one possible solution

Random Walk

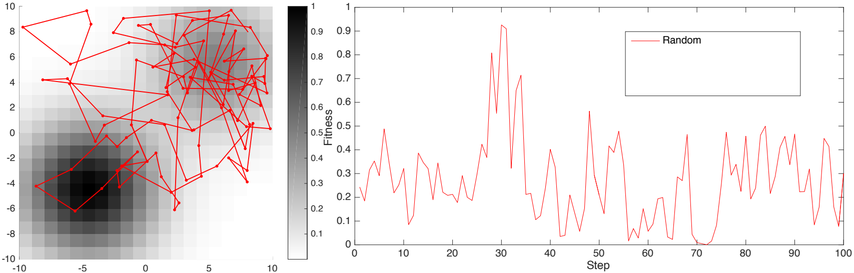
Simplest Algorithm: Randomly sample space

- Start at a given position $x = x_{t=0}$, $y = y_{t=0}$
- Randomly select new parameter values around current value:

$$
\begin{aligned}
x_t =& \mathcal{N}\left(x_{t-1}, \sigma\right) \in \left[x_{min}, x_{max}\right] \\
y_t =& \mathcal{N}\left(y_{t-1}, \sigma\right) \in \left[y_{min}, y_{max}\right]
\end{aligned}
$$

- Iterate

One alternative: Meta Heuristic Method



Problem: Wanders aimlessly around

Greedy Algorithm - Move only to better local solution

- Start at a given position $x = x_{t=0}$, $y = y_{t=0}$
- Randomly select new parameter values around current value:

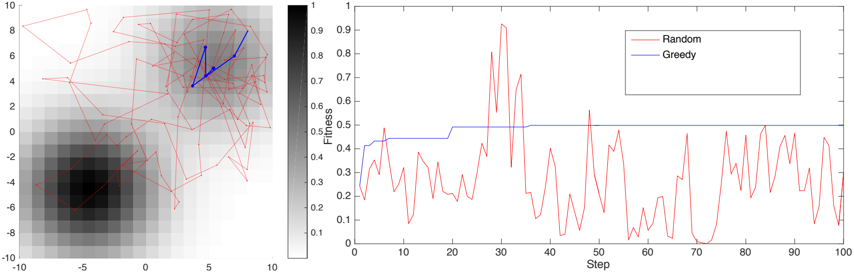
$$x' = \mathcal{N}\left(x_{t-1}, \sigma\right) \in \left[x_{min}, x_{max}\right]$$
$$y' = \mathcal{N}\left(y_{t-1}, \sigma\right) \in \left[y_{min}, y_{max}\right]$$

- if $f\left(x', y'\right) > f\left(x_{t-1}, y_{t-1}\right)$

$$x_t = x', \; y_t = y'$$

- Iterate

Meta Heuristic Method



Problem: can easily get stuck in local optima

Metropolis Algorithm - Add possibility to jump to worse solution

- Start at a given position $x = x_{t=0}$, $y = y_{t=0}$
- Randomly select new parameter values around current value:

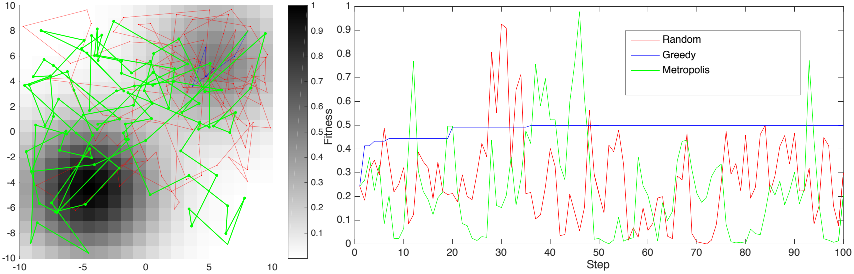
$$x' = \mathcal{N}\left(x_{t-1}, \sigma\right) \in \left[x_{min}, x_{max}\right]$$
$$y' = \mathcal{N}\left(y_{t-1}, \sigma\right) \in \left[y_{min}, y_{max}\right]$$

- if $f\left(x', y'\right) > f\left(x_{t-1}, y_{t-1}\right)$ **or** $r < \exp\left(f\left(x', y'\right) - f\left(x_{t-1}, y_{t-1}\right)\right)$, where $r = \mathcal{U}\left(0, 1\right)$

$$x_t = x', \; y_t = y'$$

- Iterate

Meta Heuristic Method



Problem: not meant for optimisation, supposed to efficiently sample parameter space around the global optimum (i.e. posterior distribution), which is not what we want to do here.

Meta Heuristic Method - Simulated Annealing

Simulated Annealing - Decay of jumping probability in time

Idea comes from metallurgy: heating and controlled cooling

- Start by large solution space sampling
- Refine search with time

Simulated Annealing - Decay of jumping probability in time

Idea comes from metallurgy: heating and controlled cooling

- Start by large solution space sampling
- Refine search with time

- Start at a given position $x = x_{t=0}$, $y = y_{t=0}$
- Randomly select new parameter values around current value:

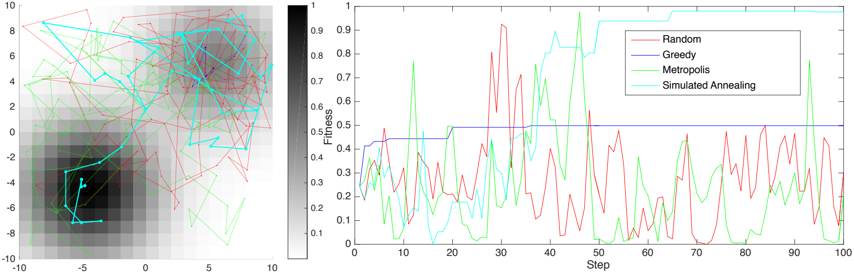
$$x' = \mathcal{N}(x_{t-1}, \sigma) \in [x_{min}, x_{max}],\ y' = \mathcal{N}(y_{t-1}, \sigma) \in [y_{min}, y_{max}]$$

- $T_t = \exp(-iteration\ \# \cdot cooling\ rate)$
- if $f(x', y') > f(x_{t-1}, y_{t-1})$ **or** $r < \exp((f(x', y') - f(x_{t-1}, y_{t-1})) / \mathbf{T_t})$

$$x_t = x',\ y_t = y'$$

- Iterate $\Rightarrow$ Save best configuration

Meta Heuristic Method - Simulated Annealing



Goal

Find parameter set $\vec{\theta}(k_{sat}, c, t_{sub}, z)$ which maximises Nash - Sutcliffe coefficient (see last week).

Procedure

1. Define a functional form for the temperature

$$T_{SA}(i) = \exp(-c_r \cdot i)$$

   where i counts the iterations, while c_r is a cooling rate

2. Attribute arbitrary values to the parameter set $\vec{\theta}(K_{sat},\, c,\, t_{sub},\, z)$. Run the hydrological model and evaluate NS_{old}.

Procedure

3. Select a new parameter set $\vec{\theta}_{\mathrm{new}}$ by drawing from a truncated normal distribution (function `TruncNormRnd.m`) around the old values (as in the Greedy Algorithm).
4. Run the hydrological model with the new parameters and evaluate NS_{new}.
5. **If** $NS_{\mathrm{new}} > NS_{\mathrm{old}}$, **then** accept the new parameter set, and save best config:
$\vec{\theta}_{\mathrm{old}} = \vec{\theta}_{\mathrm{new}},\ \vec{\theta}_{\mathrm{best}} = \vec{\theta}_{\mathrm{new}},\ NS_{\mathrm{old}} = NS_{\mathrm{new}}$
6. **Else**, accept the new parameter set with probability

$$\textbf{if} \quad r < \exp\left(\frac{NS_{new} - NS_{old}}{T_{SA}(i)}\right),\ r \in \mathcal{U}(0,1)$$

$$\textbf{then} \qquad \vec{\theta}_{\mathrm{old}} = \vec{\theta}_{\mathrm{new}},\ NS_{\mathrm{old}} = NS_{\mathrm{new}}$$

7. Repeat from 3. until convergence. A good fitting should be around $NS = 0.87$.

For the remaining of assignment:

- Use parameter set with best NS coefficient you found here
- In next week's code you will have to hardcode the values found, not rerun the code
- Don't forget to make the plots for the final report

Tune cooling rate and standard deviation of parameter distribution (σ)

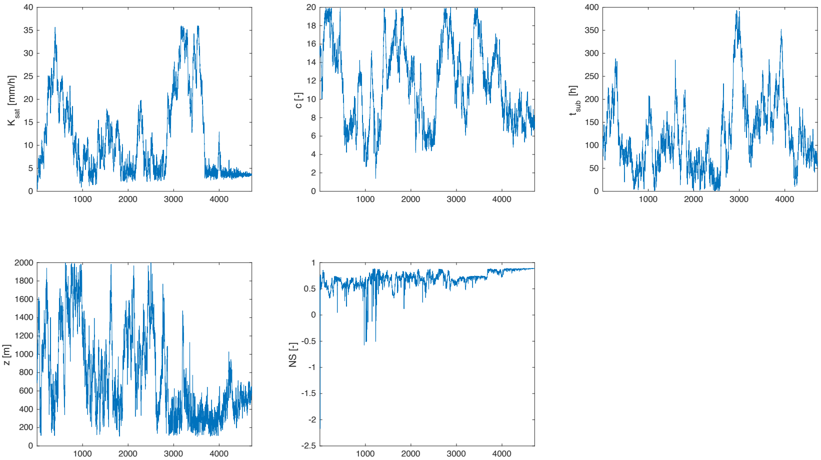
- Try with σ around 5% of parameter range (for each parameter)
- Try with cooling rate $c_r = 1/1200$
  - Too large: not efficient sampling of solution space (too greedy)
  - Too small: Sampling of parameter space accurate, but takes too long without converging

Use large enough N_{iter} to ensure convergence

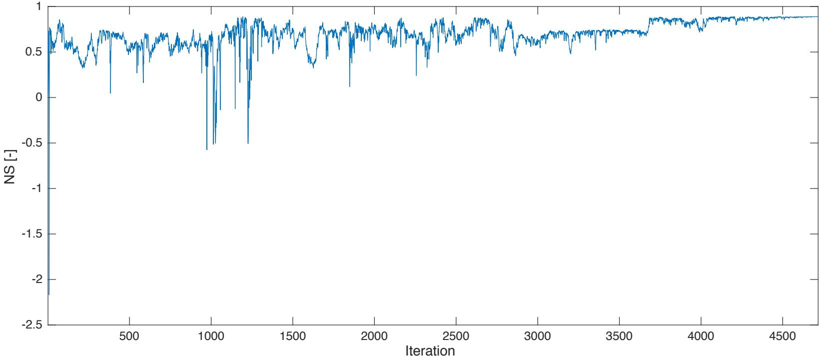
- If N_{iter} too small, the algorithm may still accept worse solutions towards the end.
- If N_{iter} too large, the chain gets stuck for too long in a (global?) optimum.

Example

Parameter Sampling and NS coefficient

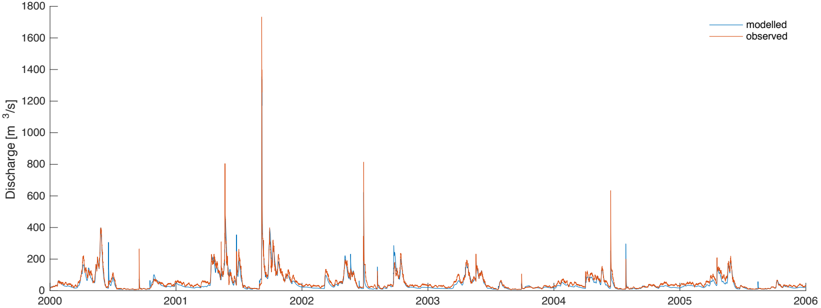


Assignment - NS Score



Use parameter set corresponding to best NS coefficient

Assignment - Model vs Observations



Find NS around .87