Heuristics - Set Covering Problem

Felipe Coutinho

October 30, 2024

FEUP

Preliminaries

Solution representation

Data

$$A \in \mathbb{A}^{n \times m}, \ \mathbb{A} = \{0, 1\}$$

- Columns: subsets
- Rows: attributes

Costs

$$\omega \in \mathbb{N}^m$$

Solution

$$s \in \mathbb{A}^m$$

Simulated Annealing

Cooling schedule

Geometric: $T_n = \alpha \times T_{n-1}$

Move

Swap(1,2): remove 1, insert *up to* 2

Repair

Move can create partial solutions

Insert sets with lowest ω until complete solution

Extra sets cost $+25\%$ (*penalty* for moving to partial sol.)

Initial temperature

Stuff that failed

Too hot causes solution cost to explode early on

Tried:

$$T_i = k\sigma, \;\; k = -3/\log p$$

Failed: needed $p \ll 0.1$ for OK behavior, but that is just local search!

SA - Example

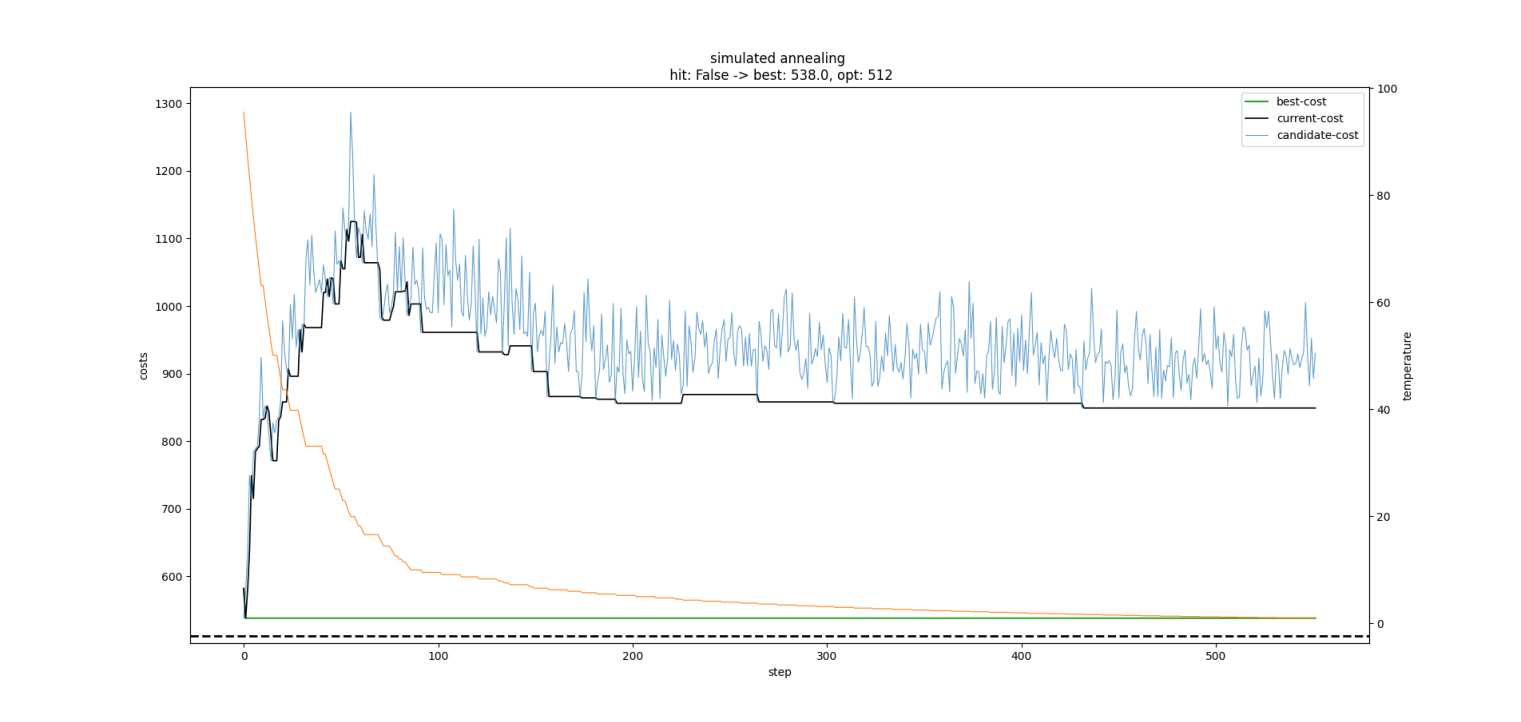

Figure 1: Too much divergence early + cooling too fast!

Initial temperature

Stuff that failed

We want candidates to oscillate around initial cost...

So match initial cost to expected fitness at 1st step!

Tried:

$$T_i = \arg\min_T |\mathbb{E}[\nu_1](T) - \nu_0|$$

Failed: behaved poorly for most instance classes

Initial temperature

Light at the end of the tunnel!

Simple grid search.

Pick $N = 5$ values of T between $T_i = [1, 100]$

For one instance for each class, find best T_i^*

Fine tune it by searching again within $T_i = [T_i - \delta T, T_i + \delta T]$ (used $\delta T = 5, N = 5$ again)

SA - Example

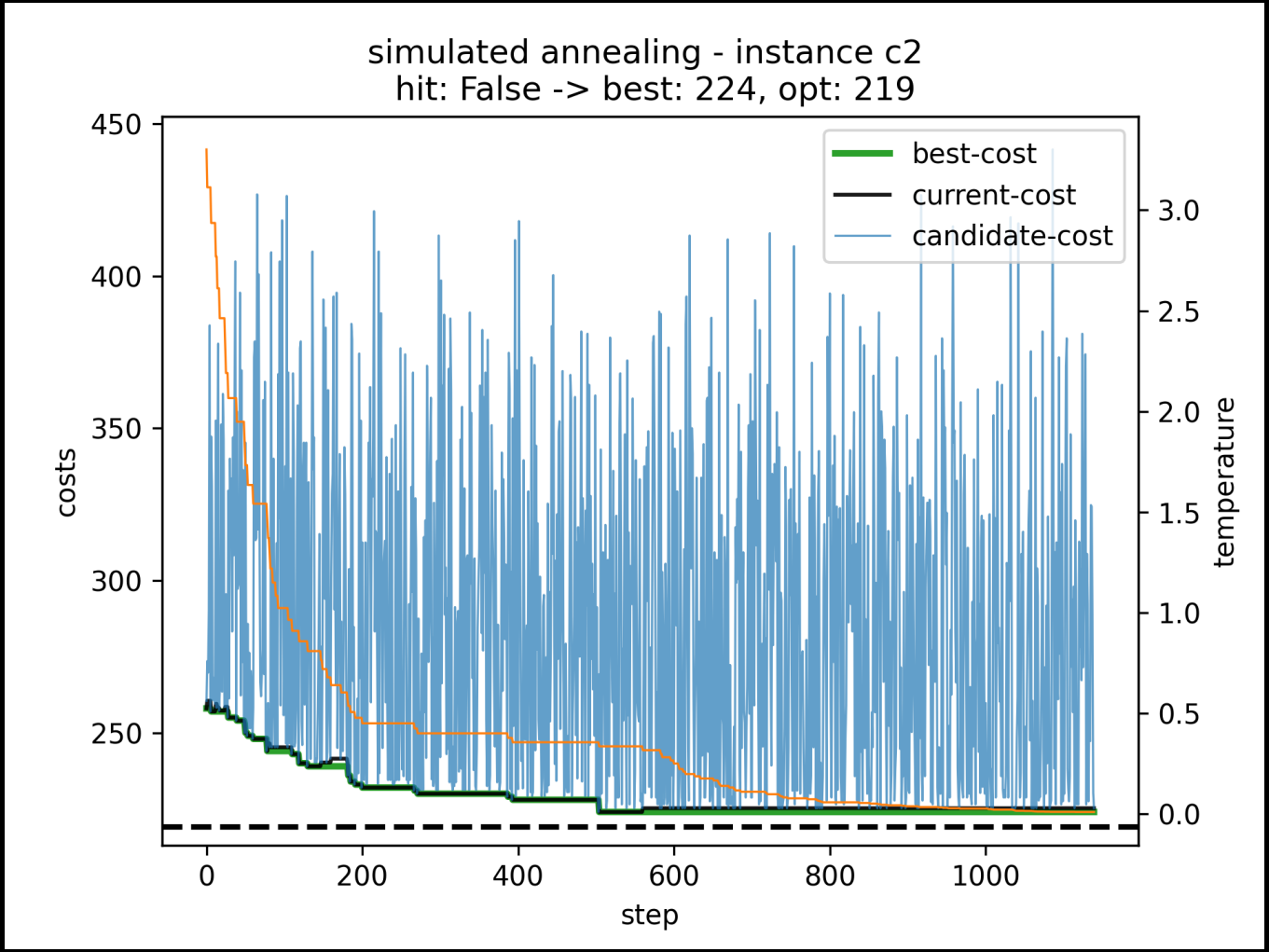

Figure 2: My algorithm running for C2 instance

SA - Results

Not very impressive...

Instance Class	4	5	6	a	b	c	d
T_i	30	3.5	1.0	6.0	1.0	3.5	1.0
Error (%)	3.9	4.6	8.2	4.6	3.2	4.2	8.7
Hits	0	0	0	0	0	0	0

Table 1: Simulated Annealing results per instance class.

Variable Neighborhood Search

Shaking

Nested neighborhood structure

- $k = 1$: Swap(1,2)
- $k = 2$: Swap(1,2) twice
- $k = 3$: etc.

Repair as before!

Local search

Move: Flip(1) (i.e. *insert* or *remove* 1 at random)

Best-improvement strategy with $N = m$ iterations

Repair + RE!

VNS - Example

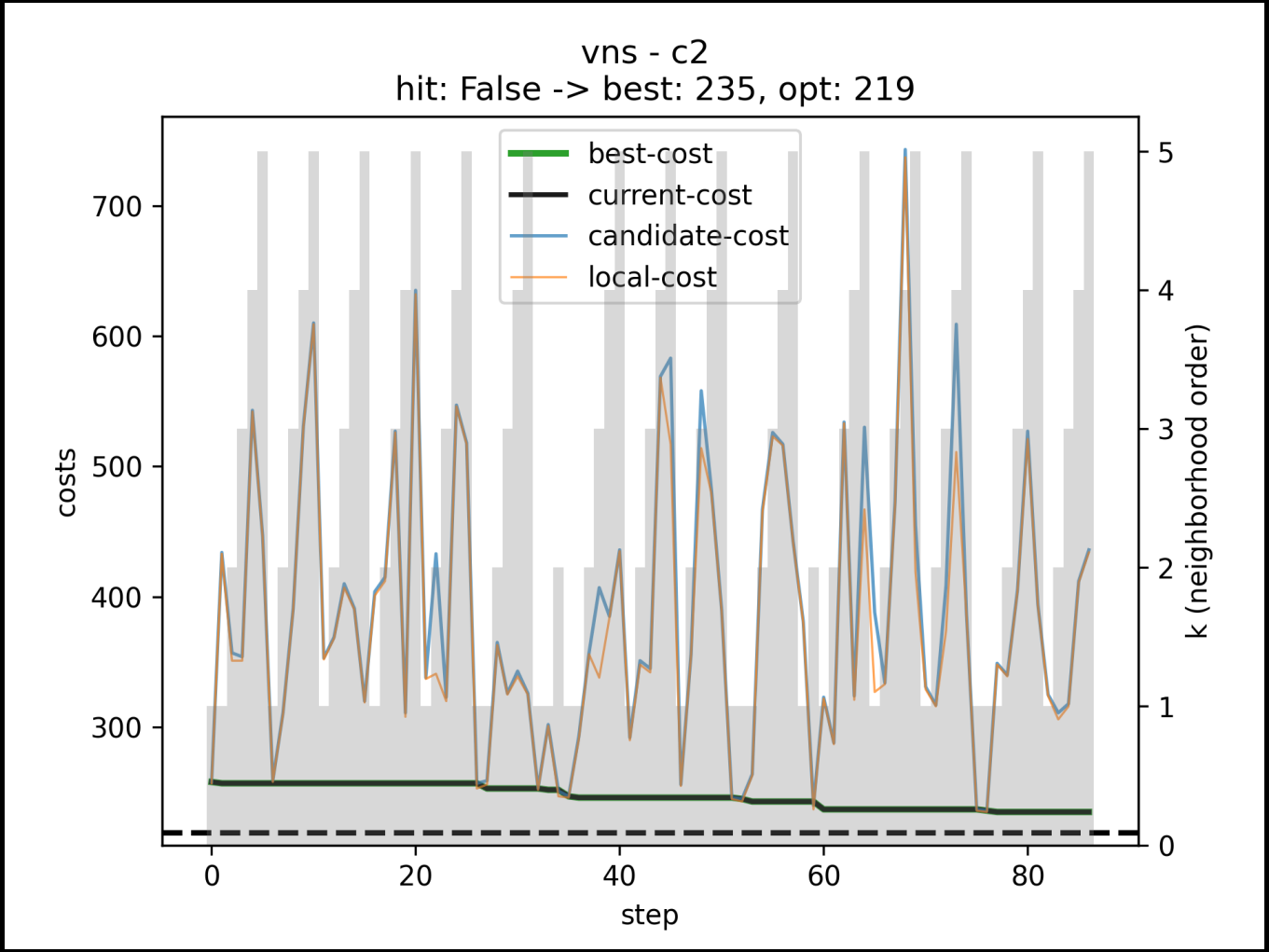

Figure 3: My algorithm running for C2 instance

VNS - Results

Not very impressive...

Instance Class	4	5	6	a	b	c	d
Error (%)	5.8	6.2	7.6	9.2	4.3	6.2	8.4
Hits	0	0	0	0	0	0	0

Table 2: VNS results per instance class, for $M = 25$.