Aalto-yliopisto
Perustieteiden korkeakoulu

Cyclic Placement Method for Capsule Packing Problem

Lauri Jokinen

27.8.2018

Ohjaaja ja valvoja: Harri Ehtamo

Systeemianalyysin laboratorio

Earlier research

- Packing problem is well studied and defined for various shapes, but only a few papers exist on capsule packing
- Mirko Ruokokoski, KONE Oy: ellipse packing
  - How many persons realistically fit in an elevator car?
  - Can be used further in larger scale simulations
  - In general a packing problem, but social couplings could be also implemented

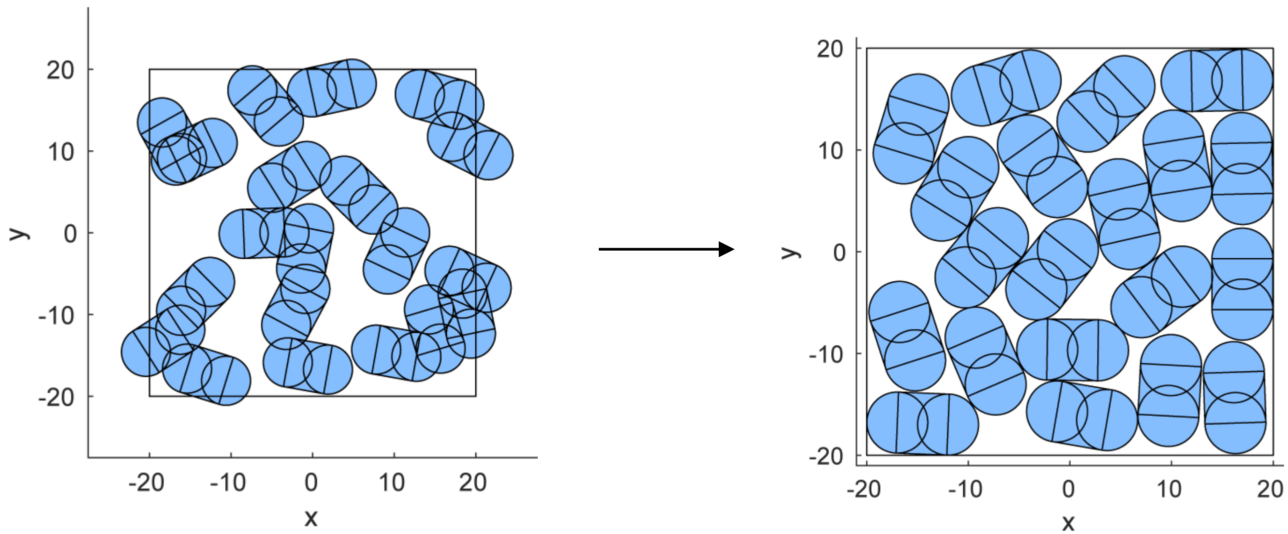
My study: capsule packing

– Approach:

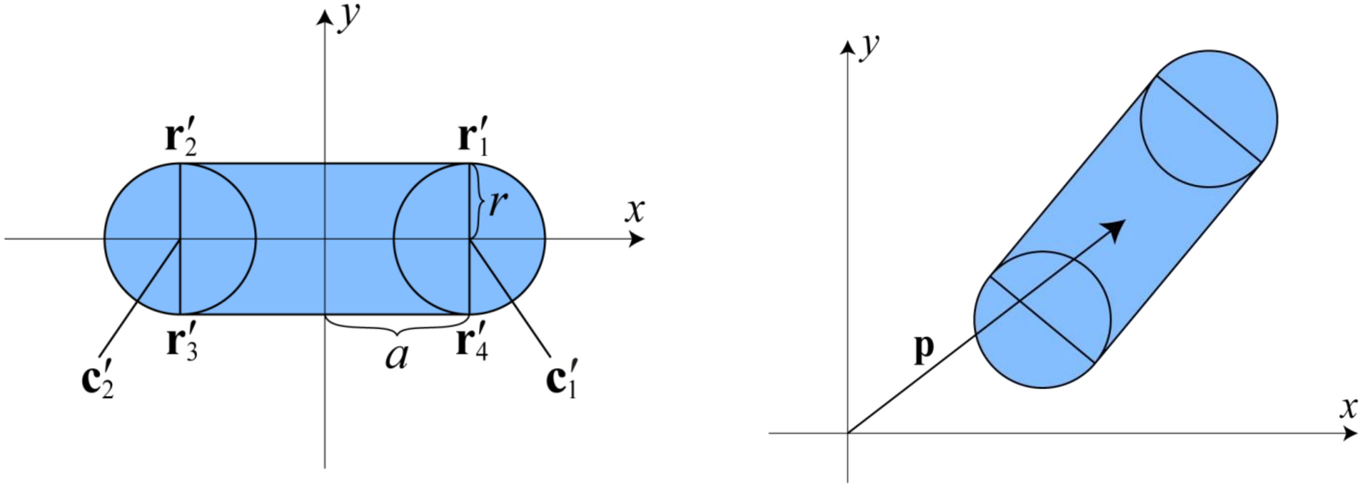
Random initial setting, overlaps allowed

→ Overlapping areas are minimized

→ Local optimum is found



Capsules



- Defined with width and height. Location is defined with $\mathbf{p}$ and θ. Let a capsule be $\mathbf{s} = [\mathbf{p} \;\; \theta]$.
- Intersection area of two capsules can be solved analytically

Objective function to be minimized

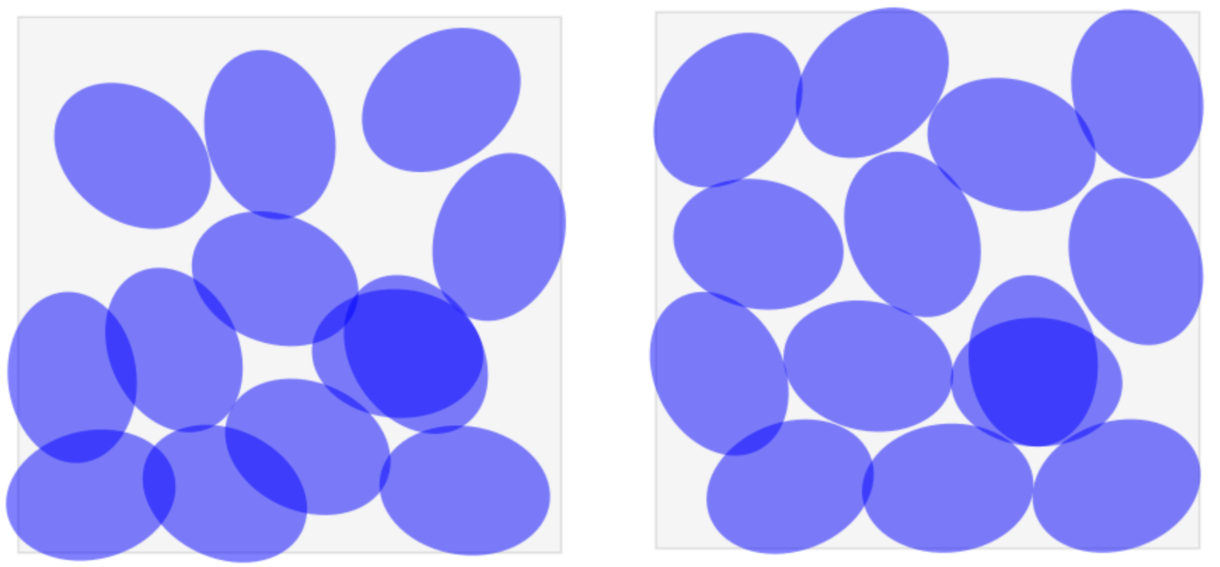
$$\min\left(\sum_{i,j\in N,\, i<j} A(\mathbf{s}_i,\mathbf{s}_j) + \sum_{i\in N}\left(A(\mathbf{s}_i) - A(\mathbf{s}_i,B)\right)\right), \qquad N=\{1,\ldots,n\}$$

Areas of capsules outside the room

Intersecting areas of
combinations of two capsules

- Minimized using a gradient method (Matlab's Opt. Toolbox)
- Used by Ruokokoski

Objective function

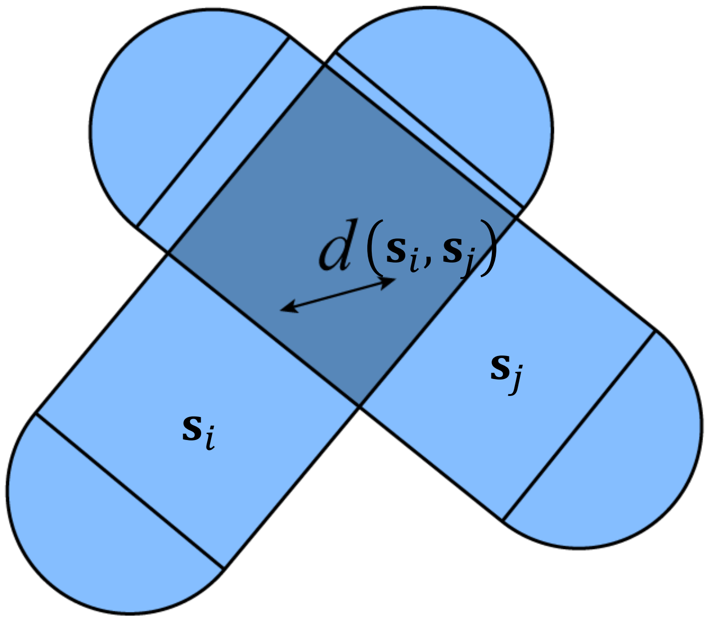


Objective function

- Repelling forces
  - Secondary optimization target; a penaulty term

$$\min\left(\frac{1}{d(\mathbf{s}_i, \mathbf{s}_j) + 1}\right)$$

$$d(\mathbf{s}_i, \mathbf{s}_j) = |\mathbf{p}_i - \mathbf{p}_j|$$



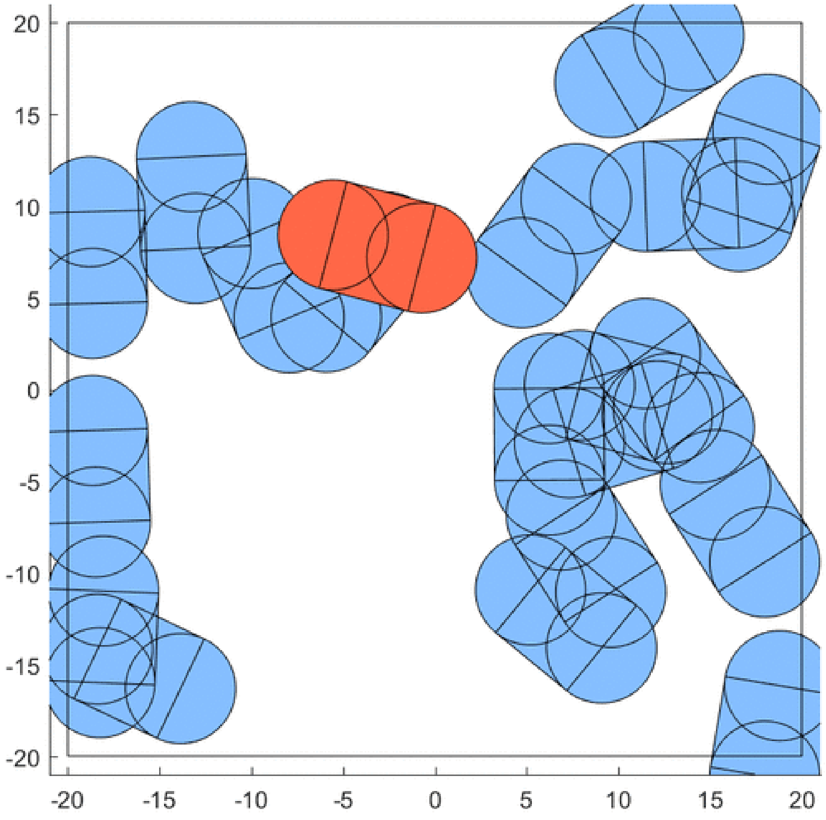
Objective function

- Objective function $F(S)$
- Set of capsules $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, $N = \{1, \ldots, n\}$.

$$F(S) = \sum_{i,j \in N,\, i<j} \left[A(\mathbf{s}_i, \mathbf{s}_j) + \frac{10^{-6}}{d(\mathbf{s}_i, \mathbf{s}_j) + 1} \right] + \sum_{i \in N} \left(A(\mathbf{s}_i) - A(\mathbf{s}_i, B) \right)$$

Cyclic placement method

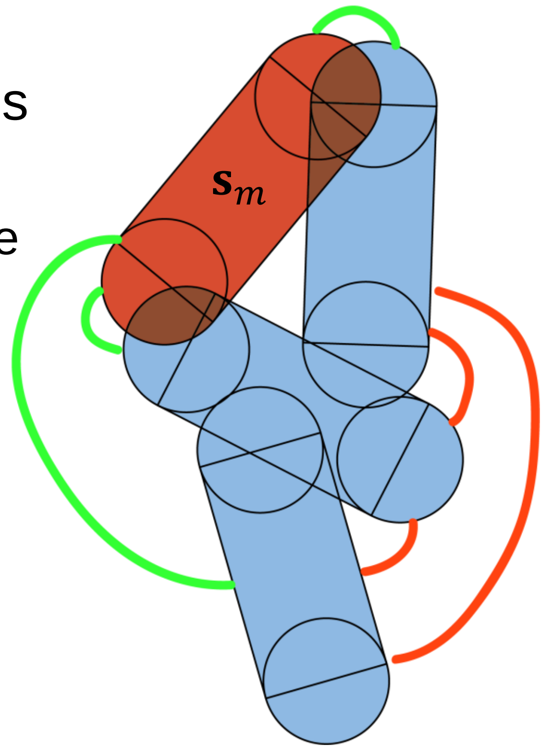
– Move a single capsule to a bit more optimal location



Cyclic placement method

– Optimize one capsule $\boldsymbol{s}_m$ at a time (with gradient method) and keep other capsules in their positions
  - We only calculate terms of $F(S)$, which are dependent of $\boldsymbol{s}_m$

    → New target function $f(\boldsymbol{s}_m)$



Objective functions

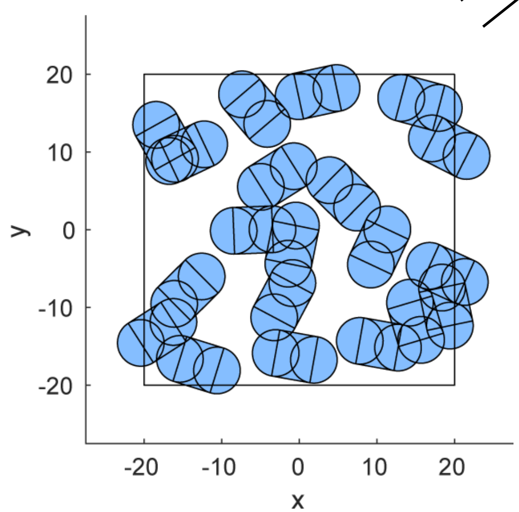
– Set of capsules $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, $N = \{1, \ldots, n\}$.

$$F(S) = \sum_{i,j \in N,\, i<j} \left[A(\mathbf{s}_i, \mathbf{s}_j) + \frac{10^{-6}}{d(\mathbf{s}_i, \mathbf{s}_j) + 1} \right] + \sum_{i \in N} \left(A(\mathbf{s}_i) - A(\mathbf{s}_i, B) \right)$$

$$f(\mathbf{s}_m) = \sum_{i \in N \setminus \{m\}} \left[A(\mathbf{s}_i, \mathbf{s}_m) + \frac{10^{-6}}{d(\mathbf{s}_i, \mathbf{s}_m) + 1} \right] + \left(A(\mathbf{s}_m) - A(\mathbf{s}_m, B) \right)$$

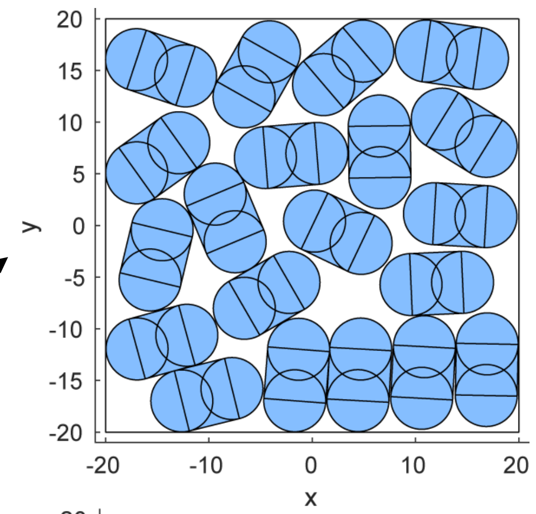
Comparison of the algorithms

– Random initial state
– Simulate until every term of gradient is less than ε. ($\varepsilon > 0$)

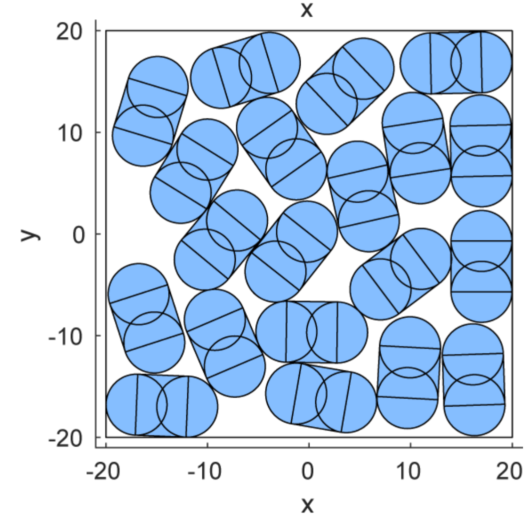


Random initial state: n = 20, b = 40

Basic gradient

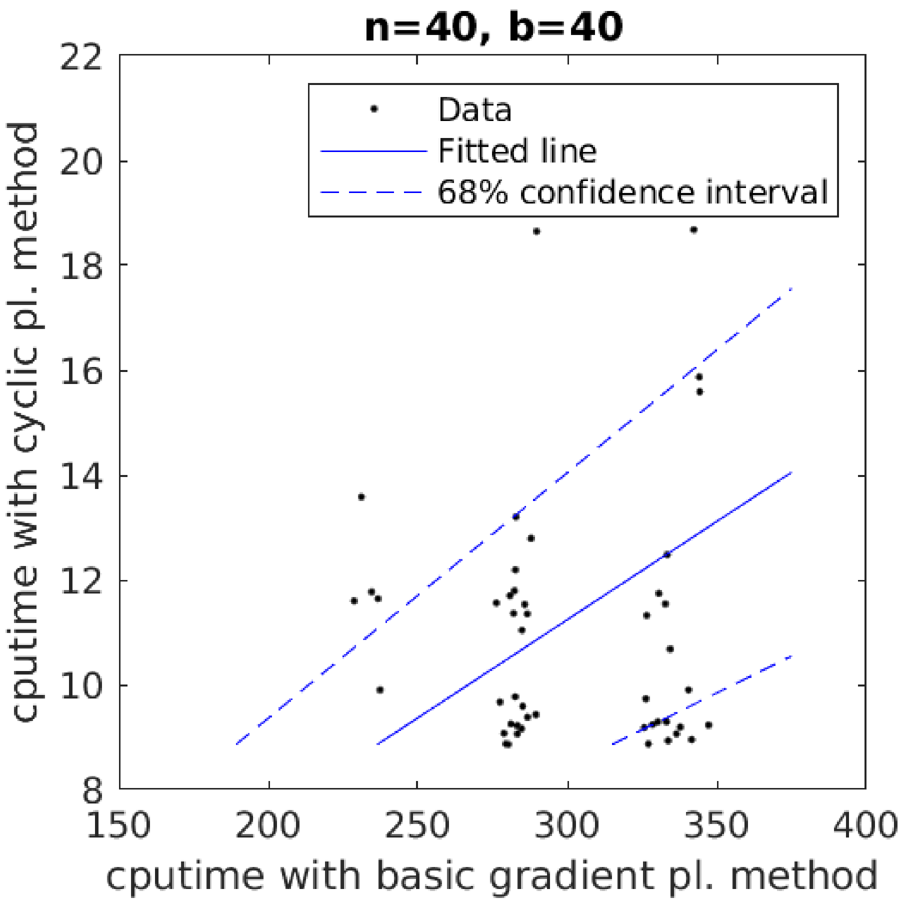


Cyclic

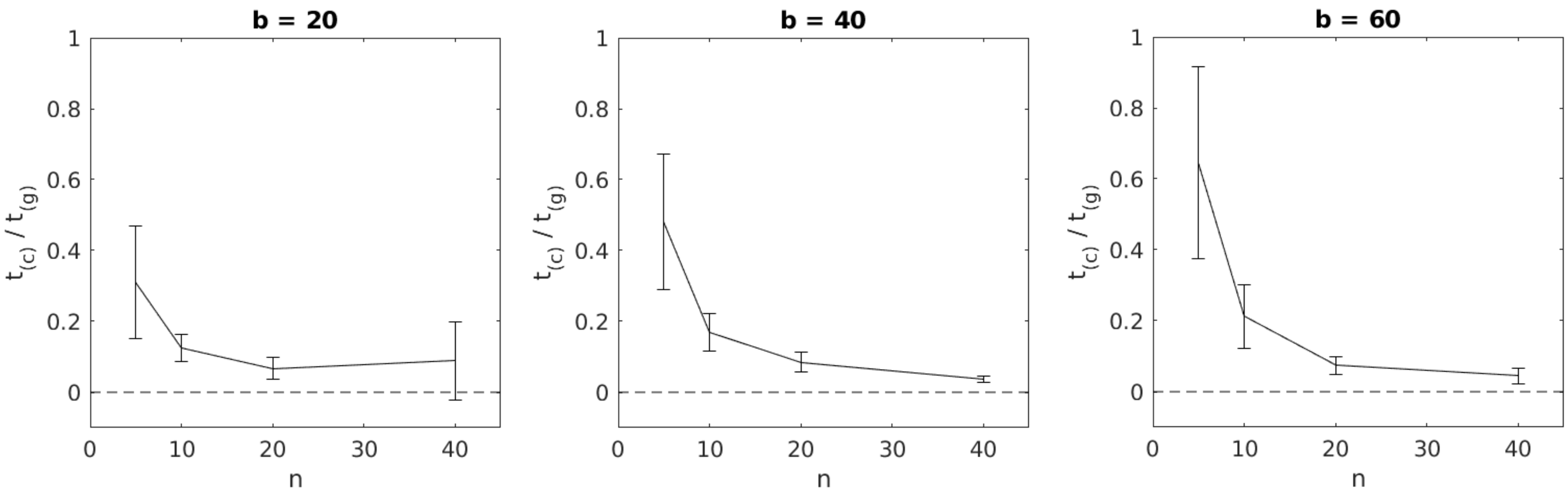


Results

- Fitted with $y = ax$
- In this case $a = 0.037 \pm 0.009$ (with 68% confidence interval)



Results



Aalto-yliopisto
Perustieteiden korkeakoulu

Thank you!

Systeemianalyysin laboratorio