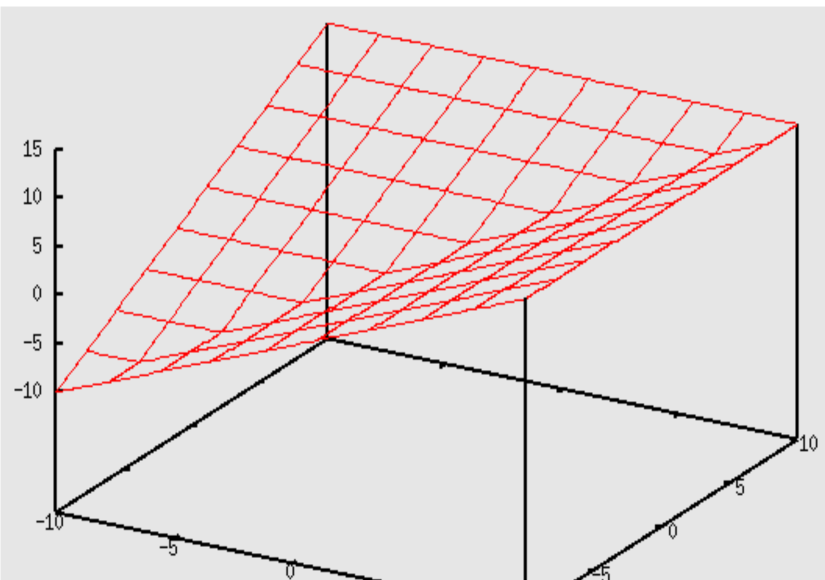
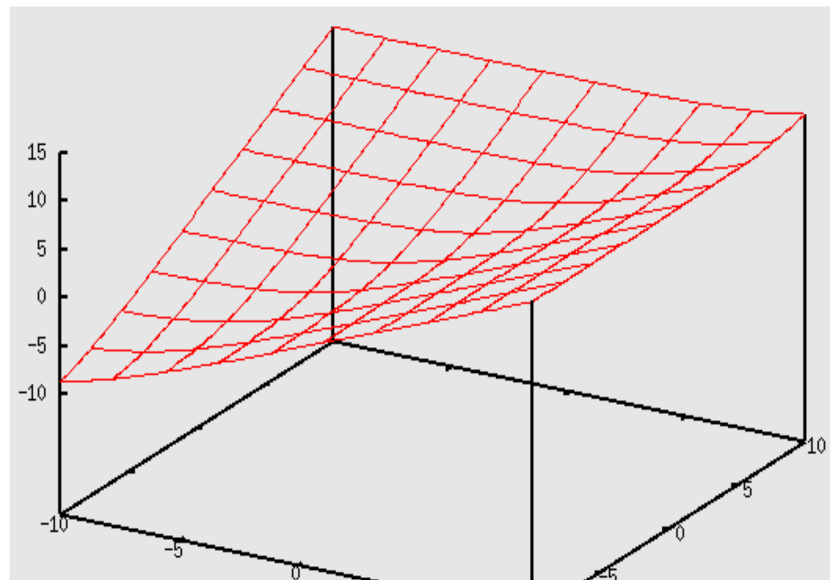
Log-Sum-Exponential (LSE) Function

- An approximation of the maximum function:
  - $\mathrm{LSE}_\alpha(z_1, \ldots, z_n) = \alpha \times \left(\log\left(\sum_{i=1}^{n} e^{z_i/\alpha}\right)\right) \approx \max(z_1, \ldots, z_n)$
  - Strictly convex and continuously differentiable
  - α : smoothing parameter (exact when $\alpha \rightarrow 0$)



α=0.1



α=2

Expression of HPWL using LSE Function

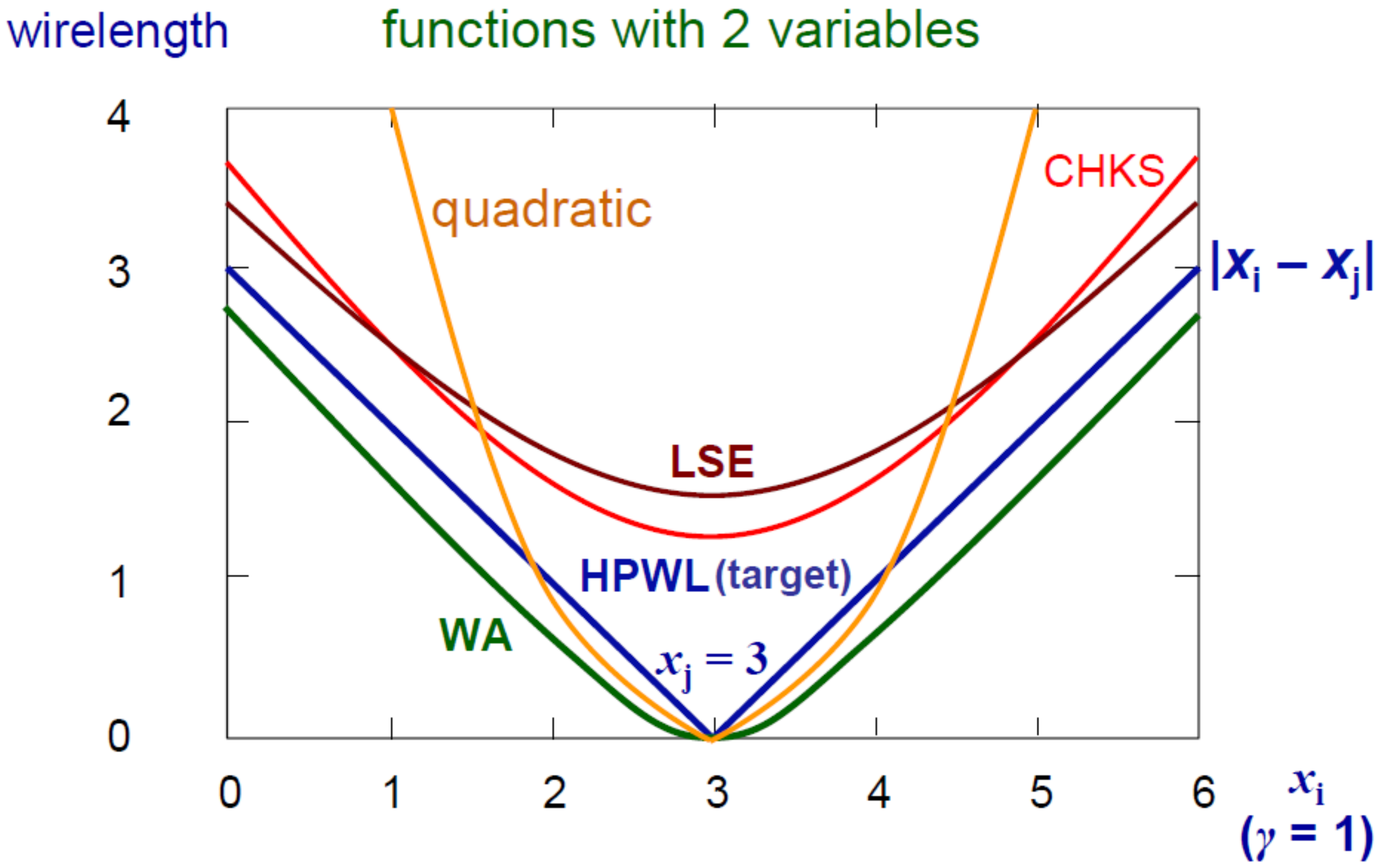
- HPWL in terms of max:

$$
\begin{aligned}
&\mathrm{HPWL}_e(x_1, \ldots, x_n,\ y_1, \ldots, y_n) \\
&= \left(\max_{i\in e}\{x_i\} - \min_{i\in e}\{x_i\}\right) + \left(\max_{i\in e}\{y_i\} - \min_{i\in e}\{y_i\}\right) \\
&= \left(\max_{i\in e}\{x_i\} + \max_{i\in e}\{-x_i\}\right) + \left(\max_{i\in e}\{y_i\} + \max_{i\in e}\{-y_i\}\right)
\end{aligned}
$$

- After approximating max by LSE:

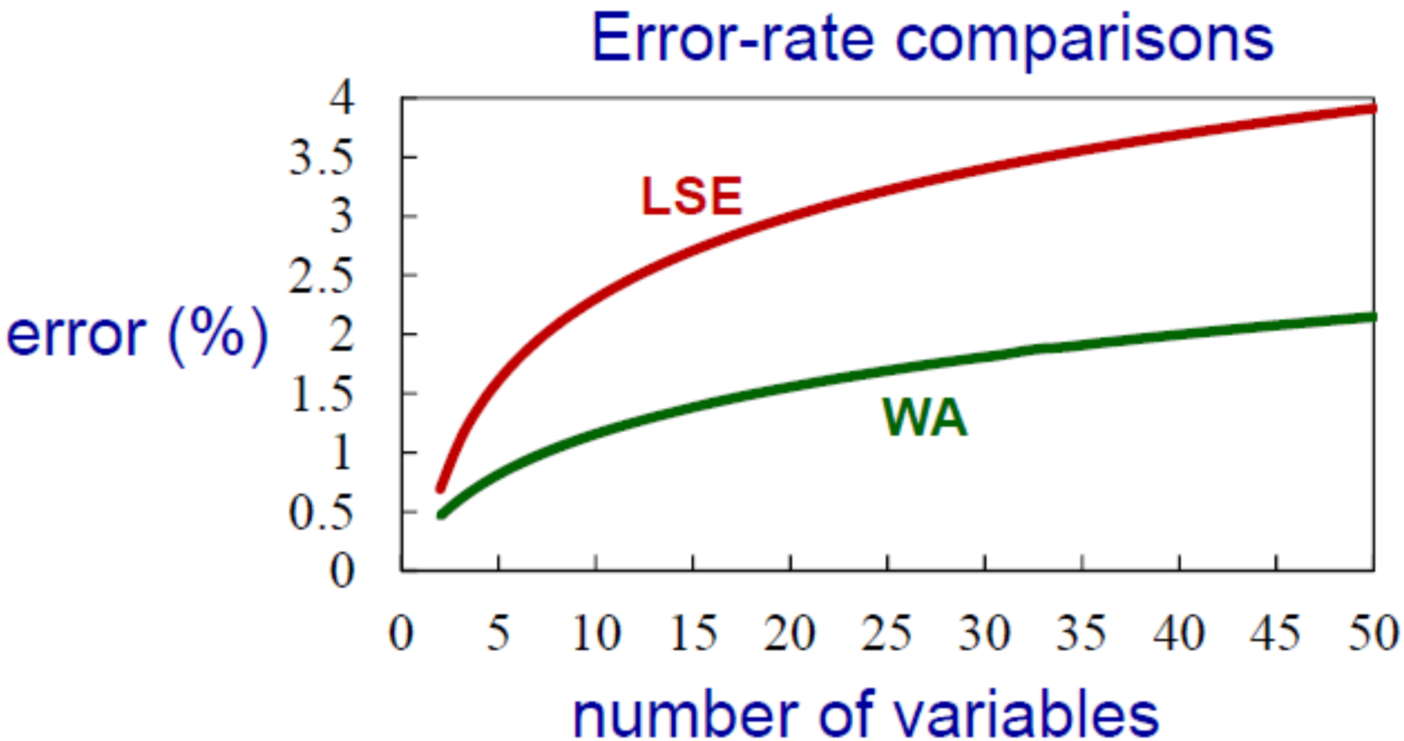
$$
\begin{aligned}
&\mathrm{LSEWL}_{e,\alpha}(x_1, \ldots, x_n,\ y_1, \ldots, y_n) \\
&= \alpha \times \left(\log\left(\sum_{i\in e} e^{x_i/\alpha}\right) + \log\left(\sum_{i\in e} e^{-x_i/\alpha}\right)\right. \\
&\qquad \left. + \log\left(\sum_{i\in e} e^{y_i/\alpha}\right) + \log\left(\sum_{i\in e} e^{-y_i/\alpha}\right)\right)
\end{aligned}
$$

Wirelength Function Comparisons



Error Rate Comparisons Between WA and LSE Functions

$$\varepsilon_{WA}(\mathbf{x}_e) \leq \varepsilon_{LSE}(\mathbf{x}_e) = \gamma \ln n$$



Density Constraint Smoothing

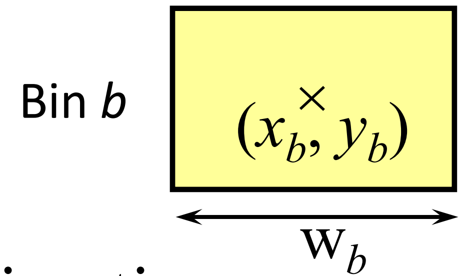
$$\text{Minimize} \sum_{e \in E} c_e \times \mathrm{WL}_e(\boldsymbol{x}, \boldsymbol{y})$$

$$\text{Subject to } D_b(\boldsymbol{x}, \boldsymbol{y}) <= T_b \text{ for all bin } b$$

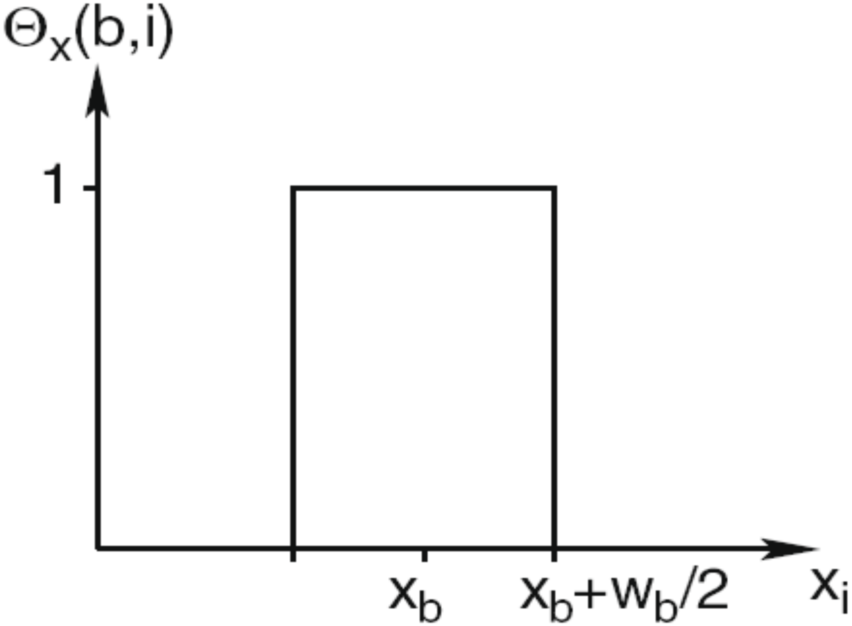
- $D_b(x,y)$ gives the density of bin b with respect to placement solution $\boldsymbol{x}$ and $\boldsymbol{y}$
- Need to be smoothed
- Smoothing techniques
  - Bell-shaped function
  - Sigmoid fucntion

Smoothing by Bell-Shaped Function

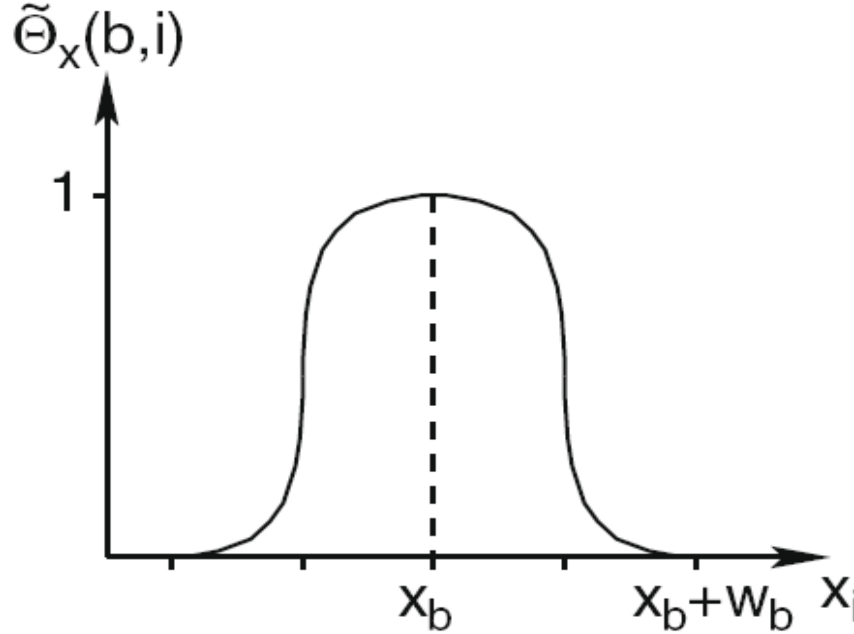
- Assume
  - Each module is much smaller than bins
  - Each module has a unit area



- Overlap between bin b and module i in x-direction:



Exact rectangle–shaped function



Smooth bell–shaped function

Equations for Bell-Shaped Function

- Let $d_x = |x_i - x_b|$

$$\tilde{\Theta}_x(b,i) = \begin{cases} 1 - 2 \times d_x^2/\omega_b^2 & \text{if } 0 \le d_x \le \omega_b/2 \\ 2 \times (d_x - \omega_b)^2/\omega_b^2 & \text{if } \omega_b/2 \le d_x \le \omega_b \\ 0 & \text{if } \omega_b \le d_x \end{cases}$$

$$D_b(x,\ y) = \sum_{i \in V} C_i \times \tilde{\Theta}_x(b,i) \times \tilde{\Theta}_y(b,i)$$

- Extension for large modules:

$$\tilde{\Theta}_x(b,i) = \begin{cases} 1 - a \times d_x^2 & \text{if } 0 \le d_x \le w_b/2 + w_i/2 \\ b \times (d_x - w_b - w_i/2)^2 & \text{if } w_b/2 + w_i/2 \le d_x \le w_b + w_i/2 \\ 0 & \text{if } w_b + w_i/2 \le d_x \end{cases}$$

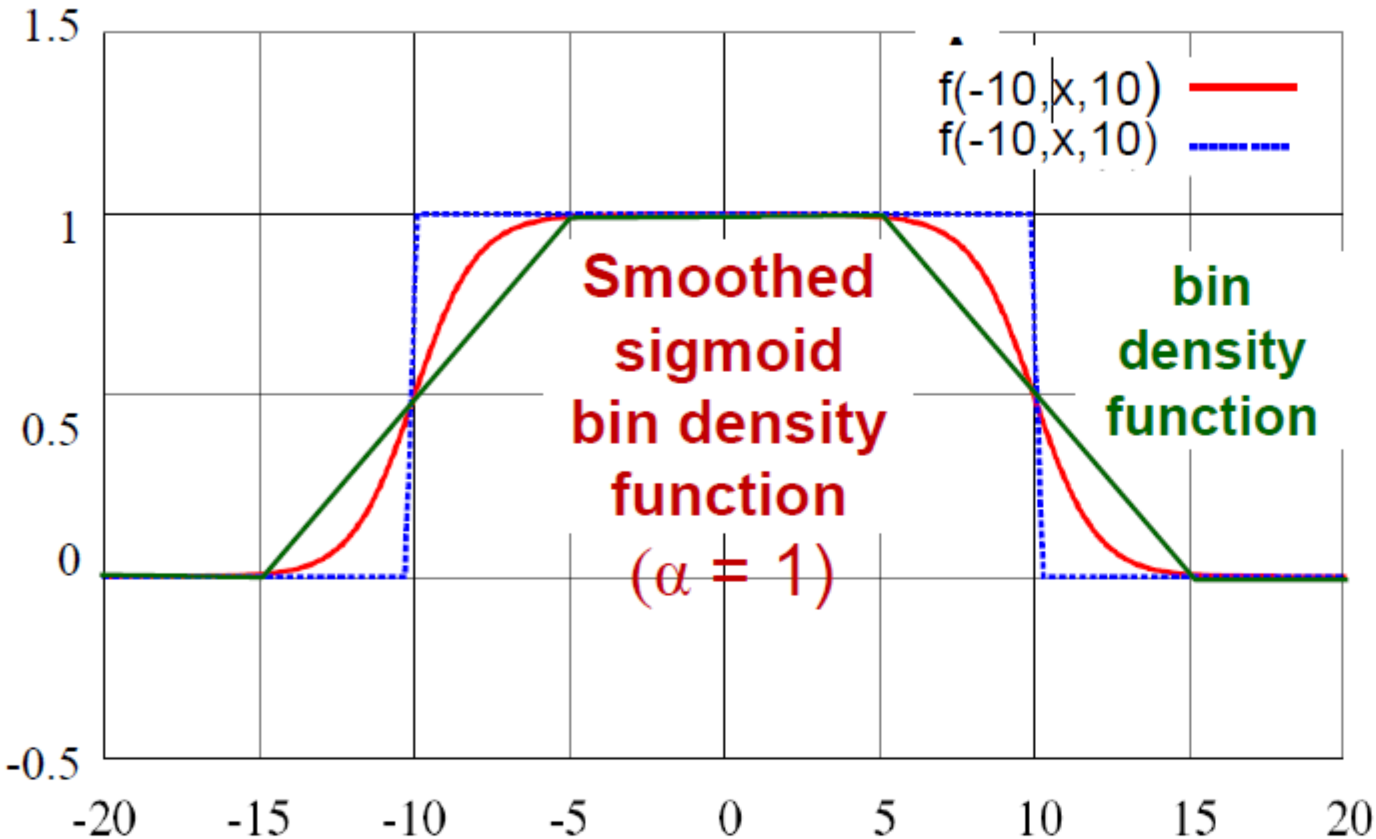
where $a = 4/((w_b + w_i)(2w_b + w_i))$

$b = 4/(w_b(2w_b + w_i))$

Smoothing by Sigmoid Function

[Hsu & Chang, ICCAD'11]

$$f(l,x,u)=\begin{cases}1, & \text{if } l<x<u\\ 0, & \text{otherwise}\end{cases} \quad \xrightarrow{\;p(t)=\frac{1}{1+e^{-\alpha t}}\;} \quad f(l,x,u)\cong p(x-l)\,p(u-x)$$



Algorithm for Nonlinear Programs

- **Quadratic penalty method**
  - Convert into a sequence of unconstrained minimization problems
  - Each unconstrained problem can be solved by the **conjugate gradient method**

$$\text{Minimize} \sum_{e \in E} c_e \times \mathrm{WL}_e(x,\ y) + \beta \times \sum_{b} (D_b(x,\ y) - T_b)^2$$

Extension to Multilevel Placement

- Three phases
  1. A hierarchy of coarser netlists is constructed
  2. An initial placement of the coarsest netlist is generated
  3. The netlist is successively unclustered, and placement at each level is refined

Clustering Technique: First Choice

- Traverse modules in an arbitrary order
- Each module *i* is clustered with an unclustered neighbor *j* with the largest affinity (or weight):

$$r_{ij} = \sum_{e \in E \wedge i,\, j \in e} \frac{c_e}{|e| - 1}$$

- To reduce variation in cluster size:

$$r_{ij} = \sum_{e \in E \wedge i,\, j \in e} \frac{c_e}{(|e| - 1) \times \text{area}(e)}$$

- To consider proximity information in initial placement:

$$r_{ij} = \sum_{e \in E \wedge i,\, j \in e} \frac{c_e}{(|e| - 1) \times \text{area}(e) \times \text{dist}(i,\ j)}$$

Clustering Technique: Best Choice

- Affinity: $r_{ij} = \sum_{e \in E \wedge i, j \in e} \frac{c_e}{|e| \times (\text{area}(i) \times \text{area}(j))}$
- Always select the globally best pair of modules for clustering (and, in principle, update netlist immediately)
- In practice, **lazy updating technique** is proposed
  - Affinities affected by previous clustering are marked as invalid and are updated only after they have been selected for clustering
- Impose a hard upper limit for cluster size

Legalization

- Legalization is a process to eliminate all overlaps (and assign cells to rows) by perturbing cells/modules as little as possible